

Automatisierte Generierung, Adaption und Rekonfiguration von Co-Simulationen für modulare Produktionsanlagen

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

von der KIT-Fakultät für Elektrotechnik und Informationstechnik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

M. Sc. Christian Härle

Tag der mündlichen Prüfung: 22. Dezember 2022

Hauptreferent: Prof. Dr.-Ing. Mike Barth

Korreferent: Prof. Dr.-Ing. Alexander Fay

Vorwort des Autors

Die vorliegende Arbeit entstand während meiner Zeit als wissenschaftlicher Mitarbeiter an der Fakultät für Technik der Hochschule Pforzheim und als externer Doktorand an der Professur für Automatisierungstechnik der Helmut-Schmidt-Universität / Universität der Bundeswehr Hamburg.

Herrn Professor Dr.-Ing. Mike Barth möchte ich herzlich für die Zeit an der Hochschule, die Ermöglichung des Forschungsprojekts mit OTEC und das in mich gesetzte Vertrauen danken. Sein fachliches Fordern und Fördern haben maßgeblich zu meiner persönlichen Weiterentwicklung und dem Gelingen dieser Arbeit beigetragen.

Mein besonderer Dank gilt Herrn Professor Dr.-Ing. Alexander Fay für die Förderung, die mir in Form von fachlichen Diskussionen, Ratschlägen und dem vermittelten methodischen Vorgehen zuteilwurde. Zudem möchte ich mich für die Übernahme des Zweitgutachtens bedanken.

Mein Dank gilt darüber hinaus dem Fakultätsrat der KIT-Fakultät für Elektrotechnik und Informationstechnik für die kurzfristige Übernahme meines Promotionsgesuches sowie den weiteren Mitgliedern des Prüfungsausschusses, Herrn Professor Dr.-Ing. Marc Hiller, Herrn Professor Dr.-Ing. Michael Heizmann und Herrn Professor Dr. rer. nat. Ulrich Lemmer.

Seitens der Firma OTEC Präzisionsfinish GmbH möchte ich mich für die Möglichkeit der Durchführung des Forschungsprojekts bedanken. Zum einen konnte ich in diesem Rahmen wertvolle Industrieerfahrung sammeln, zum anderen ist hieraus das erste Validierungsszenario für diese Arbeit entstanden. Für die Unterstützung und die kollegiale Zusammenarbeit gilt mein Dank den Herren Helmut Gegenheimer, Daniel Stelzer, Gregor Hirling, Tim Gegenheimer und Nico Gegenheimer.

Seitens der Firma C. Jentner GmbH möchte ich den Herren Chris Jentner und Marcel Scheidig für die Unterstützung und die Bereitschaft, ihre Anlage als zweites Validierungsszenario nutzen zu dürfen, danken.

Des Weiteren möchte ich meinen ehemaligen Kollegen danken, die meine Zeit an der Hochschule Pforzheim zu einer gemacht haben, an die ich mich gerne zurückerinnere. Besonders hervorheben möchte ich Herrn Max Barchet, Herrn Christoph Ußfeller und Herrn Pankaj Kolhe, die mir bei Kaffeepausen mit fachlichen Diskussionen und Ratschlägen zur Seite standen. Herrn Max Barchet möchte ich zudem für das Korrekturlesen meiner Arbeit danken.

Meinen Doktoranden-Kollegen Herrn Gary Hildebrandt, Herrn Leif-Thore Reiche und Herrn Marcel Lewke möchte ich ebenfalls für den freundschaftlichen Umgang und die Bereitschaft die vorliegende Arbeit Korrektur zu lesen danken. Zudem wünsche ich ihnen gutes Gelingen bei der Fertigstellung ihrer eigenen Arbeiten.

Meinen Eltern Martin und Susanne möchte ich für das Vertrauen, die bedingungslose Unterstützung und den Rückhalt danken. Ganz besonders möchte ich meiner Frau Tamara für die liebevolle Unterstützung, den Verzicht und das aufgebrachte Verständnis danken.

Inhaltsverzeichnis

INHALTSVERZEICHNIS	I
BEGRIFFSVERZEICHNIS	IV
VERZEICHNIS DER ABKÜRZUNGEN	V
VERZEICHNIS VERWENDETER FORMELZEICHEN	VI
KURZFASSUNG	VIII
ABSTRACT	IX
1 EINLEITUNG	1
1.1 Motivation und Problemstellung.....	1
1.2 Forschungsmethodik	3
1.3 Abgeleitete Forschungsfragen und Zielsetzung.....	3
1.4 Aufbau der Arbeit	5
2 MODELLIERUNG UND SIMULATION IM MODULAREN MASCHINEN- UND ANLAGENBAU	6
2.1 Grundlagen und Begriffe im modularen Maschinen- und Anlagenbau.....	6
2.1.1 Heterogenität im Maschinen- und Anlagenbau.....	6
2.1.2 Markt- und produktbedingte Anforderungen an Produktionsanlagen.....	7
2.1.3 Modularisierung der Produktionsanlagen	9
2.1.4 Engineering im Maschinen- und Anlagenbau	11
2.1.5 Model-Based Systems Engineering.....	13
2.2 Simulation und ihre Anwendungen im Maschinen- und Anlagenbau	21
2.2.1 Herausforderungen beim Einsatz von Simulation	22
2.2.2 Bedarf eines neuen Modellierungs- und Simulationsworkflows	26
2.2.3 Assistenzsystem im Engineering	27
2.2.4 Abgeleitete Anforderungen an ein Assistenzsystem	28
2.2.5 Definition des Automatisierungslevels des Assistenzsystems	29
2.3 Betrachtete Systeme und Validierungsszenarien	33
2.3.1 Aufbau und Wirkweise der OTEC Gleitschleifanlage	33
2.3.2 Szenario 1: Wirkende Kräfte auf Werkstücke bei Gleitschleifprozess	35
2.3.3 Aufbau und Wirkweise der Jentner Kleingalvanikanlage.....	36
2.3.4 Szenario 2: Mehrstufige Oberflächenbeschichtung in der Galvanotechnik.....	38
2.4 Zusammenfassung der Erkenntnisse	40
3 STAND VON WISSENSCHAFT UND TECHNIK	41
3.1 Bestehende Konzepte zur (teil-)automatischen Modellgenerierung, -adaption und -rekonfiguration.....	41
3.1.1 Konzepte zur (teil-)automatischen Modellgenerierung.....	41
3.1.2 Konzepte zur (teil-)automatischen Modelladaption.....	44
3.1.3 Konzepte zur (teil-)automatischen Modellrekonfiguration	49
3.1.4 Phasenübergreifende Lösungskonzepte	51
3.2 Bewertung bestehender Konzepte hinsichtlich definierter Anforderungen	53
3.3 Erkenntnisse und abgeleiteter Handlungsbedarf	55
3.4 Gesamtkonzeptübersicht	57
4 KONZEPTBAUSTEIN FÜR DIE MODELLGENERIERUNG UND -KONFIGURATION	59
4.1 Teilziele, Randbedingungen und Annahmen	59
4.2 Technologische Konzeptbausteine	60
4.2.1 Beschreibungsmittel von Produktionsanlagen	60
4.2.2 Co-Simulation im Model-Based Systems Engineering	62
4.3 Konzept zur automatischen Generierung von modulbasierten Systemsimulationen	68
4.4 Prototypische Implementierung des Assistenzsystems	70
4.4.1 Struktur und Konventionen des Topologiemodells	70

4.4.2	Aus Topologiemodell abgeleitetes Simulationsskript	71
4.4.3	Automatische Ausführung eines Simulationsskriptes.....	72
4.5	Teilkonzeptvalidierung	73
4.5.1	Validierungsszenario 1: Modellgenerierung der OTEC Schleifanlage	73
4.5.2	Validierungsszenario 2: Modellgenerierung der Jentner Galvanikanlage.....	74
4.6	Zwischenfazit	76
5	KONZEPTBAUSTEIN FÜR DIE MODELLADAPTION	78
5.1	Teilziele, Randbedingungen und Annahmen	78
5.2	Technologische Konzeptbausteine	79
5.2.1	Analyse von Ursachenclustern für notwendige Modelladaptionen	79
5.2.2	Aufbereitung der Mess- und Simulationsdaten.....	80
5.2.3	Struktureller Ablauf eines Adaptionprozesses	82
5.2.4	Detaillierungsgrad von Modellen	83
5.3	Konzept zur betriebsparallelen Adaption von Co-Simulationen	85
5.3.1	Konzeptskizze und Möglichkeiten der Adaption.....	85
5.3.2	Formulierung der Adaptionaufgabe.....	86
5.3.3	Algorithmen-Wahl für Pareto-Optimierung.....	89
5.3.4	Grundlagen evolutionärer Algorithmen	91
5.4	Prototypische Implementierung.....	93
5.4.1	Auswahl und Einbindung einer geeigneten Algorithmen-Bibliothek	93
5.4.2	Problemformulierung, Algorithmen-Setup und Analysefunktionen	93
5.4.3	Umsetzung des Adaptionablaufs.....	100
5.4.4	Anpassung des Simulationsskriptes im Rahmen der Adaption.....	103
5.5	Teilkonzeptvalidierung Teil I – OTEC Anlage.....	105
5.5.1	Betrachtetes Validierungsszenario	105
5.5.2	Physikmodelle, Parameter und Detaillierungsgrade.....	106
5.5.3	Ergebnisse der Adaption und Vergleich verschiedener Algorithmen	109
5.6	Teilkonzeptvalidierung Teil II – Jentner Anlage	111
5.6.1	Betrachtetes Validierungsszenario	111
5.6.2	Physikmodelle, Parameter und Detaillierungsgrade.....	113
5.6.3	Erkenntnisse aus Modellvalidierung.....	116
5.7	Zwischenfazit	117
6	KONZEPTBAUSTEIN FÜR DIE MODELLREKONFIGURATION	120
6.1	Teilziele, Randbedingungen und Annahmen	120
6.2	Technologische Konzeptbausteine	121
6.2.1	Durch Sensorik erfassbare Informationen	121
6.2.2	Automatische Bearbeitung des Topologiemodells	121
6.3	Konzept zur bedarfsorientierte Rekonfiguration von Co-Simulationen	122
6.3.1	Erweiterte Konzeptskizze und Ablauf der bedarfsorientierten Rekonfiguration	122
6.3.2	Detektion von Änderungen der Anlagentopologie	123
6.3.3	Rekonfiguration des Gesamtsystemmodells	124
6.3.4	Dokumentation der Änderungen.....	124
6.4	Prototypische Implementierung.....	125
6.5	Teilkonzeptvalidierung	126
6.5.1	Validierungsszenario 1: Werkstück- und prozessabhängiger Umbau bei Gleitschleifanlagen.....	126
6.5.2	Validierungsszenario 2: Prozessbedingte Elektrolytanpassung in der Galvanotechnik	127
6.6	Zwischenfazit	128
7	ZUSAMMENFASSUNG UND AUSBLICK	131
7.1	Zusammenfassung.....	131
7.2	Ausblick.....	132

ANHANG	134
Anhang A AutomationML-Modell der Gleitschleifanlage.....	134
Anhang B AutomationML-Modell der Galvanikanlage.....	142
Anhang C Listings der CustomProblem-Klasse des Assistenzsystems	143
LITERATURVERZEICHNIS	147
REFERENZIERTE NORMEN, RICHTLINIEN UND EMPFEHLUNGEN	162
VERZEICHNIS REFERENZierter INTERNETQUELLEN.....	163
VERÖFFENTLICHUNGEN DES VERFASSERS	164
STUDENTISCHE ARBEITEN	165

Begriffsverzeichnis

Adaption	Anpassung von Modellen an veränderte Bedingungen ohne vollständigen Neuaufbau [BER13].
Assistenzsystem	Technisches System, welches zur Assistenz bzw. Unterstützung des Nutzers in bestimmten Situationen oder bei bestimmten Handlungen dient [VBT17].
Black-Box-Modell	Modell, in dem eine Übergangsfunktion aufgrund der Einwirkungen auf das System (Input) die Rückwirkungen des Systems auf die Umgebung (Output) bestimmt [VDI 3633-1].
Co-Simulation	Kopplung sowohl von mehreren Modellierungsumgebungen als auch mehreren Solvern [SCH15].
Engineering	Umfasst die Planung, Realisierung und Inbetriebnahme einer Anlage. Auch spätere Ingenieurstätigkeiten, die während des Betriebs durchzuführen sind, sind dem Engineering zuzuordnen [FAY09].
Mehrkriterielles Optimierungsproblem	Optimierungsproblem mit mehreren, für gewöhnlich im Konflikt zueinanderstehenden Zielen [VAN99].
Modul	Einheit, welche funktional vollständig spezifiziert und sowohl energetisch und stofflich als auch automatisierungstechnisch integriert wird. Die vorgegebenen Funktionen werden in einen Gesamtverbund aus Modulen eingebunden. [NAMUR NE 148]
Topologiemodell	Beschreibungsmittel zur Beschreibung einer Anlage als hierarchische Struktur von Anlagenobjekten und zur Abbildung, wie diese untereinander verbunden sind. [DLP+08]

Verzeichnis der Abkürzungen

AML	AutomationML
AS	Assistenzsystem
AAS	Ausführungsassistenzsystem
EAS	Entscheidungsassistenzsystem
WAS	Wahrnehmungsassistenzsystem
EI	ExternalInterface
IE	InternalElement
IND	Individuum
FMI	Functional Mock-up Interface
FMU	Functional Mock-up Unit
GUI	Grafische Benutzeroberfläche (engl. graphical user interface)
HLA	High Level Architecture
LoD	Level of Detail
MD	Merkmals-Diagramm
MKOP	Mehrkriterielles Optimierungsproblem
M&S	Modellierung und Simulation
MTO	Metal Turn Over
OMS	OpenModelicaSimulator
SAE	Society of Automotive Engineers
TM	Topologiemodell
WS	Werkstück
XML	Extensible Markup Language

Verzeichnis verwendeter Formelzeichen

AIC	Akaike information criterion
\underline{d}_{ges}	Vektor der Detaillierungsgrade aller Modulmodelle
f, F	Güte-/ Zielfunktion(en)
\underline{f}_{res}	Resultierende Kraft auf das Werkstück
g_j	Ungleichungen, die den Lösungsraum des Optimierungsproblems einschränken
h_l	Gleichungen, die den Lösungsraum des Optimierungsproblems einschränken
k	Anzahl an Mess- und Simulationspunkten
L	Likelihood-Funktion
LoD	Detaillierungsgrad eines Modulmodells
N	Anzahl der Residuen
\underline{p}_{ges}	Vektor der Parameter aller Modulmodelle
t_i	Kommunikationspunkt einer Co-Simulation
$u(t)$	Eingangsvariablen einer FMU
v	Anzahl an Sensorpaaren (Sensoren, die sowohl in der realen Anlage als auch in der Simulation enthalten sind)
\underline{x}	Vektor der Designvariablen
\underline{x}^*	Lösungsvektor des Optimierungsproblems
$x(t)$	Zustandsvariablen einer FMU
\underline{y}_{mess}	Vektor mit Messdaten

\underline{y}_{sim}	Vektor mit Simulationsdaten
$y(t)$	Ausgangsvariablen einer FMU
z_i	Residuen aus dem nichtlinearen Least-Squares-Fit

Kurzfassung

Simulation nimmt eine zunehmend wichtige Rolle im Engineering und Betrieb von Maschinen und Anlagen ein. Simulation kann genutzt werden, um Experimente an noch nicht existierenden oder zur Verfügung stehenden Anlagen durchzuführen, um unterschiedliche Systemkonfigurationen zu testen oder um Informationen über nicht beobachtbare interne Vorgänge zu sammeln. Um volatilen Anforderungen und Randbedingungen (z. B. durch Kunden, neue Normen, Gesetze und Richtlinien, Nachhaltigkeit und Ressourceneffizienz) gerecht zu werden und dabei trotz allem die Rentabilität der Produktionsanlagen sicherzustellen, werden diese Anlagen zunehmend in modularer Bauweise konzipiert. Durch die Wandelbarkeit der Anlage nehmen die ohnehin umfangreichen manuellen Abläufe, die mit deren Modellierung und Simulation einhergehen, noch weiter zu. Diese Modellierungstätigkeiten sind fehleranfällig, kostenintensiv und erstrecken sich über den kompletten Lebenszyklus der Produktionsanlage. Folglich wird eine Lösungsmethode benötigt, welche es Ingenieuren¹ ermöglicht, Simulation unkompliziert, schnell und kosteneffizient über den kompletten Lebenszyklus einer modularen Produktionsanlage einsetzen zu können. Ziel der vorliegenden Arbeit ist es, ein Lösungskonzept für die (teil-)automatische Generierung, Adaption und (Re-)Konfiguration von Simulationsmodellen für modulare Produktionsmaschinen und -anlagen zu entwickeln. Hierzu werden einerseits die Voraussetzungen analysiert, die erfüllt werden müssen, um eine solche (Teil-)Automatisierung zu erreichen. Des Weiteren wird erläutert, welche Informationen benötigt werden und welcher Automatisierungsgrad für die Modellgenerierung und -adaption erzielt werden kann. Aus dem formulierten Handlungsbedarf werden insgesamt neun Anforderungen, die verschiedene Lebenszyklusphasen adressieren, abgeleitet. Die umfassende Analyse des Stands von Wissenschaft und Technik hat ergeben, dass keiner der bestehenden Lösungsansätze alle entwickelten Anforderungen vollumfänglich erfüllt. Der Hauptgrund hierfür ist, dass keine der bestehenden Lösungen mit dem Ziel entworfen wurde, den kompletten Lebenszyklus der Anlage abzudecken. Aus diesem Defizit ergab sich der Handlungsbedarf, ein neues, lebenszyklusübergreifendes Lösungskonzept zu entwickeln. Bei dem in dieser Arbeit entwickelten Gesamtkonzept handelt es sich, entsprechend der Lebenszyklusphasen Inbetriebnahme, Betrieb und Umbau, um ein dreistufiges Konzept. Die jeweiligen Konzeptstufen bauen dabei auf den vorherigen auf. Zur Überprüfung des vorgestellten Gesamtkonzepts hinsichtlich der definierten Anforderungen wurde ein prototypisch implementiertes Assistenzsystem auf zwei modulare Produktionsanlagen aus unterschiedlichen Domänen des Maschinen- und Anlagenbaus angewandt. Die Validierung hat gezeigt, dass das Assistenzsystem für beide Systeme über deren kompletten Lebenszyklus angewandt werden kann und dadurch der manuelle Modellierungsaufwand sowie die starke Abhängigkeit vom individuellen Wissen der Modellierungsexperten reduziert wird. Das Assistenzsystem, welches eine Implementierung des Lösungskonzepts darstellt, erfüllt alle definierten Anforderungen. Weiterhin konnten, im Rahmen der Kapitel zu den jeweiligen Teilkonzepten, die aufgestellten Forschungsfragen bezüglich der Voraussetzungen, der benötigten Informationen und des erreichten Automatisierungsgrades beantwortet werden. Den Abschluss der Arbeit bildet eine kritische Würdigung, in der die offenen Punkte des entwickelten Konzepts als Basis für Folgearbeiten erläutert werden.

¹ Aus Gründen der besseren Lesbarkeit wird auf die zusätzliche Formulierung der weiblichen Form verzichtet. Der Autor möchte darauf hinweisen, dass in der vorliegenden Arbeit sämtliche Personenbezeichnungen gleichermaßen für alle Geschlechter gelten.

Abstract

Simulation plays an increasingly important role in the engineering and operation of machines and plants. Simulation can be used to conduct experiments on plants that do not yet exist or are not yet available, to test different system configurations or to obtain information about internal processes that cannot be observed. To meet volatile requirements and boundary conditions (e. g., due to customers, new standards, laws and directives, sustainability, and resource efficiency) while still ensuring the profitability of production plants, they are increasingly designed in a modular way. Due to the changeability of the plant, the already extensive manual processes involved in its modeling and simulation increase even further. These modeling activities are error-prone, cost-intensive and extend over the complete life cycle of the production plant. Consequently, a solution method is needed that enables engineers to use simulation in an uncomplicated, fast and cost-efficient way over the complete life cycle of a modular production plant. The aim of this thesis is to develop a solution concept for the (partially) automatic generation, adaptation and (re-)configuration of simulation models for modular production machines and plants. For this purpose, the prerequisites that must be fulfilled in order to achieve such a (partial) automation are analyzed. Furthermore, it is explained which information is required and which degree of automation can be achieved for model generation and adaptation. A total of nine requirements addressing different lifecycle phases are derived from the formulated need for action. The extensive analysis of the state of the art in science and technology has shown that none of the existing solutions fully satisfies all the requirements developed. The main reason for this is that none of the existing solutions was designed with the aim of covering the complete life cycle of the plant. This deficit resulted in the need for action to develop a new cross-lifecycle solution concept. The overall concept developed in this work is a three-stage concept corresponding to the life cycle phases of commissioning, operation and modification. The respective concept stages build on the previous ones. To validate the presented overall concept with respect to the defined requirements, a prototypically implemented assistance system was applied to two modular production plants from different domains of mechanical and plant engineering. The validation showed that the assistance system can be applied to both systems over their complete life cycle, thus reducing the manual modeling effort and the strong dependency on the individual knowledge of the modeling experts. The assistance system, which represents an implementation of the solution concept, fulfills all defined requirements. Furthermore, in the context of the chapters on the respective sub-concepts, the research questions posed regarding the prerequisites, the required information and the degree of automation achieved could be answered. The thesis concludes with a critical appraisal in which the open points of the developed concept are explained as a basis for follow-up work.

1 Einleitung

1.1 Motivation und Problemstellung

Industrielle Investitionsgüter, wie beispielsweise automatisierte Produktionsanlagen, werden für einen Nutzungszeitraum von mehreren Jahren oder sogar Jahrzehnten entwickelt [MHW+18]. Dagegen führen globale Trends, wie individuelle und wechselnde Kundenanforderungen, kürzere Produktlebenszyklen [MEI19, VW12] und variable Produktportfolios zu kurzzyklisch notwendigen Änderungen in den Produktions-Anforderungen und -Randbedingungen [GRI14, BBH+19, LLP+17]. Um diesen Veränderungen Rechnung zu tragen und während der gesamten Betriebsphase funktionsfähig zu bleiben, müssen Produktionsanlagen flexibel und wandelbar sein [MHW+18]. Diesen Herausforderungen begegnet der Maschinen- und Anlagenbau zunehmend durch den Einsatz von modularen Systemlösungen [PI16, DIS19]. Durch die Aggregation vordefinierter Anlagenmodule sollen Produktionsanlagen nach einem Plug-and-Work-Ansatz [OVN18] aufgebaut werden können. So soll auch bei schwankendem Produktmengenbedarf ein schnelles "scale-up" und "scale-down" der Produktionsanlage realisiert werden können [BMW17]. Unter Modulen werden im Rahmen dieses Beitrages, entsprechend der Definition in [VDI/VDE/ NAMUR 2658-1] unter Verweis auf [NAMUR NE 148], Einheiten verstanden, welche *„funktional vollständig spezifiziert und sowohl energetisch und stofflich als auch automatisierungstechnisch integriert [werden]. Die vorgegebenen Funktionen werden [...] in einen Gesamtverbund aus Modulen eingebunden.“* In dieser Arbeit wird vornehmlich die diskrete Fertigungstechnik betrachtet.

Neben dem Vorteil der Adaptierbarkeit bringt der modulare Aufbau von Produktionsanlagen aber auch Herausforderungen mit sich [ZAM19]. Aus Sicht des modellbasierten Engineerings – insbesondere der Themen Modellierung und Simulation – stellt eine sich regelmäßig ändernde Systemkonfiguration neue Herausforderungen dar [LAD18]. Dies ist problematisch, da in [GMA20] Simulationen als *„integraler Bestandteil im gesamten Lebenszyklus, von der Planung über Design und Implementierung bis zum Betrieb und Service“* benannt werden, welche *„bestehende Wertschöpfungsketten [erweitern], neue Wertschöpfungsnetzwerke [schaffen] und neue Geschäftsmodelle [ermöglichen]“*. Zu den Einsatzzwecken der Simulation in den einzelnen Lebenszyklusphasen gehören unter anderem die Verwendung als Entwicklungswerkzeug für die Auslegung von Komponenten, vorgezogene Überprüfungen von Spezifikationen im Rahmen einer virtuellen Inbetriebnahme, wenn noch keine reale Anlage zur Verfügung steht, oder die Durchführung von Parameterstudien, um einen Produktionsprozess hinsichtlich des Energieverbrauchs oder der Taktzeit zu optimieren. Ein Beispiel für die neuen Herausforderungen ist der effiziente Einsatz von Simulationsmodellen im Engineering und Betrieb modularer Anlagen sowie während eines Anlagenumbaus. Um der Wandelbarkeit und Flexibilität der realen Produktionsanlage zu entsprechen, muss deren modulares Konzept auf die Simulationsmodelle übertragen werden. Ziel dabei ist es, dass die bereitgestellten Modulmodelle, entsprechend dem Aufbau der realen Produktionsanlage, gekoppelt werden können. Kopplung bezeichnet die Zuweisung von Ein- und Ausgängen der Modulmodelle [GTB+17]. Die Modulmodelle unterscheiden sich hinsichtlich verschiedener Aspekte, wie Herkunft, physikalischer Domäne, in der sie angesiedelt sind, oder dem Detaillierungsgrad der Modellierung [GMD+19]. Für diese Heterogenität gibt es bereits Lösungen in Form von standardisierten Schnittstellen. Die Modelle korrekt zu koppeln und

im Hinblick auf deren Modellierungstiefe zu konfigurieren, ist – trotz der standardisierten Schnittstellen – nach wie vor mit erheblichem manuellem Aufwand verbunden.

Wie angedeutet endet der Modellierungsaufwand dabei nicht nach der Inbetriebnahmephase der Produktionsanlage. Neue Modulsimulationsmodelle (MSM) müssen, entsprechend ihrer realen Gegenstücke, in eine existierende Simulation integriert, bestehende MSM ersetzt und Modellparameter angepasst werden. Sind mehrere MSM, beispielsweise von einem Antrieb mit und ohne Betrachtung thermischer Einflüsse oder mit unterschiedlich detaillierten Reibungsmodellen, vorhanden, gilt es, für verschiedene Simulationsszenarien den jeweils geeigneten Detaillierungsgrad zu wählen [PUN18]. So kann, je nach Bedarf, das reale System hinreichend genau abgebildet und der Rechenaufwand minimal gehalten werden. In [PUN18] wird konkretisierend gefordert, *„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“*. Die Anlagenteile, bei denen ein detaillierteres Modell erforderlich ist, gilt es möglichst automatisiert zu identifizieren.

Zudem ist anzunehmen, dass verschiedene Disziplinen und Stakeholder unterschiedliche Simulationsergebnisse benötigen. Denkbar sind beispielsweise zwei Simulationen einer Anlage, die erste zur Validierung der Elektroplanung und die zweite zur Überprüfung des Steuerungscode auf Kollisionsfahrten. Für die zweite Simulation ist ein Kinematikmodell nötig, mittels welchem die Fahrwege der mechanischen Komponenten berechnet und Kollisionen erkannt werden können. Für die Validierung der Elektroplanung reicht ein vereinfachtes Verhaltensmodell der Anlage, um die korrekte Verschaltung zu überprüfen. Der manuelle Aufwand für Aufbau und Pflege der Simulationen ist bislang zeit- und kostenintensiv sowie fehleranfällig [RM10]. Zudem beschränkt sich, wie in den Lebenszyklusphasen beschrieben, der Einsatz von Simulation nicht mehr ausschließlich auf das Engineering. Auch parallel zum Betrieb kann Simulation gewinnbringend eingesetzt werden [SCA+17, SDL12]. So eröffnet der kontinuierliche Vergleich von Simulationsergebnissen und Anlagenmesswerten neue Diagnosemöglichkeiten und kann zu einem robusten Anlagenbetrieb beitragen. Beispielhaft können bei einem Sensorausfall, wie z. B. einer defekten Lichtschranke, Systemzustände durch das Modell geschätzt werden [SCA+17]. Neben diesen betriebsparallelen Einsatzmöglichkeiten kann auch eine Voraussimulation für Prognose- und Optimierungszwecke, beispielsweise um Prozessparameter zu optimieren [SDL12], durchgeführt werden. Verschleiß an Anlagenkomponenten durch den Betrieb, kann es zu einem Auseinanderdriften der Mess- und Simulationsergebnisse kommen [ZAS+18]. Damit die Simulation weiterhin nutzbare Simulationsergebnisse liefert, ist eine Adaption der Modelle parallel zum Betrieb nötig. Unter Adaption wird im Rahmen dieser Arbeit entsprechend [BER13] *„die Anpassung von Modellen an veränderte Bedingungen ohne vollständigen Neuaufbau“* verstanden.

In [GMA20] wird die Vision formuliert, dass zukünftig *„Simulationsmodelle [...] so weit als möglich automatisch generiert, kombiniert oder adaptiert“* werden. Im Rahmen einer Analyse des Standes von Wissenschaft und Technik im Bereich der automatischen Modellgenerierung, -adaption und -rekonfiguration, welche in Abschnitt 3.2 detailliert vorgestellt wird, konnte der Bedarf für eine lebenszyklusübergreifende Lösung identifiziert werden. Die Ausarbeitung und Vorstellung einer Lösung für diesen Bedarf stellt den Kern dieser Arbeit dar.

1.2 Forschungsmethodik

Zunächst soll die im Rahmen dieser Arbeit angewandte Forschungsmethodik vorgestellt werden, welche die Grundlage für die Struktur dieser Arbeit liefert. Die angewandte Methodik ist dem Bereich *Design Science Research* nach Hevner et al. [HMJ+04] zuzuordnen. Untersuchungsgegenstand in Design Science ist zum einen die (Weiter-)Entwicklung von Artefakten und zum anderen die Untersuchung dieser Artefakte in ihrem Kontext. Der abstrakte Artefakt-Begriff kann in diesem Zusammenhang beispielsweise für eine Methode, ein Modell oder ein Software-Werkzeug stehen. In Bezug auf die vorliegende Arbeit könnte es sich bei dem Artefakt z. B. um eine Methode handeln, die u. a. den Einsatz von Modellierung und Simulation im Kontext des modularen Maschinen- und Anlagenbaus erleichtert. Das Artefakt dieser Arbeit ist im weiteren Verlauf noch zu spezifizieren. Unter dem Kontext, in den sich das Artefakt einordnet, sind beispielsweise bestehende Strukturen, Organisationen, Methoden und Normen zu verstehen. Im Rahmen dieser Arbeit kann man als Kontext z. B. bestehende Engineering-Abläufe im Bereich Modellierung und Simulation, das Personal, das an den Engineering-Abläufen beteiligt ist, sowie Firmen, die modulare Maschinen und Anlagen betreiben, aufführen. Es besteht eine Wechselwirkung zwischen dem Artefakt und seinem Kontext. Artefakte werden mit dem Ziel entwickelt, eine Verbesserung in ihrem jeweiligen Kontext zu bewirken. Im konkreten Fall bedeutet eine Verbesserung im Rahmen dieser Arbeit z. B., dass der Modellierungsaufwand durch die Methode reduziert oder dass applikationsspezifisch automatisch die geeigneten Detaillierungsgrade der Modelle durch ein Software-Werkzeug gewählt werden. Ein Design-Problem, im Rahmen dessen ein Artefakt entwickelt wird, setzt sich in Anlehnung an Wieringa [WIE14] aus vier Elementen zusammen. Hierzu gehören neben dem *Artefakt* die *Anforderungen*, die dieses erfüllen soll, die *Stakeholder-Ziele*, die erreicht werden sollen, sowie der *Kontext*, in dem das Artefakt eingesetzt werden soll. Als Beispiel für unterschiedliche Stakeholder-Ziele können in dem betrachteten Kontext unterschiedliche Einsatzfälle der Simulation angeführt werden. So können für dieselbe Anlage unterschiedliche Abteilungen oder Personengruppen unterschiedliche Simulationsergebnisse und daher auch unterschiedliche Modelle benötigen (z. B. Prozessplanung und Prozessoptimierung). Als Anforderung könnte von diesen Stakeholder-Zielen beispielsweise abgeleitet werden, dass ein hierfür entwickeltes Software-Werkzeug in der Lage sein muss, die Kopplung der Modulmodelle automatisch vorzunehmen, um den Modellierungsprozess zu vereinfachen. Das Design-Problem wird im Rahmen des Design-Zyklus iterativ bearbeitet und gelöst. Abschnitt 1.4 zeigt auf, in welchen Kapiteln die einzelnen Elemente aufgegriffen werden. Im nächsten Abschnitt wird das Ziel der Arbeit formuliert, und Forschungsfragen werden abgeleitet. Die Forschungsfragen dienen der Einschätzung, ob und inwieweit das Ziel der Arbeit erfüllt ist.

1.3 Abgeleitete Forschungsfragen und Zielsetzung

Wie in Abschnitt 1.1 aufgezeigt, wird ein Lösungskonzept benötigt, welches es Ingenieuren ermöglicht, Simulation unkompliziert, schnell und kosteneffizient über den kompletten Lebenszyklus einer Produktionsanlage einsetzen zu können. Dies trägt dazu bei, dass für die Ingenieure, trotz steigender Komplexität der betrachteten Systeme und wechselnder Anforderungen, Modellierung und Simulation (M&S) beherrschbar bleiben. Aus Unternehmenssicht stellt dies einen Baustein dar, um den aktuellen Marktanforderungen und den beschriebenen Herausforderungen zu begegnen [BS20a]. Hierzu müssen aus bestehenden Artefakten, die während des Engineerings und dem Betrieb anfallen, die nötigen Informationen abgeleitet werden, um die Modellierung zu automatisieren. Durch den Einsatz von Automatisierung kann

der manuelle Modellierungsaufwand sowie die starke Abhängigkeit vom individuellen Wissen der Modellierungsexperten reduziert werden [BS20a, SC20].

Von den aufgezeigten Herausforderungen

- Modellierung ist fehleranfällig, zeit- und kostenintensiv
- Modellierungsaufwand erstreckt sich über alle Lebenszyklusphasen einer Anlage und steigt entlang der Lieferkette
- Unterschiedliche Stakeholder benötigen unterschiedliche Simulationsergebnisse
- Modellierung setzt domänenübergreifendes Expertenwissen voraus
- Aus Unternehmenssicht geht mit Talentabwanderung dieser Experten der Verlust von spezifischem Wissen über M&S einher

lassen sich folgende Forschungsfragen ableiten, welche im Rahmen der vorliegenden Arbeit beantwortet werden sollen:

- 1. Ist die (teil-)automatische Generierung, Adaption und (Re-)Konfiguration von Modellen für modulare Produktionsmaschinen und -anlagen realisierbar?**
- 2. Welche Voraussetzungen müssen erfüllt werden, um diese (Teil-)Automatisierung zu erreichen?**
- 3. Welche Informationen werden für die (Teil-)Automatisierung der einzelnen Teilschritte benötigt?**
- 4. Welcher Automatisierungsgrad kann für die einzelnen Lebenszyklusphasen erzielt werden?**

Die Zielsetzung dieser Arbeit ist es – entsprechend der ersten Forschungsfrage – ein Konzept zu erarbeiten, das die (teil-)automatische Generierung, Adaption und (Re-)Konfiguration von Modellen für modulare Produktionsmaschinen und -anlagen ermöglicht. Die Ergebnisse dieser Arbeit sollen dazu beitragen, zukünftig den Aufwand, der mit Modellierung im Bereich modularer Produktionsmaschinen und -anlagen einhergeht, zu reduzieren. Dies ermöglicht z. B. eine einfache Generierung von Simulationsmodellen für verschiedene Szenarien und Stakeholder und reduziert den, entlang der Lieferkette steigenden, Modellierungsaufwand. Zudem ist eine Reduktion der benötigten Expertise für den Einsatz von M&S zu erwarten. Weiter stellen die Ergebnisse einen potenziellen Befähiger für KMUs dar, die bisher z. B. aufgrund von Personal-/ Expertenmangel auf den Einsatz von M&S verzichten müssen. Auch für aufkommende digitale Austauschplattformen wie z. B. den ISG TwinStore [Twi22[@]], bei dem es sich um eine Onlineplattform für Komponentenmodelle zur virtuellen Inbetriebnahme (VIBN) handelt, kann das erarbeitete Konzept eine sinnvolle Ergänzung sein. So könnten beispielsweise in der Cloud erste Gesamtmodelle aus einzelnen Komponentenmodellen generiert werden, um simulativ zu testen, ob Komponenten im Zusammenspiel gewünschte Anforderungen erfüllen.

Um die Forschungsfragen sinnvoll beantworten zu können, wird zunächst der aktuelle Stand von Wissenschaft und Technik untersucht. Hierzu gehört zunächst eine Analyse des Modellierungs- und Simulationsprozesses im Maschinen- und Anlagenbau. Anschließend wird aufgezeigt, welche Herausforderungen im Zusammenhang mit M&S allgemein bestehen und welche sich zusätzlich durch die Modularisierung einer Anlage ergeben. Hieraus werden Anforderungen abgeleitet, die ein zukünftiger Modellierungs- und Simulationsworkflow erfüllen muss. Auf Basis dieser Anforderungen werden bestehende Automatisierungskonzepte im Bereich M&S untersucht und bewertet. Aus dem sich ergebenden Handlungsbedarf wird ein eigenes Lösungskonzept abgeleitet. Anschließend wird das vorgeschlagene Lösungskonzept prototypisch

implementiert und mittels zweier repräsentativer industrieller Anwendungsfälle validiert. Die Ergebnisse werden abschließend mit den erhobenen Anforderungen abgeglichen, und ein entsprechendes Fazit wird gezogen.

1.4 Aufbau der Arbeit

Zunächst werden in Kapitel 2 die für das Verständnis der Domäne und des Einsatzes von Modellierung und Simulation notwendigen Grundlagen erläutert. In diesem Rahmen wird unter anderem aufgezeigt, welche Systeme im Allgemeinen betrachtet werden, wie deren modellbasiertes Engineering abläuft, wie die Systeme modelliert und letztlich simuliert werden können. Zudem werden aktuelle simulationsbezogene Herausforderungen aufgezeigt und Anforderungen an den lebenszyklusübergreifenden Einsatz von Modellierung und Simulation abgeleitet. Darüber hinaus werden zwei Beispielsysteme im Detail vorgestellt, welche für die im Rahmen dieser Arbeit entstandenen Konzepte als Validierungsfälle genutzt werden. In diesem Kapitel werden somit die Elemente des Design-Problems Artefakt, Anforderungen, Stakeholder-Ziele und Kontext im Detail beschrieben.

In Kapitel 3 wird aufgezeigt, welche Konzepte für die automatische Modellgenerierung, -adaption und -rekonfiguration in verschiedenen Domänen bereits existieren, zu welchem Grad sie die in Abschnitt 2.2.4 definierten Ziele und Anforderungen erfüllen und welcher ergänzende Handlungsbedarf sich daraus ergibt. Abschließend wird in diesem Kapitel eine erarbeitete Gesamtkonzeptübersicht präsentiert sowie eine Vorschau auf die Entwicklung der Teilkonzepte für die spezifischen Lebenszyklusphasen in den folgenden Kapiteln gegeben.

Die Kapitel 4, 5 und 6 greifen – entsprechend den Lebenszyklusphasen „Inbetriebnahme“, „Betrieb“ und „Umbau“ – jeweils einen Teilaspekt aus dieser Gesamtkonzeptübersicht auf. Die Makrostruktur der jeweiligen Kapitel ist möglichst einheitlich gehalten. Zunächst werden die Teilziele aufgezeigt, die in dem jeweiligen Kapitel abgedeckt werden und welche Annahmen und Randbedingungen gelten. Im Anschluss werden spezifische, für das jeweilige Kapitel relevante, technologische Konzeptbausteine vorgestellt und das Teilkonzept im Detail beschrieben. Auf Basis dessen wird sukzessive eine prototypische Lösung implementiert. Diese wird anschließend anhand der beiden in Abschnitt 2.3 vorgestellten Beispielsysteme und -Szenarien validiert und die Ergebnisse diskutiert. Die Kapitel 4 bis 6 schließen jeweils mit einem eigenen Zwischenfazit. Jedes dieser drei Kapitel stellt einen eigenen Design-Zyklus nach Wieringa [WIE14] für die jeweilige Lebenszyklusphase dar. Forschung im Rahmen von Design-Science wird iterativ durchgeführt [JP14]. Das trifft auch auf die zugrundeliegende Forschung im Rahmen dieser Arbeit zu. Entsprechend wurden die Schritte Problemuntersuchung, Artefaktentwicklung und Evaluation jeweils mehrfach durchlaufen. Die vielfältigen Zwischenergebnisse im Rahmen einzelner Iterationsschritte sind jedoch nicht dargestellt. Die Kapitel dieser Arbeit beschränken sich auf die Erkenntnisse, die während der Entwicklung gesammelt wurden, sowie auf die Resultate nach Erreichung des Forschungsziels.

In Kapitel 7 werden abschließend alle Resultate zusammengefasst, ein Fazit hinsichtlich des Gesamtkonzepts gezogen und ein Ausblick auf zukünftige Forschung gegeben.

2 Modellierung und Simulation im modularen Maschinen- und Anlagenbau

Dieses Kapitel zeigt in Abschnitt 2.1 auf, welche Systeme im Rahmen der vorliegenden Arbeit betrachtet werden, wie deren modellbasiertes Engineering erfolgt, welchen Zweck die Modularisierung der Systeme hat und welche Konsequenzen sich daraus für Modellierung und Simulation (M&S) der Systeme ergeben. In Abschnitt 2.2 werden aktuelle Herausforderungen im Zusammenhang mit dem Einsatz von Simulation im modularen Maschinen- und Anlagenbau aufgezeigt und daraus Anforderungen an den zukünftigen Einsatz von Simulation abgeleitet. Abschließend werden in Abschnitt 2.3 zwei Beispielsysteme im Detail vorgestellt. Diese sind unterschiedlichen Bereichen der Fertigung zuzuordnen und werden repräsentativ für die Konzeptvalidierung im weiteren Verlauf herangezogen.

2.1 Grundlagen und Begriffe im modularen Maschinen- und Anlagenbau

2.1.1 Heterogenität im Maschinen- und Anlagenbau

Um einen Eindruck über die Vielfältigkeit der Maschinen und Anlagen für die Teilefertigung zu erhalten, ist es sinnvoll, im ersten Schritt einen Überblick über Fertigungsverfahren zu schaffen. In [DIN 8580] wird ein Ordnungssystem für Fertigungsverfahren definiert, welches in Hauptgruppen, Gruppen und Untergruppen differenziert. Insgesamt werden sechs Hauptgruppen unterteilt, dazu gehören *Urformen*, *Umformen*, *Trennen*, *Fügen*, *Beschichten* und *Stoffeigenschaft ändern*. Die Merkmale der Hauptgruppen sind in Tabelle 1 zusammengefasst. Die Verfahren dienen zur Schaffung und Änderung der Form der gefertigten Bauteile oder Produkte sowie zur Änderung deren Stoffeigenschaften.

Tabelle 1: Merkmale der Hauptgruppen für Fertigungsverfahren nach [DIN 8580]

Schaffen der Form	Ändern der Form				Ändern der Stoffeigenschaften
Zusammenhalt schaffen	Zusammenhalt beibehalten	Zusammenhalt vermindern	Zusammenhalt vermehren		–
Hauptgruppe 1 Urformen	Hauptgruppe 2 Umformen	Hauptgruppe 3 Trennen	Hauptgruppe 4 Fügen	Hauptgruppe 5 Beschichten	Hauptgruppe 6 Stoffeigenschaft ändern

Weiter werden in [DIN 8580] für jede Hauptgruppe jeweils fünf bis zehn Gruppen und insgesamt über 150 Untergruppen von Fertigungsverfahren differenziert. Hieraus wird deutlich, dass sich bereits durch die unterschiedlichen Fertigungsverfahren eine Vielzahl an Maschinen- und Anlagenausprägungen ergibt. Die eigentliche Herausforderung, bedingt durch Komplexität und Mannigfaltigkeit der Produktionsprozesse und entsprechend der Maschinen und Anlagen, ergibt sich allerdings erst, wenn markt- und produktbedingte Anforderungen miteinbezogen werden. Bei komplexeren Produktionsmaschinen und -anlagen handelt es sich zunehmend um Sonderanfertigungen. Die Bedeutung des Sondermaschinenbaus und individueller Integrationslösungen nimmt nach [Pös21] unter Verweis auf [VM14, GN09] stark zu.

Abbildung 1 visualisiert die Verteilung deutscher Maschinenbauer auf Anbietertyp und Standardisierungskategorie der produzierten Maschinen und Anlagen (Stand 2014, [VM14]). Das Kreisdiagramm rechts in der Abbildung zeigt, dass sich der Markt in Anbieter aufteilt, die entweder Komplettlösungen oder Einzel-

maschinen und Komponenten herstellen. Das Balkendiagramm links in der Abbildung stellt den Standardisierungsgrad der angebotenen Produkte dar. Der Anteil an individualisierten und gemischt standardisiert-individualisierten Maschinen beträgt über 80% und macht damit den größten Teil aus. Demnach konzentriert sich ein großer Teil des deutschen Maschinenbaus auf den Bau von Maschinen, bei welchen es sich um nicht standardisierte Komplettlösungen handelt. Diese Sondermaschinen erfordern ein hohes Maß an Engineeringaufwand. [Pös21] Das Engineering wird ausführlich in Abschnitt 2.1.4 behandelt.

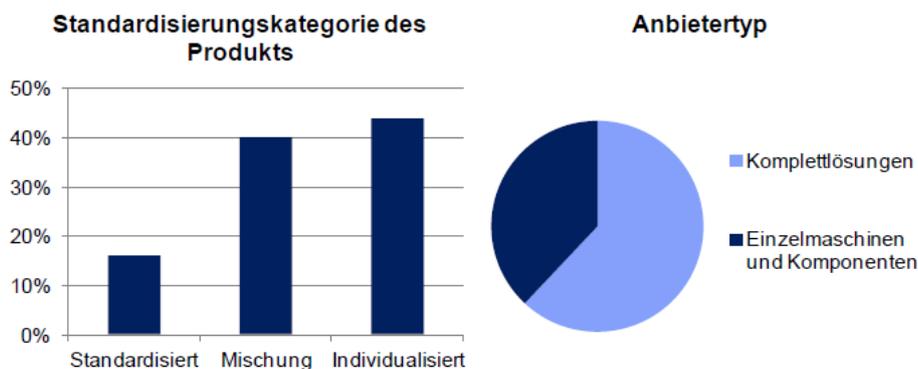


Abbildung 1: Verteilung der Unternehmen auf Anbietertyp und Standardisierungskategorie nach [Pös21] mit Daten von [VM14].

Bevor eine Lösung für die Sonderanfertigungen und das daraus resultierende Engineering der Maschinen und Anlagen betrachtet wird, sollen zunächst im nächsten Abschnitt die Anforderungen, die die Sonderanfertigungen erforderlich machen, im Detail dargelegt werden.

2.1.2 Markt- und produktbedingte Anforderungen an Produktionsanlagen

Für viele Unternehmen ist die Entwicklung von Produkt- und Prozessinnovationen ein zentraler Erfolgsfaktor [GL16]. Dabei gilt es, sich auf dynamischen und stark diversifizierten Märkten zu behaupten, die von Globalisierung und Preiskampf geprägt sind [FUL20]. Es herrscht ein Spannungsfeld aus stetig steigenden Qualitätsansprüchen bei sich verkürzenden Entwicklungszyklen, zunehmender Produkt- und Prozesskomplexität sowie Innovationsdruck bei anhaltenden Maßnahmen zur Kostenreduktion [MAT17]. Die verstärkte Nachfrage nach kundenindividuellen Produkten und Produktvarianten führt zu kleineren, variationsreichen Losgrößen. Als Beispiel sind die Ausstattungsvarianten und Konfigurationsmöglichkeiten, die die Automobilindustrie ihren Kunden anbietet, zu nennen. Klassische Produktionskonzepte geraten bei diesen Losgrößen schnell an die wirtschaftlichen und technologischen Grenzen. [GVQ+16] Eine Übersicht weiterer Änderungen der Fertigungsanforderungen in der deutschen Metall- und Elektroindustrie ist in Abbildung 2 zusammengefasst.

Fertigungsanforderungen	Tendenz
Entwicklung des Variantenangebots (Anzahl angebotener Varianten)	↑↑
Entwicklung der Losgrößen (Fertigungslose)	(↓)
Entwicklung des Materialmixes (Anzahl eingesetzter Materialien)	↗
Entwicklung der Produktgröße (Außenmaße)	↗
Entwicklung der Genauigkeitsanforderungen	↑
Entwicklung der Produktlebenszyklen (Umsatzanteil neuer Produkte)	→

Abbildung 2: Entwicklung der Fertigungsanforderungen in der deutschen Metall- und Elektroindustrie im Überblick [KIN05].

Hinzu kommen gesetzliche Anforderungen, beispielsweise die Energieeffizienz oder Nachhaltigkeit betreffend, die Einfluss auf die Technik nehmen [BOR13]. Entsprechend ist ein Umdenken der Unternehmen bezüglich der Flexibilisierung ihrer Entwicklungsprozesse gefordert [MAT17]. Weiter wird in [MAT17] eine Entwicklung von marktgetriebenen Produktentwicklungsansätzen hin zu technologiegetriebener Produktentwicklung beschrieben. Erstere gehen von der Produktfunktion aus und berücksichtigen die fertigungstechnische Realisierung erst nach dem Abschluss aller Entwicklungstätigkeiten. Das Ziel beim zweiten Entwicklungsansatz liegt „in der umfassenden Ausschöpfung der fertigungstechnologischen Potenziale bei gleichzeitiger Berücksichtigung von fertigungsbedingten Restriktionen im Entwicklungsprozess“. [MAT17] Doch allein durch die frühzeitige Berücksichtigung der fertigungstechnischen Realisierung sind Produktionsanlagen nicht mehr in der Lage, über mehrere Jahre oder Jahrzehnte den wechselnden Produktionsanforderungen und -Randbedingungen gerecht zu werden. Dies muss allerdings gegeben sein, damit sich die Produktionsanlagen als industrielle Investitionsgüter amortisieren können [MHW+18].

Nach Gorecky et al. [GVQ+16] werden daher neuartige Produktionskonzepte benötigt, um angemessen auf die aktuellen und zukünftigen Herausforderungen reagieren zu können. Gorecky et al. formulieren hierfür drei Zielkriterien, die zu erfüllen sind. Neben der individualisierten Massenproduktion und der inner- und überbetrieblichen Vernetzung nennen sie die *mechatronische Wandelbarkeit* der Produktionsanlagen als Zielkriterium. Um die Wandelbarkeit zu erreichen, werden unter anderem von Wiendahl et al. [WRN14] fünf *Wandlungsbefähiger* definiert. Hierzu gehört die Universalität, die Mobilität, die Skalierbarkeit, die Kompatibilität und die Modularität. In Abbildung 3 werden Begrifflichkeiten in Bezug auf Wandlungsfähigkeit und den Wandlungsbefähigern nach [LIE13] eingeordnet. Alle Befähiger können grundsätzlich sowohl proaktiv als auch reaktiv hinsichtlich Veränderungen innerhalb des industriellen Umfeldes eingesetzt werden. Holm [HOL16] betont unter Verweis auf [WIE02], dass Modularität unter den Wandlungsbefähigern eine herausgehobene Stellung einnimmt. Der Grund hierfür ist, dass diese als elementare Voraussetzung für Wandlungsfähigkeit angesehen wird. „Wird Modularität nicht genutzt, können Produktionsanlagen zwar zum Beispiel mobil [sein], aber in Gänze nicht die Eigenschaft der Wandlungsfähigkeit erreichen“ [HOL16]. In der weiteren Betrachtung liegt der Fokus dieser Arbeit daher auf der Modularität der Anlagen.

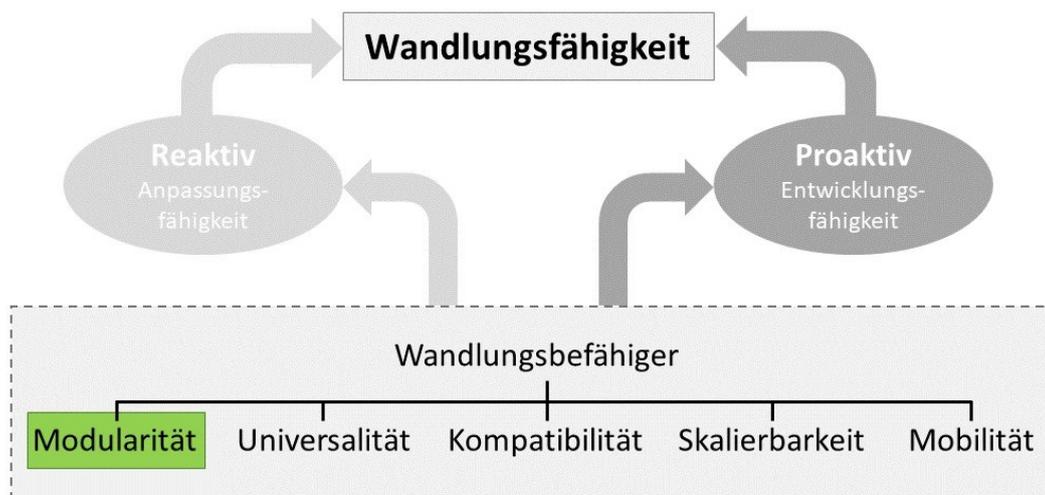


Abbildung 3: Einordnung der Wandlungsbefähiger in Anlehnung an [LIE13].

Sind „Produktionsanlagen [...] im Sinne der Minimierung von Stillstandzeiten und Kostenersparnis beim Umrüsten wandlungsfähig und modular aufgebaut; lassen [sie] sich schnell und aufwandsarm in Betrieb nehmen und an neue Produktionsanforderungen adaptieren“ [GVQ+16]. Folglich ist der Übergang von monolithisch zu modular aufgebauten Produktionsanlagen eine Möglichkeit, die nötige Wandelbarkeit und die individualisierte Massenproduktion sicherzustellen. Nach [MSC+19] werden die Anlagen hierzu in flexibel kombinierbare Module aufgeteilt. Durch Rekonfiguration entsteht ein vielseitiges Produktionssystem, das durch Austausch oder Hinzufügen einzelner Module an wechselnde Produkte und Bedarfsmengen angepasst werden kann. Eine Anlage kann somit zur Herstellung mehrerer Produkte genutzt werden, wodurch sich die hohen Ausrüstungskosten auf mehrere Produkte verteilen und somit auch kleinere Chargen wirtschaftlich rentabel werden. [MSC+19] Eine modulare Anlagenarchitektur ermöglicht zudem eine flexiblere Produktion verschiedener Teile aufgrund der Rekonfigurierbarkeit. Auch dies führt zu Kosteneinsparungen, die eine wirtschaftliche Fertigung kleinerer Losgrößen ermöglichen. [MSC+19]

Entsprechend kann so ein längerer wirtschaftlicher Einsatz der Anlagen sichergestellt werden. Dies deckt sich mit den Aussagen aus [VDI/VDE/NAMUR 2658-1], wonach es drei wesentliche Anwendungsszenarien für modulare Anlagen gibt. Hierzu zählen die kurze Time-to-Market, die schnelle Time-to-Repair sowie die individuelle Produktion kleiner Chargen. Der nächste Abschnitt befasst sich mit Begriffsdefinitionen und der Umsetzung von Modularisierung im Maschinen- und Anlagenbau sowie den daraus resultierenden Folgen für das Engineering.

2.1.3 Modularisierung der Produktionsanlagen

Zunächst ist hervorzuheben, dass keine einheitliche und konsistente Definition des Begriffs Modularität existiert. Holm [HOL16] stellt unter Verweis auf [MIL01] heraus, dass unterschiedliche Anwendungsdomänen oft ihr eigenes Verständnis von Modularisierung entwickelt haben. Das hat zur Folge, dass in der technischen Literatur und der Entwicklungspraxis die Begriffe Modul, modular und Modularisierung unterschiedlich verwendet werden [HOL16]. Bezogen auf die Anwendungsdomäne Produktion und Maschinen- und Anlagenbau kann unter anderem auf folgende Begriffsdefinitionen und Auffassungen verwiesen werden. Friedli und Schuh [FS12] verstehen unter Modularität „die Bildung von in sich möglichst abge-

schlossenen Funktionseinheiten. Größere Systeme werden durch den Zusammenbau solcher Module realisiert. Die Modularisierung dient dabei der Komplexitätsreduktion.“ Dieses allgemeine Begriffsverständnis kann noch konkretisiert werden. In [VDI 2776-1] wird unter Modularität die „Planung und [der] Bau von prozesstechnischen Anlagen mit aufeinander abgestimmten Einheiten, Abmessungen oder Schnittstellen, die einfach zusammengebaut, gewartet sowie flexibel angeordnet und betrieben werden können“ verstanden. Diese Definition kann auch auf diskrete Fertigungsanlagen übertragen werden. Hildebrandt et al. [HHD+22] verstehen unter Modularisierung im Maschinen- und Anlagenbau „die Auftrennung komplexer Produktionsprozesse in grundlegende Funktionseinheiten [...]. Die resultierenden Module können grundlegende Prozesse ausführen und auf Basis standardisierter Schnittstellen zu Gesamtanlagen kombiniert werden.“ Alle drei Auffassungen des Begriffs Modularität haben als gemeinsame Schnittmenge, dass (Funktions-)Einheiten, auch Module genannt, existieren. Das Gesamtsystem entsteht durch die Kombination und den Zusammenbau dieser Module. In [VDI 2776-1] wird beschrieben, dass die Module über Schnittstellen verfügen. Hildebrandt et al. [HHD+22] erweitern diese Auffassung und gehen von standardisierten Schnittstellen aus. Im nächsten Schritt wird daher untersucht, was unter Modulen verstanden wird und über welche Schnittstellen diese verfügen müssen.

Klein [KLE14] untersucht im Rahmen seiner Arbeit Methoden zur Definition wiederverwendbarer Module wie die Design Structure Matrix oder die Methoden Modular Function Deployment und Design for Variety. Er kommt zu dem Fazit, dass keine Methode eindeutige Vorteile für die Definition wiederverwendbarer Module bietet. Das Erfahrungswissen des Entwicklers und auch die Disziplin, aus deren Betrachtung heraus eine Produktionsanlage modularisiert wird, haben Einfluss darauf, welche Komponenten als Modul definiert und wie diese systematisiert werden [KLE14]. Kolberg et al. [KHM+18] lösen dieses Problem, indem sie unter einem Modul „ein[en] abstrakte[n] Begriff für ein in sich abgeschlossenes System, welches selbstständig eine Funktionalität anbieten kann [verstehen]. Es ist gekennzeichnet durch klar definierte Schnittstellen und eine Beschreibung der angebotenen Funktionalitäten.“ Weiter nach [KHM+18] ist „im Sinne des ‚Black-Box‘-Ansatzes [...] die genaue Realisierung der Funktion im Modul für die Verwendung nicht bekannt, sondern nur die Schnittstelle und deren Nutzung.“ Die standardisierten Schnittstellen decken Stoff-, Energie- und Informationsflüsse ab [KLE14, KHM+18].

In Abbildung 4 [KLE14] wird exemplarisch die Konfiguration einer Produktionsanlage aus standardisierten Prozess- und Strukturmodulen dargestellt. Zu den Prozessmodulen, links oben in der Abbildung, gehören z. B. Stationen für Pick and Place Aufgaben, Bohren oder Einpressen. Strukturmodule, links unten in der Abbildung, können beispielsweise Umlenk- oder Transfermodule sein. Durch die standardisierte modulare Bauweise lassen sich unterschiedliche Produktionsanlagen konfigurieren. Eine mögliche standardisierte Schnittstelle nach [KHM+18] für den Energie- und Informationsaustausch ist rechts in der Abbildung ergänzt.



Abbildung 4: Exemplarische Konfiguration einer Produktionsanlage aus standardisierten Prozess- und Strukturmodulen nach [KLE14, FN09].

Die Vorteile der Nutzung modularer Strukturen wurden bereits im vorherigen Abschnitt aufgezeigt. Die Modularisierung der Produktionsanlage hat jedoch auch Einfluss auf zahlreiche Arbeitsprozesse in den Unternehmen, da betriebliche Funktionen u.a. in der Entwicklung, Konstruktion sowie der Produktion angepasst werden müssen [Dis19]. Aus Sicht des Engineerings ergeben sich durch den Übergang von monolithischen zu modularen Produktionsanlagen neue Herausforderungen und Anforderungen. Im nächsten Schritt sollen zunächst der Engineering-Prozess von monolithischen im Vergleich zu modularen Anlagen aufgezeigt und anhand dessen die neuen Anforderungen an die Engineering-Methoden abgeleitet werden.

2.1.4 Engineering im Maschinen- und Anlagenbau

Durch die Interdisziplinarität sowie durch die zunehmende Komplexität der Systeme [WHT+12, BS20b] wächst die Herausforderung eines effektiven und effizienten Engineerings [KAI13]. Das *Systems Engineering* nimmt sich dieser Herausforderung an und setzt das Systemdenken in den Vordergrund. Hierdurch soll bei allen beteiligten Disziplinen ein einheitliches und ganzheitliches Systemverständnis erreicht werden. Um dieses ganzheitliche Systemverständnis bei allen am Engineering-Prozess Beteiligten sicherzustellen, werden Modelle zur Unterstützung eingesetzt. Der Bereich *Model-Based Systems Engineering* beschäftigt sich mit diesen Ansätzen. [KAI13] Diese Modelle und die damit durchgeführten Simulationen dienen als Engineering-Werkzeuge. Engineering-Werkzeuge repräsentieren nach [VDI 3695-1] „alle bei der Erfüllung der Engineering-Aufgabe notwendigen technischen Arbeitsmittel, sowohl auf Seiten der Software als auch auf Seiten der Hardware. Software-Werkzeuge unterstützen den Mitarbeiter beim Engineering-Prozess durch optimierte Prozessabläufe, vordefinierte Vorlagen, einfache Bedienung und Verwendung standardisierter Beschreibungsmittel“. [VDI 3695-1] Das Model-Based Systems Engineering sowie der Einsatz von M&S als Engineering-Werkzeuge werden in Abschnitt 2.1.5 im Detail vorgestellt.

Mithilfe von virtuellen, hybriden und physischen Tests lassen sich Verifikation und Validierung durchführen. Unter hybride Tests fallen beispielsweise die Software-in-the-Loop- (SiL) sowie die Hardware-in-the-Loop- (HiL) Simulation. Als HiL-Simulation bezeichnen Jaensch et al. in [JCS+18] die Kopplung realer Steuerungssysteme mit dem Simulator über die reale Kommunikationsperipherie. Bei einer SiL- Simulation

werden zusätzlich das Steuerungssystem sowie die Kommunikationsperipherie simuliert. Im praktischen Einsatz im Maschinen- und Anlagenbau helfen diese Methoden unter anderem, Zeit bei der Inbetriebnahme von Automatisierungslösungen zu sparen. Tests von regulären und irregulären Betriebszuständen gegen Modelle führen zu einer höheren Softwarequalität. [JCS+18]

Vorzeitige, virtuelle Tests stellen eine Besonderheit dar, um Teilaspekte des Systems auf der Grundlage von Systemmodellen ohne disziplinspezifische Detaillierung von Subsystemen und Systemelementen zu prüfen [VDI/VDE 2206]. Der vorzeitige Test ist in Abbildung 5 durch eine Verästelung im V-Modell dargestellt. Als Makrozyklus kann das Vorgehen im V-Modell mehrfach durchlaufen werden, wobei die Ergebnisse iterativ verfeinert werden [SCH18].

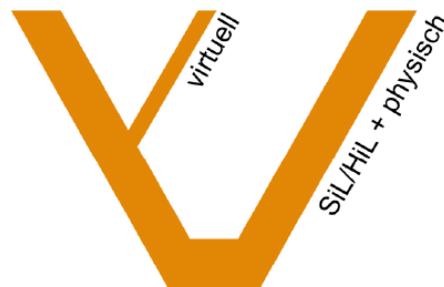


Abbildung 5: Vorzeitige Verästelung im V-Modell für virtuelle Tests nach [VDI/VDE 2206].

Wie bereits in Kapitel 1 aufgezeigt, geht das Engineering im Maschinen- und Anlagenbau über die Entwicklungsphase hinaus und umfasst auch Tätigkeiten in anderen Lebenszyklusphasen. Auch in diesen Phasen können das V-Modell als Entwicklungsmethode sowie die im Rahmen der Entwicklung entstandenen Modelle genutzt werden.

Abbildung 6 zeigt oben den Lebenszyklus einer monolithischen Produktionsanlage, angelehnt an [VDI 4499-2]. Hierbei handelt es sich um einen sequenziellen Ablauf. Unten in der Abbildung ist im Vergleich dazu der Lebenszyklus einer modularen Produktionsanlage dargestellt. Bereits während der Anlagenkonzeption werden Anlage und Module so geplant, dass eine Rekonfiguration des Gesamtsystems leicht möglich ist. Entsprechend werden die blau markierten Phasen Inbetriebnahme, Betrieb und Umbau mehrfach durchlaufen. Hierfür bietet sich daher die zyklische Darstellung unten in Abbildung 6 an. Durch die Darstellung wirkt der Lebenszyklus der modularen Anlage kürzer als der der monolithischen Anlage. Entsprechend der Darlegungen in Abschnitt 2.1.3 ist jedoch das Gegenteil der Fall; die modulare Anlage kann durch Umbau- und Rekonfigurationsmaßnahmen länger betrieben werden.

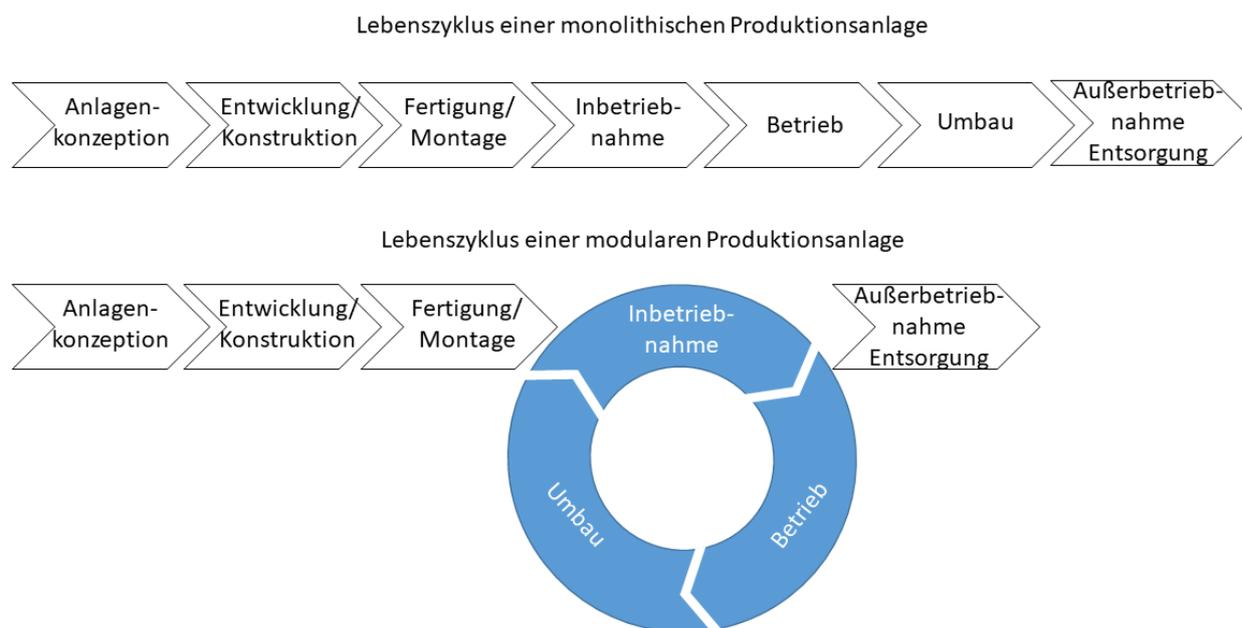


Abbildung 6: Lebenszyklusphasen einer monolithischen Produktionsanlage (oben) angelehnt an [VDI 4499-2] und einer modularen Produktionsanlage (unten).

Hierbei ist zu beachten, dass Anlagenmodule teilweise oder vollständig von Zuliefererbetrieben stammen [PB07]. Entsprechend übernehmen diese, ggfs. in Absprache mit dem Anlagenbetreiber, auf Modulebene die Lebenszyklusphasen Konzeption, Entwicklung und Fertigung/ Montage. Der Anlagenbetreiber konfiguriert die Anlage entsprechend der markt- und produktionsbedingten Anforderungen. Die Modulzulieferer nutzen für Konzeption und Entwicklung ebenfalls Engineering-Werkzeuge, Modellierung und Simulation. Die dabei entstandenen MSM stellen die Zulieferer dem Betreiber z. B. als Service oder eigenständiges Produkt zur Verfügung. Der Anlagenbetreiber kann diese MSM dann für Simulation in den Phasen Inbetriebnahme, Betrieb und Umbau nutzen.

Im weiteren Verlauf der Arbeit wird daher der Fokus auf diese drei, in Abbildung 6 blau markierten, Lebenszyklusphasen gelegt. Im Speziellen soll der Einsatz von M&S als Engineering Werkzeug in diesen Phasen untersucht werden, da diese mehrfach durchlaufen werden und Einsatzpotential sowie Nutzen von M&S groß sind [GMA20]. Der nächste Abschnitt befasst sich daher zunächst mit den Grundlagen des Model-Based Systems Engineering, bevor in Abschnitt 2.2 die Herausforderungen beim Einsatz von Simulation im modularen Maschinen- und Anlagenbau aufgezeigt werden.

2.1.5 Model-Based Systems Engineering

2.1.5.1 Grundlagen zu Modellen und Modellbildung

Zunächst sollen grundlegende Begriffe aus dem Bereich M&S, die für das weitere Verständnis dieser Arbeit relevant sind, definiert werden. In [VDI 3633-1] wird ein Modell als „vereinfachte Nachbildung eines geplanten oder existierenden Systems mit seinen Prozessen in einem anderen begrifflichen oder gegenständlichen System“ definiert. Weiter „unterscheidet [es] sich hinsichtlich der untersuchungsrelevanten Eigenschaften nur innerhalb eines vom Untersuchungsziel abhängigen Toleranzrahmens vom Vorbild“. Eine weitere Modell-Definition wird in [VDI/VDE 2206] geliefert. Demnach stellt ein Modell eine „digitale Repräsentation des Systems dar, in der die Funktionen vereinfacht dargestellt werden. Modelle werden für

Gesamtsysteme, Subsysteme und Systemelemente erstellt und unterscheiden sich in der betrachteten Problemstellung und im Detaillierungsgrad“. Ziel der Modellierung sind weiter nach [VDI/VDE 2206] „*mathematische und deskriptive Ersatzmodelle des Systems, der Subsysteme oder Systemelemente, die die Struktur und das Verhalten mit ausreichender Genauigkeit beschreiben*“. Anforderungen an die Modellgenauigkeit und Detailtiefe erhöhen sich dabei mit dem Fortschritt der Entwicklung. Alle Änderungen am System sollten auch im Modell berücksichtigt werden. [VDI/VDE 2206]

Nach Schiering et al. [SSH+21] kann der Prozess zur Bildung eines Modells in sechs Schritte unterteilt werden. Die Schritte zur Modellbildung inklusive kurzer Deskriptionen sind in Tabelle 2 dargestellt.

Tabelle 2: Schritte im Modellbildungsprozess nach [SSH+21]

Modellbildungsschritt	Beschreibung
1. Abgrenzung	Modelle haben bestimmten Geltungsbereich und sind nicht universell anwendbar. Z. B. durch Definition physikalischer Bedingungen lässt sich Bereich, auf den das Modell anwendbar ist, einschränken. Verallgemeinerbarkeit des Modells wird im Wesentlichen durch dessen Umfang bestimmt.
2. Abstraktion	Modelle werden nicht nur für einzelne Systeme, sondern für Klassen von Systemen, die sich auf gleiche Weise beschreiben lassen, erstellt. Modell kann nach Anpassung der Modellparameter auf ganze Klasse von gleichen oder ähnlichen Systemen angewandt werden.
3. Reduktion	Mit Schritt der Abstraktion geht notwendige Reduktion der Systembeschreibung einher, um diese im Sinne der Zielsetzung verständlich und erklärbar zu machen. Vernachlässigung von Einflussfaktoren und Details, die für das Systemverhalten von geringer Relevanz sind. Reduktion der Systembeschreibung und damit Reduktion der Informationen über System führt zu Abweichungen des modellierten Verhaltens vom realen Systemverhalten. Die durch Reduktion entstehenden systematischen oder zufälligen Abweichungen sind im Rahmen von Schritt 6 Verifizierung zu bewerten.
4. Dekomposition	Reale, meist komplexe Systeme lassen sich in der Regel nicht vollständig verstehen und beschreiben. Bevorzugter Ansatz besteht darin, System in überschaubare und leicht zu modellierende Teilsysteme zu zerlegen. Gesamtsystem lässt sich durch mehrere Teilmodelle in seiner Gesamtheit repräsentieren. Herausforderung besteht darin die Grenzen der Teilsysteme so zu wählen, dass möglichst einfach zu modellierende Funktionsprimitive entstehen. Alternativer Ansatz zur Definition von Teilsystemen ist die Klassifizierung nach den physikalischen Bereichen, für die sie gelten.
5. Aggregation	Generierung des Modells des Gesamtsystems durch Kombination der im Dekompositionsschritt erzeugten Teilmodelle. Voraussetzung hierfür ist, dass sich die Teilmodelle so ergänzen, dass gültiges Gesamtmodell erstellt werden kann. Zu starke Reduktion innerhalb der Teilmodelle kann zu Abweichungen im Gesamtmodell führen, die für jedes Teilmodell noch als tolerierbar einzustufen wäre. Abweichungen können sich so stark auf Gesamtmodell auswirken, dass Modellbildung überarbeitet werden muss. (Abweichungsreduktion z. B. durch Einbeziehung bisher nicht berücksichtigter Einflussgrößen oder Nichtlinearitäten.)

6. Verifizierung	<p>Empirische Ermittlung und Bewertung der Abweichungen zwischen Modell und Realität. Hierzu zählen z. B. Bewertung von systematischen und zufälligen Abweichungen sowie Fragen zur Systemdynamik, der zeitlichen Variabilität des Systemverhaltens und der Systemreaktion auf externe Störungen.</p> <p>Verifizierung schafft Vertrauen in die, durch die Simulation der Modelle erzeugten Ergebnisse. Dies stellt wiederum Grundlage für Rechtfertigung der Simulation als alternatives Untersuchungswerkzeug zum Realversuch zu nutzen [VW17].</p>
------------------	---

2.1.5.2 Modellbildung modularer Anlagen: Zulieferer- und Betreibersicht

Im folgenden Absatz soll aufgezeigt werden, welche Bedeutung dieser Modellbildungsprozess für modulare Anlagen aus Zulieferer- und Betreibersicht hat. Wie bereits in Abschnitt 2.1.4 beschrieben, kommen Module und MSM zunehmend von Zulieferern. Entsprechend führen die Zulieferer alle Schritte in Tabelle 2 im Rahmen der Modellbildung der MSM durch. Der Anlagenbetreiber muss wiederum Teile des Prozesses für das Modell der Anlage durchführen. Den Modellbildungsprozess der Anlage haben die Zulieferer, ggfs. in Absprache mit dem Anlagenbetreiber, bis einschließlich Schritt 4 *Dekomposition* durchgeführt. Der Betreiber muss nun abschließend erneut die Schritte 5 *Aggregation* und 6 *Verifizierung* durchführen, um ein Modell der Anlage zu erhalten.

Bei den zu kombinierenden Teilmodellen im Rahmen der Aggregation handelt es sich um die MSM. Der Anlagenbetreiber muss diese, entsprechend der Konfiguration der realen Anlage, kombinieren. Hierzu ist neben der Parametrierung eine korrekte Kopplung der Ein- und Ausgänge der MSM nötig. Dem in Schritt 5 beschriebenen Effekt, dass durch zu starke Reduktion innerhalb eines oder mehrerer Teilmodelle eine nicht tolerierbare Abweichung im Gesamtmodell entsteht, kann durch MSM in verschiedenen Detaillierungsgraden entgegengewirkt werden. Dies erfordert, dass die Zulieferer MSM in unterschiedlichen Detaillierungsgraden zur Verfügung stellen. Abhängig vom Anwendungsfall und der vorgegebenen Abweichungstoleranz können die Detaillierungsgrade der MSM durch den Betreiber gewählt und aggregiert werden. Auf das Thema Umgang mit Abweichungen und Wahl geeigneter Detaillierungsgrade wird in Kapitel 5 im Detail eingegangen.

Die Verifizierung von Modellen im Rahmen von Schritt 6 erfolgt nach Einschätzungen von Laroque und Skoogh [LS17] heute hauptsächlich durch Simulationsexperten in manuellen Prozessen. Bei diesem Schritt des Modellbildungsprozesses ist zu beachten, dass es sich hierbei um keine einmalige Aufgabe handelt. Zum Beispiel kann sich durch den Verschleiß von Bauteilen wie Lagern oder Ventilen sowie durch äußere Einflüsse auf die Anlage wie die Umgebungstemperatur das Systemverhalten der Anlage ändern. Diese Änderungen können temporärer (z. B. hohe Temperatur in der Produktionshalle im Sommer) oder permanenter (z. B. Bauteilverschleiß) Natur sein. In beiden Fällen muss das Modell kontinuierlich verifiziert und bei Bedarf angepasst werden. Diese Anpassung, auch Modelladaptation genannt, kann z. B. durch die Einstellung der Modellparameter vorgenommen werden. Auf Modellparameter wird im nächsten Unterabschnitt im Detail eingegangen.

Sofern sich die Konfiguration der realen Anlage, bedingt durch markt- und produktbedingte Anforderungen, im Rahmen eines Umbaus ändert, muss der Betreiber die aufgezeigten Schritte des Modellbildungsprozesses (Aggregation und Verifizierung) erneut durchlaufen. Eine Rekonfiguration des Anlagenmodells wird nötig.

Auf die mit dem Modellbildungsprozess modularer Anlagen verbundenen Herausforderungen wird in Abschnitt 2.2.1 im Detail eingegangen. Im nächsten Unterabschnitt sollen zunächst verschiedene Modellarten vorgestellt werden.

2.1.5.3 Modellarten

In [VDI/VDE 2206] werden zwei wesentliche Wege, ein System in eine abstrakte, systemunabhängige Darstellung zu überführen, differenziert. Bei der theoretischen Modellbildung werden die Systemgleichungen „durch Anwenden der physikalischen Grundgesetze abgeleitet, wobei die quantitativen Informationen aus der Geometrie des betreffenden Gebildes, aus Stoffkonstanten und aus empirischen Zusammenhängen gewonnen“ werden. Die experimentelle Modellbildung nutzt im Gegensatz dazu Messungen an dem zu beschreibenden System, um auf deren Basis rückwärts auf die Systemstruktur zu schließen. Sind bei diesen Modellen nur die Rückwirkungen des Systems auf die Umgebung (Output) aufgrund der Einwirkungen auf das System (Input) bekannt, wird das Modell nach [VDI 3633-1] als *Black-Box-Modell* bezeichnet. Modelle, die durch theoretische Modellbildung entstanden sind, sind als *White-Box-Modelle* aufzufassen, da neben dem Ein-/Ausgangverhalten durch die Systemgleichungen auch die internen Abläufe nachvollziehbar sind [EBE13]. Durch Mechanismen wie die Kompilierung des Modellcodes ist es möglich, White-Box in Black-Box-Modelle zu überführen und so das darin konzentrierte Systemwissen zu schützen. Dieser Aspekt ist relevant und gibt Modulzulieferern die Möglichkeit MSM bereit zu stellen, ohne dabei ihre Expertise offenlegen zu müssen [HBF20*b]. Gebräuchlich ist nach [VDI/VDE 2206] noch eine Kombination der beiden Varianten, „um z. B. das Verhalten des zunächst theoretisch aufgestellten Modells mit Messungen am realen System zu vergleichen oder um noch unbekannte Parameter durch Abgleich mit dem realen System zu bestimmen“. In [VDI 3633-1] wird ein Parameter im Zusammenhang mit M&S als „Attribut, das vor einem Simulationslauf gesetzt wird und sich innerhalb des Simulationslaufs nicht ändert“ definiert. Dagegen abzugrenzen sind Modellvariablen. Diese sind, wie die Parameter, Bestandteile der Systemgleichungen, können sich jedoch während eines Simulationslaufs ändern.

In Abbildung 7 sind die grundlegenden Begriffe, welche in den vorherigen Abschnitten zu den Themen Modellbestandteile und Modellbildung aufgezeigt wurden, anhand eines Beispiels zusammengefasst. Hierbei ist die Modellbildung von einem Modul aus Zulieferersicht dargestellt. Links oben in der Abbildung ist zunächst ein beliebiges reales System abgebildet, in dem konkreten Fall eine Bohrstation, welche das Modul darstellt. Die Wahl ist dadurch zu begründen, dass es sich hierbei um einen häufig zum Einsatz kommenden Bestandteil von Fertigungsanlagen handelt. Gleichzeitig ist die Systemkomplexität begrenzt. Der Zulieferer hat während und nach der Modellbildung Einsicht in sein Modell, daher handelt es sich für ihn um ein Whitebox-Modell. Das Modell verfügt, wie rechts in Abbildung 7 dargestellt, über Ein- und Ausgänge sowie Parameter, die vor jeder Simulation angepasst werden können. Mögliche Beispiele für diese Größen könnten z. B. die Vorgabe einer Motorspannung als Eingang, der Durchmesser des verwendeten Bohrers als Parameter und die Prozesszeit als Ausgang sein. Bei diesen drei Größen handelt es sich um die Schnittstellen des Modells. Im Inneren des Modells wird das Systemverhalten beschrieben. In Abbildung 7 ist dies im unteren Teil durch Systemgleichungen, welche das Ergebnis theoretischer Modellbildung sind, dargestellt. Hierbei handelt es sich um eine objektorientierte Modellierungssprache für physikalische Modelle. Auf die Vorteile dieser Art der Systemrepräsentation wird in Unterabschnitt 2.1.5.5 im Detail eingegangen. Im oberen Teil des Modells (Zeilen 2 bis 4) werden zunächst Parameter und Systemvariablen definiert. Weitere werden genutzt, um interne Systemzustände darzustellen, die sich während

einer Simulation ändern können. Im unteren Teil des Modells (Zeilen 5 bis 9) sind die Gleichungen zusammengefasst, die das Verhalten beschreiben. Wie in Schritt 5 *Aggregation* von Tabelle 2 beschrieben, kann eine zu starke Reduktion der Systembeschreibung innerhalb der Teilmodelle zu Abweichungen im Gesamtmodell führen. Konkret würde das bedeuten, dass das Modell der Bohrstation zu ungenau ist und sich dies negativ auf eine Simulation des Gesamtmodells der modularen Anlage auswirkt. In diesem Fall ist es erforderlich, dass der Zulieferer im MSM weitere Effekte wie Lagerreibung oder thermische Einflüsse, die in der Realität auftreten und die bisher vernachlässigt wurden, abbildet. Hierzu ist es erforderlich, neue Modellgleichungen einzuführen und ggfs. neue Parameter und Variablen zu deklarieren. Dies ist durch die grünen Felder in Abbildung 7 angedeutet.

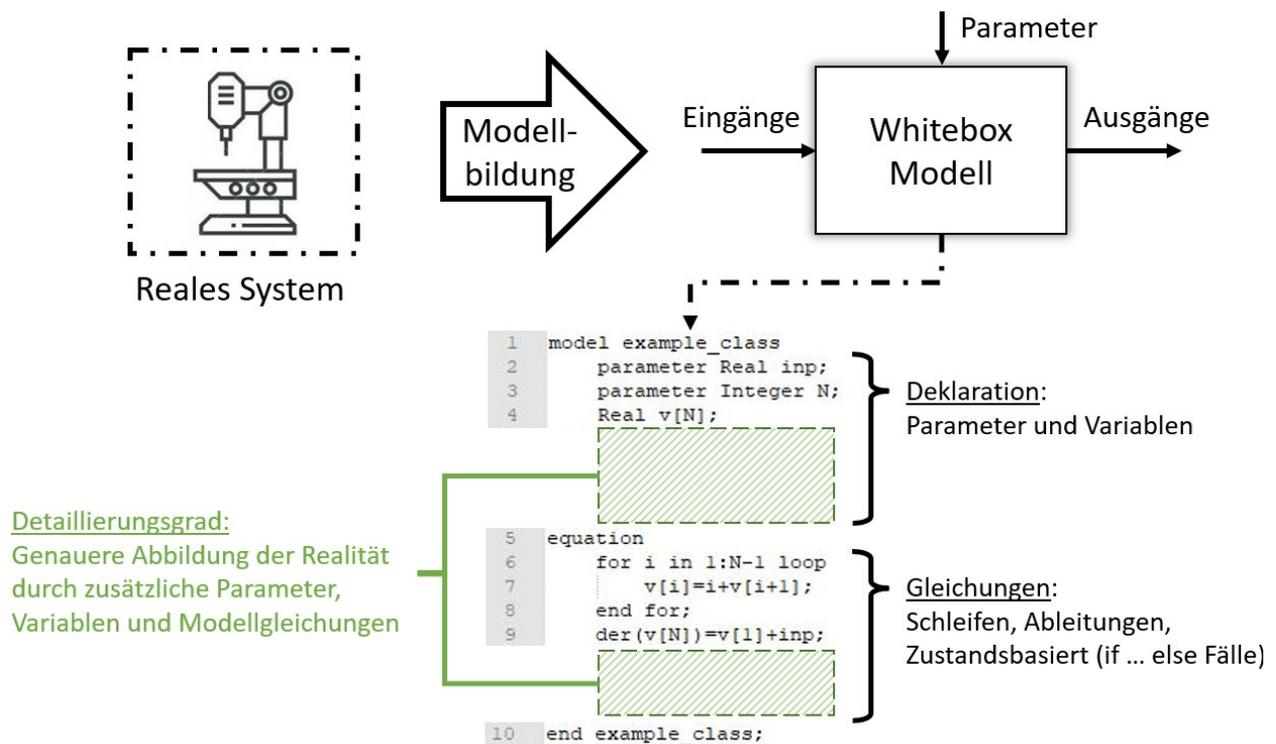


Abbildung 7: Schematische Darstellung der Modellbildung aus Zulieferersicht anhand eines einfachen Beispiels.

Aus Sicht des Zulieferers ist die Modellbildung abgeschlossen. Um seine Expertise und das Systemwissen gegenüber dem Anlagenbetreiber zu schützen, wandelt er in einem letzten Schritt, z. B. durch Kompilierung des Modellcodes, das Whitebox-Modell in ein Blackbox-Modell um. Eine Einsicht in die Modellgleichungen und die Manipulation dieser ist aus Sicht des Anlagenbetreibers nun nicht mehr möglich. Einzig über die Parameter kann er das Modell, innerhalb der durch den Zulieferer vorgegebenen Grenzen, manipulieren. Der Betreiber kann das Modell jedoch unverändert nutzen und über die Schnittstellen einbinden. Mittels dieses Modells lässt sich das Verhalten des Moduls „Bohrstation“ beschreiben und über die Ein- und Ausgänge lässt es sich im Rahmen der Aggregation zu einem Gesamtmodell einer modularen Produktionsanlage verschalten.

In der Prognose von Systemeigenschaften und Systemverhalten liegt nach [VDI/VDE 2206] der Hauptvorteil modellbasierter Ansätze. Um einen konsistenten, redundanzfreien Informationsaustausch zu ermöglichen, werden im Kontext der modellbasierten Entwicklung und des Model-Based Systems Engineerings (MBSE) die Modelle der Disziplinen mit den Modellen auf Systemebene verknüpft. Die Anwendung von MBSE ermöglicht nach [VDI/VDE 2206] die Umstellung von dokumentenzentriertem zum modellbasierten

Engineering. Durch die disziplinübergreifende und gezielte Bereitstellung von Informationen an verschiedene Personen(gruppen), wie z. B Kunden, Systemingenieure und Tester, wird die Kommunikation der an der Systementwicklung beteiligten Instanzen erleichtert und der Engineering-Prozess insgesamt verbessert. Das betrachtete System in Bezug auf Konsistenz, Korrektheit und Vollständigkeit zu bewerten, wird zudem durch ein durchgängiges Modell ermöglicht. [VDI/VDE 2206]

2.1.5.4 Grundlagen zu Simulation

Modelle, die im Rahmen der Modellbildung entstanden sind, können anschließend im Rahmen einer computergestützten Simulation genutzt werden. In [VDI 3633-1] wird Simulation als „*Nachbildung eines Systems mit seinen dynamischen Prozessen in einem experimentierfähigen Modell [beschrieben], um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind*“. In [MBG+21] wird Simulation als „*Abbildung und Ausführung realer Szenarien mit Bezug zum betrachteten Zielsystem in Simulationsmodellen [definiert], um diese Szenarien zu verstehen, vorherzusagen oder zu optimieren*“. Um mittels Simulation Erkenntnisse über das betrachtete System zu erlangen, müssen Simulationsexperimente durchgeführt werden [BER13]. Ein Simulationsexperiment wird in [VDI 3633-1] als „*die gezielte empirische Untersuchung des Verhaltens eines Modells durch wiederholte Simulationsläufe mit systematischer Parametervariation oder Strukturvariation*“ definiert. Aufgrund der Komplexität der betrachteten Systeme und Modelle ist es sinnvoll, Simulationsexperimente methodisch durchzuführen. Hierfür bieten sich Simulationsstudien an. Exemplarisch wird an dieser Stelle das Vorgehensmodell zur Durchführung einer Simulationsstudie nach dem VDI-Standard 3633 vorgestellt, da es das „im deutschsprachigen Raum wohl am häufigsten genutzte Vorgehensmodell“ darstellt [BER13]. Das Vorgehensmodell, dargestellt in Abbildung 8, umfasst die drei Hauptsegmente Vorbereitung, Durchführung und Auswertung. Bergmann [BER13] merkt an, dass im Rahmen des Vorgehensmodells „*Simulation im Rahmen eines mehr oder weniger abgeschlossenen Projektes zum Einsatz kommt [und] ein Weiternutzen oder zyklisches Einsetzen des Modells, z. B. im operativen Betrieb zur Steuerung, [...] nicht vorgesehen*“ ist. Weiter sind auch „*Reaktionen zur Anpassung (Adaption) des Modells an während der Projektzeit auftretende Änderungen /Konkretisierungen des zu modellierenden Realsystems [...] nicht Teil des Vorgehens*“.

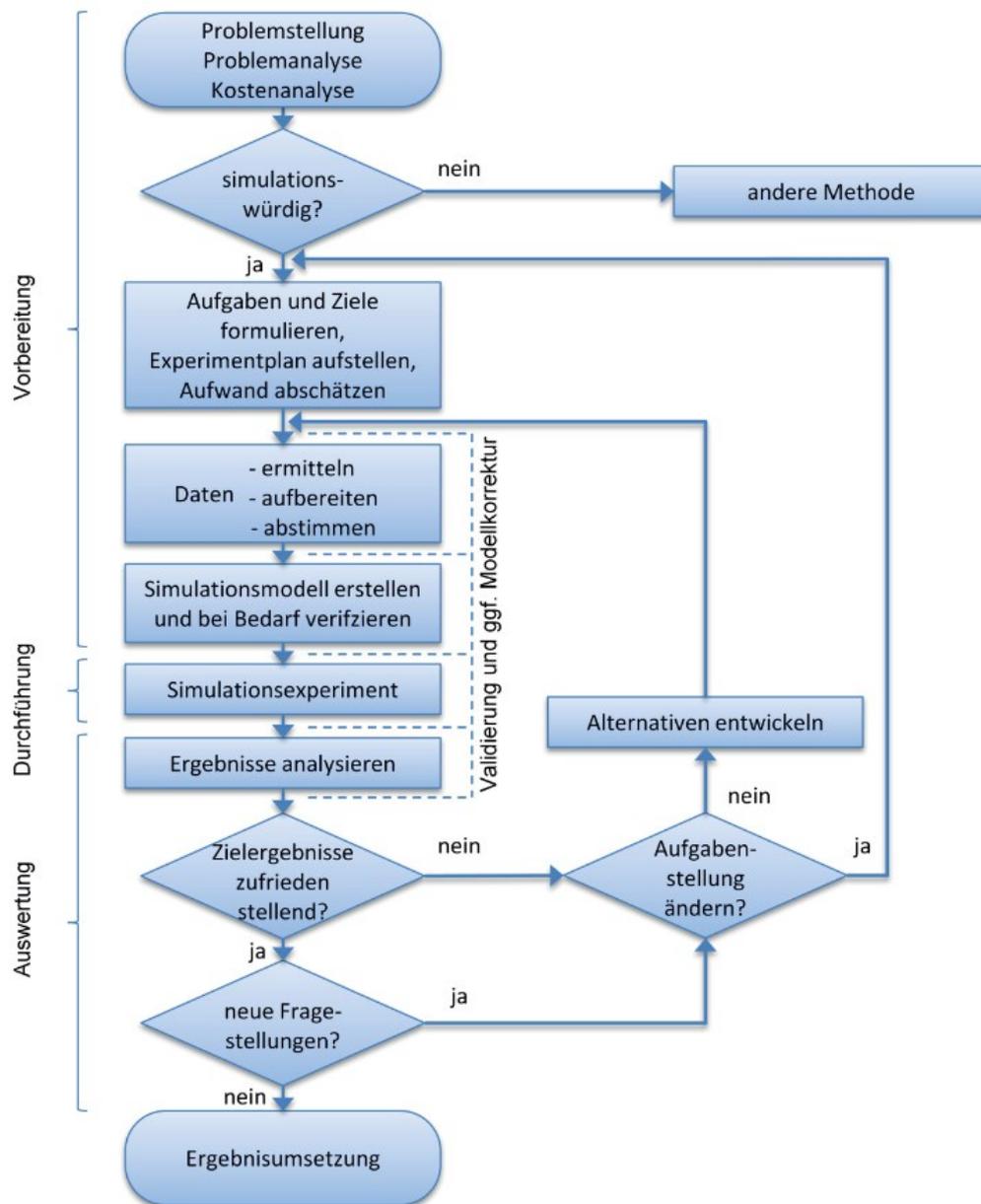


Abbildung 8: Vorgehensmodell zur Durchführung einer Simulationsstudie nach [BER13] in Anlehnung an [VDI 3633-1]

Der Block „Daten“, welcher in der Vorbereitungsphase einer Simulationsstudie relevant ist (vgl. Abbildung 8), stellt die notwendigen Eingangsdaten für Simulationsstudien dar. Diese sind nach [VDI 3633-1] in technischen Daten, Organisationsdaten und Systemlastdaten zu differenzieren. Abbildung 9 visualisiert die Quellen der Simulationsdatenbasis und listet für die verschiedenen Datenquellen Beispiele auf. Es wird ersichtlich, dass in der betrieblichen Praxis eine Vielzahl möglicher Datenquellen existiert [BS20a].

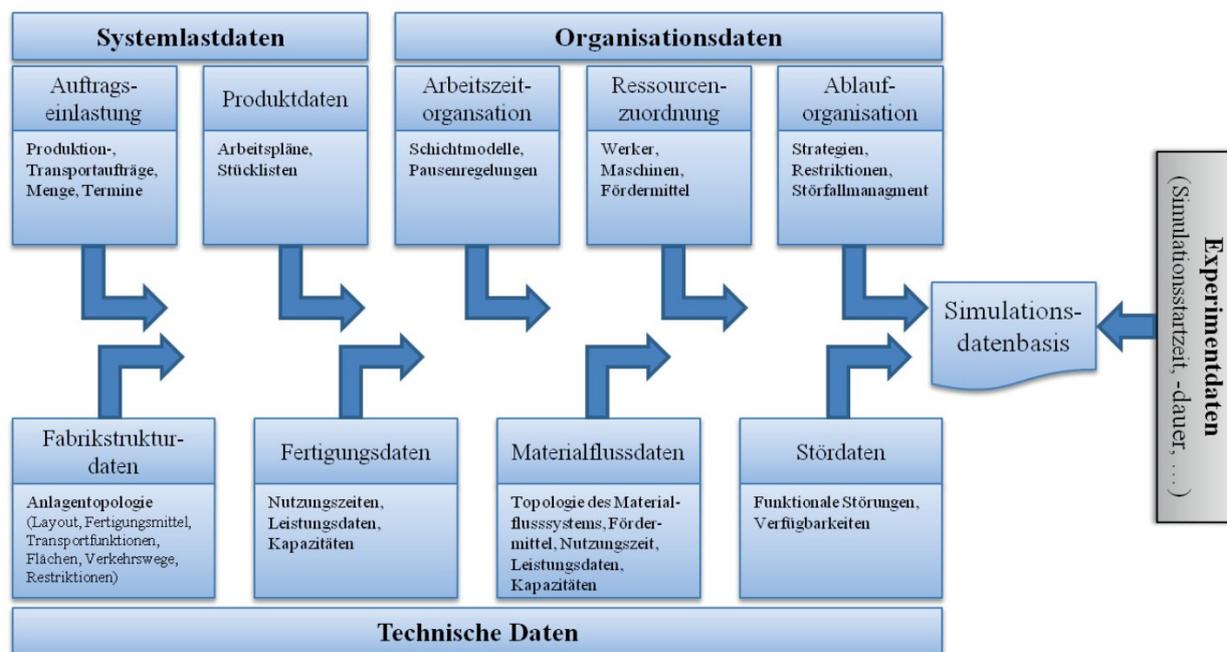


Abbildung 9: Eingangsdaten der Simulation nach [BER13] in Anlehnung an [VDI 3633-1].

Im nächsten Unterabschnitt wird aufgezeigt, welche Modellierungssprachen sich für mechatronische Systeme eignen.

2.1.5.5 Multiphysikalische Modellierungssprachen

Meinl [MEI19] schätzt Simulation auf Basis von multiphysikalischen Modellierungssprachen als geeignete Validierungsmethode für mechatronische Systeme ein. Diese sind in der Lage, „das Zusammenspiel der heterogenen, im Produkt komponierten Baugruppen und Komponenten darzustellen“. Heterogene Modelle aus allen Engineering-Domänen können über die Simulationsarchitektur „Co-Simulation“ mittels offener zugänglicher Schnittstellen in die Simulation eingebunden werden [MEI19]. In der Vorstellung benötigter technologischer Konzeptbausteine in Kapitel 4 wird Co-Simulation im Detail untersucht und betrachtet. In Abbildung 10 ist eine Zuordnung von Simulationsmethoden zu unterschiedlichen Anwendungsfeldern nach [MEI19] abgebildet. Zunächst kann eine Kategorisierung von Simulationsmodellen in „kausale“ und „nichtkausale“ Modelle vorgenommen werden. Kausale Modellbeschreibungen zeichnen sich durch einen gerichteten Informationsfluss aus. Als Beispiel sind Signalblöcke bei der Modellierung von Regelungs- oder Steuerprozessen zu nennen. Die realitätsnahe Modellierung physischer Verbindungen mechatronischer Systeme lässt sich mittels bidirektionaler Interaktionen zwischen Modellfunktionen bzw. -objekten realisieren. Diese bidirektionalen Verbindungen sind die Hauptcharakteristik nichtkausaler Modellbeschreibungen und ermöglichen die modellseitige Abbildung modularer Systeme. Derartige Verbindungen von Modellen werden von multiphysikalischen Modellierungssprachen angeboten. [KMP+08]

Die nächste Stufe der Untergliederung in Abbildung 10 stellt die Unterteilung in ereignisdiskrete und kontinuierliche Simulation dar [MBG+21]. Komplexe Abläufe, wie beispielsweise Materialflussprozesse zwischen Produktionsanlagen oder auch Regelungs- und Steuerungsprozesse, können durch ereignisdiskrete Simulationen abgebildet werden. Der Zustand des Modells hängt in diesem Fall von Aktionen ab, die an Zeiträume und andere Aktionen geknüpft sind. Für die Darstellung physikalischer Prozesse eignen sich

hingegen kontinuierliche Simulationen besser. Systemgrößen, welche als Variablen von Gleichungssystemen formuliert sind, werden über vorgegebene Anfangsbedingungen und Zeiträume numerisch berechnet. Dies ermöglicht die Nachstellung beliebiger physikalischer Prozesse. Vielerlei eindimensionale, physikalische Modelle werden von multiphysikalischen Modellierungssprachen angeboten, was zur Darstellung des Zusammenspiels von Baugruppen und Komponenten meistens ausreicht [MEI19]. Multidisziplinäre Simulationsmodelle, die multiphysikalisches Verhalten abbilden können, werden in [MBG+21] als einer der zentralen Befähiger für Industrie 4.0 angesehen. Der Fokus im weiteren Verlauf der Arbeit wird daher auf diese Modellart gelegt.

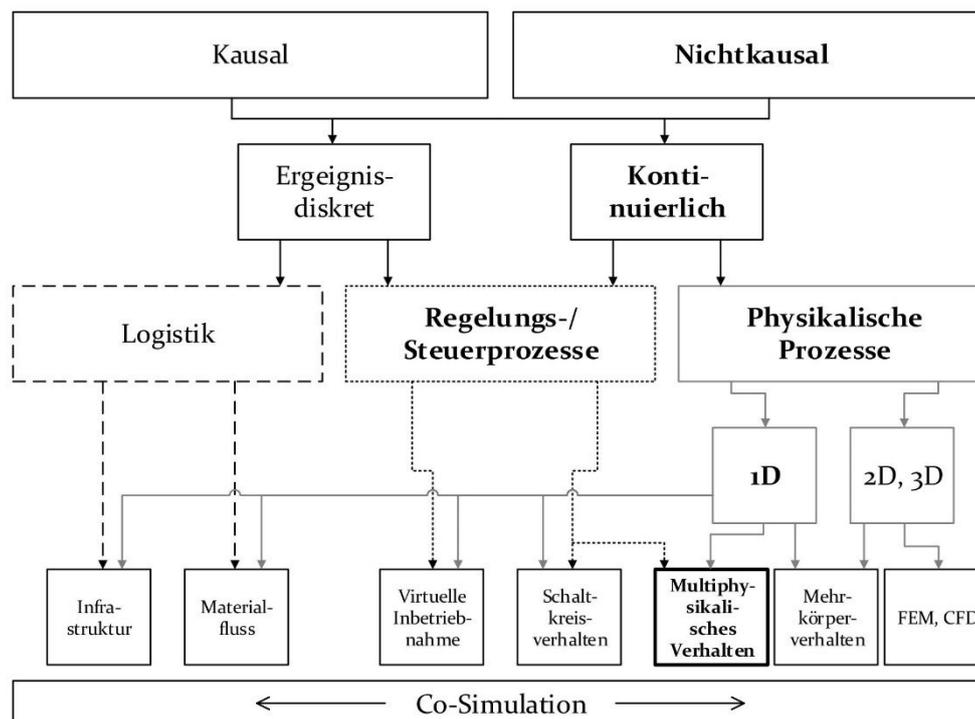


Abbildung 10: Durch Simulation adressierte Herausforderungen des Engineerings nach [MEI19].

Schluse et al. [SAR17] identifizieren MBSE und Simulationstechnik als wichtige Bausteine, um die immer komplexer werdenden technischen Systeme zu beherrschen. Doch mit der zunehmenden Komplexität der Systeme wachsen auch die Systemmodelle sowie die Simulationen zur Verifizierung, Validierung und Optimierung der Systementwürfe kontinuierlich [SAR17]. Für wandelbare Systeme erachtet die Plattform Industrie 4.0 [ABD+16] die Weiterentwicklung des Systems Engineering von „der heutigen Sammlung von ‚Best Practices‘ zu einer disziplinenübergreifenden Methodik“ als vordringliche Forschungsfrage. Für „die durchgängige Nutzung von Modellen auf System- und Komponentenebene von den frühen Phasen des Entwurfs über die Detailplanung und weiter über den gesamten Lebenszyklus“ werden daher in [ABD+16] entsprechende Methoden gefordert. Im Folgenden werden die Herausforderungen beim Einsatz von Simulation im Allgemeinen und speziell im Kontext der Modularisierung aufgezeigt.

2.2 Simulation und ihre Anwendungen im Maschinen- und Anlagenbau

Dieser Abschnitt gliedert sich in fünf Unterabschnitte. Zunächst werden die Herausforderungen beim Einsatz von Simulation über die verschiedenen Lebenszyklusphasen hinweg dargestellt. Hieraus wird der Bedarf eines neuen Modellierungs- und Simulationsworkflows abgeleitet. Im dritten Unterabschnitt werden

Assistenzsysteme als Werkzeug für die Umsetzung des neuen Workflows vorgestellt. Anschließend werden die Anforderungen formuliert, die dieser Workflow erfüllen muss, um zukünftig modellbasiertes Systems Engineering und Simulationstechnik im Bereich modularer Maschinen und Anlagen lebenszyklusübergreifend beherrschbar zu machen. Im letzten Abschnitt werden Automatisierungslevel definiert, anhand derer die im Rahmen der vorliegenden Arbeit entwickelte Lösung bewertet werden kann.

2.2.1 Herausforderungen beim Einsatz von Simulation

Simulation bildet, nicht zuletzt durch erfolgskritische Einsatzfelder wie die Virtuelle Inbetriebnahme (u. a. zum Aufdecken und Beheben von Fehlern im Steuerungscode, von Kommunikationsfehler und von Fehlverhalten einzelner Gerätekonfigurationen [BPH+15]), ein wesentliches Werkzeug im Engineering von Maschinen und Anlagen [OWB+15, VDI/VDE 3693-1]. Auch während des Betriebs der Anlage werden mit Simulationsmethoden hilfreiche Informationen ermittelt. So können eventuelle Abweichungen zwischen sensorisch erfassten und simulierten Größen, die auf potenzielle Fehler schließen lassen, festgestellt und eine Diagnose eingeleitet werden. Diese Methode wird u. a. zur Identifikation von Komponentenverschleißerscheinungen, z. B. bei Elektromotoren, verwendet [GMA20]. Alternativ können, wie beispielsweise von Donhauser [DON20] vorgeschlagen, auf Basis von Simulationsmodellen und Echtzeitdaten hinsichtlich des Ressourcenverbrauchs optimierte Fertigungsabläufe generiert werden.

Der zunehmende Einsatz von Simulationsmodellen führt jedoch zu neuen, zusätzlichen Aufgaben im Engineering. Hierzu zählen u. a. die Modellierung und Parametrierung der Modulmodelle sowie deren Koppelung zu einem Gesamtmodell. In [VDI/VDE 3693-2] werden über 20 neue Engineering-Aktivitäten im Zusammenhang mit Simulation genannt, die größtenteils manuell durchgeführt werden müssen. Die Herausforderungen, die sich dabei im Allgemeinen und speziell für modulare Produktionsanlagen ergeben, sowie die daraus resultierenden Konsequenzen sind in den folgenden Unterabschnitten aufgezeigt.

2.2.1.1 Unternehmenspolitische und technologische Herausforderungen:

Mit dem Einsatz von Simulation entstehen auch unternehmenspolitische Fragestellungen. Die realen Anlagenmodule stammen teilweise oder vollständig von Zuliefererbetrieben [PB07]. Diese Betriebe sind bestrebt, das eigene Knowhow, welches in deren Simulationsmodellen abgebildet werden muss, zu schützen. Ein Kompromiss, um dem Anlagenbetreiber dennoch Modulmodelle für eine Gesamtsystemsimulation zur Verfügung zu stellen, sind Black-Box-Modelle. Diese können zur Simulation des Ein-/Ausgangsverhaltens verwendet werden, gewähren jedoch durch vorgeschaltete Mechanismen, wie z. B. die Kompilierung des Modellcodes, keine Einsicht in die Modellierung selbst [BU16].

Da es sich bei den eingesetzten Modulen um mechatronische Systeme handelt, stammen deren Modelle aus verschiedenen physikalischen Disziplinen (z. B. Mechanik, Elektrik, Thermodynamik) [ZMP13], die jeweils spezialisierte Modellierungsansätze, Simulationstools und Solver benötigen. Auf mögliche Tools und Solver wird in Abschnitt 4.2.2 im Detail eingegangen. Diese herausfordernde Kombination von:

- Black-Box-Modellen (Zulieferer schützen ihr Wissen)
- aus unterschiedlichen Quellen (von verschiedenen Zulieferern)
- mit unterschiedlichen Solvern, sowie
- der Abbildung heterogener mechatronischer Disziplinen

wird in der simulationstechnischen Disziplin der Co-Simulation behandelt [PVL+17]. Hierauf wird in Abschnitt 4.2 im Detail eingegangen.

Folglich müssen Entwicklungsingenieure im Bereich M&S von modularen Anlagen MSM, deren interne Struktur sie nicht kennen und welche zudem aus unterschiedlichen Quellen stammen, zu einem Modell der Gesamtanlage kombinieren und dieses über den Lebenszyklus der Anlage aktuell halten.

Welche Schritte hierzu nötig sind, wird im nächsten Unterabschnitt aufgezeigt.

2.2.1.2 Herausforderungen vor und während der Inbetriebnahme

Der Engineering-Aufwand für den Aufbau einer Co-Simulation steigt entlang der Lieferkette und mündet in einem beträchtlichen Aufwand beim OEM [CBR+18]. Dies ist dadurch zu begründen, dass entlang der Lieferkette immer mehr MSM in die Co-Simulation integriert, gekoppelt und parametrisiert werden müssen. Zudem ist anzunehmen, dass verschiedene Interessengruppen unterschiedliche Simulationsergebnisse benötigen. Für die Geschäftsleitung ist beispielsweise eine Simulation zur Ermittlung der Produktionsrate relevant. Einen Steuerungstechniker interessiert hingegen das Maschinenverhalten mit neuem Steuerungscode und eine Untersuchung auf mögliche Kollisionsfahrten. Es ist daher notwendig, je nach Interessengruppe entsprechend detaillierte und ausgeprägte Co-Simulationen aufzubauen.

Um Co-Simulationen aufzubauen, müssen Entwicklungsingenieure im Bereich M&S von modularen Anlagen die benötigten MSM zunächst aggregieren. Anschließend müssen sie

- **alle MSM in eine geeignete Simulationsumgebung laden,**
- **die einzelnen MSM entsprechend der Topologie der realen Anlage verschalten. Hierzu müssen die richtigen Ein- und Ausgänge der MSM gekoppelt werden,**
- **die Parameter der MSM in der Simulationsumgebung eingestellt werden,**
- **die Parameter der Simulation (u. a. Dauer, Solver, Schrittweite, Toleranz) vorgegeben werden.**

Besonders die Kopplung der MSM sowie die Konfiguration der Co-Simulation sind, wie von Stecken et al. [SLK19] beschrieben, mit hohem manuellen Aufwand verbunden. Es stellt sich daher die Frage, welche dieser Tätigkeiten teilweise oder vollständig automatisiert werden können. Einige der M&S-Tätigkeiten sind komplex und stark vom betrachteten Produktionssystem abhängig. Als Beispiel können, wie im nächsten Unterabschnitt aufgezeigt, die Wahl geeigneter Modellparameter und -detaillierungsgrade für eine betriebsbegleitende Simulation genannt werden. Eine Automatisierung dieser Tätigkeiten ist nur mit großem Aufwand möglich. In diesem Fall ist es sinnvoll, dem Entwicklungspersonal automatisiert Vorschläge und Entscheidungshilfen anzubieten.

2.2.1.3 Herausforderungen während des Betriebs

Der Engineering-Aufwand beschränkt sich, wie bereits aufgezeigt, nicht nur auf die Phase der Inbetriebnahme und somit auf den Aufbau einer Co-Simulation. Entsprechend müssen die Entwicklungsingenieure im Bereich M&S von modularen Produktionsanlagen auch in diesen Phasen (Betrieb und Umbau) aktiv werden. Falls die Simulationsergebnisse während des Betriebs nicht zu den in der Anlage sensortechnisch erfassten Daten passen, muss ggf. eine Adaption des Gesamtmodells vorgenommen werden. Unter einer

Adaption wird im Rahmen der vorliegenden Arbeit entsprechend [BER13] „die Anpassung von Modellen an veränderte Bedingungen ohne vollständigen Neuaufbau“ verstanden. Hierauf wird in Abschnitt 5.2.3 im Detail eingegangen. Dieser Fall, dass die Datensätze nicht ausreichend übereinstimmen, kann auch ohne Rekonfiguration der Anlage auftreten, wenn beispielsweise die Maschine sukzessive verschleißt oder wenn andere Rohstoffe (z. B. Werkstücke oder Materialien mit geringfügig anderen stofflichen Eigenschaften) eingesetzt werden. In diesen Fällen ist kein Neuaufbau der Co-Simulation (MSM laden und koppeln) nötig, allerdings ist eine Anpassung der Modellparameter, z. B. besagter Materialeigenschaften der Werkstücke, und / oder der Detaillierungsgrade erforderlich.

Die Komponenten einer betriebsbegleitenden Simulation nach [VDI 4499-2] sind in Abbildung 11 dargestellt. Wird im beschriebenen Verschleißszenario die Simulation ohne Adaption der Simulationsmodelle weitergeführt, hat das Einfluss auf die zu erwartenden Produktionsergebnisse. Diese passen nicht mehr zu den Produktionszielen. In diesem Fall kommandiert die Produktionsplanung und -steuerung neue Produktionsparameter oder Steuerungsstrategien. Dies kann sich negativ auf die Produktionsergebnisse auswirken.

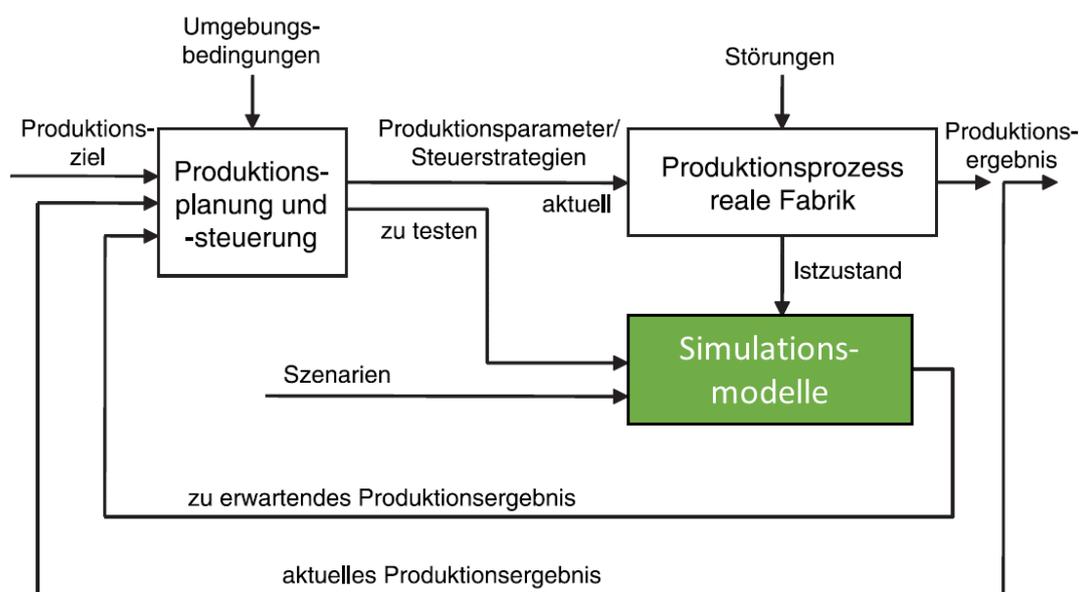


Abbildung 11: Komponenten einer betriebsbegleitenden Simulation nach [VDI 4499-2].

Um Co-Simulationen während der Betriebsphase aktuell zu halten, müssen Entwicklungingenieure im Bereich M&S von modularen Anlagen

- Mess- und Simulationsdaten vergleichen.
- diese Daten auf Abweichungen, die außerhalb einer von Interessensgruppen vorgegebenen Toleranz liegen, untersuchen.
- bei Überschreitung dieser Toleranz mögliche Ursachen für die Abweichung in den MSM lokalisieren.
- eine Adaption der betroffenen MSM durchführen.

Aufgaben wie der Datenvergleich und die Überprüfung auf die Überschreitung einer vorgegebenen Toleranz lassen sich leicht realisieren und auch automatisieren. Die Lokalisierung und Identifikation einer möglichen Ursache in einem MSM gestaltet sich dagegen schon schwieriger. In Unterabschnitt 2.1.5.2 wurde

bereits aufgezeigt, dass eine zu starke Reduktion innerhalb der Teilmodelle zu Abweichungen im Gesamtmodell führt, die für jedes Teilmodell einzeln betrachtet noch als tolerierbar einzustufen wäre. Zudem besteht durch die Kopplung der MSM eine Wechselwirkung, wodurch der negative Einfluss von einem MSM durch das Gesamtmodell propagieren kann. Mit steigender Anzahl an MSM sowie deren Parameter und Detaillierungsgrade steigt die Anzahl an möglichen Konfigurationen des Gesamtmodells schnell an.

Mögliche Konfigurationen des Gesamtmodells werden im Rahmen dieser Arbeit folgendermaßen aufgefasst. Die Topologie des Gesamtmodells (welche MSM sind enthalten und wie sind diese gekoppelt) ist durch das TM definiert und für alle Konfigurationen identisch. Variabel ist hingegen zum einen, wie die Detaillierungsgrade für die einzelnen MSM gewählt werden und zum anderen, wie die Parameter für die einzelnen MSM eingestellt werden. Somit unterscheidet sich eine Konfiguration (a) von einer zweiten Konfiguration (b) hinsichtlich der Detaillierungsgrade sowie der Parametereinstellungen für jedes im Gesamtmodell enthaltene MSM. Dieser Zusammenhang wird in Abschnitt 5.3.4, speziell in Abbildung 35, nochmals eingehend behandelt.

Die schnell steigende Anzahl an möglichen Konfigurationen des Gesamtmodells soll durch ein einfaches Rechenbeispiel verdeutlicht werden. Angenommen, die zu modellierende Anlage besteht aus vier Modulen, welche wiederum jeweils MSM in zwei Detaillierungsgraden besitzen. Hieraus ergeben sich $2^4 = 16$ Möglichkeiten für die Einstellung der Detaillierungsgrade. Wenn wiederum für jedes Modul zwei Parameter (somit insgesamt 8 Stück) mit jeweils 10 möglichen Werten gewählt werden können, ergeben sich insgesamt $2^4 * 10^8 = 1.600.000.000$ Möglichkeiten für die Einstellung der Parameter und Detaillierungsgrade der MSM. Nimmt man nun weiter an, dass ein Parameter beispielsweise den Reibungskoeffizienten in einem Lager μ_{Lager} darstellt und für dessen Grenzen $\mu_{Lager} \in [0,2 ; 0,25]$ gilt, dann wären, dargestellt als Variable vom Typ *real*, weit mehr als 10 Werte für diesen Parameter zulässig.

Hieraus geht hervor, dass sich bereits für kleinere Anlagenkonfigurationen schnell ein großer Suchraum ergibt. Entsprechend benötigt das Entwicklungspersonal softwaregestützte Lösungsvorschläge und Unterstützung bei der Entscheidungsfindung.

2.2.1.4 Herausforderungen durch einen Anlagenumbau

Wie in Abschnitt 2.1.2 aufgezeigt, können markt- und produktbedingte Anforderungen den Umbau einer Anlage erfordern. Der Anlagenbetrieb wird hierfür gestoppt und es erfolgt eine Rekonfiguration der Anlage. Im Rahmen dieser Arbeit wird unter Rekonfiguration der „Prozess [...] der Veränderung eines bereits entwickelten und operativ eingesetzten Systems [verstanden], um es an neue Anforderungen anzupassen, Funktionalität zu erweitern, Fehler zu beseitigen oder die Qualitätseigenschaften zu verbessern.“ [MAT10] Eine Rekonfiguration der realen Anlage macht es, wie zur Inbetriebnahmephase, notwendig, Modulmodelle zu instanzieren und das Gesamtmodell, entsprechend dem realen Pendant, aus bestehenden und neuen Modulen zusammenzustellen. Dies beinhaltet die Aufgaben, die bereits in Abschnitt 2.2.1.2 beschrieben wurden. Zunächst müssen die Entwicklungsingenieure im Bereich M&S von modularen Produktionsanlagen jedoch analysieren und feststellen, welche Änderungen durchzuführen sind bzw. bereits durchgeführt wurden. Nach der Modellierung müssen diese Änderungen festgehalten werden, damit z. B. in einem Wartungsfall nicht von der alten Anlagenkonfiguration ausgegangen wird und falsche Ersatzteile bestellt werden.

Um Co-Simulationen im Rahmen eines Anlagenumbaus zu aktualisieren, müssen Entwicklungsingenieure im Bereich M&S von modularen Anlagen

- **feststellen, was sich an der realen Systemkonfiguration geändert hat bzw. ändern wird (z. B. Modultausch, Entfernung Modul, zusätzliches Modul hinzugefügt),**
- **die automatische Modellgenerierung erneut durchführen und Umbauänderungen einzubeziehen,**
- **durchgeführte Änderungen an der Anlagentopologie dokumentieren.**

Die Konsequenzen aus den in den letzten Unterabschnitten aufgezeigten Herausforderungen werden im Folgenden zusammengefasst.

2.2.1.5 Konsequenzen aus diesen Herausforderungen

Die über den gesamten Anlagenlebenszyklus hinweg notwendige Generierung, Adaption und (Re)Konfiguration des Anlagenmodells ist mit manuellen Aufwänden und entsprechenden Kosten verbunden. Zudem steigt mit zunehmender Komplexität der Produktionsanlage auch die Komplexität der im Zusammenhang mit Simulation beschriebenen Aufgaben, was zu potenziellen Fehlern führen kann. Sollten Änderungen der Anlagenkonfiguration nicht sorgfältig in die Simulation eingepflegt werden, nehmen Abweichungen zwischen Mess- und Simulationsdaten zu. Das resultiert in der Verringerung der Simulationsgüte bzw. der Aussagekraft der Simulationsergebnisse. Im Extremfall werden die Abweichungen zwischen den Mess- und Simulationsdaten derart groß, dass die Simulation unbrauchbar wird.

In Kapitel 3 wird aufgezeigt, dass bereits zahlreiche Ansätze zur (Teil-)Automatisierung von notwendigen manuellen Tätigkeiten im Zusammenhang mit dem Einsatz von Simulationsmodellen existieren. Diese Arbeiten fokussieren jedoch häufig auf eine einmalige statische Modellgenerierung [BAR11, OWD+14] auf Basis homogener Modellbibliotheken, die aus einer einzigen Quelle in einem einzigen Simulationswerkzeug stammen. Es besteht dahingehend Handlungsbedarf, den erweiterten Bereich der Co-Simulationen mit ihren realen Gegenständen zu synchronisieren und die Konvergenz der physischen und digitalen Welt über deren kompletten Lebenszyklus sicherzustellen [GMA20].

2.2.2 Bedarf eines neuen Modellierungs- und Simulationsworkflows

Der vorherige Abschnitt gibt Aufschluss darüber, welchen vielfältigen Herausforderungen und Problemen das Entwicklungspersonal im Bereich M&S von modularen Produktionsanlagen begegnet. Um den Faktor Mensch als Fehler- und Kostenquelle bei Modellierungsarbeiten künftig zu reduzieren, ist es notwendig, die damit verbundenen Arbeitsschritte und Vorgänge möglichst zu automatisieren bzw. den Menschen zu assistieren. Da modulare Produktionsanlagen häufig Sonderlösungen sind [RBB+14], bleibt der Mensch als Hauptakteur und Kontrollinstanz weiterhin relevant. Vielmehr geht es darum, fehleranfällige Routinearbeiten zu reduzieren und den Menschen bei der Entscheidungsfindung zu assistieren. Hierfür ist es sinnvoll, bestehende Daten aus vorherigen Engineering-Phasen zu nutzen, um darauf aufbauend (teil-)automatisch Modelle zu generieren und Simulationen durchführen zu lassen. Auch in den weiteren Lebenszyklusphasen kann die automatische Auswertung z. B. von Sensordaten genutzt werden, um die Modelle zur Laufzeit oder nach einem Umbau anzupassen.

Es stellt sich nun die Frage, wie das Entwicklungspersonal im Bereich M&S bei diesen Tätigkeiten unterstützt werden kann. Hierfür soll zunächst untersucht werden, wie die Unterstützung bei manuellen Tätigkeiten und Entscheidungen in anderen Bereichen des Engineerings erfolgt. Stark et al. [SBL21] zeigen in ihrer Arbeit auf, dass, bezogen auf das Engineering, sowohl in der Praxis als auch in der einschlägigen Literatur ein breites Spektrum an Assistenzsystemen (AS) zu finden ist, welche die Entwickler bei unterschiedlichsten Tätigkeiten unterstützen. Exemplarisch für den Einsatz von AS im Engineering können an dieser Stelle die Arbeiten von Goldhammer und Woyand [GW11] sowie von Lang et al. [LWH+18] genannt werden, um die verschiedenen Einsatzbereiche aufzuzeigen. Goldhammer und Woyand stellen in ihrer Arbeit ein wissensbasiertes Design-AS vor, welches Konstrukteure beim Einsatz von CAD-Tools unterstützt und zur CAD-Automatisierung beiträgt [GW11]. Lang et al. stellen dagegen ein AS zur Unterstützung der Fehlersuche bei komplexen industriellen Systemen vor. Trotz der bereits zunehmenden Anzahl an AS identifizieren Stark et al. [SBL21] einen stetig wachsenden Bedarf an AS aller Art.

Es ist daher sinnvoll, dem Entwicklungspersonal im Bereich M&S ein Werkzeug in Form eines AS an die Hand zu geben. Das Konzept eines AS, welches diese Aufgaben erfüllt, stellt nach Design Science Research [WIE14] das Artefakt dieser Arbeit dar. Im nächsten Abschnitt soll zunächst eine Übersicht gegeben werden, wie Assistenzsysteme als Engineering-Werkzeuge eingesetzt werden können.

2.2.3 Assistenzsystem im Engineering

AS sind nach [VBT17] „*technische Systeme, welche zur Assistenz bzw. Unterstützung des Nutzers in bestimmten Situationen oder bei bestimmten Handlungen dienen*“. In Anlehnung an [REI17] werden in [BMV+19] Mitarbeiter-Assistenzsysteme als Systeme definiert, „*welche den Mitarbeiter bei seinen Handlungen unterstützen. Intelligent sind diese Systeme, wenn sie durch den Einsatz von Sensorik oder die Einbettung in übergeordnete IT-Systeme in der Lage sind, zu einem bestimmten Maß selbstständig auf Situation zu reagieren [...]*“. Weiter unterscheiden die Autoren intelligente Mitarbeiter-Assistenzsysteme in die drei Kategorien Wahrnehmungsassistenzsysteme (WAS), Entscheidungsassistenzsysteme (EAS) und Ausführungsassistenzsysteme (AAS). Die ersten beiden Kategorien können zu kognitiven Assistenzsystemen zusammengefasst werden. WAS unterstützen die Sinneswahrnehmung des Menschen, z. B. durch das Anzeigen von digitalen Arbeitsanweisungen mit Hilfe von Augmented Reality Anwendungen. EAS unterstützen bei der Aufnahme und Verarbeitung von Wissen und können so beispielsweise bei Problemlösungs- oder Lernaufgaben helfen. Beispielsweise könnte ein EAS mittels umfangreicher Datenanalyse Optionen für komplexe Entscheidungssituationen vorbereiten, welche der Mensch auf sich allein gestellt nicht optimal lösen kann. [REI17, BMV+19]

Als Beispielanwendungen für AAS werden in [REI17, BMV+19] ausschließlich physische Arbeiten in Betracht gezogen. Exemplarisch wird auf die ergonomische Arbeitsgestaltung und auf die Unterstützung von Bewegungsabläufen durch Exoskelette und Mensch-Roboter-Kooperation eingegangen. Die physische Assistenz von AAS lässt sich darüber hinaus auf, im Rahmen dieser Arbeit relevante, Modellierungs- und Simulationstätigkeiten übertragen und erweitern. Auch wenn bei diesen Tätigkeiten das Produkt rein digitaler Natur (in Form eines Modells oder von Simulationsergebnissen) ist, sind physische und kognitive Arbeitsschritte, beispielsweise das Einstellen von Modell- oder Simulationsparametern, nötig. Welche Ausprägung eines AS für den Modellierungs- und Simulationsworkflow im Rahmen des Lebenszyklus von modularen Produktionsanlagen nötig ist, wird im weiteren Verlauf der Arbeit herausgestellt.

Im nächsten Schritt stellt sich die Frage, welcher Automatisierungsgrad für das AS erreicht werden kann. Dies steht in direktem Zusammenhang mit Forschungsfrage 4 „Welcher Automatisierungsgrad kann für die einzelnen Lebenszyklusphasen erzielt werden?“ Eine Herangehensweise für die Einstufung des Grades stellt ein Vergleich mit den Automatisierungsleveln des autonomen Fahrens [J3016_202104] dar. Dieser Ansatz wird z. B. in den Arbeit von Gamer et al. [GKH+19] beschrieben, um eine Taxonomie für Produktionsanlagen abzuleiten. Auch in dem Arbeitspapier der Plattform Industrie 4.0 [BMW19] werden vergleichbare Level für KI-beeinflusste Autonomie-Stufen in der industriellen Produktion beschrieben. In [BMW19] werden die Level anhand eines Roboters für das Teilehandling exemplarisch beschrieben. Die Autoren nennen aber auch die Engineeringunterstützung als Beispiel für die Prozessautomatisierung, welche mit den Stufen kategorisiert werden kann.

Für automatisierte und autonome Kraftfahrzeuge hat sich die Taxonomie der Society of Automotive Engineers (SAE) [J3016_202104] in der internationalen Forschung und Praxis etabliert [RS20]. Diese unterscheidet zwischen den SAE-Leveln 0 bis 5, „Keine Automatisierung“, „Fahrassistenz“, „Teilautomatisierung“, „Bedingte Automatisierung“, „Hochautomatisierung“ und „Vollautomatisierung“ [J3016_202104]. Übertragen auf die M&S im Maschinen- und Anlagenbau würde ein höheres Automatisierungslevel bedeuten, dass weniger Eingriffe und Kontrolle durch das Bedienpersonal nötig sind und das AS komplexere Aufgaben (z. B. Kopplung, Adaption, Dokumentation) ausführen kann. Im Rahmen der SAE-Klassifizierung wird darüber hinaus definiert, in welchen Leveln der Fahrer oder das AS für Eingriffe (Lenkung, Beschleunigung, Bremsen), die Beobachtung des Fahrumfeldes sowie das dynamische Fahren (z. B. Wechseln von Fahrspuren, Abbiegen sowie Erkennen von Ampelschaltungen und Verkehrsschildern) zuständig ist. Bevor das im Rahmen dieser Arbeit entwickelte AS eingestuft werden kann, muss die SAE-Klassifizierung unter Berücksichtigung der Arbeiten von Gamer et al. [GKH+19] und der Plattform Industrie 4.0 [BMW19] an das in Kapitel 1 vorgestellte Ziel angepasst werden. Dies erfolgt in Abschnitt 2.2.5. Um diese Anpassung vornehmen zu können, müssen zunächst die Anforderungen an das AS definiert werden. Dies erfolgt im nächsten Abschnitt.

2.2.4 Abgeleitete Anforderungen an ein Assistenzsystem

Nachdem in Abschnitt 2.2.1 die Herausforderungen beim Einsatz von Simulation im modularen Anlagenbau aufgezeigt und einzelne Anforderungen abgeleitet wurden und in Abschnitt 2.2.2 der Bedarf eines neuen Modellierungs- und Simulationsworkflows mittels AS herausgestellt wurde, sollen in diesem Abschnitt die Anforderungen an ein lebenszyklusübergreifendes AS definiert und zusammengefasst werden. Die Anforderungen sind entsprechend der drei Lebenszyklusphasen in Tabelle 3 geordnet.

Tabelle 3: Anforderungen an ein Assistenzsystem für die automatische Modellgenerierung, -adaption und -rekonfiguration im modularen Maschinen- und Anlagenbau.

Lebenszyklusphase	Anforderungsnummer:	Anforderungsbeschreibung
Inbetriebnahme	A1	Das AS muss... ... in der Lage sein, selbstständig: <ul style="list-style-type: none"> • Modulmodelle zu laden, • Initialwerte/ Anfangszustände zu setzen, • die Modulmodelle, analog zur realen Systemkonfiguration, korrekt zu koppeln sowie • die Simulation zu initiieren und auszuführen

	A2	... derart konzipiert werden, dass eine Erweiterung des Einsatzbereichs auf den kompletten Anlagenlebenszyklus möglich ist.
Betrieb	A3	... in der Lage sein, selbstständig Simulations- und Messwerte, die zur Betriebsphase aufgenommen werden, miteinander vergleichen.
	A4	... in der Lage sein, selbstständig festzustellen, wenn eine Abweichung der in A3 beschriebenen Werte eine vorgegebene Toleranz überschreitet.
	A5	... in der Lage sein, selbstständig mögliche Ursachen für die in A4 erwähnte Abweichung in den Simulationsmodellen zu lokalisieren.
	A6	... in der Lage sein, eine Adaption der betroffenen Simulationsmodelle automatisch durchzuführen.
Umbau	A7	... in der Lage sein, selbstständig feststellen zu können, ob sich die reale Systemkonfiguration geändert hat (z. B. Modul ausgetauscht, Modul entfernt, zusätzliches Modul hinzugefügt)
	A8	... in der Lage sein, selbstständig die automatische Modellgenerierung (A1) erneut durchzuführen und Umbauänderungen einzubeziehen
	A9	... in der Lage sein, durchgeführte Änderungen an der Anlagentopologie automatisch zu dokumentieren.

Nach der Definition der Anforderungen an das AS können im nächsten Schritt die SAE-Automatisierungslevel auf die im Rahmen dieser Arbeit betrachtete Zielstellung angepasst werden.

2.2.5 Definition des Automatisierungslevels des Assistenzsystems

Im ersten Schritt ist es naheliegend, die Bezeichnungen der SAE-Klassifizierung auf das AS dieser Arbeit anzupassen. Die Aufgabe des AS liegt nicht in der Fahrassistenz, sondern in der Modellierungs- und Simulationsassistenz (M&S-Assistenz). Da bei den Bezeichnungen der SAE-Leveln ausschließlich bei Level 1 (Fahrassistenz) ein Bezug zu automatisierten und autonomen Kraftfahrzeugen besteht, können die restlichen Level-Bezeichnungen übernommen werden. Die sich ergebenden Level-Bezeichnungen sind der Tabelle 4 zu entnehmen.

Tabelle 4: Angepasste Klassifizierung der Automatisierungslevel eines Assistenzsystems.

Level	SAE-Klassifizierung	Klassifizierung im Rahmen dieser Arbeit
0	Keine Automatisierung	Keine Automatisierung
1	Fahrassistenz	<i>M&S-Assistenz</i>
2	Teilautomatisierung	Teilautomatisierung
3	Bedingte Automatisierung	Bedingte Automatisierung
4	Hochautomatisierung	Hochautomatisierung
5	Vollautomatisierung	Vollautomatisierung

Weiter ist die zu unterstützende Person in diesem Fall kein Fahrer, sondern der Bediener des AS, beispielsweise ein M&S-Ingenieur. Im nächsten Schritt gilt es, geeignete Beschreibungen für die Automatisierungslevel zu finden. Die Level 0 und 5 stellen dabei Grenzwerte für die zu erreichende Automatisierung durch das AS dar. Mit steigendem Level kommt es zu einer kontinuierlichen Veränderung der Verantwortung vom Menschen hin zum AS. Bei Level 0 werden alle M&S-Aufgaben durch einen Menschen bearbeitet, bei Level 5 werden entsprechend alle durch das AS übernommen. Dies beinhaltet, neben den aktiven Tätigkeiten entsprechend den Anforderungen aus Abschnitt 2.2.4, z. B. Modulmodelle zu laden oder Modellparameter einzustellen, auch die Wahrnehmungsfähigkeiten. In der SAE-Klassifikation bezieht sich die Wahrnehmung des AS auf die Erkennung und Überwachung des Fahrumfelds. Bezogen auf Abschnitt 2.2.3 ist Wahrnehmung für die Anforderungen A4 (Überwachung von Mess- und Simulationsdaten und Erkennung von Abweichungen) und A7 (Überwachung der Anlagentopologie und Erkennung von Umbaumaßnahmen) erforderlich. In [RS20] wird der Begriff „dynamische Fahraufgaben“ als weiteres Klassifizierungskriterium verwendet. Dieses umfasst komplexere Aufgaben, beispielsweise das Wechseln von Fahrspuren oder das Abbiegen, die zur Lösung Kombinationen aus den operativen Tätigkeiten und der Wahrnehmung nutzen. Übertragen auf das M&S-AS sind das die Tätigkeiten „Modellierung“, „Simulation“, „Adaption“, „Rekonfiguration“ und „Dokumentation“, die entlang der Lebenszyklusphasen auftreten. Die Tätigkeit „Modellierung“ ergibt sich beispielsweise durch die Summe aller operativen Tätigkeiten in diesem Bereich (MSM laden, koppeln und parametrieren). Diese komplexen Tätigkeiten werden im Rahmen der vorliegenden Arbeit als „Integrale Tätigkeiten“ zusammengefasst. Als letztes Kriterium nutzt [RS20] die „Systemfähigkeiten“, welche die einzelnen Fahrmodi des dynamischen Fahrens zusammenfasst. Übertragen auf das M&S-AS ist das die Summe der integralen Tätigkeiten. Nach der Anpassung der Begrifflichkeiten aus [J3016_202104, RS20] ergibt sich die in Tabelle 5 dargestellte Klassifizierung für das Automatisierungslevel eines M&S-AS.

In der ersten Spalte sind die Automatisierungslevel, in der zweiten die dazugehörigen Bezeichner eingetragen. Anschließend folgt die Beschreibung des Levels in der dritten Spalte. In den nächsten vier Spalten stehen die Tätigkeiten (operativ, integral und auf Systemebene) sowie die Wahrnehmungsfähigkeit. Zusätzlich sind noch die beiden dunkelgrau hervorgehobenen Zeilen zu beachten. Diese trennen die Level 0 bis 2 von den Leveln 3 bis 5. Bei den ersten 3 Leveln liegt die Wahrnehmung vollständig beim Bediener, bei den anderen Leveln beim AS.

Tabelle 5: SAE-Klassifizierung in Anlehnung an [J3016_202104, RS20] übertragen auf M&S-AS im Bereich modularer Maschinen und Anlagen.

Level	Bezeichnung	Beschreibung	Operative Tätigkeiten	Wahrnehmung	Integrale Tätigkeiten	Systemfähigkeiten
Bediener überwacht Mess- und Simulationsdaten sowie Anlagentopologie						
0	Keine Automatisierung	Ausführung aller Tätigkeiten durch den Bediener.	Bediener	Bediener	Bediener	Keine integralen Tätigkeiten
1	M&S-Assistenz	Ausführung einzelner operativer Tätigkeiten durch AS für spezifische Lebenszyklusphasen. Bedie-	Bediener und AS	Bediener	Bediener	Einige integrale Tätigkeiten

		ner stellt Wahrnehmungsinformationen bereit. Es gilt die Annahme, dass der Bediener alle anderen operativen Tätigkeiten übernimmt.				
2	Teilautomatisierung	Ausführung aller operativer Tätigkeiten durch AS für spezifische Lebenszyklusphasen. Bediener stellt Wahrnehmungsinformationen bereit. Es gilt die Annahme, dass der Bediener alle anderen operativen Tätigkeiten übernimmt.	AS	Bediener	Bediener	Einige integrale Tätigkeiten
AS überwacht Mess- und Simulationsdaten sowie Anlagentopologie						
3	Bedingte Automatisierung	Ausführung aller integraler Tätigkeiten durch AS. Es gilt die Annahme, dass der Bediener auf Aufforderungen zum Eingreifen angemessen reagiert.	AS	AS	Bediener	Einige integrale Tätigkeiten
4	Hochautomatisierung	Ausführung aller integraler Tätigkeiten durch AS, auch dann, wenn der Bediener nicht angemessen auf Aufforderungen zum Eingreifen reagiert.	AS	AS	AS	Einige integrale Tätigkeiten
5	Vollautomatisierung	Vollständige Übernahme aller integraler Tätigkeiten durch AS unter allen Rahmenbedingungen, die auch ein menschlicher Bediener bewältigen könnte.	AS	AS	AS	Alle integralen Tätigkeiten

Ein Vergleich der Metrik mit den von Gamer et al. [GKH+19] und der Plattform Industrie 4.0 [BMW19] erarbeiteten Ansätzen ergibt große Übereinstimmungen (z. B. Definition von 6 Levels, Überwachung durch das System ab Level 3, Verteilung der Verantwortung auf Mensch und System). Dies wird durch Abbildung 12, welche die Level (hier als Stufen bezeichnet) entsprechend des Plattform Industrie 4.0 Ansatzes [BMW19] darstellt, verdeutlicht. Die durchgängige Übereinstimmung ist dadurch zu begründen, dass sowohl die im Rahmen dieser Arbeit entwickelte Metrik als auch jene in den referenzierten Arbeiten von der SAE-Klassifizierung für autonomes Fahren abgeleitet wurden. Die vorgestellte Metrik unterscheidet sich lediglich dahingehend, dass bei der Ableitung konkret ein AS als System gewählt wurde und bei den Ansätzen von Gamer et al. und der Plattform Industrie 4.0 allgemein aus der Sicht der industriellen Produktion abgeleitet wurde. Somit handelt es sich bei der in Tabelle 7 erarbeiteten Metrik um einen Spezialfall dieser beiden allgemeineren Ansätze.

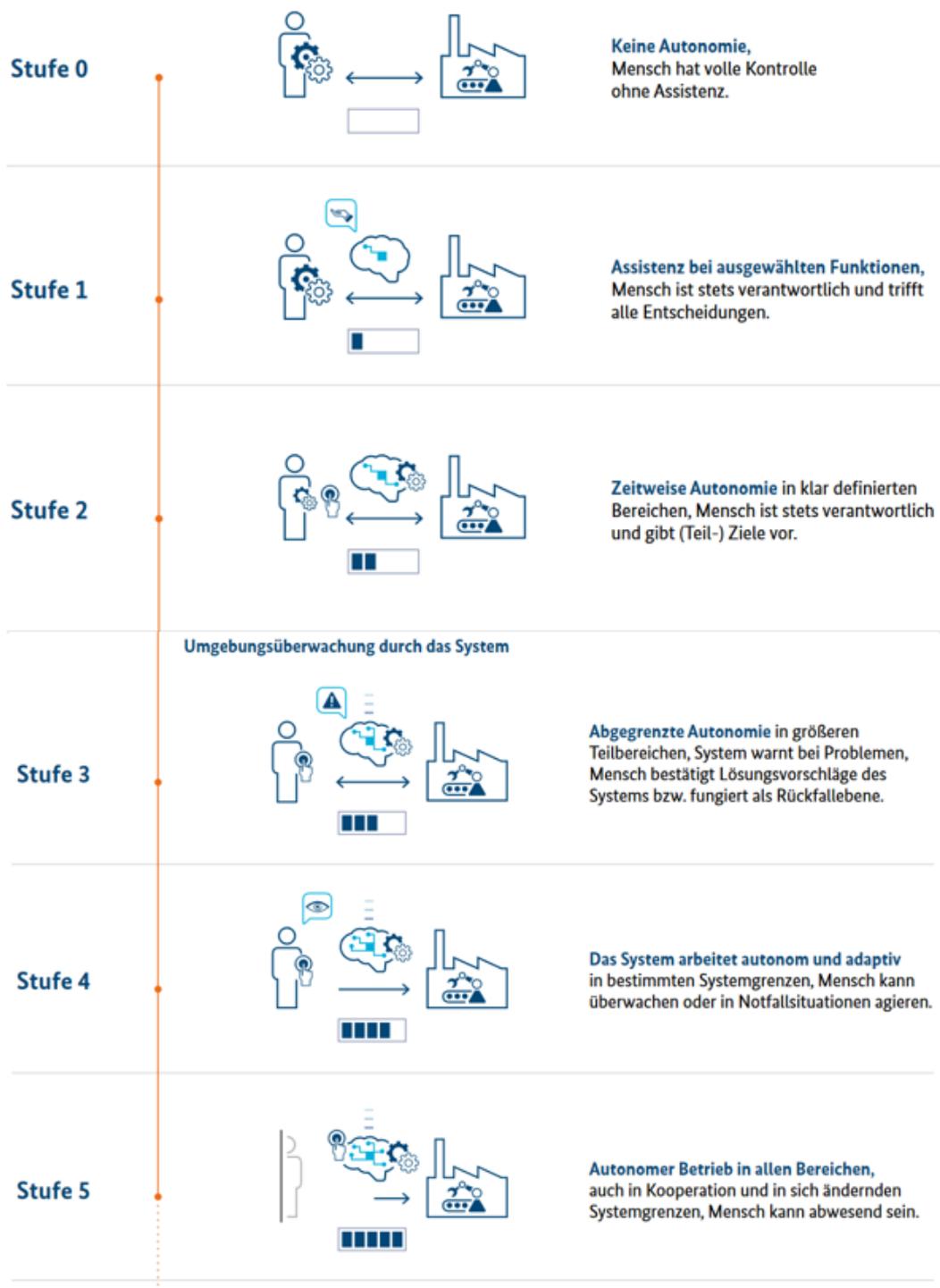


Abbildung 12: Autonomie-Stufen in der industriellen Produktion nach [BMW19].

Die erarbeitete Metrik dient im weiteren Verlauf dazu das implementierte AS hinsichtlich seiner Automatisierungslevels zu bewerten. Im Rahmen von Design Science Research ist nach [WIE14] die Validierung des Artefaktes vorgesehen. Es wird die Interaktion eines Prototyps des Artefakts mit einem Modell des Kontextes untersucht, um zu zeigen, dass das Artefakt zur Erfüllung der definierten Ziele beiträgt. Mit dieser Metrik kann bewertet werden, in welcher Lebenszyklusphase das AS welchen Automatisierungslevel erreicht. Ein möglichst hoher Level ist erstrebenswert, muss aber immer in Relation zum Aufwand hinsichtlich Implementierung und der Generalisierbarkeit (Übertragbarkeit des AS auf andere Produkti-

onsanlagen) stehen. Es ist anzunehmen, dass sich für die unterschiedlichen Lebenszyklusphasen verschiedene Level ergeben, da die integralen Tätigkeiten in ihrer Komplexität variieren. An dieser Stelle wird gefordert, dass das AS über alle Lebenszyklusphasen hinweg mindestens das Automatisierungslevel 2 erreicht. Ab diesem Level übernimmt das AS, entsprechend der Ausführung in Tabelle 5, alle operativen Tätigkeiten und ist somit in der Lage, die abgeleiteten Anforderungen selbstständig zu erfüllen.

Nach der Untersuchung von Simulation im modularen Anlagenbau, den bestehenden Herausforderungen aus Sicht des M&S-Entwicklungspersonals und der Ableitung der Anforderungen an das AS, sollen im nächsten Abschnitt die Systeme und Szenarien, die zur Validierung des AS dienen, vorgestellt werden.

2.3 Betrachtete Systeme und Validierungsszenarien

Nachdem das Ziel der Arbeit und die Anforderungen an den Einsatz von M&S im modularen Maschinen- und Anlagenbau definiert wurden, werden im Kapitel 3 bestehende (Teil-)Lösungen, deren Defizite und das erarbeitete Lösungskonzept vorgestellt. Vorab werden im folgenden Abschnitt zwei Systeme vorgestellt, die als Validierungsszenarien im Verlauf der weiteren Arbeit herangezogen werden. Anhand dieser werden die einzelnen Teile des Lösungskonzepts validiert.

Das erste System ist nach [DIN 8580] der Hauptgruppe 3 der Fertigungsverfahren *Trennen*, konkret der Untergruppe 3.3.7 *Gleitspanen* zuzuordnen. Hierbei handelt es sich um eine Gleitschleifanlage der Firma OTEC Präzisionsfinish GmbH (kurz OTEC). Diese und die damit verbundenen Validierungsszenarien werden in den Abschnitten 2.3.1 und 2.3.2 vorgestellt. Das zweite System ist nach [DIN 8580] der Hauptgruppe 5 der Fertigungsverfahren *Beschichten*, konkret der Untergruppe 5.9.1 *Galvanisches Beschichten* zuzuordnen. Betrachtet wird eine Kleingalvanik der Firma Jentner Plating Technology GmbH (kurz Jentner). Diese und die damit verbundenen Validierungsszenarien werden in den Abschnitten 2.3.3 und 2.3.4 vorgestellt. Beim ersten System handelt es sich um eine typische Sondermaschine aus dem allgemeinen Maschinen- und Anlagenbau, beim zweiten System um eine Sondermaschine aus dem Bereich Prozesstechnik. Damit werden zwei unterschiedliche Bereiche abgedeckt. Auf Basis der zu erwartenden Unterschiede kann das erarbeitete Lösungskonzept hinreichend breit validiert werden.

2.3.1 Aufbau und Wirkweise der OTEC Gleitschleifanlage

Zunächst sollen Aufbau und Wirkweise einer Gleitschleifanlage beschrieben werden. Das Fertigungsverfahren Gleitspanen wird nach [DIN 8589-17] „zum Entgraten und zur Verbesserung der Werkstückoberfläche eingesetzt“, wobei „zwischen Werkstücken und einer Vielzahl von losen Schleifkörpern bzw. einem Schleifmittel unregelmäßige Relativbewegungen stattfinden, die die Spanabnahme bewirken“. In [DIN 8589-17] wird Gleitspanen, abhängig vom verwendeten Schleifmittel, weiter in Gleitlappen und Gleitschleifen differenziert. Gleitschleifen ist eine Form des Gleitspanens, welche „Schleifkörper [verwendet], deren Form, Größe und Zusammensetzung auf die zu bearbeitenden Werkstücke abgestimmt sind“ [DIN 8589-17]. Das Gleitschleifen lässt sich nach [DIN 8589-17], je nach Bauform und Wirkprinzip der Anlage, weiter in Trommel-, Vibrations-, Fliehkraft- und Tauch-Gleitschleifen unterteilen. Ausschlaggebend ist hierbei, wie die Relativbewegung zwischen Werkstück (WS) und Schleifmittel erzeugt wird. Die betrachtete Gleitschleifanlage der Firma OTEC ist dem Tauch-Gleitschleifen zuzuordnen. [DIN 8589-17] definiert diese Form als „Gleitschleifen, bei dem die an einer Halterung befestigten Werkstücke in die sich bewe-

gende Schleifkörperfüllung eines Behälters eingetaucht werden. Die Relativbewegung zwischen Werkstücken und Schleifkörpern kann auch durch eine Bewegung der Werkstücke hervorgerufen oder bei bewegten Schleifkörpern zusätzlich verbessert werden“.

Der grundlegende Aufbau einer solchen Anlage ist der linken Seite von Abbildung 13 zu entnehmen. Rechts in der Abbildung ist ein Ausschnitt des CAD-Modells der OTEC Anlage dargestellt. Die Komponenten Arbeitsbehälter sowie Schleifkörper und Zusatzmittel unterscheiden sich dabei in den beiden Darstellungen nicht. Beim Werkstückträger gibt es hingegen einen Unterschied zwischen der Skizze links und der OTEC Anlage. Die reale Anlage verfügt über drei Arbeitsstationen (orange gerahmt). Hierbei handelt es sich um Antriebe mit Spannmechanismen (blau gerahmt), die einzelne WS (lila gerahmt) halten und in den Arbeitsbehälter absenken können. Der Spannmechanismus besitzt zusätzlich einen Freiheitsgrad in Form eines Drehgelenks, um das WS in die Strömung der Schleifkörper zu stellen und so den Angriffspunkt der wirkenden Kräfte zu verändern.

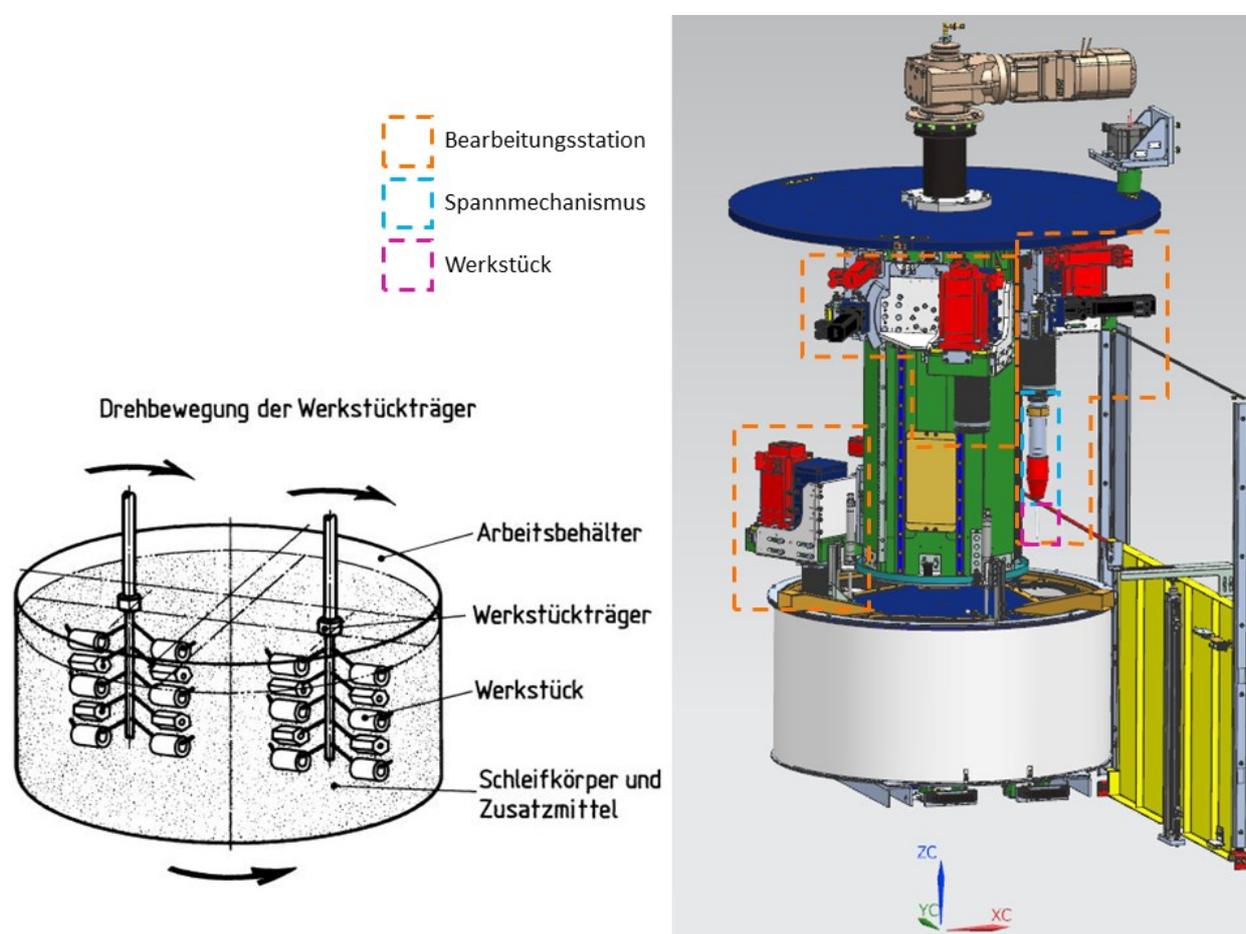


Abbildung 13: Aufbau und Wirkweise des Tauch-Gleichschleifens [DIN 8589-17] (links) und CAD-Auszug der OTEC Anlage (rechts).

Nach der Beschreibung von Aufbau und Wirkweise der Anlage gilt es im nächsten Schritt, das Szenario zu definieren, bei dem Simulation gewinnbringend über den gesamten Lebenszyklus der Anlage eingesetzt werden soll.

2.3.2 Szenario 1: Wirkende Kräfte auf Werkstücke bei Gleitschleifprozess

Im Rahmen der (virtuellen) Inbetriebnahme ist es, entsprechend den Anforderungen aus Abschnitt 2.2.4, Ziel, auf Basis eines Informationsmodells und den zur Verfügung stehenden Physikmodellen ein Modell der gesamten Anlage automatisch zu erzeugen. Relevant ist bei dem betrachteten Fall der Gleitschleifanlage die Interaktion zwischen den WS und den Schleifkörpern. Die Relativbewegung erzeugt eine Kraft auf das WS, die letztendlich den Abtrag beeinflusst. Diese Kraft kann nur bedingt und nur mit zusätzlichem Aufwand gemessen werden. Im Falle der betriebsparallelen Simulation ist die simulative Berechnung dieser Größe von Relevanz. Diese ist wiederum von weiteren Größen wie beispielsweise der Schleifmitteldichte oder der Strömungsgeschwindigkeit des Schleifmittels abhängig. Diese können nur geschätzt werden und müssen durch Modelladaption präzisiert werden. Darüber hinaus sind die physikalischen Zusammenhänge der Anlage bzw. des Prozesses komplex, was sich auf die Physikmodelle auswirkt. Die Bestimmung von passenden Detaillierungsgraden der Modelle ist daher von Relevanz, um den Simulationsaufwand zu reduzieren bzw. den Genauigkeitsanforderungen anzupassen. Anhand einer Expertenbefragung wurden für OTEC typische Umbauszenarien identifiziert. Im firmeninternen Versuchslabor werden u. a. Bearbeitungsprozesse für Kunden-WS entwickelt und Machbarkeitsstudien durchgeführt. Um neue Prozesse zu testen, ist häufig ein Anlagenumbau notwendig. So werden beispielsweise, abhängig von der WS-Geometrie, unterschiedliche Spannmechanismen verwendet. Werden nun größere WS mit den gleichen Prozessparametern (z. B. Eintauchtiefe, Drehgeschwindigkeit des Behälters) bearbeitet, treten größere Kräfte auf. Die WS-Antriebe können nur gewisse Arbeitsbereiche abdecken. Werden die Prozesskräfte zu groß, muss entweder der Prozess langsamer durchgeführt werden, oder die Antriebe müssen größer dimensioniert werden. Daher sind die Antriebseinheiten sowie WS-Halterungen modular ausgelegt. Zwei unterschiedliche Antriebseinheiten sind in Abbildung 14 dargestellt. Neben Antrieben mit Pulsfunktion (schneller Wechsel der Drehrichtung, links) können auch Schrittmotoren (rechts) verwendet werden. Diese ermöglichen es, das WS in einem bestimmten Winkel zum Schleifmittel zu halten und diesen über den Prozess zu ändern. So kann die Schleifmittelströmung das WS gezielt von verschiedenen Seiten angreifen. Ein solcher Modulwechsel soll vom AS detektiert, modelliert und dokumentiert werden.

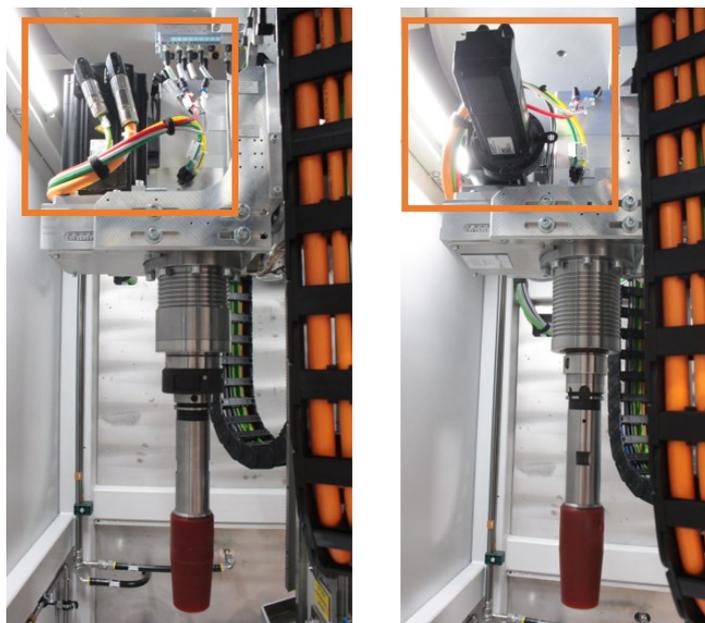


Abbildung 14: Antriebswechsel als Umbauszenario – Umbau von Pulsantrieb (links) zu Schrittmotor (rechts).

2.3.3 Aufbau und Wirkweise der Jentner Kleingalvanikanlage

Auch für die zweite Anlage, die Kleingalvanik der Firma Jentner, sollen zunächst Wirkweise und Aufbau beschrieben werden. Wie bereits zu Beginn des Abschnitts erläutert, ist das Fertigungsverfahren der Untergruppe 5.9.1 galvanisches Beschichten zuzuordnen. Beschichten wird in [DIN 8580] allgemein als „*Fertigen durch Aufbringen einer fest haftenden Schicht aus formlosem Stoff auf ein Werkstück*“ definiert. Weiter „*ist der unmittelbar vor dem Beschichten herrschende Zustand des Beschichtungsstoffes*“ maßgebend [DIN 8580]. Beim verwendeten Verfahren liegt der Beschichtungsstoff im ionisierten Zustand vor. Metallische Werkstoffe aus wässrigen Lösungen, lösemittelhaltigen Bädern oder Salzschnmelzen werden als Elektrolyte bezeichnet und zur Beschichtung eingesetzt [FF18]. Beschichtungen werden nach Warn-ecke [WAR93] eingesetzt, um Anforderungen an die Oberflächen von WS gerecht zu werden, wenn der Grundwerkstoff diese nicht erfüllen kann. Hierzu gehört u. a. der Schutz gegen äußere Einflüsse (z. B. Korrosionsschutz) oder die Erzielung spezieller physikalischer Effekte (wie Elektroisolation, Erhöhung der Leitfähigkeit) [WAR93].

Der grundlegende Aufbau eines galvanischen Bades nach [WAR93] ist in Abbildung 15 dargestellt. Beim Galvanisieren werden WS in ein elektrolytisches Bad eingetaucht. Die WS werden als Kathode geschaltet. Die WS werden dazu an Gestellen befestigt oder als Schüttgut in Trommeln gegeben. Das Beschichtungsmaterial, welches sich ebenfalls im Bad befindet, wird entsprechend als Anode geschaltet. Beide Pole sind an einen äußeren Stromkreis angeschlossen. Durch das Anlegen einer elektrischen Spannung bildet sich im Elektrolyten ein elektrisches Feld. Unter dem Einfluss des elektrischen Feldes trennen sich die Salz-moleküle in Ionen auf. Die positiv geladenen Metallionen wandern zum WS, der negativen Kathode, und lagern sich dort an. Durch Aufnahme von Elektronen, die aus der Stromquelle stammen, werden die Metallionen zu neutralen Metallatomen reduziert. Durch diesen Prozess baut sich auf dem WS eine Schicht aus Metallatomen auf. Um diesen Prozess stabil zu halten, müssen Metallionen ergänzt werden. Dies erfolgt entweder durch die Auflösung der Anoden, welche entsprechend in regelmäßigen Intervallen ausgetauscht werden müssen, oder durch Zugabe von Metallsalz in den Elektrolyten. Die negativ

geladenen Säurerest-Ionen wandern zur positiven Anode und werden an dieser oxidiert. Die abgegebenen Elektronen wandern über den äußeren Stromkreis ab und fließen zur Kathode. An den Anoden kommt es dabei zur Bildung von Metallsalzen. [WAR93, FF18]

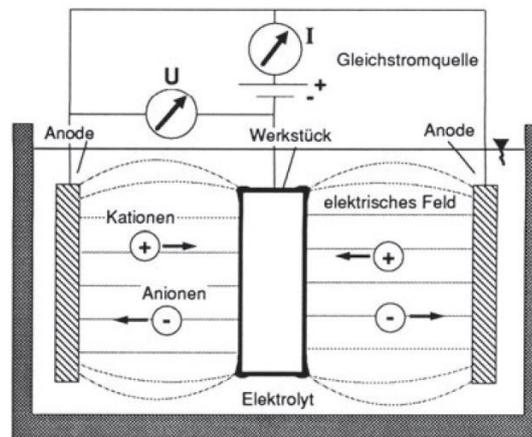


Abbildung 15: Elemente und Aufbau eines galvanischen Bades nach [WAR93].

Nach der Beschreibung der allgemeinen Wirkweise und des Aufbaus eines galvanischen Bades soll die Referenzanlage im Detail vorgestellt werden.

Die Kleingalvanik der Firma Jentner eignet sich für gängige galvanische Prozesse wie z. B. zum elektrolytischen Entfetten, Rhodinieren, Vergolden und Versilbern [JEN22[®]]. Hierfür verfügt die Anlage über insgesamt vier Spül- sowie vier Arbeitswannen. Die Anlage ist in Abbildung 16 von der Seite (a) und von oben (b) dargestellt. Links in (b) befinden sich Bedienfeld und Display, z. B. für die Einstellung von Prozessparametern, gekennzeichnet durch die Marker 1 bis 15. Im rechten Teil von (b) findet der Prozess statt. Die Wannen sind durch den Marker 16 gekennzeichnet, die zwei Anodenstangen durch 17 und die Kathodenstange durch 18. Die Kathodenstange dient als Warenstange auf der, mittels Warengestellen, die WS angebracht werden. Die Anlage ist dahingehend modular aufgebaut, dass die Elektrolyte in den vier Bearbeitungswannen unkompliziert ausgetauscht und so andere Beschichtungsprozesse durchgeführt werden können. Dies ermöglicht es ohne großen Aufwand andere WS-Typen zu bearbeiten.



Abbildung 16: Kleingalvanik der Firma Jentner. Oben in seitlicher Ansicht (a), unten in Ansicht von oben (b).
[JEN22@]

Nach dem Aufzeigen von Wirkweise und Aufbau der Anlage sollen im nächsten Schritt der Lebenszyklus der Anlage, ein möglicher Galvanisierungsprozess und die Einsatzmöglichkeiten von Simulation beschrieben werden.

2.3.4 Szenario 2: Mehrstufige Oberflächenbeschichtung in der Galvanotechnik

Beim betrachteten Referenz-Galvanisierungsprozess werden OP-Pinzetten beschichtet, u. a. um keimresistente Oberflächen zu schaffen und sowohl die Reinigung als auch die Sterilisation zu vereinfachen. Hierfür sind, neben den Vor- und Zwischenbehandlungsschritten wie Entfetten, Beizen und Spülen, mehrere Prozessschritte wesentlich. Die jeweiligen Bearbeitungsstadien der WS sind der Abbildung 17 zu entnehmen. Die fünf Stadien sind durch orangene Marker gekennzeichnet. Hierbei ist zu beachten, dass unterschiedliche Pinzetten-Rohlinge abgebildet sind, die sich in ihren geometrischen Ausprägungen unterscheiden. Relevant ist in diesem Zusammenhang lediglich der jeweilige Beschichtungszustand. Beginnend von links nach rechts ist in Abbildung 17 zunächst die Rohware aus blankem Aluminium abgebildet (Marker 1). Rechts daneben ist die Rohware nach der Vorbehandlung mit Zinkat dargestellt (Marker 2). Diese Vorbehandlung dient der besseren Haftung der im nächsten Prozessschritt aufgetragenen Schicht. Hierbei handelt es sich um das chemische Vernickeln (Marker 3). Im nächsten Schritt wird die Pinzette vergoldet (Marker 4). Die dünne Goldschicht verbessert die Haftung zwischen der Nickel- und der im nächsten Schritt aufgetragenen Silberschicht. Im letzten Bearbeitungsschritt wird die Pinzette zunächst versilbert und anschließend rhodiniert (Marker 5). Eine dicke Silberschicht wird aufgetragen, um die strom- und wärmeleitenden Eigenschaften zu verbessern. Eine dünne Rhodium-Schicht verhindert letztlich das Oxidieren der Silberschicht.



Abbildung 17: Exemplarischer Galvanikprozess - Unterschiedliche Beschichtungsstadien von OP-Pinzetten.

Sollen nun stattdessen andere WS in einem angepassten Prozess bearbeitet werden, können Bearbeitungswannen ausgetauscht, die Reihenfolge verändert, entfernt oder hinzugefügt werden. Die Flexibilität, die die Anlage bedingt durch das Wannensystem aufweist, kann durch ein monolithisches Simulationsmodell nicht abgebildet werden. Analog zum OTEC-Szenario ist es im Rahmen der (virtuellen) Inbetriebnahme Ziel, entsprechend den Anforderungen aus Abschnitt 2.2.4, auf Basis eines Informationsmodells und den zur Verfügung stehenden Physikmodellen der Elektrolyte und der WS-Träger ein Modell der gesamten Anlage für die Prozesssimulation automatisch zu erzeugen. Mit Physikmodellen der WS, deren Trägern sowie der einzelnen Becken inklusive Elektrolyte ist es z. B. möglich, abhängig von Parametern wie der anliegenden Stromstärke und der initialen Masse von Substanzen, Rückschlüsse auf den elektrochemischen Zerfall und damit auf die Stoffkonzentrationen in den Elektrolyten zu ziehen. Der Betreiber der Galvanikanlage ist bestrebt, die Stoffkonzentrationen innerhalb spezifischer Grenzwerte zu halten, da sonst negative Einflüsse auf die Qualität der WS (z. B. zu geringe Schichtdicke, Einschlüsse auf Oberfläche) zu erwarten sind. Die Bestimmung der Elektrolytzustände und Stoffkonzentrationen ist ohne Simulation allerdings nur durch aufwändige Labormessungen realisierbar. Diese sind kostenintensiv und werden daher nur in großen Zeitintervallen (z. B. einmal pro Woche) durchgeführt. Entsprechend groß ist die Unsicherheit über Konzentrationen und die Zustände der Elektrolyte zwischen diesen Messpunkten. Der betriebsparallele Einsatz von Simulation ermöglicht dahingehend Aussagen über den Zustand der Elektrolyte und Empfehlungen für das Nachdosierung von Stoffmengen. Durch die komplexe Zusammensetzung der Elektrolyte und mögliche Wechselwirkungen der einzelnen Inhaltsstoffe ist jedoch die Adaption der Mo-

dellparameter durch Messdaten wichtig, um die Modellgenauigkeit zu verbessern. Im Rahmen der Lebenszyklusphase Umbau der Anlage wird der oben beschriebene Fall betrachtet, dass sich der Prozess und damit die zu bearbeitenden WS sowie die Reihenfolge und Zusammensetzung der Elektrolyte ändert.

2.4 Zusammenfassung der Erkenntnisse

In diesem Kapitel wurde zunächst aufgezeigt, dass markt- und produktbedingte Anforderungen den Übergang von monolithischen zu modularen Produktionsanlagen fördern. Anschließend wurde dargelegt, wie das Engineering dieser Anlagen erfolgt und dass M&S in diesem Rahmen einen immer größeren Stellenwert einnimmt. Die für diese Arbeit relevanten Grundlagen für M&S wurden ebenfalls geschaffen. Anhand des aufgezeigten Modellbildungsprozesses und unternehmenspolitischer Herausforderungen (Schutz von Wissen) sowie technologischer Herausforderungen (u. a. Heterogenität der Domänen, Tools, Modellierungssprachen) wurden die aktuellen Herausforderungen für das Entwicklungspersonal im Bereich M&S modularer Anlagen für die einzelnen Lebenszyklusphasen einer modularen Anlage aufgezeigt. Als Konsequenz hieraus ergibt sich, dass die über den gesamten Anlagenlebenszyklus hinweg notwendigen M&S-Tätigkeiten mit hohen manuellen Aufwänden und entsprechenden Kosten verbunden sind.

Auf dieser Erkenntnis basierend wurde ein neuer Modellierungs- und Simulationsworkflow für das Engineering modularer Produktionsanlagen gefordert. Hierfür wurde untersucht, wie die Unterstützung bei manuellen Tätigkeiten und Entscheidungen in anderen Bereichen des Engineerings erfolgt. Diese Analyse ergab, dass die nötige Unterstützung durch Assistenzsysteme, die einem Bediener zur Verfügung gestellt werden, erfolgt. Auf Basis der identifizierten Herausforderungen wurden Anforderungen an das AS abgeleitet. Um die Eignung eines im Rahmen dieser Arbeit implementierten AS einstufen zu können, wurden Automatisierungslevel für diese definiert. Im letzten Abschnitt wurden noch zwei Systeme vorgestellt, anhand derer der implementierte AS-Prototyp validiert werden soll.

Bevor ein AS-Prototyp implementiert werden kann, ist es im nächsten Schritt erforderlich, bestehende Lösungskonzepte für die einzelnen Lebenszyklusphasen einer Produktionsanlage zu betrachten. Diese werden anschließend anhand der definierten Anforderungen geprüft, deren Defizite aufgezeigt und eine Übersicht für ein Gesamtkonzept vorgestellt. Diese Aspekte sind Inhalt des nächsten Kapitels.

3 Stand von Wissenschaft und Technik

In diesem Kapitel wird in Abschnitt 3.1 der Stand von Wissenschaft und Technik zum Thema (teil-)automatische Modellgenerierung, -adaption und -rekonfiguration untersucht. In Abschnitt 3.2 werden die vorgestellten Lösungskonzepte mit den in Abschnitt 2.2.4 formulierten Anforderungen verglichen, Erkenntnisse festgehalten und bestehende Defizite identifiziert. Aus den diskutierten Lösungskonzepten und deren Defiziten wird in Abschnitt 3.3 der Handlungsbedarf abgeleitet. Das Kapitel schließt mit der Vorstellung einer Gesamtkonzeptübersicht in Abschnitt 3.4, welche alle formulierten Anforderungen und somit aus Simulationssicht alle Lebenszyklusphasen einer modularen Produktionsanlage abdeckt.

Im nächsten Abschnitt wird der Stand von Wissenschaft und Technik zu den Themen der (teil-)automatischen Modellgenerierung, -adaption und -rekonfiguration untersucht. Die identifizierten Lösungskonzepte werden, entsprechend der Lebenszyklusphase bzw. -phasen, in der sie anwendbar sind, klassifiziert. Zuerst werden Konzepte beleuchtet, die jeweils Anforderungen aus einer der drei relevanten Phasen abdecken, der letzte Teil dieses Unterkapitels befasst sich mit phasenübergreifenden Ansätzen.

3.1 Bestehende Konzepte zur (teil-)automatischen Modellgenerierung, -adaption und -rekonfiguration

3.1.1 Konzepte zur (teil-)automatischen Modellgenerierung

Entsprechend der Abfolge der Lebenszyklusphasen einer Produktionsanlage werden zunächst Lösungskonzepte zur (teil-)automatischen Modellgenerierung für die Inbetriebnahmephase betrachtet. Wie bereits in Abschnitt 2.1.4 aufgezeigt, ist anzunehmen, dass die MSM von Zuliefererbetrieben stammen. Daher starten die M&S-Tätigkeiten aus Sicht des Betreibers in der Inbetriebnahmephase. Unter dem Begriff der (teil-)automatischen Modellgenerierung werden *„Ansätze, bei denen das Simulationsmodell nicht (rein) manuell unter Nutzung der Modellierungswerkzeuge bzw. der Programmierumgebung eines Simulationswerkzeugs erstellt wird, sondern vielmehr auf Basis externer Datenquellen, mittels Schnittstellen und Algorithmen, generiert wird“* verstanden [BS20a]. Derartige Ansätze werden bereits seit mehr als 20 Jahren verfolgt [BER13].

In [BAR11, OWD+14] werden Ansätze zur automatischen Modellgenerierung verfahrenstechnischer Anlagen für den Steuerungstest beschrieben. Der Einsatzfall der generierten Simulationsmodelle beschränkt sich hier allerdings auf die Inbetriebnahmephase von verfahrenstechnischen Anlagen. Beide Quellen nennen jeweils zahlreiche weitere Arbeiten zur Simulationsmodellgenerierung im Bereich verfahrenstechnischer Anlagen. Allen gemein ist die Fokussierung auf die einmalige Generierung von Modellen für den sogenannten Factory Acceptance Test. Pendanten hierzu existieren auch für den Maschinen- und Anlagenbau (stellvertretend in [WÜN07]). In [JSW18] wird mit dem Einsatz eines Agentensystems zur Realisierung von Co-Simulationen die Heterogenität der Quellen der Simulationsmodelle berücksichtigt. Bei diesem Ansatz wird jede Komponente in einem durch einen Agenten repräsentierten Werkzeug simuliert. Die Agenten können dynamisch zur Laufzeit in eine laufende Co-Simulation eintreten, was ein „Plug-and-Simulate“-Verhalten ermöglicht [JSW18]. Ein Lösungskonzept, welches ausgehend von Kommissionierungslisten automatisch virtuelle Maschinen aus einem CPS-Baukasten („cyber-physisches System“) erstellt, wird in [SCH20] vorgestellt. In [TFD18] stellen Thongnuch et al. eine Methode vor, mit der teilautomatisch

detaillierte Simulationsmodelle von Fertigungszellen erstellt werden können. Genutzt wird hierfür das, üblicherweise im CAE-Werkzeug vorliegende, 3D-Geometrie-Modell der Fertigungszelle. Zur Erzeugung des Simulationsmodells wird die Geometrie mit einer Verhaltensbeschreibung entsprechend VDI-Richtlinie 2860 kombiniert und so dynamisiert. Die Methode nutzt weitverbreitete Engineering-Werkzeuge und AutomationML [AUT22[®]] als Beschreibungsmittel [TFD18]. In [UWQ17] wird ebenfalls AutomationML genutzt, um ein Simulationsmodell nach dem „Plug-and-Simulate“-Konzept für eine modulare, herstellerübergreifende Montagelinie aufzubauen. In [BS20a] stellen Bergmann und Straßburger einen Ansatz für die automatische Modellgenerierung im Bereich der Produktionssimulation auf Basis des Datenformat-Standards Core Manufacturing Simulation Data (CMSD) [RL10] vor. Dieses Datenformat nutzt auch Block [BL019]. Er stellt in seiner Arbeit ein grundlegendes Vorgehensmodell für die Einführung einer durchgängigen Digitalen Fabrik auf Basis standardisierter Austauschformate und Schnittstellen vor. Es wird eine einheitliche Semantik und Syntax auf Basis von CMSD inklusive des Austausches mittels HTTP, MQTT sowie OPC-UA entwickelt. Die in cyber-physischen (Produktions-)Systemen generierten Daten können einheitlich ausgetauscht sowie aggregiert werden. Anschließend können die Daten in ein ausgewähltes Simulationswerkzeug überführt und dort automatisch zu einem ausführbaren Modell transformiert sowie simuliert werden. [BL019]

Neben diesen konkreten Beispielen für die automatische Modellgenerierung ist noch die Arbeit von Reinhardt et al. [RWP19] hervorzuheben. In ihrer Studie untersucht das Autorenteam bestehende Ansätze zur automatischen Modellgenerierung für Materialflusssimulationen im Bereich der diskreten Fertigung der letzten 20 Jahre. Die zentrale Fragestellung ihrer Arbeit ist, welche Informationen für die automatische Modellgenerierung genutzt werden können. Im Rahmen ihrer Literaturrecherche können Reinhardt et al. Unterschiede bezüglich der Datenquellen, der zeitlichen Variabilität der Daten sowie angewandte Methoden für die Informationsgewinnung identifizieren. Alle vom Autorenteam untersuchten Ansätze zur automatischen Modellgenerierung nutzen eine oder eine Kombination aus mehreren der in Tabelle 6 dargestellten Datenquellen. Die jeweiligen Datenquellen werden in der Studie mit Beispielen und Literaturreferenzen untermauert.

Tabelle 6: Von [RWP19] identifizierte Datenquellen inklusive Beispielen für die automatische Modellgenerierung.

Datenquelle	Beispiele für Datenquellen
CAD-Daten	Maschinenkonstruktions- oder Fertigungssystem-Layoutdaten
Unternehmensdaten	Strukturierte Daten aus kommerzieller Fertigungssoftware wie Produktionsplanungssysteme (PPS), Manufacturing Execution System (MES), Systeme für Enterprise Resource Planning (ERP) oder zur Produktionsdatenerfassung (PDA)
Wissensbasis	Informationen für die Simulation können unter Anwendung eines bestimmten Schemas oder Formats explizit in einem speziellen Dokument (z. B. CMSD, AML) oder einer Datenbank ausgedrückt werden
SPS oder CNC Code	Kein explizites Beispiel genannt
Sensordaten	Diskrete oder quasi-kontinuierliche Trackingdaten von RFID-getaggten Werkstücken
Stochastische Werte	Zufällige Parametereinstellungen zur Erzeugung vieler verschiedener Simulationsmodelle zum Test und zur Validierung von Methoden zur Simulationsvisualisierung

Bedienereingabe	Bedienereingabe, bei der der Bediener die Möglichkeit hat, die Eingabedaten oder das Modellierungsprogramm anzupassen oder mit ihnen zu interagieren
-----------------	--

Bezüglich der zeitlichen Variabilität der Daten unterscheiden Reinhardt et al. zwischen statischen (offline) und dynamischen (online) Daten. Statische Daten stammen typischerweise aus der Planungsphase eines Fertigungssystems und werden selten geändert. Sie zeichnen sich durch ihre Unabhängigkeit vom Fertigungsbetrieb aus. Als Beispiele nennt das Autorenteam geschätzte oder vorausberechnete Maschinenausfallwahrscheinlichkeiten, Layoutdaten oder Steuerungsquellcode. Dynamische Daten können nach Reinhardt et al. als eine Reihe von Schnappschüssen eines Teils eines gegebenen Fertigungssystems beschrieben werden, z. B. Maschinenzustände oder Daten zur Positionierung von Werkstücken. Sie sind daher durch ihre Abhängigkeit vom Fertigungsbetrieb gekennzeichnet. Dynamische Daten können für die adäquate Modellierung der Fertigungsdynamik oder die Validierung von Simulationen erforderlich sein.

In einem letzten Schritt untersuchen Reinhardt et al. das Thema Standardisierung im Bereich der automatischen Modellgenerierung. Sie kommen zu dem Schluss, dass eine beträchtliche Anzahl der von ihnen untersuchten Ansätze auf einer zwischengeschalteten Datenbank oder einem Datenaustauschformat beruht, was für die Trennung der Datenerfassung und -transformation von den Modellgenerierungsroutinen von Vorteil ist. Diese Ansätze profitieren oft von Standardisierungsbemühungen. De-facto-Standards wie AutomationML, CMSD, ISA-95 und SysML [Sys22[®]] werden nach [RWP19] häufig auf Fragestellungen der diskreten Event-Simulation angewandt. Alle Ansätze berücksichtigen dabei nur jeweils eine Standardisierungsbemühung. [RWP19]

Erkenntnisse und Defizite vorgestellter Konzepte für die Modellgenerierung

Wie aufgezeigt gibt es bereits eine Vielzahl an Lösungsansätzen, die sich mit dem Thema der automatischen Modellgenerierung befassen. Anforderung A1 (*Selbstständig MSM laden, initialisieren, koppeln und simulieren*) kann dabei von allen vorgestellten Ansätzen mit unterschiedlichen Ausprägungen erfüllt werden. Anforderung A2 (*Möglichkeit der Erweiterung des Einsatzbereichs auf kompletten Anlagenlebenszyklus*) können nicht alle Konzepte gänzlich erfüllen. Als Kriterien sind hier anzuführen, ob die Konzepte beispielsweise mit MSM aus heterogenen Quellen und Domänen arbeiten können und standardisierte Beschreibungsmittel verwenden. Diese Fähigkeiten stellen die Grundlage für einen sinnvollen Einsatz der Konzepte in den weiteren Lebenszyklusphasen dar.

Reinhardt et al. [RWP19] schließen ihre Studie mit der Erkenntnis ab, dass die von ihnen untersuchten Ansätze zur automatischen Modellgenerierung oft auf einen bestimmten Anwendungsfall zugeschnitten sind. Standardisierung wird nur für die Erstellung temporärer Datenrepräsentation und -speicherung („intermediate data representation and storage“) in Betracht gezogen. Das Autorenteam folgert, dass der Zugriff auf und das Auffinden von (allen) benötigten Informationen in typischerweise heterogenen, verteilten Datenlandschaften von Fertigungsunternehmen noch ein wesentliches Problem darstellt. Dieses muss gelöst werden, um die industrielle Nutzung der automatischen Modellgenerierung voranzutreiben. Zukünftige Forschung sollte sich daher stark auf die Universalität konzentrieren. [RWP19]

Folglich muss ein lebenszyklusübergreifendes Lösungskonzept für die (Teil-)Automatisierung der Modellierungstätigkeiten für modulare Produktionsanlagen auf standardisierten Beschreibungsmitteln und einer Simulationsarchitektur, die MSM mit standardisierten Schnittstellen fordert bzw. unterstützt, basieren. Unter Erfüllung dieser Vorgabe können Aspekte aus bestehenden Lösungen übernommen werden.

Nach der Vorstellung verschiedener Ansätze, die sich auf die erste Lebenszyklusphase konzentrieren, sowie eingesetzter Werkzeuge, Datenformate und -quellen und wie aus diesen Informationen abgeleitet werden können, liegt der Fokus im nächsten Abschnitt auf Konzepten für die Betriebsphase.

3.1.2 Konzepte zur (teil-)automatischen Modelladaption

Die Lösungskonzepte in diesem Abschnitt befassen sich mit der (teil-)automatischen Modelladaption, die parallel zum Betrieb der Anlage durchgeführt wird.

Elbattah und Molloy [EM18] beschreiben in ihrer Arbeit, dass die jüngsten Trends hin zu datengesteuerten Methoden ein grundlegendes Umdenken in der Praxis der M&S erfordern werden. Sie identifizieren Maschinelles Lernen (ML) als entscheidendes Artefakt für die Entwicklung neuer Erkenntnisse oder die Verbesserung bereits vorhandenen Wissens. In ihrem Beitrag stellen Elbattah und Molloy einen konzeptionellen Rahmen für die Integration von Simulationsmodellen mit ML vor. Diese Kombination bezeichnen sie als *hybride* Simulation. Im Kern basiert ihr Ansatz auf der Prämisse, dass Systemwissen (teilweise) erfasst und mit Hilfe von ML automatisch aus Daten gelernt werden kann. Ihr Ziel ist es, adaptive Simulationsmodelle zu realisieren, die lernen, ihr Verhalten als Reaktion auf Verhaltensänderungen im eigentlichen System zu ändern. Wird dieses dynamische Verhalten des Systems nicht im Simulationsmodell abgebildet, kommt es im Laufe der Zeit zu einer „Konzeptabweichung“ („concept drift“). In diesem Zusammenhang stellen sich Elbattah und Molloy die Frage, was wäre, wenn Simulationsmodelle neue Informationen oder neues Wissen auf automatisierte Weise wahrnehmen könnten, beziehungsweise was wäre, wenn sich ein Simulationsmodell an neue Situationen anpassen könnte, ohne dass es explizit von einem Modellierer oder Simulationsexperten informiert wird. Um das Ziel ihrer Studie zu verfolgen, formulieren die Autoren drei weitere Fragen, die an dieser Stelle sinngemäß wiedergegeben werden:

1. Wie können Simulationsmodelle mit minimalen menschlichen Eingaben über Veränderungen im Verhalten oder in den Bedingungen des tatsächlichen Systems lernen?
2. Ist es möglich, Simulationsmodelle mit ML-Modellen so zu integrieren, dass dieser Lernprozess automatisiert abläuft? Wenn ja, wie?
3. Kann, gegeben durch eine messbarere Genauigkeit der ML-Modelle, die Integration mit ML zu einem höheren Maß an Vertrauen in Simulationen führen?

Die Quellen für ihre Studie schränken die Autoren in [EM18] auf Publikationen der Konferenzen *Winter Simulation Conference* und *SIGSIM PADS* über einen Zeitraum von zehn Jahren ein. Die untersuchten Quellen stammen aus unterschiedlichen Bereichen, z. B. dem Gesundheitswesen, Bau und Umwelt sowie der autonomen Flottenkoordination. Elbattah und Molloy kommen zu dem Schluss und kritisieren, dass die meisten Versuche, Simulationsmodelle mit ML zu kombinieren, weitgehend als „ad-hoc-Versuche“ ohne klare Methodik beschrieben werden können. Weiter nach Elbattah und Molloy gibt es nur wenige Hinweise darauf, warum, wann und wie ML in die verschiedenen Phasen von M&S integriert werden sollte. Die Autoren sehen einen Bedarf an weiteren Studien für die M&S-Gemeinschaft, die verallgemeinerbare Methoden formulieren sowie eine Sammlung von Beispielen, die dazu beitragen, diese Integration von ML in der Bereichen M&S zu fördern und zu verbreiten. Sie stellen daher im Rahmen ihrer Arbeit einen eigenen Ansatz vor, der entsprechend den drei formulierten Fragen aus drei Kerngedanken besteht. Das Lösungskonzept für die automatische Modelladaption nach [EM18] ist in Abbildung 18 dargestellt. Im Folgenden werden die Teile des Lösungskonzepts vorgestellt.

modelliert werden können, um Positionierungsfehler zu reduzieren und entsprechend Zykluszeiten für die Nachkalibrierung zu erhöhen. Für die Modellierung dieser veränderlichen Einflüsse kombinieren Dahlem et al. physikbasiertes White-Box-Wissen und maschinelles Lernen. Typische White-Box-Modellierungsansätze für den vorgestellten Anwendungsfall basieren auf der Verwendung der Finite-Elemente-Methode (FEM). Diese erfordert genaue Kenntnis von Geometrie und Materialzusammensetzung der relevanten Maschinenkomponenten und der Umgebungsbedingungen. Auf dieser Grundlage kann das mechanische und thermoelastische Verhalten der Maschine berechnet werden. Den hohen Rechenaufwand, der sich aus der Aufteilung in finite Elemente ergibt, und den hohen Initialaufwand für die Geometrie- und Materialmodellierung schätzen Dahlem et al. als problematisch ein. Dass der Aufwand für die Modellerstellung nicht auf verschiedene Maschinentypen skalierbar ist und individuell vorgenommen werden muss, sehen die Autoren als weiteres Problem der White-Box Modellierung. Die alleinige Anwendung von Machine Learning zur Problemlösung schätzen Dahlem et al. ebenfalls als noch nicht zufriedenstellend ein. Als Argumente führen die Autoren an, dass Machine-Learning-Lösungen in der Regel einen Black-Box-Charakter haben und die trainierten Modelle entsprechend für den Menschen oft nicht nachvollziehbar sind. Werden solche Modelle auf unbekannte Datensätze angewandt, können sie unerwünschte und falsche Ergebnisse liefern. Die Modelle können nicht von Experten angepasst werden, wie es bei White-Box-Modellen typischerweise der Fall wäre. [DSBF+20]

Dahlem et al. schlagen daher ein neuartiges Konzept vor, bei dem verschiedene Modellierungsansätze modular zusammengeführt und ihre Ergebnisse überlagert werden. ML-Algorithmen werden verwendet, um das Residuum des Physikmodells im Vergleich zu den Messwerten zu modellieren und die Parameter des Physikmodells zu optimieren. Die Autoren beschreiben den Fall, dass beispielsweise ein White-Box-Modell deaktiviert oder ein weiterer Modellteil integriert wird. Der durch ML erstellte Modellteil würde sich nach der Trainingsphase flexibel vergrößern oder verkleinern. Dahlem et al. sehen das Ziel darin, den Anteil des maschinellen Lernens so klein wie möglich zu halten, um die Vorteile des White-Box-Modellwissens zu erhalten. Zu Validierungszwecken trennen die Autoren den Satz an Messdaten in einen Trainings- und einen Validierungsdatensatz. Für die Validierung werden die fünf ML-Algorithmen lineare Regression, neuronale Netzwerke, k-nächste-Nachbarn-Klassifikation, Random Forest Regression und Support-Vektor-Regression miteinander verglichen. Dahlem et al. kommen zu dem Schluss, dass alle getesteten Kompensationsmethoden, die mit dem Trainingsdatensatz trainiert wurden, den Restfehler im Validierungsdatensatz reduzieren. [DSBF+20]

In [MJW22] stellen Müller et al. einen weiteren Ansatz vor, der Machine Learning für die automatische Modelladaptation nutzt. In dieser Arbeit kommt Reinforcement Learning zum Einsatz. Reinforcement Learning ist ein Trial-and-Error-Lernalgorithmus, bei dem ein autonomer Agent versucht, durch automatisches Lernen die optimale Lösung für ein Problem zu finden [PV20]. In ihrem betrachteten Szenario, in dem autonome mobile Roboter einzelne Arbeitsstationen verbinden, weicht das Verhalten der Anlage von der simulierten Vorhersage ab. Müller et al. bezeichnen diese Abweichung als Simulationslücke („simulation gap“). Der vorgestellte Ansatz versucht diese Simulationslücke durch einen dreistufigen Mechanismus zu schließen. Zunächst müssen die simulierten und die realen Daten vorbereitet und synchronisiert werden, um verglichen werden zu können. Dieser Prozess der Datensynchronisierung besteht wiederum aus zwei Teilen, dem Vergleich und der Erkennung von Anomalien. Beim Vergleich der zwei Datensätze erfolgt zunächst eine Plausibilitätsprüfung, um die Prozessdaten zu validieren. Auf diese Weise können offensichtliche Messfehler aufgedeckt werden. Die Anomalie-Erkennung unterscheidet dann zwischen normalen

Abweichungen zwischen Simulation und Realität und unerklärlichem Verhalten. Durch Clustering der Prozessdaten in sich wiederholende Schritte und anschließende statistische Auswertung der jeweiligen Daten innerhalb dieser Cluster liefert die Anomalie-Erkennung Hinweise für den Vergleichsteil, wo die Grenze zwischen offensichtlichen Messfehlern und unerwarteten, aber plausiblen Daten zu ziehen ist. Im Anschluss setzen Müller et al. Reinforcement Learning ein. Die Modellverbesserung folgt zwei unterschiedlichen Ansätzen, je nachdem, ob die Abweichung zwischen Simulations- und Messwerten als normal oder anormal eingestuft wird. Den ersten Fall betrachten die Autoren in ihrer Arbeit nicht. Im Fall, dass das Modell seine Grenzen erreicht, besteht die Grundidee darin, das Synchronisationssignal als verzögerte Belohnung für den Reinforcement Learning Agenten zu verwenden, um die Aktionen zur Änderung des ursprünglichen Modells zu beurteilen. Das verbesserte Modell wird jedoch nicht direkt eingesetzt, sondern läuft parallel zum ursprünglichen Modell. Auf diese Weise ist die Anomalie-Erkennung weiterhin aktiv. Das neue Modell wird genutzt, um den Lernalgorithmus mit Beispielen zu versorgen. Sobald der Algorithmus für das Reinforcement Learning konvergiert ist, wird automatisch auf das neue Modell umgeschaltet. [MJW22]

Der Beitrag von Zipper et al. [ZAS+18] schlägt ein Konzept für die Erkennung von Änderungen in Fertigungsprozessen vor. Die Erkennung von Prozessänderungen versuchen Zipper et al. durch den kontinuierlichen Vergleich von Mess- und Simulationsdaten sowie den Zugriff auf das Steuerungsnetzwerk zu realisieren. Für den Vergleich der Daten schlagen sie ein Konzept zur Filterung, Synchronisation und Mapping der Daten vor. Die Simulation implementieren Zipper et al. als modulare Co-Simulation, die es ermöglicht, komplexe Anlagen durch Kombinationen verschiedener spezialisierter Teil-Simulationen zu simulieren. Für die Co-Simulation kommt der FMI-Standard [FUN22[®]] zum Einsatz. Um die Daten aller beteiligten Komponenten abzubilden, setzen die Autoren den AutomationML-Standard ein. Ihr Konzept validieren Zipper et al. an einer Anlage, bestehend aus zwei Modulen. Während zwei Prozessen wird ein Parameter an der realen Anlage variiert, was sich in einer Abweichung der Mess- und Simulationswerte widerspiegelt.

Erkenntnisse und Defizite vorgestellter Konzepte für die Modelladaption

Alle in diesem Abschnitt vorgestellten Arbeiten verfolgen einen ähnlichen Ansatz. Es wird versucht, Systemwissen, welches durch physikalische oder mathematische Zusammenhänge dargestellt wird, durch datenbasierte Ansätze anzureichern. Für die Anreicherung schlagen alle Ansätze, in denen eine Modelladaption angestrebt wird, Konzepte und Algorithmen, die dem Bereich des Machine Learnings zuzuordnen sind, vor. Der Einsatz von Machine Learning für die Adaption wird im Rahmen der vorliegenden Arbeit durch keine Anforderung explizit gefordert. Durch den umfangreichen und vielversprechenden Einsatz in allen vorgestellten Arbeiten wird der Einsatz von Machine Learning als Werkzeug zur Adaption von MSM jedoch als sinnvoll und zielführend eingestuft und daher im weiteren Verlauf der Arbeit verfolgt.

Die Studie und das vorgestellte Framework von Elbattah und Molloy [EM18] liefern erste wichtige Erkenntnisse, wie eine Adaption von Simulationsmodellen, basierend auf Machine Learning, grundsätzlich ablaufen kann. Auch der Hinweis, dass das Lernen inkrementell stattfindet und für die Adaption zunächst die Einflussvariablen identifiziert werden müssen, sind wichtige Anhaltspunkte. Zwar referenziert die Studie relevante Arbeiten aus dem Bereich der Modelladaption, das abgeleitete Framework wird jedoch nicht durch einen konkreten Anwendungsfall validiert. Es wird daher nicht beschrieben, welche Werkzeuge,

Beschreibungsmittel oder Simulationsumgebungen für die Umsetzung genutzt werden können. Hinsichtlich der in Abschnitt 2.2.4 definierten Anforderungen ergibt sich folgende Einschätzung. Es findet kein direkter Vergleich der Mess- und Simulationswerte statt. Stattdessen werden die Messwerte genutzt, um Prädiktoren einzulernen, welche wiederum die Einflussvariablen der Verhaltensbeschreibung der Simulationsmodelle einstellen. Die Struktur der Prädiktoren ist nicht beschrieben. Es stellt sich die Frage, was passiert, wenn diese falsch ausgelegt und dadurch die Einflussvariablen fehlerhaft eingestellt werden. Durch diesen nur indirekt durchgeführten Vergleich von Mess- und Simulationswerten wird die Anforderung A3 (*Selbstständiger Vergleich von Simulations- und Messwerten*) als teilweise erfüllt gewertet. Das Framework sieht auch an keiner Stelle vor, Toleranzen für eine mögliche Abweichung zwischen Mess- und Simulationswerten vorzugeben. Stattdessen findet das Lernen kontinuierlich statt und wird auch bei kleinsten Abweichungen der Daten durchgeführt. Diese Abweichungen sind beispielsweise schon allein durch Sensorrauschen gegeben. Es ist fraglich, wie sinnvoll der Versuch ist, diese Abweichungen im Modell abzubilden. Anforderung A4 (*Selbstständige Überprüfung der Abweichung von Simulations- und Messwerten auf Toleranzüberschreitung*) wird daher als nicht erfüllt gewertet. Die selbstständige Lokalisierung möglicher Ursachen für die Abweichung in den Simulationsmodellen wird in Anforderung A5 (*Selbstständige Lokalisierung möglicher Ursachen für Abweichung in Simulationsmodellen*) gefordert. Der Ansatz von Elbattah und Molloy verfolgt dies durch die Reduktion aller Modellvariablen auf die Einflussvariablen. Allerdings geht aus dem Konzept nicht hervor, ob dieser Ansatz auch eingesetzt werden kann, wenn das Systemmodell, entsprechend dem modularen Anlagenbau, aus mehreren Teilmodellen besteht oder als Black-Box vorliegt. Anforderung A5 wird daher als teilweise erfüllt gewertet. Da die Einflussvariablen zyklisch aktualisiert werden, kann Anforderung A6 (*Selbstständige Durchführung einer Adaption der betroffenen MSM*), die automatische Adaption der betroffenen Simulationsmodelle, als vollständig erfüllt gewertet werden.

Der von Dahlem et al. [DSBF+20] vorgestellte Ansatz liefert konkrete Vorschläge zur strukturellen Umsetzung der automatischen Modelladaption und bezüglich der potenziell einsetzbaren Algorithmen. Bezüglich Anforderung A3, dem selbstständigen Vergleich von Mess- und Simulationswerten, zeigt das Konzept jedoch ähnliche Defizite wie der Ansatz von Elbattah und Molloy [EM18]. Die ML-Algorithmen werden nur auf Basis der von den Modellen berechneten Fehlern trainiert. Die Modelle nutzen wiederum die Messwerte als Input. Es kommt daher zu einem indirekten Vergleich der Datensätze. Die Anforderung A3 wird daher als teilweise erfüllt gewertet. Auch Anforderung A4 wird, wie bei dem Ansatz von Elbattah und Molloy, nicht betrachtet. Es findet eine kontinuierliche Adaption statt. Auch bezüglich der Anforderungen A5 und A6 ergibt sich die gleiche Bewertung.

Müller et al. [MJW22] liefern in ihrer Arbeit wertvolle Ansätze für die automatische Modelladaption. Das vorgeschlagene Lösungskonzept ist in der Lage, selbstständig Mess- und Simulationswerte, die zur Betriebsphase aufgenommen werden, miteinander zu vergleichen. Damit erfüllt das Konzept die Anforderung A3 vollständig. Auch die Anforderung A4 zur Toleranzüberwachung kann mit dem Ansatz zur Anomalie-Erkennung als vollständig erfüllt gewertet werden. Anforderung A5 erfordert, dass das Lösungskonzept selbstständig mögliche Ursachen für die Abweichung in den Simulationsmodellen lokalisiert. Da Müller et al. nur das Modell eines Roboters betrachten, kann keine Aussage getroffen werden, ob das Konzept bei mehreren Modellen, wie es im modularen Anlagenbau auftritt, eingesetzt werden kann. Diese Anforderung wird daher als teilweise erfüllt gewertet. Auch die Anforderung A6, die automatische Adaption

der betroffenen Simulationsmodelle, kann als vollständig erfüllt gewertet werden. Diese Aufgabe übernimmt in [MJW22] der trainierte Reinforcement Learning Agent. Müller et al. betonen allerdings in ihrem Ausblick, dass ein entscheidender Aspekt bei ihrem Ansatz darin besteht, die Komplexität bezogen auf den Aktionsraum („action space“) des Agenten gering zu halten. Es handelt sich um einen Kompromiss, um die Konvergenz des Lernalgorithmus sicherzustellen. Es stellt sich die Frage, ob dieses Konzept auch bei mehreren MSM eingesetzt werden kann oder ob eine zu hohe Diskretisierung des Aktionsraumes nötig ist, da sonst keine Konvergenz mehr erreicht werden kann.

Der Ansatz von Zipper et al. [ZAS+18] ist, wie der Ansatz von Müller et al., in der Lage selbstständig Mess- und Simulationswerte, die zur Betriebsphase aufgenommen werden, miteinander zu vergleichen. Damit erfüllt auch dieses Konzept die Anforderung A3 vollständig. Das Konzept arbeitet ohne vorgegebene Toleranzen. Die Anforderung A4 wird daher als nicht erfüllt gewertet. Aufgrund des einfachen Validierungsfalles, bei dem nur ein Parameter variiert wird, ist nicht genau abzuschätzen, ob das Konzept in der Lage ist, Ursachen für Abweichungen in den Simulationsmodellen zu lokalisieren. Im Gegensatz zu den anderen in dieser Kategorie vorgestellten Ansätzen unterstützt dieser Co-Simulation und eignet sich daher auch für modulare Produktionsanlagen. Anforderung A5 wird daher als vollständig erfüllt gewertet. Eine automatische Adaption der betroffenen Simulationsmodelle wird nicht vorgenommen. Anforderung A6 wird daher als nicht erfüllt gewertet.

Nach der Vorstellung von Konzepten zur betriebsparallelen Modelladaption befasst sich der nächste Abschnitt mit Konzepten, die sich auf die Lebenszyklusphase Umbau konzentrieren.

3.1.3 Konzepte zur (teil-)automatischen Modellrekonfiguration

Bevor Ansätze zur Modellrekonfiguration vorgestellt werden, soll zunächst der Begriff Rekonfiguration eingeordnet werden. Im Rahmen dieser Arbeit wird unter Rekonfiguration der *„Prozess [...] der Veränderung eines bereits entwickelten und operativ eingesetzten Systems [verstanden], um es an neue Anforderungen anzupassen, Funktionalität zu erweitern, Fehler zu beseitigen oder die Qualitätseigenschaften zu verbessern.“* [MAT10] Die Anforderungen sind im Fall der automatischen Modellrekonfiguration, dass sich die reale Systemkonfiguration geändert hat und diese Änderungen in das Modell überführt werden müssen.

In ihrer Arbeit [MKW+17] stellen Marseu et al. ein *Plug & Produce* Anwendungsbeispiel aus der diskreten, lose verketteten Fließfertigung vor. Der Begriff *Plug & Produce* bezeichnet nach Dürkop und Jasperneite [DJ17] *„die Vision einer Automatisierungstechnik, in der strukturelle Änderungen möglichst ohne manuelle Konfigurationsaufwände durchgeführt werden können“*. Die Identifikation von neu eintretenden oder gewechselten Anlagenmodulen erfolgt bei Marseu et al. über eine intelligente Nachbarschaftserkennung. Um benachbarte Module zu identifizieren, *„hat jedes Modul an den Stirnseiten RFID-Lesegeräte, -Tags sowie Näherungsschalter“* [MKW+17]. Weitere potenzielle Informationsquellen, um die Anlagentopologie zu erfassen, sind u. a. der Steuerungscode der Anlage [AJS+18], das Kommunikationsnetzwerk [ZAS+18] oder bildbasierte Methoden wie 3D-Laserscans [LBV+16]. Werden diese Methoden wiederholt durchgeführt und die gesammelten Daten miteinander abgeglichen, ist es ebenfalls möglich, Änderungen in der Topologie zu identifizieren. An dieser Stelle kann auch auf die Datenquellen aus Tabelle 6 verwiesen werden, welche bereits als potenzielle Ausgangspunkte für die automatische Modellgenerierung identifiziert wurden. Als weitere Informationsquelle nutzen Marseu et al. die Verwaltungsschale (VWS) [MKW+17].

Hierbei handelt es sich um eine standardisierte, digitale Beschreibung, welche im Rahmen der Industrie-4.0-Bewegung für sämtliche Entitäten bzw. Assets innerhalb der Produktion genutzt wird. Nach Hildebrandt et al. referenziert die VWS auf „*die digitalen Engineering-Artefakte des repräsentierten Assets. Die Engineering-Artefakte werden in Teilmodellen geclustert, welche im Kontext der industriellen Produktion verschiedene Sichten auf die Informationen über das repräsentierte Assets darstellen. Die VWS selbst ist im Netz auffindbar und kann, mit entsprechenden Berechtigungen, nach den gewünschten Inhalten durchsucht werden*“. [HHD+22]

Um Daten und Beschreibungsmittel auszutauschen, schlagen Marseu et al. vor, die Module mit industrietauglichen Einplatinen-Computern auszustatten, um diese so an Datenbanken sowie das ERP-System anzubinden. So kann auch ein Retrofitting bei bestehenden Anlagenmodulen erzielt werden. Die Datenübermittlung kann beispielsweise mittels des OPC-UA-Standards oder des MQTT-Protokolls erfolgen. [MKW+17] Die Arbeit von Marseu et al. fokussiert sich auf die technische Umsetzung des Plug & Produce Ansatzes und betrachtet die Themen Modellbildung und Simulation der Produktionsanlage nicht.

Marks et al. stellen in ihrer Arbeit [MHW+18] einen Ansatz für die Anpassung von automatisierten Fertigungsmaschinen vor. Der Ansatz befasst sich mit dem gesamten Anpassungsprozess und stellt verschiedene Methoden für jede Prozessphase vor. Die Prüfung der Produzierbarkeit einer Produktionsanforderung auf der Basis eines geeigneten Fähigkeitsmodells des Systems wird ebenso behandelt wie die automatische Generierung von Anpassungsoptionen. Darüber hinaus werden Konzepte zur Analyse der Auswirkungen des Aufwands und des Nutzens der generierten Anpassungsoptionen vorgestellt. Marks et al. implementieren ihr Konzept als Entscheidungsunterstützungssystem mittels eines Agentensystems und validieren den Prototypen an einer Produktionsmaschine in Laborgröße [MHW+18]. Auch bei diesem Ansatz werden die Themen Modellbildung und Simulation der Produktionsanlage nicht betrachtet. Die Entscheidungsvorschläge des Agentensystems stellen jedoch eine weitere potenzielle Informationsquelle für die automatische Modellrekonfiguration dar.

In [MJW20] stellen Müller et al. das Konzept eines intelligenten Rekonfigurationsmanagements vor. Dieses ermittelt selbstständig Rekonfigurationsbedarfe, generiert alternative Konfigurationen, bewertet diese und wählt geeignete Konfigurationen aus. Die Ergebnisse dieses Ansatzes stellen, wie bereits beim Ansatz von Marks et al., eine weitere potenzielle Informationsquelle für die automatische Modellrekonfiguration dar. Als Voraussetzung für ein selbstorganisiertes Rekonfigurationsmanagement sehen Müller et al. die Existenz von semantischen Modellen an und identifizieren zudem „*einen Bedarf an Konzepten, die bestehende Algorithmen aus Domänen wie künstlicher Intelligenz oder mathematischer Optimierung [...] übernehmen*“.

In [YTZ+19] wird ein rein datenbasierter Ansatz zur Identifikation von cyber-physischen Systemen vorgestellt, welcher auf KI-Methoden basiert. Yuan et al. zeigen in dem Beitrag, dass der erarbeitete Algorithmus IHYDE erfolgreich komplexe mechanische Modelle direkt aus Messdaten identifiziert, einschließlich der Teilsystemdynamik und der damit verbundenen Übergangslogiken. Die vorgeschlagene Methode unterscheidet sich von klassischen maschinellen Lernverfahren, die in der Regel keinen Einblick in die zugrunde liegenden Mechanismen der Systeme geben, da die Zustandsvariablen und erlernten Parameter keine direkte Bedeutung haben.

Erkenntnisse und Defizite vorgestellter Konzepte für die Modellrekonfiguration:

Marseu et al. [MKW+17] beschreiben in ihrer Arbeit, wie Plug-&-Produce umgesetzt werden kann. Es werden Ansätze geliefert, wie die Anlagentopologie erkannt werden kann. Folglich kann Anforderung A7 (*Selbstständige Erkennung von Änderungen an realer Systemkonfiguration*) als vollständig erfüllt gewertet werden. Da Marseu et al. die Themen M&S in ihrer Arbeit nicht betrachten, wird Anforderung A8 (*Selbstständige Durchführung der automatischen Modellgenerierung (A1) unter Einbeziehung der Umbauänderungen*) als nicht erfüllt gewertet. Eine mögliche Struktur sowie die technische Realisierung für die Speicherung von Daten in der Cloud werden vorgeschlagen. Allerdings handelt es sich hierbei lediglich um einen Konzeptvorschlag ohne Validierung. Anforderung A9 (*Selbstständige Dokumentation von Änderungen an Anlagentopologie*) wird daher als teilweise erfüllt gewertet.

Der Ansatz von Marks et al. [MHW+18] ist hinsichtlich der Anforderungen ähnlich zu bewerten wie der Ansatz von Marseu et al. Das Agentensystem stellt eine weitere Informationsquelle dar, um Rekonfigurationen zu identifizieren. Anforderung A7 wird daher als vollständig erfüllt gewertet. M&S (A8) sowie die Umbaudokumentation (A9) liegen außerhalb des Fokus der Arbeit und werden daher als nicht erfüllt gewertet.

Der Ansatz von Yuan et al. [YTZ+19] unterscheidet sich hinsichtlich der eingesetzten Methoden von den anderen Arbeiten in dieser Kategorie. Dieser Ansatz verzichtet auf das Vorhandensein von Physikmodellen zur Identifikation hybrid dynamischer Systeme. Vorausgesetzt werden stattdessen die vollständige Zustandsmessung sowie Kenntnisse über die lineare Abhängigkeit der zu identifizierenden Parameter. Es stellt sich an dieser Stelle die Frage, wie realistisch diese Annahmen in der Praxis sind, wenn mit Produktionsmodulen von Zulieferern gearbeitet wird. Die Anforderungen A7 und A8 werden daher als teilweise erfüllt gewertet. Darüber hinaus wäre es nicht sinnvoll, auf Physikmodelle und das darin abgebildete Wissen zu verzichten, wenn diese zur Verfügung stehen. Der vorgestellte Ansatz könnte viel mehr eine Ergänzung darstellen, wenn für ein Anlagenmodul kein Modell existiert. Anforderung A9 zur Umbaudokumentation liegt außerhalb des Fokus der Arbeit und wird daher als nicht erfüllt gewertet.

3.1.4 Phasenübergreifende Lösungskonzepte

Nachdem in den vorherigen drei Abschnitten Konzepte und Ansätze betrachtet wurden, die sich auf eine Lebenszyklusphase beschränken, werden im letzten Schritt Lösungskonzepte aufgezeigt, die bereits Anforderungen aus mehreren Lebenszyklusphasen adressieren.

In [BER13] wird ein um die Betriebsphase erweiterter Ansatz zur automatischen Generierung von Modellen von mechatronischen Produktionssystemen beschrieben. Die automatische Modellgenerierung erfolgt auf Basis von CMSD-Daten, ähnlich den bereits in Abschnitt 3.1.1 vorgestellten Ansätzen. Darüber hinaus findet eine Adaption der generierten Modelle auf Basis von Künstlicher Intelligenz statt. Vereinfachend wird hier angenommen, dass alle Modelle aus einer homogenen Quelle stammen. Bergmann definiert Prioritätsregeln für Reihenfolgeentscheidungen bei der Bearbeitung von Aufträgen, wie „kürzeste Bearbeitungszeit“, „frühester Liefertermin“ und „kürzeste Rüstzeit“. Auf dieser Basis wird ein künstliches neuronales Netzwerk trainiert, welches die Modelladaption durchführt.

Sokoll und Clausing [SC20] stellen in ihrer Arbeit Methoden und Werkzeuge der Simulationsassistentz für Ablaufsimulationen in der Automobilindustrie vor. Es werden Ansätze zur automatischen Modellgenerierung auf Basis von Taktzeitdiagrammen, Layoutstrukturen und Prozessinformationen vorgestellt. Die Taktzeitdiagramme werden für die Simulation von Arbeitsfolgen robotergesteuerter Fertigungsanlagen im Karosseriebau eingesetzt. Um eine automatische Generierung der Materialflussstruktur in einem Modell zu ermöglichen, stellen Sokoll und Clausing einen Ansatz vor, der als Grundlage ein digitales Hallen- und Anlagenlayout des Fördertechniksystems nutzt. Dieses wird nach [SC20] in der Regel im Laufe der Planung in einer CAD-Software erstellt. Die benötigten Daten über Fördertechnikelemente und ihre Eigenschaften können mithilfe einer Schnittstelle zwischen der CAD-Software und einem Simulationswerkzeug übertragen werden. Darüber hinaus thematisieren Sokoll und Clausing in ihrer Arbeit die teilautomatische Dokumentation im Rahmen der Ablaufsimulation. Als Ansätze werden standardisierte Dokumententemplates und Assistenzfunktionen für die Erstellung von Berichten genannt.

Talkhestani [TAL20] beschreibt in seiner Arbeit zur Ankerpunktmethodologie einen Ansatz, der eine automatisierte Änderungsdetektion und Modellanpassung ermöglicht. Hierzu konzipiert und implementiert Talkhestani ein dreistufiges Assistenzsystem. Dieses führt zunächst eine automatisierte Änderungsdetektion durch, anschließend eine domänenübergreifende Abhängigkeitsanalyse und zuletzt die Modellanpassung. In der ersten Phase findet ein Vergleich zwischen der SPS-Steuerungssoftware des automatisierten Systems zum Referenzzeitpunkt und zum aktuellen Zeitpunkt unter Verwendung einer Formalisierungsmethode statt. Eine regelbasierte Analyse der erfassten Änderungen in der Steuerungssoftware wird in der zweiten Phase durchgeführt, um die domänenübergreifenden Abhängigkeiten der Änderungen sowie das auftretende Änderungsszenario im System zu identifizieren. Die Referenzierung der domänenübergreifenden Modelle und ihrer Anpassungsparameter an den geänderten Ankerpunkten des Gesamtsystemmodells erfolgt in der letzten Phase durch die Verwendung einer Komponentenbibliothek und die Erstellung von gekapselten Change-Request-Modellen. Der Ansatz von Talkhestani geht davon aus, dass abschließend ein Systemingenieur die Änderungen im Gesamtsystemmodell unter Verwendung der Informationen und Modelle innerhalb des Change-Request-Modells ausführt. [TAL20]

Erkenntnisse und Defizite vorgestellter phasenübergreifender Konzepte:

Bergmann deckt in seiner Arbeit [BER13] zwei der drei Lebenszyklusphasen teilweise ab, konkret die Phasen Inbetriebnahme und Betrieb. Die Anforderungen A1 und A2 bezogen auf die automatische Modellgenerierung und die phasenübergreifende Eignung des Ansatzes können als vollständig erfüllt gewertet werden. Bergmann führt die Adaption der Modelle allerdings nicht auf Basis von Messdaten durch. Stattdessen werden Prioritätsregeln für Reihenfolgeentscheidungen bei der Bearbeitung von Aufträgen definiert. Auf dieser Basis wird ein künstliches neuronales Netzwerk trainiert, welches zur Adaption der Anlagenmodelle genutzt werden kann, um die Prioritätsregeln umzusetzen. Die Anforderungen A3 bis A5 werden daher als nicht erfüllt gewertet. Lediglich Anforderung A6 zur automatischen Durchführung der Adaption wird als vollständig erfüllt gewertet.

Sokoll und Clausing [SC20] stellen in ihrer Arbeit Ansätze für die Ablaufsimulation vor. Dies deckt, verglichen mit Abbildung 10, allerdings nur einen Teilbereich des Einsatzbereiches von Simulationstechnik ab. Hinzu kommt, dass die Autoren nicht explizit von modularen Anlagen ausgehen und die Modelle nicht aus heterogenen Quellen stammen. Es werden standardisierte Datenformate zur (teil-)automatischen Mo-

dellgenerierung genutzt, auf welche in weiteren Lebenszyklusphasen zugegriffen werden könnte. Entsprechend werden die Anforderungen A1 und A2 als teilweise erfüllt eingeschätzt. Der betriebsparallele Modelleinsatz und entsprechend die betriebsparallele Modelladaptation wird nicht betrachtet. Das Thema Dokumentation wird von Sokoll und Clausing aufgegriffen. Der Fokus liegt hierbei allerdings nicht auf der Umbaudetektion und -dokumentation, wie in A7 bis A9 gefordert. Zwar schließt die automatische Dokumentation in [SC20] die Anlage einschließlich Prozess- und Datenbeschreibung sowie das ausführbare, valide und verifizierte Modell ein. Dabei steht jedoch die Dokumentation der Experimentbeschreibung und Ergebnisdokumentation einschließlich abgeleiteter Maßnahmen und Empfehlungen im Vordergrund.

Talkhestani [TAL20] setzt mit seiner Arbeit zur Synchronisierung der Modelle des Digitalen Zwillings automatisierter Systeme nach der Inbetriebnahmephase ein. Folglich werden die Anforderungen A1 und A2 als nicht erfüllt gewertet. Durch die Analyse der aktuellen SPS-Steuerungssoftware ist der Ansatz sowohl in der Lage Mess- und Simulationswerte zu vergleichen als auch strukturelle Änderungen an der Anlage zu identifizieren. Allerdings beschränkt sich der Anwendungsbereich auf CAD-Modelle. Weiter lassen sich mit der Ankerpunktmethod nur Änderungen feststellen, die einen Einfluss auf die Steuerungssoftware haben. Ausschließlich mechanische oder elektrische Änderungen sind mit dieser Methode nicht zu erkennen. Die Anforderungen A3 sowie A5 bis A8 werden daher als teilweise erfüllt gewertet. Eine Toleranzüberwachung entsprechend Anforderung A4 wird nicht durchgeführt und daher als nicht erfüllt gewertet. Änderungen werden nicht dokumentiert, Anforderung A9 wird daher ebenfalls als nicht erfüllt gewertet. Der Ansatz von Talkhestani geht davon aus, dass abschließend ein Systemingenieur die Änderungen im Gesamtsystemmodell ausführt: *„Die Erkennung der exakten mechanischen Lokalisierungen der Änderungen ist jedoch eine Herausforderung, die mit der Ankerpunktmethod allein nicht automatisiert zu bewältigen ist und zwangsläufig die Zusammenarbeit mit einem Systemingenieur erfordert.“* [TAL20]. Folglich handelt es sich bei dem vorgestellten Assistenzsystem lediglich um ein Entscheidungsassistenzsystem.

3.2 Bewertung bestehender Konzepte hinsichtlich definierter Anforderungen

Nachdem in Abschnitt 3.1 bestehende Lösungskonzepte vorgestellt wurden, die eine oder mehrere Lebenszyklusphasen einer Produktionsanlage abdecken, gilt es im nächsten Schritt die Konzepte hinsichtlich der definierten Anforderungen im Detail abzugleichen. Bei Konzepten, die nur eine Lebenszyklusphase abdecken, werden Anforderungen aus anderen Phasen ohne weitere Betrachtung als nicht erfüllt eingestuft. Die Bewertung der vorgestellten Ansätze hinsichtlich der Erfüllung der gestellten Anforderungen ist in Tabelle 7 zusammengefasst. In den Zeilen sind die Referenzen der jeweiligen Ansätze eingetragen, untergliedert entsprechend der Unterabschnitte 3.1.1 bis 3.1.4. In den Spalten sind die Anforderungen, ebenfalls entsprechend der Lebenszyklusphasen untergliedert, vermerkt. Um darzustellen, inwieweit Anforderungen erfüllt sind, werden Harvey-Bälle eingesetzt. Es wird unterschieden zwischen nicht, teilweise oder vollständig erfüllten Anforderungen. Die Visualisierung der Harvey-Bälle ist der Legende aus Tabelle 7 zu entnehmen.

Tabelle 7: Metrik zur Bewertung bestehender Ansätze hinsichtlich der definierten Anforderungen.

Anforderungen		Phase 1		Phase 2				Phase 3		
		A1	A2	A3	A4	A5	A6	A7	A8	A9
Referenz										
Konzepte zur (teil-)automatischen Modellgenerierung	[BAR11, OWD+14]	●	◐	○	○	○	○	○	○	○
	[WÜN07]	●	◐	○	○	○	○	○	○	○
	[JSW18]	●	●	○	○	○	○	○	○	○
	[SCH20]	●	◐	○	○	○	○	○	○	○
	[TFD18]	●	●	○	○	○	○	○	○	○
	[UWQ17]	●	●	○	○	○	○	○	○	○
	[BLO19]	●	●	○	○	○	○	○	○	○
Konzepte zur (teil-)automatischen Modelladaption	[EM18]	○	○	◐	○	◐	●	○	○	○
	[DSBF+20]	○	○	◐	○	◐	●	○	○	○
	[MJW22]	○	○	●	●	◐	●	○	○	○
	[ZAS+18]	○	○	●	○	●	○	○	○	○
Konzepte zur (teil-)automatischen Modellrekonfiguration	[MKW+17]	○	○	○	○	○	○	●	○	◐
	[MHW+18]	○	○	○	○	○	○	●	○	○
	[MJW20]	○	○	○	○	○	○	●	○	○
	[YTZ+19]	○	○	○	○	○	○	◐	◐	○
Phasenübergreifende Lösungskonzepte	[BER13]	●	●	○	○	○	●	○	○	○
	[SC20]	◐	◐	○	○	○	○	○	○	◐
	[TAL20]	○	○	◐	○	◐	◐	◐	◐	○

Legende:

vollständig erfüllt



teilweise erfüllt



nicht erfüllt

Nach der Bewertung der vorgestellten Ansätze hinsichtlich der Anforderungen sowie dem Ausarbeiten relevanter Erkenntnisse und Defizite wird im folgenden Abschnitt die Metrik ausgewertet.

3.3 Erkenntnisse und abgeleiteter Handlungsbedarf

Aus der Metrik in Tabelle 7 lässt sich ableiten, dass keine der betrachteten Lösungen den kompletten Lebenszyklus einer modularen Produktionsanlage aus der Perspektive von M&S abdeckt. Bei den Lösungskonzepten, die explizit nur eine Lebenszyklusphase betrachten, ist dieser Zusammenhang direkt gegeben. Bei den phasenübergreifenden Lösungskonzepten wird deutlich, dass jeweils nur zwei der drei betrachteten Lebenszyklusphasen (teilweise) abgedeckt werden. Entsprechend werden auch bei keiner der Lösungen alle in Abschnitt 2.2.4 formulierten Anforderungen gänzlich erfüllt. Es besteht daher die Notwendigkeit, ein neues Lösungskonzept zu erarbeiten. Dabei ist es sinnvoll, etablierte Aspekte und (Teil-)Konzepte der existierenden Lösungen in das AS zu übernehmen. Diese können, entsprechend den Anforderungen, neu kombiniert, angepasst und erweitert werden, um so Forschungsfrage 1 (*Ist die (teil-)automatische Generierung, Adaption und (Re-)Konfiguration von Modellen für modulare Produktionsmaschinen und -anlagen realisierbar?*) zu beantworten und das Ziel der Arbeit erreichen zu können.

In Abschnitt 3.4 wird eine Übersicht für ein Gesamtkonzept vorgestellt, welches alle Anforderungen erfüllt und somit eine lebenszyklusübergreifende (teil-)automatisierte Modellgenerierung, -adaption und -rekonfiguration von Simulationsmodellen für modulare Produktionsanlagen ermöglicht. Bei der Untersuchung des Standes von Wissenschaft und Technik konnten verschiedene wiederkehrende Elemente, wie Beschreibungsmittel, Methoden und Werkzeuge, identifiziert werden. Dies liefert wichtige Erkenntnisse für die Beantwortung der in Abschnitt 1.2 formulierten Forschungsfragen 2 (*Welche Voraussetzungen müssen erfüllt werden, um diese (Teil-)Automatisierung zu erreichen?*) und 3 (*Welche Informationen werden für die (Teil-)Automatisierung der einzelnen Teilschritte benötigt?*). Die Erkenntnisse aus der Analyse des Standes von Wissenschaft und Technik sind nachstehend zusammengefasst.

Zunächst wird entsprechend Forschungsfrage 1 dargestellt, welche Aspekte und (Teil-)Konzepte übernommen werden können, welche mit entsprechender Anpassung teilweise übernommen werden können und welche neu entwickelt werden müssen. Hierzu wird die Realisierung der Automatisierung von Modellierungstätigkeiten, analog zu den drei betrachteten Lebenszyklusphasen, untergliedert. Anschließend werden die identifizierten Voraussetzungen und benötigten Informationen aufgelistet.

Realisierung der Automatisierung von Modellierungstätigkeiten entsprechend Forschungsfrage 1:

a) *(teil-)automatische Modellgenerierung und -konfiguration*

Die (teil-)automatische Modellgenerierung und -konfiguration ist, entsprechend dem Stand von Wissenschaft und Technik, möglich und für verschiedene Industrieapplikationen nachgewiesen. Die Heterogenität der MSM z. B. hinsichtlich ihrer Quellen und Domänen wird durch den Einsatz von Co-Simulation handhabbar. Es ist daher erforderlich, diese Simulationsarchitektur einzusetzen. Zudem basieren alle Ansätze auf einem Beschreibungsmittel der betrachteten Anlage. Hieraus lassen sich die nötigen Informationen wie MSM Auswahl, Kopplung und Initialisierung ableiten. Durch die Wahl von Standards für Co-Simulation und Beschreibungsmittel lässt sich das erarbeitete Konzept auf die weiteren Lebenszyklusphasen erweitern.

b) *(teil-)automatische Modelladaption*

Die (teil-)automatische Modelladaption ist insgesamt betrachtet, entsprechend dem Stand von Wissenschaft und Technik, teilweise möglich. Die vorgestellten Ansätze versuchen Systemwissen,

welches durch physikalische oder mathematische Zusammenhänge dargestellt wird, durch datenbasierte Ansätze anzureichern. Hierzu werden Konzepte und Algorithmen aus dem Bereich des Machine Learnings eingesetzt. Die Ansätze, die eine Adaption durchführen, betrachten lediglich monolithische Anlagen und entsprechende Modelle. Nur ein Ansatz betrachtet modulare Anlagen und setzt für deren Simulation Co-Simulation ein. Dafür wird in diesem Ansatz keine Adaption durchgeführt. Es ist daher nötig, die Ansätze zu kombinieren, um alle Anforderungen für diese Lebenszyklusphase zu erfüllen.

c) *(teil-)automatische Modellrekonfiguration*

Die (teil-)automatische Modellrekonfiguration ist insgesamt betrachtet, entsprechend dem Stand von Wissenschaft und Technik, teilweise möglich. Es existieren verschiedene sensorbasierte Ansätze (z. B. mittels RFID-Tags, 3D-Laserscan) zur Erkennung von Änderungen in der Systemkonfiguration auf modularer Ebene. Diese Ansätze können übernommen werden. Die Abbildung vorgenommener Änderungen in das Systemmodell wird nicht vollumfänglich erfüllt. Auch eine ausreichende Dokumentation der vorgenommenen Änderungen wird nur teilweise erfüllt. Die Umsetzbarkeit dieser zwei Aspekte ist von Entscheidungen bei der Konzeptionierung der (teil-)automatischen Modellgenerierung und -konfiguration abhängig (z. B. Wahl der Standards für Co-Simulation und Beschreibungsmittel) und muss entsprechend berücksichtigt werden. Hier müssen bestehende Ansätze angepasst und neu entwickelt werden.

Hinsichtlich der zu erfüllenden Voraussetzungen und der benötigten Informationen lassen sich hieraus folgende Punkte ableiten.

Zu erfüllende Voraussetzungen entsprechend Forschungsfrage 2:

- Existenz einer Beschreibung der geplanten Produktionsanlage
- Existenz aller MSM, die im Rahmen der Gesamtsystemsimulation gekoppelt werden sollen
- MSM verfügen über standardisierte Schnittstellen
 - Ermöglicht eine Automatisierung der Komposition
 - Gewährleistet die Integration heterogener MSM
- Existenz weiterer Daten- und Informationsquellen (wird im nächsten Absatz aufgelistet)

Benötigte Informationen entsprechend Forschungsfrage 3:

- Standardisiertes Beschreibungsmittel für die Beschreibung der Anlage
- Bedienereingaben (z. B. Vorgabe der Abweichungstoleranz, Einschränkung der zu optimierenden Modellparameter)
- Sensordaten für Modelladaption
- Sensordaten für Detektion von Änderungen der Anlagentopologie

Im nächsten Abschnitt wird auf Basis dieser Erkenntnisse eine Gesamtkonzeptübersicht gegeben. Auf die jeweiligen Elemente und Bestandteile der Konzeptübersicht wird in den folgenden Kapiteln und Abschnitten entsprechend der Lebenszyklusphasen im Detail eingegangen.

3.4 Gesamtkonzeptübersicht

In Abbildung 19 ist eine Übersicht des erarbeiteten Gesamtkonzepts für den lebenszyklusübergreifenden Einsatz des AS dargestellt. Das Zentrum der Abbildung bildet das AS, welches grün hervorgehoben ist. Neben dem AS sind als Bestandteile des Gesamtkonzepts die modulare Produktionsanlage (links in der Abbildung), eine Bibliothek mit den MSM (oben), der Bediener des AS (unten) sowie Modell und Simulation der Anlage (rechts) zu nennen. Für die Interaktion mit allen genannten Bestandteilen benötigt das AS entsprechende Schnittstellen zum Daten- und Informationsaustausch. Zur Anlage benötigt das AS Schnittstellen, über die es eine Beschreibung der Anlage sowie Sensordaten erhält. Über eine Schnittstelle zur MSM-Bibliothek erhält es Zugriff auf MSMs mit standardisierten Schnittstellen. Über eine Schnittstelle, z. B. eine GUI, muss es dem Bediener möglich sein, Eingaben vorzunehmen. Im Gegenzug soll es möglich sein, dass das AS den Bediener bei komplexen Entscheidungen unterstützt. Letztlich benötigt das AS Schnittstellen zu Gesamtmodell und Simulation der Anlage. Über die Simulation erhält das AS Simulationsdaten. Alle in den Anforderungen beschriebenen M&S-Tätigkeiten (z. B. MSM laden, initialisieren, koppeln, simulieren, dokumentieren) müssen vom AS durchgeführt werden. Welchen Automatisierungsgrad das AS bei diesen M&S-Tätigkeiten, entsprechend Forschungsfrage 4 (*Welcher Automatisierungsgrad kann für die einzelnen Lebenszyklusphasen erzielt werden?*) erreichen kann, ist im Rahmen der Validierung entsprechend der in Abschnitt 2.2.5 definierten Automatisierungslevel einzustufen.

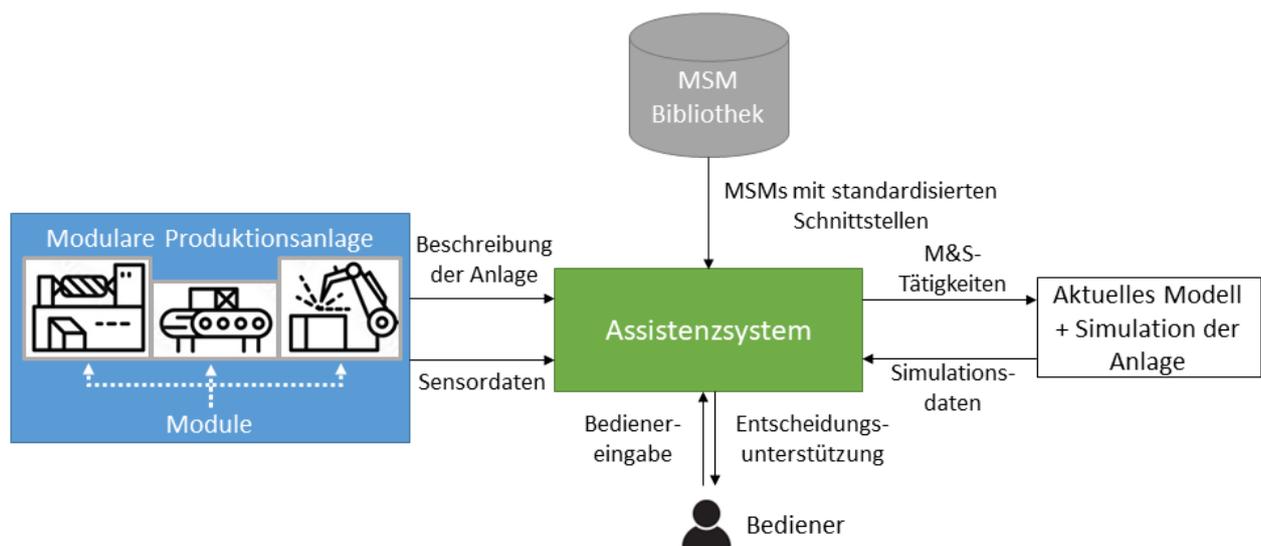


Abbildung 19: Übersicht des Gesamtkonzepts für den lebenszyklusübergreifenden Einsatz des AS.

Nachdem der Stand von Wissenschaft und Technik diskutiert, die Defizite bestehender Lösungen aufgezeigt und eine Übersicht des Gesamtkonzepts zur Erfüllung der definierten Anforderungen vorgestellt wurde, werden in den nächsten drei Kapiteln die einzelnen Teilkonzepte entsprechend der Lebenszyklusphasen herausgegriffen und im Detail behandelt. Die Kapitel folgen dabei einer einheitlichen Struktur. Zunächst werden das jeweilige Teilziel und die Anforderungen, die im Rahmen dieses Kapitels erfüllt werden sollen, aufgezeigt. Im Anschluss werden spezifische technologische Konzeptbausteine vorgestellt. Es folgt eine detaillierte Ausarbeitung des zugehörigen Teilkonzepts, abgeleitet aus der Gesamtkonzeptübersicht aus Abbildung 19. Das jeweilige Teilkonzept wird anschließend entsprechend dem Design-Zyklus nach Wieringa [WIE14] prototypisch implementiert und validiert. Jedes der drei Kapitel schließt mit einem eigenen Zwischenfazit ab.

Kapitel 4 legt den Fokus auf die (teil-)automatische Modellgenerierung und -konfiguration im Rahmen der Inbetriebnahmephase.

4 Konzeptbaustein für die Modellgenerierung und -konfiguration

Entsprechend der chronologischen Reihenfolge der Lebenszyklusphasen einer Produktionsanlage wird als erstes die Inbetriebnahmephase betrachtet. In dieser Phase stehen dem AS Engineering-Artefakte und die MSM der einzelnen Module zur Verfügung. Auf Basis dessen soll die initiale Modellgenerierung und Konfiguration erfolgen.

Im Rahmen dieses Kapitels werden zunächst in Abschnitt 4.1 die Teilziele, geltenden Randbedingungen und Annahmen aufgezeigt. Ebenfalls wird kurz rekapituliert, welche der in Abschnitt 2.2.4 definierten Anforderungen im Rahmen dieses Kapitels erfüllt werden. In Abschnitt 4.2 werden die für dieses Kapitel relevanten technologischen Konzeptbausteine behandelt. Anschließend wird in Abschnitt 4.3 das in Abschnitt 3.4 vorgestellte Gesamtkonzept auf die Inbetriebnahmephase reduziert und das Teilkonzept im Detail vorgestellt. Die prototypische Implementierung des Teilkonzepts wird in Abschnitt 4.4 behandelt und anschließend in Abschnitt 4.5 auf die beiden Industrieszenarien aus Abschnitt 2.3 angewandt und validiert. Das Kapitel schließt mit einem Zwischenfazit in Abschnitt 4.6.

4.1 Teilziele, Randbedingungen und Annahmen

Wie in Kapitel 1 und 2 aufgezeigt, bilden manuelle Tätigkeiten eine der wesentlichen Herausforderungen beim Einsatz von Simulation im Engineering modularer Produktionsanlagen. Ziel ist es daher, die manuellen Tätigkeiten durch automatische Abläufe zu ersetzen bzw. die Mitarbeiterinnen und Mitarbeiter bei deren Arbeit zu unterstützen. In einem ersten Schritt ist daher Unterstützung beim Aufbau der Gesamtsystemsimulation notwendig. Ziel des in diesem Kapitel beschriebenen Teilkonzepts ist es, dass die Komposition und Konfiguration von Co-Simulationen mithilfe von einem Assistenzsystem (AS) (teil-)automatisiert erfolgt. Der Fokus liegt auf der Phase der Inbetriebnahme. Dabei muss das AS die folgenden beiden Hauptanforderungen erfüllen:

A1 Das AS muss in der Lage sein, selbstständig:

- Modulmodelle zu laden,
- Initialwerte/ Anfangszustände zu setzen,
- die Modulmodelle, analog zur realen Systemkonfiguration, korrekt zu koppeln sowie
- die Simulation zu initiieren und auszuführen.

A2 Das AS muss derart konzipiert werden, dass eine Erweiterung des Einsatzbereichs auf den kompletten Anlagenlebenszyklus möglich ist.

Um den Vorgang der Komposition und Konfiguration zu automatisieren, müssen mehrere Voraussetzungen erfüllt sein. Zum einen müssen Simulationsmodelle für alle Module, die im Rahmen der Gesamtsystemsimulation gekoppelt werden sollen, zur Verfügung stehen. Diese Modulsimulationsmodelle (MSM) müssen zudem über standardisierte Schnittstellen verfügen, um eine Automatisierung der Komposition zu ermöglichen. Mittels standardisierter Schnittstellen wird gewährleistet, dass heterogene MSM in die Co-Simulation integriert werden können. Darüber hinaus muss eine Beschreibung der geplanten Produktionsanlage existieren. Im nächsten Abschnitt wird daher zunächst untersucht, welche Lösungen es bereits gibt, um diese Voraussetzungen zu realisieren.

4.2 Technologische Konzeptbausteine

Zunächst werden in Abschnitt 4.2.1 mögliche Beschreibungsmittel von Produktionsanlagen betrachtet und bewertet, die für das Teilkonzept der automatischen Komposition und -konfiguration relevant sein können. Im Anschluss wird in Abschnitt 4.2.2 Co-Simulation im Model-Based Systems Engineering betrachtet. Hierfür werden zunächst in Abschnitt 4.2.2.1 grundlegende Begriffe und in Abschnitt 4.2.2.2 Kompositionsabläufe einer Co-Simulation vorgestellt. Anschließend wird in Abschnitt 4.2.2.3 eine Übersicht über relevante Schnittstellenstandards und Tools gegeben. Basierend auf den Erkenntnissen dieses Abschnitts wird ein Standard für den weiteren Verlauf dieser Arbeit ausgewählt und im Detail in Abschnitt 4.2.2.4 beschrieben. Abschließend werden in Abschnitt 4.2.2.5 Laufzeitaspekte, welche für die Ausführung einer Co-Simulation relevant sind, vorgestellt. Die Abschnitte 4.2.2.1 bis 4.2.2.3 sowie der Abschnitt 4.2.2.5 beruhen maßgeblich auf den Gremiumsarbeiten im VDI/VDE-GMA-Fachausschusses 6.11 „Virtuelle Inbetriebnahme“, welche bereits in [SHR+21*] veröffentlicht wurden.

4.2.1 Beschreibungsmittel von Produktionsanlagen

Aufgrund der steigenden Komplexität moderner produktionstechnischer Systeme erfolgt deren Entwicklung zunehmend in interdisziplinären Projektteams. Diese Teams setzen sich aus Mitgliedern unterschiedlicher Fachbereiche (Domänen) zusammen, wie beispielsweise Vertriebsmitarbeiter, Konstrukteure, Informatiker oder Designer. Im iterativen Entwicklungsprozess generieren sie Inhalte, die wiederum für andere Entwickler als Arbeitsgrundlage dienen. Dieser interdisziplinäre Informationsaustausch stellt heute immer noch ein Kernproblem bei der Maschinenentwicklung dar. Um diesen interdisziplinären Informationsaustausch strukturiert auszuführen, kommen die etablierten Beschreibungsmittel der jeweiligen Entwickler zum Einsatz [HOM+18].

Nach Schnieder [SCH99] dient ein Beschreibungsmittel der formalen Formulierung der Aufgabe und Lösung eines zugehörigen Problems. Beschreibungsmittel sind eine elementare Voraussetzung „zur Formulierung der planmäßigen Vorgehensweise, der Aufgabenstellung sowie zu deren Lösung und ihrer Realisierungsmittel und letztlich sogar zur Beschreibung von Anlagenzuständen und zur Bedienungshandhabung im Betrieb“ [SCH99]. Als Beispiele für Beschreibungsmittel werden in [HOM+18] Prinzipskizzen, Schaltpläne oder Funktionsdiagramme sowie Petri-Netze und Flussdiagramme genannt. Als Skizzen, in Zeichnungen, Schaltplänen und Stücklisten werden die Entwicklungsergebnisse dokumentiert und abgelegt. Es findet ein Austausch der entstehenden Dokumente zwischen Entwicklern statt, welche im anschließenden Entwicklungsschritt als Arbeitsgrundlage dienen. [HOM+18]

Da entsprechend den Ausführungen aus Abschnitt 2.2.2 zukünftig Assistenzsysteme maßgeblich an den Modellierungs- und Simulationsaufgaben beteiligt sein sollen, gilt es zunächst zu untersuchen, wie der Austausch der Beschreibungen realisiert werden kann. Es wird an dieser Stelle angenommen, dass die nötigen Beschreibungen aus vorherigen Entwicklungsschritten der modularen Produktionsanlage durch Softwaretools erzeugt wurden und in digitaler Form vorliegen. Die Übergabe von Inhalten zwischen verschiedenen Softwaretools kann nach [HOM+18] auf drei unterschiedlichen Wegen erfolgen. Die rudimentärste Variante stellt das *manuelle Übertragen* der Inhalte durch einen Entwickler dar. Alternativ können *standardisierte Export-Import-Formate*, wie beispielsweise DXF, STL, Collada, AutomationML, SysML oder PLCopen-XML verwendet werden. Als dritte Variante ist noch die direkte Interaktion der Werkzeuge durch angebotene *Schnittstellen*, wie z. B. MS Office DDE API, TwinCAT oder IndraWorks Automation Interface

und SolidWorks API zu nennen. Alle drei Varianten, um Beschreibungsmittel zwischen Tools zu transferieren, sind weiter nach [HOM+18] mit spezifischen Problemen und Herausforderungen behaftet. Bei der manuellen Übertragung kann es zu Übertragungs- und Interpretationsfehlern kommen. Zudem ist diese entsprechend zeit- und somit kostenintensiv. Wenn der Import und Export von Dateiformaten nur unzureichend von den beteiligten Tools unterstützt wird, kann auch hier der Transfer fehlerbehaftet sein. Als mögliche Fehlerquelle wird in [HOM+18] beispielsweise die unterschiedliche Repräsentation von Inhalten genannt. Bei der letzten Variante, der direkten Interaktion von Tools über spezielle Schnittstellen, wird die Anwendung durch nicht standardisierte Implementierungen erschwert.

Da der Grundgedanke des AS die Reduktion der manuellen Tätigkeiten im Bereich M&S für modulare Produktionsanlagen ist, wird die manuelle Übertragung von Beschreibungsmitteln als nicht zielführend erachtet. Des Weiteren soll das Konzept des AS möglichst vielseitig einsetzbar sein. Die Festlegung auf eine Schnittstelle, wie es in Variante drei beschrieben ist, ist daher ebenfalls nicht zielführend. Es kann a priori nicht festgelegt werden, welche Tools in vorherigen Entwicklungsschritten bei unterschiedlichen Maschinen- und Anlagenbauern zum Einsatz kommen. Um den Anwendungsbereich des im Rahmen dieser Arbeit vorgestellten Konzepts möglichst wenig einzuschränken, wird daher der Transfer von Beschreibungen mittels standardisierter Export-Import-Formate realisiert. Nach der Untersuchung, wie Beschreibungen grundsätzlich ausgetauscht werden können, und der Festlegung, wie der Austausch im Rahmen dieser Arbeit konkret erfolgen soll, gilt es abschließend noch zu diskutieren, welches Beschreibungsmittel die Basis für das AS darstellen und welches standardisierte Dateiformat verwendet werden soll.

Um die automatische Komposition und Konfiguration der modularen Produktionsanlage durchführen zu können, benötigt das AS Informationen über die Anlage, deren Module sowie der jeweiligen MSM. Die benötigten Informationen können von den Anforderungen aus Abschnitt 4.1 abgeleitet werden. Anforderung A2 bezieht sich auf die Erweiterbarkeit des AS auf den kompletten Anlagenlebenszyklus. Hieraus ergibt sich kein direkter Bedarf an Beschreibungsmitteln. Es gilt allerdings zu überprüfen, ob in den Kapiteln 5 und 6, die sich den folgenden Lebenszyklusphasen annehmen, noch weitere Beschreibungsmittel benötigt werden. Betrachtet wird daher im Weiteren Anforderung A1. Es werden vier Aufgaben formuliert, die das AS im Rahmen dieser Anforderung selbstständig erfüllen soll. Zunächst sollen die *MSM geladen* und *Initialwerte/ Anfangszustände gesetzt* werden. Hierzu müssen dem AS Informationen darüber vorliegen, welche Module Teil der Anlagenkonfiguration sind, an welchen Speicherstellen (lokale Festplatte, Netzwerklaufwerk, Cloud, etc.) die MSM hinterlegt sind und mit welchen Parameterwerten diese initialisiert werden sollen. Anschließend sollen die Modulmodelle, analog zur realen Systemkonfiguration, korrekt vom AS gekoppelt werden. Entsprechend benötigt das AS die Kopplungsinformationen. Die Anordnung der Module wird als Topologie bezeichnet [GWQ+16]. Für die Beschreibung der Anlage sowie deren Module und der bestehenden Kopplungen bietet sich ein Topologiemodell (TM) an. Das TM beschreibt die Anlage als eine hierarchische Struktur von Anlagenobjekten und bildet ab, wie diese untereinander verbunden sind [DLP+08]. Objekte besitzen individuelle Eigenschaften und Schnittstellen, über die sie mechanisch, elektrisch oder informationstechnisch mit anderen Objekten verbunden sein können [DLP+08]. Als Basisformat für das TM wird AutomationML (AML) gewählt, da es sich hierbei um einen etablierten Standard im Engineering automatisierter Anlagen handelt [HMA+16, GRI16, DRA21, AUT22[®]]. Die jeweiligen Speicherstellen der MSM können im TM ebenfalls als Objekte abgebildet und referenziert werden. Struktur und Aufbau der TM werden im Detail in Abschnitt 4.4.1 behandelt. Abschließend soll

das AS die *Simulation initiieren und ausführen*. Für die letzte Aufgabe müssen dem AS die Simulationsparameter, beispielsweise die Start- und Stoppzeit sowie die Schrittweite, zur Verfügung gestellt werden. Da die Simulationsergebnisse, wie in Abschnitt 2.2.1 beschrieben, für unterschiedliche Interessengruppen bestimmt sind, müssen entsprechend unterschiedliche Szenarien untersucht werden können. Folglich müssen die Simulationsparameter einfach durch den Bediener des AS konfigurierbar sein. Um flexibel Simulationsszenarien vorgeben und testen zu können, bieten sich für diese Aufgabe Bedienereingaben über eine graphische Bedienoberfläche (engl. *graphical user interface*, kurz GUI) als Informationsquelle an.

Nach der Untersuchung zum Thema Beschreibungsmittel befassen sich die nächsten Abschnitte mit dem Thema Co-Simulation.

4.2.2 Co-Simulation im Model-Based Systems Engineering

Wie in Abschnitt 3.3 aufgezeigt, wird durch den Einsatz von Co-Simulation die Heterogenität der MSM z. B. hinsichtlich ihrer Quellen und Domänen handhabbar. Eine lebenszyklusübergreifende Lösung zur (Teil-)Automatisierung aller Modellierungstätigkeiten im Zusammenhang mit modularen Produktionsanlagen muss auf dieser Simulationsarchitektur basieren.

4.2.2.1 Begriff und Verständnis der Co-Simulation

In der Literatur wird der Begriff der Co-Simulation unterschiedlich verwendet (vgl. [DRO04, GKL06, SCH15, GÜN17]). Dronka [DRO04] bezeichnet die Kopplung von Gleichungslösern (engl. Solver) als verteilte Integration, modulare Simulation, Simulations-Kopplung oder Co-Simulation. Geimer et al. [GKL06] versuchen in Anlehnung an Dronka eine Begriffsvereinheitlichung, die auf der Unterteilung hinsichtlich der Anzahl verwendeter Modellierungswerkzeuge sowie der Anzahl eingesetzter Solver basiert. Dabei verwenden Geimer et al. den Begriff Co-Simulation für den Einsatz mehrerer Solver oder mehrerer Modellierungswerkzeuge. Auch Schmoll [SCH15] verwendet den Begriff der Co-Simulation, auch Integrator-, Simulator- oder Solverkopplung, für die Kopplung sowohl von mehreren Modellierungsumgebungen als auch mehreren Solvern. Günther [GÜN17] fasst die Co-Simulation allgemeiner und versteht darunter eine verteilte Simulation – die Kopplung mehrerer Solver ohne Bezug zur Anzahl der Modellierungswerkzeuge. [SHR+21*]

In der praktischen Anwendung beinhaltet eine Co-Simulation gekoppelte Teilinstanzen, welche einzelne Komponenten-, Maschinen- und Anlagenmodelle, aber auch Steuerungs- und Visualisierungskomponenten darstellen, die zu einer Gesamtmodell-Berechnung gekoppelt sind. Eine Abgrenzung der Co-Simulation entsprechend [GMA20] zu weiteren Simulationsarchitekturen hinsichtlich der Anzahl an Solvern und Simulationsmodellen ist in Abbildung 20 dargestellt. Hierbei stellt die „klassische“ monolithische Simulation die Konfiguration aus einem Simulationsmodell und einem Solver dar. Als Ausgangssituation ist sie unten links in der Abbildung verortet. Wird nun das monolithische Modell in mehrere Teilmodelle aufgespaltet und diese weiterhin von einem Solver gelöst, handelt es sich um eine Simulationsmodellkopplung. Diese Simulationsarchitektur ist oben links in Abbildung 20 dargestellt. Wenn umgekehrt, statt mehrerer Modelle und einem Solver, mehrere Solver und ein Modell zum Einsatz kommt, handelt es sich bei der Simulationsarchitektur um eine homogene Co-Simulation, dargestellt unten rechts in der Abbildung. Den

letzten Fall erhält man durch Partitionierung sowohl von den Modellen als auch von den Solvern. Die Simulationsarchitektur Co-Simulation ist oben rechts dargestellt.

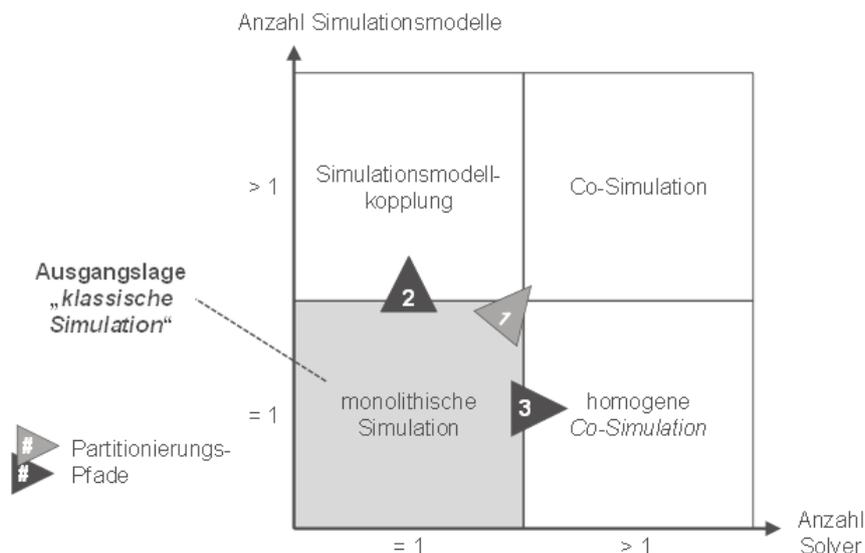


Abbildung 20: Einordnung und Abgrenzung der Co-Sim. bezüglich weiterer Simulationsarchitekturen [GMA20].

Nach der Einordnung und Abgrenzung der Co-Simulation in Bezug auf die Simulationsarchitektur sollen nun die Bestandteile einer Co-Simulation im Detail betrachtet werden. Hierzu wird das in [GV19] vorgestellte Merkmals-Diagramm (MD) für das Co-Simulations-Konzept herangezogen, welches die grundlegenden Co-Simulations-Begrifflichkeiten einführt und beschreibt. Das MD für das Co-Simulations-Konzept ist in Abbildung 21 dargestellt. Der Aufbau des MD ist an die Vorgaben von [KCH+90] angelehnt. Die Merkmale des Diagramms sind in einer Baumstruktur angeordnet. Obligatorische Merkmale werden grafisch durch einen kleinen Kreis direkt über dem Merkmalsnamen gekennzeichnet, wie beispielsweise bei dem Merkmal *Scenario* in Abbildung 21. Alternative Merkmale werden als Submerkmale desselben übergeordneten Basis-Merkmals dargestellt, wobei ein Bogen durch die Verbindungslinien aller Optionen gezeichnet wird, beispielsweise bei den untergeordneten Merkmalen *Fixed* und *Adaptive* vom Basis-Merkmal *Communication Step* in Abbildung 21. Der Bogen durch die Verbindungslinien kann in zwei Varianten auftreten. Ist der Bogen schwarz ausgefüllt, können ein oder mehrere der Submerkmale vorhanden sein. Hat der Bogen keine Füllfarbe handelt es sich um eine exklusive Oder-Verbindung. Beispielsweise kann die Schrittweite (Communication-Step) entweder fix oder adaptiv gewählt werden. Die Verbindungslinie zwischen einem untergeordneten Merkmal und einem übergeordneten Merkmal zeigt an, dass ein untergeordnetes Merkmal das Vorhandensein des übergeordneten Merkmals voraussetzt. Die linke Seite des MD in Abbildung 21 ist dem Bereich Komposition einer Co-Simulation zugeordnet, welcher in Abschnitt 4.2.2.2 behandelt wird. In Abschnitt 4.2.2.3 wird ein Überblick über die für die Komposition notwendigen Schnittstellenstandards und Tools gegeben. Die rechte Seite von Abbildung 21 beinhaltet Laufzeitaspekte einer Co-Simulation, welche in Abschnitt 4.2.2.5 aufgegriffen werden. Die für diese Arbeit relevanten Merkmale des Diagramms sind nummeriert und in den entsprechenden Abschnitten durch hochgestellte Zahlen referenziert. Betrachtet wird nun im nächsten Abschnitt zunächst die Co-Simulationskomposition.

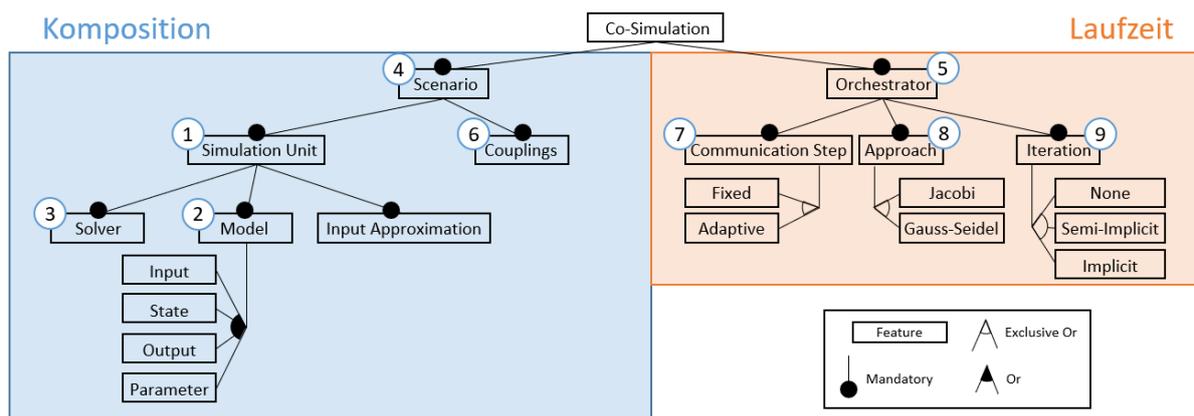


Abbildung 21: Merkmals-Diagramm nach [GV19] für das Co-Simulationskonzept.

4.2.2.2 Komposition einer Co-Simulation

Eine Co-Simulation besteht nach [PVL+17] aus gekoppelten Simulatoren, auch Simulation-Units¹ (kurz *Units*) genannt [SGE+19], welche miteinander kooperieren. Jede Unit beinhaltet ein Subsystemmodell² sowie seinen eigenen domänenspezifischen Solver³. Diesen Solver nutzt die Unit zur Lösung ihres Modells. Dieser Prozess läuft in jeder Unit parallel und zeitweise unabhängig ab. Um eine Co-Simulation auszuführen, benötigt man nach [MKB+11] neben einem Co-Simulations-Szenario⁴ einen Orchestrierungs-Algorithmus⁵ (siehe Abschnitt 4.2.2.5). Das Szenario verweist auf die Units und beschreibt, wie die Ein- und Ausgänge⁶ ihrer jeweiligen Modelle miteinander in Beziehung stehen. Die Units werden gekoppelt, indem die Modelle unter Verwendung ihrer Eingangs- und Ausgangsvariablen dynamisch miteinander verbunden werden, so dass der Ausgang einer Unit zum Eingang einer oder mehrerer anderen Units wird und umgekehrt. Grundlage für den Aufbau und Einsatz einer Co-Simulation ist das Vorhandensein von modularen Simulationsmodellen. Eine in sich geschlossene und mit klaren Schnittstellen versehene Beschreibung eines Modells ist Voraussetzung für eine Weiter- und Wiederverwendung von Modellen.

4.2.2.3 Standardisierte Schnittstellen und Werkzeuge für die Co-Simulation

Im Rahmen der in [SGE+19] durchgeführten Studie wurden über 50 Experten aus verschiedenen Bereichen von Industrie und Wissenschaft: (u. a. Energiesektor, Software-Entwicklung, Automotiv) befragt. Neben aktuellen Herausforderungen und dem Forschungsbedarf wurden vielversprechende Co-Simulations-Standards und Softwaretools untersucht. Dabei wird in drei verschiedene Co-Simulations-Paradigmen unterteilt. Der Unterschied zwischen „zeitkontinuierlich“, „diskret“ (ereignisbasiert) und „hybrid“ bezieht sich auf die Kommunikationsstrategie zwischen den einzelnen Units. Als von den Experten anerkannte und / oder selbstgenutzte Standards werden in [SGE+19] das *Functional Mock-Up Interface* (FMI) [BOA+11], *High Level Architecture* (HLA) [IEEE 1516-2010], *Simulink S-Functions*, *Discrete Event System Specification* (DEVS), sowie *SystemC* [MKB+11] genannt. Ergänzend zu diesen Standards sind agenten-[JSW18] und OPC UA-basierte [HGU+16] Ansätze zu erwähnen. Zudem wird in [SVT+20] im Bereich der virtuellen Inbetriebnahme ein Austausch von Simulationsmodellen zwischen Komponentenhersteller und Maschinenbauer in Form von *TwinStore Packages* (TSP) realisiert. Darüber hinaus gibt es Bestrebungen, die Standards FMI und HLA zu kombinieren [FG19].

Ein Auszug der Studienergebnisse aus [SGE+19] ist in Abbildung 22 dargestellt. Die Abbildung ist zweigeteilt, im oberen Teil finden sich drei Kreisdiagramme zur Fragestellung, welche Standards für Co-Simulation aus Expertensicht allgemein anerkannt sind. Hierbei wird noch spaltenweise zwischen den drei Co-Simulations-Paradigmen „zeitkontinuierlich“, „diskret“ und „hybrid“ differenziert. Im unteren Teil ist dargestellt, welche Standards die Experten selbst für welche Co-Simulations-Paradigmen nutzen. Die Einfärbung der Kreissektoren ist in der Legende über dem jeweiligen Diagramm aufgeschlüsselt. Bei der Expertenwahl ist sowohl hinsichtlich der allgemeinen Anerkennung als auch dem Selbstnutzen, insbesondere bei zeitkontinuierlichen und hybriden Co-Simulationen, eine klare Tendenz hin zum FMI-Standard auszumachen. Hier liegt der FMI-Anteil für beide Fragestellungen zwischen 90 und 94 Prozent. Auch bei der diskreten Co-Simulation überwiegt der FMI-Standard, wenn auch nicht so eindeutig. Hier liegt der Anteil bei der Anerkennung des Standards bei 39 Prozent und beim Selbstnutzen bei 68 Prozent, jeweils auf Platz eins vor der HLA.

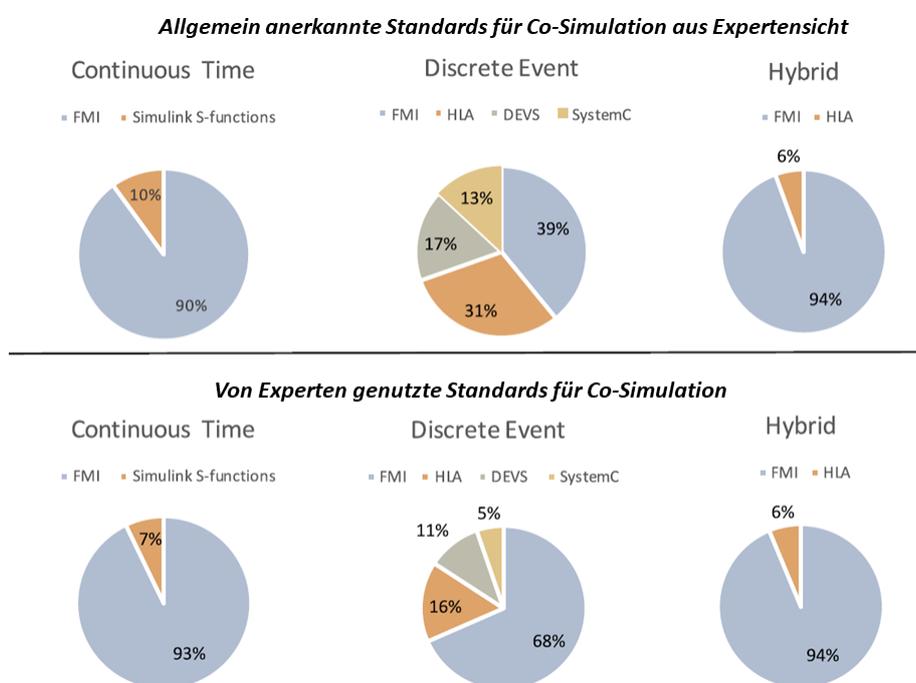


Abbildung 22: Anerkannte und selbstgenutzte Standards aus Expertensicht für verschiedene Co-Simulationsparadigmen [SGE+19].

Die Autoren von [SGE+19] betonen, dass sich aus ihrer empirischen Untersuchung im Rahmen der Studie, bedingt durch Stichprobengröße und mögliche Verzerrungen bei der Expertenwahl, keine allgemeingültigen Aussagen darüber ableiten lassen, welche Standards in Industrie und Wissenschaft weit verbreitet sind. Die Entwicklung von FMI hin zum de-facto Standard für Modellaustausch und Co-Simulation wird jedoch auch in weiteren Publikationen (u. a. [DPH17, SCS+19, PBM20]) beschrieben. Nach Aussage der offiziellen Webseite des FMI-Standards [FUN22[®]] besitzt dieser mit über 150 unterstützten Modellierungs- und Simulationsprogrammen eine signifikante Verbreitung und Akzeptanz in Industrie und Wissenschaft. Eine Übersicht der unterstützten Tools wird auf der offiziellen Webseite bereitgestellt und gepflegt. Der FMI-Standard eignet sich zudem für eine automatische Modelladaption im Rahmen von Anforderung A2, da die MSM bereits in standardisierter Form vorliegen. Ein Optimierungsalgorithmus kann auf die FMU-

Architektur zugeschnitten werden. Somit entfällt eine manuelle Aufbereitung der MSM vor einer Adaption. Hierauf wird in Kapitel 5 im Detail eingegangen. Im Rahmen dieser Arbeit fällt daher die Entscheidung auf den FMI-Standard als Realisierung der Gesamtsystems simulation als Co-Simulation. Der Standard soll im nächsten Abschnitt im Detail vorgestellt werden.

4.2.2.4 Der FMI-Standard für Co-Simulation

Beim FMI-Standard handelt es sich um einen werkzeug- und herstellerunabhängigen Standard für die Co-Simulation dynamischer Modelle, welcher im Rahmen des MODELISAR Projektes entstanden ist. Der FMI-Standard definiert eine Schnittstelle, die von ausführbaren Dateien implementiert wird. Diese Dateien werden als *Functional Mock-Up Units* (FMU) bezeichnet. [MM21] Unterschieden wird zwischen FMUs für Modellaustausch und Co-Simulation, wobei der zweite Typ neben dem Modell auch den nötigen Solver beinhaltet und im Rahmen dieser Arbeit betrachtet wird. Grund für diese Eingrenzung ist die Annahme, dass die FMUs, wie die Module der Produktionsanlage, von unterschiedlichen Zulieferern stammen (vgl. Abschnitt 2.2.1). Entsprechend benötigt der Integrator aller Submodelle die jeweiligen Solver. Bei den FMUs handelt es sich um Black-Box-Container mit definierten Schnittstellen [BOA+11]. Diese ermöglichen einen effizienten Austausch bzw. die Weitergabe von Simulationsmodellen bei gleichzeitigem Schutz von geistigem Eigentum. FMUs werden in Form von ZIP-Ordnern mit der Dateiendung „.fmu“ zur Verfügung gestellt. Die Ordner müssen mindestens folgende zwei Komponenten enthalten. Zum einen muss die FMI-Beschreibungsdatei (engl. *ModelDescriptionFile*) im XML-Format vorhanden sein. Die Beschreibungsdatei definiert den Aufbau und Inhalt einer XML-Datei, die von einer Modellierungsumgebung erzeugt wird. Diese XML-Datei enthält die Definition aller Variablen der FMU in einer standardisierten Weise. Dies ermöglicht es einem Softwaretool, die Variablendefinitionen in der Programmiersprache der Simulationsumgebung zu speichern und darauf zuzugreifen, z. B. in C++, C#, Java oder Python. Neben der Beschreibungsdatei befinden sich noch der C-Quellcode im ZIP-Ordner, einschließlich aller benötigten Laufzeitbibliotheken, die im Modell verwendet werden, und/oder Binärdateien für einen oder mehrere Zielrechner, wie z. B. Windows Dynamic Link Libraries (.dll) oder Linux Shared Object Libraries (.so). Die letztgenannte Lösung wird insbesondere dann verwendet, wenn der FMU-Anbieter den Quellcode verbergen möchte, um das enthaltene Know-how zu schützen. [MM21]

Ziel von FMI für Co-Simulation ist die Lösung von zeitabhängigen gekoppelten Systemen zu berechnen, die aus zeitlich kontinuierlichen oder zeitdiskreten Teilsystemen bestehen. In einer Blockdarstellung des gekoppelten Systems werden die Subsysteme durch Blöcke mit (internen) Zustandsvariablen $x(t)$ dargestellt, die mit anderen Teilsystemen (Blöcken) des gekoppelten Problems durch Subsystemeingänge $u(t)$ und Subsystemausgänge $y(t)$ verbunden sind [MM21]. Der schematische Aufbau einer FMU, entsprechend der FMI-Spezifikation Version 2.0.3 [MM21], ist in Abbildung 23 dargestellt. Hierbei handelt es sich um die vereinfachte Darstellung, die sowohl für Modellaustausch- als auch für Co-Simulations-FMUs zutrifft. Informationen, die von der Umgebung, beispielsweise anderen FMUs und dem Orchestrings-Algorithmus, zur Verfügung gestellt werden, sind durch rote Pfeile gekennzeichnet. Informationen, die die FMU nach der Simulation bereitstellt, sind durch blaue Pfeile dargestellt. Konkret benötigt die FMU die Eingangsvariablen $u(t)$ sowie vom Orchestrings-Algorithmus die Startzeit t_0 , einen Satz an Modellparametern p und die Startwerte der von außen zugänglichen Zustandsgrößen $v(t_0)$ [MM21]. Modellparameter sind während der Simulationsinitialisierung, innerhalb der vom Modellersteller vorgegebenen Grenzen, frei wählbar und während des Ablaufs der Simulation fix. Über diese lassen sich Eigenschaften

des modellierten physikalischen Systems wie beispielsweise die geometrische Ausprägung oder thermische Eigenschaften des Werkstoffes variieren. Auf Modellparameter wird in Kapitel 5 im Detail eingegangen. Mit diesen Informationen ist der Solver der FMU in der Lage die Simulation für den nächsten Zeitschritt auszuführen. Der Umgebung werden im Anschluss die Ausgangsvariablen $y(t)$ sowie die von außen zugänglichen Zustandsgrößen $v(t)$ bereitgestellt.

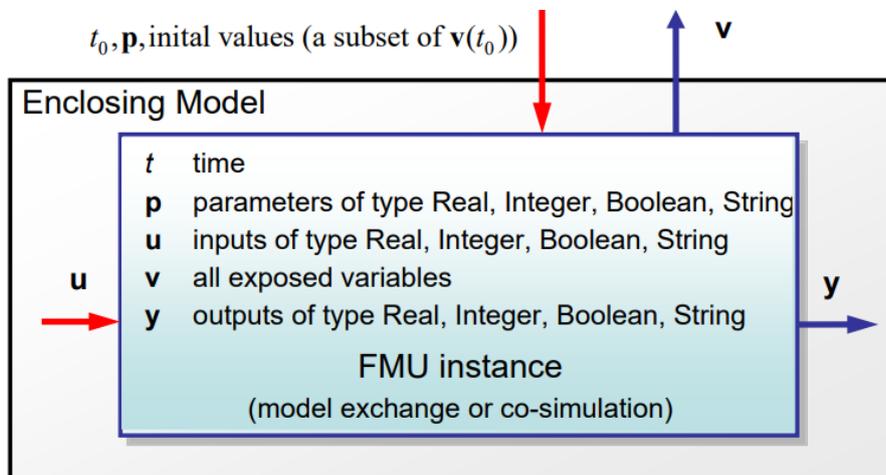


Abbildung 23: Schematische Darstellung einer FMU nach [MM21] mit Datenflüssen von und hin zur Umgebung.

Mit der Wahl des FMI-Standards zur Realisierung der Co-Simulation werden die jeweiligen MSM entsprechend in Form von FMUs benötigt. Nach der Darstellung des grundlegenden Aufbaus der Units und deren Komposition sollen im folgenden Abschnitt noch abschließend die Laufzeitaspekte einer Co-Simulation betrachtet werden.

4.2.2.5 Laufzeitaspekte einer Co-Simulation

Variablen austausch, Zeitsynchronisation und Ausführungs koordinierung werden zur Laufzeit durch einen Orchestrings-Algorithmus⁵ (OA) durchgeführt. Der OA initialisiert alle Simulationseinheiten mit den entsprechenden Werten, setzt/ erhält ihre Ein-/ Ausgänge und koordiniert ihren Verlauf über die simulierte Zeit. Sobald die Kommunikationsverbindungen zwischen jeder Unit hergestellt sind und jede Unit mit Modellparametern und Anfangsbedingungen initialisiert worden ist, wird über den OA jede Unit von einem Kommunikationspunkt t_i zum nächsten geführt. Die einzelnen Units erleben jeden Zeitschritt⁷ zwischen den Kommunikationspunkten t_i und t_{i+H} als eine Art autarke Simulation, d. h. sie erhalten zu Beginn Eingangsvariablen und eine Simulationsdauer und starten anschließend eigenständig mit der Simulation. Zwischen zwei Kommunikationspunkten führen die Units eine durch den OA vorgegebene Anzahl an Simulationsschritten durch. Diese Anzahl ist entweder fix vorgegeben oder adaptiv gesteuert. An jedem Kommunikationspunkt tauschen die Units Koppelsignale zwischen den Modellen aus. In Abbildung 24 sind verschiedene Arten von Orchestringsalgorithmen⁸ (aufgetragen über die Spalten) und Iterationsstrategien⁹ (aufgetragen über die Zeilen) anhand von Zeitdiagrammen für zwei Units S_1 und S_2 zusammengefasst. Der Datenaustausch zwischen den zwei Units wird über Pfeile mit gepunkteten Verbindungslinien dargestellt, Simulationsschritte über Pfeile mit durchgezogenen Verbindungslinien. Beim Gauss-Seidel-Ansatz in der linken Spalte fordert der OA jede Unit auf, das nächste Intervall zu berechnen, bevor er sie auffordert, Outputs $y(t_i)$ zu erzeugen. Diese Outputs werden dann an die nächste Unit als Input $u(t_i)$ übergeben, bevor diese aufgefordert wird, das nächste Intervall zu berechnen. Die Simulationen in den

Units laufen sequenziell ab. Beim Jacobi-Ansatz in der rechten Spalte fordert der OA alle Units auf, das Intervall parallel zu berechnen, wobei er ihre Eingaben am Ende des Co-Simulationsschritts setzt. Schließlich kann der OA den Co-Simulationsschritt unter Verwendung verbesserter Eingabeschätzungen, die aus den letzten Ausgaben berechnet werden, erneut durchführen. In Abbildung 24 wird diese Rückwärtsrechnung (engl. Rollback), konkret in der unteren Zeile links der Schritt 11 bzw. rechts der Schritt 9, durch gestrichelte Pfeile dargestellt. Dieser Vorgang kann so lange wiederholt werden, bis keine Verbesserung der Eingaben mehr vorliegt (vollständig implizite Iteration) oder bis eine bestimmte Anzahl von Iterationen durchgeführt wurde (semi-implizite Iteration). [GV19]

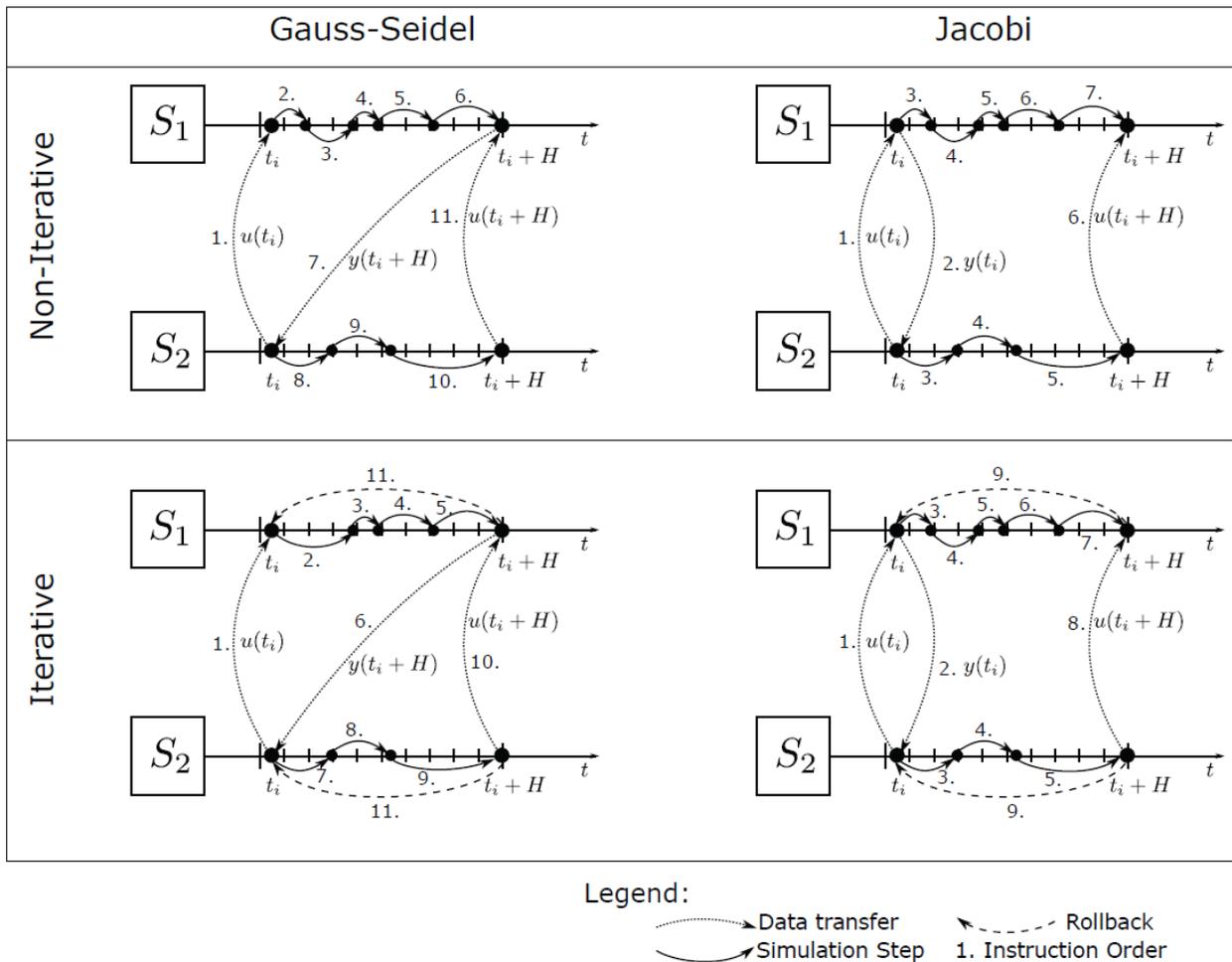


Abbildung 24: Übersicht der Zeitdiagramme gängiger Orchestrierungsalgorithmen für zwei Units [GV19].

Nachdem der Stand von Wissenschaft und Technik hinsichtlich der zur Verfügung stehenden Beschreibungsmittel von Produktionsanlagen sowie die Modellierungs- und Simulationsmöglichkeiten betrachtet wurden, wird im Folgenden das Teilkonzept für die (teil-)automatische Modellgenerierung und -rekonfiguration im Detail vorgestellt.

4.3 Konzept zur automatischen Generierung von modulbasierten Systemsimulationen

In diesem Abschnitt wird das entwickelte Teilkonzept zur automatischen Komposition und Konfiguration von Co-Simulationen mit Fokus auf die Inbetriebnahmephase vorgestellt. Hierbei handelt es sich um eine

detaillierte Darstellung der in Abbildung 19 vorgestellten Gesamtkonzeptübersicht. Es gelten die in Abschnitt 4.1 formulierten Anforderungen A1 und A2 sowie die genannten Voraussetzungen. Gemäß den Voraussetzungen existiert ein TM der Produktionsanlage in AML, links in Abbildung 25 dargestellt, und die MSM aller Module stehen in Form von FMUs zur Verfügung. Diese sind als externe Modell-Bibliothek oben in der Abbildung angedeutet. Da die Inbetriebnahmephase der Anlage betrachtet wird, ist diese noch nicht gänzlich betriebsbereit. Dies wird links unten in Abbildung 25 durch einen gestrichelten Rahmen angedeutet. Der Prozess der automatischen Modellgenerierung und -konfiguration beginnt damit, dass das AS (grün hervorgehoben) manuell gestartet wird. Als initialer Trigger wird das TM in das AS geladen. Die erste Aufgabe des AS ist es, die in der Anlage vorhandenen Module auf Grundlage des TM zu identifizieren und die benötigten MSM aus einer oder mehreren verteilten FMU-Bibliotheken zu laden. Dieser referenzbasierte Ansatz wird als zukunftsfähig betrachtet, da er sowohl mit den Umsetzungen der Industrie-4.0-Verwaltungsschale [BMW20a] als auch mit aktuellen cloudbasierten Simulationsarchitekturen [JKS+22] kompatibel ist. Anhand der im TM beschriebenen Schnittstellen sollen die FMUs, analog zum realen System, gekoppelt werden.

Die nötigen Co-Simulationsfunktionalitäten werden, wie in Abschnitt 4.2.2.3 dargelegt, in einer Vielzahl von Werkzeugen und Bibliotheken, wie Simulink, Dymola oder PyFMI, abgebildet. Für die in Abschnitt 4.4 vorgestellte prototypische Konzeptumsetzung fällt die Wahl auf den OpenModelicaSimulator (OMS) [OPE22[®]]. Das Open-Source-Werkzeug zeichnet sich unter anderem durch eine umfangreiche, nicht-proprietäre Programmierschnittstelle [OBT+19] aus, die sich auf die aktuell zum de-facto-Standard etablierenden Python-Skripte stützt [WES22[®]]. Das AS generiert auf Basis des TM, unten rechts in Abbildung 25, ein Skript, parametriert und startet die Simulation.

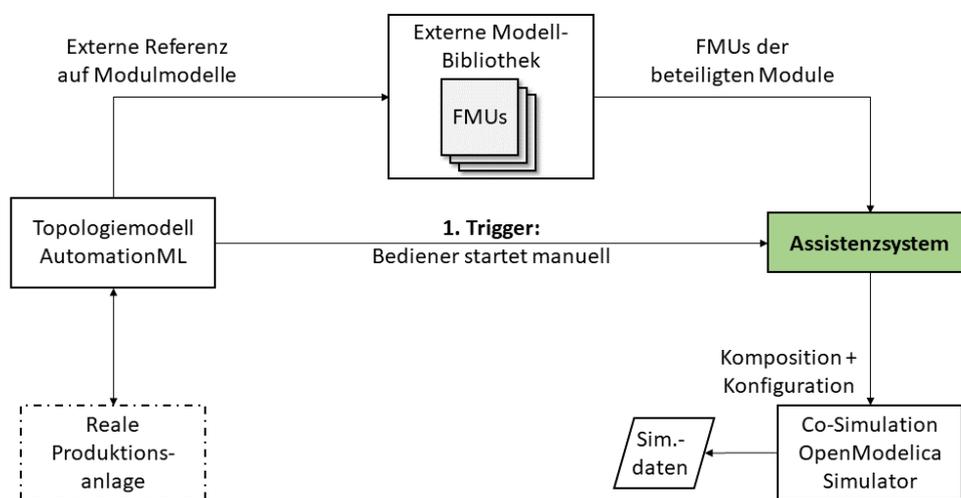


Abbildung 25: Automatische Komposition und Konfigurationen von Co-Simulationen zur Inbetriebnahmephase.

Als Nachweis der generellen Machbarkeit und dafür, dass das erarbeitete Konzept in der Praxis umgesetzt werden kann, wird im folgenden Abschnitt eine prototypische Implementierung des AS für die automatische Modellgenerierung und -konfiguration von modularen Produktionsanlagen vorgestellt.

4.4 Prototypische Implementierung des Assistenzsystems

Die Basis-Funktions-Ebene des AS stellt die automatische Komposition und Konfiguration der Co-Simulation, basierend auf dem TM, dar. Diese initiale Modellgenerierung erfolgt auf einen manuellen Trigger des Integrators. Auf dem initial generierten Simulationsmodell bauen zukünftig alle weiteren Funktionalitäten des AS auf. Als Informationsquelle steht dem AS im ersten Schritt ausschließlich das vorgegebene TM zur Verfügung. Es ist daher notwendig, zunächst die Struktur sowie die festgelegten Konventionen des TMs zu beschreiben.

4.4.1 Struktur und Konventionen des Topologiemodells

Für das TM wird, wie in Abschnitt 4.2.1 beschrieben, AML verwendet. Der Standard wird im Rahmen der Normserie IEC 62714 [IEC 62714-1] entwickelt und gepflegt. AML verfolgt einen objektorientierten Ansatz und ermöglicht es, Anlagenkomponenten als Datenobjekte zu modellieren. Schwerpunkt beim Datenaustausch im Engineering ist der Austausch von objektorientierten Datenstrukturen, Geometrie, Kinematik und Programmierung [AC18]. Anhand vordefinierter Schlüsselwörter ist es möglich, alle relevanten Informationen aus dem TM zu extrahieren. Bei der Modellierung mit AML wird in erster Linie zwischen Objekten (hier insbesondere Anlagenmodulen), dargestellt als *InternalElements* (IE), und deren Schnittstellen, sogenannten *ExternalInterfaces* (EI), unterschieden. Sofern für ein Modul ein MSM zur Verfügung steht, kann dieses durch ein EI referenziert werden. Für diese Klasse von Interfaces wird die Bezeichnung (*Tag*) *ModelReference* eingeführt. Anhand dieses Tags ist das AS in der Lage, eine Komponente als Bestandteil der Co-Simulation zu identifizieren. Wie bereits beschrieben, kann die hier gespeicherte Adresse ein lokales bzw. netzwerkbasierendes Verzeichnis oder eine URL zu einer externen Zulieferer-Webseite bzw. zu einem Cloud-Speicher referenzieren. Nach der Identifizierung aller potenziellen Komponenten werden im nächsten Schritt deren FMU analysiert. Grundbestandteil einer jeden FMU ist, neben dem eigentlichen Simulationsmodell, eine sogenannte *ModelDescription*-Datei. Hierin werden u. a. die Schnittstellen der FMU definiert. Relevant sind in erster Linie die entsprechenden Bezeichner der Schnittstellen und deren Kausalitäten, d. h. die Unterscheidung in Ein- und Ausgänge. Im Anschluss müssen die Informationen aus den jeweiligen ModelDescriptions mit den Informationen aus dem TM verknüpft werden. Hierfür werden die AML-Elemente *SignalInterface* und *InternalLink* auf Basis von Identifikatoren (IDs) verwendet.

Die Zuordnung über ID-Paare aus einem TM ist beispielhaft in Abbildung 26 dargestellt. Auf der linken Seite ist ein Auszug eines exemplarischen TMs visualisiert. Dieses ist in Anhang A im Detail sowohl als Bild als auch als XML-Datei dargestellt. Ein *InternalLink* wird durch eine blaue Verbindungslinie zwischen zwei EI dargestellt. Rechts in Abbildung 26 ist die entsprechende XML-Darstellung dieser EI abgebildet. Die relevanten IDs sind farblich hervorgehoben. Zudem sind Attribute wie Kausalität und Datentyp hinterlegt. Diese Informationen nutzt das AS, um daraus ein Skript zu generieren, anhand dessen das Co-Simulationswerkzeug sukzessive Aufbau, Initialisierung und Simulation durchführt. Im unteren Teil von Abbildung 26 ist exemplarisch eine aus dem TM abgeleitete OMS-Skriptzeile dargestellt. Die Skriptzeile beinhaltet im Wesentlichen den Projektnamen, die beteiligten Module für die betrachtete Kopplung sowie den zu koppelnden Ein- und Ausgang.

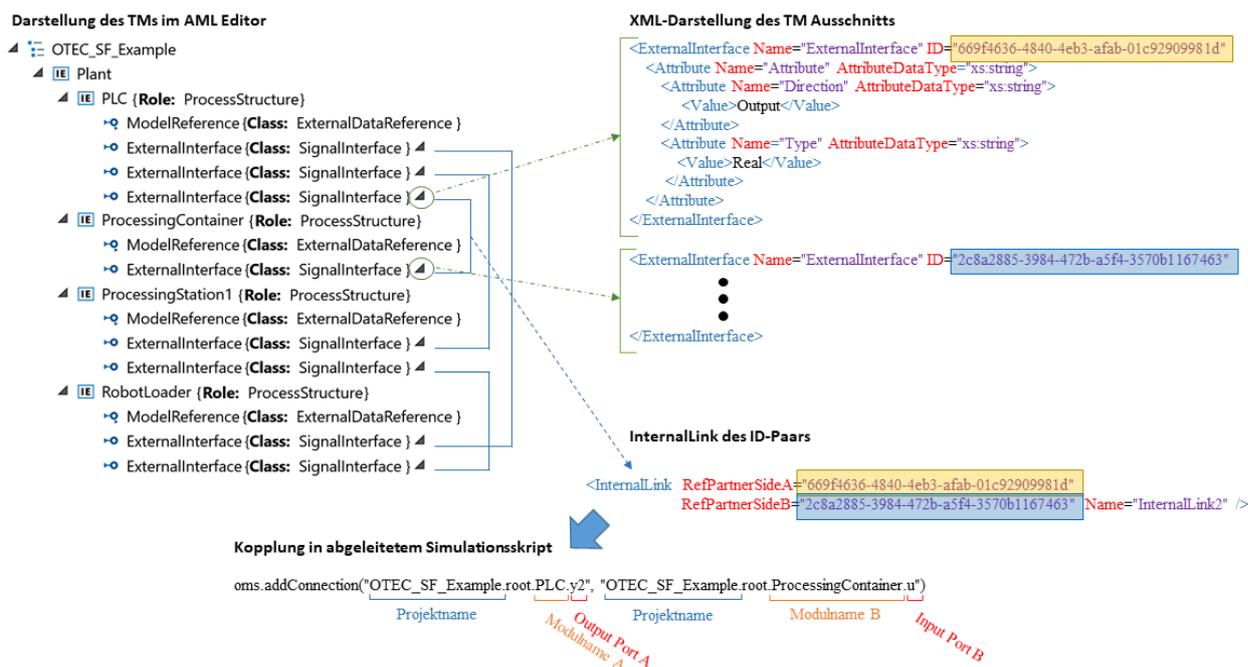


Abbildung 26: Aus TM abgeleitete ID-Paare für Simulationsskript.

4.4.2 Aus Topologiemodell abgeleitetes Simulationsskript

Das vollständige Python-Skript, welches das AS aus dem exemplarischen TM in Abbildung 26 generiert, ist in Listing 1 abgebildet. In der ersten Zeile des Skripts wird zunächst die OMSimulator-Bibliothek importiert. In Zeile 2 wird das OS-Modul importiert, welches genutzt wird, um mit relativen Pfaden zu arbeiten. So wird sichergestellt, dass das AS auch auf anderen Plattformen genutzt werden kann. In den Zeilen 5 und 6 wird das Arbeitsverzeichnis angelegt. In Zeile 7 wird das Gesamtmodell initialisiert. Diesem wird der Name des TM als String übergeben. Um die Lesbarkeit des Skripts zu erhöhen, wird in Zeile 7 der Modellname „OTEC_SF_Example“ mit „OTEC“ abgekürzt. In Zeile 8 werden die Laufzeitaspekte der Co-Simulation, wie der Umgang mit Toleranzen und die Schrittweite des OA, durch das Schlüsselwort „system_wc“ definiert. Hierbei handelt es sich um ein schwach gekoppeltes (engl. *weakly coupled*, kurz wc) System, welches die Standardeinstellung darstellt. In den Zeilen 11 bis 14 werden die MSM dem System durch die Funktion „oms.addSubModel()“ hinzugefügt. Der Funktion wird als erster Parameter das System und als zweiter Parameter der Pfad zum Submodell, der FMU des jeweiligen Moduls, übergeben. Die Pfade zu den Modellquellen entnimmt das AS dem TM, konkret den EI „ModelReference“ der Klasse „ExternalDataReference“. In Abbildung 26 stehen diese EI jeweils direkt unter den IE der Module. Ebenfalls zur Verbesserung der Lesbarkeit des Skripts werden die Pfade zu den Modellquellen gekürzt. Im Rahmen der prototypischen Implementierung wird mit Modellen, die auf einer lokalen Festplatte gespeichert sind, gearbeitet. Der Pfad zu einem Modell mit der Bezeichnung „model_xy“ würde mit einem String der Form „C:/Users/.../Model_Library/model_xy.fmu“ referenziert werden. Diese Strings werden in Listing 1 auf den Modellnamen und den Dateityp, entsprechend dem vorherigen Beispiel „model_xy.fmu“, gekürzt. Bei der Steuerung, welche die Bezeichnung „PLC“ trägt, handelt es sich nicht um ein Fertigungsmodul der Produktionsanlage. In diesem Beispiel dient sie lediglich als Signalquelle für die weiteren MSM (vgl. Abbildung 27). Nach der Integration der Submodelle werden diese, entsprechend der beschriebenen ID-Paarung der Internallinks, in den Zeilen 15 bis 18 gekoppelt. In den Zeilen 20 bis 22 werden die Simulationseinstellungen aus der Bedieneingabe über die GUI übernommen. Es wird angegeben, in welcher Datei

die Simulationsergebnisse gespeichert werden. Der Name der Datei wird automatisch über den Zeitstempel generiert. Im Anschluss wird noch die Stoppzeit der der Simulation sowie die Schrittweite des OA angegeben. Nach Zeile 24 sind die Parametereinstellungen der MSM vorgesehen. In diesem Beispiel wird an den Grundparametern der MSM keine Änderung vorgenommen. Dies wird in Kapitel 5, der betriebsparallelen Adaption, relevant und nochmals aufgegriffen. Abschließend werden in den Zeilen 27 bis 31 die Funktionen aufgerufen, um die Simulation zu initialisieren, zu simulieren und abschließend zu stoppen und die Simulationsergebnisse abzuspeichern.

```
1 from OMSimulator.lib.OMSimulator import OMSimulator
2 import os
3 oms = OMSimulator()
4
5 parentDir=os.getcwd()
6 oms.setTempDirectory(parentDir+"/temp/")
7 oms.newModel("OTEC")
8 oms.addSystem("OTEC.root", oms.system_wc)
9
10 # Instantiate FMUs
11 oms.addSubModel("OTEC.root.ProcessingStation", "ProcessingStation.fmu")
12 oms.addSubModel("OTEC.root.ProcessingContainer", "ProcessingContainer.fmu")
13 oms.addSubModel("OTEC.root.RobotLoader", "RobotLoader.fmu")
14 oms.addSubModel("OTEC.root.PLC", "PLC.fmu")
15 oms.addConnection("OTEC.root.ProcessingStation.y", "OTEC.root.RobotLoader.u1")
16 oms.addConnection("OTEC.root.PLC.y", "OTEC.root.RobotLoader.u")
17 oms.addConnection("OTEC.root.PLC.y1", "OTEC.root.ProcessingStation.u")
18 oms.addConnection("OTEC.root.PLC.y2", "OTEC.root.ProcessingContainer.u")
19 # Simulation settings
20 oms.setResultFile("OTEC", "simResults_220130_1531_09.csv")
21 oms.setStopTime("OTEC", 3)
22 oms.setFixedStepSize("OTEC.root", 1.0E-4)
23
24 # Parameter settings
25
26 # Initialize and run simulation
27 oms.instantiate("OTEC")
28
29 oms.initialize("OTEC")
30 oms.simulate("OTEC")
31 oms.terminate("OTEC")
```

Listing 1: Abgeleitetes Python-Skript des TM aus Abbildung 26.

Abschließend wird noch aufgezeigt, wie der weitere Workflow des AS nach der Generierung des Skripts abläuft.

4.4.3 Automatische Ausführung eines Simulationsskriptes

Der in Abschnitt 4.3 benannte OpenModelicaSimulator ist ein Co-Simulationswerkzeug, das den FMI-2.0-Standard für Modellaustausch und Co-Simulation unterstützt und sich zudem durch eine umfangreiche Programmierschnittstelle auszeichnet [OBT+19]. Zum Zeitpunkt des Verfassens dieser Arbeit wurde bereits der FMI-3.0 Standard veröffentlicht, aber noch von keinem Modellierungs- und Simulationsprogramm unterstützt. Daher wird die neuere Version des Standards im Rahmen der prototypischen Implementierung nicht weiter betrachtet. Das durch das AS generierte Simulationsskript wird mit dem OMS automatisch ausgeführt, das Gesamtmodell instanziiert, die Simulation durchgeführt, und die Simulationsergebnisse werden strukturiert gespeichert. Das implementierte AS ist zudem in der Lage, diese gespeicherten Ergebnisse den einzelnen Modulen zuzuordnen und zu visualisieren. Um den Benutzer bei

der Analyse zu unterstützen, ist es zusätzlich möglich, im TM eine Grafik für das Anlagenmodul zu hinterlegen. Bezogen auf eine virtuelle Inbetriebnahme stellt die Visualisierung den letzten Schritt dar. Der Anlagenintegrator hat die Möglichkeit, die Simulationsergebnisse für jedes Modul separat zu betrachten und auf Plausibilität zu überprüfen. Die Visualisierung unterstützt eine erste Einschätzung darüber, ob die Komposition und Konfiguration richtig durchgeführt wurden. Der nächste Schritt wäre nach erfolgreicher Inbetriebnahme der Abgleich mit realen Messwerten und bei Bedarf die automatische Adaption des Modells parallel zum Betrieb. Dies wird im Anschluss in Kapitel 5 untersucht. Davor ist es jedoch notwendig, die prototypische Implementierung des AS und damit das Teilkonzept zu validieren. Dies erfolgt im nächsten Abschnitt.

4.5 Teilkonzeptvalidierung

In diesem Abschnitt wird das in Abschnitt 4.3 vorgestellte Teilkonzept zur automatischen Modellgenerierung und -konfiguration für modulare Produktionsanlagen validiert. Hierzu wird das im vorherigen Abschnitt prototypisch implementierte AS auf die beiden in Abschnitt 2.3 vorgestellten Produktionsanlagen angewandt. Das erste Validierungsszenario stellt die automatische Modellgenerierung und -konfiguration für die Gleitschleifanlage von OTEC, das zweite Szenario für die Galvanikanlage von Jentner dar.

4.5.1 Validierungsszenario 1: Modellgenerierung der OTEC Schleifanlage

Für die erste Teilkonzeptvalidierung wird die in Abbildung 27 dargestellte, aus drei Modulen bestehende Anlage herangezogen. Hierbei handelt es sich um die in Abschnitt 2.3.1 vorgestellte OTEC Gleitschleifanlage inklusive eines Roboterbeladers für die WS. Die WS werden durch eine Spannlanze (Modul 1) in den rotierenden Bearbeitungsbehälter (Modul 2) getaucht. Dieser Behälter ist mit Schleif- oder Poliermedia befüllt. Durch die Parameter Eintauchtiefe und -winkel, Behälterdrehzahl und Mediazusammensetzung kann der Schleif- oder Polierprozess entsprechend angepasst werden. Die Be- und Entladung der Spannlanzen mit WS erfolgt durch den Roboterbelader (Modul 3).

Zur besseren Erläuterung werden die Ein- und Ausgangssignale der Module als vereinfachte Graphen dargestellt und die Detaillierungsstufen der Modulmodelle minimiert. Jeweils eines der drei Ausgangssignale der Steuerung dient als Eingangssignal für eines der drei Module. Die Sinusschwingung beispielsweise dient als Eingang für das Modul 1 „Spannlanze“, die linear steigende Funktion dient als Eingang für Modul 2 „Behälter“. Die Sprungfunktion ist Ausgangssignal von Modul 1 und eines der zwei Eingangssignale von Modul 3 „Roboterbelader“. Es gilt die unterschiedlichen Skalierungen bei den dargestellten Plots zu beachten. Die Komposition und Konfiguration der Co-Simulation wurde, entsprechend dem vorgegebenen TM in Abbildung 26, korrekt vom AS durchgeführt. Folglich hat das AS das TM als Beschreibungsmittel korrekt interpretiert und die nötigen MSM- und Kopplungsinformationen daraus abgeleitet. Entsprechend erfüllt das AS die Anforderung A1 für dieses Validierungsszenario.

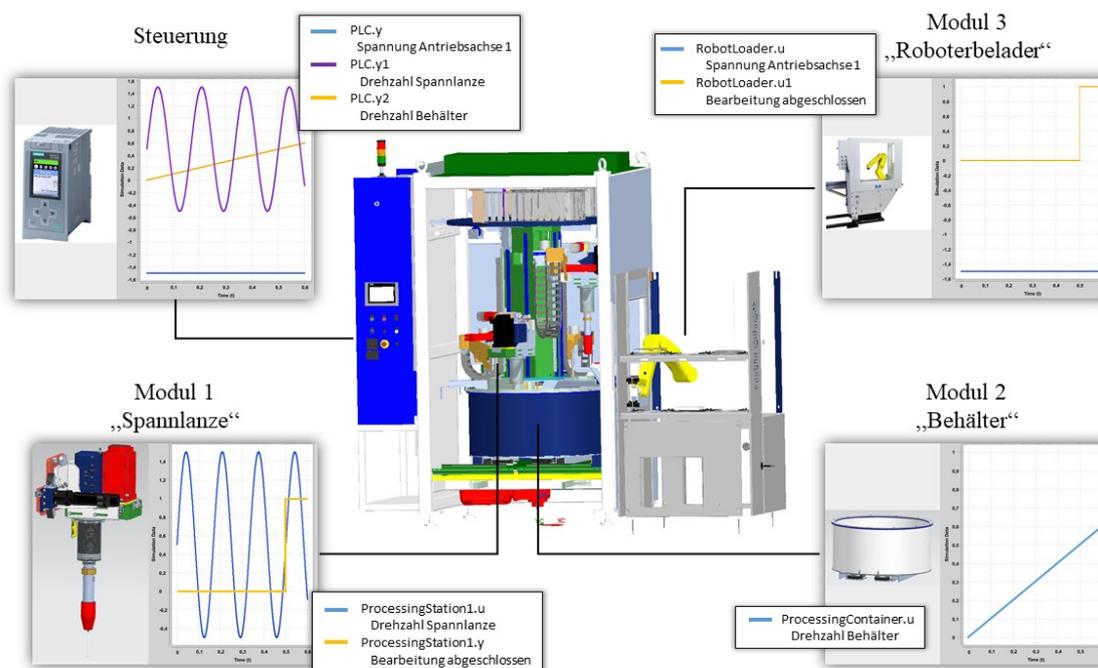


Abbildung 27: Gleitschleifanlage und von AS generierte Visualisierung der Co-Simulation.

Im nächsten Abschnitt wird das zweite Validierungsszenario betrachtet.

4.5.2 Validierungsszenario 2: Modellgenerierung der Jentner Galvanikanlage

Für das zweite Validierungsszenario wird die in Abbildung 28 dargestellte, aus vier Modulen bestehende Anlage herangezogen. Hierbei handelt es sich um die in Abschnitt 2.3.4 vorgestellte Kleingalvanik von Jentner. Das TM der Anlage, welches dem AS als Grundlage für die Co-Simulation dient, ist Anhang B zu entnehmen. Es wird ein vereinfachter Galvanisierungsprozess betrachtet. Mittels eines Werkstückträgers (Modul 1) werden WS nacheinander in ein Gold- (Modul 2), Silber- (Modul 3) und Rhodiumbad (Modul 4) abgesenkt und bearbeitet. Die Spülprozesse zur Reinigung der WS zwischen den Übergängen der Galvanisierungsbäder werden nicht betrachtet. Die MSM bestehen u. a. aus einem Zustandsautomat und Komponenten, die den elektrochemischen Zerfall sowie die aktuell vorherrschenden Konzentrationen der Inhaltsstoffe des jeweiligen Bads berechnen. In diesem Validierungsszenario wird lediglich untersucht, ob das AS in der Lage ist, die MSM korrekt zu koppeln. Hierfür ist es ausreichend, den Prozess anhand der Zustandsautomaten zu überprüfen. Die Stoffkonzentrationen der Elektrolyte werden daher in diesem Kapitel noch nicht betrachtet. Ebenfalls unbeachtet bleiben die Flüssigkeitsverschleppungen durch die WS (engl. *Drag-In* und *Drag-Out*). Auf beide Faktoren wird in Kapitel 5 im Detail eingegangen. Das auf den Zustandsautomat reduzierte Modelica-basierte MSM eines Elektrolytbades ist in Abbildung 29 dargestellt. Für dieses einfache Validierungsszenario wird für alle drei Bäder das gleiche MSM verwendet. Das MSM hat als Eingang ein boolesches Signal „*previousStepDone*“. Dieses wird *true*, wenn der vorherige

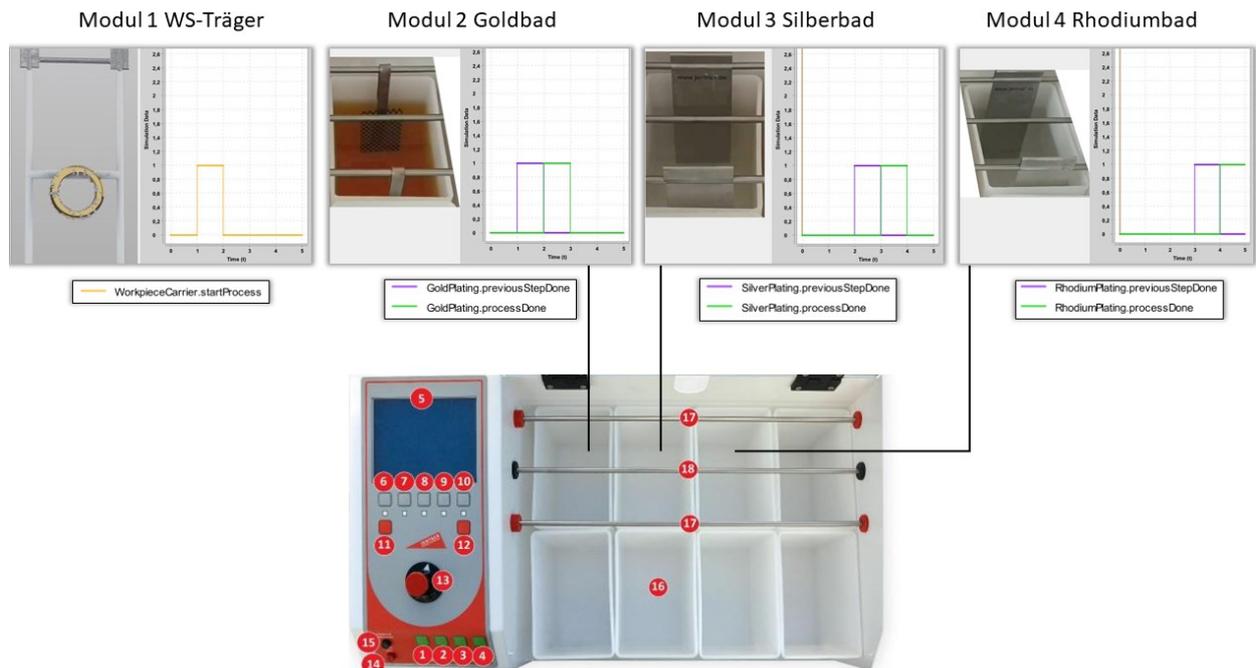


Abbildung 28: Galvanikanlage und von AS generierte Visualisierung der Co-Simulation.

Prozess abgeschlossen ist. Das erste Bad (Goldbad, Modul 2) bekommt dieses Signal vom Werkstückträger (Modul 1), sobald das WS auf den Träger befestigt und zum Galvanisieren bereit ist. In diesem Beispiel wird das WS in Sekunde 1 befestigt und ist nach einer weiteren Sekunde bereit für die Galvanisierung. Dies ist oben links in Abbildung 28 durch die gelbe Rechteck-Funktion dargestellt. Im Anschluss verweilt das WS für eine vorgegebene Zeit, in diesem Beispielszenario für jeweils eine Sekunde, im Bad und wird dann übergangslos in das nächste Bad eingetaucht. Dieser Ablauf ist für alle Bäder identisch. Nach dem jeweiligen Teilprozessende wird der boolesche Trigger „prozessDone“ als Signal an das nächste MSM übergeben. Der Abbildung 28 sind die Simulationsdaten der einzelnen MSM zu entnehmen. Es wird ersichtlich, dass das AS die Kopplung der Zustandsautomaten der einzelnen Bäder korrekt vorgenommen hat. Der violette Signalverlauf in Modul 2 bis 4 gibt jeweils an, in welchem Zustand sich der vorherige Prozess befindet und der grüne Signalverlauf gibt den Prozessverlauf des aktuellen Bades an. Das WS befindet sich von Sekunde 2 bis 3 der Simulation im Goldbad, von Sekunde 3 bis 4 im Silberbad und von Sekunde 4 bis 5 im Rhodiumbad. Nach 5 Sekunden ist die Simulation abgeschlossen und der Prozess planmäßig durchlaufen.

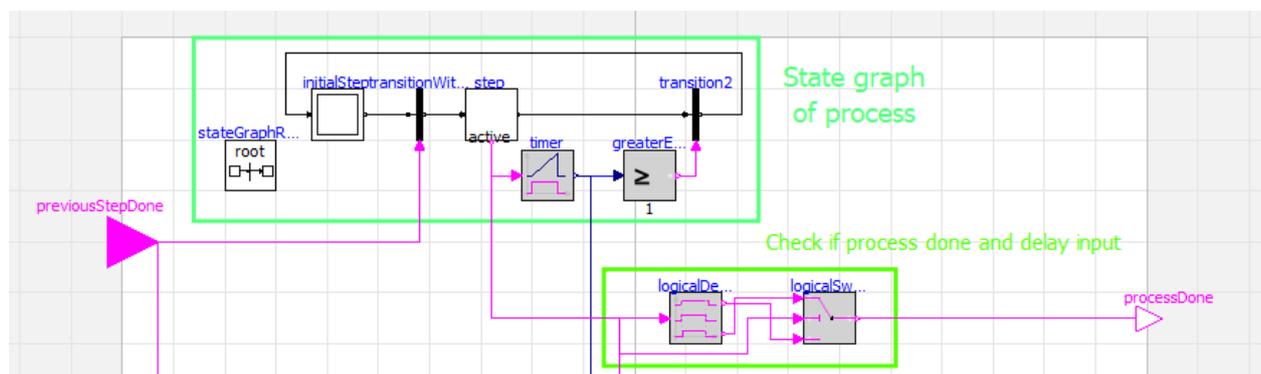


Abbildung 29: Ein auf den Zustandsautomat reduziertes MSM eines Galvanikbades.

Folglich hat das AS auch in diesem Validierungsfall das TM als Beschreibungsmittel korrekt interpretiert und die nötigen MSM- und Kopplungsinformationen daraus abgeleitet. Auch für dieses Szenario erfüllt das AS die Anforderung A1. Im folgenden Abschnitt soll abschließend ein Zwischenfazit für dieses Kapitel gezogen werden.

4.6 Zwischenfazit

In diesem Kapitel wurde die erste Lebenszyklusphase einer modularen Produktionsanlage, die Inbetriebnahmephase, betrachtet. Zunächst wurde festgelegt, welche der in Abschnitt 2.2.4 definierten Anforderungen an das AS im Rahmen dieses Kapitels erfüllt werden sollen. Anschließend wurde der aktuelle Stand von Wissenschaft und Technik zu den Themen Beschreibungsmittel und Co-Simulation diskutiert. Nach der Wahl eines geeigneten Beschreibungsmittels in Form eines TM in AML, des Co-Simulations-Standards FMI und des Co-Simulations-Tools OMS, wurde das Teilkonzept im Detail vorgestellt. Basierend darauf wurde das AS als Nachweis der generellen Machbarkeit prototypisch implementiert und anhand von zwei modularen Produktionsanlagen aus unterschiedlichen Bereichen der Fertigung validiert.

Hinsichtlich der vier in Abschnitt 1.3 formulierten Forschungsfragen ergibt sich folgende Einschätzung. Zusammenfassend erfüllt die erste Konzeptstufe des implementierten Prototyps die in Abschnitt 4.1 formulierten Anforderungen zur Komposition und Konfiguration einer Co-Simulation für die Phase der Inbetriebnahme. Das AS ist in der Lage, Modulmodelle zu laden, Initialwerte/ Anfangszustände zu setzen, die MSM, analog zur realen Systemkonfiguration, korrekt zu koppeln sowie die Simulation zu initiieren und auszuführen. Das beantwortet die erste Forschungsfrage für das Teilkonzept. Durch den Einsatz des FMI-Standards und das AML-basierte TM ist die Erweiterung des Einsatzbereichs des AS auf den kompletten Anlagenlebenszyklus möglich. Ausblickweise können so beispielsweise die MSM über die Parameter adaptiert und umbaubedingte Änderungen im TM dokumentiert werden. Die resultierende Co-Simulation kann beispielsweise auch für eine Software-/ oder Hardware-in-the-Loop-Simulation genutzt werden. In den folgenden Kapiteln werden die konzeptionellen Grundlagen der Umsetzung von Anforderung A2 und damit die Erweiterung des Einsatzbereichs auf den kompletten Anlagenlebenszyklus, beschrieben. Die standardisierten Schnittstellen der MSM stellen die einzige Voraussetzung entsprechend Forschungsfrage 2 dar. Das TM sowie die Bedieneingaben für die Modell- und Simulationsparameter stellen die benötigten Informationen entsprechend Forschungsfrage 3 dar. Vor der Beantwortung der letzten Forschungsfrage hinsichtlich des Automatisierungsgrades der Lösung, bietet es sich an, die Ausprägung des AS genau zu betrachten. Entsprechend der in Abschnitt 2.2.2 vorgenommenen Klassifizierung von AS ist dieses für die betrachtete Lebenszyklusphase als Ausführungsassistenzsystem (AAS) einzustufen. Der Bediener lädt das TM in das AS und gibt Modell- sowie Simulationsparameter vor. Die Kopplung und Konfiguration des Gesamtmodells sowie die Initialisierung, Ausführung und Visualisierung der Simulation erfolgt vollautomatisch. Bezüglich der in Abschnitt 2.2.5 definierten Automatisierungslevel kann das AS für diese Lebenszyklusphase in dem Level 5 „Vollautomatisierung“ eingestuft werden. Das AS extrahiert alle notwendigen Informationen selbstständig und führt alle M&S-Tätigkeiten vollautomatisiert aus.

Eine Herausforderung stellen zum aktuellen Zeitpunkt noch Modelle dar, die sich über mehrere Module erstrecken. Vorstellbar sind z. B. Modelle, die die Reibung zwischen Kontaktflächen unterschiedlicher Module berechnen. Ein potenzieller Lösungsansatz für den Umgang mit modulübergreifenden Modellen ist

durch geschickte Kombination hierarchischer Modellierung mit semantischer Adaption [GMD+19], Modellverbindern [CHF14] und überlagerten AML-Objekten vorstellbar. Dieser Spezialfall ist außerhalb des Fokus dieser Arbeit und stellt einen Forschungsbedarf für zukünftige Arbeiten dar. Darüber hinaus ist die Erkenntnis von Reinhardt et al. [RWP19], welche bereits in Abschnitt 3.1.1 vorgestellt wurde, zu überdenken. Die Autoren kamen in ihrer Studie zu dem Schluss, dass alle untersuchten Veröffentlichungen zum Thema automatische Modellgenerierung im Bereich der diskreten Fertigung auf einem de-facto Standard wie AutomationML, CMSD, ISA-95 und SysML beruhen. Das trifft auch auf den im Rahmen dieser Arbeit erstellten Lösungsansatz zu. Zu untersuchen ob und inwieweit die Kombination mehrerer de-facto Standards gewinnbringend für die automatische Modellgenerierung eingesetzt werden kann, stellt einen weiteren Forschungsbedarf dar.

Nach der detaillierten Vorstellung der automatischen Modellgenerierung im Rahmen der Inbetriebnahmephase liegt der Fokus im nächsten Kapitel auf der Modelladaption im Rahmen der Lebenszyklusphase Betrieb.

5 Konzeptbaustein für die Modelladaption

Nachdem die Lebenszyklusphase Inbetriebnahme in Kapitel 4 abgedeckt wurde, liegt der Fokus in diesem Kapitel auf der Betriebsphase. Dies bedeutet, dass die Anlage in Betrieb genommen wurde und, entsprechend der Produktionsanforderungen und -randbedingungen, WS bearbeiten kann. Im Rahmen der betriebsparallelen Simulation können, bedingt durch diverse anlagen- und/ oder modellseitige Ursachen, Abweichungen zwischen den Mess- und Simulationsdaten auftreten. Diesen Abweichungen und deren Ursachen muss sich das AS annehmen.

Im Rahmen dieses Kapitels werden zunächst in Abschnitt 5.1 die Teilziele, geltende Randbedingungen und Annahmen aufgezeigt. Ebenfalls wird rekapituliert, welche der in Abschnitt 2.2.4 definierten Anforderungen im Rahmen dieses Kapitels erfüllt werden. In Abschnitt 5.2 werden die für dieses Kapitel relevanten technologischen Konzeptbausteine behandelt. Anschließend wird in Abschnitt 5.3 das in Abschnitt 4.3 vorgestellte Teilkonzept auf die Betriebsphase erweitert und im Detail vorgestellt. Die prototypische Implementierung des Teilkonzepts wird in Abschnitt 5.4 behandelt. Anschließend wird das AS in den Abschnitten 5.5 und 5.6 auf die beiden Industrieszenarien aus Abschnitt 2.3 angewandt und validiert. Die Auftrennung der Konzeptvalidierung in zwei separate Unterkapitel dient der besseren Übersicht. Das Kapitel schließt mit einem Zwischenfazit in Abschnitt 5.7.

5.1 Teilziele, Randbedingungen und Annahmen

Wie in Abschnitt 2.2.1 beschrieben, treten auch während der Betriebsphase Herausforderungen auf, z. B. das Auseinanderdriften von Mess- und Simulationswerten, die manuelle Modellierungstätigkeiten nach sich ziehen. Ziel des in diesem Kapitel beschriebenen Teilkonzepts ist es, die Simulation hinsichtlich Genauigkeit und Laufzeit zu optimieren und auch in dieser Phase die manuellen Modellierungstätigkeiten durch automatische Abläufe zu reduzieren. Es besteht daher die Notwendigkeit, das AS um die Fähigkeit der Modelladaption zu erweitern. In diesem Kapitel muss das erweiterte AS daher, aufbauend auf den in Abschnitt 4.1 definierten Funktionen (z. B. Modulmodelle laden, Initialwerte setzen und Modulmodelle koppeln), die folgenden Anforderungen umsetzen. Das AS muss in der Lage sein:

- A3** Simulations- und Messwerte, die zur Betriebsphase aufgenommen werden, miteinander zu vergleichen,
- A4** feststellen zu können, wenn eine Abweichung der in A3 beschriebenen Werte eine vorgegebene Toleranz überschreitet,
- A5** mögliche Ursachen für die in A4 erwähnte Abweichung in den Simulationsmodellen zu lokalisieren,
- A6** eine Adaption der betroffenen Simulationsmodelle automatisch durchzuführen.

Es gelten weiterhin die in Kapitel 4 festgelegten Randbedingungen, dass ein TM der Anlage existiert und die MSM der Module als FMUs vorliegen. Auf die dadurch gewährleistete standardisierte Architektur aller MSM lässt sich ein Optimierungsalgorithmus anpassen, um so die automatische Modelladaption zu realisieren. Darüber hinaus benötigt das AS Zugriff auf die während des Betriebs gesammelten Messdaten. Diese müssen entsprechend in einer für das AS interpretierbaren Form vorliegen. Abschließend muss der

Bediener noch eine Möglichkeit haben, dem AS vorzugeben, welche Werte es überwachen soll und wie groß die Toleranz der potenziellen Abweichung ist. Hierfür bietet sich, wie für die Simulationsparameter im letzten Kapitel, eine Bedienereingabe per GUI an. Im nächsten Abschnitt soll der Stand von Wissenschaft und Technik bezüglich der Fragestellung, welche Realisierungsmöglichkeiten der Modelladaption existieren, untersucht werden.

5.2 Technologische Konzeptbausteine

Im Rahmen des für dieses Kapitel relevanten spezifischen Stands von Wissenschaft und Technik wird zunächst in Abschnitt 5.2.1 analysiert, welche Ursachencluster für Abweichungen, die eine Adaption nötig machen, existieren. Abschnitt 5.2.2 untersucht im Anschluss, welche Vorbereitungen nötig sind, um die Mess- und Simulationsdaten zu vergleichen. In Abschnitt 5.2.3 wird aufgezeigt, wie ein Adaptionsprozess grundsätzlich strukturell ablaufen kann. Neben den in Abschnitt 3.1.2 vorgestellten Ansätzen zur Modelladaption, welche sich hauptsächlich auf die Parameteradaption von Physikmodellen beschränken, sollen in Abschnitt 5.2.4 noch die Detaillierungsgrade von Modellen betrachtet werden. Diese stellen eine weitere Möglichkeit dar, um Modelle automatisch im Betrieb zu adaptieren.

5.2.1 Analyse von Ursachenclustern für notwendige Modelladaptionen

In der Praxis sind die Ursachen für eine Abweichung zwischen Mess- und Simulationsdaten vielfältig [CGG+05]. Abbildung 30 zeigt eine Übersicht der aus der Literatur (u. a. [AH20, Roy19]) zusammengetragenen Ursachen und unterteilt diese in die Cluster „Anlage“ und „Simulation/ Modell“. Anlagenseitig lassen sich die Ursachen in exo- und endogene Ereignisse clustern. Als von außen auf die Maschine wirkende Ursachen sind beispielsweise Umwelteinflüsse oder Ausfälle in der Energieversorgung zu nennen. Endogene Ereignisse reichen von Material- und Werkstückfehlern über Fehler bei der Systemauslegung bis hin zu Fehlern bei der Bedienung der Anlage [AH20]. Da diese Fehler, abgesehen von systematischen Fehlern, stochastischer Natur sind, werden sie im Rahmen dieser Arbeit nicht betrachtet. Systematische Fehler in Form von fehlerhafter Systemauslegung werden ebenfalls vernachlässigt, da die betrachteten Systeme mittels MBSE ausgelegt und durch Simulation im Rahmen der Inbetriebnahme geprüft werden.

Simulations- und modellseitig sind als Ursachen für eine Abweichung exemplarisch Modellierungsfehler, die Wahl von falschen Algorithmen und numerische Fehler zu nennen. Numerische Fehler können durch die Wahl geeigneter und aufgabenspezifischer Solver und entsprechender Schrittweiten minimiert werden. Aber auch wenn diese Fehler nicht auftreten und das System korrekt als Modell abgebildet wurde, ist es möglich, dass die Parameter des Modells nicht optimal eingestellt und/ oder Detaillierungsgrade falsch gewählt wurden. [Roy19]

Wie in Abschnitt 2.1.5 definiert, handelt es sich bei einem Modell um eine vereinfachte Nachbildung eines Systems. Werden charakteristische Eigenschaften des Systems bei der Modellierung zu stark vereinfacht, z. B. durch die lineare Abbildung von nichtlinearen Zusammenhängen, können große Abweichungen auftreten. Abhängig von dem jeweiligen Anwendungsfall oder den Stakeholdern, für die die Simulation bestimmt ist, können unterschiedliche Detaillierungsgrade sinnvoll sein. Dieses Thema wird in Abschnitt 5.2.4 umfänglich behandelt. Zuletzt sind noch Abweichungen zu nennen, die auftreten, wenn die gesammelten anlagen- und simulationsseitigen Daten nicht entsprechend aufbereitet werden. Dieses Ursachen-

cluster wird im nächsten Abschnitt aufgegriffen und im Detail behandelt. Der Fokus für die Modelladaption liegt somit auf den drei Ursachenclustern „Detaillierungsgrad Modell“, „Parametrierung Modell“ und „Datenaufbereitung“.

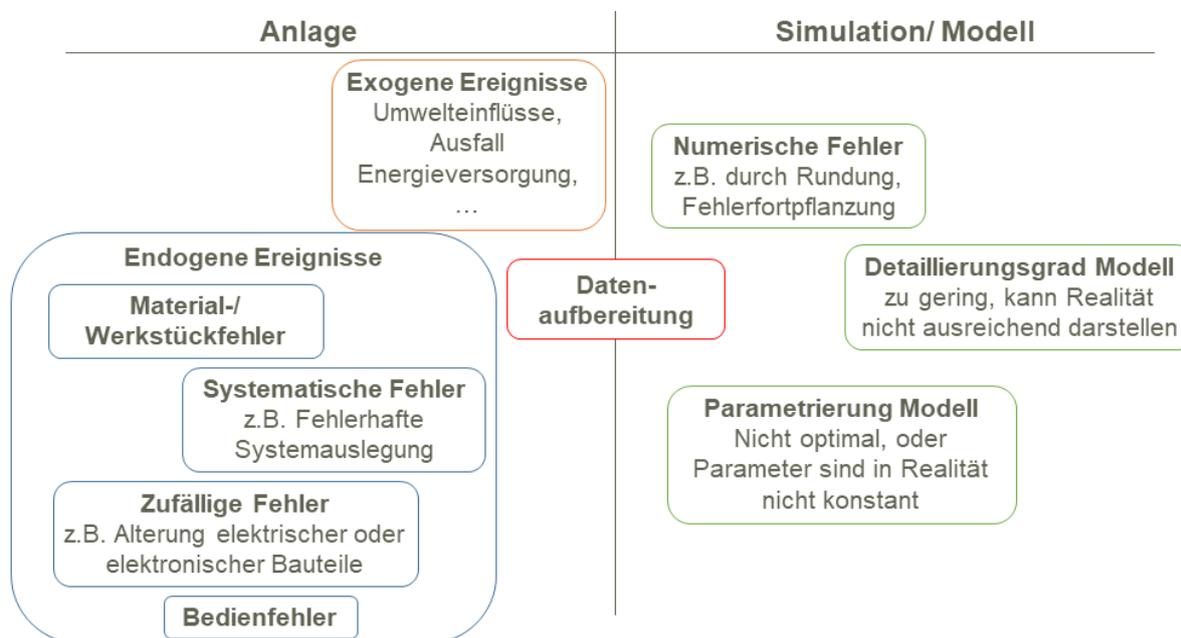


Abbildung 30: Ursachencluster für Abweichungen zwischen Mess- und Simulationsdaten.

5.2.2 Aufbereitung der Mess- und Simulationsdaten

Die in Abbildung 30 rot markierte Datenaufbereitung betrifft beide Cluster und ist grundsätzlich immer notwendig. Hierzu gehören u. a. Interpolation, wenn Abtastrate und Schrittweite nicht zusammenpassen, das Verschieben der Zeitreihen gegeneinander, wenn beispielsweise die Messung zu spät gestartet wurde, oder eine Konsistenzprüfung der Einheiten. Wird dieser Schritt nicht durchgeführt, können Abweichungen auftreten, obwohl sich sowohl Anlage als auch Simulation im Sollzustand befinden. Auch hierfür gilt es eine automatisierte Lösung zu finden, um den wirtschaftlichen Einsatz betriebsparalleler Simulation zu gewährleisten. Bevor die Adaption durchgeführt wird, muss zunächst die Abweichung der Datensätze Δ_{ist} berechnet werden. Für den Vergleich zwischen realer Anlage und Simulation muss es möglich sein, Messdaten in die AS zu laden. Für einen Vergleich der Datensätze ist es jedoch notwendig, dass die gleiche Abtastfrequenz und Schrittweite verwendet wird. Da dies in der Regel nicht zu erwarten ist, können Methoden wie Interpolation eingesetzt werden, um unterschiedliche Zeitdiskretisierungen handzuhaben. Denkbar ist z. B. auch, dass die Messung zu spät gestartet wurde und der Zeitversatz der Datensätze zu einer großen Abweichung führt, obwohl Messung und Simulation nahezu identische Werte liefern, beispielsweise bei der Phasenverschiebung einer Sinuswelle. Daher kann es notwendig sein, die Zeitreihen vor der Adaption gegeneinander zu verschieben, um die Abweichung zu minimieren. Für diese Fälle sind geeignete Kontroll- und Korrekturmechanismen erforderlich. Ein statistischer Zusammenhang zwischen den gemessenen und simulierten Werten und deren möglicher Zeitverschiebung kann mit Hilfe von Korrelationsfunktionen ermittelt werden. Nach [BCH04] wird die normierte Kreuzkorrelation $r_{xy}(k)$ für zwei zeitdiskrete Signale $x[i]$ und $y[i]$ mittels

$$r_{xy}(k) = \frac{\sum_{i=1}^N x[i] \cdot y[i+k]}{\sqrt{\sum_{i=1}^N (x[i])^2 \cdot \sum_{i=1}^N (y[i+k])^2}} \quad (1)$$

berechnet, wobei k der Zeitverschiebung entspricht. Die normierte Kreuzkorrelation r_{xy} ergibt einen Wert von 1, wenn die beiden Signale identisch sind; bei Signalen mit entgegengesetzter Phase ist der Wert -1. Signale korrelieren nicht, wenn der Wert der normierten Kreuzkorrelation für alle Werte von k nahe Null ist. [BCH04]

Vor der Durchführung der Anpassung muss der Wert bestimmt werden, für den $r_{xy}(k) \rightarrow \max$ gilt. Die Zeitreihen müssen dann entsprechend gegeneinander verschoben werden. Das oben beschriebene Beispiel einer zu spät gestarteten Messung ist in Abbildung 31 dargestellt. Im oberen Teil der Abbildung ist in orange der Verlauf eines gemessenen Signals y_{mess} und in blau des dazugehörigen simulierten Signals y_{sim} dargestellt. Über Gleichung (1) lässt sich die Kreuzkorrelation der beiden Signale, dargestellt im unteren Teil der Abbildung, berechnen. Aus dem Plot der Kreuzkorrelation geht hervor, dass die Ähnlichkeit der beiden Signale bei einer Verschiebung um etwa eine Sekunde am größten ist. Das Vorzeichen des Zeitversatzes ist abhängig von der Zuweisung von y_{mess} und y_{sim} zu x und y in Gleichung (1).

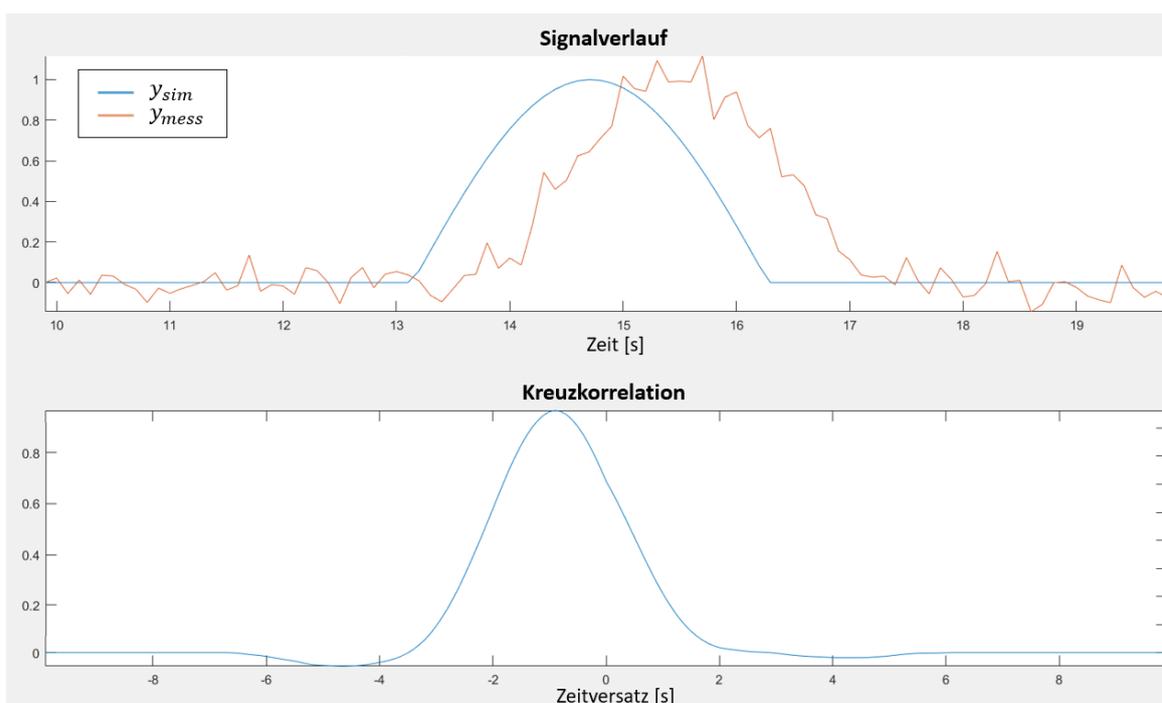


Abbildung 31: Zeitversatz zwischen Mess- und Simulationssignal (oben) und berechnete Kreuzkorrelation (unten).

Weitere negative Einflussfaktoren können Inkonsistenzen bei Einheiten oder Nachkommastellen sein. Wenn nach allen Korrekturmaßnahmen Δ_{ist} immer noch größer als eine vom Bediener vorgegebene maximale Abweichung Δ_{max} ist, muss eine Modelladaption durchgeführt werden. Die Wahl der maximalen Abweichung Δ_{max} ist von verschiedenen Faktoren, wie dem Anwendungsfall, dem betrachteten System und den Zielen der Interessengruppen abhängig und kann daher nicht pauschal durch einen fixen Wert beschrieben werden. Ein methodischer Ansatz zur Wahl von Δ_{max} könnte z. B. die Anwendung der in [BKH+20] vorgestellten Metrik zur systematischen Bewertung der Qualität von Simulationsmodellen sein. Wie ein Adaptionprozess nach der Überschreitung von Δ_{max} strukturell ablaufen kann, wird im nächsten Abschnitt untersucht.

5.2.3 Struktureller Ablauf eines Adaptionprozesses

In Abschnitt 1.1 wurde Adaption bereits als „*die Anpassung von Modellen an veränderte Bedingungen ohne vollständigen Neuaufbau*“ [BER13] definiert. Diese Modellanpassung gilt es nun im Rahmen dieser Arbeit methodisch und strukturiert durchzuführen.

Abbildung 32 veranschaulicht den strukturellen Ablauf einer Adaption und teilt diese in vier Phasen ein. Zu jeder Phase sind auf der rechten Seite in der Abbildung die Herausforderungen formuliert. Die Abbildung ist an [MHW+18] angelehnt, wobei Marks et al. in ihrer Arbeit den Adaptionprozess für Produktionsanlagen betrachtet haben. Übertragen auf die Modelladaption bleibt der Adaptionprozess auf der linken Seite in Abbildung 32, bis auf den Prozesseingang, unverändert. Bei Marks et al. wird überprüft, ob die betrachtete Produktionsanlage eine Produktionsanfrage erfüllen kann. Im Rahmen dieses Kapitels muss stattdessen überprüft werden, ob nach Vergleich der Mess- und Simulationswerte die Abweichung dieser Werte innerhalb der vorgegebenen Toleranz liegt. Die sich ergebenden Herausforderungen sind entsprechend für den Adaptionprozess von Simulationsmodellen umformuliert. Der Adaptionprozess und die jeweiligen Herausforderungen werden nun phasenweise vorgestellt. In Phase 1 des Prozesses wird überprüft, ob die nach Anforderung A4 definierte Toleranz der Abweichungen zwischen Mess- und Simulationswerten eingehalten wird. Hierzu bedarf es einer formalisierten Toleranzangabe sowie einer Datenaufbereitung, wie in Abschnitt 5.2.2 aufgezeigt. Wird die geforderte Toleranz eingehalten, kann der Prozess an dieser Stelle abgebrochen werden. Andernfalls findet ein Übergang zu Prozessphase 2 statt. In dieser Phase werden Adaptionsoptionen der Modelle generiert. Für die Generierung dieser Optionen wird ein geeigneter Mechanismus benötigt. In Phase 3 findet die Evaluation der zuvor generierten Adaptionsoptionen statt. Hierfür müssen zunächst geeignete Gütekriterien definiert und diese anschließend für alle Optionen berechnet werden. Ein Konzept für die in den Phasen 2 und 3 benötigten Mechanismen wird in Abschnitt 5.3.2 vorgestellt. Phase 4 legt abschließend den Fokus auf die Selektion einer geeigneten Option. Anhand strategischer Überlegungen, z. B., ob eine hohe Simulationsgüte oder eine hohe Performance der Simulation relevanter ist, in Kombination mit der berechneten Bewertung der Optionen hinsichtlich der Gütekriterien, kann die Entscheidung für eine Adaptionsoption getroffen werden. Es ist sinnvoll, diesen Adaptionprozess nicht einmal, sondern zyklisch und parallel zum Betrieb durchzuführen.

In Abschnitt 3.1 wurden bereits Ansätze zur (teil-)automatischen Modelladaption vorgestellt. Der Fokus dieser Ansätze lag jedoch hauptsächlich darauf, Adaptionsoptionen auf Basis von Parametervariationen zu generieren. Eine Methode, die zusätzlich genutzt werden kann, stellt die Adaption auf Basis der Detaillierungsgrade der Modelle dar. Hierbei wird der Zusammenhang ausgenutzt, wie umfänglich ein System modelliert und welche Aspekte vernachlässigt wurden. Im nächsten Abschnitt werden daher die Grundlagen zu Modelldetaillierungsgraden untersucht.

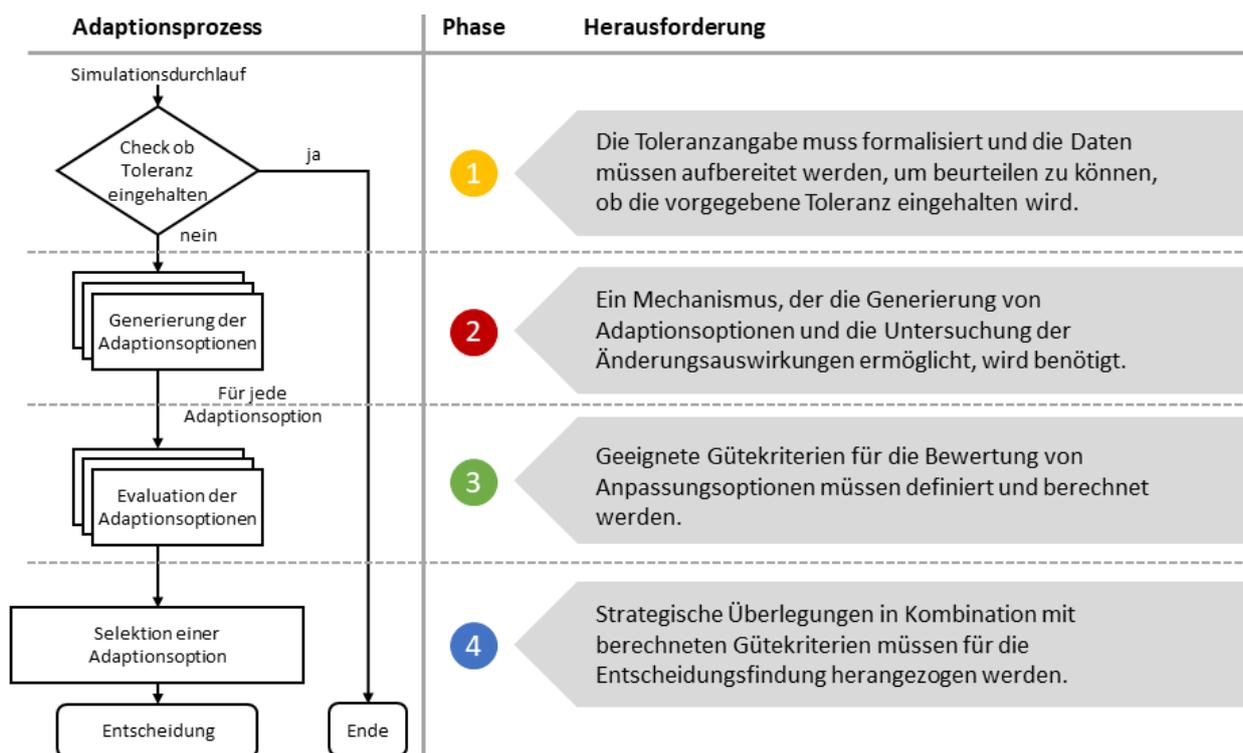


Abbildung 32: Struktureller Ablauf und Herausforderungen eines Adaptionprozesses angelehnt an [MHW+18].

5.2.4 Detaillierungsgrad von Modellen

Nach Puntel-Schmidt und Fay [PF15] können Produktionsanlagen grundsätzlich in jedem vom Kunden gewünschten Detaillierungsgrad (engl. Level-of-Detail, kurz LoD) simuliert werden, so dass eine breite Palette von Simulationsmodellen zur Verfügung steht. Sie folgern weiter, dass man grundsätzlich davon ausgehen könnte, dass die Simulation auf dem höchsten und komplexesten verfügbaren LoD immer den Anforderungen des Kunden entspricht und daher das Mittel der Wahl sein sollte. Dieser Modellierungsansatz hat nach Puntel-Schmidt und Fay jedoch gravierende Nachteile. Die Autoren nennen u. a. den Aufwand, Feldgeräte zu modellieren, den Knowhow Schutz der Hersteller sowie die Tatsache, dass komplexe Simulationsmodelle in der Regel rechenintensiv sind [RHJ+21] und verhindern können, dass größere Simulationen angemessen ausgeführt werden können. Geht mit einer Erhöhung der Modellierungstiefe eine Vergrößerung der Gleichungs- und Variablenanzahl einher, kann dies wiederum zu einer schlechteren Robustheit und numerischen Stabilität des Modells führen [BKH+20]. Es gilt somit ein angemessenes Maß für den LoD zu finden [SHR+21*]. Puntel-Schmidt und Fay schließen aus ihrer Untersuchung, dass je nach Anwendungsfall nur wenige spezifische Teile einer Produktionsanlage mit einem hohen und komplexen LoD simuliert werden sollten. Der größte Teil des Modells einer Produktionsanlage sollte so einfach wie möglich gehalten werden und leicht zu validieren sein. In Anlehnung an [MP13] formulieren Puntel-Schmidt und Fay als Regel für den LoD, dass Modelle so abstrakt und wenig komplex wie möglich und so komplex und detailliert wie nötig modelliert werden sollen. [PF15]

In [VDI/VDE 3693-1] werden verschiedene Modelltypen für die virtuelle Inbetriebnahme diskutiert und vorgeschlagen. Diese Modelltypen unterscheiden sich hinsichtlich ihrer Granularität. In der Richtlinie werden vier Modelltypen für diskrete Prozesse einschließlich Förderanlagen und Materialhandhabung definiert. Die Modelltypen werden als ereignisbasierte Modelle, kinematische Modelle, kinematische Modelle mit 3D-Modellen und dynamische Modelle bezeichnet. Das ereignisbasierte Modell ist bezüglich seines

LoD am niedrigsten, das dynamische Modell am höchsten einzustufen. Je höher der LoD, desto mehr Aspekte des realen Systems können dargestellt werden. Im ereignisbasierten Modell sind nur wenige geometrische Informationen erforderlich, die sich auf die Bereitstellung binärer Antwortsignale beschränken, um z. B. Sensoren am Ende eines Förderbands abzubilden [TFD18]. Für die ereignisbasierte Modellierung können beispielsweise Zustandsautomaten und Petri-Netze verwendet werden. Andererseits werden kinematische und dynamische Modelle auf der Grundlage von 3D-Geometrien erstellt. Ein kinematisches Modell kann als Blockdiagramm oder als mathematische Gleichung beschrieben werden, die den Bewegungsweg, die Freiheitsgrade und die kinematische Kette eines Systems definiert. Wenn ein Blockdiagramm oder eine Gleichung in das 3D-Modell eingebettet ist, handelt es sich um ein kinematisches Modell mit 3D-Modell. Diese Art von Modell ist besser geeignet als ein rein kinematisches Modell, wenn Bewegungsvisualisierung und Kollisionserkennung erforderlich sind. Das dynamische Modell ermöglicht die freie Bewegung im 3D-Raum nach den Gesetzen der Physik zusätzlich zu dem im kinematischen Modell festgelegten Pfad. Die freie Bewegung unterliegt der Masse, externen Kräften usw. Neben diesen vier Modelltypen für die Prozesssimulation umfasst der Leitfaden auch E/A-Modelle und Gerätemodelle für die Signal- und Einzelsensor- oder Aktorsimulation. Die Auswahl des geeigneten Modelltyps hängt von den Anforderungen und Testfällen der Anwender ab. Verschiedene Detailebenen können kombiniert werden, wenn ein unterschiedlicher Abstraktionsgrad erforderlich ist. [TFD18, VDI/VDE 3693-1]

Kufner [KUF12] definiert in ihrer Arbeit, im Unterschied zu [VDI/VDE 3693-1], fünf verschiedene Modellierungstiefen für Maschinenmodelle zur Hardware-in-the-Loop-Simulation. Diese tragen die Bezeichnungen logisches Abbild, logisches Abbild mit Zeitverhalten, physikalisches Prinzipabbild, physikalisches Konstruktionsabbild und physikalisches Zustandsabbild. Das erstgenannte Abbild stellt dabei die niedrigste, das letztgenannte die höchste Modellierungstiefe dar. Bei Kufner enthalten Modelle jeder Modellierungstiefe die exakte Modellierung des Kommunikationsverhaltens. Beim logischen Abbild wird lediglich die Wirkkette, der logische Zusammenhang zwischen Ein- und Ausgängen, abgebildet. Über welches physikalische Prinzip der Zusammenhang besteht, wird nicht betrachtet. Bei der nächsten Modellierungstiefe werden zusätzlich die Übergangszeiten in die Wirkkette integriert. Ab der dritten Modellierungstiefe, dem physikalischen Prinzipabbild, werden nicht mehr die logischen Zusammenhänge modelliert. Stattdessen werden die grundlegenden physikalischen Wirkprinzipien abgebildet. Hier orientiert sich Kufner an den Wirkprinzipien nach der VDI-Richtlinie 2222. Im physikalischen Konstruktionsabbild werden disziplinspezifische Informationen eingebunden, z. B. die dynamischen Eigenschaften von Bauteilen, um so physikalische Effekte zu modellieren. Bis dahin wird *„eine Maschine modelliert, die sich nach den Konstruktionsannahmen verhält“*. In der letzten Modellierungstiefe, dem physikalischen Zustandsabbild, werden zusätzlich bestehende Fehler an der Maschine, wie fehlerhaft gefertigte Einzelkomponenten oder Störgrößen durch Verschleiß und Verformung, in das Maschinenmodell integriert [KUF12]. Diese letzte Modellierungstiefe ist in [VDI/VDE 3693-1] nicht explizit vorgesehen.

Puntel-Schmidt und Fay haben in ihrer Arbeit [PF15] Publikationen der Jahre 1999 bis 2013 bezüglich Definitionen von Modell-LoD untersucht. Sie kommen zu dem Schluss, dass zwar alle genannten Ansätze in ihrem jeweiligen Anwendungsfall ihre Berechtigung haben, eine systematische Definition von LoD und passenden Modelltypen jedoch nicht zufriedenstellend erfolgt, da Granularität bzw. Art und Anzahl der Modelltypen nicht universell definiert sind. Die Arbeit von Stegmaier et al. [SDJ+22] unterstreicht diese Problematik. Neben den Arbeiten von Puntel-Schmidt und Fay [PF15], Kufner [KUF12] sowie der VDI/VDE-Richtlinie 3693 Blatt 1 [VDI/VDE 3693-1] identifizieren Stegmaier et al. noch sechs weitere Ansätze zur

Klassifikation von Detaillierungsgraden von Modellen. Auf Basis der bestehenden Ansätze führen sie zudem noch einen weiteren Klassifikationsansatz ein [SDJ+22]. Im Rahmen der vorliegenden Arbeit werden MSM gekoppelt und adaptiert, die von unterschiedlichen Zulieferern stammen können. Es ist anzunehmen, dass auch diese Zulieferer, trotz der Standardisierungsbestrebungen von [VDI/VDE 3693-1], keine einheitliche Definition von LoD verwenden. Um den LoD trotzdem für die Modelladaption nutzen zu können, wird für den weiteren Verlauf dieser Arbeit angenommen, dass ein MSM mit höherem LoD eine genauere Darstellung des Anlagenmoduls liefert und mit einem höheren Rechenaufwand einhergeht.

Nachdem der Stand von Wissenschaft und Technik hinsichtlich des strukturellen Ablaufs eines Adaptionprozesses, Möglichkeiten der Aufbereitung von Datensätzen für den Vergleich sowie den LoD von Simulationsmodellen untersucht wurde, wird im Folgenden das Teilkonzept für die (teil-)automatische Modelladaption im Detail vorgestellt.

5.3 Konzept zur betriebsparallelen Adaption von Co-Simulationen

5.3.1 Konzeptskizze und Möglichkeiten der Adaption

In diesem Abschnitt wird, wie in Abbildung 33 dargestellt, das Konzept zur automatischen Komposition und Konfiguration von Co-Simulationen aus Abbildung 25 um die betriebsparallele Adaption (blaue Pfade) erweitert. Eine Basis-Aufgabe des AS aus Kapitel 4 ist es, die in der Anlage vorhandenen Module auf Grundlage des TM zu identifizieren und die benötigten MSM aus FMU-Bibliotheken zu laden. Das AS generiert auf Basis des TM ein Skript zur Parametrierung und Steuerung der Simulation (vgl. Abschnitt 4.3). Diese Funktionen werden im Folgenden auf die Betriebsphase erweitert. Neben den Kopplungsinformationen können aus dem TM weitere Informationen, wie z. B. Modellparameter und das Vorhandensein von MSM in unterschiedlichen LoD, abgeleitet werden. Erkennt das AS, dass die Abweichung der Mess- und Simulationsdatensätzen einen vorgegebenen Wert überschreitet, wird zunächst eine Ursachenanalyse durchgeführt. Es gilt die Frage zu beantworten, ob die Ursache anlagen- oder simulationsseitig einzuordnen ist (vgl. Abbildung 30). Hierfür wird zunächst angenommen, dass das Gesamtmodell korrekt ist und die Simulation ordnungsgemäß läuft. Die Messdaten werden daraufhin mit einer Bibliothek aus Fehlerbildern abgeglichen, um Komponentenverschleiß oder -ausfall zu identifizieren. Dieser Abgleich erfolgt automatisch, z. B. kann mit historischen Fehlerbildern aus der Bibliothek ein neuronales Netz trainiert werden [TFV+19]. Nach dem Training dienen die aktuellen Messdaten als Eingangsgröße für das neuronale Netz und werden von diesem kategorisiert. Wenn ein bekanntes Fehlerbild identifiziert wird, wird eine Meldung ausgegeben und eine Anlagenwartung muss eingeleitet werden (linker Pfad). Für diese Analysefunktionen existieren bereits Lösungen aus dem Bereich der prädiktiven Instandhaltung [u. a. SSP+15, TFV+19]. Ein Großteil dieser Lösungen ist den Bereichen „*Machine Learning*“ bzw. „*Deep Learning*“ zuzuordnen [KKN+22]. Der Aufbau einer Fehlerbilder-Bibliothek sowie die Vorbereitung der datenverarbeitenden Algorithmen ist ein kontinuierlicher anlagenspezifischer Prozess, welcher meist eine beträchtliche Menge an Messdaten voraussetzt. Keleko et al. [KKN+22] identifizieren in ihrer Studie Daten als Schlüsselement für die prädiktive Instandhaltung und die Erfassung dieser Daten als eine große Herausforderung für Unternehmen. Um von Algorithmen verwertet werden zu können, müssen die gesammelten Rohdaten, bevor eine Kategorisierung möglich ist, noch vorverarbeitet werden. Dies umfasst unter anderem die Schritte Datentransformation (z. B. Normalisierung), Datenbereinigung (z. B. Behandlung fehlender

Daten und Entfernung von Ausreißern) und Datenreduktion (z. B. Dimensionalitätsreduktion und Reduktion der Anzahl) [TFV+19]. Diese Vorgänge stufen Keleko et al. als zeitaufwändig und anspruchsvoll ein. Zudem ist entsprechendes Expertenwissen erforderlich. Die manuelle Kategorisierung birgt ein Fehlerisiko und verursacht erhebliche wirtschaftliche und betriebliche Kosten. [KKN+22]

Da der Fokus auf der simulationsseitigen Adaption liegt, wird der linke Pfad in der Konzeptskizze im Rahmen der anschließenden Implementierung und Konzeptvalidierung nicht weiter betrachtet, sondern der rechte, d.h. die Ursache der Abweichung wird im Modell verortet.

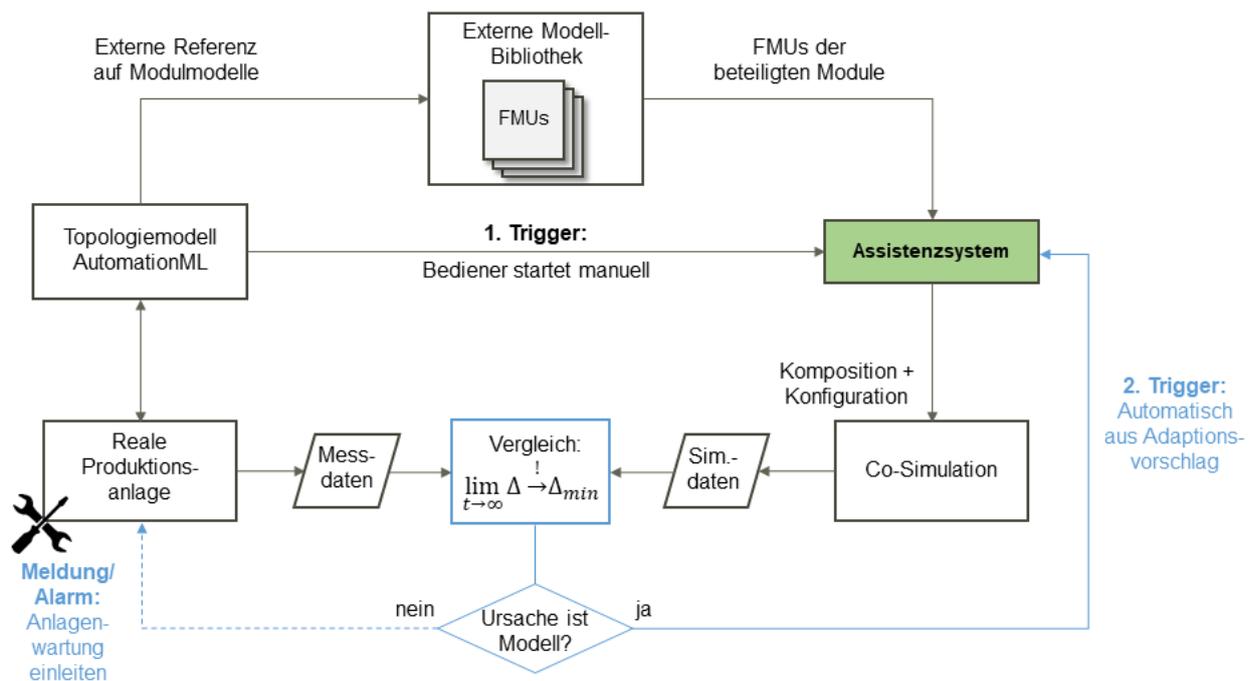


Abbildung 33: Konzept zur automatischen Modellgenerierung und betriebsbegleitenden -adaption.

Im nächsten Abschnitt wird das Problem zunächst in eine maschinell interpretier- und lösbare Form gebracht.

5.3.2 Formulierung der Adaptionaufgabe

Die Formulierung der Adaptionaufgabe führt zu einem Optimierungsproblem. Allgemein formulieren lässt sich ein Optimierungsproblem als

$$\min_{\underline{x} \in \mathbb{R}^n} f(\underline{x}) \quad , \quad f: \mathbb{R}^n \rightarrow \mathbb{R} \quad . \quad (2)$$

Die Funktion f wird als Güte- oder Zielfunktion, der Vektor \underline{x} als Designvariable bezeichnet. Möchte man beispielsweise einen Produktionsprozess hinsichtlich der benötigten Zeit optimieren, ist die Zeit die Zielfunktion. Designvariablen, über die man Einfluss auf die benötigte Zeit nehmen kann, sind beispielsweise Materialeigenschaften des Werkstücks sowie die Prozessparameter. Das Optimierungsproblem (2) ist gelöst, wenn ein \underline{x}^* gefunden wurde, für das gilt

$$f(\underline{x}^*) \leq f(\underline{x}) \quad \forall \underline{x} \in \mathbb{R}^n \quad . \quad (3)$$

Dabei müssen, insbesondere bei Optimierungen von realen Anlagen und Maschinen, unterschiedliche Rand-/ Nebenbedingungen erfüllt werden. Beispielsweise können Prozessparameter nicht beliebig variiert werden. Damit erweitert sich (2) zu

$$\begin{aligned} \min_{\underline{x} \in \mathbb{R}^n} \quad & f(\underline{x}) \quad , \quad f: \mathbb{R}^n \rightarrow \mathbb{R} \\ \text{s. t.} \quad & g_j(\underline{x}) \leq 0 \quad , \quad j = 1, \dots, e \quad , \\ & h_l(\underline{x}) = 0 \quad , \quad l = 1, \dots, r \end{aligned} \quad (4)$$

wobei die zusätzlichen Ungleichungen g_j und Gleichungen h_l den Lösungsraum einschränken. Im Rahmen der betriebsparallelen Modelladaption beschreibt eine offensichtliche Zielfunktion das Maß an Übereinstimmung zwischen Mess- und Simulationen \underline{y}_{mess} und \underline{y}_{sim} , mit $\underline{y}_{mess}, \underline{y}_{sim} \in \mathbb{R}^{k \times v}$. Es wird angenommen, dass k Mess- und Simulationspunkte vorliegen und für die Anzahl an Sensorpaaren $v \geq 1$ gilt. Als Sensorpaare werden die Sensoren aufgefasst, die sowohl in der realen Anlage als auch in der Simulation enthalten sind. Ziel ist, wie beschrieben, die Minimierung der Abweichung zwischen den simulierten und real gemessenen Werten. Zunächst gilt es die Frage zu klären, wie das Maß an Übereinstimmung über alle Sensorpaare und Mess-/Simulationspunkte bestimmt werden kann. Eine häufig eingesetzte Methode ist das Bestimmtheitsmaß R^2 [FKP+16, LEH20]. Dieses nutzt die Residuenquadratsumme, welche beispielsweise beim Least-Square-Verfahren für lineare Regressionsrechnung verwendet wird, und betrachtet zusätzlich den empirischen Mittelwert \bar{y}_{mess} . Jedoch haben eigene Untersuchungen, die auf Basis der Untersuchungen in [SN10] durchgeführt wurden, für im Anlagenbau häufig eingesetzte nichtlineare Modelle ergeben, dass sich Alternativen wie das *Akaike information criterion* (AIC), das *Corrected Akaike information criterion* (AIC_C) oder das *Bayesian information criterion* (BIC) besser eignen [HBF21*a, HBF21*b]. Als Gründe sind zu nennen, dass das Bestimmtheitsmaß R^2 zum einen wenig sensitiv ist [SN10, ZYT08]. Bei der Abbildung eines Systems durch mehrere nichtlineare Modelle mit unterschiedlichen LoD variieren die Bestimmtheitsmaße der Modelle nur geringfügig. [SN10] beobachten bei der Verwendung von ungenauen nichtlinearen Modellen im Vergleich zu detaillierten Modellen, die das Systemverhalten besser beschreiben, lediglich Änderungen in der dritten oder vierten Nachkommastelle von R^2 . Zum anderen gibt es bei einer Auswahl von Modellen eine klare Tendenz des Bestimmtheitsmaßes R^2 hin zu Modellen mit mehr Parametern. Diese Tendenz wird bei den aufgezeigten Alternativen durch einen Strafterm bei der Berechnung unterbunden. Exemplarisch berechnet sich das AIC nach [SN10]:

$$\begin{aligned} AIC &= 2n - 2 \ln(L) \quad , \text{ mit} \\ L &= 0.5 \left(-N \left(\ln 2\pi + 1 - \ln N + \ln \sum_{i=1}^k z_i^2 \right) \right) . \end{aligned} \quad (5)$$

Dabei beschreibt L die Likelihood-Funktion, n steht für die Anzahl an Modellparametern, da $\underline{x} \in \mathbb{R}^n$ angenommen wurde. Die Residuen aus dem nichtlinearen Least-Squares-Fit werden durch z_i beschrieben und N steht für die Anzahl der Residuen. Über die Likelihood-Funktion wird die Anpassungsgüte des Modells belohnt, die Anzahl an Modellparametern geht als Strafterm ($2n$ in der ersten Zeile von (5)) in die Berechnung ein. Dieser verhindert ein Overfitting der Modelle. Hawkins [HAW04] definiert Overfitting als die Verwendung von Modellen oder Verfahren, die gegen das Prinzip der Parsimonie verstoßen, d. h. die mehr Terme als nötig enthalten oder kompliziertere Ansätze als nötig verwenden. Durch mehr Modellparameter kann, bis zu einem Grenzwert, eine bessere Anpassungsgüte erreicht werden. Es ist allerdings

nicht erstrebenswert, diesen Grenzwert unter allen Umständen zu erreichen, da zusätzliche Parameter u. a. die Komplexität des Modells und den Aufwand für das Parametertuning erhöhen. Beim Vergleich mehrerer Modelle ist ein minimaler AIC-Wert zu bevorzugen. Die Berechnung der Modellgüte mittels BIC beinhaltet ebenfalls die Likelihood-Funktion, gewichtet bzw. bestraft die Anzahl an verwendeten Modellparametern jedoch unterschiedlich. Im Rahmen dieser Arbeit wird exemplarisch das AIC als Gütekriterium verwendet und die Betrachtung darauf beschränkt, um den Adaptionprozess entsprechend Abbildung 32 möglichst einfach zu halten. Die Verwendung eines der Gütekriterien genügt, um den Adaptionprozess durchführen zu können und so das vorgestellte Konzept im weiteren Verlauf der Arbeit zu validieren. Die Selektion der optimalen Adaptionsoption stellt keine Anforderung im Rahmen dieser Arbeit dar. Sollte das AS in eine Industrie-Applikation überführt werden, wäre es jedoch notwendig, verschiedene Kriterien zu betrachten oder einem Bediener zur Auswahl zu stellen, um so einen möglichst großen Vorteil aus der Modelladaption zu ziehen. Nach Brewer et al. ist es nicht sinnvoll, sich auf ein einziges Informationskriterium zu verlassen, da dieses wahrscheinlich nicht universell erfolgreich sein wird [BBC16].

Um mit den Definitionen aus (2-4) konsistent zu bleiben, wird (5) umformuliert und in Abhängigkeit der Designvariablen \underline{x} gesetzt

$$f(\underline{x}) = 2n - 2 \ln \left(0.5 \left(-N \left(\ln 2\pi + 1 - \ln N + \ln \sum_{i=1}^k \left(\underline{y}_{i,mess} - \underline{y}_{i,sim}(\underline{x}) \right)^2 \right) \right) \right). \quad (6)$$

Im nächsten Schritt gilt es zu untersuchen, von welchen Einflussfaktoren die Simulationswerte \underline{y}_{sim} abhängen. Aus Sicht eines MSM stellt der individuelle Parametersatz $\underline{p}_{individ}$ den offensichtlichsten Einfluss auf die von diesem Modell generierten Simulationswerte dar. Da bei einer Co-Simulation jedoch alle MSM über ihre Ein- und Ausgänge gekoppelt sind, müssen Wechselwirkungen zwischen den Modellen beachtet werden. Damit haben auch die Parametersätze aller anderen direkt oder indirekt gekoppelten Module \underline{p}_{global} Einfluss auf die Simulationswerte des individuell betrachteten Moduls. Die Parametersätze aller in der Co-Simulation enthaltenen Modulmodelle werden als $\underline{p}_{ges} = \begin{bmatrix} \underline{p}_{individ} \\ \underline{p}_{global} \end{bmatrix}$ zusammengefasst. Neben Modellparametern haben auch die LoD aller Modelle \underline{d}_{ges} einen Einfluss auf die Simulationsmodelle. Der Grund, alle LoD zu betrachten, ist analog zum Wechselwirkungsargument bei den Parametersätzen. Im Rahmen dieses Beitrags wird der LoD, in Anlehnung an [PUN18], als $d \in \mathbb{N}_+$ angenommen, wobei Grad 1 die niedrigste Detaillierungsstufe eines Modells darstellt. Für das AS und den Optimierungsalgorithmus ist es wichtig, dass alle Modellquellen (z. B. die einzelnen Zulieferer) die gleichen Definitionen verwenden. In Abschnitt 5.2.4 wurde hierfür angenommen, dass, selbst wenn die Zulieferer nicht die gleiche Auffassung von Modelltypen für LoD z. B. nach [VDI/VDE 3693-1] teilen, ein MSM mit höherem LoD eine genauere Darstellung des Anlagenmoduls liefert und mit einem höheren Rechenaufwand einhergeht. Damit lässt sich (6) spezifizieren zu

$$f(\underline{p}_{ges}, \underline{d}_{ges}) = 2n - 2 \ln \left(L(\underline{p}_{ges}, \underline{d}_{ges}) \right). \quad (7)$$

Wie in (4) definiert, gibt es für Optimierungsprobleme häufig Randbedingungen zu erfüllen. In dem hier betrachteten Problem sind das zum einen die Grenzen der Modellparameter \underline{p}_{ges} . Im Allgemeinen ist davon auszugehen, dass die Modellparameter eine Unter- und eine Obergrenze haben, da es sich um physikalische Größen handelt. Die Grenzwerte werden in den beiden Vektoren $\underline{p}_{ges,min}$ und $\underline{p}_{ges,max}$ zusammengefasst. Mit dieser Erkenntnis und der in (7) formulierten Zielfunktion lässt sich das Optimierungsproblem nach (4) folgendermaßen aufstellen

$$\begin{aligned} \min_{\underline{p}_{ges}, \underline{d}_{ges}} f(\underline{p}_{ges}, \underline{d}_{ges}) \\ \text{s.t. } \underline{p}_{ges} - \underline{p}_{ges,max} \leq 0 \\ -\underline{p}_{ges} + \underline{p}_{ges,min} \leq 0 \end{aligned} \quad (8)$$

In Bezug auf den LoD gilt es, ein geeignetes Maß zu finden. Detailreichere Modelle können zur Minimierung von (8) beitragen, aber auch die benötigte Rechenzeit erhöhen. Aus Sicht der Rechenzeit gilt es somit, den LoD des Gesamtmodells, welcher als Summe der Einzelmodelle angenommen wird, zu minimieren. Somit lässt sich eine zweite Zielfunktion f_2 formulieren

$$\min_{\underline{d}_{ges}} f_2(\underline{d}_{ges}) = \sum_i^m d_m, \quad (9)$$

wobei m der Anzahl an Modellen in der Co-Simulation entspricht. Das Optimierungsproblem F besteht somit aus zwei in Konflikt zueinanderstehenden Zielfunktionen.

$$\begin{aligned} \min_{\underline{p}_{ges}, \underline{d}_{ges}} F(\underline{p}_{ges}, \underline{d}_{ges}) = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} 2n - 2 \ln(L(\underline{p}_{ges}, \underline{d}_{ges})) \\ \sum_i^m d_m \end{bmatrix} \\ \text{s.t. } \underline{p}_{ges} - \underline{p}_{ges,max} \leq 0 \\ -\underline{p}_{ges} + \underline{p}_{ges,min} \leq 0 \end{aligned} \quad (10)$$

Diese Art von Optimierungsproblem wird in [VAN99, LAM00] als mehrkriterielles Optimierungsproblem (MKOP) oder *Pareto Optimierung* bezeichnet. Im nächsten Schritt gilt es geeignete Algorithmen zur Lösung dieser Problemklasse zu identifizieren.

5.3.3 Algorithmen-Wahl für Pareto-Optimierung

In [VAN99] werden globale Such- und Optimierungsansätze in die drei Kategorien enumerativ, deterministisch und stochastisch eingeteilt. Obwohl eine enumerative Suche deterministisch ist, differenziert Van Veldhuizen hier, da diese Suchmethode keine Heuristik verwendet. Die beschriebene Klassifizierung ist mit relevanten Vertretern der jeweiligen Kategorien in Abbildung 34 zusammengefasst. Jeder dieser Ansätze besitzt nach Lampinen [LAM00] Vorteile, aber auch grundlegende Einschränkungen. Der enumerative ist der am wenigsten effiziente, aber durch die Betrachtung aller möglichen Lösungen ein robuster Optimierungsansatz. Durch die Betrachtung aller möglichen Lösungen wird das globale Optimum garantiert gefunden. Hier wird jedoch schnell die Schwäche des Ansatzes deutlich. Eine Aufzählung oder Betrachtung aller Lösungen ist nur bei diskreten Problemen effektiv, da hier ein endlicher Lösungsraum vor-

liegt. Auch kontinuierliche Probleme können enumerativ gelöst werden, hier ist jedoch zunächst eine Diskretisierung des Suchraums nötig. Nach der Umwandlung eines kontinuierlichen in ein diskretes Problem nimmt die Auflösung der Lösung umgekehrt proportional zur Anzahl der diskreten Punkte ab, die den (ursprünglich kontinuierlichen) Entwurfsraum repräsentieren. In der Praxis kann die Aufzählung aufgrund der schnell ansteigenden kombinatorischen Möglichkeiten und der daraus resultierenden Zeitkomplexität nur bei kleinen diskreten Designdomänen angewandt werden, d. h. bei wenigen Variablen, die lediglich mit einer geringen Auflösung diskretisiert werden [LAM00]. In [KW07] wird dem enumerativen Ansatz (hier als *Brute Force* bezeichnet) die Eignung für realistische Optimierungsprobleme aberkannt und mit dem, von Richard Bellmann geprägten Begriff, „*Fluch der Dimensionalität*“ argumentiert. Nach Verleysen und François ist der Fluch der Dimensionalität der Ausdruck für alle Phänomene, die bei hochdimensionalen Daten auftreten und meist unvorteilhafte Auswirkungen auf das Verhalten und die Leistung von Lernalgorithmen haben [VF05].

Deterministische Optimierungsansätze sind nach [LAM00] auf eine kleine Teilmenge der allgemeinen MKOP beschränkt. Die Verwendung von deterministischen Algorithmen basiert typischerweise auf Annahmen, beispielsweise hinsichtlich der Eigenschaften der Zielfunktion, des Optimierungsproblems oder des Suchraumes. Wenn diese Annahmen nicht erfüllt sind, ist ein Fehlschlagen des Optimierungsalgorithmus wahrscheinlich. Deterministische Algorithmen beruhen nach [LAM00] des Weiteren auf einer Reihe von Annahmen, wie z. B. das Vorliegen eines linearen Problems in analytischer Form oder dem Ausbleiben von Wechselwirkungen zwischen Parametern. In Anbetracht realitätsnaher MSM gelten für das in Gleichung (10) beschriebene Optimierungsproblem die meisten der formulierten Annahmen nicht. So ist f_2 beispielsweise unstetig und f_1 als nichtlinear anzunehmen. Wenn mindestens eine der Zielfunktionen oder der Randbedingungen nichtlinear ist, ist Gleichung (10) ein nichtlineares Problem [CP07]. Zusammengefasst ist die Klasse der deterministischen Algorithmen für das in Gleichung (10) beschriebene Optimierungsproblem ebenfalls nicht zu bevorzugen.

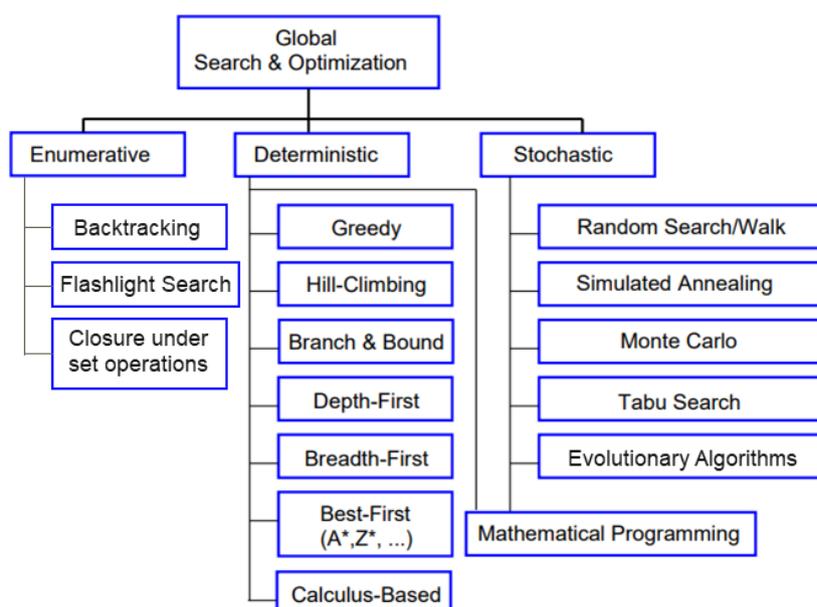


Abbildung 34: Einteilung globaler Such- und Optimierungsalgorithmen nach [VAN99] mit populären Vertretern.

Im Bereich der stochastischen Verfahren werden evolutionäre Algorithmen (engl. *evolutionary algorithms*) routinemäßig bei der Lösung von Pareto-Optimierungsproblemen in verschiedenen Zweigen der

Technik, Wissenschaft und Wirtschaft eingesetzt [ZT99, DEB15]. Die zunehmende Akzeptanz dieser Algorithmen ist nach [SUM04] auf ihre Fähigkeit zurückzuführen, mehrere Lösungen in einem einzigen Durchlauf zu finden, ohne Ableitungen zu arbeiten, zu Pareto-optimalen Lösungen zu konvergieren und sowohl für kontinuierliche Funktions- als auch kombinatorische Optimierungsprobleme genutzt werden zu können. Darüber hinaus sind evolutionäre Algorithmen weniger anfällig für die Form oder Kontinuität der Pareto-Front. Der Einsatz bei diskontinuierlichen oder konkaven Pareto-Fronten ist möglich, während diese beiden Aspekte für mathematische Programmieretechniken ein Problem darstellen [COE01]. Für die Implementierung der betriebsbegleitenden Adaption der Co-Simulation werden daher im Folgenden evolutionäre Algorithmen gewählt.

5.3.4 Grundlagen evolutionärer Algorithmen

Bei evolutionären Algorithmen dient die natürliche Evolution als Vorbild für die Lösung von Optimierungsproblemen. Diese Algorithmen haben nach Weicker [WEI15] gemeinsam, dass sie *„Vorgänge und Begriffe aus der Biologie entlehnen, um in einer simulierten Evolution (als Wechselspiel zwischen Modifikation und Auswahl besserer Individuen) möglichst gute Näherungslösungen für eine gegebenes Problem zu entwickeln.“* Für ein besseres Verständnis ist es sinnvoll, zunächst die wesentlichen Begriffe einzuführen. Im weiteren Verlauf dieser Arbeit sind die Begriffe *„Individuum“* und *„Population“* sowie die Evolutionsfaktoren [WEI15] *„Selektion“*, *„Mutation“* und *„Kreuzung“* relevant.

Der Kerngedanke, der allen evolutionären Algorithmen zugrunde liegt, ist nach Eiben und Smith [ES15] derselbe: Bei einer Population von Individuen in einer Umgebung mit begrenzten Ressourcen führt der Wettbewerb um diese Ressourcen zu einer natürlichen Auslese (Überleben des Stärkeren). Dies wiederum führt zu einem Anstieg der Leistungsfähigkeit (auch *„Fitness“* bezeichnet) der Population. Bei einer zu maximierenden Qualitätsfunktion wird zunächst nach dem Zufallsprinzip eine Reihe von Lösungsvorschlägen (Individuen) erstellt. [ES15] Im Rahmen dieser Arbeit ist die zu maximierende Qualitätsfunktion $F(\underline{p}_{ges}, \underline{d}_{ges})$ durch Gleichung (10) beschrieben. Per Zufall werden in diesem Fall der Vektor der Parameter \underline{p}_{ges} und der Vektor der LoD \underline{d}_{ges} gewählt. In Abbildung 35 ist dieser Vorgang exemplarisch für das erste und das n -te Individuum einer Population aufgezeigt. In diesem Beispiel besteht das zu adaptierende Gesamtmodell aus drei MSM. Das erste und dritte MSM besitzen jeweils zwei, das zweite MSM drei Parameter. Der evolutionäre Algorithmus kann für die Parameter reelle Werte zwischen 0 und 1 mit einer Nachkommastelle wählen. Die Wahl der Parametergrenzen und deren Genauigkeit dient in diesem Beispiel dazu, die Parameter in Abbildung 35 leicht von den LoD differenzieren zu können. Für jedes der MSM kann der Algorithmus zudem noch den LoD zwischen 1 und 2 wählen. Die einzelnen Einträge sind als Gene des Individuums aufzufassen. Die Gesamtheit aller Gene eines Organismus wird als Genom bezeichnet [WEI15].

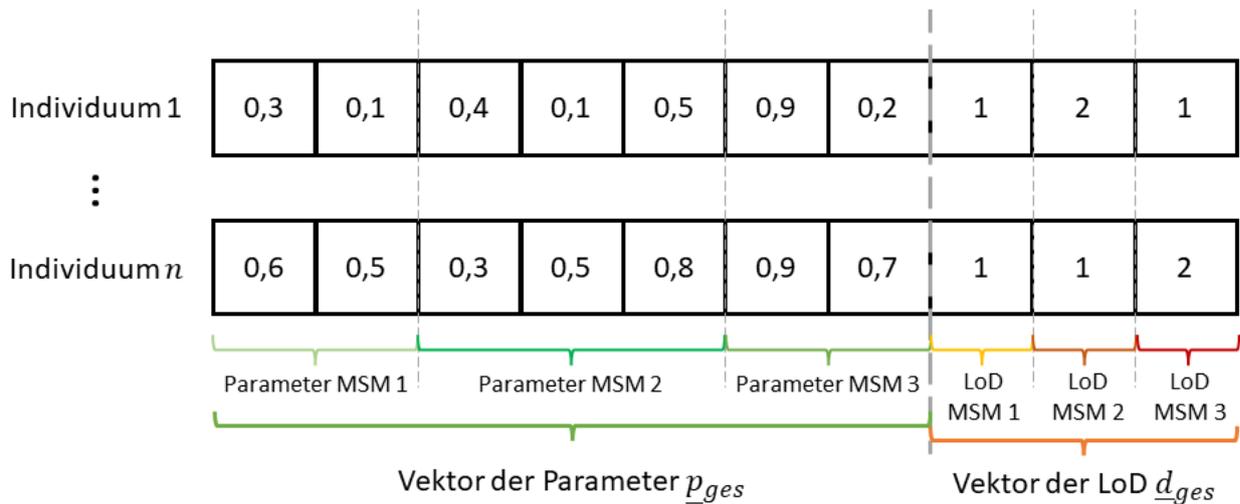


Abbildung 35: Exemplarisches zufallsgeneriertes Genom von zwei Individuen mit 7 Parametern und 3 MSM.

Weiter stellt nach Eiben und Smith die Menge der zufallsgenerierten Lösungsvorschläge die initiale Population dar. Auf jeden einzelnen Lösungsvorschlag wird die Qualitätsfunktion als abstraktes Fitnessmaß angewandt. Auf Grundlage dieser Fitnesswerte werden einige der besseren Kandidaten ausgewählt, um die nächste Generation zu erzeugen. Dies geschieht durch Anwendung von Kreuzung und/oder Mutation auf die bestehenden Kandidaten. Kreuzung ist ein Operator, der auf zwei oder mehr ausgewählte Kandidaten (die sogenannten Eltern) angewandt wird und einen oder mehrere neue Kandidaten (die Kinder) hervorbringt. Hierzu werden Teile der Eltern-Genome kombiniert. Die Mutation wird auf einen Kandidaten angewandt und führt zu einem neuen Kandidaten. Hierbei werden zufällig einzelne Gene im Genom variiert. Die Ausführung der Operationen Kreuzung und Mutation an den Eltern führt zur Erzeugung einer Reihe neuer Kandidaten (den Nachkommen). Diese werden auf ihre Fitness hin bewertet und konkurrieren dann, basierend auf ihrer Fitness, mit den alten Kandidaten um einen Platz in der nächsten Generation. Dieser Prozess kann so lange wiederholt werden, bis ein oder mehrere Kandidaten mit ausreichender Qualität gefunden werden oder eine zuvor festgelegte Berechnungsgrenze erreicht ist. Eiben und Smith identifizieren zwei Hauptkräfte, die die Grundlage für evolutionäre Systeme bilden. Zum einen schaffen die Evolutionsfaktoren Kreuzung und Mutation die notwendige Vielfalt innerhalb der Population und ermöglichen dadurch Neuartigkeit. Zum anderen wirkt die Selektion als Kraft, die die durchschnittliche Qualität der Lösungen in der Population erhöht. [ES15]

Der beschriebene Evolutionsprozess ist in Abbildung 36 schematisch dargestellt.

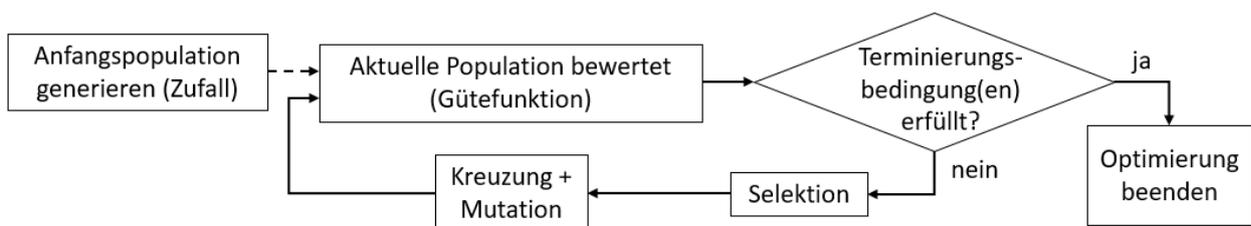


Abbildung 36: Schematische Darstellung der iterativen Optimierung mittels evolutionärer Algorithmen in Anlehnung an [WE15].

Nachdem die konzeptionellen Grundlagen geschaffen sind, wird im nächsten Abschnitt die prototypische Implementierung des Lösungskonzepts vorgestellt.

5.4 Prototypische Implementierung

Die prototypische Implementierung der Funktionen für die betriebsparallele Modelladaption erfolgt in vier Schritten. In Abschnitt 5.4.1 wird aufgezeigt, welche Lösungen bereits für das Lösen von MKOP mittels evolutionärer Algorithmen existieren, und ein geeigneter Kandidat ausgewählt. Abschnitt 5.4.2 befasst sich mit der Integration des formulierten MKOP, dem Setup von Algorithmen sowie möglichen Analysefunktionen. In Abschnitt 5.4.3 wird anschließend der Adaptionprozess entsprechend der Vorlage aus Abschnitt 5.2.3 umgesetzt. Abschnitt 5.4.4 behandelt abschließend, wie die in Abschnitt 4.4.2 generierten Simulationsskripte im Rahmen der Adaption angepasst werden.

5.4.1 Auswahl und Einbindung einer geeigneten Algorithmen-Bibliothek

Das bestehende AS soll um die in Abschnitt 5.3 beschriebene Möglichkeit der betriebsparallelen Modelladaption erweitert werden. Hierfür bietet es sich an, etablierte Bibliothekselemente zu verwenden, die die notwendigen Funktionalitäten bereitstellen. Zu diesen Funktionalitäten gehören beispielsweise die unkomplizierte Integration der formulierten Adaptionaufgabe als MKOP sowie die Bereitstellung verschiedener EA für die Lösung der Optimierungsprobleme im Rahmen der Konzeptvalidierung. Unter anderem werden Opt4J [LGR+11], ECJ [SL19], JENETICS [WIL22[®]], JMetal [NDV15] und das MOEA Framework [MOE22[®]] als potenziell geeignete Kandidaten identifiziert. Für die prototypische Implementierung fällt die Wahl auf die Open-Source-Bibliothek MOEA Framework. Diese zeichnet sich durch eine lange Entwicklungsgeschichte, eine umfangreiche API inklusive Dokumentation und ihre Benutzerfreundlichkeit aus. Eigene Optimierungsprobleme können schnell formuliert werden und es gibt vordefinierte Methoden zum Vergleich und zur Analyse von Algorithmen. Das MOEA Framework beinhaltet zudem die Algorithmen der JMetal-Bibliothek und ergänzt diese noch um zusätzliche Algorithmen. So steht eine umfangreiche Algorithmen-Bibliothek für die Konzeptvalidierung zur Verfügung. Sollte sich ein EA für ein bestimmtes Problem als nicht geeignet erweisen (z. B. gut geeignet für Jentner-Anlage, ungeeignet für OTEC-Anlage), kann die Adaption ohne großen Aufwand mit einem anderen Algorithmus durchgeführt werden. Im nächsten Abschnitt wird im Detail aufgezeigt, wie ein Optimierungsproblem formuliert und das Setup der Algorithmen im Betrieb durchgeführt werden können und welche Analysefunktionen zur Verfügung stehen.

5.4.2 Problemformulierung, Algorithmen-Setup und Analysefunktionen

Hadka [HAD16] beschreibt in seiner Arbeit zunächst, wie ein eigenes Problem im MOEA Framework definiert werden kann. Hierfür gibt es zwei Möglichkeiten. Entweder kann die eigene Problem-Klasse das Interface „*Problem*“ implementieren oder alternativ kann die Klasse „*AbstractProblem*“ erweitert werden. Die zweite Variante schätzt Hadka als einfacher ein, daher wird diese für die Implementierung genutzt. Bei der Erweiterung der abstrakten Klasse müssen der Konstruktor und die zwei Methoden „*newSolution*“ und „*evaluate*“ implementiert werden. Zunächst wird der Aufbau des Konstruktors und die Schnittstelle zum Bediener der AS aufgezeigt. Anschließend werden die beiden Methoden im Detail vorgestellt.

5.4.2.1 Problemformulierung und Setup der Adaptionaufgabe durch Bediener

Dem Konstruktor werden die Anzahl an Designvariablen sowie die Anzahl an Zielfunktionen als Parameter übergeben. Der Aufbau des Konstruktors der für den Adaptionprozess formulierten Problem-Klasse „CustomProblem“ ist in Listing 2 dargestellt.

```
1 public class CustomProblem extends AbstractProblem
2 {
3     public CustomProblem(int numberOfVariables, int numberOfObjectives)
4     {
5         super(numberOfVariables, numberOfObjectives);
6     }
7 }
```

Listing 2: Konstruktor der Klasse CustomProblem

In Zeile 3 von Listing 2 werden dem Konstruktor die Parameter als Variablen übergeben. Die Anzahl an Zielfunktionen „numberOfObjectives“ könnte direkt mit dem Integer-Wert 2 vorgegeben werden. Die Implementierung mit einer Variablen ermöglicht es jedoch, bei Bedarf die Adaptionaufgabe zukünftig um weitere Zielfunktionen zu erweitern. Die Anzahl an Designvariablen „numberOfVariables“ fasst die Parameter und LoD aller im Anwendungsfall betrachteten MSM zusammen. Der Bediener des AS gibt die Designvariablen durch eine GUI vor. Über das Schlüsselwort „super“ wird der Konstruktor der Klasse „AbstractProblem“ aufgerufen und ein unbeschränktes abstraktes Problem mit der angegebenen Anzahl an Designvariablen und Zielfunktionen erzeugt.

Eine mögliche Implementierung der beschriebenen GUI zur Vorgabe der Designvariablen ist in Abbildung 37 dargestellt. Im konkreten Fall wird die aus zwei MSM bestehende OTEC Anlage betrachtet. Die MSM, deren Parameter und LoD werden in Abschnitt 5.5 im Detail vorgestellt. Da bereits zur Inbetriebnahme-phase das TM der Anlage in das AS geladen wurde (vgl. Kapitel 4), sind dem AS die Anzahl an MSM und deren Bezeichner bekannt. Darüber hinaus ist es möglich aus dem TM die Parameter einschließlich deren Initial- und Grenzwerte sowie die LoD der MSM zu extrahieren. Mit diesen automatisch ausgelesenen Informationen kann das AS die GUI aufbauen. In Abbildung 37 sind, markiert durch die dunkelblauen Pfeile, die beiden Module *Bearbeitungsbehälter* und *Bearbeitungsstation* aufgeführt. Für jedes MSM werden die Parameter separat aufgelistet. Das Modul *Bearbeitungsbehälter* besitzt in diesem Fall einen Parameter, das Modul *Bearbeitungsstation* besitzt drei Parameter. In den vier Spalten (durchnummeriert mit grünen Markern) steht zunächst der Name des Parameters, sein Startwert bzw. die initiale Schätzung, sofern noch keine Adaption durchgeführt wurde, sowie der minimale und maximale Grenzwert. Über die Checkboxen links neben jedem Parameter (grün umrandet) hat der Bediener die Möglichkeit, die Parameter einzeln für die Adaption zu berücksichtigen oder zu sperren. Bei einer Sperrung, dargestellt durch das Ausgrauen der Schaltflächen für die Grenzwerteingabe, werden die Parameter mit ihrem aktuellen Wert als fix gesetzt. Die in Abbildung 37 dargestellte Einstellung würde bedeuten, dass für das Modul *Bearbeitungsbehälter* kein Parameter und für das Modul *Bearbeitungsstation* die zwei Parameter „Eintauchtiefe WS“ und „Schleifmittel-Dichte“ innerhalb der vorgegebenen Grenzwerte beliebig durch den EA variiert werden können. Zusätzlich ist unter der Parameternaufzählung noch pro MSM der maximal zu verwendende LoD (orange umrandet) einstellbar. In der in Abbildung 37 dargestellten Konfiguration kann der EA im Rahmen der Adaption den LoD 1 oder LoD 2 für das Modul *Bearbeitungsbehälter* wählen. Beim Modul *Bearbeitungsstation* ist der LoD 1 fix durch den Bediener vorgegeben. Hier kann der EA keine Än-

derungen vornehmen. Die Vorgabe eines zu hohen LoD durch den Bediener wird vom AS verhindert, indem die referenzierte FMU-Bibliothek analysiert wird. Entsprechend dem Setup in Abbildung 37 umfasst der Vektor mit Designvariablen zwei Parameter und einen zu optimierenden LoD. Dem Konstruktor der CustomProblem-Klasse aus Listing 2 würde für die Variable `numberOfVariables` entsprechend der Wert drei übergeben werden. Über den Button „File“ oben links im Fenster kann abschließend die Datei mit den Messdaten in das AS geladen werden.

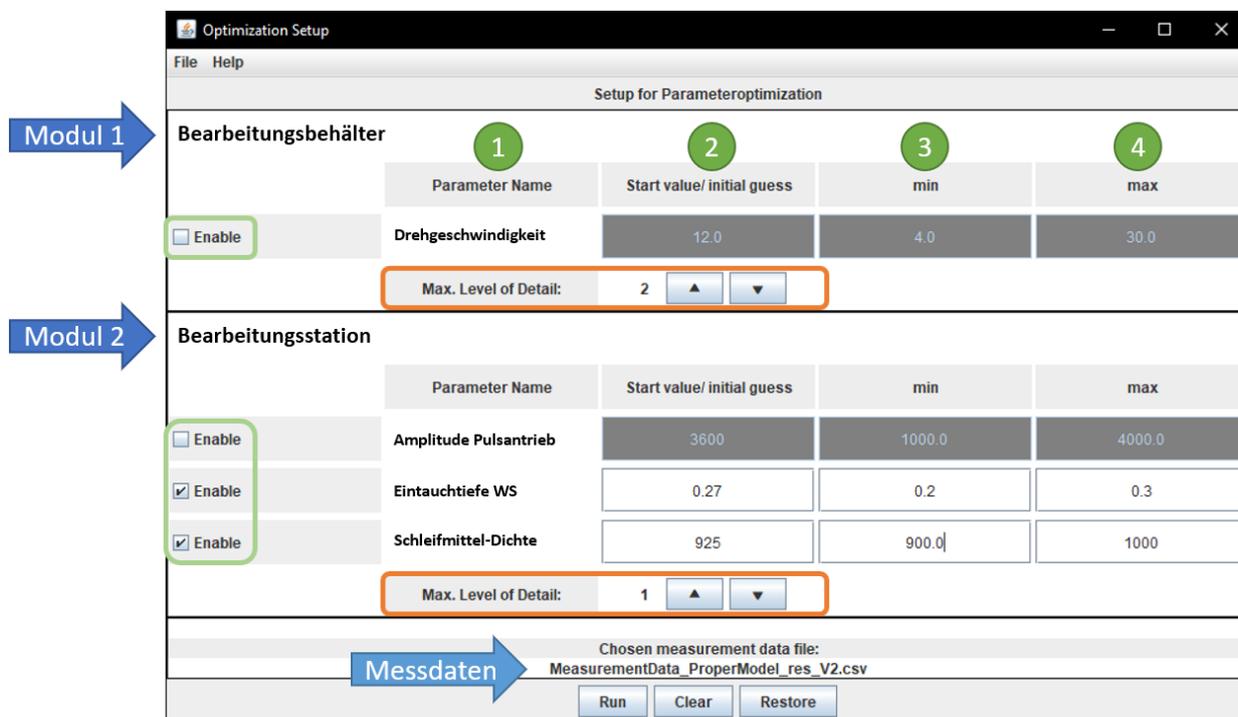


Abbildung 37: GUI für das Setup der Adaptionaufgabe für die OTEC Anlage.

Nach der Beschreibung des Konstruktors und dem Setup des Optimierungsproblems liegt der Fokus auf den beiden zu implementierenden Methoden.

5.4.2.2 Erzeugung und Validierung neuer Lösungen für das Optimierungsproblem

Die `newSolution`-Methode erzeugt beim Aufruf eine neue, zufällig initialisierte Lösung. Der EA nutzt diese Methode im Rahmen der Generierung der initialen Population. Jede Lösung stellt, innerhalb der vorgegebenen Grenzen, eine zufällige Konstellation des Vektors der Designvariablen dar. Die Implementierung der `newSolution`-Methode ist in Listing 3 dargestellt. Hierbei wird die Methode der Oberklasse überschrieben. Zunächst wird in Zeile 4 eine neue Lösung `sol` initialisiert. Dieser wird, wie schon dem Konstruktor der Problem-Klasse, die Anzahl an Designvariablen und Zielfunktionen übergeben. Anschließend werden innerhalb der Zeilen 6 bis 24 alle Designvariablen durchlaufen. Der Designvektor wurde so gewählt, dass zuerst alle Parameter der MSM in der Reihenfolge, wie sie im TM auftreten, gelistet werden und anschließend entsprechend alle LoD. Ein anderer Aufbau des Designvektors hinsichtlich der Reihenfolge ist ebenfalls denkbar. Über eine `if`-Abfrage wird entschieden, ob es sich bei der jeweiligen Designvariable um einen Parameter oder einen LoD handelt. Die Unterscheidung ist dahingehend relevant, da die LoD im natürlichen Zahlenraum definiert wurden und daher durch Integer-Werte repräsentiert werden.

Die Parameter werden der Einfachheit halber alle durch Double-Werte repräsentiert. In realen Applikationen sind auch andere Datentypen für die Parameter denkbar, z. B. ein boolescher Wert für einen Schalter. Der Datentyp der Parameter ist ebenfalls im TM abgebildet. Die Umsetzung einer entsprechenden Abfrage ist leicht möglich, würde an dieser Stelle jedoch die Lesbarkeit von Listing 3 reduzieren. Über Listen wird anschließend auf die entsprechenden Parameter und LoD zugegriffen, vgl. Zeile 10 und Zeile 18. In den Zeilen 14 und 19 werden letztlich die Werte der Parameter und LoD innerhalb der vorgegebenen Grenzwerte randomisiert festgelegt. Hierzu werden die Methoden `EncodingUtils.newReal` bzw. `EncodingUtils.newInt`, die das MOEA Framework zur Verfügung stellt, aufgerufen. In Zeile 23 wird abschließend die Lösung `sol` zurückgegeben. Diese zufällig generierte Lösung kann im nächsten Schritt durch den EA hinsichtlich der Gütefunktionen bewertet werden.

```

1  @Override
2  public Solution newSolution()
3  {
4      Solution sol = new Solution(getNumberOfVars(), getNumberOfObjectives());
5
6      for (int i = 0; i < getNumberOfVars(); i++)
7      {
8          if (i < getNumberOfVars() - numberOfModulesWithMultipleLevelsOfDetail)
9          {
10             Parameter currentParameter = parameterList.get(i);
11             double currentMin = currentParameter.getMinValue();
12             double currentMax = currentParameter.getMaxValue();
13
14             sol.setVar(i, EncodingUtils.newReal(currentMin, currentMax));
15         }
16         else
17         {
18             String currentLoD = lodList.get(i - (getNumberOfVars() - ..
19                 numberOfModulesWithMultipleLevelsOfDetail)).getLoD();
20             sol.setVar(i, EncodingUtils.newInt(1, ..
21                 Integer.parseInt(currentLoD)));
22         }
23     }
24     return sol;
25 }

```

Listing 3: Implementierung der Methode `newSolution` mittels welcher der EA neue Lösungen für das MKOP erzeugt.

Die Berechnung der Güte einer Lösung erfolgt innerhalb der `evaluate`-Methode. Diese ist in Auszügen in Listing 4 dargestellt. Zunächst wird der Methode eine Lösung als Parameter übergeben. In den Zeilen 4 und 5 werden entsprechend der Designvariablen zwei Vektoren `x_double` und `x_int` initialisiert. Die Designvariablen werden anschließend in die Vektoren eingetragen. Dieser Codeabschnitt ist zugunsten einer leichteren Lesbarkeit von Listing 4 nicht dargestellt. Anschließend werden in den Zeilen 7 und 8 zwei boolesche Werte angelegt (`simulationDone`, `calculateDeltaDone`), über die abgebildet werden kann, ob die Simulation mit den neuen Designvariablen bereits abgeschlossen ist und ob die Abweichung zwischen den Mess- und Simulationsdaten berechnet wurde. In den Zeilen 9 und 10 werden die beiden Werte der Gütefunktionen initialisiert. Anschließend erfolgt zunächst die Berechnung der f_2 -Güte (Summe der Detaillierungsgrade) in den Zeilen 13 bis 17. Hierfür ist es nicht nötig, die Simulation durchzuführen, da diese Gütefunktion nur von den gewählten Detaillierungsgraden abhängt. Diese können aus der Variable `x_int` ausgelesen werden. Anschließend wird in der Zeile 20 die Simulation mit den neuen Designvariablen durchgeführt. Wie die Simulationsskripte hierfür angepasst werden müssen, ist in Abschnitt 5.4.4 im Detail beschrieben. Die Durchführung der Simulation und die Berechnung der Abweichung erfolgt in einer

anderen Funktion. Sobald die beiden Funktionen `runSimScript` und `calculateDelta` abgeschlossen sind, werden die beiden booleschen Werte `simulationDone` und `calculateDeltaDone` auf `true` gesetzt. Da diese Berechnungen, je nach Simulationszeit und Größe der Datensätze, eine gewisse Zeit benötigen, wartet die Methode `evaluate`, bis beide Schritte sicher erfolgt sind. Sofern dies sichergestellt ist, wird in Zeile 31 die f_1 -Güte berechnet. Die Methode `objectiveOne` ist eine Implementierung des Akaike information criterion, welches in Abschnitt 5.3.2 durch Gleichung 6 beschrieben wurde. Abschließend werden die berechneten Gütewerte der Lösung übergeben.

```

1  @Override
2  public void evaluate(Solution solution)
3  {
4      double[] x_double = new double[getNumberOfVars()-numberOfModulesWithMultipleLevelsOfDetail];
5      int[] x_int = new int[numberOfModulesWithMultipleLevelsOfDetail];
6
7      boolean simulationDone = false;
8      boolean calculateDeltaDone = false;
9      double f1 = 0.0;
10     int f2 = 0;
11
12     // Funktionsauswertung für f2
13     for (int j = 0; j < x_int.length; j++)
14     {
15         f2 += x_int[j];
16     }
17     f2 += numberOfModulesWithoutDifferentLoDs;
18
19     // Skript updaten und Simulation ausführen
20     opt.updateSimScript(updatedParameterList, updatedLoDList);
21     updatedParameterList.clear();
22     updatedLoDList.clear();
23
24     simulationDone = opt.runSimScript();
25     calculateDeltaDone = opt.calculateDelta();
26
27     while (true) {
28         if (simulationDone && calculateDeltaDone)
29         {
30             // Funktionsauswertung für f1
31             f1 = opt.objectiveOne(opt.getCurrentDelta());
32
33             // Funktionswerte für Lösung setzen
34             solution.setObjective(0, f1);
35             solution.setObjective(1, f2);
36
37             break;
38         }
39     }
40 }

```

Listing 4: Implementierung der Methode `evaluate` mittels welcher der EA die Güte einer Lösung berechnen kann.

Die im Rahmen dieser Arbeit implementierte `CustomProblem`-Klasse ist in Anhang C vollständig dargestellt.

5.4.2.3 Setup des evolutionären Algorithmus durch Bediener

Nach der Definition des Problems, der Struktur der Lösungen und wie diese hinsichtlich der Gütefunktionen bewertet werden können, kann im nächsten Schritt das Problem gelöst werden. Hierfür wird im MOEA Framework die „`Executor`“-Klasse verwendet. Der `Executor` ist verantwortlich für die Erstellung von Instanzen von Optimierungsalgorithmen und deren Verwendung zur Problemlösung. Die Klasse lässt eine

komplexe Konfiguration zu, beispielsweise kann eingestellt werden, dass das Problem nacheinander mit mehreren Algorithmen berechnet wird. Im Minimalfall benötigt die Klasse mindestens drei Informationen. In Listing 5 ist ein Minimalbeispiel zur Verwendung des Executors dargestellt. In Zeile 5 wird ein Objekt `result` der Klasse `NondominatedPopulation` aus dem MOEA Framework erzeugt. Hierbei handelt es sich um eine Menge nicht-dominierter Lösungen. Die Bedeutung dieser nicht-dominierten Lösungen und wie sie berechnet werden wird im nächsten Abschnitt im Detail aufgezeigt. Kurz zusammengefasst handelt es sich hierbei um die Lösungen des MKOP nach Erreichen des vorgegebenen Abbruchkriteriums. Das Objekt `result` wird durch die Erzeugung, Parametrierung und Ausführung eines Objekts der Executor-Klasse generiert. Als Parameter werden dem Executor-Objekt in den Zeilen 6 bis 9 der Algorithmus, die Problemklasse und mindestens ein Abbruchkriterium übergeben. In diesem Beispiel sind das der Standard-Algorithmus `NSGAI`, die `CustomProblem`-Klasse aus Listing 2 und 1000 Lösungsauswertungen, nach denen der EA abbricht. Abschließend können die Ergebnisse über einen Plot für den Bediener visualisiert werden.

```
1 public class TestOptimization
2 {
3     public static void main(String[] args)
4     {
5         NondominatedPopulation result = new Executor()
6             .withAlgorithm("NSGAI")
7             .withProblemClass(CustomProblem.class)
8             .withMaxEvaluations(1000)
9             .run();
10
11         new Plot()
12             .add("NSGAI", result)
13             .show();
14     }
15 }
```

Listing 5: Minimalbeispiel zur Verwendung des Executors.

Das Setup für den Executor bzw. den EA erfolgt ebenfalls durch Bedienereingaben mittels einer GUI im AS. Eine mögliche Implementierung der GUI-Maske ist in Abbildung 38 dargestellt. Zunächst muss der Bediener im oberen Teil der Maske die maximal zulässige Abweichung Δ_{max} vorgeben, bevor das AS eine Adaption einleitet. Hierfür können keine Standardwerte vorgegeben werden. Dieser Wert ist stark vom Anwendungsfall abhängig, z. B. wie viele virtuelle und reale Sensoren über welchen Zeitraum verglichen werden und welche Modellabweichungen tolerierbar sind. Dies lässt keine ad-hoc Aussagen zu und setzt beim Bediener ein gewisses Maß an Systemverständnis sowie empirische Erkenntnisse voraus. Im nächsten Abschnitt in Abbildung 38 kann der Bediener das Setup des EA vornehmen. Hierbei handelt es sich um eine vereinfachte Darstellung, da die Algorithmen noch mehr Parameter, z. B. hinsichtlich der Abbruchkriterien, besitzen und nicht alle Algorithmen über die gleichen Parameter eingestellt werden. In dieser Beispielmaske kann der Bediener über ein Dropdown-Menü den EA auswählen und anschließend auf der rechten Seite die Populationsgröße, die maximale Anzahl an Generationen und die Mutationsrate vorgeben.

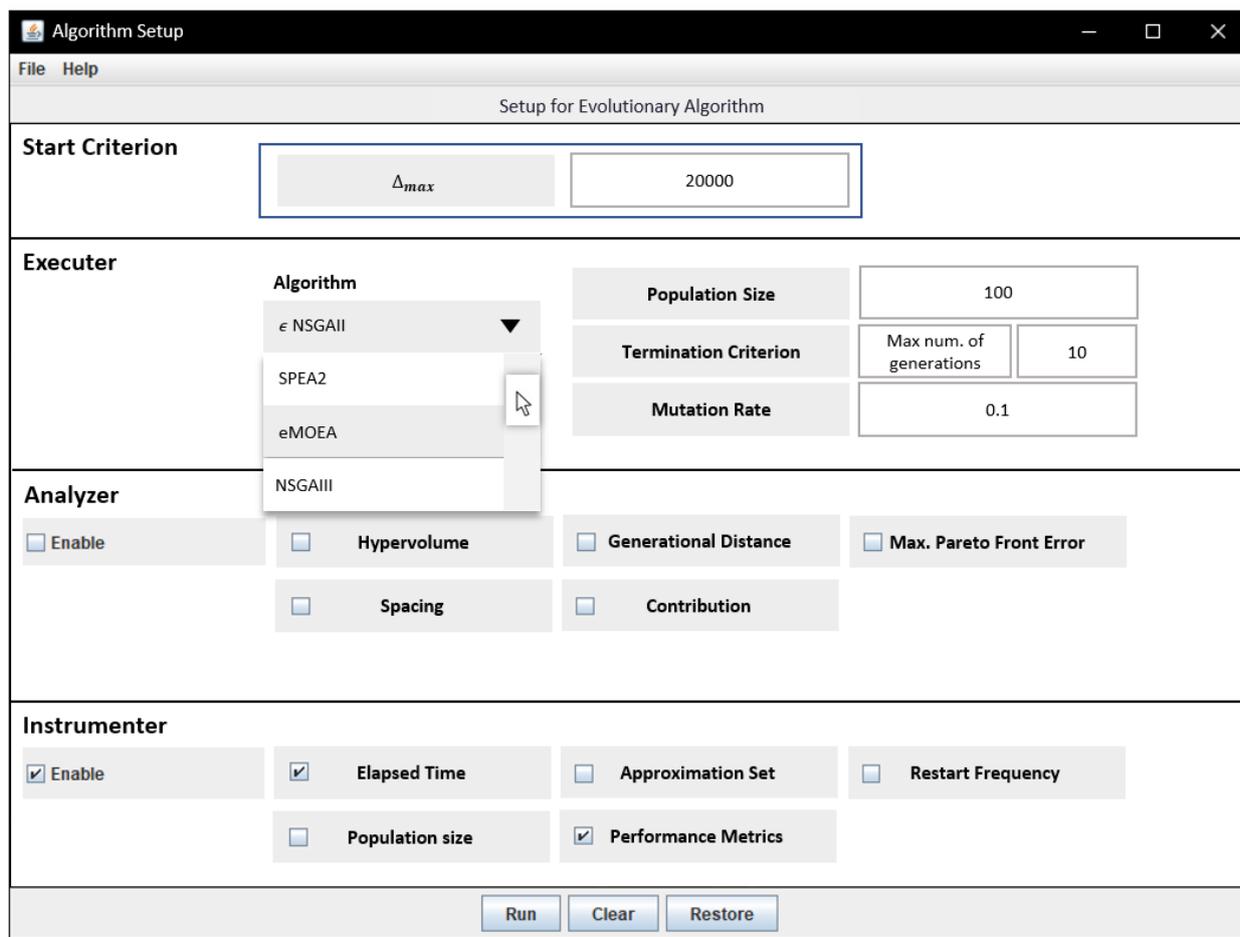


Abbildung 38: GUI für das Setup des EA sowie weiterer Analysefunktionen.

In der unteren Hälfte von Abbildung 38 können noch weitere Einstellungen, z. B. zur Laufzeitanalyse des gewählten EA, vorgenommen werden. Diese werden an dieser Stelle der Vollständigkeit halber kurz vorgestellt, werden aber nicht für die Erfüllung der Anforderungen benötigt. Für die Umsetzung des AS in einer industriellen Applikation sowie für weitere Forschung am AS und dessen Potenzial im Rahmen der automatischen Modelladaption können diese Funktionen allerdings von Bedeutung sein.

5.4.2.4 Optionale Analysefunktionen für evolutionäre Algorithmen

Das MOEA Framework bietet einerseits die „Analyzer“-Klasse, die in der Regel in Verbindung mit der Executor-Klasse verwendet wird, um die Leistung von Algorithmen zu analysieren. Beispielsweise kann der Analyzer genutzt werden, um die Leistung von zwei oder mehr Algorithmen statistisch zu vergleichen oder um festzustellen, ob es einen statistischen Unterschied zwischen den beiden Algorithmen gibt. Um Informationen über die Laufzeitdynamik zu erhalten, kann andererseits die „Instrumenter“-Klasse genutzt werden. Diese Klasse wird ebenfalls in Verbindung mit der Executor-Klasse verwendet. Es kann die Frequenz vorgegeben werden, mit der Informationen aufgezeichnet werden, beispielsweise alle 100 Auswertungen. Hierüber können nach Durchlauf der Optimierung z. B. Aussagen über die benötigte Zeit und Populationsgrößen getroffen und Leistungsmetriken erstellt werden. Eine weitere nützliche Funktion ist beispielsweise, dass man berechnete Pareto-Fronten speichern und für zukünftige Analysen als Referenz-Set speichern kann. So kann z. B. ein neuentwickelter Algorithmus gegen etablierte Algorithmen getestet werden oder eine Datenbank mit Pareto-Fronten aufgebaut werden.

Nach der Vorstellung der Problemimplementierung, des Algorithmen-Setups sowie möglicher Analysefunktionen soll im nächsten Abschnitt dargestellt werden, wie der in Abschnitt 5.2.3 aufgezeigte Adaptionsprozess umgesetzt werden kann.

5.4.3 Umsetzung des Adaptionsablaufs

Unter der Annahme, dass eine Anpassung notwendig ist, wird zunächst eine initiale Lösungspopulation mit einer vorgegebenen Anzahl an Individuen und jeweils zufälligen Genomen generiert. Die Wahl der Populationsgröße sowie weiterer Parameter des verwendeten EA, welche beispielsweise die Mutation und Selektion steuern, stellt eine Herausforderung dar. Lu und Yen [LY02] beschreiben in ihrer Arbeit, dass es bei evolutionären Algorithmen zu einer vorzeitigen Konvergenz hin zu einem lokalen Optimum kommen kann, wenn die Populationsgröße zu klein gewählt wird. Die Wahl einer zu großen Population führt hingegen zu einem unerwünscht hohen Berechnungsaufwand verbunden mit einer langen Wartezeit. Da die Pareto-Front eines MKOP a priori oft nicht bekannt ist, ist es schwierig, die optimale Anzahl von Individuen abzuschätzen, die für eine effektive Erkundung des Lösungsraums und eine gute Darstellung der Pareto-Front erforderlich ist. [LY02]

Die größte Herausforderung für die Anwender evolutionärer Algorithmen besteht nach Eiben und Smith [ES15] demnach darin, dass die Wahl der Parameterwerte (wie Populationsgröße und Mutationsrate) einen so großen Einfluss auf die Leistung des Algorithmus hat. Die Parametrierung von EAs stellt daher selbst ein Optimierungsproblem dar [ES15]. Sipper et al. kommen in ihrer Studie zu dem Schluss, dass sich gute Parameter über das ganze Parameterspektrum erstrecken und in der gegenwärtigen Praxis Parameterwerte meist durch Konventionen, ad-hoc-Entscheidungen und sehr begrenzte experimentelle Vergleiche ausgewählt werden [SFA+18]. Es gibt daher bereits Ansätze, die sich mit selbstanpassenden evolutionären Algorithmen, z. B. in [RW22], beschäftigen. Im Weiteren wird die initiale Lösungspopulation exemplarisch mit 100 Individuen angenommen, da es sich hierbei um die Standardeinstellung der meisten EAs im MOEA Framework handelt.

Wie in Abbildung 35 aufgezeigt, setzt sich das Genom von jedem Individuum durch die Wahl der Modellparameter und LoD innerhalb vorgegebener Grenzen zusammen. Auf Grundlage dieser Konfigurationen werden im nächsten Schritt neue Simulationsskripte erstellt. Der nächste Abschnitt befasst sich im Detail mit der Abbildung der Adaption im Simulationsskript. Für 100 Individuen müssen entsprechend 100 Varianten des ursprünglichen Simulationsskriptes erzeugt werden. Im Anschluss werden die Simulationen sukzessive durchgeführt und die Ergebnisse hinsichtlich der Zielfunktionen ausgewertet (vgl. Gleichung (10)). Um die f_1 -Güte (Abweichung Mess- und Simulationswerte) berechnen zu können, muss zunächst die Abweichung zwischen den Datensätzen berechnet werden. Es wird ersichtlich, dass dieser Prozess je nach Komplexität und Dauer der Simulation, Größe der Datensätze sowie Größe der Population mit entsprechendem Zeitaufwand einhergeht. Nimmt man beispielsweise für jede durchzuführende Simulation eine Dauer von fünf Sekunden an, ergibt sich bereits in der ersten Generation für 100 Individuen (ohne Parallelisierung) ein Zeitbedarf von über acht Minuten. Erfolgt die Optimierung über mehrere Generationen hinweg, ist mit einer Optimierungsdauer von mehreren Stunden zu rechnen. Im Rahmen der Validierungsabschnitte werden die Zeitbedarfe für Optimierungen konkret aufgezeigt.

Im Anschluss an die Güteberechnung der initialen Population werden die Phasen Selektion, Kreuzung und Mutation durchlaufen. *Dominierte Lösungen*, die in Bezug auf keine der Zielfunktionen bessere Ergebnisse

liefern, werden ausgeschlossen [WEI15]. Die Population wird im Anschluss durch Kreuzung von Genomen und zufällige Mutation einzelner Designvariablen wieder auf die ursprüngliche Anzahl von 100 Individuen aufgefüllt. Für diese neuen Individuen werden wieder neue Simulationsskripte generiert. Bis ein Abbruchkriterium erreicht ist, wird dieser Prozess iterativ wiederholt. Die Identifikation von dominierten und *nicht-dominierten* Lösungen ist in Abbildung 39 dargestellt. Im Koordinatensystem wird die f_2 -Güte (Summe der LoD) über die f_1 -Güte aufgetragen. Der Wert auf der Abszisse läuft im Ursprung gegen 0 und kann potenziell unendlich groß werden, die Grenzen der Ordinate sind der minimale LoD und der maximale LoD. Bei m Modulen mit jeweils einem minimalen LoD von 1 und einem maximalen LoD von 3 liegt die untere Grenze auf der Ordinate bei m und die obere Grenze bei $3m$. Im Ursprung befindet sich der Utopie-Punkt. Hierbei handelt es sich um einen theoretischen Zustand, in dem alle Gütefunktionen minimal bzw. null werden. Dieser ist, dem Namen entsprechend, in realen Anwendungen nicht erreichbar [DK19]. Er liegt beim betrachteten MKOP im Punkt $(0|m)$. Zu interpretieren ist der Punkt mit keiner Abweichung der Mess- und Simulationswerte bei minimalem LoD aller MSM. In Abbildung 39 sind weiter die *Pareto-optimalen Lösungen* als Sterne markiert. Hierbei handelt es sich um diskrete Punkte, da der LoD als natürliche Zahlen repräsentiert wird. Die Pareto-Front ist durch die blaue Kurve dargestellt und entsteht durch Interpolation der Pareto-optimalen Lösungen. In realen Applikationen kann die ideale Pareto-Front nur mit viel Aufwand gefunden werden und wird daher näherungsweise berechnet. Die Front ist an dieser Stelle der Vollständigkeit halber eingezeichnet. Im betrachteten MKOP und mit der Definition des LoD würde es keinen Sinn machen, Werte, die nicht Element der natürlichen Zahlen sind, für den LoD zu wählen.

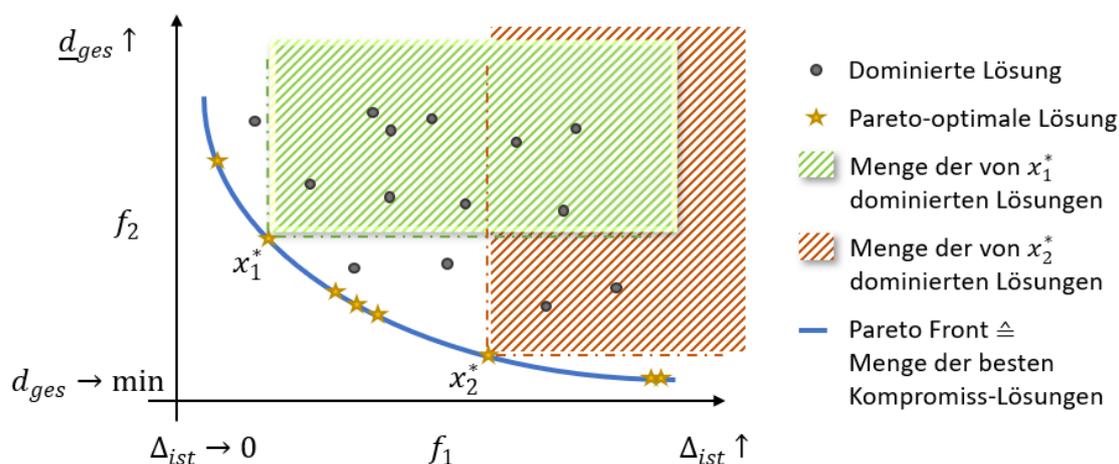


Abbildung 39: Ideale Pareto-Front mit dominierten Lösungen.

Um den Zusammenhang zwischen dominierten und nicht dominierten Lösungen aufzuzeigen, werden exemplarisch die Punkte x_1^* und x_2^* herausgegriffen. Weitere gefundene Lösungen sind durch graue Punkte visualisiert. Hierbei handelt es sich um dominierte Lösungen. Dominierte Lösungen zeichnen sich dadurch aus, dass es „bessere“ Lösungen gibt. Besser bedeutet in diesem Zusammenhang, dass eine andere Lösung (hierbei muss es sich nicht um eine Pareto-optimale Lösung handeln) hinsichtlich einer oder mehrerer Gütekriterien besser abschneidet und hinsichtlich der anderen Gütekriterien mindestens gleich gut [WEI15]. Die Mengen der von x_1^* und x_2^* dominierten Lösungen sind durch die grün und rot schraffierten Flächen visualisiert. Die dominierten Lösungen werden über den Evolutionsprozess sukzessive aus der Population entfernt.

In Abbildung 40 ist exemplarisch die iterative Verbesserung und Konvergenz einer Pareto-Front aus der Arbeit von Saputra et al. [SHI+20] dargestellt. Die Wahl der Abbildung ist damit zu begründen, dass sowohl die OTEC- als auch die Jentner-Anlage, bedingt durch die geringe Anzahl an MSM und entsprechend LoD, hinsichtlich der f_2 -Güte nur einen kleinen Bereich abdecken. Das Konvergenzverhalten könnte mit diesen Beispielen nicht so anschaulich dargestellt werden. In Abbildung 40 verbessert sich über 200 Generationen hinweg die Lösungsmenge hinsichtlich der beiden Gütefunktionen. Hervorzuheben sind zum einen die schnelle Konvergenz zu Beginn der Optimierung und zum anderen die Abnahme an Verbesserungen im späteren Verlauf der Optimierung. Um das Konvergenzverhalten zu verdeutlichen, sind die Individuen der unterschiedlichen Generationen mit unterschiedlichen Farben und Formen gekennzeichnet. Die initiale Population, dargestellt durch grüne Dreiecke, liegt weit auseinander. Dieser Effekt ist beabsichtigt, um zu Beginn einen möglichst großen Lösungsraum abzudecken. Dies reduziert die Wahrscheinlichkeit, dass sich die Population in einen lokalen Extrempunkt entwickelt. Innerhalb der ersten 50 Generationen ist noch eine starke Verbesserung hinsichtlich der beiden Gütefunktionen zu erkennen. Beim Übergang von der 100. zur 200. Generation sind nur noch kleine Verbesserungen zu erzielen. Daher ist eine sinnvolle Wahl der Abbruchkriterien relevant, die allerdings Erfahrungen beim Bediener voraussetzt. Die resultierende Lösungsmenge ist eine Annäherung an die tatsächliche Pareto-Front des MKOP.

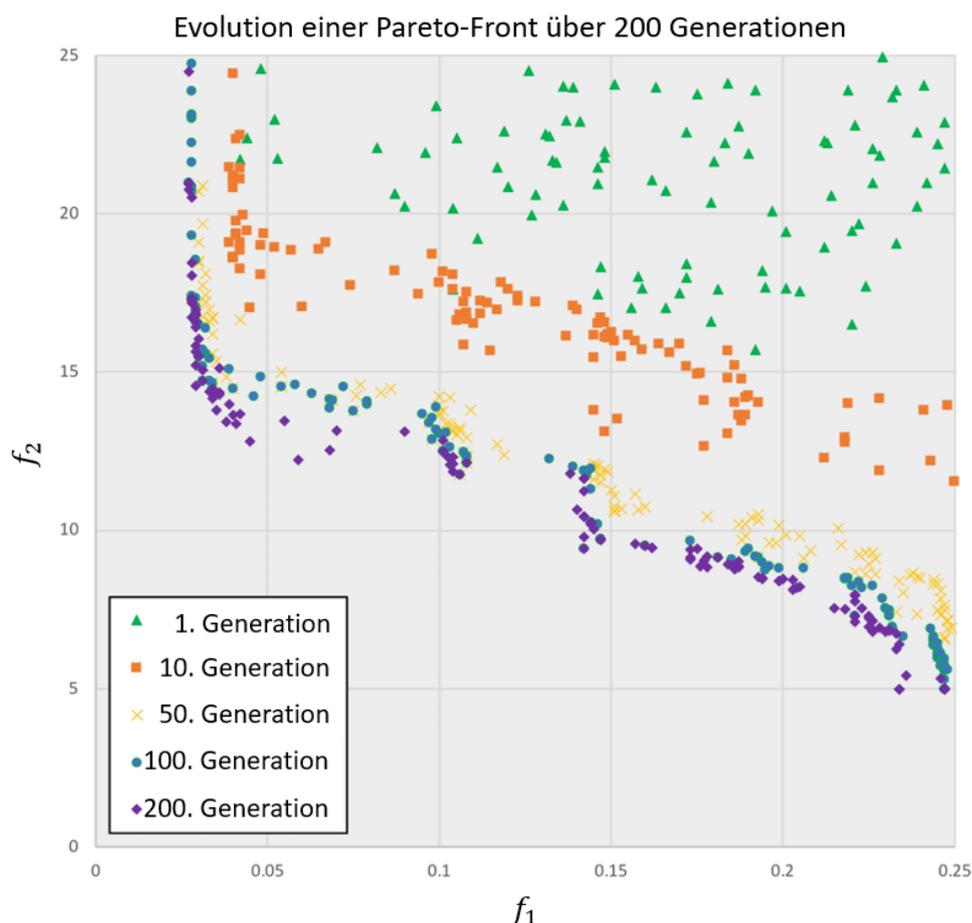


Abbildung 40: Exemplarische Evolution und Konvergenz einer Pareto-Front über 200 Generationen hinweg [SHI+20].

Nachdem ein Abbruchkriterium für den Evolutionsprozess erreicht ist, wird die Lösungsmenge vom AS visualisiert. Der Bediener des AS kann anschließend aus der Lösungsmenge einen entsprechend den An-

forderungen geeigneten Lösungskandidaten für den weiteren betriebsparallelen Einsatz der Co-Simulation auswählen. Im letzten Abschnitt zur prototypischen Implementierung soll aufgezeigt werden, wie das Simulationsskript im Rahmen der Adaption angepasst wird.

5.4.4 Anpassung des Simulationsskriptes im Rahmen der Adaption

In Listing 6 ist das vom AS im Rahmen der Adaption angepasste Simulationsskript für ein Individuum dargestellt. Hierbei handelt es sich im Wesentlichen um das Simulationsskript aus Abschnitt 4.4.2. Die vom AS vorgenommenen Änderungen sind gelb hervorgehoben. Die erste Änderung tritt in den Zeilen 11 und 12 auf. Die Bezeichner der FMUs wurden dahingehend angepasst, dass der LoD direkt vom AS identifiziert werden kann. Im Rahmen dieser Arbeit wird der Bezeichner „ModulName_LoD_LoDdesMSM“ vorgeschlagen. Auch andere Bezeichner sind denkbar. Relevant ist nur, dass alle MSM Quellen der gleichen Konvention folgen, um die Automatisierung der Adaption zu ermöglichen. Die zweite Änderung betrifft die Zeilen 25 und 26. Hier ruft das AS automatisch für jeden, vom Bediener für die Adaption freigegebenen, MSM-Parameter eine *setter*-Methode auf. Aus dem TM kann das AS die Datentypen und die initialen Werte der Parameter entnehmen und so die geeignete setter-Methode, z. B. *setReal* oder *setBoolean*, wählen. Bei jedem Methodenaufruf wird der zu setzende Parameter und der neue Wert übergeben.

```

1  from OMSimulator.lib.OMSimulator import OMSimulator
2  import os
3  oms = OMSimulator()
4
5  parentDir=os.getcwd()
6  oms.setTempDirectory(parentDir+"/temp/")
7  oms.newModel("OTEC")
8  oms.addSystem("OTEC.root", oms.system_wc)
9
10 # Instantiate FMUs
11 oms.addSubModel("OTEC.root.ProcessingStation", "ProcessingStation_LoD_2.fmu")
12 oms.addSubModel("OTEC.root.ProcessingContainer", "ProcessingContainer_LoD_1.fmu")
13 oms.addSubModel("OTEC.root.RobotLoader", "RobotLoader.fmu")
14 oms.addSubModel("OTEC.root.PLC", "PLC.fmu")
15 oms.addConnection("OTEC.root.ProcessingStation.y", "OTEC.root.RobotLoader.u1")
16 oms.addConnection("OTEC.root.PLC.y", "OTEC.root.RobotLoader.u")
17 oms.addConnection("OTEC.root.PLC.y1", "OTEC.root.ProcessingStation.u")
18 oms.addConnection("OTEC.root.PLC.y2", "OTEC.root.ProcessingContainer.u")
19 # Simulation settings
20 oms.setResultFile("OTEC", "simResults_220130_1531_09.csv")
21 oms.setStopTime("OTEC", 3)
22 oms.setFixedStepSize("OTEC.root", 1.0E-4)
23
24 # Parameter settings
25 oms.setReal("OTEC.root.ProcessingStation.rotationalSpeed", 10.26)
26 oms.setReal("OTEC.root.ProcessingContainer.rhoMedia", 964.17)
27
28 # Initialize and run simulation
29 oms.instantiate("OTEC")
30
31 oms.initialize("OTEC")
32 oms.simulate("OTEC")
33 oms.terminate("OTEC")

```

Listing 6: Vom AS im Rahmen der Adaption angepasstes Simulationsskript für ein Individuum.

Wie das AS bzw. konkreter der EA-Algorithmus die LoD und die Parameter wählt, soll an dieser Stelle nochmals im Detail beschrieben werden. Wie in Abbildung 35 aufgezeigt, stellt die Menge an Parametern und der LoD das Genom eines Individuums (IND) dar. Ist das IND Teil der initialen Population, wird dessen Genom zufällig, innerhalb der vorgegebenen Grenzen, durch den EA gewählt. In diesem Fall trägt das AS

in die Zeilen 11 und 12 sowie 25 und 26 (gelb markierte Stellen) von Listing 6 die vom EA übergebenen zufälligen Werte ein. In allen weiteren Generationen kann das Genom des IND immer noch zufällig zusammengesetzt sein, falls es nicht durch Selektion aus der Population entfernt wurde. Anderenfalls setzt sich das Genom durch Kreuzung aus Genomen von Elternindividuen zusammen. Zusätzlich können durch Mutation einzelne Gene zufällig neu bestimmt werden. Dieser Zusammenhang ist in Abbildung 41 exemplarisch dargestellt. In diesem Beispiel sind oben zunächst die Genome aller IND der initialen Population dargestellt. Eine Populationsgröße von vier wäre für eine reale Applikation zu gering oder nur für kleine Suchräume sinnvoll. An dieser Stelle dient sie der besseren Übersicht. Ein einfacher Farbcode verdeutlicht, dass die Gene zum jeweiligen IND gehören. Jedes IND besitzt vier Gene, zwei repräsentieren die Parameter in Listing 6 Zeile 11 und 12 und zwei repräsentieren die LoD in Zeile 25 und 26. Nach der Berechnung der Güte der jeweiligen IND wird in der ersten Selektionsphase IND 4 aus der Population entfernt. Im nächsten Schritt füllt der stochastische Mechanismus *Kreuzung* die Population wieder auf ihre ursprüngliche Größe auf. Aus den Genomen zufällig gewählter IND wird durch Kreuzung ein neues IND erzeugt (IND 5). IND 5 könnte auch eine beliebige andere Zusammensetzung aus den IND 1 bis 3 besitzen. Der stochastische Mechanismus *Mutation* bestimmt im letzten Schritt eine kleine Menge an Genen neu. Die Menge an neu bestimmten Genen je Durchlauf wird durch die Mutationsrate (bspw. Neubestimmung von 1% aller Gene pro Durchlauf) festgelegt. Diese kann durch den Bediener der AS über eine GUI vorgegeben werden. In Abbildung 41 ist IND 2 von der Mutation betroffen, wodurch das dritte Gen Neubestimmt wird. Dem Gen wird in diesem Beispiel durch Zufall ein Wert zugewiesen, welcher nicht Teil des ursprünglichen Genpools war (vgl. IND 1 bis 4). Um diesen Zusammenhang darzustellen, wird das Gen grau schraffiert dargestellt. Aus IND 2 entsteht IND 6. Die IND 1, 6, 3 und 5 stellt im Rahmen der Optimierung die nächste Generation (2. Generation) dar. Dieser Zyklus wiederholt sich nun, je nach vorgegebenem Abbruchkriterium, n mal.

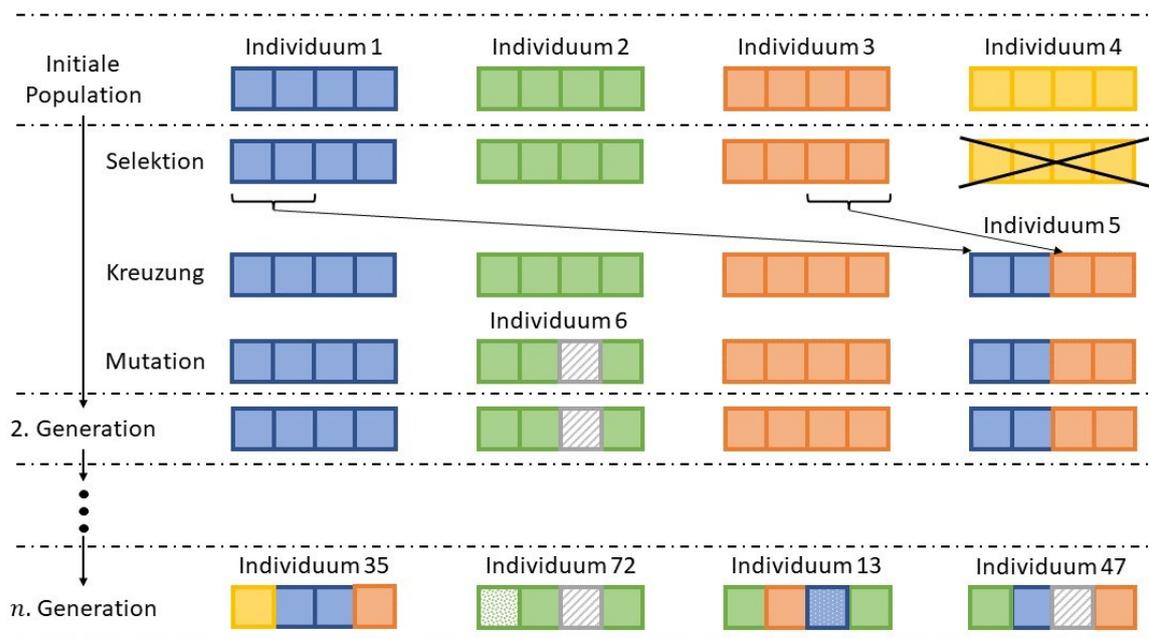


Abbildung 41: Genomzusammensetzung und -veränderung einzelner Individuen über Generationen hinweg.

Im unteren Teil von Abbildung 41 sind die IND der n . Generation abgebildet. Hervorzuheben sind hier zum einen die Nummerierung der IND. IND 13 besitzt in diesem Beispiel eine deutlich niedrigere Nummer als

die anderen IND. Dies weist darauf hin, dass das IND bereits mehrere Selektionsphasen überlebt und früh in der Optimierung gefunden wurde. Dies deckt sich mit der Aussage aus Abbildung 40 (schnelle Konvergenz zu Beginn der Optimierung, Abnahme an Verbesserungen im späteren Verlauf der Optimierung). Weiter fällt auf, dass das erste Gen von IND 35 die gleiche Farbe wie IND 4 besitzt. Obwohl das Genom bereits in der ersten Selektion aus dem Genpool gelöscht wurde, kann durch den stochastischen Charakter der Mutation ein Teil oder das ganze Genom in neuer Kombination wieder auftauchen. Dieser Mechanismus verhindert somit, dass nach einmaliger Untersuchung einer Konfiguration Teile des Suchraums vernachlässigt werden und die Optimierung so ggfs. in Richtung eines lokalen Optimums läuft.

Nach der prototypischen Implementierung der Funktionen für die betriebsparallele Adaption soll das erweiterte AS anhand der beiden in Abschnitt 2.3 vorgestellten Systeme validiert werden. In Abschnitt 5.5 erfolgt die Teilkonzeptvalidierung zunächst an der OTEC Anlage.

5.5 Teilkonzeptvalidierung Teil I – OTEC Anlage

5.5.1 Betrachtetes Validierungsszenario

Für die erste Konzeptvalidierung wird die Gleitschleifanlage der Firma OTEC betrachtet. Das CAD-Modell ist in Abbildung 42 (a) dargestellt, die 3D-Visualisierung des resultierenden Physik-Modells, welches in Abschnitt 5.5.2 vorgestellt wird, in Abbildung 42 (b). Mittels der Anlage können Werkstücke (WS) wie Bohrer und Fräser geschliffen und poliert werden. Hierzu wird das WS (I) in eine Halterung (II) eingespannt, in (b) durch den grauen Zylinder dargestellt, und in einen Bearbeitungsbehälter (III) abgesenkt, dargestellt in dunkelblau.

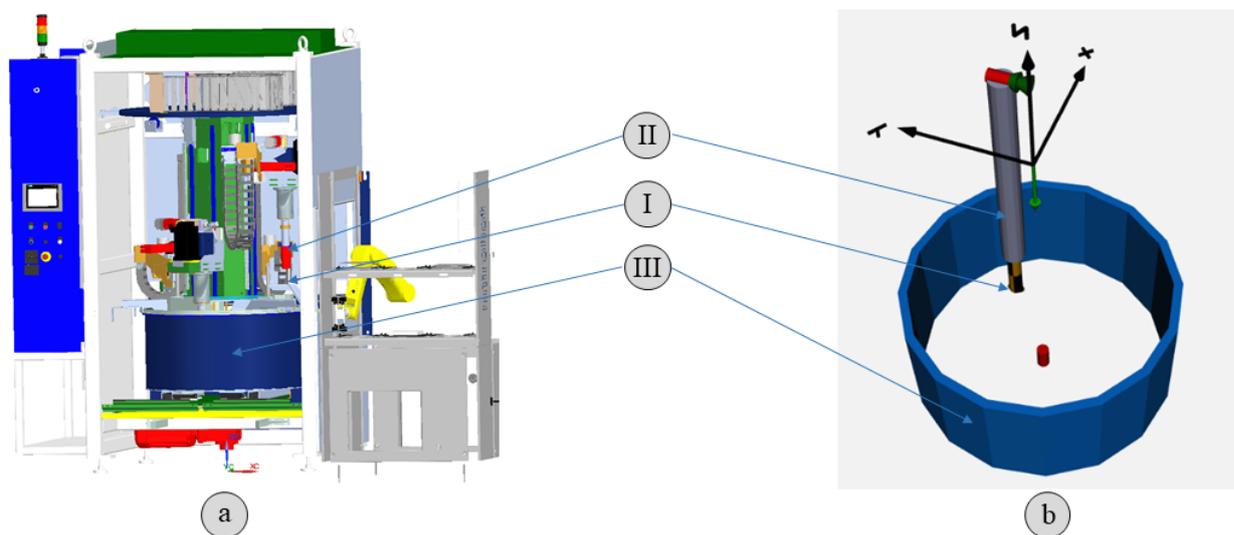


Abbildung 42: Modelle der betrachteten Anlage. CAD Modell (a), 3D-Visualisierung des Physik-Modells (b).

Der Behälter ist mit Schleifmittel befüllt und wird in Rotation versetzt. Das Schleifmittel ist auf dieser Abstraktionsebene nicht dargestellt. Durch die Strömung des Schleifmittels und die Rotationsbewegung des WS wird der Materialabtrag erzielt. Die resultierende Kraft $\underline{f}_{res} = \begin{pmatrix} f_x \\ f_y \end{pmatrix}$ auf das WS soll im Rahmen einer Simulation berechnet werden. Hierfür werden die Module „Bearbeitungscontainer“ und „Spannlanze mit WS“ betrachtet. In Abbildung 43 wird aufgezeigt, wie sich diese resultierende Kraft zusammensetzt. Links

(a) ist ein zylindrisches WS mit Radius r in Draufsicht dargestellt, das von dem Schleifmittel umströmt wird. Die Strömungsgeschwindigkeit u_0 des Schleifmittels wird durch die Rotation des Bearbeitungsbehälters erzeugt. Das Schleifmittel umströmt das WS und wird dabei auf die Geschwindigkeit u_M abgebremst. Hierdurch entsteht bereits eine Kraft auf das WS, welche allerdings nicht f_{res} entspricht. Wird das WS in Rotation versetzt (b), kommt es zu einer Überlagerung der beiden Effekte. Die Schleifmittelströmung trifft auf das mit der Bahngeschwindigkeit ω_r drehende WS und wird abgelenkt. Dieses Verhalten wird durch den Magnus-Effekt [DUB79] beschrieben. In Bezug zur Drehrichtung des WS wird der Schleifmittelstrom in die Richtung der resultierenden Auftriebskraft f_A einseitig stark abgelenkt. Die resultierende Kraft f_{res} ist eine Linearkombination der durch die Strömung wirkenden Kraft und der Auftriebskraft.

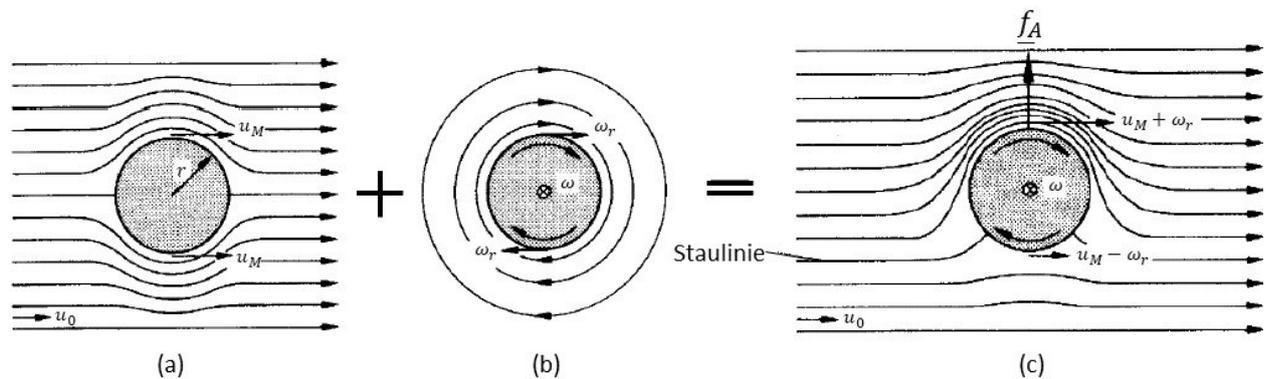


Abbildung 43: Überlagerung der Umströmung des WS (a) mit dessen Rotationsbewegung (b). Überlagerung führt zu Auftriebskraft f_A (Magnus-Effekt (c)) in Anlehnung an [REN18].

Im nächsten Abschnitt werden die Physikmodelle, deren Parameter sowie LoD vorgestellt.

5.5.2 Physikmodelle, Parameter und Detaillierungsgrade

Um den Modellierungsprozess zu beschleunigen und die Vorteile einer 3D-Visualisierung nutzen zu können, werden Elemente der Modelica-Standardbibliothek (*Mechanics* \rightarrow *Multi-Body*) verwendet. Zuerst wird der Bearbeitungsbehälter betrachtet. Dieser besteht aus einer Masse mit Trägheit (*Body*), einem Drehgelenk, einem Achsantrieb sowie einem Sensor für die Drehgeschwindigkeit. Das Physik-Modell des Drehbehälters ist in Abbildung 44 dargestellt. Um die Komplexität des Modells zu reduzieren, wird in einem zweiten Modell der orange umrahmte Block (DC-Maschine mit Reibung, Trägheit und Kaskadenregler) durch die daneben abgebildete Rampen-Funktion ersetzt. Diese Vereinfachung reduziert das Gleichungssystem des Modells von 1721 auf 1186 Modellgleichungen. Die Drehgeschwindigkeit wird aus dem Modell geführt und dient als Eingang des Spannlanzenmodells. Diese wird für die Berechnung von f_{res} benötigt, da sich hierüber und durch den Abstand des WS zum Behälterrand die Strömungsgeschwindigkeit des Schleifmittels annähern lässt.

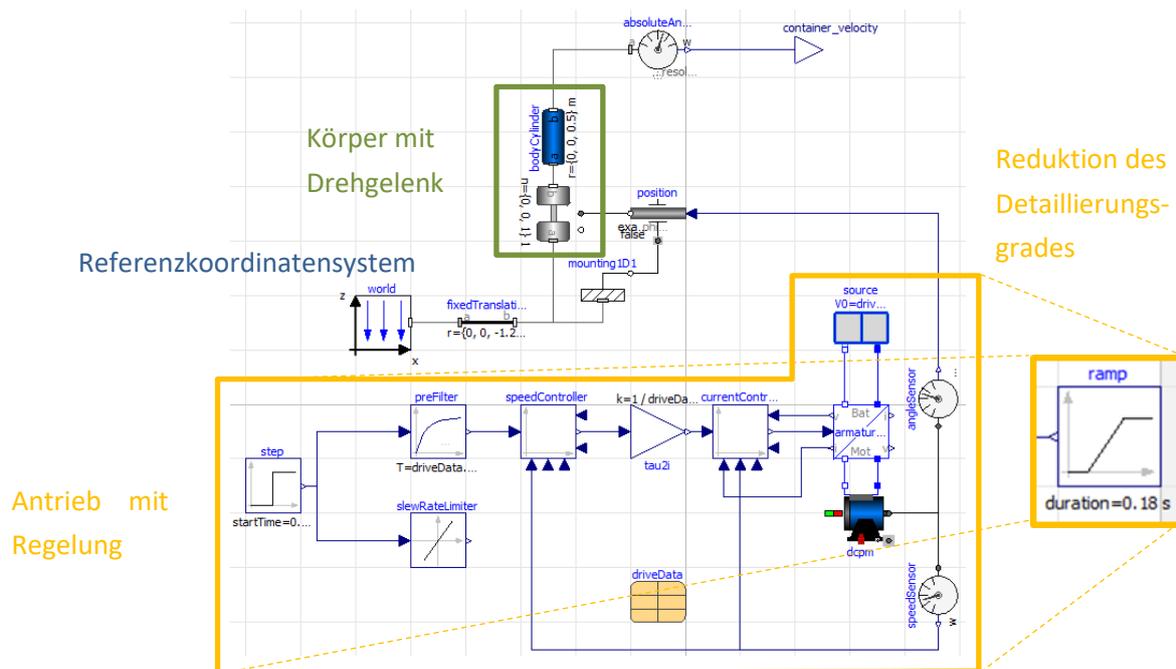


Abbildung 44: Physikmodell des Bearbeitungsbehälters in hohem Detaillierungsgrad und gewählte Vereinfachung in orange.

Das Physikmodell der Spannlanze mit WS ist in Abbildung 45 dargestellt. Die Spannlanze besitzt ein Drehgelenk mit Antrieb (grüner Block), um das WS in die Strömung zu stellen. Ein weiteres Drehgelenk ermöglicht dem WS eine Drehung um dessen Rotationsachse (blauer Block). Dies entspricht dem sogenannten Puls-Antrieb der Anlage. Die Drehrichtung des WS geht als Faktor in die Berechnung von f_{res} (roter Block) ein. Existierende Messdaten der auf das WS wirkenden Kräfte werden für die Plausibilisierung des Modells verwendet. Für die Validierung der Adaption durch das AS wird das Behälter- mit dem Spannlanzenmodell gekoppelt. Dabei stellen der LoD des Bearbeitungsbehälters sowie die frei gewählten Parameter *Schleifmittel-Dichte*, *Amplitude der Trapezoid-Funktion für Pulsantrieb*, *Drehgeschwindigkeit des Behälters* und die *effektive Eintauchtiefe WS in Schleifmittel* mögliche Designvariablen für die Adaption dar.

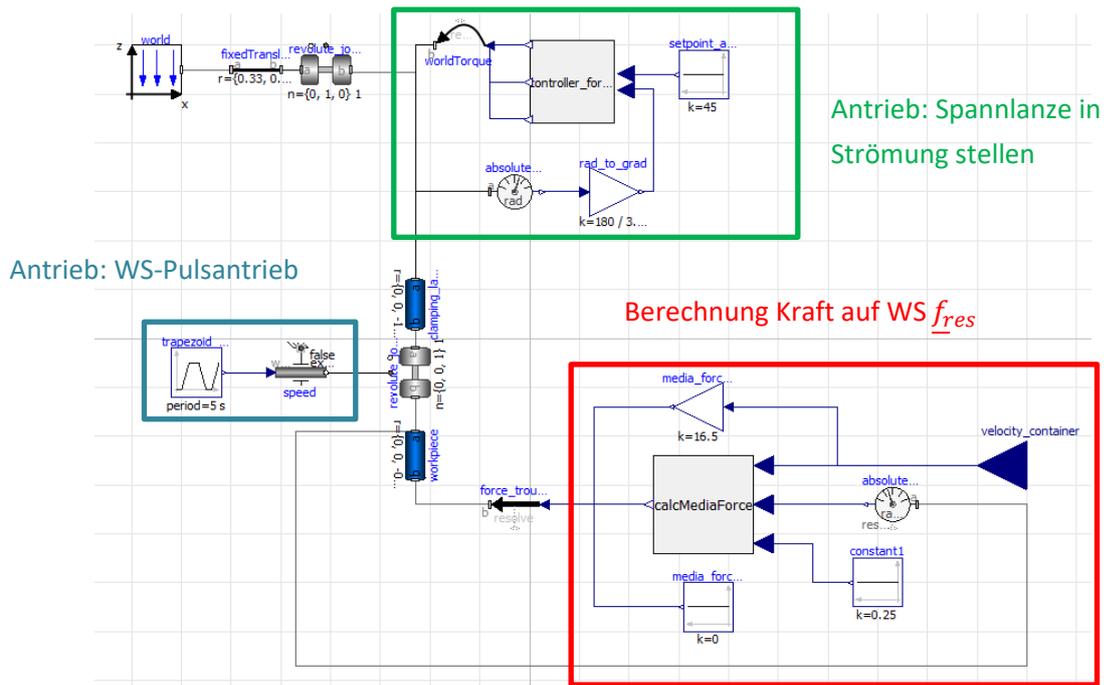


Abbildung 45: Physikmodell der Spannungslanze mit Werkstück inklusive Berechnung der resultierenden Kraft.

Die 3D-Visualisierung des resultierenden Physikmodells, welches bereits in Abbildung 42 (b) dargestellt wurde, ist in Abbildung 46 nochmals im Detail und in zwei unterschiedlichen Perspektiven dargestellt. Links zeigt das Mehrkörpersystem von schräg oben. In der Mitte und rechts ist die Sicht in den Behälter entlang der z-Achse dargestellt. Grüne Pfeile symbolisieren Kräfte, wirkende Momente werden durch grüne Doppelpfeile dargestellt. Rote Zylinder repräsentieren die Drehgelenke der Spannungslanze sowie des Bearbeitungsbehälters. In der mittleren und rechten Ansicht wird deutlich, dass die resultierende Kraft auf das WS f_{res} nicht parallel zum Behälterrund wirkt. Dieses Verhalten tritt durch den in Abschnitt 5.5.1 beschriebenen Magnus-Effekt auf. Beim Wechsel der Drehrichtung des WS (vgl. mittlere und rechte Abbildung) ändert sich die Richtung, aus der die Kraft auf das WS wirkt.

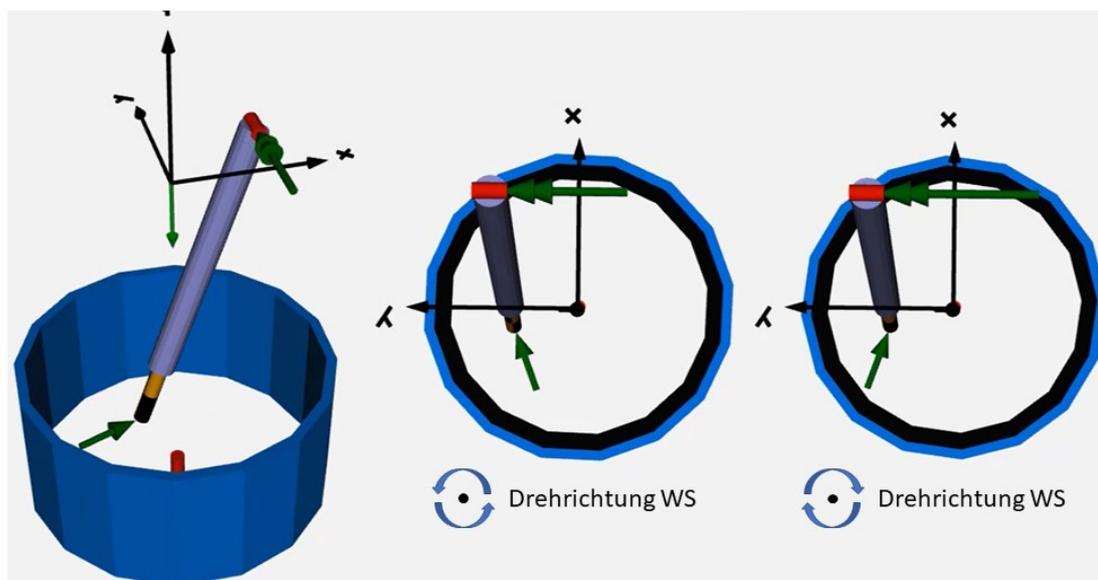


Abbildung 46. Resultierende 3D-Visualisierung der Mehrkörpersimulation.

Nach der Vorstellung der Physikmodelle und der Designvariablen für die Adaption wird diese im nächsten Abschnitt durchgeführt und die Ergebnisse präsentiert.

5.5.3 Ergebnisse der Adaption und Vergleich verschiedener Algorithmen

Das MOEA Framework bietet mehr als 20 Algorithmen (u. a. NSGAI, SPEA2, VEGA) und die Möglichkeit, diese problemspezifisch zu konfigurieren (z. B. Populationsgröße, Mutationsrate). Zudem können die Abbruchkriterien für die Adaption eingestellt werden, z. B. nach vorgegebener Berechnungszeit oder Anzahl an Evolutionsschritten. Erfolg und Performance einer Adaption hängen neben den stochastischen Einflüssen von den Erfahrungen und Kenntnissen des Bedieners hinsichtlich Wahl und Setup des evolutionären Algorithmus ab. Für das betrachtete Beispiel-Szenario kann gezeigt werden, dass bereits mit den Standard-Einstellungen eine Verbesserung der Simulationsergebnisse hinsichtlich der beiden Gütefunktionen erreicht werden kann. In einem ersten Versuch wird exemplarisch der Algorithmus ϵ -NSGAI gewählt und nach 1000 Evolutionsschritten terminiert. Dieser kombiniert den Standard-Algorithmus NSGAI, welcher in der Literatur häufig als Referenz herangezogen wird [KR06], mit einem ϵ -dominanz Archiv [KR06], um Konvergenz sicherzustellen. In Abbildung 47 sind im oberen Teil die gemessenen und simulierten Kräfte f_x und f_y vor und nach der Adaption aufgezeigt. Vor der Adaption werden die geschätzten Parameter aus dem TM der Anlage und für alle Module der LoD 1 verwendet. Da die initialen Parameter bereits sehr nah an den korrekten Werten liegen, ist die Differenz zu den Messwerten gering. In diesem Szenario sind die initialen Schätzwerte dem Algorithmus nicht bekannt. Dieser ist nach der Optimierung der Parameter dennoch in der Lage, mit dem einfachen Behältermodell (LoD_B 1) unmittelbar gute Ergebnisse zu erzielen, und mit dem komplexeren Behältermodell (LoD_B 2) den Anstieg der Kraft f_1 bei Sekunde 0.2 genauer nachzubilden.

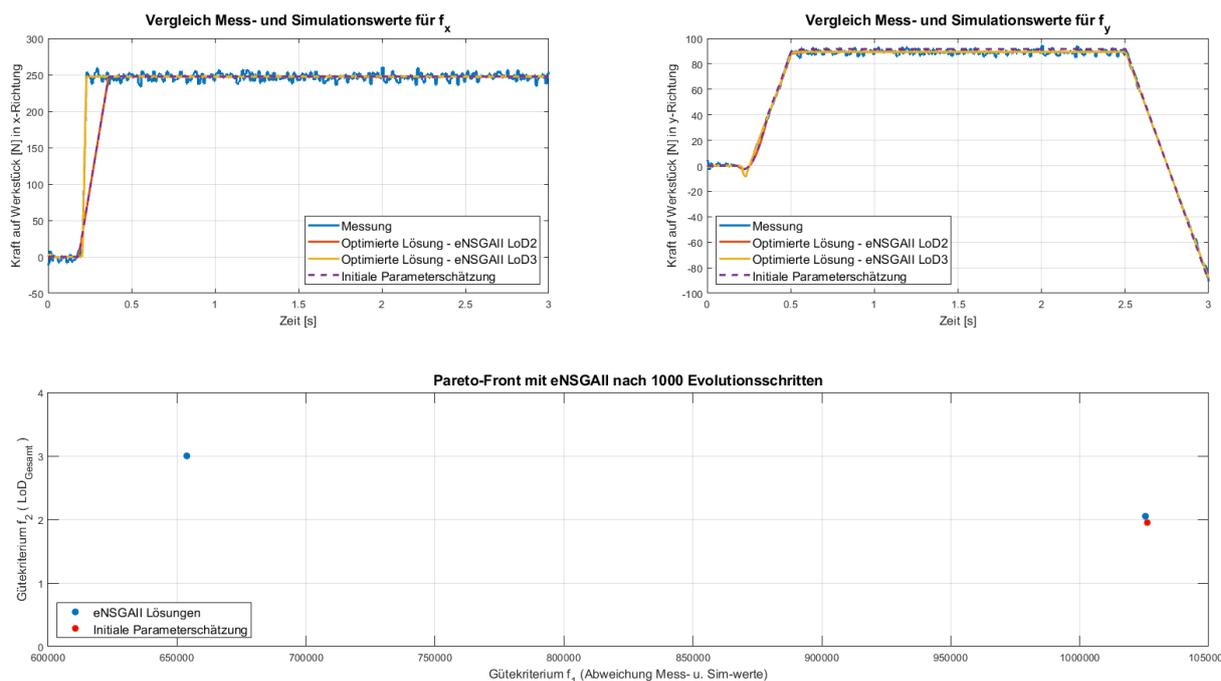


Abbildung 47: Plot der Kräfte auf das WS vor und nach der Adaption (oben) und Pareto-Front des ϵ -NSGAI nach 1000 Evolutionsschritten.

Im unteren Teil von Abbildung 47 ist die Pareto-Front des MKOP dargestellt. Auf der x-Achse ist die Gütefunktion f_1 , auf der y-Achse f_2 aufgetragen. Die Pareto-Front beinhaltet lediglich zwei Punkte (blau), da auf der y-Achse nur zwei Werte angenommen werden können. Hier ist die Summe der LoD eingetragen, das Spannlanzenmodul besitzt nur ein Modell mit LoD_s 1, das Behältermodul zwei mit LoD_B 1 bzw. LoD_B 2. Sollte das AS das komplexere Behältermodell wählen, ist der Gesamt-LoD LoD_{Ges} 3, ansonsten LoD_{Ges} 2. Der rote Punkt beschreibt die Güte der initialen Parameterschätzung. Ein Evolutionsschritt (Vektor mit Designvariablen generieren, Simulationsskript anpassen, Simulation mit drei Sekunden Simulationszeit durchführen, Gütefunktionen berechnen, ggfs. Pareto-Front aktualisieren) benötigt bei diesem Szenario im Schnitt etwa drei Sekunden auf einem durchschnittlichen Büro-Rechner. Bei 1000 Evolutionsschritten ist der komplette Adaptionsvorgang in unter einer Stunde abgeschlossen.

In Abbildung 48 ist exemplarisch noch der Vergleich verschiedener Algorithmen dargestellt. Das MOEA Framework ermöglicht, wie in Abschnitt 5.4.2 beschrieben, über die Analyzer-Klasse den Vergleich von Algorithmen. Die Abbildung zeigt farblich differenziert die Pareto-Fronten der unterschiedlichen Algorithmen (NSGAI, SPEA2, eMOEA). Um den stochastischen Einfluss, der durch die zufällige Initialisierung der ersten Population und durch die Mutationsrate auftritt, zu reduzieren, wurden mit allen Algorithmen zwei Adaptionsprozesse durchlaufen. Dies wird durch die Bezeichnung „seed“ in der Abbildung dargestellt. Offensichtlich wird der stochastische Einfluss beim Vergleich der zwei Durchläufe des NSGAI Algorithmus. Beim Durchlauf „seed 0“ wurde eine Lösung mit LoD_{Ges} 2 und mit LoD_{Ges} 3 gefunden, beim Durchlauf „seed 1“ lediglich eine Lösung mit LoD_{Ges} 2.

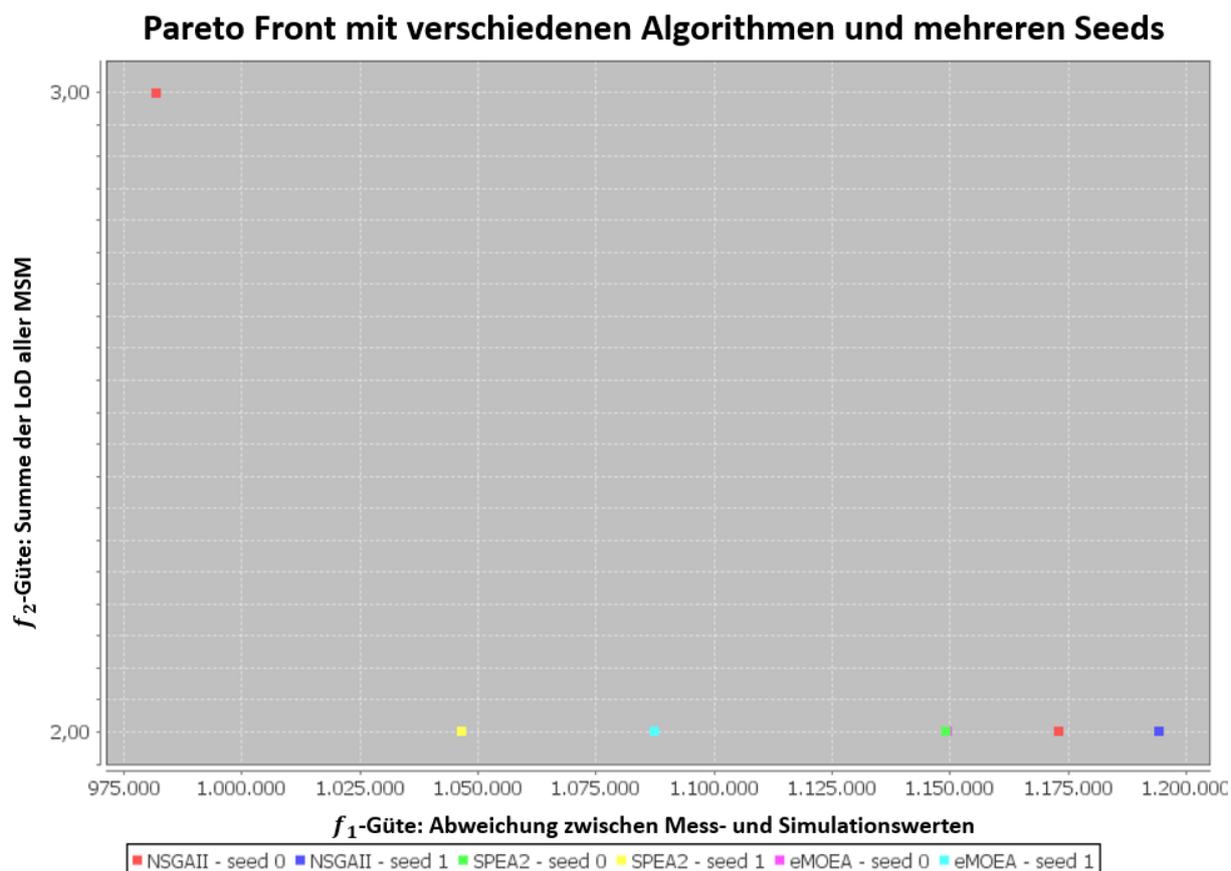


Abbildung 48: Pareto-Front verschiedener Algorithmen mit mehreren Seeds.

Daher ist es wichtig, dem Bediener verschiedene Algorithmen zur Verfügung zu stellen und diese in mehreren Durchläufen zu testen. Gerade ersteres ist wichtig, da die Performance eines Algorithmus stark vom zu optimierenden System abhängt. Gómez und Rojas argumentieren in ihrer Arbeit [GR16] mit dem „No-Free-Lunch“-Theorem. Im Kontext des maschinellen Lernens impliziert dieses Theorem, dass alle Lernalgorithmen gleich gut abschneiden, wenn sie über alle möglichen Datensätze gemittelt werden. Dieses nicht intuitive Theorem drückt aus, dass die Suche nach einem allgemeinen, hochgradig prädiktiven Algorithmus nicht sinnvoll ist [GR16]. Es wird daher darauf verzichtet, die optimale Einstellung für den ϵ -NSGAI Algorithmus zu suchen, da diese nur für die Adaption der OTEC Anlage optimal wäre. Nachdem das AS erfolgreich auf das erste System angewandt wurde, folgt nun die Vorstellung des zweiten Validierungsszenarios.

5.6 Teilkonzeptvalidierung Teil II – Jentner Anlage

5.6.1 Betrachtetes Validierungsszenario

Im Rahmen der zweiten Konzeptvalidierung wird ein Beschichtungsprozess in der Kleingalvanik-Anlage der Firma Jentner betrachtet. Wie in Abschnitt 2.3.4 aufgezeigt, besteht die Kleingalvanik-Anlage aus vier Spül- sowie vier Arbeitswannen. In diesem Szenario wird der Fokus auf den Beschichtungsprozess in einer Arbeitswanne beschränkt. Konkret wird betrachtet, wie sich der Silbergehalt in dem Elektrolytbecken über einen Zeitraum von vier Monaten ändert, wenn darin Pinzetten beschichtet werden. In Tabelle 8

sind relevante Informationen und Prozessdaten, die von der Firma Jentner bereitgestellt wurden, zusammengefasst. Über den genannten Zeitraum wurden etwa 7000 WS bearbeitet. Pro beschichtete Charge werden etwa 121 g Silber vom Elektrolyt abgeschieden. Das summiert sich über den betrachteten Zeitraum auf einen Silberverbrauch von ca. 27,5 kg. Die initiale Silbermenge von 8 kg muss daher kontinuierlich ergänzt werden. Dies erfolgt zum einen durch die Anoden, die sich mit der Zeit auflösen, und zum anderen durch Zugabe von außen durch den Bediener der Anlage. In diesem Zusammenhang ist die Anzahl an *Metal Turn Overs* (MTOs) relevant. Ein MTO ist erfolgt, wenn die initiale Silbermenge durch den Prozess verbraucht wurde. Es existiert ein direkter Zusammenhang zwischen dem Alter des Elektrolyts und der Anzahl an MTOs. Mit steigender Anzahl an MTOs nimmt nach Kerr et al. [KBW97] die Abscheidungsrate ab. Bei dem von Kerr et al. untersuchten Beschichtungsprozess mit Nickel ist das Ende der Lebenszeit des Elektrolyts nach 6 MTOs erreicht. Ist dies der Fall, muss das Bad neu aufgesetzt werden.

Tabelle 8: Relevante Prozessdaten zum betrachteten Validierungsszenario.

Volumen Becken	Ca. 192 Liter
Absolute gelöste Silbermenge	Ca. 8 kg
Betrachtungszeitraum	01.03.2021 – 30.06.2021
Anzahl bearbeiteter Chargen	227
WS pro Charge	30
Anzahl bearbeiteter WS	6810
Silberauftrag pro Charge	121 g
Silberverbrauch insgesamt	27,5 kg
MTOs (Metal Turn Over)	3,43

Das Neuaufsetzen eines Elektrolytbades ist durch die verwendeten Edelmetalle und Zusatzstoffe wie Einbner und Glanzbildner teuer. Es ist daher erstrebenswert, dieses Ereignis möglichst lange hinauszuzögern. Dies kann erzielt werden, indem beim Nachdosieren nicht zu viel Silber hinzugefügt wird. Wird zu viel Silber ergänzt, steigt dadurch die Abscheidungsrate. Dies führt wiederum zu einer größeren Schichtdicke auf den WS. Es ist daher erstrebenswert, den Silbergehalt konstant im niedrigen Arbeitsbereich zu halten. So kann die Prozessqualität sichergestellt und ein Neuaufsetzen des Elektrolytbades möglichst lange hinausgezögert werden. Jeder Beschichtungsprozess hat einen optimalen Betriebsbereich, der von den Konzentrationen der einzelnen Bestandteile abhängt. Kölle et al. [KMS+19] schätzen die richtige Konzentration der Elektrolytsubstanzen als entscheidend für einen ressourcen- und energieeffizienten Galvanisierungsprozess ein. Diesen Bereich zu finden und beizubehalten ist ein aufwändiger Prozess, der zudem viel Expertenwissen erfordert. Leiden et al. [LKT+20] identifizieren die Steuerung und Dosierung von Elektrolyten als eine zentrale Herausforderung beim Betrieb von galvanotechnischen Prozessketten.

Weiter beschreiben Leiden et al., dass die Nachdosierung üblicherweise nach Vorgaben des Prozesslieferanten zwischen zwei Messungen erfolgt. Die Steuergröße für die automatische Nachdosierung ist in der Regel Amperestunden. Auf dieser Basis werden die Additive mit Dosierpumpen in den Elektrolyten nach-

dosiert. Zwischen den Messungen sind die aktuellen Konzentrationen der Elektrolytsubstanzen nicht bekannt. Analytische Messungen, insbesondere für organische Stoffe, sind aufwendig und werden in der Regel nur einmal im Monat durchgeführt. Nur wenn Probleme auftreten, werden zusätzliche Messungen durchgeführt. Dies ist jedoch zu spät, da die Probleme bereits aufgetreten sind und hohe Kosten für Nacharbeiten und Reparaturen verursachen. Daher ist nach Leiden et al. ein vorausschauender Wartungsansatz erforderlich, der den Gehalt an Elektrolytsubstanzen mit geringem Aufwand und geringen Kosten überwacht [LKT+20]. Eine betriebsparallele Simulation kann eingesetzt werden, um die benötigten Nachdosierempfehlungen zu liefern. Die hierfür im Rahmen der Arbeit entworfenen Modelle werden im folgenden Abschnitt vorgestellt.

5.6.2 Physikmodelle, Parameter und Detaillierungsgrade

Für den Galvanikprozess werden zwei Hauptbestandteile, die Einfluss auf den Prozess und den Verbrauch der Elektrolytsubstanzen haben, identifiziert. Den ersten Bestandteil stellen die WS und der WS-Träger dar. Um einen konsistenten Modellierungsansatz sicherzustellen, werden ebenfalls, wie in Abschnitt 5.5.2, Elemente der Modelica-Standardbibliothek verwendet. In Abbildung 49 ist das Physikmodell des WS-Trägers inklusive WS dargestellt. In diesem Modell ist es möglich, über Parameter drei verschiedene Modelleigenschaften einzustellen. Zunächst kann die Anzahl der WS auf dem Träger vorgegeben werden. Darüber hinaus schlagen Leiden et al. [LKT+20] vor, die Oberfläche der WS sowie deren Verschleppungsfaktor zu berücksichtigen. Die Verschleppung durch die WS ist dahingehend relevant, dass hierdurch Elektrolyt aus den Bearbeitungswannen in die Spülwannen eingetragen und im folgenden Prozessschritt Wasser aus den Spülwannen in die nächste Bearbeitungswanne eingetragen wird. Der Aus- und Eintrag verursacht jeweils eine Änderung der vorherrschenden Konzentrationen. Der Verschleppungsfaktor ist eine Abschätzung, wie viel Flüssigkeit aufgrund der Geometrie im Schnitt verschleppt wird [LKT+20]. Das Produkt dieser drei Parameter ist der Austrag (engl. Dragout) pro bearbeiteter Charge. Dieser stellt die erste Modellschnittstelle dar. Daneben besitzt das Modell des WS-Trägers noch einen Pulsgenerator. Dieser erzeugt ein boolesches Signal, das dem nachgelagerten Elektrolytbad signalisiert, dass WS für die Bearbeitung bereitstehen. Das Signal wird über die zweite Modellschnittstelle aus dem Modell herausgeleitet. Das Modell des WS-Trägers liegt nur im LoD 1 vor.

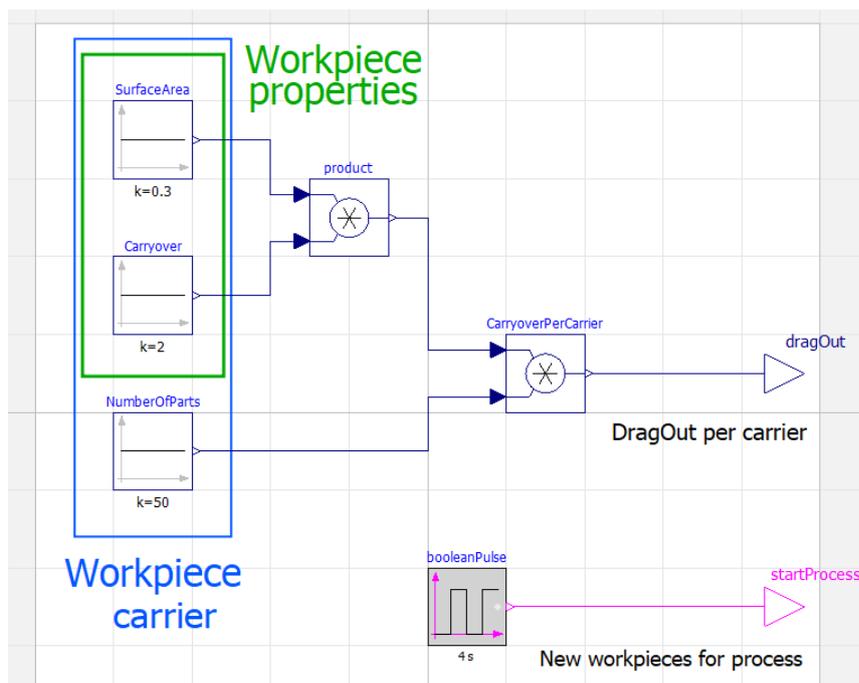


Abbildung 49: Physikmodell des Werkstückträgers.

Der zweite für den Galvanikprozess als relevant identifizierte Hauptbestandteil stellt der Elektrolyt dar. In Abbildung 50 ist das Physikmodell des Elektrolyts mit LoD 1 dargestellt. Die zugrundeliegenden Gleichungen für dieses Modell sowie für das Modell mit LoD 2 sind der Arbeit von Leiden et al. [LKT+20] entnommen. Das Elektrolytmodell mit LoD 1 besteht aus fünf relevanten Komponenten, die jeweils durch einen farbigen Rahmen hervorgehoben sind. Die Komponente oben in der Abbildung stellt den Bearbeitungsprozess dar. Dieser wird durch einen Zustandsautomaten abgebildet. Sobald über die boolesche Schnittstelle „*previousStepDone*“ das Signal kommt, dass die WS bereitstehen, beginnt, nach kurzer Verzögerung durch das Eintauchen, der Beschichtungsprozess. Dieses Signal kann entweder vom WS-Träger kommen, wenn es sich um den Elektrolyten in der ersten Arbeitswanne handelt, oder vom Elektrolytmodell aus der vorherigen Wanne. Dafür gibt es direkt unter dem Zustandsautomat eine weitere, ebenfalls grün umrandete Komponente, die überprüft ob der Beschichtungsprozess abgeschlossen ist. Nach Abschluss wird wieder ein boolesches Signal über die Schnittstelle „*processDone*“ an das nächste Elektrolytmodell geleitet, sofern es existiert. Die Weiterleitung erfolgt mit einstellbarer Verzögerung, um den Transport der WS von einem Becken zum nächsten abzubilden. In der blau umrandeten Komponente wird der elektrochemische Zerfall [LKT+20] der Komponenten berechnet. Jede Substanz besitzt ihre eigene Zerfallsrate. Die anliegende Stromstärke beeinflusst maßgeblich, wie viel der Elektrolytkomponenten abgeschieden wird und sich an den WS ablagert. Der elektrochemische Zerfall wird daher als Produkt der jeweiligen Zerfallsrate und der Stromstärke berechnet. Gelb umrandet ist die Nachdosierlogik dargestellt. Diese schlägt, aufgrund der simulierten Konzentrationen und des Gesamtvolumens, Nachdosiermengen vor und führt diese aus. In der darüber liegenden dunkelgrün umrahmten Komponente werden die aktuell vorherrschenden Konzentrationen, die sich aus Prozessdauer, elektrochemischem Zerfall und Nachdosierung ergeben, berechnet. In diesem Modell wird die Verschleppung (*DragIn* und *DragOut*) durch die WS vernachlässigt. Der Effekt der Verschleppung ist im Physikmodell des Elektrolyts mit LoD 2 berücksichtigt. Damit im Rahmen der Simulation eines mehrstufigen Beschichtungsprozesses eine Kombination von Elektrolytmodellen mit unterschiedlichem LoD möglich ist, wird beim Elektrolytmodell mit LoD 1 die Verschleppung

durch das Modell durchgeführt (vgl. Abbildung 50 unten). Dies stellt sicher, dass die im Modell des WS-Träger berechnete Verschleppung weiterhin genutzt werden kann.

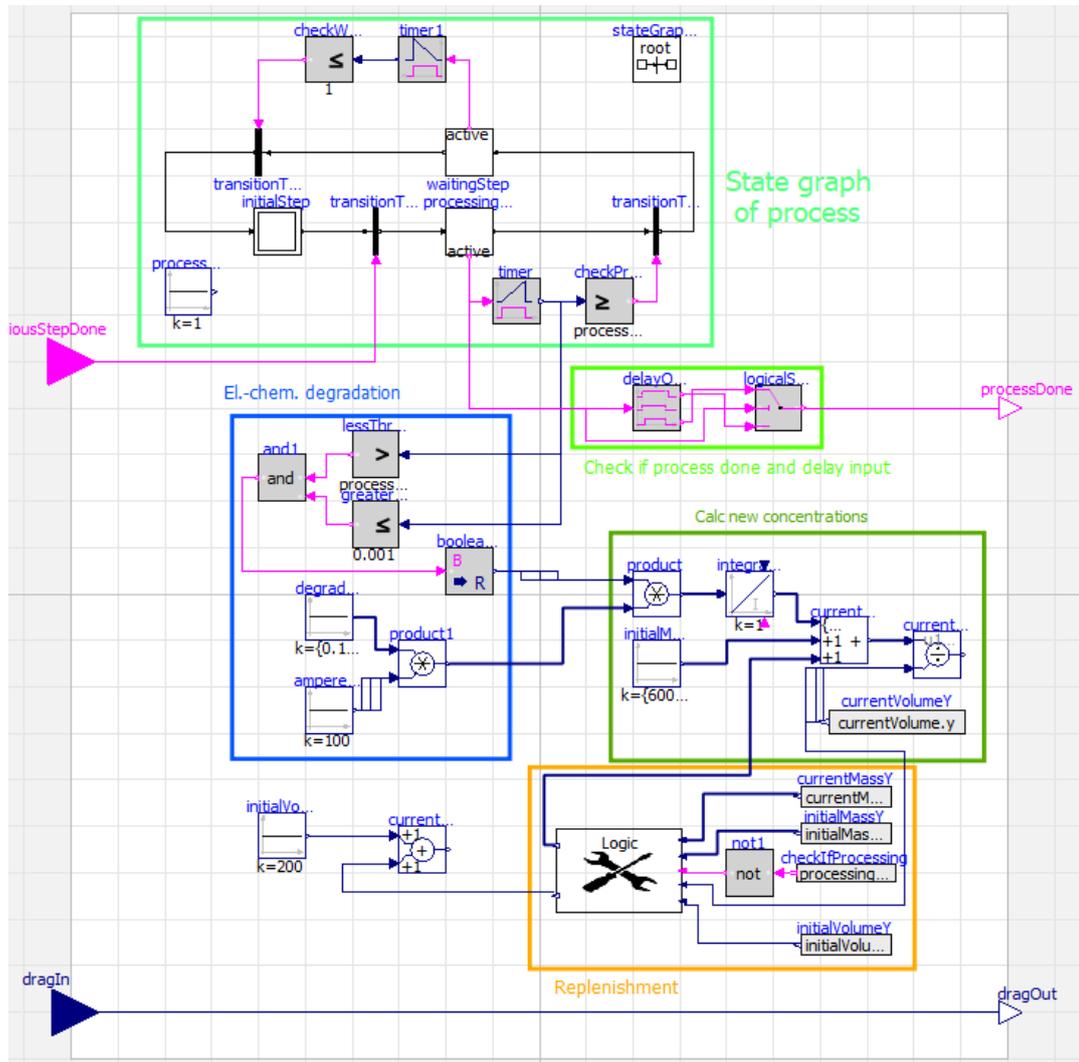


Abbildung 50: Physikmodell des Elektrolyts mit Vernachlässigung der WS-Verschleppung.

Das Elektrolytmodell mit LoD 2 ist in Abbildung 51 dargestellt. Neben den fünf Komponenten, über die auch das Modell mit LoD 1 verfügt, verfügt dieses Modell noch zusätzlich über zwei Komponenten (hellblau und rot umrandet), die die Verschleppung aus der vorherigen bzw. in die nächste Wanne berechnen.

Zu den Parametern, die bisher auf Expertenwissen beruhen, gehören unter anderem der Verschleppungsfaktor sowie die elektrochemische Zerfallsrate. Gerade letztere ist stark stoff- und prozessabhängig und dadurch aufwändig zu bestimmen. Es wäre sinnvoller, diese Parameter im Rahmen einer betriebsparallelen Adaption durch das AS genauer zu bestimmen. Zusätzlich wäre es interessant herauszufinden, ob das komplexere Elektrolytmodell einen signifikanten Mehrwert liefert. Die Ergebnisse der Adaption werden im nächsten Abschnitt vorgestellt.

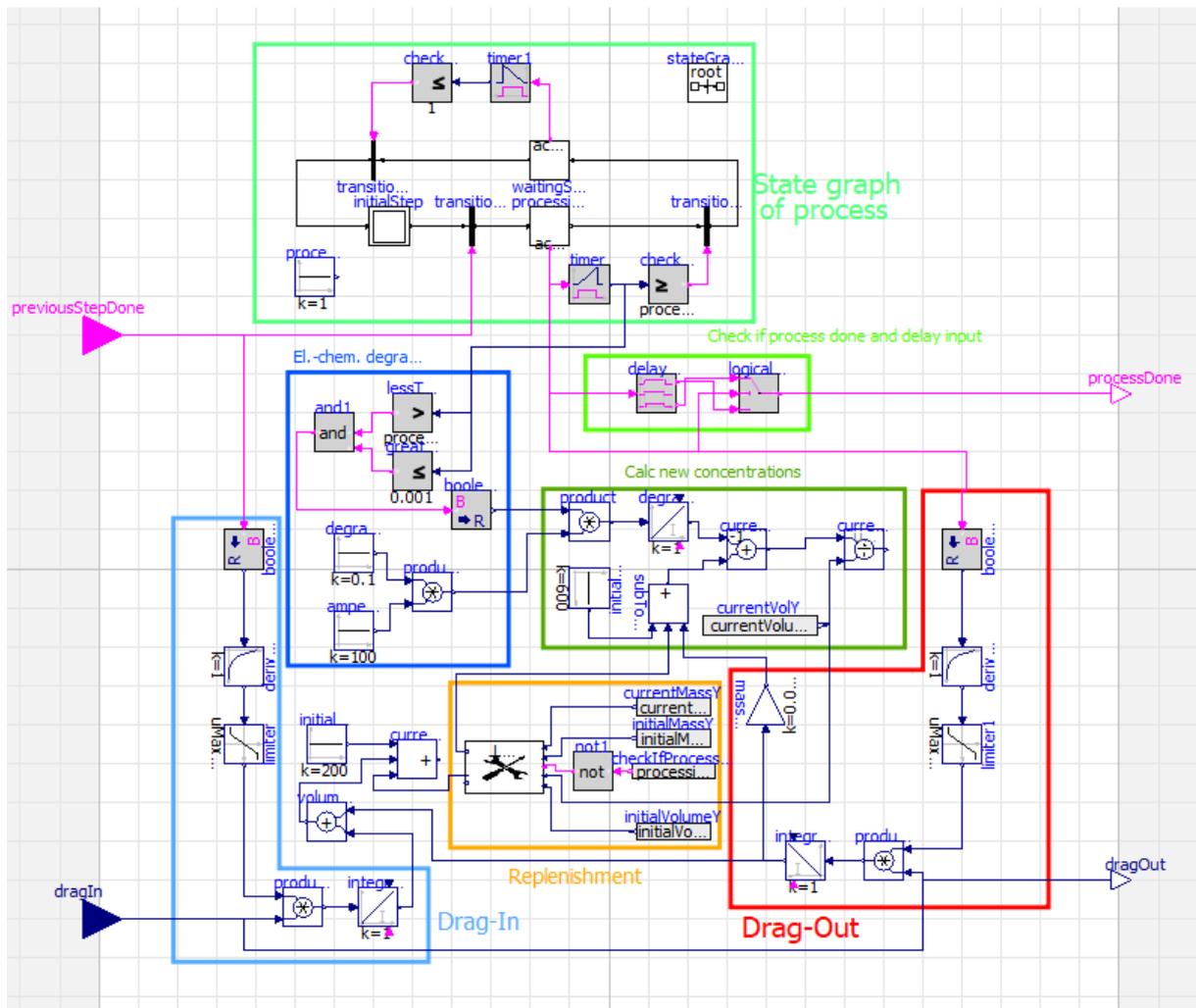


Abbildung 51: Physikmodell des Elektrolyts mit Betrachtung der WS-Verschleppung.

5.6.3 Erkenntnisse aus Modellvalidierung

Die drei Physikmodelle sind sowohl einzeln als auch in Kombination im Rahmen einer Co-Simulation lauffähig. In Abbildung 52 sind die Simulationsergebnisse eines dreistufigen Beschichtungsprozesses dargestellt. Die Struktur der Co-Simulation, bestehend aus dem Modell des WS-Trägers und drei in Reihe geschalteten Elektrolytmodellen, ist unten links abgebildet. Dargestellt sind die Konzentrationen der drei für die Beschichtung verwendeten Metalle Gold, Silber und Rhodium über den Verlauf einer Arbeitsschicht. Die Schicht wird mit acht Stunden bzw. 480 Minuten angenommen. Die Verweilzeit der WS beträgt in der Gold- und Rhodiumwanne jeweils eine Minute und in der Silberwanne eine Stunde. Entsprechend nimmt die Konzentration hier deutlich stärker ab und nach drei Chargen ist ein Nachdosieren nötig.

Aufgrund komplexer Wechselwirkungen zwischen den Bestandteilen der Elektrolyte in Kombination mit zu wenig Messdaten (33 Messpunkte für die Silberkonzentration über einen Zeitraum von vier Monaten) können die Modelle allerdings nicht ausreichend validiert werden. Ohne ausreichend validierte Modelle und Messdaten ist eine betriebsparallele Modelladaption nicht zielführend. Der Mangel an Daten beim Einsatz von Machine Learning führt nach Jones et al. [JGF16] zu Problemen, wie z. B. zu Overfitting sowie Schwierigkeiten beim Nachweis der Verlässlichkeit und Generalisierbarkeit der Modelle. Hofmann et al.

[HOR+19] identifizieren in ihrer Arbeit den Mangel an Daten als größte und am häufigsten genannte technische Herausforderung beim Einsatz von ML. In einer realen Applikation, bei der zu wenig Messdaten vorliegen, könnte das AS daher nicht zur Verbesserung der Modelle und damit zur Reduktion der Abweichung zwischen Mess- und Simulationen beitragen.

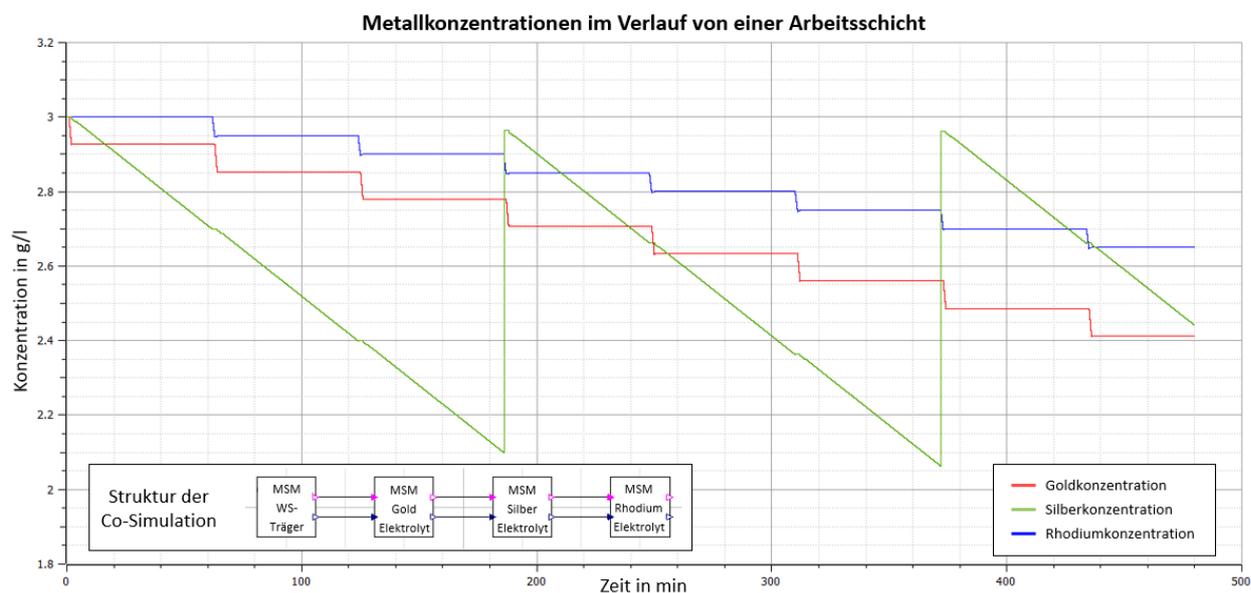


Abbildung 52: Ergebnisse einer Co-Simulation zur Bestimmung der Metallkonzentration im Verlauf einer Arbeitsschicht.

An dieser Stelle könnte das nichtvalidierte Modell genutzt werden, um synthetische Messdaten zu generieren. Jordon et al. [JSH+22] definieren synthetische Daten als Daten, die mit Hilfe eines mathematischen Modells oder Algorithmus erzeugt wurden, um Data Science Aufgaben zu lösen. Hierzu könnten in dem konkreten Betrachtungsfall LoDs und Parameter für die MSM vorgegeben und die Simulationsergebnisse mit weißem Rauschen überlagert werden, um Messeffekte nachzuahmen. Die Aufgabe des EA wäre es dann, die gewählten LoD und Parameter der MSM zu bestimmen. Auf diese Weise würde allerdings nur der EA als Teil des Adaptionalgorithmus validiert werden. Da der EA, wie in Abschnitt 5.4.1 aufgezeigt, Teil einer etablierten Open-Source-Bibliothek ist, wird an dieser Stelle auf eine Validierung von diesem verzichtet.

Im nächsten Abschnitt wird das Zwischenfazit für das Teilkonzept der betriebsparallelen Modelladaption gezogen.

5.7 Zwischenfazit

In diesem Kapitel wurde die zweite Lebenszyklusphase einer modularen Produktionsanlage, die Betriebsphase, betrachtet. Zunächst wurde festgelegt, welche der in Abschnitt 2.2.3 definierten Anforderungen an das AS im Rahmen dieses Kapitels erfüllt werden sollen. Anschließend wurde der aktuelle Stand von Wissenschaft und Technik zu den Themen Datenaufbereitung, Durchführung von Adaptionprozessen und LoD von Modellen untersucht. Das Teilkonzept wurde im Detail vorgestellt, die Adaptionaufgabe als mathematisches Optimierungsproblem formuliert und eine geeignete Algorithmik für die Lösung des Problems identifiziert. Basierend darauf wurde das AS als Nachweis der generellen Machbarkeit prototypisch um die nötigen Funktionen erweitert. Hierfür wurde eine bestehende Bibliothek eingebunden

und aufgezeigt, wie der Adaptionprozess umgesetzt und das bestehende Simulationsskript angepasst werden kann. Die Validierung erfolgt anhand von exemplarischen Prozessen auf den in Abschnitt 2.3 vorgestellten modularen Produktionsanlagen.

Hinsichtlich der vier in Abschnitt 1.3 formulierten Forschungsfragen ergibt sich folgende Einschätzung. Zusammenfassend erfüllt die zweite Konzeptstufe des implementierten Prototyps die in Abschnitt 5.1 formulierten Anforderungen zur betriebsparallelen Adaption einer Co-Simulation. Über die in (7) formulierte Zielfunktion f_1 wird Anforderung A3, der Vergleich der Simulations- und Messwerte, erfüllt. Durch die mögliche Vorgabe einer maximalen Abweichung zwischen diesen Datensätzen wird Anforderung A4 abgedeckt. Die automatische Adaption betroffener Simulationsmodelle wird, wie in Abschnitt 5.5 gezeigt, ebenfalls abgedeckt. Anforderung A5, die Lokalisierung möglicher simulationsseitiger Ursachen für die Abweichung, wird teilweise erfüllt. So kann der Bediener einzelne Lösungen der angenäherten Pareto-Front betrachten und beispielsweise nachvollziehen, dass ein Modulmodell nicht ausreichend detailliert dargestellt oder Modellparameter nicht optimal eingestellt waren. Für eine umfassende Ursachenidentifikation ist es allerdings notwendig, das bestehende Konzept um eine Sensitivitätsanalyse [SB17] zu erweitern. Bei dieser Analyse wird nach Zusammenhängen zwischen der Varianz von Eingangsfaktoren und Systemantworten gesucht.

Als Voraussetzung entsprechend Forschungsfrage 2 ist eine ausreichende Menge an Messdaten anzuführen. Wie in Abschnitt 5.6.3 aufgezeigt, ist eine betriebsparallele Modelladaption sonst nicht sinnvoll durchführbar. Hierauf wird am Ende des Abschnitts im Detail eingegangen. Benötigte Informationen entsprechend Forschungsfrage 3 sind, neben den Informationen aus Kapitel 4, die Eingaben des AS-Bediener. Diese beinhalten die Vorgabe der maximalen Abweichung, bevor das AS eine Adaption einleitet, die Wahl und die Einstellung des evolutionären Algorithmus sowie die Vorgabe der zu optimierenden Designvariablen und deren Grenzen.

Wie im vorherigen Kapitel bietet es sich an, vor der Beantwortung der letzten Forschungsfrage hinsichtlich des Automatisierungsgrades der Lösung, die Ausprägung des AS genau zu betrachten. Entsprechend der in Abschnitt 2.2.2 vorgenommenen Klassifizierung von AS ist dieses für die betrachtete Lebenszyklusphase als Kombination eines Entscheidungsassistenzsystems (EAS) und eines Ausführungsassistenzsystems (AAS) einzustufen. Der Bediener gibt die Abweichungstoleranz sowie das Setup des Algorithmus und der Designvariablen vor. Das AS gleicht automatisch die zur Verfügung stehenden Mess- und Simulationsdaten ab, berechnet die Abweichung und führt bei Bedarf die Adaption vollautomatisch durch. Bis zu diesem Punkt handelt es sich bei dem AS um ein AAS. Nach der Durchführung werden dem Bediener Lösungen aus der Pareto-Front vorgeschlagen. Der Bediener kann nun aus der Front eine Lösung für den weiteren Betrieb der Co-Simulation auswählen. Es findet der Wechsel vom reinen AAS zum EAS statt. Bezüglich der in Abschnitt 2.2.5 definierten Automatisierungslevel ergibt sich folgende Einschätzung. Das AS kann selbstständig die Adaption durchführen und die Pareto-Front bilden. Das AS ist jedoch nicht in der Lage, selbstständig zu entscheiden, welche Designparameter innerhalb welcher Grenzen variiert werden sollen und welcher Algorithmus für das MKOP geeignet ist. Auch liegt die Entscheidung, welche Lösung aus der Pareto-Front für den weiteren Simulationsbetrieb genutzt werden soll, beim Bediener. Das AS wird daher für diese Lebenszyklusphase mit dem Level 3 „Bedingte Automatisierung“ eingestuft.

Kritische Würdigung des Ansatzes hinsichtlich der industriellen Anwendbarkeit:

Mit steigender Anzahl an Modulen, mit multiplen LoD und einer entsprechend größeren Anzahl an Parametern wächst der Suchraum nichtlinear an („Fluch der Dimensionalität“). Der Adaptionvorgang wird entsprechend aufwendiger. Es müssen entweder mehr Evolutionsschritte durchgeführt werden, was sich negativ auf die Dauer der Adaption auswirkt, oder es ist mit einer schlechteren Güte des Adaptionsergebnisses zu rechnen. Studien zeigen, dass durch die Parallelisierung evolutionärer Algorithmen große Suchräume effizient untersucht, die Qualität der Lösung verbessert und somit diesem Problem begegnet werden kann [UJ13]. Darüber hinaus ist das aufgetretene Problem beim zweiten Teil der Konzeptvalidierung anzuführen. Ohne eine ausreichend große Menge an Messdaten sind keine verwendbaren Ergebnisse durch die Adaption zu erwarten. Eine mögliche Lösung hierfür stellt die synthetische Datengenerierung, wie sie beispielsweise in [AIR18, KMY20] vorgestellt wird, dar. Hierfür muss jedoch zunächst ein validiertes Modell der Anlage existieren, welches belastbare Daten generiert.

Es ist zu betonen, dass es sich in beiden Fällen nicht um spezifische Probleme des im Rahmen dieser Arbeit vorgestellten Konzepts handelt. Kakade [KAK03] formuliert hierzu in seiner Arbeit zwei Fragen, die sich auf Datenmengen und Rechenleistung beziehen: Die Frage der Stichprobenkomplexität (*sample complexity*) lautet sinngemäß: Wie viele Daten müssen wir sammeln, um „Lernen“ zu ermöglichen? Die entsprechende Frage der Berechnungskomplexität (*computational complexity*) lautet: Wie viele Berechnungen sind bei der Verwendung dieser Daten erforderlich, um „Lernen“ zu ermöglichen? Folglich handelt es sich um allgemeine Herausforderungen beim Einsatz von Machine Learning, die es anwendungsspezifisch zu lösen gilt.

Nach der detaillierten Vorstellung der automatischen Modelladaption im Rahmen der Betriebsphase liegt der Fokus im nächsten Kapitel auf der Modellrekonfiguration im Rahmen der Lebenszyklusphase Umbau.

6 Konzeptbaustein für die Modellrekonfiguration

Nachdem die Lebenszyklusphasen Inbetriebnahme und Betrieb in den Kapiteln 4 und 5 abgedeckt wurden, liegt der Fokus in diesem Kapitel auf der Umbauphase. Dies bedeutet, dass aus Sicht des Betreibers ein Zeitpunkt erreicht wird, in dem die Anlage den Produktionsanforderungen und -randbedingungen in ihrer aktuellen Modulkonfiguration nicht mehr gerecht werden kann. Ein struktureller Umbau der Anlage auf Modulebene ist erforderlich. Nachdem dieser Umbau vollzogen wurde, müssen diese Änderungen im nächsten Schritt in das Simulationsmodell der Anlage überführt werden, um die nötige Übereinstimmung zwischen realer Anlage und Modell zu gewährleisten.

Im Rahmen dieses Kapitels werden zunächst in Abschnitt 6.1 die Teilziele, geltende Randbedingungen und Annahmen aufgezeigt. Ebenfalls wird kurz rekapituliert, welche der in Abschnitt 2.2.4 definierten Anforderungen im Rahmen dieses Kapitels erfüllt werden. In Abschnitt 6.2 werden die für dieses Kapitel relevanten technologischen Konzeptbausteine behandelt. Anschließend wird in Abschnitt 6.3 das in Abschnitt 5.3 vorgestellte Teilkonzept auf die Umbauphase erweitert und im Detail vorgestellt. Die prototypische Implementierung des Teilkonzepts wird in Abschnitt 6.4 behandelt und anschließend in Abschnitt 6.5 auf die beiden Industrieszenarien aus Abschnitt 2.3 angewandt und validiert. Das Kapitel schließt mit einem Zwischenfazit in Abschnitt 6.6.

6.1 Teilziele, Randbedingungen und Annahmen

Wie in Abschnitt 2.2.1 beschrieben, ergeben sich auch während der Umbauphase vielzählige manuelle Modellierungstätigkeiten, die sich mit denen aus der Inbetriebnahmephase decken und den Einsatz von Simulation erschweren. Hinzu kommen in dieser Phase Dokumentationstätigkeiten, die das manuelle Arbeitspensum erhöhen. Umbaubedingte Änderungen der Anlage werden jedoch selten ausreichend dokumentiert, was zu Diskrepanzen im weiteren Lebenszyklus der Anlage führen kann. Ziel des in diesem Kapitel beschriebenen Teilkonzepts ist es daher, Modellierungs- und Dokumentationstätigkeiten durch automatische Abläufe zu ersetzen, bzw. die Mitarbeiterinnen und Mitarbeiter bei deren Arbeit zu unterstützen. Hieraus ergibt sich die Notwendigkeit, das AS um die Fähigkeit der Modellrekonfiguration und Dokumentation zu erweitern. In diesem Kapitel muss das erweiterte AS daher, aufbauend auf den in den Abschnitten 4.1 und 5.1 definierten Funktionen (z. B. Modulmodelle laden, Initialwerte setzen, Modulmodelle koppeln und messdatenbasierte Adaptionen durchführen), die folgenden Anforderungen umsetzen. Das AS muss in der Lage sein:

- A7** feststellen zu können, ob sich die reale Systemkonfiguration geändert hat (z. B. Modul ausgetauscht, Modul entfernt, zusätzliches Modul hinzugefügt);
- A8** die automatische Modellgenerierung erneut durchzuführen und Umbauänderungen einzubeziehen;
- A9** durchgeführte Änderungen an der Anlagentopologie automatisch zu dokumentieren.

Das Vorhandensein der MSM für alle Module als FMUs sowie ein TM als Beschreibung der Anlage im AML wird weiterhin als gegeben vorausgesetzt. Um eine Änderung der Systemkonfiguration auf modularer

Ebene, wie beispielsweise einen Modultausch, feststellen zu können, muss die Anlage über entsprechende Sensorik (z. B. RFID-Tags) verfügen und eine Schnittstelle zum Informationsaustausch mit dem AS existieren. Alternativ muss das AS eine Bedieneingabe bereitstellen, in der die Änderungen manuell eingetragen werden können.

6.2 Technologische Konzeptbausteine

Wie die sensorische Erkennung neuer Anlagenmodule grundsätzlich erfolgen kann, wurde bereits durch die Untersuchung des Stands von Wissenschaft und Technik in Abschnitt 3.1.3 aufgezeigt. Auf eine Wiederholung der vorgestellten Methoden soll an dieser Stelle verzichtet werden. Es ist jedoch relevant zu untersuchen, welche Informationen durch die Sensorik erfasst und an das AS übermittelt werden können. Darüber hinaus gilt es darzulegen, welche Methoden und Werkzeuge existieren, um ein bestehendes TM mittels automatischer Abläufe zu bearbeiten.

6.2.1 Durch Sensorik erfassbare Informationen

Zunächst stellt sich die Frage, welche Informationen durch die zur Verfügung stehende Sensorik erfasst und an das AS übermittelt werden können. Durch Laserscanner können beispielsweise die Abmaße eines Anlagenmoduls erfasst werden. Es wäre jedoch sinnvoll, wenn neu eintretende Module alle Informationen über sich selbst, z. B. zu Schnittstellen und MSM, zur Verfügung stellen könnten. Hierfür kann ein Typenschild genutzt werden. Ein Typenschild enthält nach [BMW20b] identifizierende, beschreibende und anzeigende Informationen über ein Wirtschaftsgut. Angesichts der vielfältigen Anforderungen stoßen herkömmliche Typenschilder jedoch an die Grenzen der Darstellung notwendiger Inhalte. Das von der Plattform Industrie 4.0 in Kooperation mit dem ZVEI entwickelte Teilmodell „Nameplate“ dient der Vereinheitlichung der Informationsstruktur zur Modellierung eines Typenschildes gemäß der EU-Maschinenrichtlinie 2006/42/EG. Die Verfügbarkeit von Anlageninformationen soll von der lokalen auf die globale Ebene ausgeweitet werden, so dass weitere Partner entlang der Wertschöpfungskette Zugriff auf die Typenschildinformationen haben. Die maschinelle Lesbarkeit kann mit Hilfe von semantischen Informationen ohne Mehrdeutigkeit realisiert werden. Das Teilmodell „Nameplate“ stellt Informationen eines Typenschildes zur Verfügung. So kann dieses Teilmodell genutzt werden, um identifizierende, klassifizierende Informationen über ein Wirtschaftsgut zu erhalten, wie z.B. den Herstellernamen, den Modelltyp oder die Seriennummer und die vorhandenen Produktkennzeichnungen. [BMW20b]

Im Rahmen dieses Kapitels wird angenommen, dass jedes Anlagenmodul über einen QR-Code verfügt, der auf das Teilmodell „Nameplate“ des Moduls verweist. So kann das AS mit allen nötigen Informationen versorgt werden. Im nächsten Abschnitt soll betrachtet werden, wie das TM automatisch bearbeitet werden kann.

6.2.2 Automatische Bearbeitung des Topologiemodells

Entsprechend Anforderung A9 muss das AS in der Lage sein, durchgeführte Änderungen an der Anlagentopologie automatisch zu dokumentieren. Die Topologie vor dem Umbau ist im TM beschrieben. In Kapitel 4 wurde hierfür bereits der AML-Standard ausgewählt. Um Redundanzen und die Einführung zusätzlicher Datenformate zu vermeiden, bietet es sich an, den AML-Standard auch zur Dokumentation der Änderungen zu nutzen. Es stellt sich nun die Frage, wie das TM mittels automatischer Abläufe bearbeitet werden

kann. Die AML-Datei enthält Daten in einem neutralen XML-basierten Format, daher können die AML-Daten mit normalen XML-Parsern geparkt werden [VBG+17]. Die AutomationML Association stellt zusätzlich ein kostenloses Framework zur Verfügung, um die Anwendung und Integration des AML-Standards zu erleichtern. Die *AutomationML Engine* ist eine Softwarekomponente, die eine Software-API zur Verarbeitung von AML-Dokumenten auf der Basis von C#.NET bereitstellt [DRA21]. Neben Methoden zum Parsen von AML-Standardaspekten wie Rollenklassen, Schnittstellen usw. [VBG+17] unterstützt das Framework weitere wichtige Funktionalitäten, wie das Laden von XML-Dokumenten aus Dateien oder Streams, das Erstellen neuer XML-Dokumente, Suchen und Abfragen innerhalb der XML-Dokumente, die Manipulation der In-Memory-XML-Elemente sowie die Validierung von XML-Dokumenten mit XSD [DRA21]. Da alle notwendigen Funktionen bereitgestellt werden, wird die AutomationML Engine daher im Weiteren als Werkzeug für die Änderungsdokumentation verwendet.

6.3 Konzept zur bedarfsorientierte Rekonfiguration von Co-Simulationen

In Abschnitt 6.3.1 wird zunächst die erweiterte Konzeptskizze vorgestellt und ein Überblick über die nötigen Abläufe gegeben. Die darauffolgenden Abschnitte befassen sich mit jeweils einer der im Rahmen dieses Kapitels zu erfüllenden Anforderungen. Abschnitt 6.3.2 legt dar, wie Änderungen der Anlagentopologie detektiert werden sollen. Die Rekonfiguration des Gesamtsystemmodells wird in Abschnitt 6.3.3 behandelt. Abschließend befasst sich Abschnitt 6.3.4 mit der Dokumentation der Änderungen.

6.3.1 Erweiterte Konzeptskizze und Ablauf der bedarfsorientierten Rekonfiguration

Abbildung 53 erweitert das in Abbildung 33 dargestellte AS um die Lebenszyklusphase Umbau und entspricht der bereits in Abschnitt 3.4 vorgestellten Gesamtkonzeptskizze. Die für die Umbauphase zusätzlich erforderlichen Aspekte sind in Orange dargestellt. Nach dem Übergang von der Betriebs- in die Umbauphase erhält das AS zunächst einen externen Trigger. Dieser ist oben rechts in der Abbildung 53 dargestellt. Sofern die Produktionsanlage über die notwendige Sensorik verfügt, um neue Module zu erkennen, z. B. über RFIDs oder Kontaktschalter, kann der Trigger direkt von der Anlage kommen. Alternativ soll das AS über eine Bedieneingabe verfügen, über welche die Änderungen manuell eingetragen werden können. Unabhängig von der Trigger-Quelle ist es notwendig, dass dem AS Informationen über die Modul-IDs und über die Schnittstellen zur Verfügung gestellt werden. Im nächsten Schritt werden diese Informationen mit dem bestehenden TM abgeglichen und Unterschiede identifiziert. Diese werden in einer neuen Version des TMs dokumentiert. Auf diese Weise kann die Umbau-Historie nachvollzogen werden. Im Anschluss wird die Co-Simulation entsprechend dem neuen TM aufgebaut. Die notwendigen FMUs werden aus einer bestehenden Modellbibliothek geladen, gekoppelt und parametrisiert. Abschließend werden die Simulationen ausgeführt und die Simulationsdaten visualisiert. Die simulationsspezifischen Mechanismen werden bereits durch das Teilkonzept zu Modellgenerierung und -konfiguration bereitgestellt und können wiederverwendet werden. In Abbildung 54 ist der gesamte Ablauf exemplarisch für die Erweiterung einer modularen Produktionsanlage von ursprünglich zwei auf drei Module dargestellt. Vorstellbar ist beispielsweise die Erweiterung eines Galvanikprozesses in der Jentner Kleingalvanik um ein zusätzliches Elektrolytbecken oder die Umstellung der OTEC Gleitschleifanlage von manueller auf automatische Beladung der WS durch eine Roboterzelle.

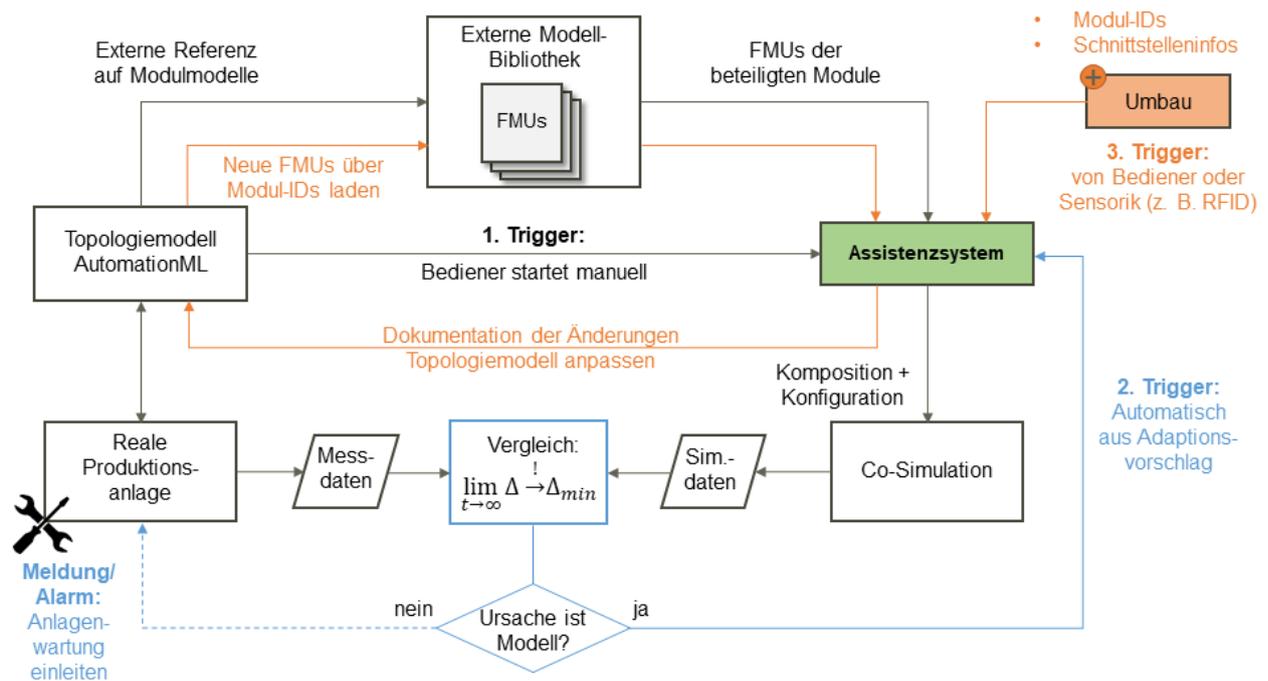


Abbildung 53: Gesamtkonzeptskizze des AS für die drei Lebenszyklusphasen Inbetriebnahme, Betrieb und Umbau.

In den folgenden Abschnitten werden die einzelnen Elemente der Abbildung herausgegriffen und im Detail vorgestellt. Zunächst müssen die umbaubedingten Änderungen detektiert werden.

6.3.2 Detektion von Änderungen der Anlagentopologie

Entsprechend den in Abschnitt 3.1.3 vorgestellten Konzepten zur (teil-)automatischen Modellrekonfiguration kann davon ausgegangen werden, dass umbaubedingte Änderungen der Anlage auf modularer Ebene detektiert werden können. Sollte dies, z. B. bei Bestandsanlagen, nicht der Fall sein, muss das Bedienpersonal des AS die Informationen manuell in das AS einpflegen. Dieser Zusammenhang ist links unten in Abbildung 54 veranschaulicht. Die ursprüngliche Anlagenkonfiguration vor dem Umbau ist grau hinterlegt, die Konfiguration nach dem Umbau ist grün markiert. Unter der Annahme, dass die nötige Sensorik zur Verfügung steht und einen Umbau detektiert hat (durch Sensor S3), wird im ersten Schritt die initiale Topologie der Anlage aus dem TM mit den Sensordaten verglichen. Da eine Produktionsanlage grundsätzlich eine beliebige Anzahl an Umbauphasen durchlaufen kann, muss es sich hierbei nicht um das TM aus der Inbetriebnahmephase (hier als Version 1 angenommen) handeln. Rechts in Abbildung 54 wird daher allgemein eine Version n des TM angenommen und vom AS geladen. Zusätzlich zu den Sensordaten sind noch Mapping-Informationen nötig, die z. B. den RFIDs die entsprechenden MSM-Bezeichner und AML-IDs zuordnen. Da die Module und entsprechend die MSM von unterschiedlichen Herstellern stammen können, ist es möglich, dass keine einheitliche Bezeichnung oder Namenskonvention vorliegt. Es ist vorstellbar, die Mapping-Informationen durch eine Bedienerabfrage mittels GUI an das AS zu übergeben, oder eine Beschreibung zur Verfügung zu stellen. Im einfachsten Fall kann es sich hierbei um eine Excel- oder CSV-Datei handeln, die dem RFID-Tag die AML-ID und den Modellnamen bzw. den Quellpfad des Modells zuordnet. Dies ist oben in Abbildung 54 durch die Tabelle veranschaulicht. Verfügt das AS über die nötigen Mapping-Informationen, können die Informationen verglichen und Änderungen identifiziert werden. Zu beachten ist, dass der Sensor S3 vor dem Umbau kein RFID-Tag bzw. kein Modul erfassen konnte. Entsprechend existiert im TM keine AML-ID für dieses Modul. Um dem AS diese Information zu

verdeutlichen, kann beispielsweise der Wert „0“ übergeben werden. Durch den Vergleich der alten AML-IDs der Module mit Verweisen auf die MSM im TM und den RFID-Tags mit Verweisen auf die potenziell neuen MSM können Änderungen letztlich durch das AS identifiziert werden.

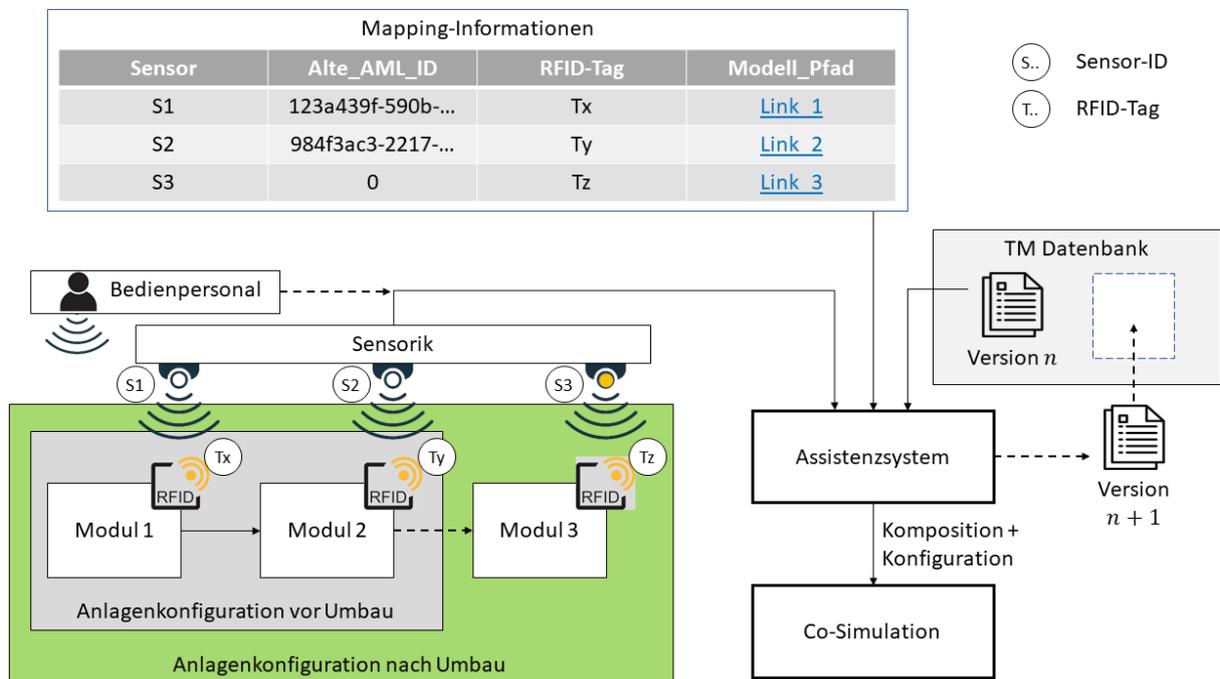


Abbildung 54: Detailskizze zur Umbaudetektion, Rekonfiguration der Co-Simulation und Dokumentation.

Nach der Detektion der Änderungen an der Anlage wird im nächsten Schritt das Gesamtmodell rekonfiguriert und die Co-Simulation erneut ausgeführt.

6.3.3 Rekonfiguration des Gesamtsystemmodells

Die Rekonfiguration des Gesamtsystemmodells läuft weitgehend identisch zu dem in Kapitel 4 beschriebenen Vorgehen bei der initialen Komposition und Konfiguration des Gesamtsystemmodells ab. Der Hauptunterschied besteht in den zur Verfügung stehenden Informationsquellen. Nachdem die korrekten MSM identifiziert wurden, werden die zugehörigen FMUs in eine Co-Simulation geladen und entsprechend gekoppelt. Für unveränderte Module können Modellparameter und der LoD, welche im Rahmen der betriebsparallelen Adaption (vgl. Kapitel 5) bestimmt wurden, übernommen werden. Für neu eintretende Module wird das MSM im niedrigsten LoD initialisiert, und die in der *ModelDescription*-Datei der FMU hinterlegten Modellparameter werden verwendet. Für die neue Anlagenkonfiguration bietet es sich an, die betriebsparallele Adaption erneut durchzuführen, um Parameter und LoD neu zu bestimmen. Zudem sollten Parameter und LoD in einer Datenbank gespeichert werden, falls bestimmte Anlagenkonfigurationen öfter genutzt werden. Abschließend müssen noch die Änderungen des TM dokumentiert werden.

6.3.4 Dokumentation der Änderungen

Um die vorgenommenen Änderungen zu dokumentieren wird, wie in Abschnitt 6.2 dargelegt, die *AutomationML Engine* verwendet. Diese ermöglicht es, das bestehende TM zu durchsuchen und zu bearbeiten,

z. B. Elemente umzubenennen und neue IDs zu vergeben. Der Aufbau eines TM ist identisch zu dem, welcher in Abschnitt 4.4.1 für die Lebenszyklusphase Inbetriebnahme beschrieben wurde. Im Fall von neuen Modulen, die bisher noch nicht Teil der Anlage waren, muss zunächst ein *InternalElement* (IE) erzeugt werden, welches das Modul repräsentiert. Diesem IE werden *ExternalInterfaces* (EI) entsprechend den real vorhandenen Schnittstellen des Moduls hinzugefügt. Ein zusätzliches EI wird genutzt, um auf das MSM zu verweisen. Die Kopplungen der einzelnen Module werden durch *InternalLinks* (IL) der entsprechenden EI abgebildet. Für entfernte Module müssen die zugehörigen IE mit den damit verbundenen EI im bestehenden TM gelöscht werden. Zusätzlich werden alle IL von und zu diesen EI gelöscht. Im Fall von getauschten Modulen müssen die IE umbenannt und neue IDs vergeben werden. Zudem müssen die Referenzen auf die MSM angepasst werden. Das resultierende TM wird abschließend unter neuer fortlaufender Versionsnummer $n + 1$ gespeichert. So kann die Umbauhistorie abgebildet werden (vgl. hierzu TM Bibliothek rechts in Abbildung 54).

6.4 Prototypische Implementierung

In Listing 7 ist ein Code-Ausschnitt in C# dargestellt, der die Verwendung der AML Engine exemplarisch aufzeigt. In diesem Beispiel wird über eine vorgegebene AML-ID nach einem spezifischen IE im TM gesucht mit dem Ziel, den Namen und die ID des IE zu ändern. Dieses IE könnte z. B. ein bestehendes Modul repräsentieren, welches durch ein neu in die Anlagenkonfiguration eingetretenes Modul getauscht wird. In Zeile 2 wird hierfür zunächst die AML Engine eingebunden. In Zeile 6 wird das Topologiemodell mit dem Dateinamen „myTopologyModel“ geladen und in die Variable „caexDocument“ gespeichert. Anschließend wird in Zeile 9 die *InstanceHierarchy* des CAEX Dokuments in eine Variable „IH“ gespeichert. In Zeile 9 wird noch die zu suchende ID in die Variable „searchedID“ gespeichert. Anschließend folgt in Zeile 15 bis Zeile 26 eine Schleife, in der alle IE der *InstanceHierarchy* durchlaufen werden. In einer if-Schleife von Zeile 17 bis Zeile 25 werden die IDs der durchlaufenen IE mit der gesuchten ID verglichen. Bei einem Fund wird in Zeile 18 das gesuchte IE in eine Variable „searchedIE“ gespeichert und abschließend in Zeile 21 zuerst der Name und in Zeile 24 die ID des IE überschrieben.

```
1 // AML Engine einbinden
2 using AML.Engine.CAEX;
3
4 // Topologiemodell "myTopologyModel" laden und in Variable "caexDocument"
5 // speichern
6 var caexDocument = CAEXDocument.LoadFromFile("myTopologyModel.aml");
7
8 // InstanceHierarchy in var abspeichern
9 var IH = caexDocument.CAEXFile.InstanceHierarchy[0];
10
11 // Gesuchte ID in Variable speichern
12 var searchedID = "1234";
13
14 // aller Knoten und Subknoten in TM durchlaufen
15 foreach (XmlNode n in node.ChildNodes) {
16     // Abgleich aller IDs der XML Knoten mit gesuchter ID
17     if (n.getID() == searchedID) {
18         var searchedIE = IH.FindCaexObjectFromId<InternalElement>(searchedID);
19
20         // Bezeichner des gesuchten InternalElements ändern
21         searchedIE.SetAttributeName("newName");
22
23         // ID des gesuchten InternalElements ändern
24         searchedIE.AssignNewGuidAsID();
25     }
26 }
```

Listing 7: Exemplarischer Code-Ausschnitt in C# für die Verwendung der AML Engine.

6.5 Teilkonzeptvalidierung

In diesem Abschnitt wird das in Abschnitt 6.3 vorgestellte Teilkonzept zur bedarfsorientierten Rekonfiguration einer Co-Simulation validiert. Hierzu werden Umbauszenarien der beiden in Abschnitt 2.3 vorgestellten Produktionsanlagen herangezogen.

6.5.1 Validierungsszenario 1: Werkstück- und prozessabhängiger Umbau bei Gleitschleifanlagen

Für dieses Validierungsszenario soll der bereits in Abschnitt 2.3.1 vorgestellte Fall betrachtet werden, dass größere und entsprechend schwerere WS mit der Gleitschleifanlage bearbeitet werden sollen. Aus vorgegebenen Gründen, z. B. den Anforderungen an die WS-Oberfläche oder die maximale Prozesszeit, können die Prozessparameter, beispielweise die Eintauchtiefe oder die Drehzahlen des Behälters und der WS, nicht angepasst werden. Bei höherer WS-Masse und gleichbleibenden Prozessparametern erhöhen sich folglich die maximal auftretenden Prozesskräfte. Es wird angenommen, dass die auftretenden Prozesskräfte außerhalb der Arbeitsbereiche der bisher verbauten Antriebseinheiten liegen. Entsprechend werden die Module der Antriebseinheiten ersetzt (vgl. Abbildung 14). In diesem Validierungsszenario wird somit der Fall eines Modultauses betrachtet. Die restliche Anlagenkonfiguration bleibt unverändert.

In Abbildung 55 wird das TM der Anlage nach dem Umbau und der entsprechenden Umbaudokumentierung visualisiert. Links in der Abbildung ist das TM der OTEC-Anlage dargestellt. Darunter ist eine schematische Skizze der resultierenden Co-Simulation abgebildet. Die Topologie der Anlage hat sich nicht geändert. Die getauschte Antriebseinheit wird durch ein neues Physikmodell (orange hervorgehoben) repräsentiert. Entsprechend wird der Pfad zur referenzierten FMU angepasst. Dieser Zusammenhang ist rechts in der Abbildung dargestellt. Das EI *ModelReference* besitzt hierfür eine *refURI*, deren Wert vom AS um-

geschrieben wird. Der hierfür nötige Methodenaufruf der AML Engine ist unten in der Abbildung eingefügt. Die nötigen Informationen entnimmt das AS den Mapping-Informationen. Die Schnittstellen der Module, repräsentiert durch IL (blaue Verbindungslinien), bleiben bestehen.

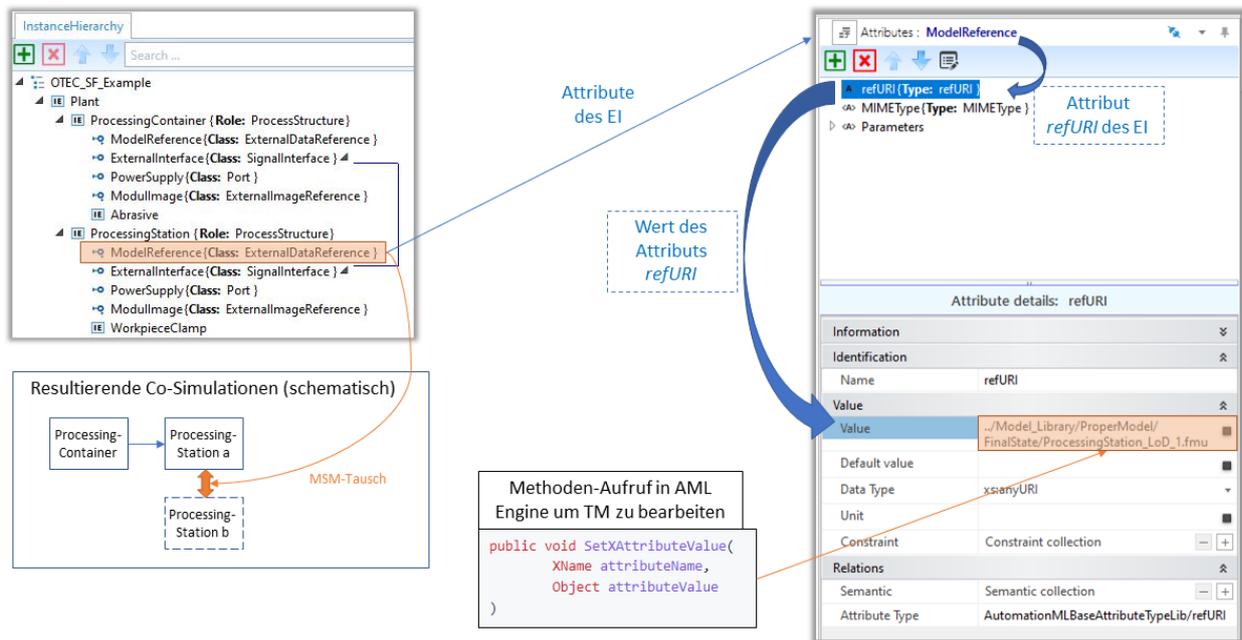


Abbildung 55: Resultierendes TM und Schema der Co-Simulation nach Umbau für OTEC Anlage.

Folglich hat das AS die Umbaumaßnahme „Modultauch“ korrekt erkannt, das Modell entsprechend rekonfiguriert und die Änderung im TM abgebildet. Entsprechend erfüllt das AS die Anforderungen A7 bis A9 für dieses Validierungsszenario. Im nächsten Abschnitt wird das zweite Validierungsszenario betrachtet.

6.5.2 Validierungsszenario 2: Prozessbedingte Elektrolytanpassung in der Galvanotechnik

In diesem Validierungsszenario wird der in Abschnitt 2.3.4 vorgestellte Fall betrachtet, dass mit der Jentner Kleingalvanik andere WS in einem angepassten Prozess bearbeitet sollen. Hierfür können in der Anlage Bearbeitungswannen ausgetauscht, deren Reihenfolge verändert, entfernt oder hinzugefügt werden. Im konkreten Beispiel soll von einem zweistufigen (zwei Elektrolytbäder) auf einen dreistufigen Beschichtungsprozess umgestellt werden. In Abbildung 56 wird links das TM der Anlage nach dem Umbau und der entsprechenden Umbaudokumentierung dargestellt. Die resultierende Co-Simulation ist rechts unten schematisch abgebildet. Aus Modellierungs- und Dokumentationsicht ist der Umbau der Jentner-Anlage umfangreicher als der im vorherigen Validierungsfall. Es wird der bestehende Beschichtungsprozess (vergoldet und anschließend versilbern) um einen weiteren Schritt, das Rhodinieren (Modul grün hervorgehoben), erweitert. Das AS lädt hierfür ein zusätzliches Galvanikbad-Modell. Die beiden Schnittstellen der Bäder werden nach dem gleichen Schema gekoppelt (WS-Transfer und Flüssigkeitsverschleppung durch WS) wie die bereits vorhandenen Bäder. Die AML Engine muss hierfür zunächst das IE, welches das Modul repräsentiert, anlegen. Anschließend können diesem die EI zugewiesen werden. Der Pfad für das MSM wird analog zum ersten Validierungsfall im EI *ModelReference* hinterlegt. Die EI, welche die MSM-Schnittstellen repräsentieren, benötigen die Attribute *Direction* und *Type* (rechts in Abbildung 56 dargestellt). Analog zum vorherigen Validierungsfall kann über den Befehl *SetXAttributeValue* hierüber die Kausalität

(z. B. *Input*) und der Datentyp (z. B. *real*) vorgegeben werden. Abschließend müssen die neuen EI noch mit denen des vorherigen Bades (Silberbad) über IL gekoppelt werden. Die Wannen für spezielle Elektrolyte können mit RFID-Tags ausgestattet werden, die zusätzlich relevante Modellparameter zur Verfügung stellen. So können z. B. Parameter wie elektrochemische Zerfallsraten oder initiale Konzentrationen dem AS übermittelt werden, um so das MSM (mittig in Abbildung 56) zu initialisieren. Diese Informationen können zusätzlich in den Attributen des EI *ModelReference* dokumentiert werden.

Da in diesem Fall die Elektrolytmodelle standardisiert sind, wäre es möglich über die objektorientierten Mechanismen von AML eine Vorlage in der *SystemUnitClassLib* anzulegen, die für neue Module genutzt werden kann. So müssen nicht jedes Mal die EI und IE einzeln neu angelegt werden, sondern nur noch die Vorlagen geladen und deren Schnittstelleninformationen, wie die IDs der IL, angepasst werden. Der Einsatz einer Vorlagenbibliothek in AML kann den Automatisierungsgrad erhöhen, ist aber nur sinnvoll, wenn gleiche oder zumindest ähnliche Module häufiger zum Einsatz kommen. Anstatt ganze Module als *SystemUnitClass* abzubilden, gibt es auch die Möglichkeit, Schnittstellen in der *InterfaceClassLib* oder Attribute in der *AttributeClassLib* für die wiederkehrende Verwendung abzulegen. So können auch mehrfach auftretende Teilaspekte als Vorlagen genutzt werden.

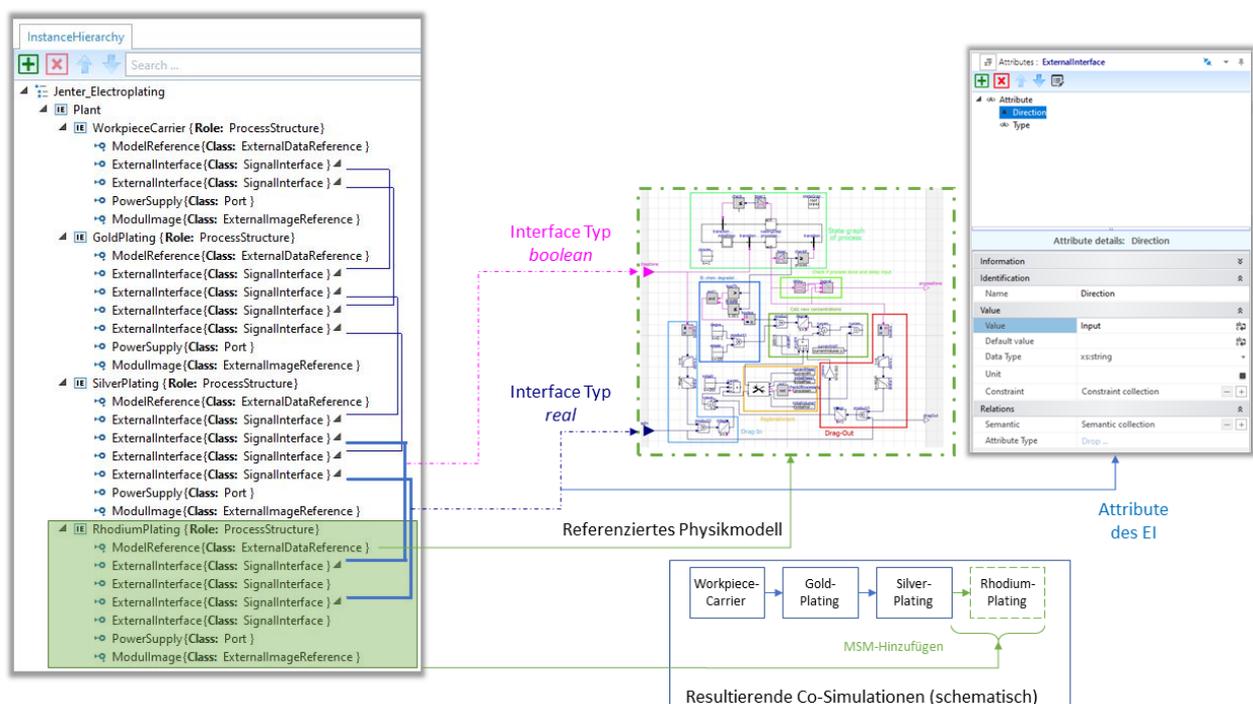


Abbildung 56: Resultierendes TM und Schema der Co-Simulation nach Umbau für Jentner Anlage.

Auch die Umbaumaßnahme „Hinzufügen eines Moduls“ hat das AS korrekt erkannt, das Modell entsprechend rekonfiguriert und die Änderung im TM abgebildet. Für dieses Validierungsszenario erfüllt das AS ebenfalls die Anforderungen A7 bis A9. Abschließend folgt das Zwischenfazit für dieses Kapitel.

6.6 Zwischenfazit

In diesem Kapitel wurde das bestehende Konzept für die automatische Modellgenerierung und -adaption im Bereich der diskreten Fertigung um Funktionen für die Lebenszyklusphase Umbau erweitert. Hierfür

werden sensorbasierte Methoden für die Erkennung von umbaubedingten Änderungen entsprechend Anforderung A7 angenommen. Die Sensordaten werden mittels Mapping-Informationen den MSM der entsprechenden Module zugeordnet. Um Anforderung A8 (die automatische Rekonfiguration) zu erfüllen, können im Anschluss die Mechanismen des AS, welche bereits zur Inbetriebnahmephase Einsatz finden, erneut für die automatische Modellgenerierung genutzt werden. Abschließend werden die identifizierten Änderungen in einer neuen Version des TM dokumentiert und in einer Datenbank gespeichert. Somit wird auch Anforderung A9 (die automatische Dokumentation) erfüllt. Das Teilkonzept wurde anhand von Umbauszenarien an zwei Anlagen aus unterschiedlichen Bereichen der Fertigung validiert. Es konnte gezeigt werden, dass die manuellen Tätigkeiten, die nach bisherigem Stand der Technik mit M&S einhergehen, für die Lebenszyklusphase Umbau einer Produktionsanlage gesenkt werden können.

Hinsichtlich der vier in Abschnitt 1.3 formulierten Forschungsfragen ergibt sich folgende Einschätzung. Zusammenfassend erfüllt die dritte Konzeptstufe des implementierten Prototyps die in Abschnitt 6.1 formulierten Anforderungen zur bedarfsorientierten Rekonfiguration einer Co-Simulation für die Umbauphase. Das AS ist in der Lage festzustellen, ob sich die reale Systemkonfiguration geändert hat, die automatische Modellgenerierung unter Einbeziehung der Umbauänderungen erneut durchzuführen und durchgeführte Änderungen an der Anlagentopologie automatisch zu dokumentieren. Das beantwortet die erste Forschungsfrage für das Teilkonzept. Bezüglich Forschungsfrage 2 muss, neben den in den beiden vorherigen Kapiteln genannten Voraussetzungen, die Anlage zusätzlich über die nötige Sensorik zur Erkennung der Module verfügen oder das AS bei Bestandsanlagen eine Bedienereingabe zulassen. Die daraus resultierenden Daten stellen, zusammen mit den Mapping-Informationen, die benötigten Informationen entsprechend Forschungsfrage 3 dar. Vor der Beantwortung der letzten Forschungsfrage hinsichtlich des Automatisierungsgrades der Lösung wird auch in diesem Kapitel zunächst die Ausprägung des AS diskutiert. Entsprechend der in Abschnitt 2.2.2 vorgenommenen Klassifizierung von AS ist dieses für die betrachtete Lebenszyklusphase, wie bereits in Kapitel 4, als Ausführungsassistenzsystem (AAS) einzustufen. Sofern eine Sensorik zur Umbauererkennung vorhanden ist und eine Schnittstelle zum AS existiert, muss der Bediener lediglich die Mapping-Informationen bereitstellen. Sämtliche Modellierungs- und Simulationsaufgaben werden vom AS, analog zu Kapitel 4, vollautomatisch ausgeführt. Gleiches gilt für die Dokumentation und die Versionsverwaltung dieser. Die automatische Versionierung und Abbildung der Umbauhistorie ermöglichen es, beim Auftreten von Produktionsanforderungen und -randbedingungen, die in der Vergangenheit in gleicher oder ähnlicher Form vorlagen, zielgerichtet alte Anlagentopologien zu laden und Simulationsstudien durchzuführen. Mit wachsender Umbauhistorie könnten dem Bediener beim Auftreten neuer Produktionsanforderungen und -randbedingungen vom AS auch Topologien vorgeschlagen werden. Dies würde das reine AAS um die Fähigkeiten eines EAS erweitern.

Bezüglich der in Abschnitt 2.2.5 definierten Automatisierungslevel ergibt sich, je nach Ausprägung, folgende Einschätzung. Sollte keine Sensorik zur Verfügung stehen, um Umbaumaßnahmen wahrzunehmen, ist das AS für diese Lebenszyklusphase mit dem Level 2 „Teilautomatisierung“ eingestuft. Grund ist, dass der Bediener die Wahrnehmung übernimmt. Sollte die nötige Sensorik vorhanden sein, aber die Mapping-Informationen aufwendig manuell erstellt werden, ist das AS mit dem Level 3 „Bedingte Automatisierung“ einzustufen. Das AS kann den Umbau wahrnehmen, jedoch nicht ausreichend interpretieren, um die Rekonfiguration der Co-Simulation vorzunehmen. Sollten Sensorik vorhanden und Mapping-Informationen nicht nötig sein, liegt das AS im Level 5 „Vollautomatisierung“ vor. Alle Aufgaben von der Umbauererkennung über die Rekonfiguration bis zur Dokumentation werden selbstständig vom AS erfüllt.

Eine offene Fragestellung bei der Rekonfiguration der Gesamtsystemmodelle ist zum aktuellen Zeitpunkt noch, wie Kopplungsinformationen erfasst werden können, die nicht mechanischer Art sind. Vorstellbar ist z. B., dass ein Physikmodell zur Simulation die Umgebungstemperatur verwendet. Hierfür würde es keine mechanische Kopplung geben, die durch Sensorik erfasst und in die Co-Simulation überführt werden könnte. Darüber hinaus wäre es sinnvoll, eine Standardisierung der Mapping-Informationen anzustreben, um zukünftig den Grad an Automatisierung zu erhöhen. Es entwickeln sich bereits unabhängige proprietäre Lösungen wie Marktplätze, über welche Komponenten- und Modulmodelle zur Verfügung gestellt werden. Durch frühzeitige Gremienarbeit kann der Standardisierungsaufwand reduziert werden. Zu erwähnen sind an dieser Stelle z. B. die Arbeiten des SG AAS Teilmodelle Simulation der Plattform Industrie 4.0 [KSE+20, JKS+22].

Nach der Vorstellung des letzten Konzeptteils folgt im nächsten Kapitel die Gesamtzusammenfassung und der Ausblick der Arbeit.

7 Zusammenfassung und Ausblick

Im Rahmen der vorliegenden Arbeit wurde ein Konzept für die (teil-)automatische Modellgenerierung, -adaption und -rekonfiguration für modulare Maschinen und Anlagen im Bereich der diskreten Fertigung entworfen. Der Neuheitsgrad der Arbeit begründet sich darin, dass erstmals aus Modellierungs- und Simulationsicht der komplette Lebenszyklus von modularen Produktionsanlagen durch ein AS unterstützt wird. Es wurden ein neues Gesamtkonzept entwickelt und bestehende Lösungen für die einzelnen Lebenszyklusphasen substantiell, auf Basis abgeleiteter Anforderungen aus der Analyse des aktuellen Vorgehens im Model-Based Systems Engineering, erweitert. Die wesentlichen Aspekte werden nachfolgend im Abschnitt 7.1 zusammengefasst. Der Ausblick in Abschnitt 7.2 schließt die vorliegende Arbeit ab. Es wird aufgezeigt, welche Optimierungsmöglichkeiten und Grenzen das vorgestellte Lösungskonzept besitzt und welche Potentiale sich für zukünftige Forschungsfragen aus den Erkenntnissen dieser Arbeit ergeben.

7.1 Zusammenfassung

Ziel der Arbeit war die Ausarbeitung eines Lösungskonzepts für die (teil-)automatische Generierung, Adaption und (Re-)Konfiguration von Modellen für modulare Produktionsmaschinen und -anlagen. Im Rahmen der Arbeit wurden zusätzlich die Fragen beantwortet, welche Voraussetzungen erfüllt werden müssen, um diese (Teil-)Automatisierung zu erreichen, welche Informationen benötigt werden und welcher Automatisierungsgrad erzielt werden kann. Aus dem formulierten Handlungsbedarf wurden insgesamt neun Anforderungen, die verschiedene Lebenszyklusphasen adressieren, abgeleitet. Eine Analyse des Standes von Wissenschaft und Technik ergab, dass bereits diverse Ansätze existieren, die Lösungen zur Automatisierung von Abläufen im Bereich M&S vorschlagen. Diese Einsatzmöglichkeiten wurden hinsichtlich der zuvor definierten Anforderungen untersucht. Es konnte festgestellt werden, dass keiner der Lösungsansätze alle Anforderungen vollumfänglich erfüllt. Der Hauptgrund hierfür ist, dass keine der Lösungen mit dem Ziel entworfen wurde, den kompletten Lebenszyklus der Anlage abzudecken. Der Großteil der Lösungen deckt eine Lebenszyklusphase ab, vereinzelt wird noch übergreifend eine zweite Lebenszyklusphase behandelt. Aus diesem Defizit ergab sich der Handlungsbedarf, ein neues lebenszyklusübergreifendes Lösungskonzept zu entwickeln. Viele Aspekte konnten aus den bestehenden Lösungen übernommen und zu einem Gesamtkonzept kombiniert werden. Bei dem Gesamtkonzept handelt es sich, entsprechend der Lebenszyklusphasen Inbetriebnahme, Betrieb und Umbau, um ein dreistufiges Konzept. Die jeweiligen Konzeptstufen bauen dabei auf den vorherigen auf.

Im ersten Konzeptionierungsschritt musste zunächst festgelegt werden, welche Beschreibungen der Anlage als Informationsquellen zur Verfügung stehen und in welcher Ausprägung die MSM vorliegen. Die Ausprägung hat maßgeblich die Wahl der Co-Simulationsart beeinflusst. Auf Basis der gewählten Beschreibung und MSM mit standardisierten Schnittstellen konnte der Ablauf der Generierung und Konfiguration des Gesamtmodells realisiert werden. Aus der Beschreibung wurden Informationen über die Anlagentopologie abgeleitet und das Gesamtmodell entsprechend aus den MSM gekoppelt. Im nächsten Schritt wurde das bestehende Teilkonzept für die automatische Modellgenerierung um Funktionen für die betriebsparallele Modelladaption erweitert. Hierfür wurde ein mehrkriterielles Optimierungsproblem for-

muliert, welches mittels evolutionärer Algorithmen gelöst werden kann. Das Ergebnis dieses Optimierungsproblems ist eine Menge an Lösungen, welche unterschiedliche Güteeigenschaften hinsichtlich Abweichung der Mess- und Simulationswerte sowie der Detaillierungsgrade der Modelle und damit der benötigten Rechenzeit haben. Der Bediener des AS kann letztlich entscheiden, welche Lösung(en) für den weiteren Simulationsbetrieb sinnvoll sind. Im letzten Schritt wurde das Teilkonzept um Funktionen für die Lebenszyklusphase Umbau erweitert. Hierfür wurden sensorbasierte Methoden für die Erkennung von umbaubedingten Änderungen angenommen. Die Sensordaten wurden mittels Mapping-Informationen den MSM der entsprechenden Module zugeordnet. Im Anschluss können die Mechanismen des AS, welche bereits zur Inbetriebnahmephase Einsatz finden, für die automatische Modellgenerierung genutzt werden. Abschließend wurden die identifizierten Änderungen in einer neuen Version des TM dokumentiert.

Zur Überprüfung des vorgestellten Gesamtkonzepts hinsichtlich der definierten Anforderungen wurde das prototypisch implementierte AS auf zwei modulare Produktionsanlagen aus unterschiedlichen Bereichen der Fertigung angewandt. Die Validierung jedes Teilkonzepts erfolgte hierbei isoliert betrachtet. Es konnte gezeigt werden, dass das AS für beide Systeme über deren kompletten Lebenszyklus angewandt werden kann und dadurch der manuelle Modellierungsaufwand sowie die starke Abhängigkeit vom individuellen Wissen der Modellierungsexperten reduziert wird. Beim AS-Prototyp handelt es sich um eine Kombination aus einem AAS (in allen Lebenszyklusphasen) und einem EAS (in der Betriebsphase). Das Automatisierungslevel des AS variiert über die drei Lebenszyklusphasen. Unter der Annahme, dass Sensorik für die Umbauerfassung zur Verfügung steht, kann für alle Phasen durchgängig mindestens das Level 3 „*Bedingte Automatisierung*“ erreicht werden. Das AS, welches eine Implementierung des Lösungskonzepts darstellt, erfüllt dahingehend alle definierten Anforderungen. Die Erfüllung aller Anforderungen führt zur Erfüllung der Zielsetzung der Arbeit. Weiter konnten im Rahmen der Kapitel zu den jeweiligen Teilkonzepten die aufgestellten Forschungsfragen bezüglich der Voraussetzungen, benötigten Informationen und des erreichten Automatisierungsgrades beantwortet werden.

7.2 Ausblick

Es ist zu erwarten, dass der vorgestellte Ansatz zukünftig eine Lösung für die in Abschnitt 2.2.1 beschriebenen Herausforderungen und Probleme beim Einsatz von M&S im modularen Maschinen- und Anlagenbau darstellt. Im Rahmen der Implementierung und Validierung wurde allerdings auch deutlich, dass trotz des AS-Einsatzes keinesfalls komplett auf M&S-Experten verzichtet werden kann. Als Beispiel ist die Wahl der Designvariablen und deren Grenzen für die Adaption zu nennen. So konnte lebenszyklusübergreifend auch nicht durchgängig das Automatisierungslevel 5 „*Vollautomatisierung*“ erreicht werden. Das vorgestellte Lösungskonzept und der Einsatz des AS können allerdings zu einer Verschiebung der Tätigkeitsschwerpunkte im Bereich M&S im modularen Maschinen- und Anlagenbau beitragen und so repetitive und fehleranfällige manuelle Tätigkeiten reduzieren.

Im Rahmen zukünftiger Arbeiten sollte die Frage beantwortet werden, durch welche ergänzenden Methoden, Informationen und Werkzeuge das vorgeschlagene Konzept durchgängig auf das Automatisierungslevel 5 gehoben werden kann, um so den M&S -Workflow für modulare Anlagen zusätzlich zu verbessern. Weitere offene Fragestellungen wurden bereits im Rahmen der drei Zwischenfazits formuliert und sollen an dieser Stelle nochmals kompakt dargestellt werden. Für die Inbetriebnahmephase wurden

Modelle als Herausforderung identifiziert, die sich über mehrere Module erstrecken. Es gilt eine sinnvolle Abbildung dieser Modelle zu finden und die automatische Modellgenerierung auf diesen Fall zu erweitern. Des Weiteren ist auf die Erkenntnis von Reinhardt et al. [RWP19] zu verweisen, die zu dem Schluss kamen, dass alle untersuchten Veröffentlichungen zum Thema automatische Modellgenerierung im Bereich der diskreten Fertigung auf einem de-facto Standard wie AutomationML, CMSD, ISA-95 und SysML beruhen. Ob und inwieweit die Kombination mehrerer de-facto Standards gewinnbringend für die automatische Modellgenerierung eingesetzt werden kann, stellt einen weiteren Forschungsbedarf dar. Im Rahmen der Betriebsphase sind die beiden Herausforderungen zur Stichprobenkomplexität (*sample complexity*) und zur Berechnungskomplexität (*computational complexity*) aufgetreten. Es stellt sich die Frage, wie mit diesen beiden Problemen im Rahmen der betriebsparallelen Modelladaption umzugehen ist und ob generelle Aussagen über die Anforderungen getroffen werden können, um in allen Fällen eine Modelladaption mit dem AS realisieren zu können. Für die Umbauphase wurde als Frage formuliert, wie im Rahmen der Rekonfiguration Kopplungsinformationen erfasst werden können, die nicht mechanischer Art sind. Darüber hinaus wurde eine Standardisierung der Mapping-Informationen angestoßen, um zukünftig den Grad an Automatisierung zu erhöhen.

Anhang

Anhang A AutomationML-Modell der Gleitschleifanlage

AutomationML-Modell der Gleitschleifanlage für Kapitel 4

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <CAEXFile SchemaVersion="3.0" FileName="Otec_SF_Example.aml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.dke.de/CAEX"
  xsi:schemaLocation="http://www.dke.de/CAEX CAEX_ClassModel_V.3.0.xsd">
3 <SuperiorStandardVersion>AutomationML 2.10</SuperiorStandardVersion>
4 <SourceDocumentInformation OriginName="AutomationML Editor"
  OriginID="916578CA-FE0D-474E-A4FC-9E1719892369" OriginVersion="5.1.2.0"
  LastWritingDateTime="2019-10-25T12:11:22.5917758+02:00"
  OriginProjectID="unspecified" OriginProjectTitle="unspecified"
  OriginRelease="5.1.2.0" OriginVendor="AutomationML e.V."
  OriginVendorURL="www.AutomationML.org" />
5 <InstanceHierarchy Name="Otec_SF_Example">
6 <Version>0</Version>
7 <InternalElement Name="Plant" ID="76c76bcf-63c9-4c02-95ed-d070b6e1382c">
8 <InternalElement Name="PLC" ID="8c4d51fc-9209-432c-a01f-f3681e467e33">
9
10 <ExternalInterface Name="ModelReference"
  ID="0584adb9-62ef-4b4c-8dde-2cbc62c211fa"
  RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/Ext
  ernalDataConnector/ExternalDataReference">
11 <Attribute Name="refURI" AttributeDataType="xs:anyURI"
  RefAttributeType="AutomationMLBaseAttributeTypeLib/refURI">
12 <Value>../Model_Library/PLC/PLC_LoD_1.fmu</Value>
13 </Attribute>
14 <Attribute Name="MIMETYPE" AttributeDataType="xs:string"
  RefAttributeType="AutomationMLBaseAttributeTypeLib/MIMETYPE" />
15 <Attribute Name="Parameters" AttributeDataType="xs:string">
16 <Attribute Name="LoadingTime.freqHz" AttributeDataType="xs:double"
  RefAttributeType="AttributeTypeLib/Parameter">
17 <Value>6.0</Value>
18 <Attribute Name="Min" AttributeDataType="xs:double">
19 <Value>4.0</Value>
20 </Attribute>
21 <Attribute Name="Max" AttributeDataType="xs:double">
22 <Value>8.0</Value>
23 </Attribute>
24 </Attribute>
25 <Attribute Name="LoadingTime.amplitude" AttributeDataType="xs:double"
  RefAttributeType="AttributeTypeLib/Parameter">
26 <Value>1.0</Value>
27 <Attribute Name="Min" AttributeDataType="xs:double">
28 <Value>0.8</Value>
29 </Attribute>
30 <Attribute Name="Max" AttributeDataType="xs:double">
31 <Value>1.2</Value>
32 </Attribute>
33 </Attribute>
34 </Attribute>
35 </ExternalInterface>
36 <ExternalInterface Name="ExternalInterface"
  ID="46998bd9-621e-4b7b-8ba1-8a27ab7b6d19"
  RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/Comm
  unication/SignalInterface">
37 <Attribute Name="Attribute" AttributeDataType="xs:string">
38 <Attribute Name="Direction" AttributeDataType="xs:string">
39 <Value>Output</Value>
40 </Attribute>
41 <Attribute Name="Type" AttributeDataType="xs:string">
42 <Value>Real</Value>
43 </Attribute>
44 </Attribute>
45 </ExternalInterface><ExternalInterface Name="ExternalInterface"
  ID="c9dc0d77-de44-4b0c-a43a-d4bacf06763c"
  RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/Comm
  unication/SignalInterface">
46 <Attribute Name="Attribute" AttributeDataType="xs:string">
47 <Attribute Name="Direction" AttributeDataType="xs:string">
48 <Value>Output</Value>

```

```

49 </Attribute>
50 <Attribute Name="Type" AttributeDataType="xs:string">
51 <Value>Real</Value>
52 </Attribute>
53 </Attribute>
54 </ExternalInterface><ExternalInterface Name="ExternalInterface"
ID="669f4636-4840-4eb3-afab-01c92909981d"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/Comm
unication/SignalInterface">
55 <Attribute Name="Attribute" AttributeDataType="xs:string">
56 <Attribute Name="Direction" AttributeDataType="xs:string">
57 <Value>Output</Value>
58 </Attribute>
59 <Attribute Name="Type" AttributeDataType="xs:string">
60 <Value>Real</Value>
61 </Attribute>
62 </Attribute>
63 </ExternalInterface><ExternalInterface Name="PowerSupply"
ID="c3ebd19a-c8d0-4464-bfff-ff743334ed3b"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/Port
"><Attribute Name="Direction" AttributeDataType="xs:string"
RefAttributeType="AutomationMLBaseAttributeTypeLib/Direction">
64 <Constraint Name="AllowedValues">
65 <NominalScaledType>
66 <RequiredValue>In</RequiredValue>
67 <RequiredValue>Out</RequiredValue>
68 <RequiredValue>InOut</RequiredValue>
69 </NominalScaledType>
70 </Constraint>
71 </Attribute><Attribute Name="Cardinality"
RefAttributeType="AutomationMLBaseAttributeTypeLib/Cardinality">
72 <Attribute Name="MinOccur" AttributeDataType="xs:unsignedInt" />
73 <Attribute Name="MaxOccur" AttributeDataType="xs:unsignedInt" />
74 </Attribute><Attribute Name="Category" AttributeDataType="xs:string"
RefAttributeType="AutomationMLBaseAttributeTypeLib/Category"
/></ExternalInterface><ExternalInterface Name="ModulImage"
ID="0276598f-aaa9-469b-977f-58fa86fa9460"
RefBaseClassPath="InterfaceClassLib/ExternalImageReference"><Attribute
Name="refURI" AttributeDataType="xs:anyURI"
RefAttributeType="AutomationMLBaseAttributeTypeLib/refURI">
75 <Value>../Modulimages/PLC.jpg</Value>
76 </Attribute></ExternalInterface>
77 <InternalElement Name="DecentralPeriphery"
ID="05319a6c-5266-4790-8a41-8694a2b0b72e">
78 <InternalElement Name="ET200SP" ID="df6f431f-2a61-44d3-b9bd-bff07f7adbc2" />
79 </InternalElement>
80 <RoleRequirements
RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Struct
ure/ProcessStructure" />
81 </InternalElement>
82 <InternalElement Name="ProcessingContainer"
ID="24b34f9b-5437-424c-bd07-5c8befcf3ab5">
83
84 <ExternalInterface Name="ModelReference"
ID="128f3fa2-bda9-4469-822e-d511f5119bf1"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/Exte
rnalDataConnector/ExternalDataReference">
85 <Attribute Name="refURI" AttributeDataType="xs:anyURI"
RefAttributeType="AutomationMLBaseAttributeTypeLib/refURI">
86
<Value>../Model_Library/ProcessingContainer/ProcessingContainer_LoD_1.fmu <
/Value>
87 </Attribute>
88 <Attribute Name="MimeType" AttributeDataType="xs:string"
RefAttributeType="AutomationMLBaseAttributeTypeLib/MimeType" />
89 </ExternalInterface><ExternalInterface Name="ExternalInterface"
ID="2c8a2885-3984-472b-a5f4-3570b1167463"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/Comm
unication/SignalInterface">
90 <Attribute Name="Attribute" AttributeDataType="xs:string">
91 <Attribute Name="Direction" AttributeDataType="xs:string">
92 <Value>Input</Value>
93 </Attribute>
94 <Attribute Name="Type" AttributeDataType="xs:string">
95 <Value>Real</Value>
96 </Attribute>
97 </Attribute>
98 </ExternalInterface><ExternalInterface Name="PowerSupply"

```

```

ID="a110936e-00aa-49b9-b7e3-ffba638d7506"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/Port"
">>Attribute Name="Direction" AttributeDataType="xs:string"
RefAttributeType="AutomationMLBaseAttributeTypeLib/Direction">
99 <Constraint Name="AllowedValues">
100 <NominalScaledType>
101 <RequiredValue>In</RequiredValue>
102 <RequiredValue>Out</RequiredValue>
103 <RequiredValue>InOut</RequiredValue>
104 </NominalScaledType>
105 </Constraint>
106 </Attribute><Attribute Name="Cardinality"
RefAttributeType="AutomationMLBaseAttributeTypeLib/Cardinality">
107 <Attribute Name="MinOccur" AttributeDataType="xs:unsignedInt" />
108 <Attribute Name="MaxOccur" AttributeDataType="xs:unsignedInt" />
109 </Attribute><Attribute Name="Category" AttributeDataType="xs:string"
RefAttributeType="AutomationMLBaseAttributeTypeLib/Category"
/></ExternalInterface><ExternalInterface Name="ModulImage"
ID="edd58c87-d075-4c7d-8cc4-bb77ea773305"
RefBaseClassPath="InterfaceClassLib/ExternalImageReference"><Attribute
Name="refURI" AttributeDataType="xs:anyURI"
RefAttributeType="AutomationMLBaseAttributeTypeLib/refURI">
110 <Value>../Modulimages/ProcessingContainer.jpg</Value>
111 </Attribute></ExternalInterface><InternalElement Name="Media"
ID="7917903c-b22d-4b15-97a6-a4dc42b0315a" />
112 <RoleRequirements
RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Struct
ure/ProcessStructure" />
113 </InternalElement>
114 <InternalElement Name="ProcessingStation"
ID="99052937-ddaf-4bc0-9f5b-ee4ff526095b">
115
116 <ExternalInterface Name="ModelReference"
ID="97d042eb-84ad-450f-a79e-ed3ddd0a7d59"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/Exte
rnalDataConnector/ExternalDataReference">
117 <Attribute Name="refURI" AttributeDataType="xs:anyURI"
RefAttributeType="AutomationMLBaseAttributeTypeLib/refURI">
118
<Value>../Model_Library/ProcessingStation/ProcessingStation_LoD_1.fmu </Val
ue>
119 </Attribute>
120 <Attribute Name="MimeType" AttributeDataType="xs:string"
RefAttributeType="AutomationMLBaseAttributeTypeLib/MimeType" />
121 </ExternalInterface>
122 <ExternalInterface Name="ExternalInterface"
ID="14bbadba-f034-405e-ad3f-09d5a08c7776"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/Comm
unication/SignalInterface">
123 <Attribute Name="Attribute" AttributeDataType="xs:string">
124 <Attribute Name="Direction" AttributeDataType="xs:string">
125 <Value>Input</Value>
126 </Attribute>
127 <Attribute Name="Type" AttributeDataType="xs:string">
128 <Value>Real</Value>
129 </Attribute>
130 </Attribute>
131 </ExternalInterface>
132 <ExternalInterface Name="ExternalInterface"
ID="9f2155f1-a993-4c36-b447-6bc958401aef"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/Comm
unication/SignalInterface">
133 <Attribute Name="Attribute" AttributeDataType="xs:string">
134 <Attribute Name="Direction" AttributeDataType="xs:string">
135 <Value>Output</Value>
136 </Attribute>
137 <Attribute Name="Type" AttributeDataType="xs:string">
138 <Value>Boolean</Value>
139 </Attribute>
140 </Attribute>
141 </ExternalInterface><ExternalInterface Name="PowerSupply"
ID="d61fe960-90b2-41b0-bd89-a05e0854a1b5"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/Port"
">>Attribute Name="Direction" AttributeDataType="xs:string"
RefAttributeType="AutomationMLBaseAttributeTypeLib/Direction">
142 <Constraint Name="AllowedValues">
143 <NominalScaledType>
144 <RequiredValue>In</RequiredValue>

```

```

145 <RequiredValue>Out</RequiredValue>
146 <RequiredValue>InOut</RequiredValue>
147 </NominalScaledType>
148 </Constraint>
149 </Attribute><Attribute Name="Cardinality"
RefAttributeType="AutomationMLBaseAttributeTypeLib/Cardinality">
150 <Attribute Name="MinOccur" AttributeDataType="xs:unsignedInt" />
151 <Attribute Name="MaxOccur" AttributeDataType="xs:unsignedInt" />
152 </Attribute><Attribute Name="Category" AttributeDataType="xs:string"
RefAttributeType="AutomationMLBaseAttributeTypeLib/Category"
/></ExternalInterface><ExternalInterface Name="ModulImage"
ID="dac38c7b-b83e-4ddf-96f6-c1f1c8d4caaa"
RefBaseClassPath="InterfaceClassLib/ExternalImageReference"><Attribute
Name="refURI" AttributeDataType="xs:anyURI"
RefAttributeType="AutomationMLBaseAttributeTypeLib/refURI">
153 <Value>../Modulimages/ProcessingStation.jpg</Value>
154 </Attribute></ExternalInterface><InternalElement Name="WorkpieceClamp"
ID="2a3f08c9-fc14-4d55-9f2b-9d0ec74a2b18" /><InternalElement
Name="PulseFinishingDrive" ID="30d414a7-3271-4586-8839-122917b36faa" />
155 <RoleRequirements
RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Struct
ure/ProcessStructure" />
156 </InternalElement>
157 <InternalElement Name="RobotLoader" ID="a74f8f11-e2cd-405d-b9b9-046a330e6208">
158
159 <ExternalInterface Name="ModelReference"
ID="9026e28f-ab70-482e-bd21-0f987507bf86"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/Exte
rnalDataConnector/ExternalDataReference">
160 <Attribute Name="refURI" AttributeDataType="xs:anyURI"
RefAttributeType="AutomationMLBaseAttributeTypeLib/refURI">
161 <Value>../Model_Library/RobotLoader/RobotLoader_LoD_1.fmu</Value>
162 </Attribute>
163 <Attribute Name="MIMETYPE" AttributeDataType="xs:string"
RefAttributeType="AutomationMLBaseAttributeTypeLib/MIMETYPE" />
164 </ExternalInterface>
165 <ExternalInterface Name="ExternalInterface"
ID="f16ae902-4020-457c-8c19-cb7b8183efc8"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/Comm
unication/SignalInterface">
166 <Attribute Name="Attribute" AttributeDataType="xs:string">
167 <Attribute Name="Direction" AttributeDataType="xs:string">
168 <Value>Input</Value>
169 </Attribute>
170 <Attribute Name="Type" AttributeDataType="xs:string">
171 <Value>Real</Value>
172 </Attribute>
173 </Attribute>
174 </ExternalInterface><ExternalInterface Name="ExternalInterface"
ID="374a0c17-f723-41e1-8439-ebc03f19449e"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/Comm
unication/SignalInterface">
175 <Attribute Name="Attribute" AttributeDataType="xs:string">
176 <Attribute Name="Direction" AttributeDataType="xs:string">
177 <Value>Input</Value>
178 </Attribute>
179 <Attribute Name="Type" AttributeDataType="xs:string">
180 <Value>Boolean</Value>
181 </Attribute>
182 </Attribute>
183 </ExternalInterface>
184 <ExternalInterface Name="PowerSupply"
ID="79c8c69e-768c-4500-b801-4e5dc0f828db"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/Port
"><Attribute Name="Direction" AttributeDataType="xs:string"
RefAttributeType="AutomationMLBaseAttributeTypeLib/Direction">
185 <Constraint Name="AllowedValues">
186 <NominalScaledType>
187 <RequiredValue>In</RequiredValue>
188 <RequiredValue>Out</RequiredValue>
189 <RequiredValue>InOut</RequiredValue>
190 </NominalScaledType>
191 </Constraint>
192 </Attribute><Attribute Name="Cardinality"
RefAttributeType="AutomationMLBaseAttributeTypeLib/Cardinality">
193 <Attribute Name="MinOccur" AttributeDataType="xs:unsignedInt" />
194 <Attribute Name="MaxOccur" AttributeDataType="xs:unsignedInt" />
195 </Attribute><Attribute Name="Category" AttributeDataType="xs:string"

```

```

RefAttributeType="AutomationMLBaseAttributeTypeLib/Category"
/></ExternalInterface><ExternalInterface Name="ModulImage"
ID="862ade21-aa7d-475a-8f4e-26454af0aa7f"
RefBaseClassPath="InterfaceClassLib/ExternalImageReference"><Attribute
Name="refURI" AttributeDataType="xs:anyURI"
RefAttributeType="AutomationMLBaseAttributeTypeLib/refURI">
196 <Value>../Modulimages/RobotLoader.jpg</Value>
197 </Attribute></ExternalInterface>
198 <RoleRequirements
RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Struct
ure/ProcessStructure" />
199 </InternalElement>
200
201
202 <InternalLink RefPartnerSideA="9f2155f1-a993-4c36-b447-6bc958401aef"
RefPartnerSideB="374a0c17-f723-41e1-8439-ebc03f19449e" Name="InternalLink1" />
203 <InternalLink RefPartnerSideA="669f4636-4840-4eb3-afab-01c92909981d"
RefPartnerSideB="2c8a2885-3984-472b-a5f4-3570b1167463" Name="InternalLink2"
/><InternalLink RefPartnerSideA="c9dc0d77-de44-4b0c-a43a-d4bacf06763c"
RefPartnerSideB="14bbadba-f034-405e-ad3f-09d5a08c7776" Name="Link"
/><InternalLink RefPartnerSideA="46998bd9-621e-4b7b-8ba1-8a27ab7b6d19"
RefPartnerSideB="f16ae902-4020-457c-8c19-cb7b8183efc8" Name="InternalLink" />
204 </InternalElement>
205 </InstanceHierarchy>
206 <InterfaceClassLib Name="AutomationMLInterfaceClassLib" ChangeMode="state">
207 <Description>Standard Automation Markup Language Interface Class Library - Part
1 Content </Description>
208 <Version>2.10</Version>
209 <Copyright>AutomationML e.V.</Copyright>
210 <InterfaceClass Name="AutomationMLBaseInterface">
211 <InterfaceClass Name="Order"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface" >
212 <Attribute Name="Direction"
RefAttributeType="AutomationMLBaseAttributeTypeLib/Direction"
AttributeDataType="xs:string">
213 <DefaultValue>InOut</DefaultValue>
214 <Constraint Name="AllowedValues">
215 <NominalScaledType>
216 <RequiredValue>In</RequiredValue>
217 <RequiredValue>Out</RequiredValue>
218 <RequiredValue>InOut</RequiredValue>
219 </NominalScaledType>
220 </Constraint>
221 </Attribute>
222 </InterfaceClass>
223 <InterfaceClass Name="Port"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface" >
224 <Attribute Name="Direction" AttributeDataType="xs:string"
RefAttributeType="AutomationMLBaseAttributeTypeLib/Direction">
225 <Constraint Name="AllowedValues">
226 <NominalScaledType>
227 <RequiredValue>In</RequiredValue>
228 <RequiredValue>Out</RequiredValue>
229 <RequiredValue>InOut</RequiredValue>
230 </NominalScaledType>
231 </Constraint>
232 </Attribute>
233 <Attribute Name="Cardinality"
RefAttributeType="AutomationMLBaseAttributeTypeLib/Cardinality">
234 <Attribute Name="MinOccur" AttributeDataType="xs:unsignedInt" />
235 <Attribute Name="MaxOccur" AttributeDataType="xs:unsignedInt" />
236 </Attribute>
237 <Attribute Name="Category" AttributeDataType="xs:string"
RefAttributeType="AutomationMLBaseAttributeTypeLib/Category" />
238 </InterfaceClass>
239 <InterfaceClass Name="PPRConnector"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface" />
240 <InterfaceClass Name="ExternalDataConnector"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface" >
241 <Attribute Name="refURI" AttributeDataType="xs:anyURI"
RefAttributeType="AutomationMLBaseAttributeTypeLib/refURI" />
242 <InterfaceClass Name="COLLADAInterface"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/Exte
rnalDataConnector">
243 <Attribute Name="refType" AttributeDataType="xs:string" />
244 <Attribute Name="target" AttributeDataType="xs:token" />
245 </InterfaceClass>
246 <InterfaceClass Name="PLCopenXMLInterface"

```

```

RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector" />
247 <InterfaceClass Name="ExternalDataReference"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector">
248 <Attribute Name="MIMETYPE" AttributeDataType="xs:string"
RefAttributeType="AutomationMLBaseAttributeTypeLib/MIMETYPE" />
249 </InterfaceClass>
250 </InterfaceClass>
251 <InterfaceClass Name="Communication"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface" >
252 <InterfaceClass Name="SignalInterface"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/Communication" />
253 </InterfaceClass>
254 </InterfaceClass>
255 </InterfaceClassLib><InterfaceClassLib
Name="InterfaceClassLib"><Version>0</Version><InterfaceClass
Name="ExternalImageReference" /></InterfaceClassLib>
256 <RoleClassLib Name="AutomationMLBaseRoleClassLib">
257 <Description>Automation Markup Language Base Role Class Library - Part
1</Description>
258 <Version>2.10.0</Version>
259 <RoleClass Name="AutomationMLBaseRole" ChangeMode="state">
260 <RoleClass Name="Group"
RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole">
261 <Attribute Name="AssociatedFacet"
RefAttributeType="AutomationMLBaseAttributeTypeLib/AssociatedFacet"
AttributeDataType="xs:string" />
262 </RoleClass>
263 <RoleClass Name="Facet"
RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole" />
264 <RoleClass Name="Resource"
RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole" />
265 <RoleClass Name="Product"
RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole" />
266 <RoleClass Name="Process"
RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole" />
267 <RoleClass Name="Structure"
RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole">
268 <RoleClass Name="ProductStructure"
RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Structure"
/>
269 <RoleClass Name="ProcessStructure"
RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Structure"
/>
270 <RoleClass Name="ResourceStructure"
RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Structure"
/>
271 </RoleClass>
272 <RoleClass Name="ExternalData"
RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole" />
273 </RoleClass>
274 </RoleClassLib>
275 <AttributeTypeLib Name="AutomationMLBaseAttributeTypeLib" ChangeMode="state">
276 <Description>Standard Automation Markup Language Attribute Type
Library</Description>
277 <Version>2.10.0</Version>
278 <Copyright>AutomationML e.V.</Copyright>
279 <AttributeType Name="AssociatedExternalValue">
280 <Attribute Name="refCAEXAttribute" AttributeDataType="xs:string" />
281 <Attribute Name="refURI" AttributeDataType="xs:string" />
282 <Attribute Name="Direction" AttributeDataType="xs:string" />
283 </AttributeType>
284 <AttributeType Name="AssociatedFacet" AttributeDataType="xs:string" />
285 <AttributeType Name="Cardinality">
286 <Attribute Name="MinOccur" AttributeDataType="xs:unsignedInt" />
287 <Attribute Name="MaxOccur" AttributeDataType="xs:unsignedInt" />
288 </AttributeType>
289 <AttributeType Name="Category" AttributeDataType="xs:string" />
290 <AttributeType Name="Direction" AttributeDataType="xs:string">
291 <DefaultValue>InOut</DefaultValue>
292 <Constraint Name="AllowedValues">
293 <NominalScaledType>
294 <RequiredValue>In</RequiredValue>
295 <RequiredValue>Out</RequiredValue>
296 <RequiredValue>InOut</RequiredValue>
297 </NominalScaledType>

```

```
298 </Constraint>
299 </AttributeType>
300 <AttributeType Name="DocLang" AttributeDataType="xs:string" />
301 <AttributeType Name="Frame" AttributeDataType="">
302 <Attribute Name="x" AttributeDataType="xs:double">
303 <DefaultValue>0</DefaultValue>
304 <Value>0</Value>
305 </Attribute>
306 <Attribute Name="y" AttributeDataType="xs:double">
307 <DefaultValue>0</DefaultValue>
308 <Value>0</Value>
309 </Attribute>
310 <Attribute Name="z" AttributeDataType="xs:double">
311 <DefaultValue>0</DefaultValue>
312 <Value>0</Value>
313 </Attribute>
314 <Attribute Name="rx" AttributeDataType="xs:double">
315 <DefaultValue>0</DefaultValue>
316 <Value>0</Value>
317 </Attribute>
318 <Attribute Name="ry" AttributeDataType="xs:double">
319 <DefaultValue>0</DefaultValue>
320 <Value>0</Value>
321 </Attribute>
322 <Attribute Name="rz" AttributeDataType="xs:double">
323 <DefaultValue>0</DefaultValue>
324 <Value>0</Value>
325 </Attribute>
326 </AttributeType>
327 <AttributeType Name="ListType" AttributeDataType="xs:string" />
328 <AttributeType Name="LocalizedAttribute" AttributeDataType="xs:string" />
329 <AttributeType Name="MimeType" AttributeDataType="xs:string" />
330 <AttributeType Name="OrderedListType" AttributeDataType="xs:string" />
331 <AttributeType Name="refURI" AttributeDataType="xs:anyURI" />
332 </AttributeTypeLib><AttributeTypeLib
Name="AttributeTypeLib"><Version>0</Version><AttributeType Name="Parameter"
AttributeDataType="xs:string" /></AttributeTypeLib>
333 </CAEXFile>
```

Listing 8: AML TM der Gleitschleifanlage für Kapitel 4.

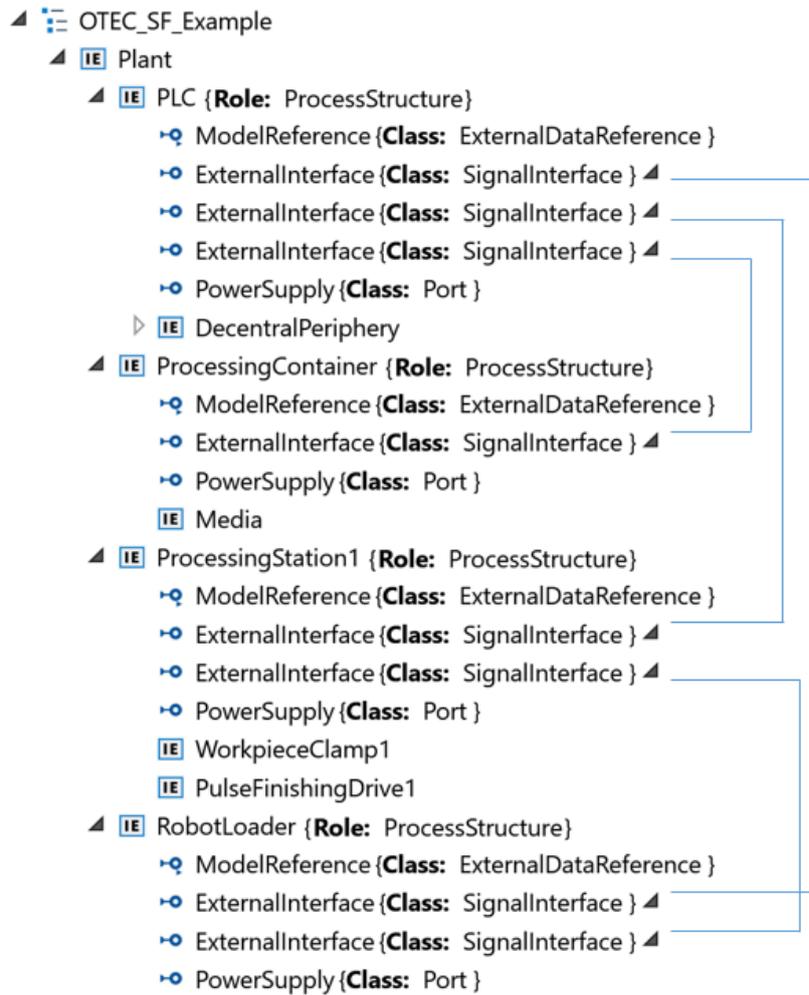


Abbildung 57: Grafische Darstellung des AML TM der Gleitschleifanlage für Kapitel 4

Anhang B AutomationML-Modell der Galvanikanlage

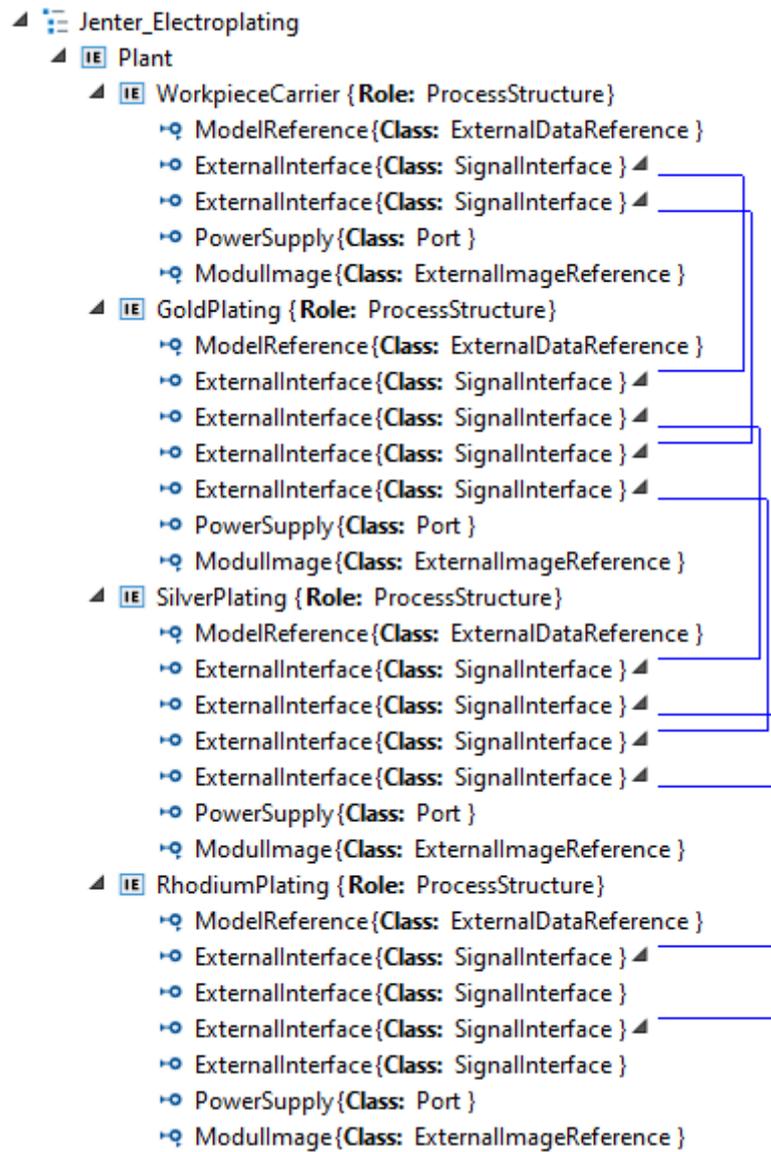


Abbildung 58: TM der Jentner Galvanikanlage.

Anhang C Listings der CustomProblem-Klasse des Assistenzsystems

```
1 package bin.classes;
2
3 import java.util.Arrays;
4 import java.util.ArrayList;
5 import java.util.List;
6 import java.util.stream.Collectors;
7
8 import java.text.DecimalFormat;
9
10 import org.moeaframework.core.Solution;
11 import org.moeaframework.core.variable.EncodingUtils;
12 import org.moeaframework.core.variable.RealVariable;
13 import org.moeaframework.problem.AbstractProblem;
14
15 import bin.classes.LevelOfDetail;
16
17 public class CustomProblem extends AbstractProblem {
18
19     // Variablen können über setter-Methode gesetzt werden
20     int numberOfModulesWithMultipleLevelsOfDetail;
21     int numberOfModulesWithoutDifferentLoDs = 0;
22
23     List<Parameter> parameterList;
24     List<Parameter> updatedParameterList;
25     List<LevelOfDetail> lodList;
26     List<LevelOfDetail> updatedLoDList;
27
28     Optimization opt;
29
30     // numberOfVariables = Summe aus Parametern und Anzahl Modulen mit
31     // mehreren Detaillierungsgraden
32     // numberOfObjectives = 2 (minimiere Abweichung und minimiere
33     // Detaillierungsgrad)
34     public CustomProblem(int numberOfVariables, int numberOfObjectives) {
35         super(numberOfVariables, numberOfObjectives);
36     }
37
38     @Override
39     public Solution newSolution() {
40         Solution solution = new Solution(getNumberOfVariables(),
41             getNumberOfObjectives());
42         System.out.println("Creating new solution (Nr.: " +
43             solution.hashCode() + ").");
44
45         // Aufteilen der Designvariablen: i=0 bis i-x sind RealVariables,
46         // i-x+1 bis i == getNumberOfVariables sind IntegerVariables
47         for (int i = 0; i < getNumberOfVariables(); i++) {
48             // Der erste Teil der Variablen sind die Parameter der Module,
49             // diese werden als Real-Werte initialisiert
50             // Aus der Parameterliste werden für den jeweiligen Parameter
51             // die Min- und Maxwerte extrahiert
52             if (i < getNumberOfVariables() -
53                 numberOfModulesWithMultipleLevelsOfDetail) {
54                 Parameter currentParameter = parameterList.get(i);
55                 double currentMin = currentParameter.getMinValue();
56                 double currentMax = currentParameter.getMaxValue();
57
58                 solution.setVariable(i, EncodingUtils.newReal(currentMin,
59                     currentMax)); // newReal(lowerBound, upperBound)
60             }
61         }
62     }
63 }
```

```
52     }
53     // Der zweite Teil der Variablen sind die Detaillierungsgrade
    // der Modulmodelle
54     else {
55         // Die max. Detaillierungsgrade der Modelle werden über
    // GUI vorgegeben und mittels Liste übergeben
56         String currentLoD = lodList.get(i-(getNumberOfVariables()
    - numberOfModulesWithMultipleLevelsOfDetail)).getLoD();
57         solution.setVariable(i, EncodingUtils.newInt(1,
    Integer.parseInt(currentLoD)));
58     }
59 }
60 return solution;
61 }
62
63 @Override
64 public void evaluate(Solution solution) {
65     System.out.println("Evaluating solution ...");
66
67     double[] x_double = new double[getNumberOfVariables() -
    numberOfModulesWithMultipleLevelsOfDetail];
68     int[] x_int = new int[numberOfModulesWithMultipleLevelsOfDetail];
69
70     boolean simulationDone = false;
71     boolean calculateDeltaDone = false;
72
73     String variablesAsString = "";
74     // wird wie parameterList initialisiert und hier upgedated,
    // ursprüngliche Liste wird erhalten für neue Lösungen
75     updatedParameterList = parameterList.stream()
76         .collect(Collectors.toList());
77     // wird wie lodList initialisiert und hier upgedated, ursprüngliche
    // Liste wird erhalten für neue Lösungen
78     updatedLoDList = lodList.stream()
79         .collect(Collectors.toList());
80
81
82     // Variablen in die Vektoren x_double und x_int aufteilen
83     for (int i = 0; i < getNumberOfVariables(); i++) {
84         // Der erste Teil der Variablen sind die Parameter der Module
85         if (i < getNumberOfVariables() -
    numberOfModulesWithMultipleLevelsOfDetail) {
86             x_double[i] = EncodingUtils.getReal(solution.getVariable(i));
87             // updatedParameterList mit neuen Werten updaten
88             updatedParameterList.get(i).setInitValue(x_double[i]);
89             variablesAsString += String.format("%.4f", x_double[i])+" ";
90         }
91         // Der zweite Teil der Variablen sind die Detaillierungsgrade der
    // Modulmodelle
92         else {
93             int tempLoD = EncodingUtils.getInt(solution.getVariable(i));
94             x_int[i-(getNumberOfVariables() -
    numberOfModulesWithMultipleLevelsOfDetail)] = tempLoD;
95             variablesAsString += x_int[i-(getNumberOfVariables() -
    numberOfModulesWithMultipleLevelsOfDetail)] + " ";
96             // Liste updaten
97             String lodAsString = Integer.toString(tempLoD);
98             updatedLoDList.get(i-(getNumberOfVariables() -
    numberOfModulesWithMultipleLevelsOfDetail)...
    ).setLoD(lodAsString);
99
100     }
101 }
```

```
102
103     double f1 = 0.0; // Funktionswert der Lösung hinsichtlich
                          Zielfunktion 1
104     int f2 = 0;      // Funktionswert der Lösung hinsichtlich
                          Zielfunktion 2
105
106     /*
107     * Mittels x können nun Sim.skripte upgedated werden
108     --> Teilmenge von x liefert f2
109     * Anschließend muss Simulation ausgeführt werden
110     * Mit Sim.- und Messwerten kann neues Delta berechnet werden
111     --> liefert f1
112     */
113     // Funktionsauswertung für f2
114     for (int j = 0; j < x_int.length; j++) {
115         f2 += x_int[j];
116     }
117     f2 += numberOfModulesWithoutDifferentLoDs; // Module, die
118     ausschließlich LoD = 1 haben werden mitberücksichtigt
119
120     // Skript updaten und Simulation ausführen
121     opt.updateSimScript(updatedParameterList, updatedLoDList);
122     updatedParameterList.clear();
123     updatedLoDList.clear();
124
125     simulationDone = opt.runSimScript();
126     calculateDeltaDone = opt.calculateDelta();
127
128     while (true) {
129         if (simulationDone && calculateDeltaDone) {
130             // Funktionsauswertung für f1
131             f1 = opt.objectiveOne(opt.getCurrentDelta());
132
133             // Funktionswerte für Lösung setzen
134             solution.setObjective(0, f1);
135             solution.setObjective(1, f2);
136
137             DecimalFormat df = new DecimalFormat();
138             df.setMaximumFractionDigits(1);
139             System.out.println("Solution " + solution.hashCode() +
140                 " has variable composition: \n\t" +
141                 "(" + variablesAsString.substring(0,
142                     variablesAsString.length()-2) + ") " + "\n\t" +
143                 "and objective function quality: \n\t" +
144                 df.format(f1) + " for f_1 and \n\t" +
145                 f2 + " for f_2.");
146             break;
147         }
148         else {
149             }
150     }
151
152     public void setNumberOfModulesWithMultipleLevelsOfDetail(int
153         numberOfModulesWithMultipleLevelsOfDetail) {
154         this.numberOfModulesWithMultipleLevelsOfDetail =
155             numberOfModulesWithMultipleLevelsOfDetail;
156     }
157
158     public void setParameterList(List<Parameter> parameterList) {
159         this.parameterList = parameterList;
160     }
161 }
```

```
156
157 public void setLoDList(List<LevelOfDetail> lodList) {
158     this.lodList = new ArrayList<>();
159     for (LevelOfDetail lod : lodList) {
160         if (!lod.getLoD().equalsIgnoreCase("1")) {
161             this.lodList.add(lod);
162         }
163         else {
164             numberOfModulsWithoutDifferentLoDs++;
165         }
166     }
167 }
168
169 public void setOptimization(Optimization opt) {
170     this.opt = opt;
171 }
172
173 }
```

Listing 9: Formulierung der Adaptionaufgabe in CustomProblem-Klasse.

Literaturverzeichnis

- [ABD+16] Anderl, R.; Bauer, K.; Diegner, B. et al.: Aspekte der Forschungsroadmap in den Anwendungsszenarien. Ergebnispapier der Plattform Industrie 4.0, Bundesministerium für Wirtschaft und Energie (BMWi) (Hrsg.), 2016.
- [AC18] AutomationML consortium: *Whitepaper AutomationML Edition 2.1 Part 1 – Architecture and General Requirements*. 2018.
- [AH20] Aschoff D.; Heitmann J.: *Klassische Risiken im Überblick*. In: Mahnke A.; Rohlf T. (Hrsg.) *Betriebliches Risikomanagement und Industrieversicherung*. Springer Gabler, Wiesbaden, 2020.
- [AIR18] Akinnikawe, O.; Lyne, S.; Roberts, J.: *Synthetic Well Log Generation Using Machine Learning Techniques*. In Unconventional Resources Technology Conference, 2018.
- [AJS+18] Ashtari, B. T.; Jazdi, N.; Schloegl, W. et al.: *Consistency check to synchronize the Digital Twin of manufacturing automation based on anchor points*. In *Procedia CIRP Volume 72*, S. 159-164, 2018.
- [BAR11] Barth, M.: *Automatisch generierte Simulationsmodelle verfahrenstechnischer Anlagen für den Steuerungstest*. Dissertation, Helmut-Schmidt-Universität / Universität der Bundeswehr Hamburg, Institut für Automatisierungstechnik, 2011.
- [BBC16] Brewer, M. J.; Butler, A.; Cooksley, S. L.: *The relative performance of AIC, AICC and BIC in the presence of unobserved heterogeneity*. In *Methods Ecol Evol* 7, S. 679-692, 2016.
- [BBH+19] Bernhausen, J.; Bloch, H.; Hoernicke, M. et al.: *Plug & Produce auf dem Sprung in den Markt*. In *atp edition*, Vol. 61 (1-2), 56-69, 2019.
- [BCH04] Benesty, J.; Chen, J.; Huang, Y.: *Time-delay estimation via linear interpolation and cross correlation*. In *IEEE Transactions on Speech and Audio Processing*, 12(5), S. 509-519, 2004.
- [BER13] Bergmann, S.: *Automatische Generierung adaptiver Modelle zur Simulation von Produktionssystemen*. Dissertation, Technischen Universität Ilmenau, 2013.
- [BPH+15] Barth, M.; Puntel Schmidt, P.; Hoernicke, M. et al.: *Methoden und Modelle der Virtuellen Inbetriebnahme - Eine Übersicht der Richtlinienarbeit des GMA FA 6.11*. In VDI-Kongress „Automation“, S. 107-120, 2015.
- [BKH+20] Barth, M.; Kübler, K.; Heinzerling, T. et al.: *Eine systematische Bewertung der Qualität von Simulationsmodellen für die Automatisierungstechnik – Identifikation und Clustering von Qualitätskriterien*. In VDI-Kongress „Automation“, S. 499 – 516, 2020.
- [BLO19] Block, C.: *Methodik zur teilautonomen Modellierung und ereignisdiskreten Simulation in CPPS - Entscheidungsunterstützung in der Planung mit dem Digitalen Zwilling*. Dissertation, Ruhr-Universität Bochum, 2019.

- [BMV+19] Busse, A.; Merhar, L.; Vernim, S. et al.: *Digitale Helfer im Arbeitsalltag - Praxisleitfaden für Assistenzsysteme in der Produktion*. Mittelstand-Digital, Fraunhofer-Einrichtung für Gießerei-, Composite- und Verarbeitungstechnik IGCV, 2019.
- [BMW17] Federal Ministry for Economic Affairs and Energy (BMWi) *Industrie 4.0: Industrie 4.0 Plug-and-Produce for Adaptable Factories: Example Use Case Definition, Models, and Implementation*. Working Paper, 2017.
- [BMW19] Plattform Industrie 4.0: *Technologieszenario „Künstliche Intelligenz in der Industrie 4.0“*. Bundesministerium für Wirtschaft und Energie (BMWi), Working Paper, 2019.
- [BMW20a] Plattform Industrie 4.0 in Kooperation mit ZVEI: *Details of the Asset Administration Shell Part 1 - The exchange of information between partners in the value chain of Industrie 4.0 (Version 2.0.1)*. BMWi, 2020.
- [BMW20b] Plattform Industrie 4.0 in Kooperation mit ZVEI: *Submodel Templates of the Asset Administration Shell - ZVEI Digital Nameplate for industrial equipment (Version 1.0)*. Bundesministerium für Wirtschaft und Energie (BMWi), 2020.
- [BOA+11] Blochwitz, T.; Otter, M.; Arnold, M. et al.: The Functional Mockup Interface for Tool independent Exchange of Simulation Models. In Proceedings of the 8th International Modelica Conference, S. 105-114, 2011.
- [BOR13] Borchering H.: *Der mittelständische Maschinenbau – flexibel und höchst innovativ auch in der Systementwicklung*. In Sendler U. (Hrsg.) *Industrie 4.0*. Xpert.press. Springer Vieweg, Berlin, Heidelberg, 2013.
- [BS20a] Bergmann, S.; Straßburger, S.: *Automatische Modellgenerierung – Stand, Klassifizierung und ein Anwendungsbeispiel*. In Ablaufsimulation in der Automobilindustrie, Herausgeber: Mayer, G., Pöge, C., Spieckermann, S., Wenzel, S. (Hrsg.), Springer Vieweg, 2020.
- [BS20b] Brinzer, B.; Schneider, K.: *Complexity assessment in production: linking complexity drivers and effects*. In 53rd CIRP Conference on Manufacturing System, S. 694–699, 2020.
- [BU16] Bohn, C.; Unbehauen, H.: *Identifikation dynamischer Systeme - Methoden zur experimentellen Modellbildung aus Messdaten*. Springer Vieweg, 2016.
- [CBR+18] Couto, L. D.; Basagiannis, S.; Ridouane, E. H. et al.: *Lessons Learned Using FMI Co-simulation for Model-Based Design of Cyber Physical Systems*. In ISoLA 2018, S. 488-503, 2018.
- [CGG+05] Christie, M.; Glimm, J.; Grove, J. et al.: *Error analysis and simulations of complex phenomena*. In Los Alamos Science 29, S. 6-25, 2005.
- [CHF14] Christiansen, L.; Hoernicke, M.; Fay, A.: Entwicklung eines Modellverbinders zur Vermeidung von Inkonsistenzen zwischen Modellen im Kontext des modellbasierten Engineerings. In VDI-Kongress „Automation“, 2014.

- [COE01] Coello Coello, C. A.: *A Short Tutorial on Evolutionary Multiobjective Optimization*. In International Conference on Evolutionary Multi-Criterion Optimization, S. 21-40, 2001.
- [CP07] Chinchuluun, A.; Pardalos, P.M.: *A survey of recent developments in multiobjective optimization*. In Annals of Operations Research 154, S. 29–50, 2007.
- [DEB15] Deb K.: *Multi-Objective Evolutionary Algorithms*. In Kacprzyk J., Pedrycz W. (Hrsg.) Springer Handbook of Computational Intelligence. Springer Handbooks. Springer, Berlin, Heidelberg, 2015.
- [Dis19] Dispan, J.: *Modulare Bauweise und neue Produktionskonzepte im Werkzeugmaschinenbau*. In Working Paper Forschungsförderung, No. 118, Hans-Böckler-Stiftung, Düsseldorf, 2019.
- [DJ17] Dürkop L., Jasperneite J.: *„Plug & Produce“ als Anwendungsfall von Industrie 4.0*. In: Vogel-Heuser B., Bauernhansl T., ten Hompel M. (Hrsg.) *Handbuch Industrie 4.0*, Bd.2. Springer Reference Technik. Springer Vieweg, Berlin, 2017.
- [DK19] Dhiman, G.; Kaur, A.: HKn-RVEA: a novel many-objective evolutionary algorithm for car side impact bar crashworthiness problem. In International Journal of Vehicle Design, Vol. 80, Nos. 2/3/4, 2019.
- [DLP+08] Drath, R.; Lüder, A.; Peschke, J.; Hundt, L.: *AutomationML – the glue for seamless Automation Engineering*. In IEEE International Conference on Emerging Technologies and Factory Automation, 2008.
- [DON20] Donhauser, T.: *Ressourcenorientierte Auftragsregelung in einer hybriden Produktion mittels betriebsbegleitender Simulation*. Dissertation, Friedrich-Alexander-Universität Erlangen-Nürnberg. In FAU Studien aus dem Maschinenbau, Band 346, 2020.
- [DPH17] Durling, E., Palmkvist, E., Henningsson, M.: *FMI and IP Protection of Models: A Survey of Use Cases and Support in the Standard*. In Proceedings of the 12th International Modelica Conference, S. 329-335, 2017.
- [DRA21] Drath, R.: *AutomationML: A Practical Guide*, Berlin, Boston: De Gruyter Oldenbourg, 2021.
- [DRO04] Dronka, S.: *Die Simulation gekoppelter Mehrkörper- und Hydraulik-Modelle mit Erweiterung für Echtzeitsimulation*, Zugl.: Dresden, Techn. Univ., Diss. Aachen: Shaker. Berichte aus der Fahrzeugtechnik, 2004.
- [DSBF+20] Dahlem, P.; Sanders, M. P.; Birck Fröhlich, H. et al.: *Hybrid model approaches for compensating environmental influences in machine tools using integrated sensors*. In at - Automatisierungstechnik, vol. 68, no. 6, S. 465-476, 2020.
- [Dub79] Dubs, F.: *Der Magnus-Effekt*. In: Aerodynamik der reinen Unterschallströmung. Flugtechnische Reihe, vol 1. Birkhäuser, Basel, 1979.

- [EBE13] Ebert, C.: *Systemidentifikation zur Modellierung mechanischer Strukturen: Markovparameter zur experimentellen Schadenserfassung*. In Schriftenreihe der Arbeitsgruppe für Technische Mechanik im Institut für Mechanik und Regelungstechnik - Mechatronik, Dissertation, Universität Siegen, 2013.
- [EM18] Elbattah, M.; Molloy, O.: *ML-Aided Simulation: A Conceptual Framework for Integrating Simulation Models with Machine Learning*. In SIGSIM-PADS '18: SIGSIM Principles of Advanced Discrete Simulation, S. 33-36, 2018.
- [ES15] Eiben, A. E.; Smith, J. E.: *What Is an Evolutionary Algorithm?* In: Introduction to Evolutionary Computing. Natural Computing Series. Springer, Berlin, Heidelberg, 2015.
- [FAY09] Fay, A.: *Effizientes Engineering komplexer Automatisierungssysteme*. In Schnieder (Hrsg.): Wird der Verkehr automatisch sicherer? Beschreibungsmittel, Methoden und Werkzeuge des integrierten Systementwurfs zur Fahrzeug- und Verkehrsautomatisierung. Braunschweig, S. 43–60, 2009.
- [FF18] Förster, R.; Förster, A.: *Einführung in die Fertigungstechnik - Lehrbuch für Studenten ohne Vorpraktikum*. Springer Vieweg, Berlin, Heidelberg, 2018.
- [FG19] Falcone, A.; Garro, A.: *Distributed Co-Simulation of Complex Engineered Systems by Combining the High Level Architecture and Functional Mock-up Interface*. In: Simulation Modelling Practice and Theory (97), 2019.
- [FKP+16] Fahrmeir, L.; Künstler, R.; Pigeot, I.; Tutz, G.: *Statistik. Der Weg zur Datenanalyse*. Springer Spektrum, Berlin/ Heidelberg, 2016.
- [FN09] Frager, O.; Nehr, W.: *Modularität und Wiederverwendung im Engineering des Maschinen- und Anlagenbaus*. In Vogel-Heuser, B. (Hrsg.): Automation & Embedded Systems – Effizienzsteigerung im Engineering. kassel university press GmbH, 2009.
- [FS12] Friedli, T.; Schuh G.: *Wettbewerbsfähigkeit der Produktion an Hochlohnstandorten*. 2. Auflage, Springer-Verlag Berlin Heidelberg, 2012.
- [FUL20] Fulda, B.: *Die neue Komplexität von Wertschöpfung: Warum Wertschöpfung immer globaler und komplexer wird und welche Effekte das auf Beschäftigung in Deutschland hat*. Forschungsförderung Report, No. 7, Hans-Böckler-Stiftung, 2020.
- [GKH+19] Gamer, T.; Kloepper, B.; Hoernicke et, M. al.: *Autonomy from an industrial perspective - Applying a taxonomy to autonomous plants*. In atp magazine Bd. 61 Nr. 6-7, S. 62-68, 2019.
- [GKL06] Geimer, M.; Krüger, T.; Linsel, P.: *Co-Simulation, gekoppelte Simulation oder Simulator-kopplung? Ein Versuch der Begriffsvereinheitlichung*. In O+ P Zeitschrift für Fluidtechnik 50 (11-12), S. 572–576, 2006.

- [GL16] Gürtler, M. R.; Lindemann, U.: Innovationsmanagement. In: Lindemann, U. (Hrsg.): Handbuch Produktentwicklung. München: Carl Hanser Verlag, 2016.
- [GMA20] VDI/VDE-Gesellschaft für Mess- und Automatisierungstechnik, Fachausschuss 6.11: *Simulation und digitaler Zwilling im Anlagenlebenszyklus*. VDI-Statusreport, Februar 2020.
- [GMD+19] Gomes, C.; Meyers, B.; Denil, J. et al.: *Semantic adaptation for FMI co-simulation with hierarchical simulators*. In Simulation: Transactions of the Society for Modeling and Simulation International 95(3), S. 241–269, 2019.
- [GN09] Gosling, J.; Naim, M. M.: *Engineer-to-order supply chain management: A literature review and research agenda*. In International Journal of Production Economics, Volume 122, Issue 2, S. 741-754, 2009.
- [GR16] Gómez, D.; Rojas, A.: *An Empirical Overview of the No Free Lunch Theorem and Its Effect on Real-World Machine Learning Classification*. In Neural Computation; 28 (1): 216–228, 2016.
- [GRI14] Grimske, S.: *Multifunktionale Schnittstellen für kleine modulare Werkzeugmaschinen*. Dissertation, Helmut-Schmidt-Universität / Universität der Bundeswehr Hamburg, Fakultät für Maschinenbau, 2014.
- [GRI16] Grimm, B.: *AutomationML in den Engineeringprozess einführen - Herausforderungen und Herangehensweise*. In atp edition 58 Nr. 05, S. 42-51, 2016.
- [GTB+17] Gomes, C.; Thule, C.; Broman, D. et al.: *Co-simulation: State of the art*. In Engineering, Computer Science ArXiv, 2017.
- [GÜN17] Günther, C.: *Beitrag zur Co-Simulation in der Gesamtsystementwicklung des Kraftfahrzeugs*. Dissertation an der TUM - Lehrstuhl für angewandte Mechanik, 2017.
- [GV19] Gomes, C.; Vangheluwe, H.: *Co-Simulation of Continuous Systems: A Hands-on Approach*. In Winter Simulation Conference (WSC), National Harbor, MD, USA, S. 1469-1481, 2019.
- [GW11] Goldhammer, L.; Woyand, H.: *Towards a knowledge based design assistance system*. In 15th IEEE International Conference on Intelligent Engineering Systems, S. 269-274, 2011.
- [GWQ+16] Gorecky, D.; Weyer, S.; Quint, F. et al.: *Definition einer Systemarchitektur für Industrie 4.0-Produktionsanlagen*. In VDI-Kongress „Automation“, 2016.
- [HAD16] Hadka, D.: *Beginner's Guide to the MOEA Framework*. CreateSpace Independent Publishing Platform, 2016.
- [HAW04] Hawkins, D. M.: The Problem of Overfitting. In Journal of Chemical Information and Computer Sciences, 44, S. 1-12, 2004.
- [HGU+16] Hensel, S.; Graube, M.; Urbas, L. et al.: *Co-simulation with OPC UA*. In: IEEE 14th International Conference on Industrial Informatics (INDIN), Poitiers, S. 20-25, 2016.

- [HHD+22] Hildebrandt, G.; Habiger, P.; Drath, R. et al.: *Virtualisierungsarchitektur für heterogene CAx-Tools im Engineering modularer Produktionssysteme*. In AALE 2022 - 18. Fachkonferenz für Angewandte Automatisierungstechnik in Lehre und Entwicklung an Hochschulen, 2022.
- [HMA+16] Hoernicke, M.; Messinger, C.; Arroyo, E. et al.: *Topologiemodelle in AutomationML*. In atp edition 58 Nr. 05, S. 28-41, 2016.
- [HMJ+04] Hevner, A. R.; March, S. T.; Jinsoo, P. et al.: *Design Science in Information Systems Research*. In MIS Quarterly 28.1, 2004.
- [Hol16] Holm, T.: *Aufwandsbewertung im Engineering modularer Prozessanlagen*. Dissertation, Helmut-Schmidt-Universität / Universität der Bundeswehr Hamburg, Fakultät für Maschinenbau, 2016.
- [HOM+18] Holowenko, O.; Oehm, L.; Majschak, J. P. et al.: *Informationsaustausch im interdisziplinären Entwicklungsprozess*. In VVD 2018 Verarbeitungsmaschinen und Verpackungstechnik - Verarbeitung & Verpackung 4.0: 9. wissenschaftliche Fachtagung, 2018.
- [HOR+19] Hofmann, P.; Oesterle, S.; Rust, P. et al.: *Machine Learning Approaches Along The Radiology Value Chain – Rethinking Value Propositions*. In Twenty-Seventh European Conference on Information Systems, 2019.
- [JCS+18] Jaensch, F.; Csiszar, A.; Scheifele, C. et al.: *Digital Twins of Manufacturing Systems as a Base for Machine Learning*. In 25th International Conference on Mechatronics and Machine Vision in Practice (M2VIP), S. 1-6, 2018.
- [JGF16] Jones, D. E.; Ghandehari, H.; Facelli, J. C.: *A review of the applications of data mining and machine learning for the prediction of biomedical properties of nanoparticles*. In Computer Methods and Programs in Biomedicine Vol. 132, S. 93-103, 2016.
- [JKS+22] Juhlin, P.; Karaagac, A.; Schlake, J. C. et al.: *Cloud-enabled Drive-Motor-Load Simulation Platform using Asset Administration Shell and Functional Mockup Units*. In 27th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), 2022.
- [JP14] Johannesson, P.; Perjons, E.: *An Introduction to Design Science*. Springer International Publishing, Switzerland, 2014.
- [JSH+22] Jordon, J.; Szpruch, L.; Houssiau, F. et al.: *Synthetic Data -- what, why and how?* The Alan Turing Institute & The Royal Societe, arXiv, 2022.
- [JSW18] Jung, T.; Shah, P.; Weyrich, M.: *Dynamic Co-Simulation of Internet-of-Things-Components using a Multi-Agent-System*. In 51st CIRP Conference on Manufacturing Systems, Volume 72, 2018.
- [KAI13] Kaiser, L.: *Rahmenwerk zur Modellierung einer plausiblen Systemstruktur mechatronischer Systeme*. Dissertation, Universität Paderborn, 2013.

- [KAK03] Kakade, S. M.: *On the Sample Complexity of Reinforcement Learning*. Dissertation, University College London, 2003.
- [KBW97] Kerr, C.; Barker, D.; Walsh, F.: *Physical and electrochemical characterisation of electroless nickel coatings on carbon steel*. In Transactions on Engineering Sciences vol 17, S. 47-59, 1997.
- [KCH+90] Kang, K. C.; Cohen, S.; Hess, J. et al.: *Feature-Oriented Domain Analysis. Feasibility Study*. In Technical Report CMU/SEI-90-TR-21, Carnegie Mellon University, Pittsburgh, PA, USA, 1990.
- [KHM+18] Kolberg, D.; Hermann, J.; Mohr, F. et al.: *SmartFactory KL Systemarchitektur für Industrie 4.0-Produktionsanlagen*. Whitepaper SF-1.2, 2018.
- [KIN05] Kinkel, S.: *Anforderungen an die Fertigungstechnik von morgen – wie verändern sich Variantenzahlen, Losgrößen, Materialeinsatz, Genauigkeitsanforderungen und Produktlebenszyklen tatsächlich?* Mitteilungen aus der Produktionsinnovationserhebung No. 37, Fraunhofer ISI, 2005.
- [KLE14] Klein, P. W.: *Methode zum Engineering von Produktionsanlagen durch Wiederverwendung von Modulen*. Dissertation, Universität Siegen, 2014.
- [KMP+08] Kofránek, J.; Mateják, M.; Privitzer, P. et al.: *Causal or acausal modelling: labour for humans or labour for machines*. Proceedings of the Technical Computing Conference, S. 1-16, 2008.
- [KMS+19] Kölle, S.; Mock, C.; Schmid, K. et al.: *Von der Industrie 4.0 zu Galvanik 4.1 - Elektrolytführung neu gedacht*. In WOMag Nr. 4, 2019.
- [KMY20] Kuchin, Y. I.; Mukhamediev, R. I.; Yakunin, K. O.: One method of generating synthetic data to assess the upper limit of machine learning algorithms performance. In Cogent Engineering, 7:1, 2020.
- [KKN+22] Keleko, A.T.; Kamsu-Foguem, B.; Ngouna, R.H. et al.: Artificial intelligence and real-time predictive maintenance in industry 4.0: a bibliometric analysis. In AI Ethics, 2022.
- [KR06] Kollat, J.B.; Reed, P.M.: Comparing state-of-the-art evolutionary multi-objective algorithms for long-term groundwater monitoring design. In Advances in Water Resources 29(6), S. 792-807, 2006.
- [KSE+20] Kiele-Dunsche, M.; Schlake, J.; Engler, W. et al.: *Verwaltungsschale: Interoperabilität für Simulationsmodelle*. In atp magazin 62 (11-12), S. 28-32, 2020.
- [KUF12] Kufner, A.: *Automatisierte Erstellung von Maschinenmodellen für die Hardware-in-the-Loop-Simulation von Montagemaschinen*. In ISW/IPA-Forschung und -Praxis Nr. 188, Dissertation, Universität Stuttgart, 2012.

- [KW07] Kleijnen, J.P.C.; Wan, J.: *Optimization of simulated systems: OptQuest and alternatives*. In Simulation Modelling Practice and Theory, Volume 15, Issue 3, S. 354-362, 2007.
- [LAD18] Ladiges, J.: *Automatisierte Bestimmung von Eigenschaften industrieller Produktionssysteme unter Einfluss evolutionärer Änderungen*. Dissertation, Helmut-Schmidt-Universität / Universität der Bundeswehr Hamburg, Institut für Automatisierungstechnik, 2018.
- [LAM00] Lampinen, J.: *Multiobjective Nonlinear Pareto-Optimization - A Pre-Investigation Report*. Lappeenranta University of Technology, 2000.
- [LBV+16] Lindskog, E.; Berglund, J.; Vallhagen, J. et al.: *Layout Planning and Geometry Analysis Using 3D Laser Scanning in Production System Redesign*. In Procedia CIRP Volume 44, S. 126-131, 2016.
- [LEH20] Lehmann, T.: *Methodenbaukasten zur Quantifizierung der statistischen Güte und deren Sensitivität von Last- und Verschleißanalysen mit einem Beispiel im Kontext alternativer Antriebskonzepte*. Dissertation, Fakultät Statistik, Technische Universität Dortmund, 2020.
- [LGR+11] Lukasiwycz, M.; Glaß, M., Reimann, F. et al.: *Opt4J – A Modular Framework for Meta-heuristic Optimization*. In Proceedings of the Genetic and Evolutionary Computing Conference (GECCO), S. 1723-1730, 2011
- [LIE13] Lier, S.: *Entwicklung einer Bewertungsmethode für die Modularisierung von Produktionssystemen in der Chemieindustrie*. Dissertation, Ruhr-Universität Bochum, Shaker Verlag, Aachen, 2013.
- [LKT+20] Leiden, A.; Kölle, S.; Thiede, S. et al.: *Model-based analysis, control and dosing of electroplating electrolytes*. In The International Journal of Advanced Manufacturing Technology 111, S. 1751–1766, 2020.
- [LLP+17] López-Gómez, C.; Leal-Ayala, D.; Palladino, M. et al.: *Emerging trends in global advanced manufacturing: challenges, opportunities and policy responses*. Report of University of Cambridge's Institute for Manufacturing, in collaboration with the United Nations Industrial Development Organization (UNIDO), 2017.
- [LS17] Laroque, C.; Skoogh, A.: *Functional Interaction of Simulation and Data Analytics – Potentials and Existing Use-Cases*. In Wenzel, S.; Peter, T. (Hrsg.) Simulation in Produktion und Logistik 2017. kassel university press, S. 403-412, 2017.
- [LWH+18] D. Lang et al.: *Assistance system to support troubleshooting of complex industrial systems*. In 14th IEEE International Workshop on Factory Communication Systems (WFCS), S. 1-4, 2018.

- [LY02] Lu, H.; Yen, G.: *Dynamic population size in multiobjective evolutionary algorithms*. In Proceedings of the 2002 Congress on Evolutionary Computation, S. 1648-1653, 2002.
- [MAT10] Matevska, J.: *Rekonfiguration komponentenbasierter Softwaresysteme zur Laufzeit*. Vieweg+Teubner Verlag, 2010.
- [MAT17] Mattmann I.: *Modellintegrierte Produkt- und Prozessentwicklung*. Dissertation, Technische Universität Darmstadt, 2017.
- [MBG+21] Manoury, M.; Brünnhäuser, J.; Geiger, A. et al.: *Expertise des Forschungsbeirats der Plattform Industrie 4.0 - Modellierungs- und Simulationsbedarfe der intelligenten Fabrik*. Forschungsbeirat der Plattform Industrie 4.0; acatech – Deutsche Akademie der Technikwissenschaften (Hrsg.), 2021.
- [MEI19] Meinel, D.: *Architektur applikationsspezifischer Multi-Physics-Simulationskonfiguratoren am Beispiel modularer Triebzüge*. Dissertation aus dem Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik (FAPS), Erlangen FAU University Press, 2019.
- [MHW+18] Marks, P.; Hoang, X. L.; Weyrich, M.; Fay, A.: *A systematic approach for supporting the adaptation process of discrete manufacturing machines*. In Research in Engineering Design 29, S. 621–641, Springer-Verlag London Ltd., 2018.
- [MIL01] Miller, D. T.: *Modular Engineering - An approach to structuring business with coherent, modular architectures of artifacts, activities, and knowledge*. Dissertation, Technical University of Denmark, 2001.
- [MJW20] Müller, T.; Jazdi, N.; Weyrich, M.: *Intelligentes Rekonfigurationsmanagement selbstorganisierter Produktionssysteme in der diskreten Fertigung*. In VDI-Kongress „Automation“, S. 517-530, 2020.
- [MWj22] Müller, T.; Jazdi, N.; Weyrich, M.: *Self-improving Models for the Intelligent Digital Twin: Towards Closing the Reality-to-Simulation Gap*. 14th IFAC Workshop on Intelligent Manufacturing Systems, 2022.
- [MKB+11] Mendoza, F.; Kollner, C.; Becker, J. et al.: *An automated approach to SystemC/Simulink co-simulation*. In 22nd IEEE International Symposium on Rapid System Prototyping, S. 135–141, 2011.
- [MKW+17] Marseu, E.; Kolberg, D.; Weyer, S. et al.: *Exemplarische Übertragung der RAMI 4.0-Verwaltungsschale auf die SmartFactoryKL - Systemarchitektur für Industrie 4.0-Produktionsanlagen*. In Whitepaper SF-2.1: 04/2017.
- [MM21] MODELISAR consortium and Modelica Association Project “FMI”. *Functional Mock-up Interface for Model Exchange and Co-Simulation*. Version 2.0.3 der Spezifikation. <https://fmi-standard.org/downloads>, 2021.
- [MP13] Mayer, G., Pöge, C.: *Quo vadis material flow simulation – visions for the future from the view of the automotive industry*. In Simulation in Produktion und Logistik, S. 11–20, 2013.

- [MSC+19] Moll, P.; Schäfer, M.; Coutandin, S. et al.: *Reconfigurable modular production plant for thermoplastic hybrid composites*. In *Production Engineering* 13, S. 469–477, 2019.
- [NDV15] Nebro, A.J.; Durillo, J.J.; Vergne, M.: *Redesigning the jMetal Multi-Objective Optimization Framework*. In *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation (GECCO Companion)*, S. 1093–1100, 2015.
- [OBT+19] Ochel, L.; Braun, R.; Thiele, B. et al.: *OMSimulator – Integrated FMI and TLM-based Co-simulation with Composite Model Editing and SSP*. In *Proceedings of the 13th International Modelica Conference*, 2019.
- [OVN18] Otto, J.; Vogel-Heuser, B.; Niggemann, O.: *Automatic Parameter Estimation for Reusable Software Components of Modular and Reconfigurable Cyber-Physical Production Systems in the Domain of Discrete Manufacturing*. In *IEEE Transactions on Industrial Informatics*, Vol 14 (1), S. 275–282, 2018.
- [OWB+15] Oppelt, M., Wolf, G., Barth, M., & Urbas, L.: *Simulation im Lebenszyklus einer Prozessanlage*. In *ATP - Automatisierungstechnische Praxis*, 57 (10), S. 38–50, 2015.
- [OWD+14] Oppelt, M.; Wolf, G.; Drumm, O. et al.: *Automatische Generierung von Simulationsmodellen für die virtuelle Inbetriebnahme auf Basis von Planungsdaten. Vorstellung eines generischen Konzepts und einer prototypischen Implementierung*. In *VDI-Kongress „Automation“*, 2014.
- [PB07] Picot, A.; Baumann, O.: *Modularität in der verteilten Entwicklung komplexer Systeme: Chancen, Grenzen, Implikationen*. In *Journal für Betriebswirtschaft* 57, S. 221–246, 2007.
- [PBM20] Palmieri, M., Bernardeschi, C., Masci, P.: *A framework for FMI-based co-simulation of human-machine interfaces*. In *Software and Systems Modeling* 19, S. 601–623, 2020.
- [PF15] Puntel-Schmidt P.; Fay A.: *Levels of Detail and Appropriate Model Types for Virtual Commissioning in Manufacturing Engineering*. In *IFAC-PapersOnLine* 48(1), S. 922–927, 2015.
- [Pi16] Platform Industrie 4.0: *Aspects of the Research Roadmap*. In *Application Scenarios I4.0. Platform Industrie 4.0*. Berlin, Germany, 2016.
- [Pös21] Pöschl, S.: *Prozessplanungsmodell für eine Effizienzsteigerung von Inbetriebnahmeprozessen im Maschinenbau*. Dissertation, Stuttgarter Beiträge zur Produktionsforschung 124, Fraunhofer Verlag, 2021.
- [PUN18] Puntel Schmidt, P.: *Methoden zur simulationsbasierten Absicherung von Steuerungscode fertigungstechnischer Anlagen*. Dissertation, Helmut-Schmidt-Universität / Universität der Bundeswehr Hamburg, Institut für Automatisierungstechnik, 2018.
- [PV20] Puiutta, E.; Veith, E.: *Explainable Reinforcement Learning: A Survey*. In: Holzinger A., Kieseberg P., Tjoa A., Weippl E. (Hrsg.) *Machine Learning and Knowledge Extraction*. CD-MAKE 2020. Lecture Notes in Computer Science, vol 12279. Springer, Cham, 2020.

- [PVL+17] Palensky, P.; Van Der Meer, A.; Lopez, C. D et al.: *Cosimulation of Intelligent Power Systems – Fundamentals, Software Architecture, Numerics, and Coupling*. In IEEE Industrial Electronics Magazine 11, S. 34-50, 2017.
- [RBB+14] Radziwon, A.; Bilberg, A.; Bogers, M. et al.: *The Smart Factory: Exploring Adaptive and Flexible Manufacturing Solutions*. In Procedia Eng. 69, S. 1184–1190, 2014.
- [REI17] Reinhart, G.: *Handbuch Industrie 4.0 – Geschäftsmodelle, Prozesse, Technik*. Carl Hanser Verlag GmbH & Co. KG, 2017.
- [REN18] Renninger, U.: *Modellierung und Analyse des Temperaturverhaltens bei Streamfinish-Bearbeitung*. Bachelorarbeit, OTEC Präzisionsfinish GmbH und Dualen Hochschule Baden-Württemberg Karlsruhe, 2018.
- [RHJ+21] Rosen, R.; Heinzerling, T.; Jäkel, J.; et al.: *Die Rolle der Simulation im Kontext des Digitalen Zwillings- Virtuelle Inbetriebnahme-Modelle als Mittler zwischen Phasen und Anwendungen?* In atp magazin 63(04), S.82-89, 2021.
- [RL10] Riddick, F.; Lee, Y.: *Core Manufacturing Simulation Data (CMSD): A Standard Representation for Manufacturing Simulation-related Information*. NIST Interagency/Internal Report (NISTIR), National Institute of Standards and Technology, Gaithersburg, 2010.
- [RM10] Rose O., März L.: *Simulation*. In: März L., Krug W., Rose O., Weigert G. (eds) „Simulation und Optimierung in Produktion und Logistik.“ VDI-Buch. Springer, Berlin, Heidelberg, 2010.
- [ROY19] Roy C. J.: *Errors and Uncertainties: Their Sources and Treatment*. In: Beisbart C.; Saam N. (Hrsg.) *Computer Simulation Validation. Simulation Foundations, Methods and Applications*. Springer, 2019.
- [RS20] Roos, M.; Siegmann, M.: *Technologie-Roadmap für das autonome Autofahren: Eine wettbewerbsorientierte Technik- und Marktstudie für Deutschland*. Working Paper Forschungsförderung No. 188, Hans-Böckler-Stiftung, Düsseldorf, 2020.
- [RW22] Rajabi, A., Witt, C.: *Self-Adjusting Evolutionary Algorithms for Multimodal Optimization*. In Algorithmica 84, S. 1694–1723, 2022.
- [RWP19] Reinhardt, H.; Weber, M.; Putz, M.: *A Survey on Automatic Model Generation for Material Flow Simulation in Discrete Manufacturing*. In 52nd CIRP Conference on Manufacturing Systems, S. 121-126, 2019.
- [SAR17] Schluse, M.; Atorf, L.; Rossmann, J.: *Experimentable digital twins for model-based systems engineering and simulation-based development*. In 2017 Annual IEEE International Systems Conference (SysCon), S. 1-8, 2017.
- [SB17] Siebertz K.; Bebbler D.; Hochkirchen T.: *Sensitivitätsanalyse*. In: Statistische Versuchsplanung. VDI-Buch. Springer Vieweg, Berlin, Heidelberg, 2017.

- [SBL21] Stark, R.; Brandenburg, E.; Lindow, K.: *Characterization and application of assistance systems in digital engineering*. In CIRP Annals Volume 70, Issue 1, S. 131-134, 2021.
- [SC20] Sokoll, K.; Clausing, M.: *Methoden und Werkzeuge der Simulationsassistenz*. In Mayer, G. et al. (Hrsg.): *Ablaufsimulation in der Automobilindustrie*, Springer-Verlag GmbH Deutschland, 2020.
- [SCA+17] Scaramellini, S.; Cerri, P.; Astorino, G. et al.: *Parallel Simulation Techniques for Real-Time Field Management: An Innovative Integrated Production Optimization Tool Application*. In Offshore Mediterranean Conference and Exhibition, Ravenna, Italy, 2017.
- [SCH15] Schmoll, R.: *Co-Simulation und Solverkopplung: Analyse komplexer multiphysikalischer Systeme*. Zugl.: Kassel, Univ., Diss. Kassel University Press. Berichte des Instituts für Mechanik, 3, 2015.
- [SCH18] Scholz, A.: *Unterstützung des Engineerings von fertigungstechnischen Produktionssystemen mit Hilfe von Maschinenfunktionen: Methode, Modell und Beschreibungsmittel für ein funktionsorientiertes Planen*. Dissertation, Helmut-Schmidt-Universität / Universität der Bundeswehr Hamburg, Institut für Automatisierungstechnik, 2018.
- [SCH20] Scheifele S. M.: *Generierung des Digitalen Zwillings für den Sondermaschinenbau mit Losgröße 1*. Stuttgarter Beiträge zur Produktionsforschung, Band 107. Hrsg.: Fraunhofer IPA, Stuttgart. Fraunhofer Verlag, 2020.
- [SCH99] Schnieder, E.: *Methoden der Automatisierung: Beschreibungsmittel, Modellkonzepte und Werkzeuge für Automatisierungssysteme*. Studium Technik. Wiesbaden: Vieweg+Teubner Verlag, 1999.
- [SCS+19] Saidi, S.; Charif, A.; Sassolas, T. et al.: *Fast Virtual Prototyping of Cyber-Physical Systems using SystemC and FMI: ADAS Use Case*. In Proceedings of the 30th International Workshop on Rapid System Prototyping, S. 43-49, 2019.
- [SDJ+22] Stegmaier, V. et al.: *A Structure of Modelling Depths in Behavior Models for Digital Twins*. In IEEE International Conference on Emerging Technologies And Factory Automation, 2022.
- [SDL12] Svensson, B.; Danielsson, F.; Lennartson, B.: *Time-synchronised hardware-in-the-loop simulation - Applied to sheet-metal press line optimization*. In Control Engineering Practice 20(8), S. 792-804, 2012.
- [SFA+18] Sipper, M.; Fu, W.; Ahuja, K. et al.: *Investigating the parameter space of evolutionary algorithms*. In BioData Mining 11, 2, 2018.

- [SGE+19] Schweiger, G.; Gomes, C.; Engel, I. et al.: *An empirical survey on co-simulation: Promising standards, challenges and research needs*. In: Simulation Modelling Practice and Theory 95, S. 148-163, 2019.
- [SHI+20] Saputra, A. A.; Hong, C. W.; Ijspeert, A. J. et al.: *A Muscle-Reflex Model of Forelimb and Hindlimb of Felidae Family of Animal with Dynamic Pattern Formation Stimuli*. In International Joint Conference on Neural Networks (IJCNN), S. 1-8, 2020.
- [SL19] Scott, E.; Luke, S.: *ECJ at 20: Toward a General Metaheuristics Toolkit*. In Proceedings of the Genetic and Evolutionary Computing Conference (GECCO), S. 1391–1398, 2019.
- [SLK19] Stecken, J.; Lenkenhoff, K.; Kuhlenkötter, B.: *Classification method for an automated linking of models in the co-simulation of production systems*. In Procedia CIRP 81, S. 104–109, 2019.
- [SN10] Spiess, A.-N.; Neumeyer, N.: *An evaluation of R2 as an inadequate measure for nonlinear models in pharmacological and biochemical research: a Monte Carlo approach*. In BMC Pharmacology. 10: 6, 2010.
- [SSH+21] Schiering, N.; Schnelle-Werner, O.; Heizmann, M. et al.: *Modellbildung in der Messtechnik und ihre künftige Entwicklung*. In VDI-Berichte Nr. 2390, S. 179-188, 2021.
- [SSP+15] Susto, G. A.; Schirru, A.; Pampuri, S. et al.: *Machine Learning for Predictive Maintenance: A Multiple Classifier Approach*. In IEEE Transactions on Industrial Informatics 11(3), S. 812-820, 2015.
- [Sum04] Suman, B.: *Study of simulated annealing based algorithms for multiobjective optimization of a constrained problem*. In Computers & chemical engineering, 28(9), S. 1849-1871, 2004.
- [SVT+20] Scheifele, C.; Verl, A.; Tekouo, W. et al.: *Eine Online-Plattform für Digitale Zwillinge - Austausch, Pflege und Bereitstellung von Simulationsmodellen zur virtuellen Inbetriebnahme*. In atp magazin 62 (11-12), S. 58-65, 2020.
- [TFD18] Thongnuch, S.; Fay, A.; Drath, R.: *Semi-automatic generation of a virtual representation of a production cell: Combining 3D CAD and VDI-2860 behavior models by means of AutomationML*. In at - Automatisierungstechnik, vol. 66, no. 5, S. 372-384, 2018.
- [TFV+19] Thyago P.; Fabrizzio A.; Vita R. et al.: *A systematic literature review of machine learning methods applied to predictive maintenance*. In Computers & Industrial Engineering 137, 2019.
- [UJ13] Umbarkar, A.J.; Joshi, M.S: *Review of parallel genetic algorithm based on computing paradigm and diversity in search space*. In ICTACT Journal on Soft Computing 3(4), S. 615-622, 2013.

- [UWQ17] Um, J.; Weyer, S.; Quint, F.: *Plug-and-Simulate within Modular Assembly Line enabled by Digital Twins and the use of AutomationML*. In IFAC-PapersOnLine, Volume 50, Issue 1, S. 15904-15909, 2017.
- [VAN99] Van Veldhuizen, D. A.: *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. Dissertation, Graduate School of Engineering of the Air Force Institute of Technology, 1999.
- [VBG+17] Vathoopan, M.; Brandenbourger, B.; George, A. et al.: *Towards an integrated plant engineering process using a data conversion tool for AutomationML*. In IEEE International Conference on Industrial Technology ICIT, S. 1205-1210, 2017.
- [VBT17] Vogel-Heuser, B.; Bauernhansl, Ten Hompel, M.: *Handbuch Industrie 4.0 Bd.2 – Automatisierung*. Springer Vieweg, Berlin, Heidelberg, 2017.
- [VF05] Verleysen, M.; François, D.: *The Curse of Dimensionality in Data Mining and Time Series Prediction*. In Cabestany, J.; Prieto, A.; Sandoval, F. (Hrsg.): *Computational Intelligence and Bioinspired Systems*. IWANN. Lecture Notes in Computer Science, vol 3512. Springer, Berlin, Heidelberg, 2005.
- [VM14] VDMA; McKinsey&Company: *Zukunftsperspektive deutscher Maschinenbau - Erfolgreich in einem dynamischen Umfeld agieren*. Report, 2014.
- [VW12] Van Iwaarden, J.; van der Wiele, T.: *The effects of increasing product variety and shortening product life cycles on the use of quality management systems*. In International Journal of Quality & Reliability Management 29 (5), 2012.
- [VW17] Viehof, M.; Winner, H.: *Stand der Technik und der Wissenschaft: Modellvalidierung im Anwendungsbereich der Fahrdynamiksimulation*. Forschungsbericht, Technische Universität Darmstadt, 2017.
- [WAR93] Warnecke H. J.: *Beschichten*. In: Einführung in die Fertigungstechnik. Vieweg+Teubner Verlag, Wiesbaden, 1993.
- [WEI15] Weicker, K.: *Evolutionäre Algorithmen*. Springer Vieweg, Wiesbaden, 2015.
- [WHT+12] ElMaraghy, W.; ElMaraghy, H.; Tomiyama, T. et al.: *Complexity in engineering design and manufacturing*. In CIRP Annals Volume 61, Issue 2, S. 793-814, 2021.
- [WIE02] Wiendahl, H. P.: *Wandlungsfähigkeit*. In: Werkstatttechnik WT-Online, 92 Nr. 4, S. 122-127, 2002.
- [WIE14] Wieringa, R. J.: *Design Science Methodology for Information Systems and Software Engineering*. Springer Berlin Heidelberg, 2014.
- [WRN14] Wiendahl, H. P.; Reichardt, J.; Nyhuis, P.: *Handbuch Fabrikplanung - Konzept, Gestaltung und Umsetzung wandlungsfähiger Produktionsstätten*. Hanser Verlag, 2014.

- [WÜN07] Wunsch, G.: *Methoden für die virtuelle Inbetriebnahme automatisierter Produktionssysteme*. Dissertation, Technische Universität München, 2007.
- [YTZ+19] Yuan, Y., Tang, X.; Zhou, W. et al.: *Data driven discovery of cyber physical systems*. In Nature Communications 10, 4894 (2019).
- [ZAM19] Zhou, Y.; Allmaras, M.; Massalimova, A. et al.: *Assist System Framework for Production Prioritization: Flexible Architecture to integrate Simulation in Run-Time Environment*. In VDI-Kongress „Automation“, 2019.
- [ZAS+18] Zipper, H.; Auris, F.; Strahilov, A. et al.: *Keeping the Digital Twin up-to-date – Process Monitoring to Identify Changes in a Plant*. In IEEE International Conference on Industrial Technology (ICIT), S. 1592-1597, 2018.
- [ZMP13] Zhang, H.; Mao, H.; Peng, G.: *Modular collaborative simulation in multibody system dynamics of complex mechatronic products*. In IEEE International Conference on Systems, Man, and Cybernetics, 2013.
- [ZS12] Zeigler, B.P.; Sarjoughian, H.S.: *Guide to Modeling and simulation of systems of systems*. Springer Science & Business Media, 2012.
- [ZT99] Zitzler, E.; Thiele, L.: *Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach*. In IEEE Transactions on Evolutionary Computation, vol. 3, no. 4, S. 257-271, 1999.
- [ZZT08] Zeng Q. C.; Zhang E.; Tellinghuisen, J.: *Univariate calibration by reversed regression of heteroscedastic data: a case study*. In The Analyst 133, S. 1649-1655, 2008.

Referenzierte Normen, Richtlinien und Empfehlungen

- [DIN 8580] DIN 8580. *Fertigungsverfahren - Begriffe, Einteilung*, 2003.
- [DIN 8589-17] DIN 8589-17. *Fertigungsverfahren Spanen - Teil 17: Gleitspanen. Einordnung, Unterteilung, Begriffe*, 2003.
- [IEC 62714-1] IEC 62714-1:2018. *Engineering data exchange format for use in industrial automation systems engineering - Automation Markup Language - Part 1: Architecture and general requirements*, 2018.
- [IEEE 1516-2010] IEEE 1516-2010. *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules*. Technical report, IEEE, 2010.
- [J3016_202104] SAE International (Society of Automotive Engineers). *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, 2021.
- [NAMUR NE 148] NAMUR Empfehlung NE 148. *Anforderungen an die Automatisierungstechnik durch die Modularisierung verfahrenstechnischer Anlagen*, 2013.
- [VDI/VDE/NAMUR 2658-1] VDI 2658 Blatt 1. *Automatisierungstechnisches Engineering modularer Anlagen in der Prozessindustrie - Allgemeines Konzept und Schnittstellen*, 2019.
- [VDI 2776-1] VDI 2776 Blatt 1. *Verfahrenstechnische Anlagen - Modulare Anlagen- Grundlagen und Planung modularer Anlagen*, 2020.
- [VDI 3633-1] VDI 3633 Blatt 1. *Simulation von Logistik-, Materialfluss- und Produktionssystemen – Grundlagen*, 2014.
- [VDI 3695-1] VDI 3695 Blatt 1. *Engineering von Anlagen - Evaluieren und Optimieren des Engineerings - Grundlagen und Vorgehensweise*, 2010.
- [VDI 4499-2] VDI 4499 Blatt 2. *Digitale Fabrik – Digitaler Fabrikbetrieb*, 2011.
- [VDI/VDE 2206] VDI/VDE 2206. *Entwicklung mechatronischer und cyber-physischer Systeme*, 2021.
- [VDI/VDE 3693-1] VDI/VDE 3693 Blatt 1. *Virtuelle Inbetriebnahme - Modellarten und Glossar*, 2016.
- [VDI/VDE 3693-2] VDI/VDE 3693 Blatt 2. *Virtuelle Inbetriebnahme Einführung der virtuellen Inbetriebnahme in Unternehmen*, 2018.

Verzeichnis referenzierter Internetquellen

- [AUT22[®]] AutomationML, <http://www.automationml.org>, [letzter Zugriff 17.10.2022].
- [FUN22[®]] Functional Mock-Up Interface, <https://fmi-standard.org>, [letzter Zugriff 17.10.2022].
- [JEN22[®]] Jentner Plating Technology GmbH, *Kleingalvanik DIGITAL II Version 2.0 – Bedienungsanleitung*, <https://shop.jentner.de/media/a8/40/ca/1617282672/DIGITAL-II-Ver-2-0-de.pdf>, [letzter Aufruf: 17.10.2022].
- [MOE22[®]] MOEA Framework: <http://moeaframework.org/index.html>, [letzter Aufruf: 30.09.2022].
- [OPE22[®]] OpenModelicaSimulator, <https://openmodelica.org/uncategorised/191-omsimulator>, [letzter Zugriff 17.10.2022].
- [SYS22[®]] OMG System Modeling Language Specification Version 1.6: <https://www.omg.org/spec/SysML/>, [letzter Aufruf: 17.10.2022].
- [TWI22[®]] ISG Industrielle Steuerungstechnik GmbH, *TwinStore*, <https://www.virtuos-twinstore.de/about>, [letzter Zugriff 17.10.2022].
- [WES22[®]] Westhouse Group GmbH: *Weiterbildung: Programmiersprachen für Ingenieure* <https://www.westhouse-group.com/weiterbildung-programmiersprachen-fuer-ingenieure>, 21.01.2020, [letzter Zugriff 17.10.2022].
- [WIL22[®]] Wilhelmstötter, F.: *Jenetics Library User's Manual 7.0*, 2022, <https://jenetics.io/manual/manual-7.0.0.pdf>, [letzter Zugriff 17.10.2022].

Veröffentlichungen des Verfassers

- [HBF18*a] C. Härle, M. Barth, A. Fay: *Modelica-basierte Embedded-Simulation – Einplatinenrechner als Plattform für modellbasiertes Engineering*. EKA 2018 – 15. Fachtagung Entwurf komplexer Automatisierungssysteme, Magdeburg, 02.-03. Mai 2018.
- [HBF18*b] C. Härle, M. Barth, A. Fay: *Process simulation on single-board computers – the digital twin in the field*. In: 14th IEEE Conference on Automation Science and Engineering (CASE 2018), S. 1548-1555, München, 20.-23. August 2018.
- [HBF18*c] C. Härle, M. Barth, A. Fay: *Einplatinenrechner als Simulationsplattform*. In: atp-magazin 60(11-12), S. 56-67, 2018.
- [HBF19*] C. Härle, M. Barth, A. Fay: *Hybride Modellierung und verteilte Multi-Agenten zur Co-Simulation auf Einplatinen-Computern*. In: AALE 2019 – 16. Fachkonferenz für Angewandte Automatisierungstechnik in Lehre und Entwicklung an Hochschulen, S. 69-77, Heilbronn, 28. Februar – 01. März 2019.
- [HBF20*a] C. Härle, M. Barth, A. Fay: *Simulationsmodellgenerierung im modularen Maschinen- und Anlagenbau - Assistenzsystem zur automatischen Komposition und Konfiguration von Co-Simulationen*. In atp magazin 62(9), S. 72-79, 2020.
- [HBF20*b] C. Härle, M. Barth, A. Fay: *Assistance system for the automated composition and configuration of a Co-Simulation*. 34th Annual European Simulation and Modelling Conference, S. 155-162, Frankreich, Oktober 2020.
- [SHR+21*] C. Scheifele, C. Härle, R. Rosen, T. Heinzerling, M. Röhler: *Co-Simulation als Realisierung digitaler Zwillinge – Austausch und Kopplung von (Teil-)Modellen für eine virtuelle Inbetriebnahme*. In: atp-magazin 63(4), S. 90-96, 2021.
- [HBF21*a] C. Härle, M. Barth, A. Fay: *Betriebsparallele Adaption einer Co-Simulation für diskrete Fertigungsanlagen*. In: VDI-Automationskongress, S. 517-534, 2021.
- [HBF21*b] C. Härle, M. Barth, A. Fay: *Operation-parallel adaptation of a co-simulation for discrete manufacturing plants*. In: 26th IEEE International Conference on Emerging Technologies And Factory Automation (ETFA 2021), S. 1-8, September 07-10, 2021, Västerås, Sweden.
- [HBF22*] C. Härle, M. Barth, A. Fay: *Bedarfsorientierte Rekonfiguration einer Co-Simulation für diskrete Fertigungsanlagen*. In: AALE 2022 – 18. Konferenz für Angewandte Automatisierungstechnik in Lehre und Entwicklung an Hochschulen, S. 155-164, Pforzheim, 09. März – 11. März 2022.
- [HJS22*] C. Härle, J. Jäkel, G. Sand: *Tagungsband AALE 2022 – Wissenstransfer im Spannungsfeld von Autonomisierung und Fachkräftemangel (Hrsg.)*. AALE 2022 – 18. Konferenz für Angewandte Automatisierungstechnik in Lehre und Entwicklung an Hochschulen, Pforzheim, 09. März – 11. März 2022.

Studentische Arbeiten

Dieses Verzeichnis enthält eine Liste der studentischen Arbeiten, die vom Autor betreut wurden.

- [DDG18[%]] A. Dayan, F. Delz, T. Grunau: *Virtual commissioning - Definition of design guidelines and implementation of a SIL demo*. Pforzheim, Hochschule Pforzheim, Forschungsprojekt, 2018.
- [KD18[%]] J. Dehnen, J. Knittel: *Bidirektionale Interaktion zwischen virtuellem Rundgang und bestehender Prozessanlage*. Pforzheim, Hochschule Pforzheim, Projektarbeit, 2018.
- [KD19[%]] J. Dehnen, J. Knittel: *Implementierung einer Sicherheitsunterweisung mittels Augmented Reality für eine verfahrenstechnische Prozessanlage*. Pforzheim, Hochschule Pforzheim, Projektarbeit, 2019.
- [GEG19[%]] T. Gegenheimer: *Hardware-in-the-Loop (HiL) Simulation*. Karlsruhe, Dualen Hochschule Baden-Württemberg Karlsruhe, Studienarbeit, 2019.
- [SCH19[%]] L. Schumacher: *The Digital Twin in industrial engineering*. Pforzheim, Hochschule Pforzheim, Forschungsprojekt, 2019.
- [BBS19[%]] M. Blumenstein, N. Bohnert, A. Spindler: *Parameter Optimization using Reinforcement Learning based on Functional Mockup Units*. Pforzheim, Hochschule Pforzheim, Forschungsprojekt, 2019.
- [BAY19[%]] T. Bayhan: *Vergleich von genetischen Algorithmen und Least-Squares für die Optimierung eines Temperaturmodells*. Pforzheim, Hochschule Pforzheim, Bachelorarbeit, 2019.
- [HS20[%]] N. Hartmann, J. Schweidler: *Ansteuerung eines CAD Modells per SPS Programmierung*. Pforzheim, Hochschule Pforzheim, Projektarbeit, 2020.
- [CS21[%]] M.A. Caycioglu, A. Serzysko: *Aufbau eines Drehpendels zur Bestimmung des Massenträgheitsmoments durch Auswertung von Inertialsensordaten*. Pforzheim, Hochschule Pforzheim, Projektarbeit, 2021.
- [JS21[%]] L. Jany, C. Schmid: *Konzeption und Simulation eines Torque Vectoring Systems*. Pforzheim, Hochschule Pforzheim, Projektarbeit, 2021.
- [SP21[%]] M. Seegen, A. Pavlidis: *Simulation und 3D-Visualisierung eines Ball on Plate Systems*. Pforzheim, Hochschule Pforzheim, Projektarbeit, 2021.
- [KS21[%]] A. Kurz, D. Schanz: *Entwicklung eines digitalen Zwillings einer Füllstandsregelanlage in Simscape*. Pforzheim, Hochschule Pforzheim, Projektarbeit, 2021.
- [BDS21[%]] L. Britsch, T.-K. Dinh, P. Schenk: *Fahrdynamiksimulation eines elektrischen Formula Student Rennwagens*. Pforzheim, Hochschule Pforzheim, Projektarbeit, 2021.