

Modellierung und Validierung einer Steuerungs- architektur für autonome Cyber-Physische Matrixproduktionssysteme

Vom Fachbereich Maschinenbau und Verfahrenstechnik
der Rheinland-Pfälzischen Technischen Universität Kaiserslautern-Landau
zur Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

genehmigte

Dissertation

von

Herrn

M.Sc. Simon Komesker

aus Georgsmarienhütte

Tag der mündlichen Prüfung: 21.03.2025

Dekan: Herr Prof. Dr. rer. nat. Roland Ulber

Promotionskommission:

Vorsitzender: Prof. Dr.-Ing. Jan C. Aurich

1. Berichterstatter: Prof. Dr.-Ing. Martin Ruskowski

2. Berichterstatter: Prof. Dr.-Ing. Alexander Fay

Ergebnisse, Meinungen und Schlüsse dieser Dissertation sind nicht notwendigerweise die der Volkswagen Aktiengesellschaft.

Danksagung

An dieser Stelle möchte ich all denjenigen meinen tiefsten Dank aussprechen, die mich während der dualen Promotion und der Erstellung dieser Dissertation unterstützt haben.

Zuallererst möchte ich mich bei meinem Doktorvater, Prof. Dr.-Ing. Martin Ruskowski für die Förderung, Unterstützung und vertrauensvolle Zusammenarbeit während der Promotionszeit bedanken. Seine Expertise und sein Engagement haben maßgeblich zum Gelingen dieser Arbeit beigetragen. Ein besonderer Dank gilt auch Dr.-Ing. habil. Achim Wagner, Leiter des Forschungsbereichs Innovative Fabrikssysteme, der mir wertvolle Hinweise während der Promotion gab und mich engagiert in das Forschungsprojekt, seine Forschungsgruppe und das Institut integrierte. Ebenfalls möchte ich mich bei Prof. Dr.-Ing. Alexander Fay für die Begutachtung und seinem Interesse an meiner Arbeit sowie den inspirierenden Austausch bedanken. Mein weiterer Dank für die Übernahme des Vorsitzes der Prüfungskommission gilt Prof. Dr.-Ing. Jan C. Aurich.

Ich möchte darüber hinaus den Menschen danken, die Teil der verschiedenen Organisationen sind und waren, mit denen ich während meines Weges zusammenarbeiten durfte. Danke an Stefan Strickrodt, der mir die Vision und den Glauben an ihre Umsetzung mit auf den Weg gegeben hat und an Jonathan Bartels, der sie mit mir zum Leben erweckt hat und weiterentwickelt. Ich kann mich leider aus rechtlichen Gründen nicht namentlich bei den Mitarbeitenden der Volkswagen AG bedanken, aber ich hoffe, dass meine Wertschätzung dennoch deutlich wird. Außerdem möchte ich meinen Kolleginnen und Kollegen bei der Smart Factory^{KL}, dem Forschungsbereich Innovative Fabrikssysteme am DFKI sowie den Kolleginnen und Kollegen am Lehrstuhl Werkzeugmaschinen und Steuerungen der RPTU in Kaiserslautern danken. Die organisationsübergreifende Zusammenarbeit im Rahmen der Industriepromotion mit akademischen wie praxisnahen Anregungen, Diskussionen und Einblicken haben meine Forschung bereichert und mir neue Perspektiven eröffnet. Durch die mir übertragene Verantwortung in der Bearbeitung und Leitung von Forschungsprojekten konnte ich mich nicht nur fachlich, sondern auch persönlich weiterentwickeln.

Ich möchte vor allem meiner Familie und meinen Freunden danken, die mir in den letzten Jahren immer zur Seite standen. Meine Eltern Günter und Mechtilde Komesker und meine Brüder Stefan und Magnus Komesker haben mich in jeder Lebenslage unterstützt und stets an mich geglaubt. Mein Onkel Dr. Heinrich Komesker hat mir stets ein offenes Ohr geschenkt und stand als Ratgeber und Motivator zur Seite. Danke ihm und meiner Tante Brigitta für das Korrekturlesen dieser Arbeit. Ganz besonders danke ich meiner Partnerin Alicia Kühn für den unerschütterlichen Rückhalt und ihr Verständnis für die zahllosen Abende, Wochenenden und Urlaubstage, die ich während den arbeitsintensiven und herausfordernden Phasen für die Dissertation genutzt habe.

Diese Arbeit wäre ohne die Unterstützung von wunderbaren Menschen nicht möglich gewesen. Danke für die Erfahrungen, die ich auf diesem aufregenden, herausfordernden und mit Sicherheit einmaligem Weg machen durfte.

Kurzfassung

CPMPS erhalten als alternative Organisationsform für die industrielle Produktion immer stärkere Beachtung. Aufgrund ihrer gesteigerten Veränderungsfähigkeiten bieten sie Herstellern schnellere Anpassungsmöglichkeiten an externe und interne Ereignisse. Allerdings fehlt es an industriellen Umsetzungsbeispielen, was mit den gesteigerten Freiheitsgraden und der damit verbundenen Steuerungskomplexität begründet wird. Es bedarf einer entscheidungsunterstützenden Steuerungsstruktur, die sich durch eine autonome Optimierung und Rekonfiguration auszeichnet. Wesentliche Anforderungen an die autonome Steuerungsarchitektur sind Selbstorganisation, Anpassungsfähigkeit und Skalierbarkeit. Die entwickelte Architektur wird modellbasiert mithilfe der DACS Methode erstellt. Dies beinhaltet ein Entscheidungsmodell, ein holonisches Agentenmodell und ein Interaktionsmodell. Ergänzt werden diese mit einem MILP-Optimierungsmodell zur Lösung eines erweiterten Flexible Job Shop Scheduling Problems (FJSP). Die Implementierung erfolgt in einem agentenbasierten Simulationsframework mit Optimierungsservice, in dem verschiedene Technologien wie JADE, Tecnomatix-DES, Node-Red Middleware und SQL-Datenbanken enthalten sind. Die Simulationsexperimente als Teil der Validierung zeigen, dass die Steuerungsarchitektur eine skalierbare Auftragsplanung und adaptive Auftragsausführung ermöglicht. Durch serviceorientierte Integration eines Tabu-Search Verfahrens wird ein erweitertes FJSP skalierbar gelöst. Die Ergebnisse bestätigen die Eignung einer holonischen, serviceorientierten Steuerungsarchitektur für eine ganzheitliche autonome Steuerung von CPMPS. Vor allem die Verwendung der holonischen Interaktionsmodelle als Entwurfsmuster wird für die weitere Forschung empfohlen.

Abstract

CPMPS are gaining increasing attention as an alternative organizational form for industrial production. Due to their enhanced adaptability, they offer manufacturers quicker response options to external and internal events. However, there is a lack of industrial implementation examples, which is attributed to the increased degrees of freedom and the associated control complexity. A decision-supportive control structure characterized by autonomous optimization and reconfiguration is needed. Key requirements for the autonomous control architecture are self-organization, adaptability, and scalability. The developed architecture is created using the model-based DACS method. This includes a decision model, a holonic agent model, and an interaction model, complemented by an MILP optimization model to solve an extended Flexible Job Shop Scheduling Problem (FJSP). The implementation is carried out in an agent-based simulation framework with service-oriented scheduling, that incorporates various technologies such as JADE, Tecnomatix-DES, Node-Red Middleware, and SQL databases. Simulation experiments show that the control architecture enables scalable scheduling and adaptive dispatching. By service-oriented integration of a tabu search algorithm, the extended FJSP is solved in a scalable manner. The results confirm the suitability of a holonic, service-oriented control architecture for holistic autonomous control of CPMPS. Especially the deployment of the holonic interaction models as design pattern is recommended for future research.

Inhaltsverzeichnis

Danksagung	III
Kurzfassung/Abstract	IV
Inhaltsverzeichnis	V
Abkürzungsverzeichnis	VII
Formelzeichenverzeichnis	VIII
1 Einleitung	1
1.1 Motivation und Problembeschreibung	1
1.2 Zielsetzung und Forschungsfragen.....	4
1.3 Vorgehensweise und Struktur der Arbeit	6
2 Grundlagen und Stand der Forschung zur Steuerung von CPMPS	8
2.1 Grundlagen zu Systemen und autonomer Steuerung	8
2.2 Definition und Einordnung von Cyber-Physischen Matrixproduktionssystemen.....	14
2.3 Steuerungsarchitekturen veränderungsfähiger CPPS.....	17
2.3.1 Multi-Agenten-Systeme und holonische Steuerungsarchitekturen.....	25
2.3.2 Ansätze holonischer Steuerungsarchitekturen für rekonfigurierbare CPPS ...	31
2.4 Produktionsplanung und -steuerung veränderungsfähiger CPPS.....	36
2.4.1 Dynamische Reihenfolgeoptimierung und erweitertes FJSP	40
2.4.2 Ansätze zur Lösung von Reihenfolgeoptimierungsproblemen flexibler CPPS	43
2.5 Stand der Forschung zur autonomen Steuerung von CPMPS.....	46
2.6 Zusammenfassung und Analyse.....	51
3 Agentenbasierte Modellierung einer holonischen Steuerungsarchitektur	58
3.1 Entscheidungsmodell	60
3.1.1 Autonomer Steuerungsprozess	61
3.1.2 Identifikation der Steuerungsentscheidungen	63
3.2 Agentenmodell.....	67
3.2.1 Kapselung der Steuerungsentscheidungen zu Agenten	67
3.2.2 Holonische Kapselung der Agenten.....	71
3.3 Interaktionsmodell	73
3.3.1 Intraholonische Interaktion	74
3.3.2 Interholonische Interaktion	75
3.4 Optimierungsmodell	77

3.4.1	Spezifikation des erweiterten FJSP	78
3.4.2	Mathematische Formulierung des erweiterten FJSP	81
4	Implementierung der Steuerungsarchitektur als Simulation	87
4.1	Aufbau des agentenbasierten Simulationsframeworks	89
4.1.1	Multi-Agenten-System	90
4.1.2	Diskrete Event Simulation	92
4.1.3	Middleware	94
4.1.4	Datenbank	95
4.1.5	Optimierungsservice	97
4.1.6	Schnittstellen	97
4.2	Implementierung des holonischen Agentenmodells	98
4.2.1	Work Center Holon	99
4.2.2	Station Holon	100
4.2.3	Control Device Holon	102
4.3	Implementierung des Optimierungsmodells	105
4.3.1	Auswahl und Adaption eines Optimierungsverfahrens	105
4.3.2	Eigenschaften des Optimierungsverfahrens	108
5	Inbetriebnahme und Validierung	116
5.1	Vorstellung des Anwendungsfalls	116
5.2	Simulationsexperimente	120
5.3	Evaluation	129
6	Zusammenfassung	133
6.1	Fazit	134
6.2	Ausblick	136
Anhang		141
A.I.	UML-Klassendiagramm	141
A.II.	Technische Vorrangbeziehungen	142
A.III.	Holonische Kapselung des Entscheidungsmodells	144
A.IV.	Detailansicht DES	145
A.V.	Übersicht Datenbank	146
A.VI.	Weitere Zielgrößen der Simulationsexperimente	147
A.VII.	Korrelationsanalysen zum Optimierungsverfahren	149
A.VIII.	Exemplarische Optimierungsverläufe	152
Literaturverzeichnis		154

Abkürzungsverzeichnis

A _{L1}	Auftrag der Ebene L1
CDH	Control Device Holon
CPMPS	Cyber-Physisches Matrixproduktionssystem (engl.: Cyber Physical Matrix Production System)
CPMPSoS	Cyber-Physical Matrixproduction System of Systems
CPPM	Cyber-Physisches Produktionsmodul
CPPS	Cyber-Physische Produktionssysteme
CPS	Cyber-Physische Systeme
D	Dependency
E	Event
FIPA	Foundation for Intelligent Physical Agents
FJSP	Flexible Job Shop Problem
FTF	Fahrerloses Transportfahrzeug
I	Interaction
I4.0	Industrie 4.0
IKT	Informations- und Kommunikationstechnologien
JSP	Job Shop Problem
MAS	Multi-Agenten-System
PPS	Produktionsplanung und -steuerung
RAMI4.0	Referenzarchitekturmodell Industrie 4.0
RH	Ressourcenholon
SH	Stationsholon
SK _{L1}	Skill der Ebene L1
SoA	Serviceorientierte Architektur
SoS	System of Systems
TH	Transportholon
VWS	Verwaltungsschale
WCH	Work Center Holon

Formelzeichenverzeichnis

$j \in \mathcal{J}$	Arbeitsgänge j des Jobs $\mathcal{J}$
$k \in \mathcal{K}$	Arbeitsgangtypen k der Gesamtanzahl $\mathcal{K}$
$m \in \mathcal{M}$	Ressourcen m der Gesamtanzahl $\mathcal{M}$
$m \in \mathcal{M}_k$	Ressourcen, die Arbeitsgänge vom Typ k bearbeiten können
$t \in \mathcal{T}$	Zeitschritte t des Planungszeithorizonts $\mathcal{T}$
p_{kjm}	Dauer des Arbeitsgangs bei Bearbeitung durch Ressource m
st_{khm}	Rüstzeit von Ressource m , wenn von Arbeitsgangtyp k auf h umgerüstet wird
$tt_{mm'}$	Dauer des Transports zwischen den Standorten von unterschiedlichen Ressourcen m und m'
pc_{jkh}	wenn für Job j der Arbeitsgang vom Typ h erst gestartet werden darf, wenn der vom Typ k beendet ist
Θ	hinreichend große Zahl
d_m	Freigabezeitpunkt von Ressource m
a_{kj}	Primärarbeitsgang
C_{max}	Makespan
S_{kj}	Beginn der Bearbeitung von Arbeitsgang O_{kj}
X_{kjhim}	Reihenfolge der Arbeitsgänge auf einer Ressource
Y_{jhk}	Reihenfolge der Arbeitsgänge eines Jobs
G_{kjm}	Ausführung eines Arbeitsgangs auf einer Ressource

1 Einleitung

Industrielle Produktionssysteme mit global vernetzten Lieferketten werden durch langfristige und kurzfristige Veränderungstreiber ständig herausgefordert [Ba14; WW20]. Langfristige Veränderungstreiber sind z. B. in der Automobilindustrie der Trend zu alternativen Antriebssystemen, wohingegen kurzfristige Veränderungstreiber durch unvorhergesehene Einflussfaktoren, wie lokale oder globale Krisen hervorgerufen werden und mit Versorgungsengpässen einhergehen [PI22]. Dies hat Einfluss auf die Planung und Steuerung der Produktion. Stand bei der Gestaltung von Produktionssystemen bisher vor allem die Effizienz im Vordergrund, gewinnt mittlerweile die Resilienz zunehmend an Bedeutung [Ar21]. Damit einhergehend wird die Anpassungsfähigkeit immer wichtiger für Unternehmen, um auf Veränderungen innerhalb und außerhalb der Fabrik reagieren zu können.

Ein Produktionssystem mit gesteigerter Anpassungsfähigkeit ist z.B. das Cyber-Physische Matrixproduktionssystem (CPMPS) [Ke21; PI22]. CPMPS erweisen sich im Vergleich zu linearen Produktionssystemen als widerstandsfähiger gegenüber Nachfrage- und Kapazitätsschwankungen und stellen das Prinzip der Linienproduktion mit „Band und Takt“ infrage [Ba15b; Ke21]. Um liniengeführte Produktionssysteme während ihres Lebenszyklus an veränderte Marktanforderungen anzupassen, werden Produktionssysteme um- oder neugestaltet, was in der Regel nur während einer Produktionsunterbrechung erfolgen kann [Ba14].

Die Fähigkeit, dynamisch und kurzfristig auf unvorhergesehene Abweichungen zu reagieren, wird damit zu einem Wettbewerbsvorteil für Unternehmen. Dies erfordert einen ganzheitlichen Steuerungsansatz in einem horizontal und vertikal integrierten System. Dies soll durch Industrie 4.0 und den Einsatz von Cyber-Physischen Systemen (CPS) mit leistungsfähigen Informations- und Kommunikationstechnologien (IKT) ermöglicht werden [Ka13]. Statt individuell hochtechnisierter Einzelsysteme sollten autonome Module in Gesamtsystemen koordiniert werden und als Wertschöpfungsnetzwerke zusammenwirken [Ru21]. Erst durch die Balance von Effizienz und Resilienz können Ökosysteme erfolgreich und nachhaltig zusammenwirken [Li10].

1.1 Motivation und Problembeschreibung

In industriellen Produktionssystemen gilt die Linienproduktion seit der Einführung durch Henry Ford Anfang des 20. Jahrhunderts als Status Quo. Im Laufe der Jahre wurde der Automatisierungsgrad stetig erhöht und damit die taktgebundene Effizienz gesteigert. Mittlerweile existieren spezielle Methoden im Bereich der Produktionsplanung und -steuerung (PPS) für eine ganzheitliche Steuerung der liniengeführten Produktionssysteme [He12]. Die unterstützenden Planungs- und Steuerungssysteme mit dem Fokus auf einer detaillierten Planung und automatisierten Prozesssteuerung setzen die Methoden der PPS um und zeichnen sich vor allem durch den

Einsatz von robusten Automatisierungstechnologien und einer starr hierarchischen Auftragsplanung aus. Mit der Umsetzung alternativer Konzepte in der Automobilindustrie werden die bestehenden System- und Steuerungsarchitekturen infrage gestellt [Au16].

Die historisch gewachsenen monolithischen Steuerungssysteme für die automatisierte Linienfertigung erweisen sich aufgrund ihrer eingeschränkten Anpassungsfähigkeit als ungeeignet, um den Anforderungen vernetzter dynamischer Cyber-Physical Production Systems (CPPS) mit verteilten autonomen CPS gerecht zu werden [Tr19]. CPPS bestehen aus verteilten intelligenten Systemen, die die physische und die virtuelle Welt abbilden [Ba14]. CPPS erfordern digitale End-to-End Prozesse mit vertikaler und horizontaler Integration der Anlagen und Produkte innerhalb der Produktion [Ka13]. Für eine integrierte Produktionsplanung und -steuerung sollen die verteilten Systeme Entscheidungen autonom und in Abstimmung untereinander treffen können [Ru22]. Durch den Einsatz von Multiagentensystemen (MAS) könnten Möglichkeiten zur ganzheitlichen Entscheidungsfindung für die flexible Produktion über die Fabrikhalle hinaus ermöglicht werden [Ru21]. Matrixproduktionssysteme zeichnen sich durch einen hohen Flexibilitätsgrad aus, besitzen aber neben den verschiedenen und viel diskutierten Flexibilitätsarten (siehe z. B. [Da96; EIM05]) noch weitere Veränderungsfähigkeiten über die Produktionsebenen. Neben der angesprochenen Flexibilität sind noch die Wandlungsfähigkeit, Rekonfigurierbarkeit und Umrüstbarkeit zu nennen, die bei der Entscheidungsfindung auf mehreren Ebenen berücksichtigt werden müssen (siehe Abbildung 1.1 [PI22]). Zur Umsetzung der Matrixproduktion besteht vor allem Forschungsbedarf im Bereich der autonomen Optimierung und Rekonfiguration [PI22]. Dies bedeutet vor allem eine Koordination der Arbeitsschritte unter Berücksichtigung der system- und ebenenspezifischen Veränderungsfähigkeiten auf den Ebenen der Matrixproduktion über das Prozessmodul bis zum Ressourcenmodul.

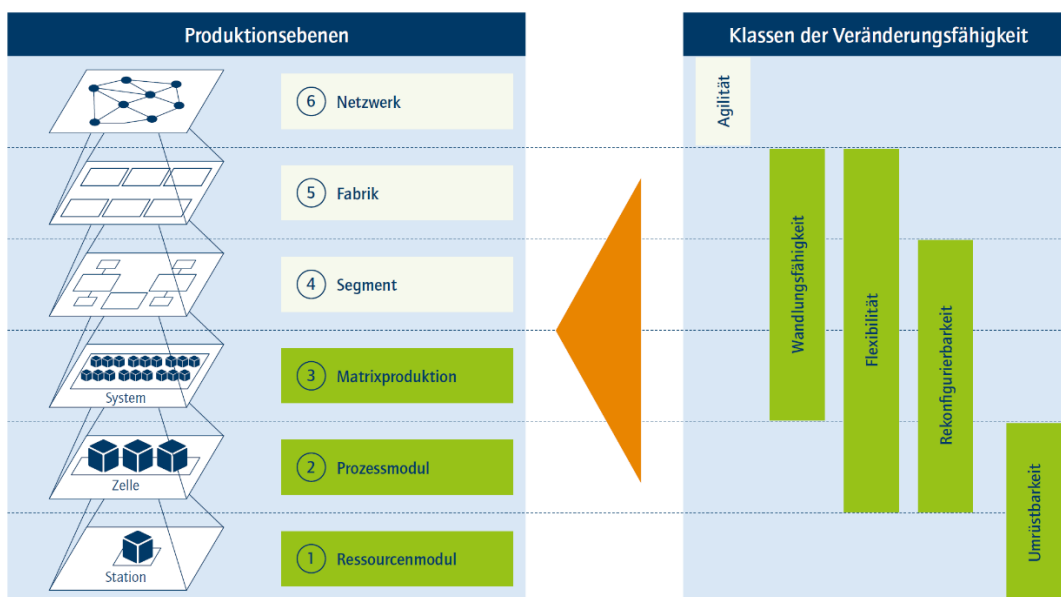


Abbildung 1.1: Struktur und Veränderungsfähigkeiten von CPMPs nach [PI22] in Anlehnung an [Wi09; Wi14]

CPMPS sollen die statischen und dynamischen Bedarfe und Kapazitäten eines integrierten Gesamtsystems für eine dynamische Optimierung des flexiblen Produktions- und Materialflusses nutzen [Gr14; Sc15; KRB16; TFB20]. Ein Zusammenschluss mehrerer Systeme kann gemäß des System Engineering als „System of Systems“ (SoS) betrachtet werden [Ha19]. Gemäß der bereits existierenden Bezeichnung von CPSoS bzw. CPPSoS (siehe [We15]) können CPMPS im Sinne integrierter Gesamtsysteme in dieser Arbeit als CPMPSoS bezeichnet und betrachtet werden.

Eine ganzheitliche Betrachtung der system- und ebenenspezifischen Veränderungsfähigkeiten geht allerdings mit einer gesteigerten Anzahl an Freiheitsgraden und einer Vielzahl an Entscheidungsmöglichkeiten einher. Dadurch steigt die Komplexität einer ganzheitlichen Produktionsplanung und -steuerung von Matrixproduktionssystemen. Diese benötigen ein anpassungsfähiges Steuerungssystem mit integrierter Auftragsplanung und Auftragsausführung, welches die einzelnen Assets in einem Netzwerk koordinieren kann. Dafür sollten die Informationen über die herzustellenden Produkte, deren Standort sowie die bereits durchgeführten und noch ausstehenden Arbeitsschritte berücksichtigt werden können [Ru21].

Um die Komplexität in dynamischen Systemen zu bewältigen, bedarf es nach dem Gesetz von Ashby eines Steuerungssystems mit vergleichbarer interner Komplexität [Sc96; Pl22]. Ein Steuerungssystem von der Produktionsplanung bis zur Prozessautomatisierung von CPMPS ist bisher nicht am Markt zu finden [Pl22]. Die Realisierung einer integrierten Steuerungslösung bedarf komplexitätsbeherrschender Maßnahmen mit intelligenter Entscheidungsfindung und Orchestrierung der Produktionsprozesse. Einen Ansatz zur Darstellung einer RAMI-konformen Fertigungsarchitektur für Unternehmen mit integrierter Auftragsplanung und Auftragsausführung ist in Abbildung 1.2 dargestellt. Die fähigkeitsbasierte Auftragsplanung und Auftragsausführung von Cyber-Physischen Produktionsmodulen (CPPM) in einer serviceorientierten Architektur kann mit Multi-Agenten-Systemen gesteuert werden.

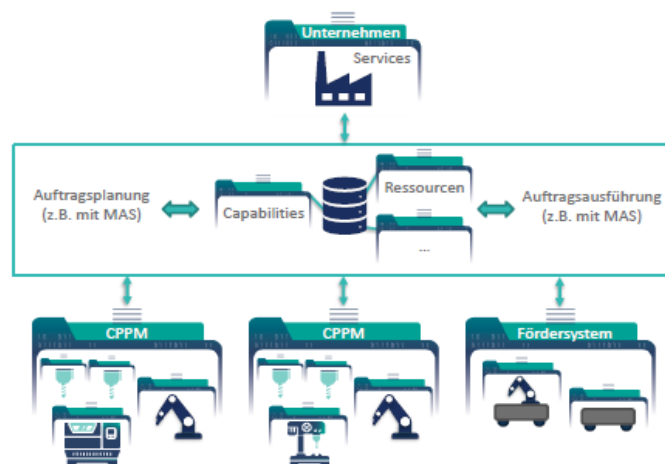


Abbildung 1.2: Darstellung der RAMI-konformen Fertigungsarchitektur eines Unternehmens von [Vo23]

1.2 Zielsetzung und Forschungsfragen

Das Ziel dieser Arbeit ist die Befähigung eines autonom gesteuerten CPMPs als System of Systems (CPMPSoS) mit einer geeigneten Steuerungsarchitektur. Durch die Auflösung der starr verketteten Ablauffolgen gewinnt ein CPMPs mehr Flexibilität für eine auftragsindividuelle Produktion [Au16; KRB16]. Die Herausforderung besteht darin, die verschiedenen systemindividuellen Anpassungsmöglichkeiten für die Auftragsbearbeitung zu koordinieren. Dazu wird untersucht, wie eine Steuerungsarchitektur aufgebaut sein sollte, die in der Lage ist, CPMPs unter Berücksichtigung system- und ebenenspezifischer Freiheitsgrade autonom optimieren und rekonfigurieren zu können [PI22]. Der hierarchische Betrachtungsraum der Arbeit konzentriert sich auf die Work Center Ebene, die Station Ebene und die Control Device Ebene, wie in Abbildung 1.3 dargestellt.

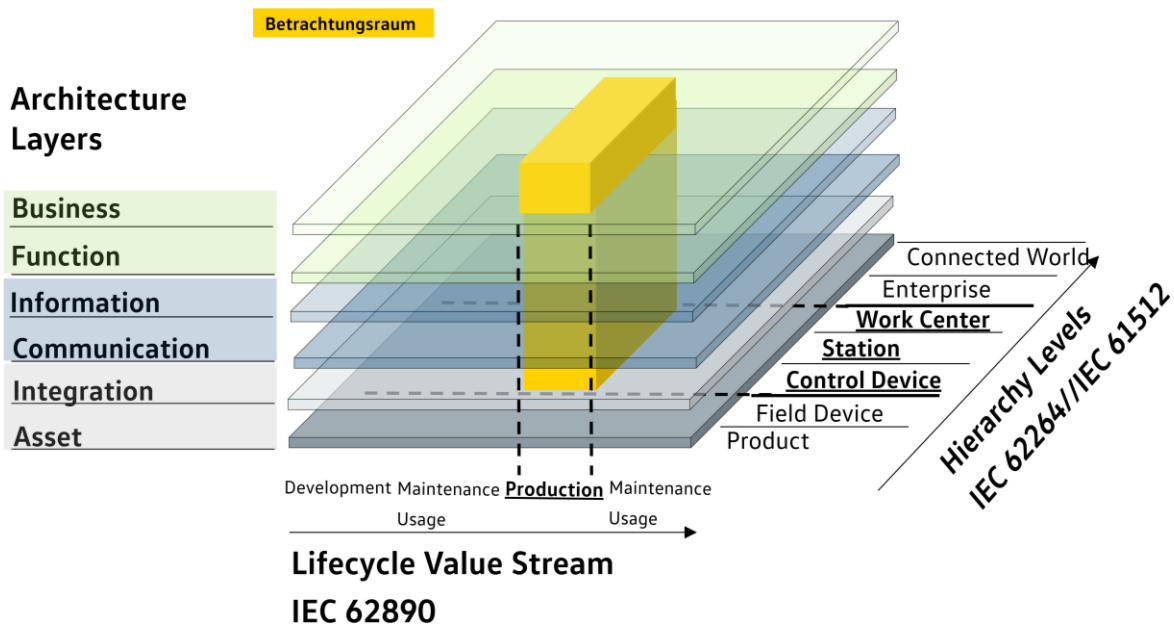


Abbildung 1.3: Einordnung des Betrachtungsraums der Arbeit (in gelb) nach RAMI4.0 [DIN SPEC 91345]

Das RAMI 4.0 stellt einen Ansatz zur Entwicklung von Industrie 4.0 (I4.0) fähigen Systemen und CPPS dar. Während zuvor Architekturen für spezifische Anwendungsfälle entwickelt wurden, bietet das RAMI 4.0 einen generischen und technologieneutralen Startpunkt für die Entwicklung spezifischer I4.0 Architekturen und Systeme [Tr19]. Das Modell kann ebenfalls als Grundlage zur Einordnung von Technologien und zur Schaffung eines einheitlichen Verständnisses sowie der Identifikation von Interaktionen und der Verortung von Standards und Normen verwendet werden [Sc19b]. Für die Entwicklung von Steuerungsarchitekturen gemäß der I4.0-Initiative von der Informationsverarbeitungs- bis zur Automatisierungsebene ist daher die Definition von Standards und das Ableiten von Entwurfsmustern wesentlich, ähnlich der Entwicklung des Internets in den 90er Jahren [EH10].

Im RAMI4.0 (DIN SPEC 91345) wird über eine dreidimensionale Zusammenführung der wichtigsten Aspekte von Industrie 4.0 ein gemeinsames Verständnis zur Beschreibung von Assets und deren Kombination aufgebaut. Besonders bei Betrachtung der Hierarchie-Achse zeigt sich der Unterschied zur starr hierarchisch organisierten Automatisierungspyramide. Die Assets, wie Anlagen und Maschinen oder auch Produkte und Aufträge, bilden durch horizontale und vertikale Beziehungen ein interoperables Netzwerk. Die Steuerungsarchitektur eines CPMPS soll unter Berücksichtigung der individuellen Veränderungsfähigkeiten seiner modular hierarchisch strukturierten Assets eine autonome Optimierung und Rekonfiguration ermöglichen. Dabei sollen die CPMPS gemäß der Industrie 4.0 Initiative horizontal und vertikal integrierte Gesamtsysteme bilden [Ka13]. Für die vertikale Integration muss das CPMPS von der Matrixproduktionssystemebene über die Prozessmodulebene bis zur Ressourcenmodulebene berücksichtigt werden. Eine abstrahierte Darstellung des CPMPS als RAMI-konformes modular hierarchisches System ist in Abbildung 1.4 dargestellt. Dies kann nach RAMI4.0 in die Hierarchieebenen von der Work Center Ebene bis zur Control Device Ebene definiert werden und ähnelt in der Darstellung dem fraktalen Aufbau nach Warnecke [Wa92].

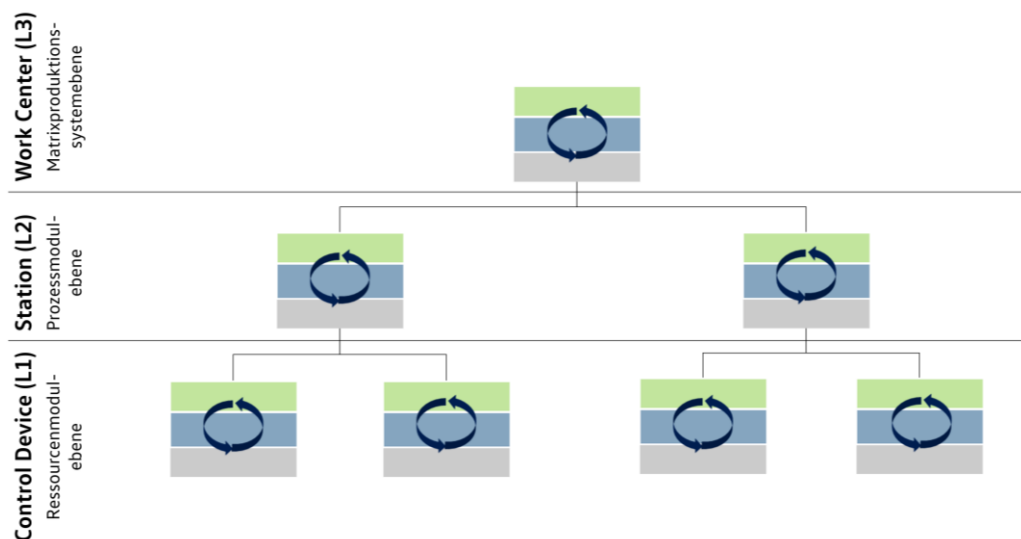


Abbildung 1.4: Abstraktion eines CPMPS als RAMI4.0 konformes modular hierarchisches System

In dieser Arbeit soll erforscht werden, welche Möglichkeiten für eine autonome ganzheitliche Entscheidungsfindung in einem CPMPS existieren. Das CPMPS wird dazu als modular hierarchisches System of Systems mit seinen Stationen und Control Devices modelliert. CPMPS benötigen integrierte Steuerungslösungen zur autonomen Optimierung und Rekonfiguration auf horizontaler und vertikaler Ebene. Dazu sollen die folgenden Forschungsfragen in dieser Arbeit beantwortet werden.

Forschungsfragen:

1. Ein CPMPS besteht aus mehreren Ebenen mit verteilten Produktionsressourcen. Wie können die Abläufe in einem industriellen modular hierarchischen Produktionssystem unter Berücksichtigung zustandsaktueller Ereignisse autonom rekonfiguriert werden?

2. CPMPs besitzen charakteristische Veränderungsfähigkeiten mit system- und ebenenspezifischen Freiheitsgraden. Wie können CPMPs ganzheitlich optimiert werden und wie kann die Komplexität beherrscht werden?

1.3 Vorgehensweise und Struktur der Arbeit

Die methodische Vorgehensweise zur Erstellung der Steuerungsarchitektur orientiert sich an der Systems Engineering Method [Ko11] und gliedert sich wie in Abbildung 1.5 auf Kapitelebene und Abbildung 1.6 bis auf Abschnittsebene dargestellt in Einleitung (Kapitel 1), Grundlagen und Stand der Forschung (Kapitel 2), Modellierung (Kapitel 3), Implementierung (Kapitel 4), Validierung (Kapitel 5) und Zusammenfassung (Kapitel 6). Ziel ist die strukturierte Modellierung einer Steuerungslösung (Solution) ausgehend des Forschungsbedarfs (Need) mit rekursiver Beziehung zwischen den einzelnen Kapiteln (siehe Abbildung 1.5).

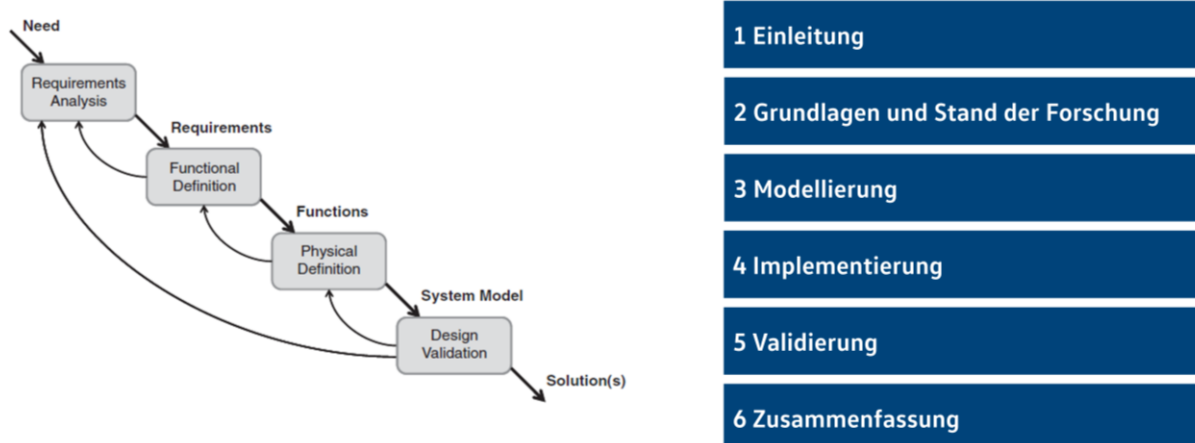


Abbildung 1.5: Struktur der Arbeit in Anlehnung an das Top-Level Flow Diagramm aus [Ko11]

In Kapitel 1 wurde in der Einleitung die Problemstellung zur Steuerung von CPMPs und der damit verbundenen Zielstellung mit den zu beantwortenden Forschungsfragen vorgestellt. Im Anschluss in Kapitel 2 werden wegen der systemischen Betrachtung eines CPMPs als SoS die wesentlichen Grundlagen der Systemtheorie und der autonomen Steuerung vorangestellt. Anschließend werden die Grundlagen von CPMPs betrachtet mit den charakteristischen Eigenschaften dieser Organisationsform und den unterschiedlichen Veränderungsfähigkeiten. Die relevanten Grundlagen und Ansätze zur autonomen Steuerung von CPMPs werden im Anschluss als veränderungsfähige CPPS zweigeteilt untersucht. Es werden zunächst veränderungsfähige CPPS bezüglich relevanter Forschungsbereiche zur autonomen Steuerung analysiert. Es werden die Bereiche der Steuerungsarchitekturen sowie der PPS für veränderungsfähige CPPS untersucht. Darauf aufbauend wird der Stand der anwendungsorientierten Forschung zur autonomen Steuerung von CPMPs analysiert. Dazu werden Vorhaben und Projekte zur Steuerung von CPMPs gezielt auf ihre

Betrachtung von mehreren Ebenen und der Berücksichtigung von Veränderungsfähigkeiten untersucht. Als Ergebnis werden die gewonnenen Erkenntnisse zusammengefasst und die weiteren Anforderungen an eine Steuerungsarchitektur für autonome CPMPs herausgestellt.

Die Modellierung der Steuerungsarchitektur erfolgt in Kapitel 3 und definiert das grundlegende Steuerungsverhalten durch die Definition des Entscheidungsmodells, des Agentenmodells, des Interaktionsmodells und des Optimierungsmodells. Darauf aufbauend findet eine Implementierung der Steuerungsarchitektur in Kapitel 4 als Simulation statt, bevor diese in Abschnitt 5 in Betrieb genommen und mithilfe von Simulationsexperimenten validiert, eingeordnet und bezüglich der Anforderungserfüllung evaluiert wird. In Kapitel 6 wird die Arbeit zusammengefasst und ein Fazit gezogen, bevor ein Ausblick für die weitere Forschung gegeben wird. Das Vorgehen ist schematisch in Abbildung 1.6 dargestellt.

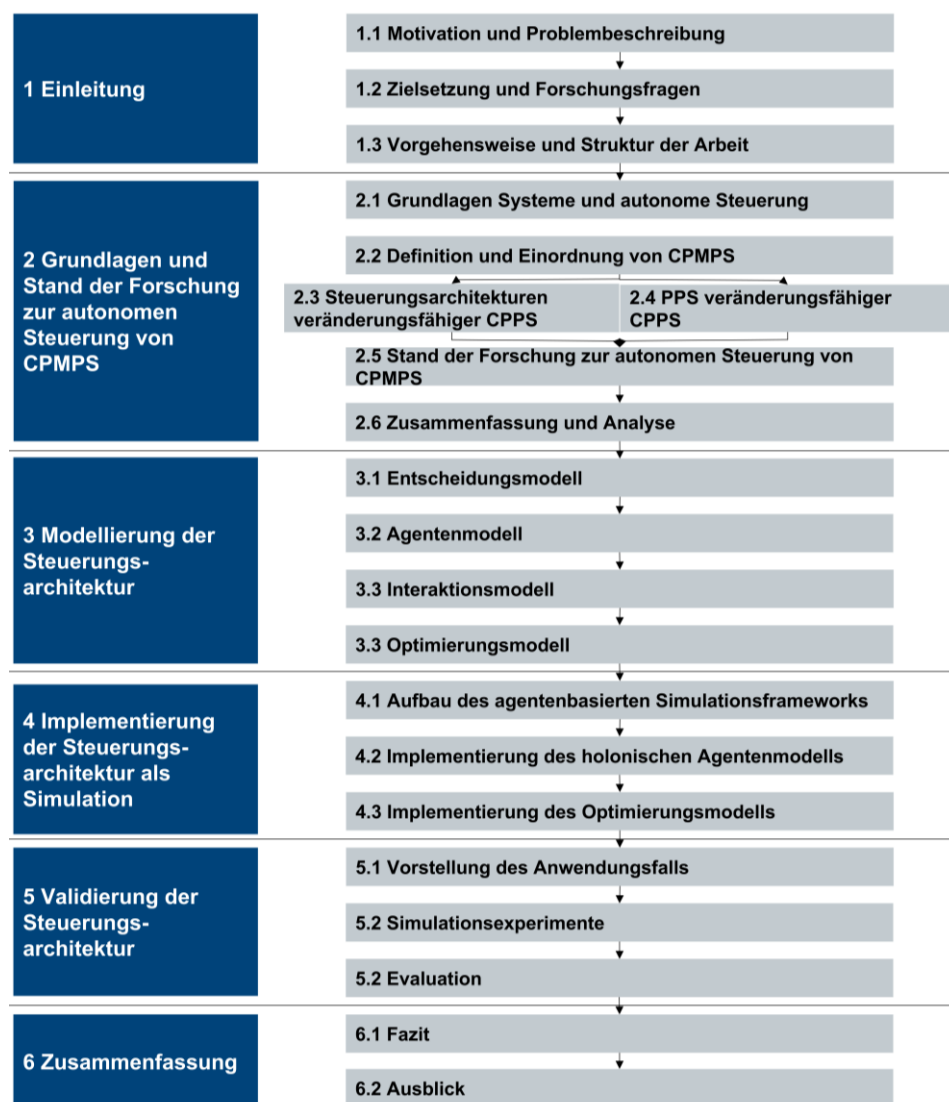


Abbildung 1.6: Vorgehensweise strukturiert nach Kapiteln und Abschnitten

2 Grundlagen und Stand der Forschung zur Steuerung von CPMPS

Ein CPMPS kann als ein System bestehend aus mehreren Systemen auf mehreren Ebenen verstanden werden. In Abschnitt 1.1 wurde bereits die Herausforderung im Umgang mit komplexen Systemen zur Produktionsplanung im Kontext verteilter CPS in CPPS oder CPMPS einleitend beschrieben. In dieser Arbeit wird eine systemorientierte Betrachtung der CPMPS vorgenommen und die CPMPS als System of Systems bzw. Cyber-Physisches Matrixproduktionssystem of Systems (CPMPSoS) betrachtet. Für den weiteren Umgang werden die wichtigsten Grundlagen und Eigenschaften des Systems Engineerings, Systems Thinking und der Systems Theory im Kontext der „System of Systems“ (SoS) und der autonomen Steuerung in Abschnitt 2.1 vorgestellt. Im Anschluss daran erfolgt die Definition und Einordnung von CPMPS mit Fokus auf den Herausforderungen für die Steuerung durch Betrachtung der Veränderungsfähigkeiten in Abschnitt 2.2. Darauf aufbauend werden in Abschnitt 2.3 und 2.4 zwei zentrale Bereiche zur Steuerung für veränderungsfähige CPPS betrachtet und erste Ansätze präsentiert. Abschnitt 2.3 stellt die Grundlagen und Ansätze mit Nähe zur rekonfigurierbaren Prozess- und Ressourcensteuerung vor, während Abschnitt 2.4 den Bereich der Produktionsplanung und -steuerung mit Fokus auf der Optimierung von flexiblen Abläufen behandelt. In Abschnitt 2.5 wird der Stand der angewandten Forschung der autonomen Steuerung von CPMPS analysiert. Abschließend erfolgt in Abschnitt 2.6 eine Zusammenfassung der Erkenntnisse und eine Analyse der zu berücksichtigenden Aspekte für die weitere Gestaltung der Steuerungsarchitektur im Zusammenhang mit der weiteren Bearbeitung der Forschungsfragen.

2.1 Grundlagen zu Systemen und autonomer Steuerung

Der Systembegriff umfasst drei grundlegende, einander ergänzende Konzepte der Systemtheorie, die in Abbildung 2.1 dargestellt sind. Es handelt sich um das funktionale, das strukturelle und das hierarchische Systemkonzept. Dem Systems Engineering zugrunde liegt das Systems Thinking, welches Ansätze zur Strukturierung von Systemen liefert [Ha19].

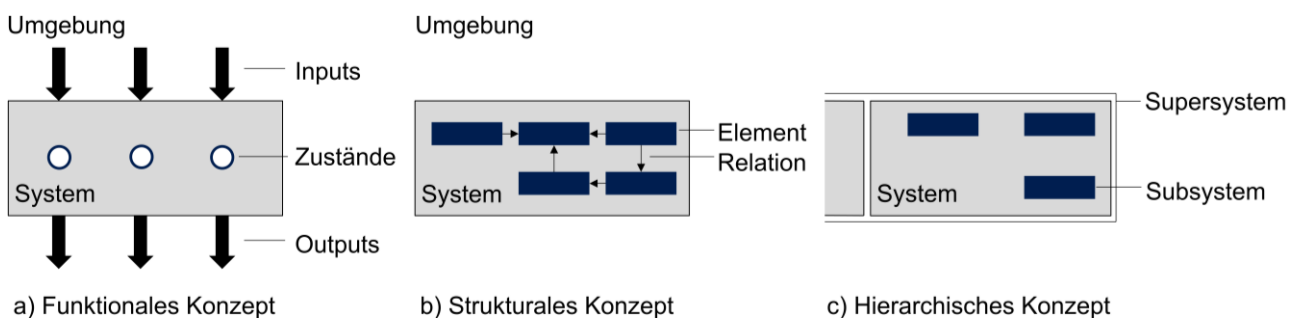


Abbildung 2.1: Konzepte der Systemtheorie nach [Rop09]

Ein System besteht aus einer Vielzahl an Elementen mit Eigenschaften und Handlungsmöglichkeiten sowie einer Menge von Beziehungen zwischen diesen Elementen [Ro09]. Die Elemente des Systems operieren zweckorientiert. Die Vernetzung der Beziehungen der Systemelemente wird als Struktur bezeichnet. Jedes System kann in logische Subsysteme niederer Ordnung zerlegt werden. Können Elemente weiterhin in eine andere Beziehungsform gesetzt werden, spricht man von Teilsystemen. Ein System wird grundsätzlich durch eine Systemgrenze gegenüber seiner Umwelt abgegrenzt. Liegen keine Eingangs- und Ausgangsgrößen vor, so spricht man von einem geschlossenen System. Anderenfalls handelt es sich um ein offenes System [He03].

Das funktionale Konzept oder „effect-oriented view“ setzt den Fokus der Betrachtung auf die Zusammenhänge zwischen seinen äußeren Eigenschaften und stellt das System als eine Black Box dar, welches Input in Output umwandelt [Ro09]. Systeme können nach dem hierarchischen Konzept weiter detailliert und in Subsysteme unterschiedlicher Hierarchien strukturiert werden. Das strukturelle Konzept definiert das System als eine Einheit von miteinander interagierenden Elementen innerhalb der Systemgrenze. Die Eigenschaften eines Systems ergeben sich somit aus den individuellen Eigenschaften der Elemente und aus dem Zusammenwirken dieser.

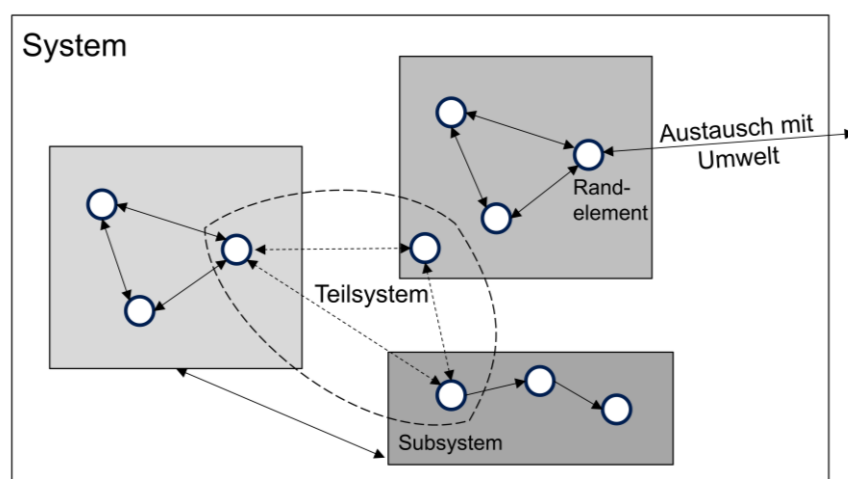


Abbildung 2.2: Begriffe der Systemtheorie nach [Mue09]

Dieses strukturelle Systemdenken baut auf dem holistischen Gesetz auf, nachdem das Ganze mehr als bloß die Summe seiner Einzelteile darstellt [Ro09]. Grundsätzlich kann zwischen hierarchischen und heterarchischen Beziehungen zwischen den Systemen und Teilsystemen unterschieden werden [Ha19].

Maier führte 1996 erstmals eine formale Diskussion über SoS ein, indem er deren Merkmale identifizierte. Die für diese Arbeit relevante Definition umfasst die Zusammenfassung der Kriterien von [Ma96; Ma98] durch [SC01], die u. a. in [Ko11] und [We15] zu finden sind:

1. Locker gekoppelte Elemente (Loose Coupling & Cohesion): Ein SoS besteht aus locker gekoppelten Komponentensystemen, die hauptsächlich über Datenaustausch verbunden sind und oft geografisch verteilt sind.

2. Emergentes Verhalten (Emergent Behavior): Das Verhalten von SoS entsteht durch die Interaktion seiner Systeme und ist mehr als die Summe ihrer individuellen Fähigkeiten.
3. Operative Unabhängigkeit (Operational Independence): Systeme können unabhängig außerhalb des SoS arbeiten oder sich anderen SoS anschließen.
4. Unabhängiges (Selbst-)Management (Managerial Independence): Komponentensysteme werden unabhängig entwickelt, verwaltet und haben eigene Lebenszyklen.
5. Evolutionäre Entwicklung (Evolutionary Development): Ein vollständig entwickeltes SoS existiert selten. Die Entwicklung ist evolutionär, wobei Komponenten je nach Bedarf hinzugefügt, entfernt oder modifiziert werden.

Die beiden folgenden Merkmale wurden nachträglich ergänzt und sind in dieser Arbeit für die autonome Optimierung von CPMPS von großer Bedeutung.

6. Selbstorganisation: Ein SoS verfügt über eine dynamische Organisationsstruktur, die auf Veränderungen in der Umgebung und auf interne Änderungen der Ziele und Zwecke des SoS reagieren kann.
7. Anpassung: Ähnlich wie eine dynamische Organisation ist die (physische) Struktur des SoS dynamisch und kann auf externe Veränderungen und Wahrnehmungen der Umgebung reagieren.

Die Selbstorganisation und Anpassung eines SoS ist eine zentrale Eigenschaft für die Entscheidungsfindung. Ein System of Systems kann in vier Kategorien eingeteilt werden [Ma96; Ma98]:

- Virtuell: Die Systeme sind durch lockere Kopplung verbunden und streben nach Zielen ohne eine zentrale Autorität oder ohne einen gemeinsamen Zweck, sondern durch unsichtbare Mechanismen.
- Kollaborativ: Die Systeme kollaborieren freiwillig für gemeinsame Zwecke, ohne die zentrale Durchsetzung von Standards.
- Anerkannt: Die Systeme verfolgen anerkannte Ziele, aber behalten dabei ihre Unabhängigkeit und dezentrale Kontrolle.
- Gelenkt: Das SoS besitzt eine zentrale Verwaltung für spezifische Zwecke, wobei Systeme ihre Unabhängigkeit beibehalten, aber dem zentralen Zweck untergeordnet sind.

Die Begriffe Selbstorganisation oder Autonomie werden mitunter redundant verwendet. Der Begriff Autonomie kann in der Produktion als die Freiheit von Entscheidungen und Handlungen in einer klar definierten und abgegrenzten Umgebung innerhalb der Gesamtorganisation definiert werden [WWB98]. Das Ziel besteht darin, eine möglichst störungsresistente Steuerung der Produktionsentität zu erreichen, ohne direkte Einwirkung eines Mitarbeiters. Die Kombination autonomer Ressourcen einerseits und autonomer Teile, Baugruppen und Produkte andererseits führt zu autonomen Prozessen, bei denen intelligente Assets ihre Montage selbst steuern, wie in Abbildung 2.3 dargestellt ist [SF07]. Die Koordination verteilter autonomer Prozesse kann nach dem Prinzip der Subsidiarität erfolgen, das auf die Produktion angewendet werden kann und besagt,

dass die Entscheidungsfindung so niedrig wie möglich und nur so hoch wie nötig angesiedelt sein sollte [Be22].

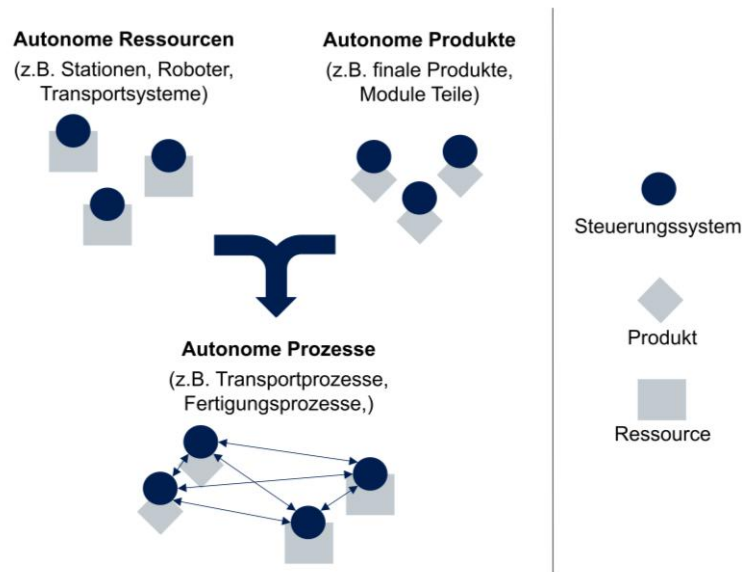


Abbildung 2.3: Autonome Ressourcen, Produkte und Prozesse nach [SF07]

Für eine autonome Steuerung kann auch ein selbstoptimierender, adaptiver Regelkreis für das Systemverhalten zugrunde gelegt werden [GES05; Bö06]. Das System kann durch selbstoptimierende Regelung, welche durch äußere Einflüsse ausgelöst wird, oder eine adaptive Regelung, welche durch interne Einflüsse ausgelöst wird, gesteuert werden. Dabei entscheidet ein Adaptionsalgorithmus wie in Abbildung 2.4 dargestellt, entweder über neue Sollwertvorgaben im Sinne einer Selbstoptimierung oder passt bestehende Sollwertvorgaben durch einen adaptiven Regler innerhalb des Systems an.

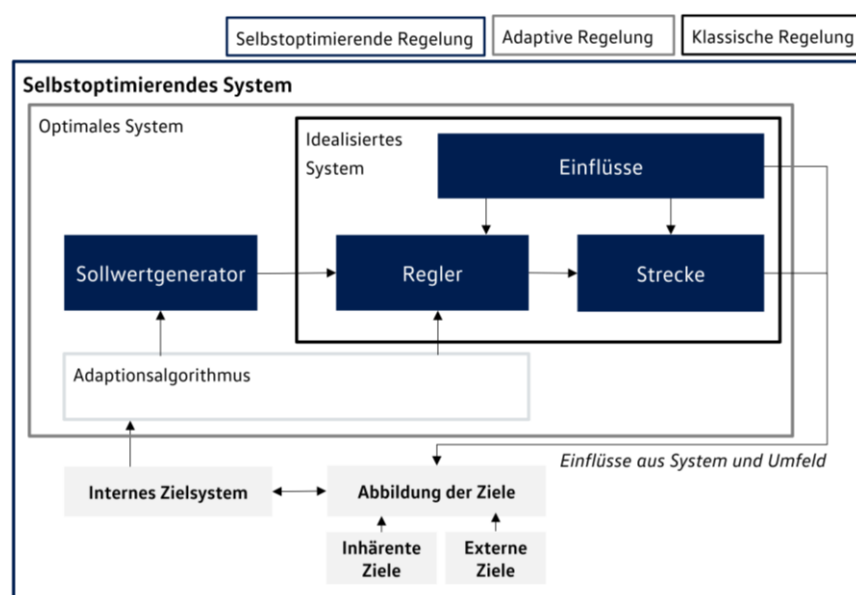


Abbildung 2.4: Selbstoptimierende Systeme nach [Bö06]

In einem industriellen Produktionssystem sollen die Ziele einer hohen Stückzahl, geringer Durchlaufzeit, hoher Termintreue und optimaler Auslastung erreicht werden. Inhärente Ziele sind die Ziele, die dem Entwurfszweck des Systems dienen und die Grundfunktionalität sichern, wie z.B. grundsätzlich die Herstellung von Produkten. Von außen herangetragene externe Ziele sind übergeordneter Natur wie z. B. die Zielvorgabe zur Stückzahlerreichung. Das interne Zielsystem vereint die unterschiedlichen Einflüsse und ist ausschlaggebend für das Optimierungsverhalten des Systems.

Grundsätzlich kann zwischen drei Steuerungsarten unterschieden werden (Abbildung 2.5). Der zentralisierte Ansatz einer hierarchisch organisierten Architektur, der dezentralisierte Ansatz mit rein heterarchischer Architektur sowie den Mischformen, die auch als hybride oder semi-heterarchische Architekturen beschrieben werden. Wie in Abbildung 2.5 dargestellt, wirken Steuereinheiten dabei auf unterschiedliche Weise auf Produkt und Ressource ein.

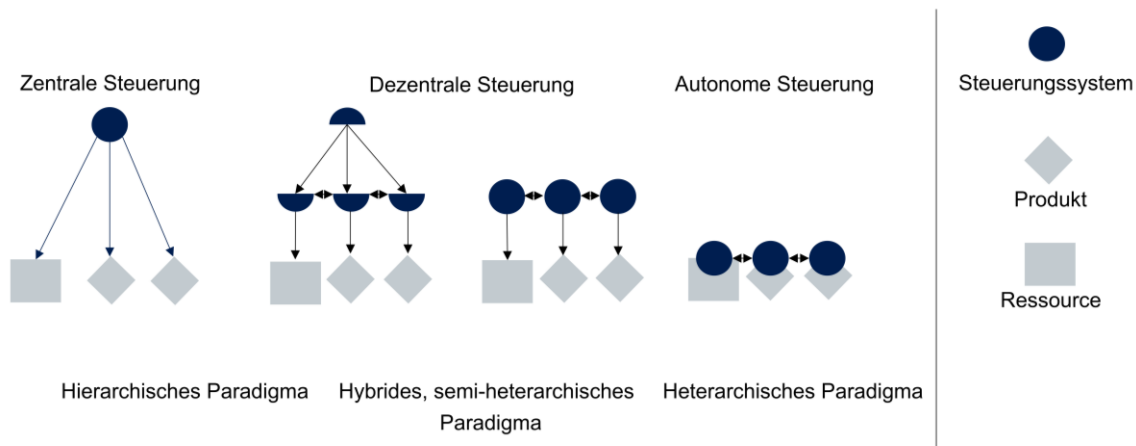


Abbildung 2.5: Steuerungsarten nach [SF07]

Hierarchische Steuerungsstruktur: Die hierarchische Steuerung existiert sowohl als rein zentralisierter als auch als (teil-)dezentralisierter Steuerungsansatz. In diesem Ansatz wird eine übergeordnete Einheit für die globale Steuerung verwendet. Bei vollständig zentralisierten Strukturen kann die auszuführende Ressource nicht selbstorganisiert auf Störungen reagieren. Stattdessen müssen Abweichungen an die übergeordnete Einheit weitergeleitet werden, um neue zentrale Anweisungen zu erhalten [VvB15]. Insbesondere bei starken Wechselwirkungen der untergeordneten Systeme ist dies oft nur mit der zentralen Steuerung möglich [Re17]. Bei steigender Komplexität allerdings neigen die zentralen Steuerungen zu längeren Reaktionszeiten [Un15]. Die hierarchische Steuerung folgt dem zentralen Ansatz mit streng hierarchischem Kommunikationsmuster nach dem Prinzip „top-down“ und „bottom-up“. Eine Kooperation ist nur in einem zuvor definierten Maße möglich [Un15].

Heterarchische Steuerungsstruktur: Die heterarchische Steuerung bietet durch eine rein dezentrale Steuerung Flexibilitätsvorteile durch verkürzte Entscheidungswege und minimierte Reaktionszeiten. Im Vergleich zur hierarchischen Steuerung können frühe Entscheidungen vorteilhaft

angesichts möglicher langer Optimierungszeiten sein [Sp13]. Bei lokalen Störungen ist die dezentrale Steuerung in der Lage selbstorganisiert zu reagieren. Dies führt zu erhöhter Robustheit in dynamischen Umgebungen. Durch die dezentrale Entscheidungsfindung wird die Komplexität des Planungsproblems reduziert [Os12]. Die logistische Leistungsfähigkeit ist allerdings nicht zwangsläufig höher als bei einer zentralen Steuerung, sondern kann dort, wo zentrale Optimierung möglich und erforderlich ist, leistungsbezogene Nachteile mit sich bringen [Re17].

Hybride Steuerungsstruktur: Die hybride Steuerung, auch als semi-heterarchisch bezeichnet, kombiniert Elemente der zentralen und dezentralen Steuerung. In Bezug auf das Produktionssystem kann die Grobplanung zentral erfolgen, während reaktionsschnelle Ablaufänderungen dezentral durchgeführt werden [VvB15; Re17]. Holonische Produktionssysteme setzen diesen hybriden Ansatz um, indem sie zentrale und dezentrale Steuerungsvorteile nutzen. Holone sind autonome, selbstregulierende und kooperative Blöcke in Produktionssystemen. Sie bestehen in der Regel aus einem informationsverarbeitenden und einem physikalischen Teil [Ch94; CM00]. Holonische Produktionssysteme folgen dem „Divide and Conquer“-Prinzip, bei dem eine zentrale Instanz Aufgaben in Teilprobleme zerteilt und diese dezentralen Problemlösern zuweist. Die Lösungen der Teilprobleme werden dann in das zentrale Problem integriert [FSS03]. Dieses Prinzip der Komplexitätsreduktion ermöglicht es, verschiedene Ebenen eines Produktionssystems abzubilden [Sc19a].

Die Vor- und Nachteile der Steuerungsparadigmen wurden von Cruz et al. in Bezug auf verschiedene Merkmale des Systemverhaltens in Tabelle 2.1 gegenübergestellt [Cr19]. Dabei sind die Paradigmen in hierarchisch und (semi-)heterarchisch unterteilt. Eine dynamische Steuerungsarchitektur zeichnet sich durch die Synergie der Vorteile beider Paradigmen aus. In der Tabelle 2.1 werden ebenso Auswertungen nach Trentesaux und Leitão berücksichtigt [LR08; Tr09].

Tabelle 2.1: Merkmale hierarchischer (Kl. 0-1) und semi-heterarchischer (Kl. 2-3) Steuerungen nach [Cr19] und [LR08; Tr09]

Merkmal	Klasse	
	0-1	2-3
Anpassungen und schnelle Rekonfigurierbarkeit	-	++
Produktverfolgung und Einsatz intelligenter Produkte	-	++
Robustheit bei unvorhergesehenen Änderungen am System	-	++
Optimierung der Ressourcenauslastung	-	++
Vorhersagbarkeit, Ermittlung analytischer Lösung	++	-
Flexibilität und Reaktivität auf Störungen	-	++
Adressierung des globalen Optimums	++	-
Erweiterbarkeit des Systems	-	++

2.2 Definition und Einordnung von Cyber-Physischen Matrixproduktionssystemen

Ein CPMPs wird in dieser Arbeit als ein modular hierarchisches System of Systems (SoS) betrachtet, in dem die verteilten Systeme hierarchisch verschachtelte Beziehungen aufweisen. CPMPs werden laut [PI22] folgendermaßen definiert:

Die **Cyber-Physischen Matrixproduktionssysteme (CPMPs)** besitzen ein charakteristisches Layout in Form einer Matrix und bestehen aus flexibel verketteten, ortsfesten Prozessmodulen, die logistisch beplanbar sind. Die Prozessmodule wiederum bestehen aus nicht logistisch beplanbaren Ressourcenmodulen. CPMPs zeichnen sich durch Möglichkeiten zur Veränderung auf unterschiedlichen Systemebenen aus. So entsteht ein verschachteltes Produktionssystem mit unterschiedlichen systemspezifischen Freiheitsgraden [PI22].

Neben den CPMPs existieren die Matrixproduktionssysteme auch als Software-definierte Matrixproduktionssysteme (SDMPs). Die SDMPs zeichnen sich ebenfalls durch eine modulare Struktur und die Verwendung einer automatisierten, optimierten Prozesssteuerung aus. Aufgrund von technologischen Restriktionen oder der Hardware-Konfiguration der Produktionsressourcen sind sie allerdings, im Gegensatz zu CPMPs, in der Regel nicht rekonfigurierbar und weisen einen sehr hohen Automatisierungsgrad auf. Häufig entwickeln sich SDMPs aus herkömmlichen Werkstattfertigungen, die um eine komplexe Werkstattsteuerung erweitert werden [PI22].

Umsetzungsbeispiele für CPMPs finden sich bisher vor allem für die Produktion kleiner Stückzahlen und hoher Variantenvielfalt in der Kleinserienfertigung oder auch für den Anlauf neuer Produktvarianten in der Vorserienfertigung [Au16; Ke21]. Umsetzungsbeispiele der SDMPs sind in der Serienfertigung von kleinteiligen Produkten mit hoher Änderungsdynamik wie der Halbleiterfertigung zu finden [Wa20]. Die Systeme werden in Umgebungen mit hoher Änderungsdynamik und gesteigerten Individualitätsanforderungen des Marktes verwendet. Bisher wurden CPMPs hauptsächlich im Bereich der Montage umgesetzt [PI22].

Im Vergleich zu klassischen Produktionssystemen kann mit CPMPs z. B. eine höhere Flexibilität bei ebenfalls höherer Produktivität erreicht werden. Abbildung 2.6 gibt einen Überblick über die Unterschiede der Organisationsformen. Ein CPMPs unterscheidet sich in der Ablauforganisation nicht von einem verketteten Produktionssystem mit Einzelplätzen oder einer Werkstatt- bzw. Gruppenfertigung. Im Gegensatz zu den Einzelplatz- und Werkstattssystemen, wo die Aufträge zu Losen starr zusammengefasst werden, besitzen die Produktionsobjekte in der Matrixproduktion hingegen individuelle Ablaufoptionen. Die flexiblen Ablaufoptionen und rekonfigurierbaren Systemzustände steigern neben der Materialflusskomplexität auch die Automatisierungskomplexität des Produktionssystems. CPMPs eignen sich sowohl für standardisierte als auch konfigurierbare Produkte mit unterschiedlicher Teilevarianz. Es ist aber zu beachten, dass für ein Szenario

mit standardisierten Produkten und geringer Teilevarianz über einen bekannten Einsatzzeitraum eine Linienproduktion wirtschaftlicher ist [PI22].

Kriterien	Einzelplätze	Werkstatt	Gruppe	Linie	Cyber-physische Matrixproduktion	Fluide Produktion
Verkettung	keine	flexibel	flexibel	starr	flexibel	flexibel
Stationäre Elemente	Produktionsobjekt, Arbeitskräfte, Betriebsmittel, Gebäudeinfrastruktur	Betriebsmittel, Gebäudeinfrastruktur	Betriebsmittel, Gebäudeinfrastruktur	Betriebsmittel, Gebäudeinfrastruktur	Betriebsmittel, Gebäudeinfrastruktur	Gebäudeinfrastruktur
Bewegliche Elemente	keine	Produktionsobjekt, Arbeitskräfte	Produktionsobjekt, Arbeitskräfte	Produktionsobjekt, Arbeitskräfte	Produktionsobjekt, Arbeitskräfte	Produktionsobjekt, Arbeitskräfte, Betriebsmittel
Funktionsumfang der Betriebsmittel & Zeitbezug der Änderung	fest definiert	fest definiert	fest definiert	fest definiert	periodisch rekonfigurierbar	dynamisch rekonfigurierbar
Segmentierungsmerkmal der Ressourcen	Produkt	Technologie	Produkt	Produkt	Prozesse	Auftrag
Zeitpunkt der Ressourcenzuordnung	vor Auftragsdurchlauf	vor Auftragsdurchlauf	vor Auftragsdurchlauf	vor Auftragsdurchlauf	während Auftragsdurchlauf	während Auftragsdurchlauf
Losgrößen (im Regelfall)	Batch, One-piece-flow	Batch	Batch, One-piece-flow	Batch, One-piece-flow	One-piece-flow	One-piece-flow
Möglicher Varianzgrad der Produkte	Individualisierung	Individualisierung	Konfiguration	Konfiguration	Individualisierung	Individualisierung
Anzahl Produktvarianten	hoch	hoch	mittel	mittel	hoch	hoch

Aufbauorganisation
 Ablauforganisation
 Markt

Abbildung 2.6: Abgrenzung der cyber-physischen Matrixproduktion von klassischen Organisationsformen aus [PI22]

Die Gestaltung und Umsetzung von CPMPS wird aktuell in verschiedenen Studien und Projekten diskutiert [Ke21; PI22]. CPMPS vereinen die Bestrebungen zur Einführung von flexiblen und rekonfigurierbaren Produktionssystemen (FMS und RMS). Bereits in den 90er Jahren begegnete Volvo in Uddevalla mit dem Konzept einer Boxenfertigung den steigenden Flexibilitätsanforderungen des Marktes [TS97]. Mit den zu der Zeit verfügbaren Technologien konnte dieses Projekt aber nicht die notwendige Flexibilität entfalten, u.a. durch eine zu geringe Rechenleistung zur Koordination der komplexen Prozessorganisation [Wi14]. Mittlerweile sind weitere Projekte und Studien hinzugekommen, die sich mit der modularen Produktionsform als Alternative für die Automobilindustrie beschäftigen. Durch den Einsatz fahrerloser Transportfahrzeuge und die Einführung der Elektromobilität kamen in den vergangenen Jahren weitere Anforderungen an die Produktionsplanung und -steuerung in industriellen Produktionssystemen hinzu [Bo16].

Die Veränderungsfähigkeiten von Matrixproduktionssystemen können aus verschiedenen Perspektiven betrachtet werden. Sie werden durch Veränderungen der Qualifikation der Mitarbeiter sowie Veränderungen in den Stationen, Einrichtungen und Produkten bedingt. Dabei sollten sowohl die Produkt- als auch die Produktionsebenen berücksichtigt werden [Wi09]. Die Veränderungsfähigkeiten, die auf den verschiedenen Systemebenen der CPMPS zu finden sind, werden nachfolgend von der Ressourcenebene bis zur Matrixproduktionssystemebene vorgestellt.

Die **Umrüstbarkeit** kann als operative Fähigkeit definiert werden, wonach eine Maschine oder ein Arbeitsplatz jederzeit reaktiv mit minimalem Aufwand und in kürzester Zeit zwischen verschiedenen Fertigungsaufgaben umgeschaltet werden kann. Diese Umschaltung kann sowohl manuell als auch automatisch durchgeführt werden [Wi09]. In diesem Zusammenhang wird auch auf die Definition von Lorenzer verwiesen, der eine Werkzeugmaschine als umrüstbar betrachtet, wenn sie sich jederzeit und mit minimalem Aufwand durch den Austausch oder die Veränderung der Komponenten, wie Werkzeuge oder Spannvorrichtungen, an die Ausführung definierter Arbeitsoperationen innerhalb eines festgelegten Rahmens anpassen lässt, ohne in die Steuerungskonfiguration einzugreifen [Lo11].

Rekonfigurierbarkeit ermöglicht eine Veränderung der Struktur mit minimalem Aufwand, bei gleichbleibender operativer Aufgabenerfüllung [KS10]. Dies bezieht sich nicht nur auf die Möglichkeit, Kapazitäten hinzuzufügen oder zu entfernen, sondern auch auf den Wechsel zwischen bestimmten Teilefamilien und Baugruppen [Wi97; Wi07; Wi09]. Um dies umzusetzen, sind autonome und standardisierte Funktionsmodule erforderlich, wie sie beispielsweise durch Plug & Produce-Module dargestellt werden [Ny08b]. Rekonfiguration kann jedoch auch logische Komponenten wie „Umlenkung“, „Neuplanung“ und „Neutermminierung“ einschließen, die in Bezug auf die Steuerungstechnik eine wichtige Rolle spielen [EIM05]. Die Zeit, die für die Berechnung einer neuen Konfiguration benötigt wird, hängt von der Größe und Anpassungsfähigkeit des Produktionssystems ab [VGL15].

Flexibilität im industriellen Kontext ist ein weit gefasster und weit interpretierter Begriff. Veränderungen innerhalb flexibler Strukturen werden durch ein vordefiniertes Maßnahmenbündel und festgelegte Handlungskorridore definiert [Wi07]. Ein System kann als flexibel bezeichnet werden, wenn es sich an veränderte Gegebenheiten innerhalb eines geplanten Umfangs anpassen lässt [We00]. Das bedeutet auch, dass Bedingungen, die in der bisherigen Planung des Produktionssystems nicht berücksichtigt wurden, nicht abgedeckt werden können. Zudem ist Flexibilität kurzfristig immer mit zusätzlichen Kosten verbunden. Es können verschiedene Arten von Flexibilität nach Das und El Maraghy definiert werden [Da96; EIM05]. Für CPMPs können die Ressourcen-, Produkt-, Prozess- und Routing- bzw. Sequenzflexibilität nach [Da96] berücksichtigt werden. Die Umrüstbarkeit und Rekonfigurierbarkeit sind mitunter Teil der Flexibilität. So kann z. B. die Ressourcenflexibilität die Umrüstbarkeit miteinschließen und die Produktflexibilität durch eine auftragsindividuelle Rekonfigurierbarkeit erreicht werden [Wi14].

Im Unterschied zur Flexibilität bestehen bei der **Wandlungsfähigkeit** keine vordefinierten Grenzen. Wandlungsfähigkeit beschreibt daher die Anpassungsfähigkeit eines Systems über vorgesehene Systemgrenzen hinaus [ZMV05]. Anpassungen in wandlungsfähigen Systemen sollen relativ geringe Kosten verursachen und dem Unternehmen eine schnelle Adaption an sich ändernde Umweltbedingungen ermöglichen [SAV12].

Die Zuordnung der Veränderungsfähigkeiten als operative, taktische oder strategische Dimension ist nicht eindeutig. Wiendahl z. B. ordnet Flexibilität und Wandlungsfähigkeit als taktische

Fähigkeiten ein, wobei sich die Flexibilität auf einzelne Produktionsbereiche und die Wandlungsfähigkeit auf die gesamte Fabrikstruktur bezieht [Wi14]. Andere Untersuchungen ordnen die Veränderungsfähigkeiten Wandlungsfähigkeit und Rekonfigurierbarkeit taktisch ein und Flexibilität und Umrüstbarkeit operativ [PI22].

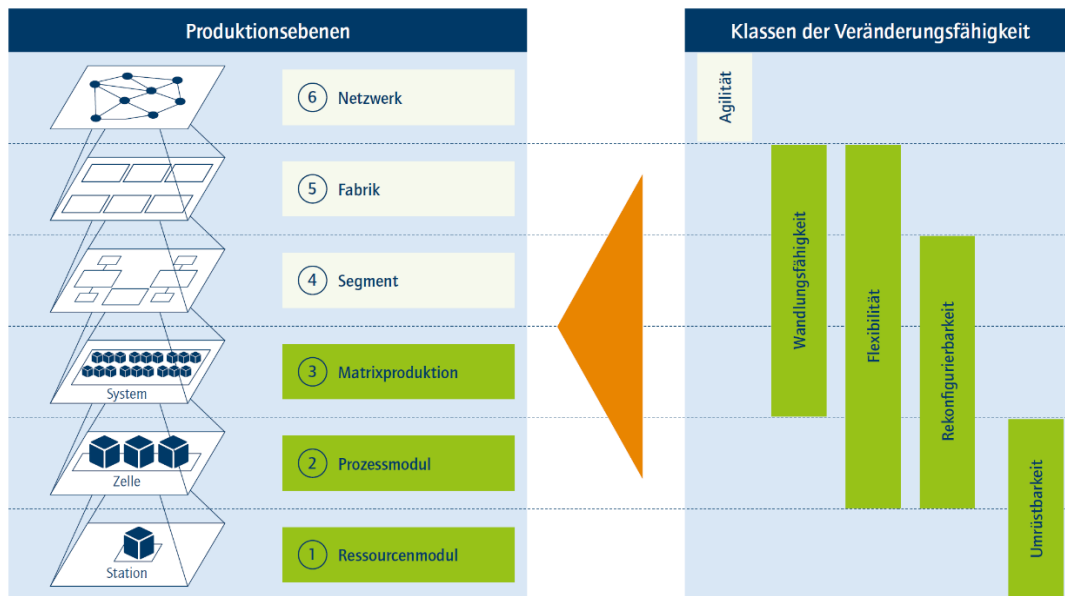


Abbildung 2.7: CPMPs mit Klassen der Veränderungsfähigkeit unter Berücksichtigung des Betrachtungsraumes der Forschungsarbeit [PI22]

In dieser Arbeit werden die operativ-steuernden Veränderungsfähigkeiten fokussiert, wie die nach [PI22] modifizierte Abbildung 2.7 darstellt. Dies sind die Flexibilität und Umrüstbarkeit, sowie die Rekonfigurierbarkeit in Bezug auf die Neuplanung bzw. Neuterminierung. Somit wird für CPMPs eine Steuerungsarchitektur gesucht, die die operativen Veränderungsfähigkeiten Flexibilität, Umrüstbarkeit und Rekonfigurierbarkeit berücksichtigt. Im nächsten Abschnitt werden die Grundlagen und Ansätze für Steuerungsarchitekturen von CPPS mit den für CPMPs benötigten Veränderungsfähigkeiten untersucht.

2.3 Steuerungsarchitekturen veränderungsfähiger CPPS

Der Forschungsbedarf im Bereich der CPMPs besteht, wie in Abschnitt 1.1 eingeleitet, in der autonomen Optimierung und Rekonfiguration [PI22]. In diesem Abschnitt werden die Grundlagen und Ansätze von Steuerungsarchitekturen vorgestellt, die eine autonome Rekonfiguration bis auf die Ebene der Control Devices (Level1-Ebene) ermöglichen können. Die Möglichkeiten zur autonomen Optimierung und Rekonfiguration veränderungsfähiger CPPS werden gemäß der am Ende von Abschnitt 2.2 getroffenen Abgrenzung bezüglich der operativ-steuernden Veränderungsfähigkeiten Rekonfigurierbarkeit und Umrüstbarkeit analysiert. Wegen der besonderen Berücksichtigung der Autonomie in dieser Arbeit werden im Abschnitt 2.3.1 die agentenbasierten Steu-

erungsarchitekturen vertieft. Grundlegend wird zunächst ein allgemeiner Überblick zu Steuerungssystemen und -architekturen verteilter Systeme im Kontext veränderungsfähiger CPPS gegeben, bevor MAS und holonische Steuerungsarchitekturen detaillierter betrachtet werden.

Für Architekturen im Systemkontext existieren viele unterschiedliche Begriffe, die mitunter redundant verwendet werden. Die bekanntesten Begriffe Referenzarchitektur (Meta-Level), Systemarchitektur (Model Level) und Systemstruktur (System Level) wurden von [Ka23] in Beziehung gesetzt und mit weiteren Begriffen ergänzt, wie in Abbildung 2.8 auf der linken Seite dargestellt.

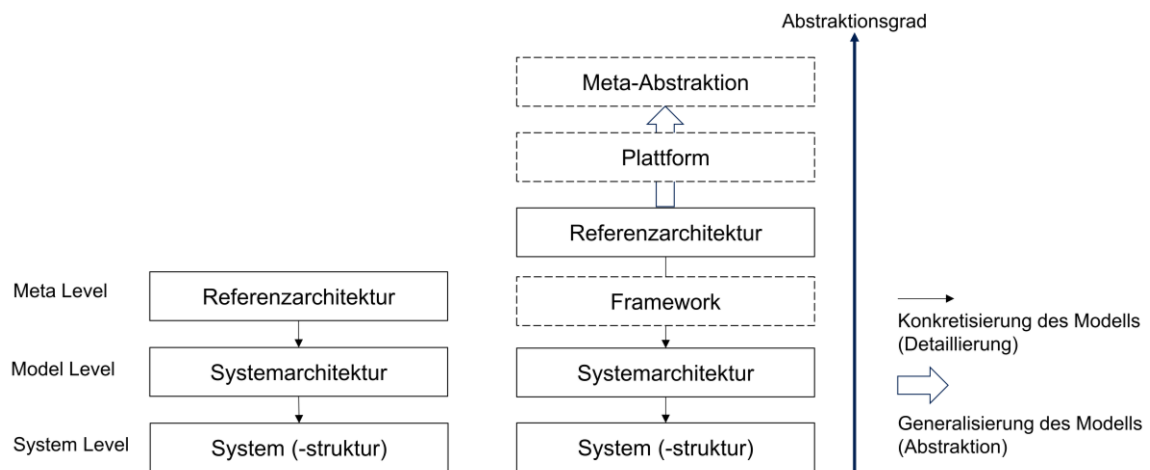


Abbildung 2.8: Einordnung der Begriffe zu Architekturen [Ka23]

Es folgt eine Definition der verschiedenen Begriffe im Kontext von Steuerungsarchitekturen. Dabei werden entsprechend einer Einteilung von Kaiser et al. die verschiedenen Ausprägungen System (-struktur), Systemarchitektur, Framework, Referenzarchitektur und Plattform in der Reihenfolge ihres zunehmenden Abstraktionsgrades, wie in Abbildung 2.8 nach [Ka23] dargestellt, beschrieben.

Ein **System** ist ein digitales Fertigungssystem, das physische und digitale Komponenten (wie z. B. Maschinen, Sensoren, Steuerungen und Software) integriert [Ka23]. Es realisiert die in der Systemarchitektur definierten Strukturen, Beziehungen und Interaktionen unter Verwendung adäquater Technologien und möglicher Systemimplementierungen. Ein System kann auch als Systemstruktur beschrieben werden mit dem Fokus auf der Verwendung kompatibler Komponenten [Ko11].

Systemarchitekturen formalisieren die Struktur und Funktionalität eines Steuerungssystems durch eine abstrakte Beschreibung seiner Assets und ihrer Interaktionen und Wechselwirkungen [Ka23]. Mithilfe standardisierter Modellierungstechniken werden die Elemente und ihre interoperablen Informations- und Kommunikationsbeziehungen sowie der (Daten-)Verwaltung beschrieben.

Frameworks sind Referenzarchitekturen mit vorgeschriebenen Komponenten oder Interaktionsmustern [Ka23]. Sie besitzen strenge Design- und Implementierungsregeln für spezifische Anwendungen. Frameworks können für ad-hoc Lösungen eingesetzt werden.

Referenzarchitekturen bieten Unterstützung für die strukturierte Entwicklung und die Gestaltung von Systemarchitekturen durch geeignete Templates und standardisierte Terminologien mit Designrichtlinien, Entwurfsmustern und Standards [Ka23]. Sie umfassen die Beziehung zwischen Steuerelementen (hierarchisch oder heterarchisch) und die Art der Systemzerlegung (serviceorientiert oder objektorientiert). Sie beschreiben dabei keine Prinzipien, Konzepte oder Technologien.

Digitale **Plattformen** kombinieren Designrichtlinien mit einem geeigneten digitalen Ökosystem [Ka23]. Die Anbieter und Anwender können serviceorientiert Dienstleistungen austauschen und (Prozess-)Wissen teilen. Die Plattformen nutzen geeignete Technologien für die Bereitstellung und Vermittlung von Anwendungen und unterstützen Anwender bei der Integration neuer Technologien. Plattformen umfassen neben den operativen auch die geschäftlichen und technischen Aspekte.

Metaabstraktionen sind organisatorische Ansätze und Ontologien, die Kernabstraktionen für Referenzarchitekturen bereitstellen [Ka23]. Die Metaabstraktionsebene enthält die wichtigsten Interoperabilitätsmerkmale einer Referenzarchitektur. Ontologien definieren Konzepte und Eigenschaften von Referenzarchitekturen und die Beziehungen zwischen ihnen. Metaabstraktionen beschreiben eine Reihe von Kernmerkmalen des digitalen Fertigungssystems, wie zum Beispiel die Serviceorientierung oder einen holonischen Entwurfsansatz. Referenzarchitekturen kombinieren eine Teilmenge dieser Merkmale.

Um die effiziente Nutzung der verfügbaren Daten zur Koordination der RAMI4.0-Architecturebenen der Geschäftsprozesse und Funktionen bis zur Steuerung zu ermöglichen, ist es von entscheidender Bedeutung, ganzheitliche Unternehmensinformationssysteme mit vernetzten, interoperablen, flexiblen und reaktiven Steuerungsarchitekturen zu kombinieren [Vo09; DCT21]. In dieser Hinsicht kann auf Vorarbeiten, die einen ähnlichen gestalterischen Ansatz haben, aufgebaut werden. Unter dem Begriff der CPMPS werden CPPS mit den in Abschnitt 2.2 vorgestellten Veränderungsfähigkeiten generalisiert als veränderungsfähige CPPS betrachtet. In der Forschung werden Konzepte zur alternativen Gestaltung und Steuerung von Produktionssystemen bereits vor Industrie 4.0 betrachtet. Dazu gehören besonders die flexiblen und rekonfigurierbaren Fertigungssysteme (FMS und RMS) [Ko99; EIM05], die holonischen Fertigungssysteme (HMS) [vBV98], bionische Fertigungssysteme [Ue92] und erweiterbare Montagesysteme [OBF06]. Kombiniert werden diese vor allem mit den Multiagentensystemen (MAS) [MVK06] und den serviceorientierten Architekturen (SoA) [Co14], welche durch die Industrie 4.0-Initiative zur Gestaltung von Produktionssystemen mit verteilten intelligenten Systemen wieder aufgegriffen wurden [Vo20].

Der Ansatz von Industrie 4.0 ist die Verteilung von Steuerungsentscheidungen auf viele miteinander vernetzte autonom agierende Entitäten (I4.0 Komponenten), die durch standardisierte

Schnittstellen (I4.0 Semantik) und dem Konzept der Verwaltungsschale (VWS) kommunizieren können. Wie in Abbildung 2.9 dargestellt, werden die Komponenten nach [Vo23] für eine Industrie 4.0-fähige Produktion hierarchisch gegliedert.

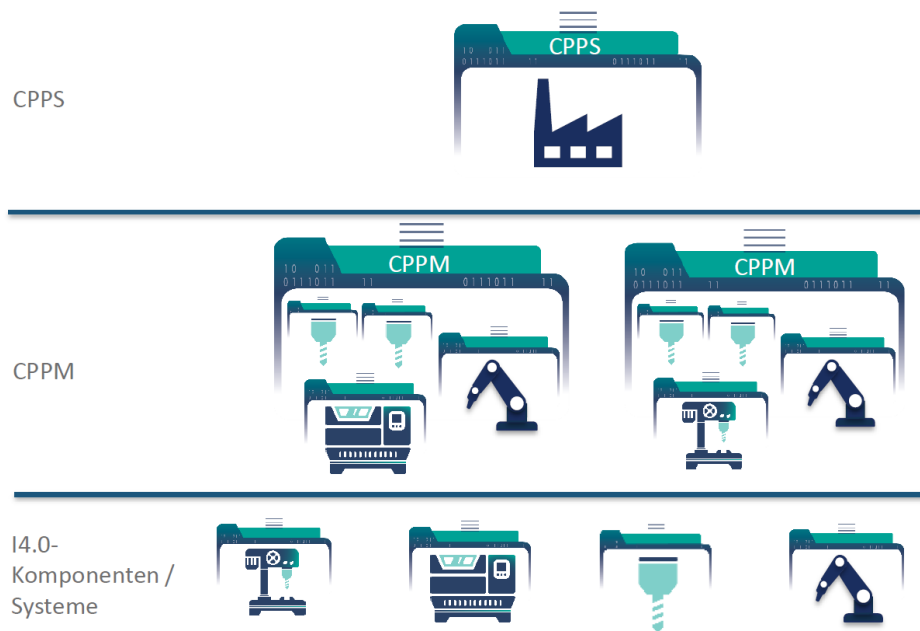


Abbildung 2.9: Hierarchische Gliederung einer Industrie 4.0 fähigen Produktion in CPPS, CPPM und I4.0 Komponenten nach [Vo23] in Anlehnung an [KHM18]

Im Folgenden werden die Begriffe Assets, I4.0-Komponente, Verwaltungsschale (VWS), I4.0-System sowie die Cyber-Physischen Systeme (CPS), die Cyber-Physischen Produktionsmodule (CPPM) und die Cyber-Physischen Produktionssysteme (CPPS) definiert:

Cyber-Physische Systeme (CPS) sind definiert als eingebettete Systeme mit physischen Komponenten und einer integrierten Recheneinheit sowie eigener Logik. Das CPS kann über eine Systemschnittstelle in einem globalen Netzwerk kommunizieren oder benutzt werden [Br10; LS15; Co17].

Cyber-Physische Produktionsmodule (CPPM) sind ein Zusammenschluss unterschiedlicher Komponenten, einer Steuerungseinheit, sowie der cyber-physischen Präsentation des Moduls. Die technischen Details werden dabei fähigkeitsbasiert abstrahiert und die Interaktion erfolgt über cyber-physische Schnittstellen und Algorithmen [KHM18; RB18; RH18].

Cyber-Physische Produktionssysteme (CPPS) beschreiben die Vernetzung von CPS, CPPM, menschlichen Ressourcen, Subsystemen und Produkten über cyber-physische Schnittstellen. Dabei soll wissensbasiert durch Überwachung während der Laufzeit eine kontinuierliche Verbesserung des Systems ermöglicht werden [Mo14; RB18; RH18].

Die Kombination der **I4.0-Konzepte** und **CPS** ermöglicht die Vernetzung der verteilten Systeme zu Wertschöpfungsnetzwerken. Die verteilten Systeme in diesen Wertschöpfungsnetzwerken können hierarchisch gekapselt werden. Diese Kapselung wird durch die DIN SPEC91345 beschrieben, wonach ein I4.0-System zur I4.0-Komponente eines übergeordneten Systems werden kann.

Die **I4.0 Komponente** und eine entsprechende **I4.0-konforme Semantik** wird mittels des Konzeptes der Verwaltungsschale zur zuverlässigen Identifikation und Kommunikation definiert. Das Zusammenspiel aus Verwaltungsschale, I4.0-Komponente und I4.0-Sprache ermöglicht die Vernetzung verschiedener Systeme. Die I4.0-Komponenten können zu I4.0-Systemen gekapselt werden.

Die **Verwaltungsschale (VWS)** repräsentiert die Eigenschaften einer I4.0-Komponente oder eines I4.0-Systems digital. Die Implementierung erfolgt dabei technologieneutral unter Berücksichtigung standardisierter Informationen der gekapselten Assets über Teilmodelle. Die Verwaltungsschale ist in drei Typen unterschiedlicher Autonomie definiert:

- **Passive VWS:** Die Beschreibung der Assets anhand standardisierter Teilmodelle (TM) sind in der VWS als Datei hinterlegt (XML oder JSON-Format) und besitzen keine Autonomie.
- **Reaktive VWS:** Die Informationen der standardisierten TM der Assets werden mittels Application Interface (API) abgerufen und ermöglichen eine teilautonome Ausführung.
- **Proaktive VWS:** Ein Interaktionsmanager innerhalb der VWS organisiert die horizontale Integration und ermöglicht ein zielgerichtetes Verhalten durch autonome Entscheidungs- und Optimierungsalgorithmen.

Um Produktionssysteme als autonom optimierende anpassungsfähige Netzwerke zu befähigen, bedarf es einer integrierten Betrachtung von Steuerungstechnik (OT) und Informationstechnik (IT) [Vo09]. Für die breite Anwendung der Konzepte RAMI4.0 und der VWS ist weiterer Standardisierungsaufwand und eine spezifischere Funktionsdefinition und Abgrenzung notwendig [RZL21].

Zur Steuerung verteilter Cyber-Physischer Systeme in dynamischen Produktionssystemen wird durch die I4.0-Initiative, wie bereits erwähnt, vor allem der Architekturstil der serviceorientierten Architekturen wieder aufgegriffen und weiterentwickelt [Vo20]. Durch die Einführung von Mikroprozessoren und der Entwicklung von verteilten intelligenten mechatronischen Systemen und CPS begann die Abkehr von der traditionellen starr hierarchischen ISA95 Automatisierungspyramide, die lange Zeit als die vorherrschende Steuerungsstruktur galt [Vo09]. Die Automatisierungspyramide in Abbildung 2.10 links beschreibt grundsätzlich alle Automatisierungsfunktionen eines Systems und umfasst sämtliche Geräte von der Feldebene über die Steuerungsebene (SPS) und die Prozessleitebene (SCADA) bis hin zur Betriebsleitebene (MES) und schließlich zur Unternehmensebene (ERP). Diese Ebenen sind klar voneinander abgegrenzt und die Gerätetypen sind

jeder Ebene fest zugeordnet. Die Datenmengen nehmen mit steigender Hierarchiestufe ab, von hochfrequenten Daten in der Feldebene bis zu hoch aggregierten, qualitativ hochwertigen Daten für Unternehmensanalysen auf der obersten Entscheidungsebene. Die Kommunikation erfolgt über individuell definierte Schnittstellen zwischen den jeweils angrenzenden Ebenen. Intelligente Feldbuskomponenten und die Möglichkeit der dezentralen Automatisierung durch den Einsatz von kleinen und leistungsstarken Controllern haben dazu geführt, dass die tatsächliche Automatisierungsstruktur in Produktionsanlagen nicht mehr zwangsläufig der klassischen Pyramide entspricht. Stattdessen kann sie alternativ wie in Abbildung 2.10 rechts dargestellt, als Automatisierungs-Diabolo beschrieben werden [Vo09]. Ein systemweites Informationsmodell bildet die Grundlage für die Funktionen und Geschäftsprozesse und dient zur standardisierten Verknüpfung mit der Prozessebene.

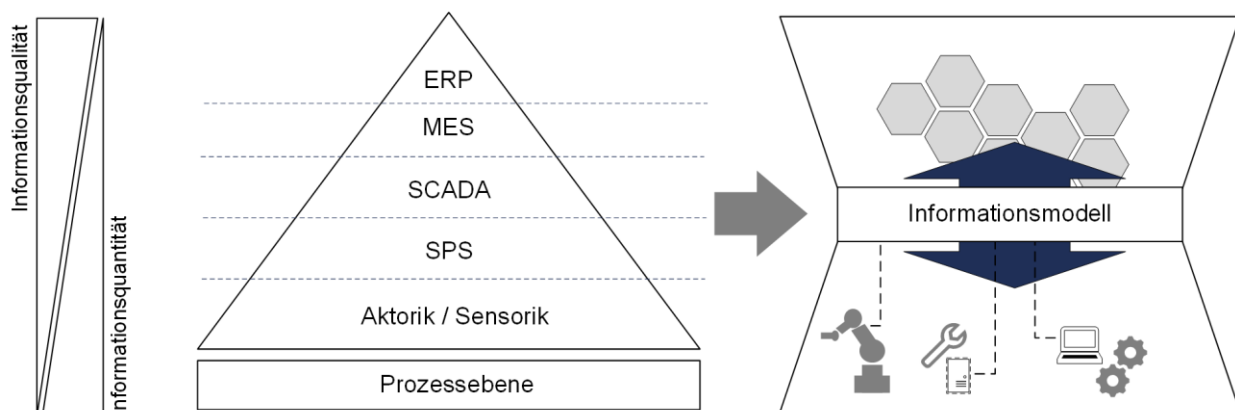


Abbildung 2.10: Wandlung der Automatisierungspyramide nach [Vo09; FWF20]

Aus IT-Perspektive basiert das Konzept der Serviceorientierten Architekturen (SoA) auf der Idee, verteilte Systeme in einem Netzwerk zu entwickeln [Er05]. Dieses Konzept wurde ursprünglich für Geschäftssysteme und den elektronischen Geschäftsverkehr entwickelt [Co15]. Verschiedene funktionale Aspekte in der Fertigung haben sich zu Dienstleistungen entwickelt, wie zum Beispiel IT-as-a-Service (ITaaS) oder Infrastructure-as-a-Service (IaaS). Diese Entwicklung wird voraussichtlich in verschiedenen Bereichen fortgesetzt, angefangen von Platform-as-a-Service (PaaS), die maßgeblich von [Ma06] geprägt wurden, bis hin zu Manufacturing-as-a-Service (MaaS) [Ku19]. Das RAMI 4.0 mit DIN SPEC 91345 bildet die Grundlage für die Entwicklung von serviceorientierten Architekturen im Bereich der Fertigung. Services und serviceorientierte Architekturen werden nach [Ma06] wie folgt definiert:

Ein **Service** ist ein Mechanismus, der den Zugriff auf eine oder mehrere Fähigkeiten ermöglicht, wobei der Zugriff über eine vorgeschriebene Schnittstelle erfolgt und in Übereinstimmung mit den in der Dienstbeschreibung angegebenen Einschränkungen und Richtlinien ausgeübt wird.

Die **serviceorientierte Architektur (SoA)** ist ein Paradigma für die Organisation und Nutzung verteilter Fähigkeiten, die unter der Kontrolle verschiedener Besitzerdomänen stehen können.

SoA zielen darauf ab, komplexe monolithische Unternehmenssoftwarearchitekturen durch flexible, verteilte Anwendungslandschaften zu ersetzen. Dabei werden Funktionalitäten oder Steuerungslogiken in Services gekapselt und wiederverwendet. Diese Services sind lose miteinander gekoppelt und orchestrieren Geschäftsprozesse, hauptsächlich auf den Funktions- und Business-Layern von RAMI4.0. Erste Beispiele für die Umsetzung von SoA in Produktionssystemen sind die Architekturen SOCRADES oder IMC AESOP [Co14; Co15]. Ebenso können die SoA IMPROVE, PERFoRM und BaSys genannt werden, die sich individuell durch unterschiedliche spezifische Funktionalitäten auszeichnen und in einer Kombination zu interoperablen, echtzeitfähigen flexiblen und rekonfigurierbaren SoA ergänzt werden konnten [Tr19]. Eine Weiterentwicklung von Services sind Microservices. Nach dem Konzept der Microservice-Architektur wird eine Anwendung als eine Ansammlung kleiner Services entwickelt, von denen jeder als eigenständiger Prozess läuft und über leichtgewichtige Mechanismen, häufig basierend auf HTTP-APIs, mit anderen Systemkomponenten kommuniziert. Diese Services sind zur Unterstützung von Geschäftsfunktionen organisiert und können unabhängig voneinander bereitgestellt werden. Sie können in verschiedenen Programmiersprachen implementiert werden und auf verschiedene Datenspeichermethoden zugreifen [Fo]. Durch den Einsatz standardisierter Implementierungstechnologien können RAMI 4.0 basierte SoA Architekturen über die Architekturlayer interoperabel umgesetzt werden [YH18]. In Abbildung 2.11 ist eine beispielhafte RAMI4.0-konforme SoA für ein Prozesssteuerungssystem dargestellt mit Verweis auf verschiedene Implementierungstechnologien.

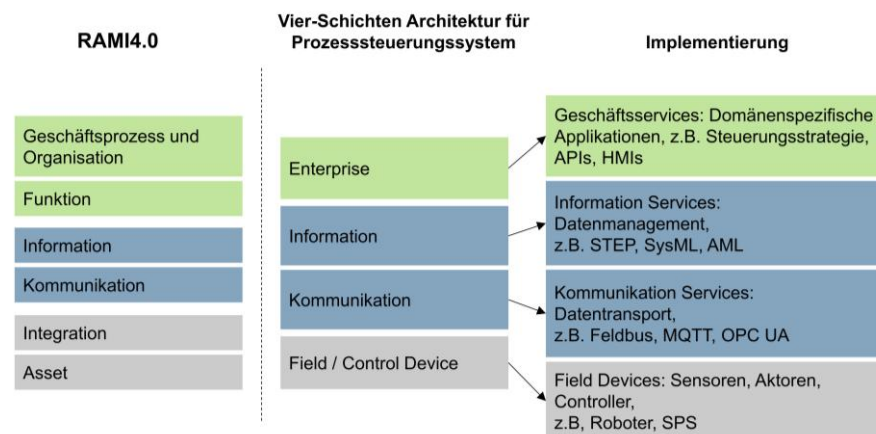


Abbildung 2.11: 4-Schichten SoA zur Prozesssteuerung mit möglichen Implementierungstechnologien nach [YH18]

Die Serviceorientierung in I4.0-Architekturen kann durch kooperative Aspekte und autonome Entscheidungsfindung mithilfe von MAS und agentenbasierten Systemen ergänzt werden [RBC08; Le16]. Für die Umsetzung von agentenbasierten Architekturen sind serviceorientierte Architekturkonzepte und Webservices wesentliche Aspekte [Le16]. Serviceorientierte Agentensysteme (SOMAS) unterscheiden sich von traditionellen MAS durch die serviceorientierte Gestaltung der Agenten [Me09]. Die Kapselung von Prozessschritten zur Steuerung verteilter CPS kann durch den fähigkeitsbasierten Ansatz unterstützt werden. Die generische skillbasierte Beschreibung der Fähigkeiten einer Ressource ermöglicht das Prinzip des „Plug-and-Produce“ [Pf15].

Durch die interoperable semantische Beschreibung der Hardwarehierarchie können die Produktionsentitäten nach der Initialisierung fähigkeitsbasiert identifiziert werden und als Skills ausgeführt werden [Zi19]. Diese beiden Prinzipien ermöglichen eine flexible Erweiterung von vorhandenen Systemen durch das Hinzufügen von Teilsystemen oder Komponenten. Die Summe der Fähigkeiten der Teilsysteme ergeben die Fähigkeiten des darüberliegenden Systems. Gleichzeitig dient das Identifizieren von generischen Fähigkeiten und der Abgleich mit den Anforderungen eines Auftrags der Komplexitätsreduktion. Darüber hinaus können so Ebenen berücksichtigt werden [Br18]. Das rechte Schema in Abbildung 2.12 zeigt das fähigkeitsbasierte Produkt (P) - Prozess (P') - Ressource (R) - Konzept. Das Produkt benötigt eine bestimmte Fähigkeit, die von der Ressource bereitgestellt wird. Für das Matching von Ressourcen mit geeigneten Skills für die flexible Fertigung von Produkten mit speziellen Features werden entsprechende Informationsmodelle benötigt, die zu einer serviceorientierten Produktion beitragen [He21; Kö23]. Wenn dieser Abgleich erfolgreich ist, kann der entsprechende Prozess am Produkt durchgeführt werden [Va19b].

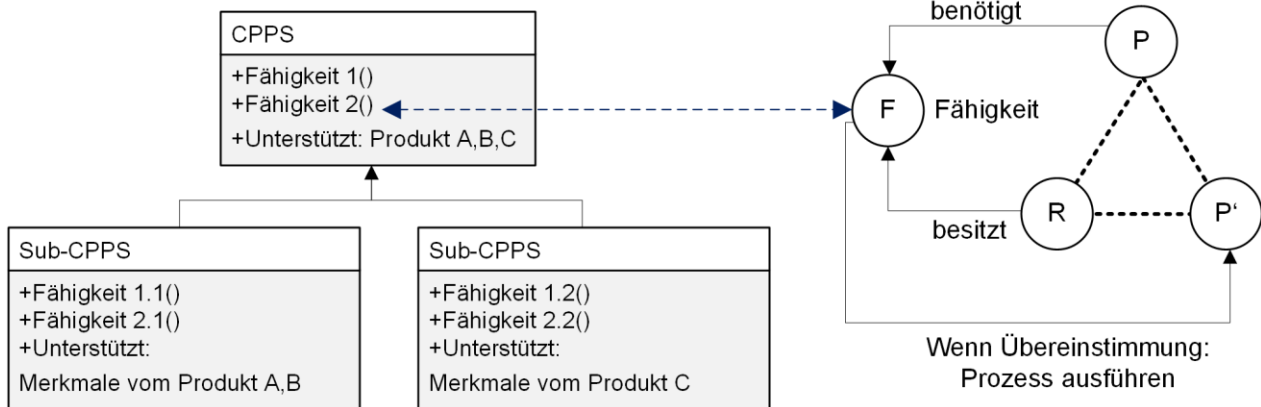


Abbildung 2.12: Fähigkeitsbasierte Architektur (links) und Einordnung in das PPR-Modell (rechts) nach [Br18; Va19a]

Die Plattform Industrie 4.0 definiert ein Geschäftsmodell, das auf dem Prinzip der „Shared Production“ basiert und sich auf die flexible Konfiguration der Produktion konzentriert, nicht nur innerhalb einer Fabrik, sondern auch auf die Optimierung über Standorte und Unternehmen hinweg. Im CSS (Capability, Skill, Service) - Modell werden die bereits weiter oben definierten Services mit den Fähigkeiten (Capabilities) und Fertigkeiten (Skills) verknüpft [Ba20a; Kö23]:

Fähigkeit (Capability): Eine Fähigkeit spezifiziert Produktionsfunktionen mit Auswirkungen in der physischen oder virtuellen Welt, die von Produktionsressourcen bereitgestellt werden.

Fertigkeiten (Skill): Eine Fertigkeit besitzt Parameter, die verschiedene Aspekte der Produktion beeinflussen können, zum Beispiel den Produktionsprozess und die Ausführung der Steuerung. Jede Fertigkeit sollte über mindestens eine Schnittstelle zugänglich sein.

Produkt, Prozess, Ressource: Das Produkt steht für alle Materialien, Komponenten und Zwischenerzeugnisse, die in das endgültige Produkt einfließen. Der Prozess steht für alle Aktivitäten, die zur Fertigung des Produktes beitragen und die Produktion steuern. Dabei wird der gesamte Produktionsprozess in Teilprozesse und schließlich in einzelne Operationen unterteilt. Ressourcen stehen für alle benötigten Ressourcen, um die Produktion erfolgreich durchzuführen, einschließlich der Fabrik und der einzelnen Komponenten, Betriebsmittel, Systeme und menschlichen Arbeitskräfte.

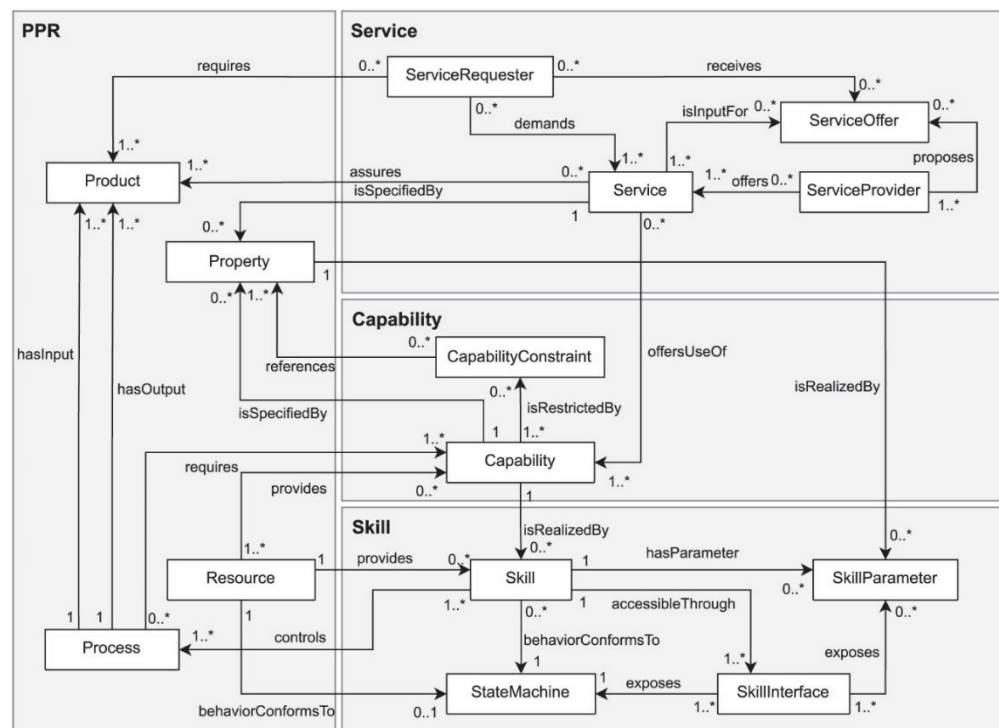


Abbildung 2.13: PPR CSS Informationsmodell nach [Kö23]

Im Rahmen dieser Arbeit sollen Möglichkeiten zur Umsetzung autonom gesteuerter CPMPs als modular hierarchische Systeme untersucht werden. Der Fokus liegt dabei auf der Entscheidungsfindung und -unterstützung und setzt auf einer fähigkeitsbasierten Informationsmodellierung auf. Um fähigkeitsbasierte Ansätze umsetzen zu können, existieren bereits Entwurfsmuster u.a. für die Ressourcenallokation und für den Ressourcenzugriff, aber auch für andere grundlegende Funktionen in CPPS, welche Ansätze zur standardisierten Umsetzung liefern können [Cr19]. Im folgenden Abschnitt werden die MAS und holonischen Steuerungsarchitekturen vertieft, die die autonome Steuerung in komplexen Systemen wie den veränderungsfähigen CPPS ermöglichen können.

2.3.1 Multi-Agenten-Systeme und holonische Steuerungsarchitekturen

Serviceorientierte Prinzipien können in MAS integriert werden, um die Funktionalitäten im Hinblick auf Interoperabilität und vertikale und horizontale Integration zu verbessern. Obwohl beide Paradigmen die verteilte Steuerung eines Produktionssystems ermöglichen, weisen MAS und SoA

einige wichtige Unterschiede auf, insbesondere hinsichtlich der Rechenanforderungen und Interoperabilität, wie in Tabelle 4.1 dargestellt [RBC08]. Es existieren verschiedene Möglichkeiten zur Kombination der beiden Konzepte, sowohl über Gateways als auch Middleware Ansätze [Me09; Co15].

Tabelle 2.2: Vergleich MAS vs. SoA nach [RBC08]

Multi-Agent Systems	Service-oriented Architectures
Etablierte Methoden zur Beschreibung des Verhaltens eines Agenten	Fokus auf der Detaillierung der öffentlich zugänglichen Schnittstelle statt der Beschreibung von Ausführungsdetails
Agenten repräsentieren soziale Fähigkeiten, die durch interne oder externe Einflüsse reguliert werden.	Sozialität ist keine definierte Eigenschaft von SoA
Implementierungen sind für eine Nutzung im lokalen Netzwerk (LAN) optimiert, aber auch die Verwendung über das Internet ist möglich	Implementierung durch webbasierte Technologien für Ausführung im Internet
Rekonfiguration durch reaktives Verhalten auf Veränderungen in der Umgebung	Rekonfiguration erfordert Neuprogrammierung
Die Interoperabilität ist maßgeblich von der Einhaltung FIPA-ähnlicher Standards abhängig	Die Interoperabilität wird durch die Verwendung von Webtechnologien sichergestellt
Hoher Rechenaufwand erforderlich	Hohe Leistungsfähigkeit ohne Einschränkung der Interoperabilität

Mit dem Einsatz von Agenten kann die Flexibilität, Rekonfigurierbarkeit, Zuverlässigkeit und Anpassungsfähigkeit eines Systems gesteigert werden [BJW04; Cr19]. Gemäß den Definitionen in [WJ95; RN03; MVK06; Ka20] wird ein Agent und ein Multiagentensystem wie folgt charakterisiert:

Ein **Agent** ist ein autonom handelndes Softwaresystem, das darauf ausgerichtet ist, ein vorgegebenes Ziel zu erreichen. Sollte ein Agent nicht in der Lage sein, ein Problem eigenständig zu lösen, kann er mit anderen Agenten kommunizieren, um kooperativ eine Aufgabe zu bewältigen.

Ein **Multiagentensystem** (MAS) bildet ein Netzwerk von verteilt vorhandenen Agenten, die fähig sind, miteinander zu kommunizieren und gemeinsam Probleme zu lösen.

Nach Fischer können Agenten in menschliche Agenten, Hardware- und Softwareagenten gegliedert werden, die für ein Zielobjekt oder -subjekt definierte Aufgaben übernehmen [FSS03]. Mit der Integration von CPS im Kontext von Industrie 4.0 rückt der Einsatz von Agenten sogar bis auf

die Feldebene in den Fokus [Vo18]. CPS besitzen Aktoren und Sensoren und können Informationen und Befehle über einen internen Controller ausführen. Agenten können somit auch Teil von CPS sein, entweder als integrierte „embedded“ Agenten in den Prozessgeräten oder in Form von separaten „standalone“ Komponenten, die keine vollständige Kontrolle übernehmen, sondern das Verhalten der Komponenten mittel- bis langfristig beeinflussen [Le16]. Agenten werden über alle Hierarchieebenen und im Bereich der Verhaltenssimulation und des Managements von großen CPS und Cyberphysischen System of Systems (CPSoS) eingesetzt [Le16; Vo18; Cr19]. Dies ermöglicht die Kombination einer dezentralen Steuerung der Prozesskomponenten mit einer ausgelagerten rechenintensiven koordinierenden Planung, z. B. über Cloud Computing [Le16; DCT21]. Nachfolgende Abbildung 2.14 zeigt die grundsätzlichen Funktionsbereiche eines Agenten.

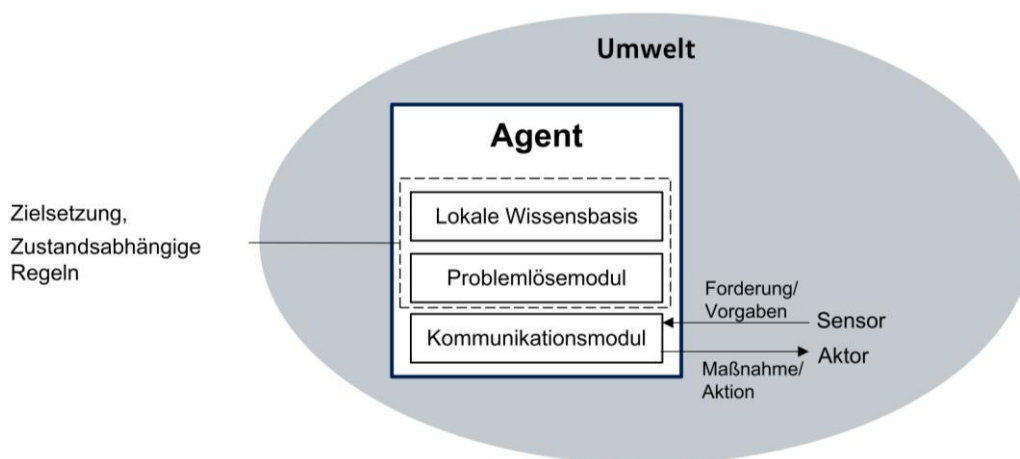


Abbildung 2.14: Agentenfunktionen und -interaktion mit seiner Umgebung in Anlehnung an [Br15] nach [Hi00] und [RN03]; MVK06; Ka20]

Das Wesen von Softwareagenten unterscheidet sich von herkömmlichen Computerprogrammen oder Softwaresystemen durch ihre zugeordneten spezifischen Eigenschaften. Besitzt ein Agent eine steuernde Funktion einer Produktionsentität, so befähigt er diese mit den Eigenschaften eines Agenten. Nachfolgende Beschreibungen der Eigenschaften basieren auf [WJ95; RN03; MVK06; CH12; KPK18].

- **Autonomie:** Agenten können ohne direkten Einfluss eines Menschen in ihrer Umgebung existieren und interagieren und haben dabei Kontrolle über ihre eigenen Aktionen und den internen Zustand.
- **Soziabilität:** Agenten können sowohl mit anderen Agenten als auch Menschen Informationen austauschen.
- **Reaktivität:** Agenten nehmen ihre Umgebung wahr, welche dabei eine physische Umgebung, eine Schnittstelle oder eine Sammlung weiterer Agenten, das Internet oder eine Kombination all dieser sein kann.

- **Proaktivität:** Agenten agieren nicht nur in Reaktion auf die Umgebung, sondern können eigene Aktionen in zielgerichtetem Sinne ableiten. Ein Ziel ist dabei nicht zwangsläufig immer die Maximierung eines Zielwerts.

Aufgrund dieser grundlegenden Eigenschaften können Agenten anpassungsfähig umgesetzt werden. Eine Weiterführung dieser Eigenschaften sind Lernfähigkeit, Echtzeitfähigkeit, Mobilität sowie Ehrlichkeit, Vertrauenswürdigkeit und Zuverlässigkeit [RN03]. Nicht jeder Agent verfügt allerdings über die gleichen Eigenschaften. Agenten können zu Multi-Agenten-Systemen kombiniert werden und kooperieren zur Erreichung ihrer individuellen Ziele [Le09]. Die Umgebung von Agenten lässt sich dabei grundsätzlich anhand der folgenden Eigenschaften beschreiben [RN03; SL09]:

Deterministisch vs. stochastisch: In deterministischen Umgebungen hat eine Aktion bei gleichem Ausgangszustand immer dieselbe Wirkung. In stochastischen Umgebungen führt dieselbe Aktion mit bestimmten Wahrscheinlichkeiten zu unterschiedlichen Zuständen.

Statisch vs. dynamisch: Dies betrifft die Kontrolle der Umgebung durch den Agenten. In statischen Umgebungen werden Veränderungen durch Aktionen des Agenten ausgelöst, während in dynamischen Umgebungen Veränderungen außerhalb des Einflusses der Agenten stattfinden.

Diskret vs. kontinuierlich: Dies bezieht sich auf die Art der Aktionen der Agenten. Diskrete Aktionen lassen sich klar voneinander abgrenzen, während kontinuierliche Aktionen fließend ineinander übergehen.

Episodisch vs. sequenziell: In episodischen Umgebungen führt ein Agent eine Reihe von einander unabhängiger Entscheidungen aus. In sequenziellen Umgebungen beeinflusst eine Entscheidung auch zukünftige Bewertungen und Entscheidungen des Agenten.

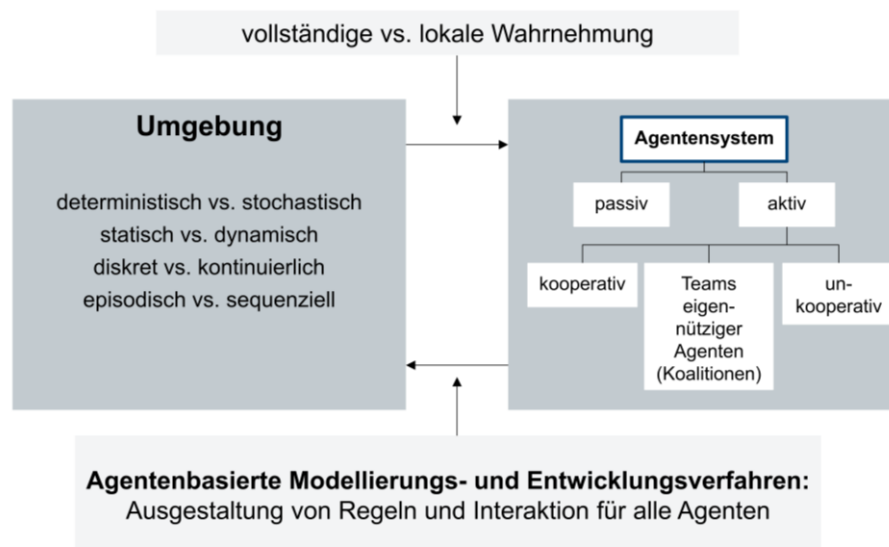


Abbildung 2.15: Charakterisierung von MAS und der Umgebung nach [RN03] und [We01] in Anlehnung an die Darstellung in [Wa20]

Der Umfang der Wahrnehmung der Umgebung kann entweder der Umgebung oder dem Agenten zugeordnet werden, wobei zwischen partieller bzw. lokaler und vollständiger bzw. globaler Wahrnehmung unterschieden wird. In MAS lassen sich Agenten in passive und aktive Kategorien einteilen. Passive Agenten beobachten ihre Umgebung, ohne diese zu verändern oder zu beeinflussen. Im Gegensatz dazu sind die meisten Agenten aktiv und modifizieren die Umgebung durch ihre Handlungen. Grundsätzlich unterscheidet man zwischen kooperativen und unkooperativen bzw. kompetitiven Agenten. Durch die Möglichkeit Agenten zu Koalitionen zusammenzuschließen, können diese sich gegenseitig unterstützen. Zur Strukturierung der Entscheidungsfindung von MAS kann auf agentenbasierte Entwicklungs- und Modellierungsverfahren zurückgegriffen werden [We01; FR15; LF15, WR17].

Agentenbasierte Modelle eignen sich gut für Szenarien in denen z.B. Interaktionen zwischen den Akteuren von Bedeutung sind, in denen Heuristiken vorherrschen, in denen die Heterogenität der Akteure relevant ist, in denen eine Vielzahl granularer Daten verfügbar ist und in denen herkömmliche analytische Methoden nicht ausreichen [Le16]. Die anwendungsfallspezifische Entwicklung und Modellierung für den Einsatz in Steuerungssystemen kann mithilfe von Referenzarchitekturen oder speziellen Entwicklungsmethoden erfolgen. Ebenso können Entwurfsmuster verwendet werden, die bei der Umsetzung von Agentensystemen wiederverwendet werden können [Lu17; Cr19]. Wassermann und Fay haben vier Grundsätze für die Koordination von Agentensystemen festgelegt [WF17]. So ist es erstens notwendig, dass die Aussagen der Agenten wahrheitsgemäß sind. So können andere Agenten darauf vertrauen, dass zugesagte Aktionen tatsächlich ausgeführt werden. Zweitens müssen die von den Agenten eingegangenen Verpflichtungen eingehalten werden. Diese Verpflichtungen müssen für eine bestimmte Zeitspanne gültig sein, damit andere Agenten ihre Entscheidungen auf dieser Grundlage treffen können. Eine Neubewertung der Entscheidungen sollte daher für einen gewissen Zeitraum ausgeschlossen sein. Drittens sollte die Aktivität der Agenten begrenzt werden, beispielsweise durch Festlegung einer

Obergrenze, um zu verhindern, dass die Anfragen übermäßig ehrgeiziger Agenten das System überlasten. Schließlich sollte als viertes Prinzip vermieden werden, dass Agenten zu langsam oder überhaupt nicht reagieren, da dies zum Stillstand des Systems führen könnte [WF17].

Anwendung findet das MAS-Konzept in der Produktion als Holonic Manufacturing System (HMS). Das HMS bietet Flexibilität auf struktureller Ebene für die Zuweisung von Aufgaben und die Zusammenhänge zwischen Entitäten über Holone. Die Skalierbarkeit von HCAs ermöglicht es, verschiedene Funktionen mit unterschiedlichen Zeitskalen in derselben Steuerungsarchitektur zu integrieren, von der Prozessgerätesteuerung bis zur Lieferkettensynchronisation [DCT21]. Dabei kann die Holonic Component Based Architecture (HCBA) als die erste Referenzarchitektur genannt werden, die eine solche Steuerung umsetzt [CM00]. Ein Beispiel für die Anwendung integrierter Produktionsplanung und Prozesssteuerung wurde Anfang der 2000er Jahre bereits in der Automobilindustrie im Projekt 2000+ umgesetzt [MB03; Bu]. Ein anderes Beispiel für den Einsatz einer agentenbasierten Steuerung in der industriellen Fertigung ist das Projekt FABMAS in der Halbleiterfertigung, das zu einem ähnlichen Zeitpunkt durchgeführt wurde [MSZ03].

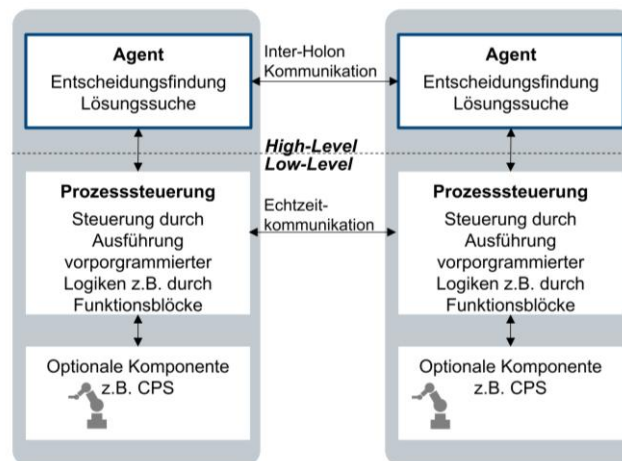


Abbildung 2.16: Agenten übernehmen die Nicht-Echtzeit Kommunikation zwischen Holonen in Anlehnung an [MM05; Ka20]

Holone können Agenten in sich kapseln und die Verantwortung zur Aufgabenbearbeitung delegieren. Ein Beispiel dafür ist das Delegate MAS Entwurfspattern von [Ma19a]. Der hierarchische Aufbau von Holonen wird auch als Holarchie bezeichnet und kann das hybride Steuerungsparadigma umsetzen, das flexible Informationsbeziehungen ermöglicht [FSS03]. Diese lose Hierarchie unterscheidet sich von herkömmlichen Hierarchien insofern, als dass sie verschiedenen Hierarchien angehören, temporäre Hierarchien bilden und unabhängig von anderen Holonen operieren können [VvB15]. Diese Eigenschaften ermöglichen es einem System selbstorganisiert zwischen unterschiedlichen Steuerungsparadigmen wechseln zu können [LCR05; Mö06]. Während Agenten individuelle Ziele verfolgen, ist der aus der Kooperation der Agenten entstehende Effekt der Emergenz nicht vordefinierbar. Ebenfalls ist die Anwendung und Portabilität der MAS nicht ausreichend erforscht, sodass sie häufig in geschlossenen Softwareumgebungen ausgeführt werden.

Es ist daher zu beachten, dass die Skalierbarkeit hinsichtlich der Problemkomplexität nicht völlig deterministisch und abschätzbar ist [KPK18].

Die Topologie von MAS spielt eine entscheidende Rolle bei der Gewährleistung der Interoperabilität zwischen den Agenten und externen Systemen [WR17]. Agenten kommunizieren mithilfe von Nachrichten, die vielfältige Formen und Dimensionen annehmen können [Vo20]. Dabei sind bestimmte Standards für die Nachrichten von großer Bedeutung, darunter die Sprachen Knowledge Query and Manipulation Language (KQML) und die Agent Communication Language (ACL), die von der Foundation for Intelligent Physical Agents (FIPA) standardisiert wurde. Diese Protokolle gehen über die gängige M2M Kommunikation hinaus und ermöglichen komplexe Kommunikation zwischen Agenten [Le16]. Sie sind unabhängig von der Ontologie der Nachricht sowie dem zugrunde liegenden Transportmechanismus. FIPA-ACL ist das am weitesten verbreitete Protokoll für agentspezifische Kommunikation [Ga18]. Es unterstützt klassische Request/Response-Muster sowie das Publish/Subscribe-Muster und enthält die essentiellen Funktionen für das Akzeptieren/Ablehnen von Vorschlägen, was die Verhandlungsfähigkeiten der Agenten ermöglicht, sowie das Fehler- und Informations-Performativ [FI02]. Die Integration in die Ressourcenebene kann mithilfe gängiger M2M Protokolle gateway-, broker-, middleware- oder serverbasiert erfolgen.

MAS sind durch die Umsetzung der hierarchischen, heterarchischen und hybriden Steuerungsparadigmen geeignet, Produktionssysteme mit verteilten intelligenten Systemen autonom zu steuern. Agenten können individuell modelliert werden und zusammenarbeiten, um die Ziele des Gesamtsystems zu erreichen und adaptiv auf Veränderungen in ihrer Umgebung zu reagieren [Zi08; WR17]. Die Kombination von HMS und SoA in einer Steuerungsarchitektur bietet den Vorteil einer autonomen Zuweisung von Aufgaben zu Ressourcen kombiniert mit der serviceorientierten Kapselung von (Prozess-) Funktionen für die flexible Ausführung auf verschiedenen Ressourcen. Die Skalierbarkeit von holonischen Steuerungsarchitekturen ermöglicht es, Funktionen mit unterschiedlichen Zeithorizonten in einer Steuerungsarchitektur zu unterstützen, sodass sowohl die Prozessgerätesteuerung als auch die Synchronisation von Lieferketten integriert betrachtet werden können [DCT21]. Im folgenden Abschnitt werden Architekturen vorgestellt, die als Ansätze holonischer Steuerungsarchitekturen für eine autonome Steuerung veränderungsfähiger CPPS verwendet werden können.

2.3.2 Ansätze holonischer Steuerungsarchitekturen für rekonfigurierbare CPPS

In CPMPs wird die autonome Anpassung des Systems unter Berücksichtigung der system- und ebenenspezifischen Veränderungsfähigkeiten benötigt. In der Literatur existieren Ansätze zur agentenbasierten Steuerung in hierarchisch organisierten CPPS mit unterschiedlichem Grad der Veränderungsfähigkeit. Zur Übersicht über den Einsatz von Agentensystemen in der Produktion kann vor allem auf die Recherchen in [Le16; DCT21; Ka23] verwiesen werden. Der Einsatz holonischer Steuerungsarchitekturen wurde als geeigneter Architekturstil zur Befähigung autonomer CPPS mit verteilter rekonfigurierbarer Ressourcensteuerung identifiziert [Cr19; DCT21]. In diesem Abschnitt werden agentenbasierte und holonische Referenzarchitekturen für die Steuerung

hierarchischer CPPS vorgestellt, die bereits vielfach in Forschung und Industrie eingesetzt wurden [LF15]. Dabei werden die Ansätze holonischer Architekturen von auf dem Gebiet führenden Forschern um McFarlane, van Brussel, Valckenaers, Leitão, Ribeiro, Lüder, Vogel-Heuser, Cardin, Trentesaux und weiteren vorgestellt.

Die **HCBA** von McFarlane und Chirn besteht aus zwei Arten von Komponenten im Produktionssystem: der Ressourcenkomponente und der Produktkomponente mit zweigeteilter Struktur [CM00; MM05]. Die Ressourcenkomponente besteht aus einem physischen Teil und einem virtuellen Steuerungsteil, durch den die Ressourcen sich selbst organisieren und ihre internen Abläufe optimieren. Die Produktkomponente besteht aus einem physischen Teil und einem informationsbezogenen Teil. Ihr physischer Teil kann Materialien, Teile, Paletten usw. repräsentieren. Der informationsbezogene Teil verwaltet das Produktionsprogramm, einschließlich der Routensteuerung, der Prozesskontrolle, der Entscheidungsfindung und der Produktinformation. Der informationsbezogene Teil besteht aus Agenten mit spezifischen Rollen. Jede Produktkomponente bezieht sich auf einen Produktkoordinator, der Work-In-Process Agenten erstellt. Beide überwachen die Fertigstellung von Aufträgen auf unterschiedlichen Ebenen. Der Produktkoordinator gewährleistet die Produktionsüberwachung einer Charge, während WIP-Agenten für die Produktionsüberwachung eines einzelnen Teils verantwortlich sind. Als Ergebnis verhandeln WIP-Agenten mit den Ressourcen, um die Bearbeitung des Teils auf dem Shopfloor zu definieren. Diese Verhandlungen erfolgen innerhalb eines vom Produktkoordinator festgelegten Ziels.

Van Brussel und Valckenaers haben die **PROSA**-Architektur entwickelt, die sowohl hierarchisches als auch heterarchisches Steuerungsverhalten implementiert [vBV98]. Diese Architektur ist eine weit verbreitete HMS-Referenzarchitektur, die durch vier Holon-Typen repräsentiert wird: Produkt-Holon, Ressourcen-Holon, Order-Holon und Staff-Holon [Ve06]. Das Staff-Holon ist optional und unterstützt andere Holone bedarfsorientiert, indem es spezialisiertes Know-how bereitstellt. RHs bestehen aus einem physischen und einem informationsverarbeitenden Teil und bieten Features und Fähigkeiten anderen Holonen an. PHs stellen Wissen über Produkte und Prozesse bereit, einschließlich Arbeitsplänen mit Qualitätsvorschriften für die korrekte Ausführung der Herstellungsschritte [Mö06; BG08]. In der ersten Holarchie kooperieren Produkt-, Ressourcen- und Order-Holone zur Ressourcenallokation, während in der zweiten Holarchie PHs und RHs zur Prozesssteuerung zusammenarbeiten [BG08]. Die PROSA-Architektur wird für die Integration von Planungs- und Produktionssteuerungssystemen verwendet [VV06]. Die ARTI-Architektur von Valckenaers ist eine Weiterentwicklung und realisiert eine Digital Twin-Schicht. Diese Form der Virtualisierung der physischen Steuerungskomponente ist eine alternative Form gegenüber den bis dahin existierenden Architekturen, die entweder zentralisiert oder dezentralisiert virtualisiert wurden [Va19a].

Leitão et al. haben die **ADACOR**-Architektur entwickelt, die sich durch die Möglichkeit der selbstorganisierten Anpassung des Produktionssystems bei Störungen auszeichnet und wechselt dazu zwischen den Steuerungsparadigmen [LR03; LCR05]. Sie orientiert sich an PROSA und besteht aus

vier Arten von Holonen: Produkt-, Order-, Operational- und Supervisor Holons [LR02]. Die Produkte sind repräsentiert durch Produktholone, die produkt- und prozessbezogenes Wissen enthalten und die Planung des Fertigungsprozesses übernehmen. Diese Fertigungsaufträge werden mit Task Holons abgearbeitet. Sie übernehmen die Steuerung und Überwachung der Produktionsschritte und das dynamische Speichern der zugehörigen Informationen. Die Operational Holons repräsentieren physische Produktionsressourcen und steuern die Ausführung und Optimierung der Produktionspläne [LR03]. Das SH überwacht Aktivitäten der übrigen Holone und integriert die Planungs- und Steuerungsmechanismen. Identifiziert ein OH eine Störung, kann heterarchisch eine Alternative gewählt werden [LCR05]. Die ADACOR Architektur hat eine breite Anwendung. Aufbauend auf der ADACOR wurde die ADACOR² Architektur durch Barbosa, Leitão um Trentesaux et al. entwickelt und erweitert das Steuerungsverhalten um die Möglichkeit Übergangszustände einzunehmen, die Abweichungen evaluiert und mittels dynamischer Online Konfiguration anpasst [Ba13]. Die Ablaufplanung erfolgt z. B. mit einem genetischen Algorithmus [Ba15a].

Die **GRACE-Architektur** wurde von Leitão et al. gemäß den Prinzipien der ADACOR-Architektur entwickelt, wobei ihr Hauptziel die Integration von Prozess- und Qualitätskontrollfunktionen ist [Le15]. GRACE umfasst fünf Arten von Agenten. Product Type Agents, die Produkt- und Prozesswissen sowie historische Daten nutzen, um neue Aufträge zu verfolgen und Anpassungen vorzunehmen. Product Agents verwalten Kundenaufträge und interagieren mit weiteren Agenten, um Auftragsfolgen zu ändern und produktionsprozessbezogene Daten zu sammeln. Resource Agents repräsentieren physische Ressourcen und führen Produktionsprozessschritte aus, einschließlich Anpassungen und Optimierungen. Qualitätsagenten sind für die Qualitätssicherung verantwortlich und steuern QS-Messungen unter Anpassung von Algorithmen und Parametern basierend auf Umgebungsinformationen und historischen Daten, die zentral systemweit verfügbar sind [Ca11]. Unabhängige Meta-Agenten organisieren das System hierarchisch zur Überwachung und Entscheidungsfindung. Die QAs der GRACE Architektur sind mit Ressourcenagenten anderer Referenzarchitekturen vergleichbar.

Die **PABADIS'PROMISE** von Lüder, Vogel-Heuser et al. basiert auf dem Projekt PABADIS, das eine Steuerungsarchitektur mit Agenten für MES bis Feldebenesteuerung beschreibt [Fe06; PLK05]. Ressourcen werden durch Resource Agents repräsentiert, die über die Bearbeitung offener Produktionsaufträge verhandeln [Pe14]. Das Herzstück der PABADIS'PROMISE-Architektur sind der Order Agent, der Ability Broker und der Information Collector. Ergänzt werden die internen Agenten mit Resource Agent Supervisors zur MES und ERP-Systemen und Order Agent Supervisors zur Verbindung in ein Product Data Repository [Fe06]. Eine Hierarchie wird implizit durch eine flexible Auftragsstruktur definiert. Die Architektur ermöglicht eine verteilte Entscheidungsfindung und integriert Feedback aus der ERP-Ebene über periodisches und ereignisbasiertes Reporting [BLT10]. Die Möglichkeit dynamischer Anpassungen durch Rekonfiguration des Fertigungssystems ermöglicht die Nutzung der Anlagenflexibilität [VGL15]. Vogel Heuser, Lüder, Leitao haben agentenbasierte Entwurfsmuster für die Ressourcenallokation und den Ressourcenzugriff auf Control Device und Field Device Ebene entwickelt und im Modell agent@PLC umgesetzt [VLL15].

Diese Architektur umfasst einen CPPS-Fabrikanten, der für die Koordination eines Produktionssystems in einem CPPS-Netzwerk verantwortlich ist. In dieser Struktur sind verschiedene Komponenten enthalten, darunter ein Systemagent mit Kenntnissen über die Werkstruktur, ein Kommunikationsagent, der die Kommunikation zwischen den Einheiten des CPPS koordiniert, ein Prozessagent, der den Produktionsprozess und die erforderlichen Schritte steuert, ein Whiteboard, das Aufgaben und ihren Status verfolgt, sowie ein Automatisierungssteuerungsagent, der den physischen Teil der Anlage repräsentiert und die Ausführung der notwendigen Schritte des Prozessagenten übernimmt.

Die **SoHMS**-Architektur um Cardin basiert hauptsächlich auf den Prinzipien und Konzepten von PROSA und kombiniert diese mit Interaktionskonzepten aus HCBA und ADACOR [Ga16]. Auch wenn die Architektur den Konzepten ähnlich ist, wurde ihr Verhalten so angepasst, dass die Services zur Planung und Steuerung flexibel eingesetzt und wiederverwendet werden können in dem verteilten System. In einem SoHMS wird jeder Arbeitsschritt, der an Produkten durchgeführt werden soll, durch Services beschrieben, die von einer oder mehreren verfügbaren Ressourcen erbracht werden können. Die Autoren adressieren mit ihrer Architektur die Herausforderung der Wiederholgenauigkeit der Prozessausführung auf verschiedenen Ressourcen. Ziel ist die flexible Bereitstellung und ressourcenspezifische Ausführung auftragsindividueller Fertigungsservices unter Echtzeitbedingungen. Die Holone der SoHMS-Architektur werden in Holone zur Entscheidungsfindung und zur Prozessausführung unterschieden. Durch einen Service Interface Layer werden Services für Ressourcen Holone verfügbar gemacht [Ga16; DCT21].

Trentesaux et al. haben die **SURFER**-Architektur, die nach Le Mortellec ursprünglich für Transportsysteme entwickelt wurde, für den Einsatz in CPPS mit hohem Datenaufkommen weiterentwickelt [LeM13; Bo19]. Es handelt sich dabei um die Adaption eines Steuerungssystems für die Wartung und Überwachung von Zügen für die Transportsteuerung. Dazu werden rekursive Diagnosefunktionen für die Systembereiche auf verschiedenen Hierarchieebenen eingesetzt. Jedes System besteht aus einem Steuerungsteil und einem gesteuerten Teil. Der Steuerungs-Part führt einen Algorithmus aus, und der kontrollierte Part führt ein erwartetes Verhalten aus. Die unterste Ebene der holonischen Struktur besteht in der Regel aus physischen Elementen wie z. B. Sensoren und Aktoren. Durch Aggregation der Holone ist es möglich, eine Architektur zu erstellen, in der Daten auf verschiedenen semantischen Ebenen verarbeitet werden und bei der jedes Holon die Daten der unteren Ebene verstehen kann. Mit diesem Mechanismus wird die Datenmenge so weit wie möglich verdichtet und die Daten werden nur dann gesendet, wenn sie relevant sind [Bo19].

Heragu entwickelte eine hybride holonische Architektur als Alternative zu den bereits genannten agentenbasierten Referenzarchitekturen mit Fokus auf der integrierten Produktionsplanung und -steuerung von Produktionssystemen [He02; He03a]. Das agentenbasierte Framework strukturiert die Interaktion hierarchisch über mehrere Ebenen. Der Fokus liegt auf der Betrachtung einer integrierten Produktionsplanung mit Ablauf- und Prozesssteuerung unter Berücksichtigung indi-

vidueller Auftrags-, Prozess- und Ressourceninformationen. Entscheidungen basieren auf selbst-organisierten Interaktionen zwischen Materialfluss- und Fertigungssystementitäten. Die Architektur kombiniert Optimierungsfunktionen mit lern- und wissensbasierten Ansätzen für die Verfolgung global optimierter Pläne und lokal reaktiver Anpassungsmöglichkeiten. Die Architektur kann bei Bedarf zwischen hierarchischem und heterarchischem Interaktionsverhalten wechseln. Agenten höherer Hierarchie besitzen Koordinationsfunktionen für die Agenten der Prozessausführungsebene. Die Architektur berücksichtigt die Fertigungshierarchie und ermöglicht eine globale Optimierung des Produktions- und Materialflusses. Das System besitzt Entscheidungsstrukturen für die hierarchische Aufgabenzuweisung und -ausführung. Holone auf niedrigerer Ebene können autonom Verhandlungen innerhalb der von den Agenten höherer Ebene gesetzten Rahmenbedingungen treffen. Ein Agent niedrigerer Hierarchie muss die Freigabe eines höherstufigen Agenten einholen, um vom bereitgestellten Zeitplan abweichen zu dürfen. Die Informationen über Prozesspläne und Zustandsinformationen werden über eine Datenbank zentral bereitgestellt. Die Berücksichtigung mehrerer Produktebenen findet nicht statt.

Neben den holonischen Steuerungsarchitekturen konnten generische Agententypen für die Verwendung in hierarchischen CPPS identifiziert werden. Durch eine Analyse verschiedener agentenbasierter Architekturen von Vogel-Heuser et al. wurden vier Unteragententypen definiert, die in hierarchischen Steuerungssystemen universell eingesetzt werden können [Vo18; Cr19]. Diese sind Ressourcenagent (RA), Prozessagent (PA), Agentenmanagementsystem (AMS) und Kommunikationsagent (CA) und werden in die Funktionsgruppen Ressourcenzugriff (RA), Wissensbasis (KB), Koordinationsprozess (CP) und Kommunikationsschnittstelle (CI) klassifiziert. Die Unteragenten verwenden die Wissensbasis, um formale Methoden für ihre Implementierung abzuleiten. Der Ressourcenzugriff beschreibt die funktionale Anforderung, Daten von physischen Ressourcen in Echtzeit zu erfassen und zu verarbeiten. Der RA deckt in der Regel die untersten 0-2 Automatisierungsstufen ab. Der Koordinationsprozess enthält Verfahren und Abgrenzungen der Unteragenten zur Verwaltung von MAS in einer höheren Hierarchie. Diese Funktionalität ist in der Regel in AMS auf den Automatisierungsstufen L1-L2 und PA auf den Automatisierungsstufen L2-L3 enthalten. Die CI ermöglicht offene Kommunikation zwischen den Automatisierungsstufen mit verschiedenen Datenformaten und unterstützt die Aufgaben von AMS und CA auf den Automatisierungsstufen L1-L3. Die Funktionalitäten der CI sind oft in allen Automatisierungsstufen vertreten. Das vorgeschlagene Muster der vier Unteragenten kann relevante MAS-Funktionen für Entwickler liefern, um ähnliche Lösungen auf dieser Grundlage zu entwerfen [Cr19].

Zusammenfassend kann festgestellt werden, dass bereits verschiedene Ansätze holonischer Steuerungsarchitekturen existieren, die mehrere Hierarchieebenen abdecken und in Wissenschaft und Industrie angewendet werden. Dies bezieht sich hauptsächlich auf die Steuerung und Überwachung. Die hierarchische Struktur der Produktionssystemhardware sowie der Aufträge und Produkte wird noch nicht explizit angewendet. Mit den existierenden holonischen Architekturen kann eine auftragsindividuelle Prozessausführung in rekonfigurierbaren Produktionssystemen mit flexibler Auftragsallokation ermöglicht werden. Das agentenbasierte Framework von

Heragu kommt der Problemstellung für die autonome Steuerung von CPMPs unter Berücksichtigung mehrerer Hierarchieebenen bereits sehr nahe. Durch den expliziten Fokus auf der integrierten Planung und Steuerung unterscheidet es sich von bisher verwendeten Referenzarchitekturen. Für die Anwendung in CPMPs wird eine spezifische Modellierung der Interaktionsgestaltung sowie des Optimierungsverhaltens mit Berücksichtigung der Flexibilitätsarten eines CPMPs benötigt. Der Einsatz von Supervisor- und Koordinationsagenten sorgt für eine globale Optimierung in holonischen Architekturen. Daher werden im nächsten Abschnitt die Grundlagen und Ansätze zur Optimierung von veränderungsfähigen CPPS mit Fokus auf flexiblen Belegungs- und Reihenfolgemöglichkeiten vorgestellt.

2.4 Produktionsplanung und -steuerung veränderungsfähiger CPPS

Mit Industrie 4.0 sollen CPPS als vertikal integrierte Gesamtsysteme befähigt werden. Dazu sollen die verschiedenen Steuerungshierarchien und Zeithorizonte integriert betrachtet werden können. Neben dem in Abschnitt 2.2 vorgestellten Bereich der Steuerungsarchitekturen wird in diesem Abschnitt der Bereich der Produktionsplanung und -steuerung (PPS) für veränderungsfähige CPPS behandelt. Während Steuerungsarchitekturen die Bereiche der Prozesssteuerung und Automatisierung fokussieren, werden in der PPS die Methoden zur Optimierung des Produktionsflusses ausgeführt. Es werden zunächst die Grundlagen der PPS in Bezug auf die Reihenfolgeorganisation in veränderungsfähigen bzw. komplexen Produktionssystemen vorgestellt, bevor explizit auf die dynamische Ablaufplanung und das erweiterte Flexible Job Shop Scheduling mit möglichen Lösungsansätzen eingegangen wird.

Innerhalb der Produktion existieren unterschiedliche Planungsarten, die im englischen als Planning, Scheduling und Dispatching bezeichnet werden. Die Planungsarten verfolgen die Aufgabe, die zur Verfügung stehenden Ressourcen so zu organisieren, dass die Produktionsziele erreicht werden. Die Planungsarten unterscheiden sich in ihrem Entscheidungshorizont. Die Zusammenhänge und Einordnung von Planning, Scheduling und Dispatching nach Wu et al. [WS20] sowie die groben Aufgabenbeschreibungen und Zeitrahmen im Rahmen der Fertigungsregelung nach Wiendahl und Mönch [Wi97; MFM13] sind in Abbildung 2.17 kombiniert dargestellt. Die Abgrenzung zwischen den Planungsarten ist fließend.

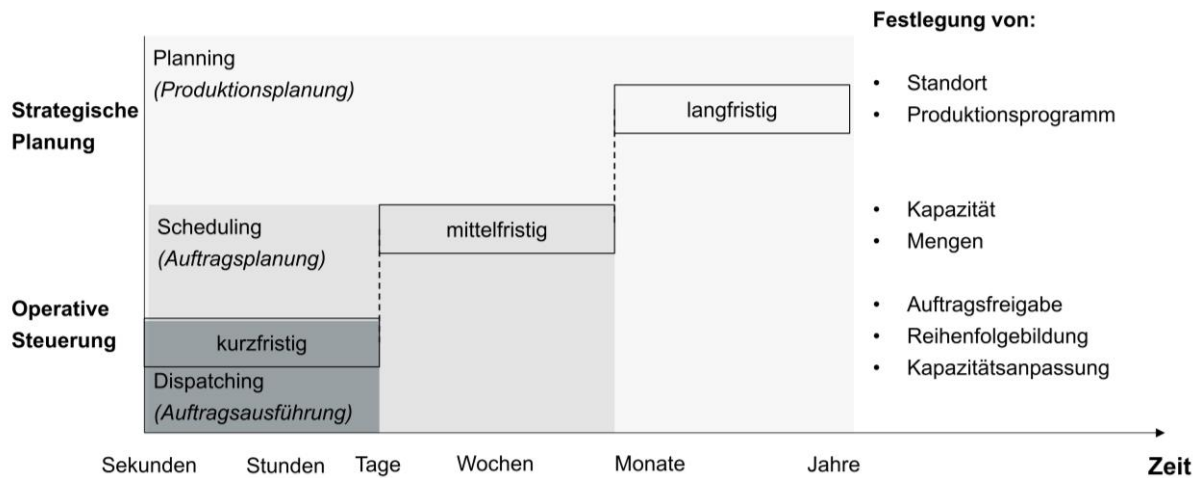


Abbildung 2.17: Planungsarten in der Produktion, eigene Darstellung in Anlehnung an [Wi97; MFM13; WS20]

Es existiert keine allgemeingültige Definition der drei zentralen Planungsbegriffe, daher werden die Planungsbegriffe auf Basis der englischen Bezeichnung von Wu et al. [WS20] definiert.

Das (Production) **Planning** beschreibt grundlegend den Prozess der Entscheidungsfindung darüber, wie die Fertigung von Produkten zur Gewinnmaximierung organisiert werden sollte [WS20]. Dazu wird ein Plan für die zukünftige Produktion definiert, bei dem die benötigten Kapazitäten bestimmt und bereitgestellt werden [Te98]. Die Festlegung des Produktmixes und der Kapazität erfolgt unter Berücksichtigung der Kundenanforderungen [HL96]. Sie bezieht sich ausschließlich auf die langfristige Planung oder den Zeitraum davor [FMW08]. Der Zeithorizont umfasst Monate bis Jahre [MFM13]. Im weiteren Verlauf der Arbeit wird diese Planungsart als **Produktionsplanung** bezeichnet.

Das (Job / Resource) **Scheduling** zielt darauf ab, die Herstellung oder Fertigung von Produkten aus Komponenten oder Rohstoffen zu planen, um die auf der Grundlage der Produktionsplanung getroffenen Entscheidungen mit begrenzten Ressourcen innerhalb eines vordefinierten Zeitraums zu terminieren. Das Scheduling definiert die Startzeiten der Auftragsbearbeitung auf den dafür infrage kommenden Maschinen [Ke00]. Im weiteren Verlauf der Arbeit wird diese Planungsart als **Auftrags- oder Ablaufplanung** bezeichnet.

Das (Job) **Dispatching** (engl. *Disposition oder Einlastung*) bedeutet Waren oder Nachrichten für einen bestimmten Zweck an einen bestimmten Ort zu senden [WS20]. Durch Dispatching wird der Auftrag aus einem Pool an verfügbaren Aufträgen zur unmittelbaren Bearbeitung ausgewählt, um die Auftragspläne umzusetzen [BPH82]. Der Zeithorizont des Dispatching wird in der Regel in Minuten, Stunden und Tagen gemessen [MW03]. Im weiteren Verlauf der Arbeit wird diese Planungsart der **Auftragsausführung oder -steuerung** zugeschrieben.

Die Begriffe werden im betriebswirtschaftlichen Kontext als Produktionsplanung und -steuerung (PPS) bezeichnet. Die Hauptaufgabe einer ganzheitlichen PPS besteht darin, die Fertigung termingerecht, unter Berücksichtigung der Kapazität und der erforderlichen Mengen auf Grundlage der erwarteten oder bekannten Kundennachfrage zu organisieren [SS12]. Dabei sind grundsätzlich die logistischen Zielgrößen zu berücksichtigen, darunter eine gleichmäßige Auslastung der Kapazitäten, Flexibilität, niedrige Lagerbestände, pünktliche Lieferungen und kurze Durchlaufzeiten. Ein ganzheitlicher Ansatz zur Optimierung des Produktionssystems ist das Aachener PPS-Modell, das sowohl die Ressourcen und Prozesse im Unternehmen als auch den Kundennutzen berücksichtigt und Zielgrößen für das System festlegt. Wie in Abbildung 2.18 dargestellt, unterteilt das Modell die Aufgaben in überbetriebliche Netzwerkaufgaben und innerbetriebliche Kern- und Querschnittsaufgaben.

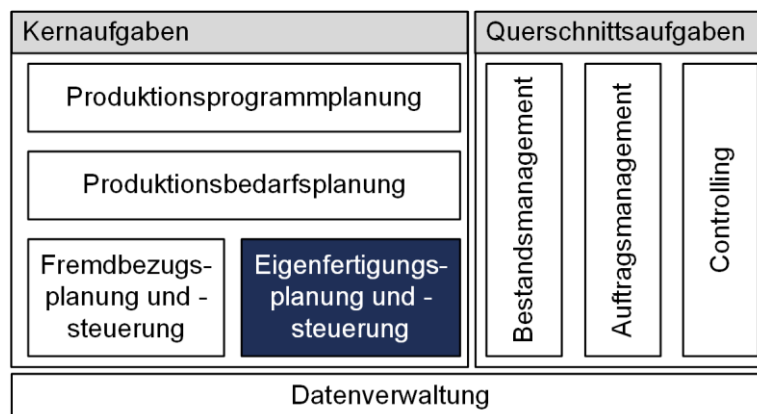


Abbildung 2.18: Kern- und Querschnittsaufgaben im Aachener PPS-Modell nach [SS12]

Die Eigenfertigungsplanung und -steuerung setzt die Fertigungsaufträge aus der Produktionsprogramm- und Produktionsbedarfsplanung um. Eine zentrale Herausforderung dabei ist die Zuweisung von Aufträgen zu den verfügbaren Ressourcen, wobei festgelegt werden muss, wann und wo jeder Auftrag bearbeitet wird [Mü70; SS12]. Je nach Produkt- und Fertigungsstruktur können diese Aufträge verschiedene Fertigungsprozesse oder Arbeitsschritte umfassen. Die Losgrößenrechnung zielt darauf ab, wirtschaftlich optimale Losgrößen für die Fertigung festzulegen, häufig basierend auf Mitarbeitererfahrung oder Losgrößenformeln, um Unwirtschaftlichkeit zu vermeiden. Das Zusammenfassen von Aufträgen kann die Durchlaufzeiten verkürzen, indem Rüstzeiten minimiert werden [Da09; SS12]. Die Arbeitsgangfeinterminierung folgt anschließend mit der Terminierung der Arbeitsschritte unter Berücksichtigung ihrer Bearbeitungs- und Übergangszeiten.

Die Ressourcenfeinplanung berücksichtigt die tatsächlich vorhandenen Kapazitäten an Material, Personal, Betriebsmitteln und Hilfsmitteln. Die Reihenfolgeplanung erfolgt durch Prioritätsregeln, Kumulationskriterien oder die dezentrale Abstimmung durch Mitarbeiterkommunikation. Das Produktionsprogramm wird mit den verfügbaren Kapazitäten abgeglichen und eine Neuplanung erfolgt bei unzureichender Verfügbarkeit. Schließlich erfolgt als letzter Schritt bei der klassischen Eigenfertigungsplanung die Auftragsfreigabe, bei der die Aufträge in das Produktionssystem eingelastet werden, entsprechend dem erstellten Ablaufplan. In komplexen Systemen wie

den CPMPs sind die letzten beiden Schritte vertauscht, sodass die Auftragsfreigabe vor der Reihenfolgeplanung erfolgt [Eh11]. Schuh begründet dies damit, dass in komplexen Systemen durch intensive Kommunikation der Teilnehmer eine Reihenfolge selbstorganisiert festgelegt werden könne, um so der Dynamik gerecht zu werden [SS14].



Abbildung 2.19: Schritte im Planungsprozess in der Linienfertigung (oben) und in komplexer Werkstattfertigung (unten) nach [SS14] in [Wa20]

Die Produktionssteuerung fungiert als Bindeglied zwischen den planenden und operativen Abteilungen in einem produzierenden Unternehmen und übernimmt die Aufgaben der Auftragsplanung und Auftragsausführung. Ihre zentrale Verantwortung besteht darin, die Vorgaben aus der Produktionsplanung bestmöglich umzusetzen, auch wenn sich Änderungen in Bezug auf Auftragsmenge oder Fertigstellungstermin ergeben oder unvorhergesehene Störungen auftreten. Infolgedessen trägt die Produktionssteuerung erheblich zur Erreichung der logistischen Zielgrößen Bestand, Durchlaufzeit, Auslastung und Termintreue bei [Da09]. Die operative Produktionssteuerung beinhaltet die folgenden operativ-steuernden Aufgaben [Lö16]:

1. **Auftragserzeugung:** Hier werden die geplanten Werte für den Zugang und Abgang von Arbeitsaufträgen sowie die Bearbeitungsreihenfolge festgelegt. Dieser Schritt ist normalerweise der Produktionsplanung zugeordnet.
2. **Auftragsfreigabe:** Die Auftragsfreigabe bestimmt den Zeitpunkt, zu dem die Aufträge von der Planung zur Umsetzung in die Fertigung übergehen. Dies hat direkten Einfluss auf den Umlaufbestand in der Produktion und erfordert die Sicherstellung der Verfügbarkeit der eingeplanten Ressourcen.
3. **Kapazitätsanpassung:** Die kurzfristige Anpassung der Kapazitäten in den Arbeitsprozessen erfolgt hier. Dies beinhaltet die Festlegung der Arbeitszeiten der Mitarbeiter und die Zuordnung zu den Arbeitsprozessen. Es werden Maßnahmen ergriffen, um auf Störungen innerhalb der flexiblen Kapazitätsgrenzen zu reagieren.
4. **Reihenfolgebildung:** Diese Aufgabe legt fest, in welcher Reihenfolge Aufträge an einem Arbeitsplatz bearbeitet werden. Dazu stehen unterschiedliche Prioritätsregeln und Steuerungsverfahren zur Verfügung.

5. Auftragsüberwachung: Dies ist entscheidend für die Verfolgung des Auftragsdurchlaufs von der Freigabe bis zur Fertigstellung. Störungen und unvorhergesehene Ereignisse sollen identifiziert und deren Einfluss minimiert werden.

Die Veränderungsfähigkeiten aus Abschnitt 2.1 können in strategisch-planerische und operativ-steuernde Veränderungsfähigkeiten unterteilt werden [Wi97; Wi09]. Die Wandlungsfähigkeit und Agilität sind dabei gestalterisch-planerisch einzuordnen, wobei die Rekonfigurierbarkeit, Flexibilität und Umrüstbarkeit operativ-steuernd mit kurzfristigerem Zeithorizont einzuordnen sind. Die Rekonfigurierbarkeit kann beiden Kategorien zugeordnet werden, wird in dieser Arbeit aber vorwiegend operativ-steuernd verstanden und bezieht sich dabei sowohl auf die physische als auch auf die logische Interpretation dieser Veränderungsfähigkeit. Für die Produktionssteuerung von CPMPS sind somit die operativen Veränderungsfähigkeiten der Flexibilität, Rekonfigurierbarkeit und Umrüstbarkeit zu berücksichtigen. Dazu werden im folgenden Kapitel die Methoden zur Reihenfolgeoptimierung in dynamischen flexiblen Systemen vorgestellt.

2.4.1 Dynamische Reihenfolgeoptimierung und erweitertes FJSP

Für die autonome Optimierung veränderungsfähiger CPPS ist die dynamische Bildung von Ablaufreihenfolgen grundlegend. Die Ablauf- bzw. Auftragsplanung in CPMPS kann der Produktionssteuerung zugeordnet werden und fungiert als Bindeglied zwischen der strategischen Produktionsplanung und der operativen Prozesssteuerung. Die Optimierung von vorranggraphbasierten Ablauf- und Prozessfolgen auf Steuerungs- und Materialflussebene für veränderungsfähige CPPS kann als integrierte Prozess- und Ablaufplanung betrachtet werden. Diese beiden Aufgaben werden in der Literatur oft separat betrachtet, obwohl die Prozessplanung, insbesondere die Wahl der Prozessschrittfolge, Auswirkungen auf die Ablaufplanung haben kann [SK99; Pe16].

Das grundlegende Optimierungsproblem bei der Ablaufplanung besteht darin, eine geeignete Reihenfolge für die Bearbeitung einer Gruppe von Aufträgen oder Jobs zu finden, bei denen verschiedene Arbeits- bzw. Prozessschritte von Ressourcen durchgeführt werden müssen [JP19]. In modularen Produktionssystemen existieren durch die Flexibilitätsarten mitunter mehrere Produktionspfade, die zur Fertigstellung eines Auftrags gewählt werden können, wie in Abbildung 2.20 dargestellt ist.

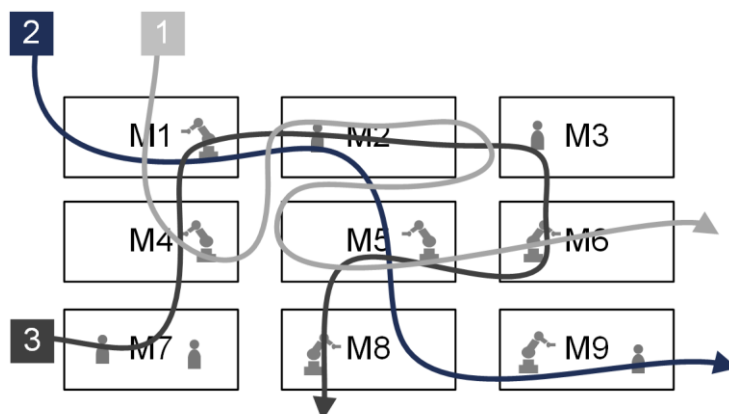


Abbildung 2.20: Modulares Produktionssystem und drei exemplarische mögliche Produktionspfade nach [TFB20]

Abbildung 2.21 zeigt eine beispielhafte Zuordnung von zwei Fertigungsaufträgen, bestehend aus je drei Arbeitsgängen auf zwei Ressourcen. Abgebildet ist hierbei die Durchführungszeit auf den jeweiligen Ressourcen. Diese unterteilen sich in die tatsächliche Bearbeitungszeit und die Rüstzeit, welche schraffiert zwischen einzelnen Arbeitsgängen markiert ist. Die Zeit zwischen den jeweiligen Arbeitsgängen eines Fertigungsauftrags ist die Zwischenzeit, welche wiederum aus Transport- und Liegezeiten vor und nach der Bearbeitung gekennzeichnet sind. Die Durchlaufzeit ergibt sich aus der Summe der Durchführungszeiten und der Zwischenzeiten für den jeweiligen Fertigungsauftrag [Ny08a].

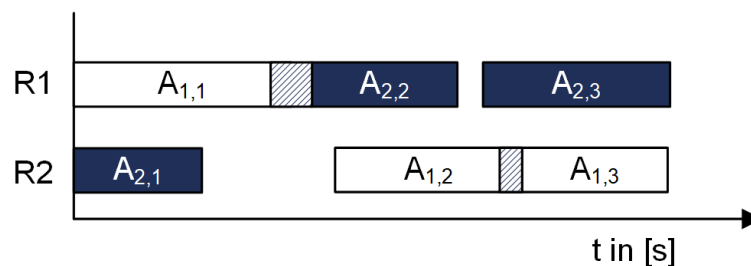


Abbildung 2.21 Maschinenbelegungsplan für zwei Aufträge mit je drei Arbeitsgängen

CPMPS als modulare Systeme mit flexiblen Auftragsfolgen werden als flexible Job Shop Probleme (FJSP) behandelt, welche durch eine hohe Anzahl an Aufträgen und eine Vielfalt an Produkten und Ressourcen gekennzeichnet sind. Das FJSP ähnelt dem Job Shop-Scheduling und ist jedoch aufgrund der vielen potenziellen Operationen in einer Fertigungssequenz noch komplexer [vB90]. Bereits das Finden des kürzesten Ablaufplans für ein JSP mit mehr als zwei Ressourcen ist ein NP-schweres Optimierungsproblem [GJS76]. In autonomen modularen Produktionssystemen muss der Ablaufplan zustandsaktuell anpassbar sein, um Unsicherheiten und Störungen zu bewältigen, wobei Störungen auf der Systemebene behandelt werden sollten, auf der sie auftreten [vB90].

Die Auftrags- oder Ablaufplanung mit Berücksichtigung von Echtzeitereignissen kann als dynamische Ablaufplanung bezeichnet werden [OP09]. Lösungsansätze dafür können in vollständig reaktiv, prädiktiv-reaktiv und robust proaktiv unterschieden werden [VHL03]. Vollständig reaktive Ablaufplanung erfolgt ohne vorher festgelegten Zeitplan. Es erfolgt eine lokale Zuweisung, und Maschinen werden erst nach Abschluss auf einer vorherigen Maschine für die Bearbeitung ausgewählt. Diese Ansätze sind für den Einsatz in Umgebungen mit geplanter Teileanlieferung (wie JIT/JIS-Anstellung) ungeeignet [OP09]. Prädiktiv-reaktive Ansätze erstellen offline einen vorhersehbaren Zeitplan und korrigieren ihn online bei Bedarf aufgrund unvorhergesehener Störungen. Dies kann zentralisiert und dezentralisiert durchgeführt werden, wodurch Berechnungszeiten verkürzt werden [VHL03]. Diese Methode eignet sich sowohl zur Einhaltung von JIS/JIT-Anforderungen als auch zur Anpassung bei Störungen [OP09]. Robust proaktive Ablaufplanung berücksichtigt vorab Unsicherheiten, beispielsweise hinsichtlich der Verfügbarkeit von Stationen während der ursprünglichen Planung, ohne dass eine Neuplanung erforderlich ist. Dies stellt einen

Worst-Case-Ansatz dar und kann bei geringeren als erwarteten Störungen auch zu einer schlechteren Performance führen [OP09].

In der Praxis gibt es verschiedene Ausprägungen von modularen Fertigungskonzepten und Optimierungsproblemen wie das Flow Shop-Problem (FSP), das Job Shop Problem (JSP) und das Open Shop-Problem (OSP). Diese Probleme berücksichtigen zusätzliche Annahmen zur Reihenfolge, in welcher die Arbeitsschritte eines Jobs bearbeitet werden können. Diese Restriktionen werden z. B. durch Vorrangbeziehungen der Arbeitsgänge, ressourcenbezogene Rüstzeiten oder Transportzeiten zwischen Arbeitsgängen definiert. Gerade die Vorrangbeziehungen sind unterschiedlich stark ausgeprägt in Abhängigkeit des Fertigungskonzeptes. Das FSP ist das restriktivste Problem und basiert auf dem Prinzip der Fließbandfertigung, bei dem angenommen wird, dass die Arbeitsschritte streng miteinander verbunden sind und Jobs in einer festen Reihenfolge die Ressourcen durchlaufen. Das JSP orientiert sich an der Werkstattfertigung und erfordert zwar eine feste Reihenfolge der Arbeitsschritte, diese kann jedoch je nach Job variieren. Das OSP hingegen legt keine Beschränkungen für die Reihenfolge der Arbeitsschritte eines Jobs fest [Pi16; GW18]. Um die Flexibilität in der realen Fertigung besser zu berücksichtigen, können die Maschinenflexibilität, Prozessflexibilität und Sequenzflexibilität hervorgehoben werden. Die Maschinenflexibilität bedeutet, dass ein Arbeitsschritt nicht an eine bestimmte Ressource gebunden ist, sondern von einer Untermenge aller verfügbaren Ressourcen durchgeführt werden kann. Die Prozessflexibilität ermöglicht es, alternative Arbeitsschritte zur Herstellung eines Produkts zu berücksichtigen. Die Sequenzflexibilität erlaubt es, von starren Bearbeitungsreihenfolgen abzuweichen, wie sie beim FSP und JSP vorgeschrieben sind und beim OSP gänzlich fehlen [SK99; BFR15; Zh20]. Für das erweiterte FJSP mit Maschinen-, Prozess- und Sequenz- oder Routingflexibilität können zwei Arten hervorgehoben werden, die von [Li22] neben anderen klassifiziert wurden:

Das **FJSP mit Prozessplanung** fokussiert das Scheduling mit flexiblen Prozess- und Auftragsreihenfolgen und mehrstufigen Auftragsstrukturen. Das FJSP mit Prozessflexibilität kombiniert die Prozess- und Ablaufplanung, um die Leistung beider Funktionen zu verbessern und dem Fertigungssystem zu einer höheren Reaktionsschnelligkeit im dynamischen Wettbewerbsumfeld zu verhelfen. Zanjirani und Özgüven haben das FJSP mit Prozessplanung als Optimierungsproblem mit integrierter Prozess- und Routingflexibilität beschrieben [ÖÖY10; ZE20]. Dieses Optimierungsproblem kommt der Problemstellung in CPMPS bereits sehr nahe und wurde im Kontext von CPMPS bereits angewendet [Bo18].

Das **FJSP mit Montageplanung** berücksichtigt die Tatsache, dass die meisten Produkte aus mehreren Teilen und Teilbaugruppen bestehen, wodurch die Montagebeziehung auf dem Shopfloor berücksichtigt werden muss [Sh10]. Die Betrachtung des FJSP mit integrierter Montageplanung ermöglicht die Betrachtung realer Produktionsumgebungen und -beziehungen. Es beinhaltet die flexible Bearbeitung von Teilen und das Fügen zu Teilbaugruppen (Modulen) und Produkten. Aufgrund der Montagebeziehungen steigt die Komplexität im Vergleich zum reinen FJSP deutlich, weshalb aktuelle Forschungsarbeiten darauf abzielen, effiziente mathematische Modelle und Algorithmen zu entwickeln [Li22].

Die verschiedenen Varianten der Ablaufplanungsprobleme besitzen drei gemeinsame grundsätzliche Annahmen. Erstens kann eine Ressource immer nur einen Arbeitsschritt gleichzeitig bearbeiten. Zweitens kann nach Abschluss eines Arbeitsschritts die Ressource für weitere Arbeitsschritte erneut genutzt werden. Drittens erfordert die Bearbeitung eines Arbeitsschritts immer eine einzelne Ressource zur gleichen Zeit [JP19]. Dabei muss beachtet werden, dass Arbeitsabläufe oft nicht vollständig linear bearbeitet werden, sondern Vorrangbeziehungen zwischen Arbeitsschritten bestehen. Eine Vorrangbeziehung bedeutet, dass ein Arbeitsschritt erst durchgeführt werden kann, nachdem bestimmte andere Arbeitsschritte abgeschlossen wurden.

Zur Lösung eines erweiterten FJSP werden Optimierungsverfahren benötigt, die im nächsten Abschnitt vorgestellt werden.

2.4.2 Ansätze zur Lösung von Reihenfolgeoptimierungsproblemen flexibler CPPS

Die Ablaufplanung in CPMPS wird als erweitertes Flexible Job Shop Scheduling behandelt. Erweiterte FJSP stellen ein kombinatorisches Optimierungsproblem dar, welches typischerweise nach einer Lösung sucht, die unter gegebenen Einschränkungen das bestmögliche Ergebnis erzielt. Diese Probleme zeichnen sich durch einen endlichen Lösungsraum aus, der die Gesamtheit aller zulässigen Lösungen für das Problem darstellt, in dem diskrete Entscheidungen getroffen werden müssen [SHS09; LM13]. Die Lösung dieser Probleme erfolgt entweder durch exakte Lösungsverfahren oder durch Heuristiken. Probabilistische Lösungsverfahren sind hingegen nicht darauf ausgelegt einen großen Lösungsraum effizient zu durchsuchen. Heuristiken werden oft eingesetzt, um die Optimierung in praktisch relevanten Zeiträumen zu ermöglichen, obwohl sie keine Garantie für eine optimale Lösung bieten [LM13].

Domschke et al. bieten eine umfassende Übersicht und sinnvolle Kategorisierungen der bekannten Methoden [Do15]. Anschließend wird eine Übersicht über einige zentrale Verfahren gegeben.

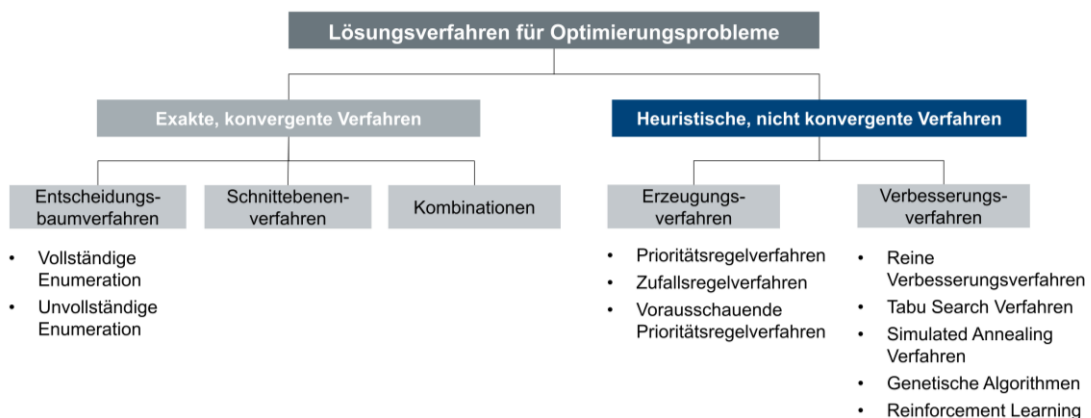


Abbildung 2.22 Übersicht von Lösungsverfahren für Optimierungsprobleme nach [Do15]

In komplexen Planungsproblemen mit vielen Variablen und Lösungsoptionen wird oft auf computerbasierte Entscheidungsunterstützung zurückgegriffen. Mittlerweile werden die Lösungsverfahren mit Verfahren der künstlichen Intelligenz ergänzt und kombiniert, was z. B. mit agentenbasierten Verfahren möglich ist. Dadurch kann u. a. die Qualität von vorausschauenden Prioritätsregelverfahren gesteigert werden. In der Studie von Li et al. ist eine detaillierte Übersicht des Einsatzes von Optimierungsverfahren zu verschiedenen Ausprägungen des erweiterten FJSP zu finden [Li22]. Nachfolgend werden mögliche Verfahren als Ansätze für die Implementierung von Optimierungsmodellen vorgestellt.

Exakte Lösungsmethoden zielen darauf ab, die optimale Lösung gemäß des Zielkriteriums zu finden. Sie verwenden mathematische Modelle wie Mixed Integer Linear/Non-Linear Programming (MIP, MILP oder MINLP) und spezialisierte Software wie IBM ILOG CPLEX. Exakte Methoden sind jedoch auf kleinere bis mittelgroße Probleme beschränkt, da die Berechnungszeit mit der Problemgröße exponentiell steigt, was als "kombinatorische Explosion" bekannt ist [SG13].

Heuristiken sind Lösungsansätze für mathematische Modelle, die zwar keine optimale Lösung garantieren, aber in akzeptabler Zeit gute Lösungen generieren und demnach einen Kompromiss zwischen der Planungszeit und der Qualität der Lösung bieten [LM13]. Sie sind besonders nützlich für große und komplexe Probleme, bei denen die optimale Lösung nicht praktikabel ist. Bei heuristischen Verfahren kann zwischen Erzeugungsverfahren und Verbesserungsverfahren unterschieden werden. Zu den Erzeugungsverfahren gehören regelbasierte Verfahren, Zufallsregelverfahren oder (vorausschauende) Prioritätsregelverfahren. Diese werden häufig für Dispatching Verfahren verwendet. Bei den Verbesserungsverfahren existieren entweder reine Verbesserungsverfahren, oder aber auch Metaheuristiken. Hier sind z.B. die Metaheuristiken der lokalen Suche zu nennen. Diese beschreiben Strategien, wie der Lösungsraum durch gezielte Änderungen an den Lösungen systematisch durchsucht wird, um ein globales Optimum zu finden, das eine der potenziell besten Lösungen hinsichtlich des Zielfunktionswerts ist. Die Suche nach einem globalen Optimum endet anhand eines festgelegten Abbruchkriteriums, wie beispielsweise der maximalen Anzahl von Iterationen. Die Herausforderung besteht darin, eine vorzeitige Konvergenz zu einem lokalen Optimum zu verhindern [SG13].

Das **Tabu Search Verfahren** ist eine Methode, die von Grund auf so entwickelt wurde, dass sie lokale Optima umgehen kann. Sie erlaubt vorübergehende Verschlechterungen von Lösungen und verwendet ein "Gedächtnis" in Form einer sogenannten Tabuliste, um die kürzlich vorgenommenen Lösungsänderungen rückgängig zu machen. Dadurch wird ermöglicht, dass die Suche, trotz vorübergehender Verschlechterungen, aus einem lokalen Optimum ausbrechen und sich weiter durch den Lösungsraum bewegen kann [SG13].

Die **Variable Neighborhood Search** verfolgt die Strategie zwischen verschiedenen Nachbarschaftsoperatoren gezielt zu wechseln. Zwei oder mehr solcher Operatoren werden abwechselnd eingesetzt. Sie werden zuerst priorisiert, und die Suche beginnt mit dem ersten Operator. Wenn bestimmte Kriterien erfüllt sind, wie zum Beispiel das Erreichen eines lokalen Optimums, wird auf den nächsten Operator gewechselt. Dies basiert auf der Erkenntnis, dass eine Lösung, die

lokal optimal in Bezug auf einen Nachbarschaftsoperator ist, nicht zwangsläufig auch lokal optimal für einen anderen Operator ist. Das globale Optimum ist jedoch in Bezug auf jeden Operator ein lokales Optimum [Ha10].

Simulated Annealing wird ebenfalls verwendet, um lokale Optima zu überwinden. Die Strategie zur Überwindung lokaler Optima unterscheidet sich von der Tabusuche insofern, als dass Lösungen mit einer festgelegten Wahrscheinlichkeit akzeptiert werden, auch wenn sie zu einer Verschlechterung der Zielfunktion führen. Diese probabilistische Komponente wird eingeführt, um lokale Optima zu überwinden und das globale Optimum zu erreichen. Laut Domschke hängt die Wahrscheinlichkeit, eine schlechtere Lösung anzunehmen, vom Ausmaß der Verschlechterung und dem Fortschritt des Lösungsprozesses ab [Do15]. Zur Steuerung dieser Wahrscheinlichkeit wird ein sogenannter Temperaturparameter verwendet, der sich im Laufe des Lösungsprozesses dem Wert Null annähert.

Genetische Algorithmen basieren auf dem Prinzip der natürlichen Evolution („survival of the fittest“), die als heuristische Methode beschrieben wurde [Ho92]. In der klassischen Vorgehensweise eines genetischen Algorithmus erfolgt zunächst die Initialisierung durch die Erzeugung einer zufälligen Startpopulation potenzieller Lösungen. In einer Optimierungsschleife wird die Fitness jedes Individuums anhand einer zuvor definierten Fitnessfunktion bewertet, wobei eine Normierung der Leistungskennzahlen erfolgt, um einheitliche Einflüsse sicherzustellen. In einem weiteren Schritt werden durch genetische Operationen wie Selektion, Rekombination und Mutation eine neue Population erzeugt. Die Fitness beeinflusst das Überleben der Individuen, und die genetischen Operationen werden mit festgelegten Wahrscheinlichkeiten angewendet. Die Optimierungsschleife endet, wenn die Abbruchbedingung erfüllt ist, und das Individuum mit der höchsten Fitness in der aktuellen Population gilt als Optimum [SD08].

Zur Lösung von erweiterten FJSP spielen auch die in Abschnitt 2.2.1 vorgestellten **MAS** eine Rolle. Agentenbasierte Lösungsverfahren stehen vor der Herausforderung der verteilten Erfüllung von Nebenbedingungen bei gleichzeitiger globaler Optimierung. Obwohl alle Agenten eine Nebenbedingung erfüllen, kann diese von der Gesamtheit der Agenten verletzt werden. Eng verbunden mit MAS sind Ant-Colony-Optimization (ACO)-Ansätze, die auf Schwarmintelligenz basieren [IB13; DS19]. Ein weiterer Ansatz im Agentenkontext ist das Reinforcement Learning (*engl. Bestärkendes Lernen*), einer Art des Maschinellen Lernens, in dem Agenten trainiert werden, Entscheidungen aus den Folgen ihrer Handlungen zu treffen [Er21].

Zusammenfassend lässt sich feststellen, dass das Optimierungsproblem, welches zur Ablaufplanung von veränderungsfähigen CPMPs gelöst werden muss, als erweitertes FJSP in Kombination mit den relevanten Flexibilitätsarten in der Literatur behandelt wird und dazu bereits verschiedene Modelle und Lösungsverfahren existieren. Die Herausforderung besteht in der geeigneten anwendungsfallspezifischen Modellierung der Optimierung. Dabei scheint die Kombination von FJSP mit Prozessplanung und FJSP mit Montageplanung für produkt-, prozess-, ressourcen- und sequenzflexible Produktionssysteme mit mehrstufigen Auftragsstrukturen interessant.

2.5 Stand der Forschung zur autonomen Steuerung von CPMPS

Nach der Vorstellung der Grundlagen und Ansätze für die autonome Steuerung veränderungsfähiger CPPS wird im folgenden Abschnitt ein Überblick über den Stand der anwendungsorientierten Forschung zur autonomen Steuerung von CPMPS gegeben. Dazu werden relevante Forschungsvorhaben verschiedener Forschergruppen bezüglich ihrer Fähigkeit zur autonomen Optimierung und Rekonfiguration untersucht. Die Vorhaben können dabei gestalterisch-planerisch oder operativ-steuernd unterschieden werden. Die Matrixproduktionssysteme werden zwar abhängig von den Forschergruppen mitunter auch als „frei verkettete Montagesysteme“ oder „Modulare Produktion“ benannt, sind aber mittlerweile als eigenständiges Forschungsfeld unter dem Begriff CPMPS etabliert [PI22]. Mit Fokus auf der Zielsetzung dieser Arbeit wird der Einsatz von Ansätzen für die Lösung von erweiterten FJSP und die Umsetzung agentenbasierter Architekturansätze für eine autonome Rekonfiguration analysiert. Besonders wird die Berücksichtigung von mehreren Produkt- und Produktionssystemebenen sowie die ebenen- und systemspezifische Anpassungsfähigkeit der Steuerung geprüft.

Eine prägende Forschergruppe auf dem Gebiet der CPMPS ist die Gruppe um Bauernhansl mit Foith-Förster, Fechter et al.. An der Universität Stuttgart, am Fraunhofer IPA und am Forschungscampus ARENA2036 werden seit 2015 Umsetzungsmöglichkeiten für Produktionssysteme „ohne Band und Takt“ untersucht [Ba15b; Ba20b]. Gegenstand der Untersuchungen sind sowohl die arbeitsorganisatorische Gestaltung der Produktionssysteme als auch mögliche Untersuchungen zur optimierten Planung. Dort wird grundlegend die Gestaltung von Prozessmodulen bzw. Stationen für effiziente Montagesysteme untersucht [FB15; FEB17]. Durch die Gestaltung adaptiver Montagezellen soll die Umsetzung von Matrixproduktionssystemen grundsätzlich ermöglicht werden [TFB20]. Die Gestaltung und Auslegung für integrierte Materialfluss- und Produktionssysteme wird aus dieser Arbeitsgruppe heraus betrachtet. Fries et al. schlagen eine Methodik vor, um systematisch die Materialversorgungsstrategie eines Matrixproduktionssystems zu gestalten [FWF20]. Die Entwicklung eines Gesamtkonzepts zur Umsetzung eines Matrixproduktionssystems „ohne Band und Takt“ in der Automobilindustrie wurde von Kern anwendungsorientiert bei der Audi AG in Ingolstadt erforscht [KRB16; KLB17; Ke21]. Kern hat ein Gesamtkonzept zur „Modularen Produktion“ und eine Methodik für die Gestaltung eines wandlungsfähigen und flexiblen Montagesystems mit integrierter Logistiksteuerung und Qualitätsprüfung entwickelt. Dabei wird das System produkt- bzw. bauteilorientiert strukturiert und für eine zustandsaktuelle Auftragsbearbeitung mit zustandsaktueller reaktiver (ad-hoc) Optimierung unter Berücksichtigung des Vorranggraphprinzips umgesetzt. Kern konnte das Potenzial des wandlungsfähigen und flexiblen Montagekonzepts für eine mitarbeitergerechte Arbeitsorganisation und Fertigungszeiteinsparung bei flächenneutraler integrierter Logistik- und Fertigungskapazität zeigen. Die Steuerungslogik wurde innerhalb einer Simulationsumgebung implementiert. Neben der Forschung zur Gestaltung und Planung in der Forschergruppe hat Waschneck eine flexible Produktionssteuerung im Rahmen des Projekts „Autonome Entscheidungsfindung in der Produktionssteuerung komplexer Werkstattfertigungen“ für die Halbleiterindustrie erforscht. Dies ist

nach Definition den SDMPs zuzuordnen. Dazu wurde die Ablaufplanung eines flexiblen Produktionssystems mithilfe eines Reinforcement-Learning-Modells zustandsaktuell optimiert. Die neuronalen Netzwerke, die durch Trainingsläufe eine Entscheidungslogik verinnerlicht haben, werden zur Optimierung von Reihenfolgen in einer autonom gesteuerten Halbleiterfertigung mit rein heterarchischem Steuerungsparadigma eingesetzt [Wa18; Wa20]. Waschneck empfiehlt die Nutzung realitätsnaher Simulationen oder eines digitalen Zwillings für die Einführung von autonom gesteuerter Reihenfolgeoptimierung. Die Erforschung einer Steuerungsarchitektur und die Betrachtung von mehreren Ebenen wird in den Forschungsarbeiten der Arbeitsgruppe bislang nicht explizit verfolgt.

Von Greschke wurde das Konzept einer Matrix-Produktion mitgeprägt und als produkt- und prozessflexibles System erforscht [Gr14]. Dieses Vorhaben gilt neben der Untersuchung von Kern als eine der ersten Untersuchungen modularer taktungebundener Fließfertigung in der Automobilindustrie. Im Gegensatz zum produktorientierten Ansatz von Kern sind die Reihenfolgen der Prozessschritte hier starr vorgegeben und berücksichtigen nicht das Vorranggraphprinzip mit flexibler Auftragsreihenfolge. Das System besteht aus redundanten prozessflexiblen Arbeitsstationen, sodass ein Auftrag flexibel mehreren Stationen zugeordnet werden kann. Es wurden erste Simulationsstudien durchgeführt, um das Systemverhalten zu analysieren. Dabei wurde die Notwendigkeit von Steuerungsstrategien und der Bedarf generischer Steuerungsarchitekturen für weitere Untersuchungen herausgestellt [Sc15]. In weiterführenden Analysen von Filz et al. wurde die Abhängigkeit des Produktions- und Materialflusses von verschiedenen Materialversorgungsstrategien untersucht [Fi19]. Die Untersuchungen wurden mittels einer agentenbasierten Simulation unter Verwendung des heterarchischen Steuerungsparadigmas auf Stationsebene vorgenommen. Büth wählte eine agentenbasierte Modellierung in einer DES für Matrixproduktionssysteme [Bu17]. Das System besteht aus neun Ressourcen mit je zwei Puffern und folgt einem vollständig dezentralen Ansatz. Dabei wird die Maschinenflexibilität für einen festen nächsten Arbeitsschritt pro Produkt berücksichtigt. Es wird keine Sequenzflexibilität oder Prozessflexibilität berücksichtigt. Die Agenten repräsentieren Produkte, Puffer und Ressourcen auf einer Ebene. Die Berücksichtigung von FTF und möglichen Engpässen im Transport wird vernachlässigt. Das Produktionssystem und das Produkt werden lediglich auf einer Ebene betrachtet und die Steuerung wird mit verschiedenen rein dezentralen Strategien analysiert. Die Simulationsstudien wurden mit entsprechender Simulationssoftware durchgeführt ohne weitere Erforschung der Steuerungsarchitektur.

An der RWTH Aachen werden Matrixproduktionssysteme auch unter dem Begriff „line less assembly systems“ oder zu deutsch „frei verkettete Montagesysteme“ von Schmitt et al. und Burggräf et al und Göppert et al. mit Fokus für die Umsetzung in der Montage untersucht [Gö18; Gö21; Gö22]. So wurde u. a. in den Projekten freeMoVe, AIMFREE und LoCoMo die Gestaltung und der Einsatz mobiler Prozessgeräte für eine agile Montage sowie verschiedene Optimierungsalgorithmen und erste Informationsmodellierungen untersucht. Hierfür wurden flexible und rekonfigurierbare Komponenten und mobile Prozessgeräte entwickelt [He19a]. Die Gestaltung er-

folgt nach prozessorientierten Kriterien [LHS19]. Hüttemann analysiert die Gestaltung und Leistungsfähigkeit von frei verketteten agilen Montagesystemen [HBS19]. Ebenso untersucht und vergleicht Göppert die verschiedenen Konfigurationsmöglichkeiten sowohl für den Linien- als auch für den Matrixbetrieb [Gö21]. Dieser Ansatz ermöglicht rekonfigurierbare Anpassungen während der Planungsphase des Systems. In einer Simulation konnte zudem unter Einsatz eines Monte Carlo Tree Search-Algorithmus nachgewiesen werden, dass durch flexible Materialflussorganisation der Einfluss von Ressourcenstörungen reduziert werden konnte [GRS20]. Im Projekt LoCoMo haben Burggräf et al. einen fähigkeitsbasierten Ansatz zur Ablaufplanung agiler Montagesysteme angewendet, mit dem Ziel eine skalierbare Optimierung unter Echtzeit-Bedingungen mit exakter Lösung von MILP zu erreichen. Für die Ablaufplanung wurde ein erweitertes Flexible Job Shop Problem modelliert mit parallelen und multifunktionalen Montagestationen und reihenfolgeabhängigen Rüstzeiten [Bu20]. Außerdem wurden Maschinenausfälle, flexible Sequenzen und Ablaufrouten bei der Planung berücksichtigt. Die exakte Lösung des MILP konnte in kleinen nicht industriell vergleichbaren Probleminstanzen keine optimale Lösung in angemessener Zeit generieren. Eine Neuplanung innerhalb weniger Sekunden konnte ebenfalls nicht erreicht werden. Für die globale Optimierung wird für die weitere Forschung ein MAS-Ansatz in Kombination mit einer Heuristik vorgeschlagen. Der Umfang der Betrachtung umfasst zwei Ebenen [Bu20]. Auf Basis der Forschungen zur Auslegung werden auch Untersuchungen für die Informations- und Steuerungsmodellierung vorgenommen. Grunert untersucht ein serviceorientiertes Informationsmodell für die integrierte Planung und Steuerung eines produkt- und routingflexiblen Matrixproduktionssystems, bei dem durch Matching identifizierte mögliche auftragsgesteuerte Produktionsrouten koordiniert werden [Gr19]. Neben den Untersuchungen zur Informationsmodellierung werden mittlerweile auch Softwarearchitekturen betrachtet. Matthews hat dazu eine modulare Softwarearchitektur zur Planung und Steuerung eines Montagesystems mit einem interoperablen Digital Twin entwickelt [Ma23]. Die Softwarearchitektur wurde in einem Demonstrator mit drei manuellen Montagestationen und einem FTF implementiert. Es wurde ebenfalls ein Ansatz mit holonischer Steuerungsarchitektur verfolgt. Entsprechend der Hardwarehierarchie in einer Montage wurde eine holonische Kapselung nach PROSA über zwei Ebenen modelliert [Bu22]. Dazu wurde unter Berücksichtigung standardisierter Ontologien eine Interaktion innerhalb der Holarchie mit zwei Ebenen zur Gesamtsystemoptimierung entwickelt. Die Validierung erfolgte als simulationsbasierte Planausführung mit zwei Ebenen und reduzierter Systemgröße unter Normalbedingungen ohne Störeinflüsse.

Aus der Arbeitsgruppe um Lanza et al. am KIT wurden Untersuchungen eines matrixorientierten Ansatzes mit Automatisierungsbezug im Projekt „Smart Body Synergy“ identifiziert. Die flexiblen Fertigungszellen mit vorrichtungsfreier Fügetechnik sollen eine Steigerung der Variantenmixflexibilität ermöglichen. Zur Lösung eines FJSP werden zentrale und dezentrale Optimierungsverfahren mit dem Ziel der Erstellung robuster Ablaufpläne kombiniert [KCF18; EL19]. Ziel ist der Einsatz und die Kombination verschiedener Optimierungsverfahren für eine dynamische Ablaufplanung für produktflexible CPMPs [ES20]. Die Untersuchungen von May behandeln eine verhandlungsbasierte Entscheidungsfindung in einer heterarchisch organisierten Produktion

[Ma21b]. Dazu werden auch die produktbezogenen Abhängigkeiten in Bezug auf die Auswirkungen auf Effizienz und Volumenflexibilität untersucht [Ma21a]. Zur Mehrzieloptimierung in der Montage wurde außerdem ein Monte Carlo Tree Search Verfahren von Hofmann et al. angewendet [St21]. Durch eine vergleichende Analyse der Optimierungsergebnisse zwischen einer Matrix- und einer Linienfertigung konnten Vorteile durch die Berücksichtigung von Routen- und Prozessflexibilität herausgestellt werden.

Behrendt et al. untersuchen CPMPs im Bereich der Intralogistik zur Optimierung der Materialbereitstellung. Dabei wird ein Algorithmus eingesetzt, der sich auf die Steuerung des Materialflusses konzentriert, ohne die Rekonfiguration auf Stationsebene zur Optimierung des CPMPs zu berücksichtigen [Sc21].

Aus der Forschergruppe um Nielsen wird eine integrierte Steuerung von Produktionszellen und FTFs in einer zentralisierten Architektur unter Verwendung von RL-Optimierung vorgeschlagen [Ni23]. Es wird eine modulare, containerbasierte Softwarearchitektur nach dem C4-Modell vorgeschlagen, die die Steuerungsfunktionen mit zentraler Steuerung auf Stationsebene umsetzt. Zur Lösung des integrierten Optimierungsproblems mit rekonfigurierbaren Stationen und flexiblen FTF wurde ein Reinforcement-Learning-Verfahren eingesetzt, das mithilfe einer Simulation eines CPMPs trainiert wurde. Zudem wurde die Wechselwirkung zwischen modularem Produktdesign und modularen Produktionssystemen hervorgehoben [NY22]. Die Steuerung wurde mit einem Digital Twin für den Einsatz in einem Matrixproduktionssystem vorbereitet. Das Steuerungssystem berücksichtigt die Gesamtsystemebene ohne weitere Granularität. Es konnten keine Angaben zur Systemgröße des betrachteten Anwendungsfalles identifiziert werden.

Die Forschergruppe um Ruskowski et al. erforscht Konzepte zur menschenzentrierten modularen Produktion unter dem Begriff „Production Level 4“ im Verbund DFKI Kaiserslautern, RPTU Kaiserslautern-Landau (bis 2022 TU Kaiserslautern) und Smart Factory e.V.. Die Untersuchungen werden u. a. demonstratorbasiert in der Forschungsumgebung der Smart Factory e.V. und dem Lehrstuhl Werkzeugmaschinen und Steuerungen Kaiserslautern (WSKL) implementiert und validiert. Dabei werden im Bereich der flexiblen Fertigung Informationsmodelle und Steuerungskonzepte nach dem fähigkeitsbasierten Ansatz von Hermann, Volkmann und weiteren erforscht [He19b; HRR20; He21; Vo21; Vo23]. Hermann et al. entwickeln eine Systemarchitektur nach dem C4 Modell für die Generierung flexibler Prozessketten nach dem PPR Prinzip [HRR20]. Darüber hinaus werden simulationsbasierte Untersuchungen zur ganzheitlichen Ablaufoptimierung von modularen Fertigungszellen und mobilen Transportmitteln von Popper und Yfantis vorgenommen, die auf Methoden des Deep Reinforcement Learning, ILP oder prädiktiver Verfahren basieren [PYR21; Yf22a; Yf22b]. Die Möglichkeit resilient zu fertigen wird im Verbund autonomer Module durch das Prinzip der subsidiären Entscheidungsfindung gefördert und steht im Gegensatz zu den vorherrschenden starr durchgeplanten Produktionsplänen [Be22]. Diesem Prinzip folgend wurden Untersuchungen mit agentenbasierten holonischen Ansätzen zur Entwicklung einer autonomen Steuerungsarchitektur für die ganzheitliche Produktionsplanung und -steuerung von CPMPs mit mehreren System- und Produktebenen vorgenommen [Ko21; Ko22; Ko23].

Es existieren neben den vorgestellten Forschungsaktivitäten der Arbeitsgruppen auch Einzelprojekte, die keiner übergeordneten Forscher- oder Arbeitsgruppe zugeordnet werden können. Ein weiterer Anwendungsfall im Bereich modularer Montagesysteme zur Optimierung von produkt- und prozessflexiblen Systemen wurde im Rahmen des Projekts "Modulares Produktionsmanagement" mit Anwendung bei der Audi AG in der Motorenfertigung am Standort Győr identifiziert. In diesem Projekt wurde die Entwicklung von Algorithmen zur dynamischen und echtzeitfähigen Ablaufoptimierung eines flexiblen und rekonfigurierbaren Systems auf Gesamtsystemebene fokussiert. In dem Projekt wurde ein „partial look-ahead“ Optimierungsalgorithmus eines erweiterten FJSP mit Produkt- und Prozessflexibilität für eine prädiktiv-reaktive Ablaufoptimierung genutzt [ME19]. Mayer verfolgt neben dem zentralen Ansatz auch einen verhandlungsbasierten dezentralen Ansatz mit Reinforcement Learning Optimierung [Ma19a; MCE21]. Zur Validierung der Algorithmen für die Steuerung des Montagesystems hat Mayer ein agentenbasiertes Simulationsframework verwendet [Ma19b].

Die anwendungsorientierten Untersuchungen von Fay und Gehlhoff in flexiblen Automatisierungsumfeldern erforschen die integrierte Produktions- und Transportsteuerung mit agentenbasierter dezentraler Architektur. Gehlhoff hat aufbauend auf bekannten Referenzarchitekturen wie PROSA und ADACOR effiziente Kommunikationsprotokolle und Informationsmodelle für die Prozess- und Ablaufplanung von Produktions- und Transportressourcen entwickelt. Mithilfe eines effizienten Kommunikationsprotokolls und des Informationsmodells wird eine skalierbare Kommunikationsarchitektur entwickelt [Ge20; Ge23]. Neben der Modellierung der Kommunikations- und Informationsstruktur werden geeignete Algorithmen für die agentenbasierte Prozess- und Ablaufplanung untersucht mit eingeschränktem Flexibilitätsgrad [Ge23]. Die Architektur ist rein heterarchisch und umfasst die Ressourcenebene in Systemen von nicht-industrieller Größe. Eine Untersuchung von mehreren Produkt- oder Systemebenen ist nicht erfolgt, die Berücksichtigung zusätzlicher Flexibilitäten und Kombination von Algorithmen wird für zukünftige Untersuchungen vorgeschlagen.

Im Rahmen des Forschungsprojekts „*SmartFace – Smarte micro Factory für Elektrofahrzeuge*“ wurden Ansätze zur Produktionsplanung und -steuerung von verteilten dezentralen Stationen mit flexibler Fördertechnik für die Montage untersucht. Darin wurden Aspekte der Programmplanung und Kommunikationsfähigkeit mit dezentralen Systemen für flexible Materialflusspfade berücksichtigt. Im Rahmen des Projektes hat Bochmann eine Methodik zur Auslegung von dezentral gesteuerten Produktionssystemen mit integrierter Transport-, Layout- und Maschinenbelegungsplanung nach dem Vorranggraphprinzip auf Gesamtsystemebene entwickelt. Der Fokus lag dabei auf den Veränderungsfähigkeiten Flexibilität und Wandlungsfähigkeit [Bo16]. Die Validierung der Methodik erfolgt in einer zentralisierten Architektur mithilfe einer agentenbasierten Simulation ohne Mehrebenenbetrachtung [Bo17; Bo18].

Zusammenfassend lässt sich feststellen, dass zur Planung und Steuerung von CPMPs bislang vor allem Algorithmen zur Lösung von Optimierungsproblemen erforscht werden, die dem gestalterisch-planerischen Bereich zugeordnet werden können. Dazu werden in der Forschung Modelle

und Methoden mit Bezug zur Layout-, Transport-, Belegungs- und Materialbereitstellungsplanung untersucht. Die Anwendung erfolgt zumeist im Kontext der Montage. Die autonome Optimierung und Rekonfiguration der CPMPs mit globaler Zielverfolgung von der obersten Matrixproduktionssystemebene des CPMPs bis zur Ressourcenmodulebene und den entsprechenden Produktebenen mit den ebenen- und systemspezifischen Freiheitsgraden wird dagegen noch nicht explizit behandelt. Aufgrund der limitierten Systemgröße bisher betrachteter Anwendungsfälle stellt sich die Frage nach der möglichen Skalierbarkeit der Ansätze für den Einsatz in industriellen Produktionssystemen mit mehreren Hierarchieebenen unter Berücksichtigung zustandsaktueller Informationen im dynamischen Betrieb. Zur Befähigung von autonom gesteuerten CPMPs für automatisierte Produktionssysteme sollten auch Aspekte der SDMPs zur Prozesssteuerung berücksichtigt werden. In Bezug auf Steuerungsarchitekturen für CPMPs wurde in der jüngeren Vergangenheit ein Anstieg der Forschungsaktivitäten identifiziert, die holonische und agentenbasierte Umsetzung berücksichtigen.

2.6 Zusammenfassung und Analyse autonomer CPMPs

Im folgenden Abschnitt werden die bisherigen Erkenntnisse zusammengefasst und der weitere Forschungsbedarf in Bezug auf die Forschungsfragen herausgestellt. CPMPs zeichnen sich durch eine Kombination veränderungsfähiger Systeme aus, die ganzheitlich koordiniert werden sollen. Nach der Definition und Einordnung von CPMPs wurde der Bereich der Produktionsplanung und -steuerung sowie der Bereich der Steuerungsarchitekturen für veränderungsfähige CPPS mit autonomer Entscheidungsfindung untersucht.

Durch die I4.0-Initiative wurde das Paradigma der serviceorientierten Architekturen auf CPPS mit verteilten dezentral gesteuerten CPS angewendet. SoA zeichnen sich vorwiegend durch ihre Interoperabilität aus und können mit agentenbasierten Architekturen für autonome Entscheidungsfindung kombiniert werden. Serviceorientierte holonische oder agentenbasierte Steuerungsarchitekturen verbinden dadurch die charakteristischen Eigenschaften der beiden Architekturstile zu interoperablen und autonomen Steuerungsarchitekturen. Multi-Agenten Systeme und holonische Steuerungsarchitekturen setzen hierarchische, heterarchische und semi-heterarchische Steuerungsparadigmen um. Holonische Steuerungsarchitekturen besitzen die Möglichkeit, die verschiedenen RAMI4.0-Architecturebenen abzudecken und Holarchien zu bilden. Sie zeichnen sich besonders durch Rekursivität und Reaktivität aus und können selbstorganisiert zwischen den Steuerungsparadigmen wechseln und ein System autonom rekonfigurieren. Neben den weit verbreiteten holonischen (Referenz-) Steuerungsarchitekturen konnte ein Ansatz identifiziert werden, welcher die Produktionsplanung mit Ablauf- und Prozesssteuerung über mehrere Ebenen betrachtet mit eingeschränkter Berücksichtigung system- und ebenenspezifischer Freiheitsgrade [He02; He03a]. Es kann somit festgestellt werden, dass bereits autonome Fertigungssteuerungssysteme für die Umsetzung flexibler und rekonfigurierbarer Fertigungssysteme existieren. Allerdings sind diese Systeme nicht gezielt dafür ausgelegt, um die Komplexität eines CPMPs mit

all seinen unterschiedlichen Veränderungsfähigkeiten ganzheitlich über mehrere System- und Produktebenen zu bewältigen.

Die Produktionsplanung und -steuerung behandelt die Planung und Ausführung von Aufträgen unter Berücksichtigung der Material- und Ressourcenverfügbarkeiten. Die Optimierung eines veränderungsfähigen CPPS stellt ein integriertes Produktionsplanungs-, Ablauf- und Prozessplanungsproblem dar, welches als erweitertes Flexible Job Shop Problem beschrieben werden kann. Zur Ermittlung von durchführbaren möglichen flexiblen Auftrags- und Ressourcenreihenfolgen sind die ressourcenspezifischen Freiheitsgrade und die produktspezifischen Verbaurestriktionen nach dem Vorranggraphprinzip zu berücksichtigen. Das Ablaufplanungsproblem in modularen Systemen stellt ein NP-schweres Optimierungsproblem dar mit verschiedenen Flexibilitätsarten. Die anwendungsfallspezifische Komplexität für CPMPS mit mehreren Auftragsstufen kann über die integrierte Berücksichtigung der Prozess-, Produkt-, Ressourcen- und Routing- bzw. Sequenzflexibilität erfolgen. Zur Lösung dieser kombinatorischen Optimierungsprobleme mit anwendungsfallabhängiger Komplexität eignen sich Modelle und Methoden des Operation Research. Die Herausforderung des FJSP mit NP-schwerem Optimierungsverhalten liegt in dem Dilemma des Erreichens von bestmöglicher Lösungsgüte bei kürzestmöglicher Lösungszeit. Die Fokussierung auf lediglich einen der beiden Aspekte kann dazu führen, dass die Potenziale der veränderungsfähigen Systeme nicht ausgeschöpft werden können. Es existieren Modelle und Methoden für erweiterte FJSP mit integrierter Prozessplanung und Montageplanung, welche für die Optimierung von CPMPS berücksichtigt werden können.

In der Analyse des Standes der Forschung zur autonomen Steuerung von CPMPS als Anwendungsbeispiel integrierter flexibler, rekonfigurierbarer und umrüstbarer CPPS konnte eine Vielzahl von Forschungsarbeiten unterschiedlicher Forschergruppen auf diesem verhältnismäßig neuen Forschungsfeld identifiziert werden. Die Forschung auf diesem Gebiet beschäftigt sich bei Unterscheidung in gestalterisch-planerischen und operativ-steuernden Aspekten bisher vor allem mit der Gestaltung und Planung der CPMPS. Dazu werden vor allem Optimierungsverfahren für dynamische Ablaufplanungsprobleme für produkt- und prozessflexible Systeme untersucht und kombiniert. Zur Lösung flexibler Ablaufplanungsprobleme werden z. B. Prioritätsregelverfahren, Heuristiken, Metaheuristiken oder auch RL-basierte Optimierungsverfahren untersucht. Die Validierung von Optimierungsalgorithmen zur Auslegung der CPMPS als integrierte Gesamtsysteme erfolgt mithilfe von Simulationen in agentenbasierten Frameworks wie in [Ma19b] und [Bo18]. Dabei wird der Materialfluss auf Stationsebene mit hierarchischen oder heterarchischem Steuerungsparadigma optimiert und idealisiert simuliert. Es ist zu beobachten, dass mittlerweile auch agentenbasierte Ansätze für die Kollaboration verteilter dezentral gesteuerter Systeme nach heterarchischem Steuerungsparadigma verfolgt werden. Ein Ansatz von Burggräf et al. zur Kombination von agentenbasierter und heuristischer Methoden zur Verbesserung der Lösungszeit wurde für die weitere Betrachtung als interessant identifiziert. Die vorhandenen Steuerungssysteme schöpfen noch nicht das volle Veränderungspotenzial eines Matrixproduktionssystems über die verschiedenen Ebenen aus. Aufgrund der bisherigen Berücksichtigung der Produkt-, Pro-

zess- und Routing- bzw. Sequenzflexibilität für die Produktionssteuerung können Materialflussrouten und Stationsbelegungen auf Stationsebene bereits abgebildet werden. Die Veränderungsmöglichkeiten der Ressourcenebene werden aktuell noch nicht integriert berücksichtigt.

Nach der Analyse bisheriger Vorhaben kann festgestellt werden, dass die Erforschung von Steuerungsarchitekturen für eine autonome Optimierung und Rekonfiguration von CPMPS bisher noch nicht stark ausgeprägt ist. Neben der vorliegenden Forschungsarbeit konnten zwei weitere Ansätze von den Forschergruppen um Nielsen und Göppert zur Erforschung von Steuerungsarchitekturen für CPMPS identifiziert werden. Die explizite Erforschung und Entwicklung von Steuerungsarchitekturen für autonom gesteuerte CPMPS von der Ressourcenmodul- bis zur Matrixproduktionssystemebene ist bisher allerdings noch nicht erfolgt. Der holonische Architekturansatz mit agentenbasierter Entscheidungsfindung kann grundsätzlich als geeignet für die ganzheitliche autonome Optimierung und Rekonfiguration von modularen Produktionssystemen mit hierarchischer Struktur herausgestellt werden. Aufgrund der begrenzten Systemgröße von betrachteten Anwendungsfällen kann bisher keine Aussage zur Skalierbarkeit der Steuerungsansätze gemacht werden.

Zunächst werden die bisher gewonnenen Erkenntnisse eingeordnet und weitere Anforderungen abgeleitet.

Forschungsfrage 1

Ein CPMPS besteht aus mehreren Ebenen mit verteilten Produktionsressourcen. Wie können die Abläufe in einem industriellen modular hierarchischen Produktionssystem unter Berücksichtigung zustandsaktueller Ereignisse autonom rekonfiguriert werden?

Durch holonische Steuerungsarchitekturen und eine agentenbasierte Entscheidungsfindung können Produktionssysteme sich autonom rekonfigurieren. Im Kontext der CPMPS werden agentenbasierte Ansätze bisher für die Planung und Gestaltung von CPMPS in Verbindung mit Simulationsframeworks zur Verhaltenssimulation oder der Algorithmenvalidierung verwendet. Mittlerweile werden Agentensysteme auch vermehrt zur operativen Steuerung von CPMPS eingesetzt. Eine autonome Entscheidungsfindung in veränderungsfähigen Produktionssystemen mit mehreren Ebenen und hybrider Kommunikation kann mittels einer entsprechenden Strukturierung der Interaktion der Multi-Agenten-Systeme ermöglicht werden. Es wird eine holonische Steuerungsarchitektur benötigt, die ein CPMPS als modular hierarchisches SoS in einer Holarchie mit mehreren Systemebenen unter Berücksichtigung der charakteristischen operativen Veränderungsfähigkeiten des CPMPS ganzheitlich autonom rekonfigurieren kann.

Forschungsfrage 2:

CPMPS besitzen charakteristische Veränderungsfähigkeiten mit system- und ebenenspezifischen Freiheitsgraden. Wie können CPMPS ganzheitlich optimiert werden und wie kann die Komplexität beherrscht werden?

Die Veränderungsfähigkeiten eines CPMPs können mithilfe von geeigneten Optimierungsmodellen und -methoden als erweiterte Flexible Job Shop Probleme modelliert werden. Dabei können verschiedene Flexibilitätsarten berücksichtigt werden. Für auftragsgesteuerte CPMPs sind die Produkt-, Prozess-, Ressourcen-, und Routing- bzw. Sequenzflexibilität auf ebenenspezifisch zu berücksichtigen. Die ressourcenspezifischen Freiheitsgrade zur Auftragsbearbeitung sollten bedarfsgerecht genutzt und gleichzeitig müssen die produktspezifischen Vorranggraphrestriktionen bei der Prozess- und Ablaufplanung beachtet werden. Angesichts der möglichen Individualisierung und Modularisierung von Produkt und Produktionssystem ist es entscheidend, dieses Veränderungspotenzial bei der dynamischen Auftragsallokation über mehrere Ebenen zu berücksichtigen. Produktionsstrukturen sollten in ihrer tatsächlichen Form und Granularität in der Steuerungsarchitektur repräsentiert werden, um die Komplexität angemessen abbilden zu können.

Zusammenfassend kann festgestellt werden, dass für die ganzheitliche Optimierung eines CPMPs eine dynamische Ablaufplanung eines erweiterten FJSP mit integrierter Prozess- und Montageplanung und prädiktiv reaktivem Verhalten innerhalb eines modular hierarchischen Systems benötigt wird. Auf höherer Hierarchieebene könnte eine globale Optimierung mit größerem Entscheidungshorizont erfolgen, wohingegen auf unterer Hierarchieebenen eine Entscheidung zur Auftragsausführung kurzfristig getroffen wird. Für die autonome globale Optimierung und Rekonfiguration sind die system- und ebenenspezifischen Freiheitsgrade und gleichzeitig die produkt- bzw. auftragspezifischen Vorranggraphrestriktionen zu beachten. Die Anforderungen an die Steuerungsarchitektur werden im folgenden für die daran anschließende Modellierung herausgestellt.

Anforderungen an die Steuerungsarchitektur

Die Herausforderung der autonomen Steuerung von CPMPs liegt in der ganzheitlichen Optimierung des Systems of Systems mit den systemindividuellen Freiheitsgraden und den auftragsindividuellen Vorranggraphrestriktionen. Die Berücksichtigung der Granularität von Produkt und Produktionssystem spielt eine entscheidende Rolle, um ein CPMPs realitätsnah abbilden zu können. Für die auftragsgesteuerte autonome Optimierung des Systems ist die Berücksichtigung der zustandsaktuellen Produkt-, Prozess- und Ressourceninformationen eine notwendige Grundlage. CPMPs als System of Systems zeichnen sich durch eine modulare Erweiterbarkeit und Modifizierbarkeit aus, welche allerdings den Lösungsraum zur Ablaufplanung nicht-linear ansteigen lassen. Daher werden Mechanismen zur Entscheidungsfindung benötigt, die die Komplexität eines CPMPs mit seinen unterschiedlichen Veränderungsfähigkeiten auf verschiedenen Ebenen beherrschen kann.

Nachfolgend werden Anforderungen definiert, die zur Schließung der Forschungslücke für autonom gesteuerte CPMPs als modular hierarchische SoS als Orientierung dienen sollen. Zentrale Anforderungen für eine autonome Steuerung eines CPMPSoS sind Selbstorganisation (A1) und Anpassungsfähigkeit (A2), die eng miteinander zusammenhängen. Aufgrund der Lösungskomplexität in modifizier- und erweiterbaren SoS wird die Skalierbarkeit (A3) ebenfalls ergänzend betrachtet [Ma98; Ko11; We15].

A1 Selbstorganisation

Selbstorganisation bezieht sich auf die Fähigkeit eines Systems, ohne externe Steuerung oder Anleitung zu funktionieren. Es beinhaltet die spontane Bildung einer Ordnung oder Struktur innerhalb eines Systems aus den Wechselwirkungen seiner Bestandteile. Das Gesamtsystemverhalten eines SoS kann nicht spezifisch einem System zugeordnet werden, sondern ergibt sich emergent aus dem Zusammenwirken der Systeme mit ihren individuellen Zielen. Die Steuerungsarchitektur soll somit durch die Gestaltung der internen Kommunikation und der Interaktion der Systeme definiert werden und strukturiert die Entscheidungsfindung und das Systemverhalten [Ma98]. Dadurch wird die Performance des Systems bestimmt und es können Rückschlüsse zur Gestaltung des Produktionssystems gezogen werden [He02]. In einem CPMPSoS mit mehreren Ebenen wird die Umsetzung des hybriden Steuerungsparadigmas mit einer ebenen- und systemübergreifenden hierarchischen und heterarchischen Kommunikation benötigt [Ma96; Ma98]. Durch die Autonomie von Agenten können dynamische und regelbasierte Verhalten abgebildet werden [VGL15]. Agenten können dabei auch als Teil von Planungs- und Steuerungsverfahren eingesetzt werden, sodass sie sowohl die Ziele des Gesamtsystems hierarchisch verfolgen als auch bei Störungen dezentral heterarchisch reagieren können [LCR05; LR08]. Ein CPMPSoS benötigt die Fähigkeit eines selbstorganisierten Wechsels zwischen heterarchischer Kollaboration und hierarchischer Koordination innerhalb der Holarchie [LCR05]. Durch fähigkeitsbasierte Interoperabilität und das PPR-Matching wird die Grundlage für eine selbstorganisierte Allokation der Aufträge geschaffen [He19b, He20, He21]. Das Gesamtsystem sollte über alle Hierarchieebenen hinweg abgebildet werden können und die zur Entscheidungsfindung notwendige hybride Kommunikation unterstützen. Zur autonomen Steuerung des CPMPs soll daher eine selbstorganisierte Optimierung und Rekonfiguration hierarchisch und heterarchisch erfolgen können. Dies wird für die weitere Modellierung in der Anforderung **A1.1 Hierarchische fähigkeitsbasierte Optimierung und Rekonfiguration**, sowie **A1.2 Heterarchische fähigkeitsbasierte Optimierung und Rekonfiguration** definiert.

A2 Anpassungsfähigkeit

Die Anpassungsfähigkeit beschreibt die Fähigkeit eines Systems, sich an veränderte Bedingungen, Umstände oder Anforderungen anzupassen. Es beinhaltet die Bereitschaft und Fähigkeit, sich flexibel zu ändern, um mit neuen Gegebenheiten umzugehen. Dies ermöglicht auch die Integration neuer Technologien und eine damit verbundene Erweiterbarkeit und Modifizierbarkeit. Die Anpassungsfähigkeit wird in dieser Arbeit mit Optimierung und Rekonfigurierbarkeit beschrieben. Demnach soll die Steuerungsarchitektur das CPMPs ganzheitlich optimieren und rekonfigurieren können und baut auf der Selbstorganisation (A1) auf. Durch die Nutzung der statischen und dynamischen Prozess-, Produkt- und Ressourceninformationen können die system- und ebenenspezifischen Freiheitsgrade identifiziert werden. Diese werden zur Optimierung und Rekonfiguration des CPMPs entsprechend der durch die Veränderungsfähigkeiten definierten Freiheitsgrade zielgerichtet koordiniert. Das CPMPs stellt ein veränderungsfähiges System mit mehreren Ebenen dar, welches sich entsprechend der aktuellen Auftragsbedarfe auftragsindividuell ebenen- und systemspezifisch anpassen soll. Für die ganzheitliche autonome Optimierung

und Rekonfiguration eines CPMPS mit modular hierarchischer Struktur soll eine dynamische Ablaufplanung mit prädiktiv reaktivem Verhalten unter Berücksichtigung der system- und ebenenspezifischen Flexibilitäten ermöglicht werden [Pl22]. Die für diese Arbeit relevanten Flexibilitätsarten Produkt-, Prozess-, Ressourcen- und Routing-, bzw. Sequenzflexibilität erfordern die Lösung eines erweiterten FJSP mit integrierter Prozess- und Montageplanung.

Ein CPMPS soll seine Auftragsbearbeitungsfolgen und die Systemkonfiguration gemäß einer selbstoptimierenden und adaptiven Regelung [Boe06] steuern können. Dabei soll das System globale Ziele des Gesamtsystems verfolgen und gleichzeitig den lokalen Ressourcen die Freiheit gewähren, Teilprobleme eigenständig zu lösen, um Robustheit und Reaktionsfähigkeit sicherstellen zu können [BJW04]. Farid und Ribeiro unterstreichen in [FR15], dass eine modulare Produktion an Flexibilität verlieren würde, wenn übergeordnete Ablauffolgen ohne lokale Optimierungsmöglichkeiten ausgeführt würden. Ein CPMPS könnte durch die Berücksichtigung von Agenten auf mehreren Ebenen in der Lage sein, lokal zu entscheiden und gleichzeitig globale Ziele zu verfolgen. Systeme höherer Hierarchieebenen nehmen eine globale Optimierung mit größerem Entscheidungshorizont vor, wohingegen Systeme unterer Ebenen ihre Entscheidungen kurzfristig lokal optimieren und anpassen können [He02]. Die integrierte Reihenfolgeoptimierung des CPMPS soll die system- und ebenenspezifischen Veränderungsfähigkeiten für eine zustandsaktuelle Optimierung und Rekonfiguration berücksichtigen. Dabei wird der Einsatz von Modellen und Methoden zur Ablaufoptimierung unter Berücksichtigung der lokalen system- und auftragsspezifischen Produkt-, Prozess-, Ressourcen- und Routingflexibilität benötigt. Zur anpassungsfähigen Steuerung eines CPMPS soll daher eine lokale und globale Optimierung und Rekonfiguration ermöglicht werden, welche Anforderung **A2.1: System- und ebenenübergreifende Optimierung und Rekonfiguration** und **A2.2: System- und ebenenspezifische Optimierung und Rekonfiguration** beschreiben.

A3 Skalierbarkeit

Ein CPMPSoS zeichnet sich durch eine modulare Erweiterbarkeit und Modifizierbarkeit aus. Durch das Hinzufügen von Assets (Systeme, Produkte) oder auch die Erweiterung existierender Assets um ergänzende Fähigkeiten und Skills steigt der Lösungsraum zur Optimierung der Ablaufreihenfolgen nicht-linear an. Daher werden Mechanismen benötigt, die die Entscheidungsfindung eines CPMPS mit seinen ebenen- und systemspezifischen Veränderungsfähigkeiten in seiner Komplexität beherrschbar machen. Erste Ergebnisse zu Lösungen eines erweiterbaren FJSP mit MILP im Kontext von CPMPS konnte keine optimalen Ergebnisse in annehmbarer Zeit und keine Neuplanung innerhalb von Sekunden erreichen [Bu20]. Ein vorgeschlagener Ansatz für eine effizientere Lösung könnten Multi-Agenten Systeme in Verbindung mit Metaheuristiken darstellen [Bu20]. Holonische Steuerungsarchitekturen wie SURFER implementieren systemindividuelle Diagnosefunktionen zur effizienten Datenverarbeitung, die die Komplexität des Gesamtsystems begrenzen [Bo19]. Die zu entwickelnde Steuerungsarchitektur soll die skalierbare autonome Optimierung von CPMPS durch rekursive Strukturen ermöglichen, die in holonischen Architekturen grundlegend sind und dem Prinzip der Fraktale entsprechen [Wa92]. Eine agentenbasierte Steuerung mit hybridem Steuerungsparadigma soll so modelliert werden, dass die Komplexität eines

CPMPs in modular hierarchische Teilprobleme zerlegt wird, um dann mit seinen system- und ebenenspezifischen Veränderungsfähigkeiten rekursiv und skalierbar optimiert zu werden [Ku19; DCT21].

Bei der Umsetzung von agentenbasierten Steuerungsarchitekturen ist es notwendig, die Granularität eines Produktionssystems zu berücksichtigen [FR15]. Dies unterstützt die globale Optimierung mit verteilter Ablaufplanung durch eine Dekomposition in Teilprobleme und kann dadurch auch eine erhöhte Flexibilität abbilden [BJW04; VGL15]. Die Rekonfigurierbarkeit in Agentensystemen sollte mehrere Ebenen entsprechend der technischen Grundstruktur des Systems berücksichtigen [FR15]. Diese Vorgehensweise verfolgt somit außerdem das Ziel die Komplexität durch die Reduktion der Freiheitsgrade zu begrenzen, sodass Optionen, die nicht für die Erfüllung des Kundenauftrags erforderlich sind, flexibel vernachlässigt werden können [BJW04]. Gemäß des Subsidiaritätsprinzip erfolgt die Koordination und Optimierung nach Möglichkeit lokal auf Grundlage der dezentral verfügbaren Informationen [Be22]. Für die skalierbare Steuerung von CPMPs werden daher die Anforderungen **A3.1 Rekursive Optimierung und Rekonfiguration** und **A3.2 Subsidiäre Optimierung und Rekonfiguration** an die Steuerungsarchitektur gestellt.

3 Agentenbasierte Modellierung einer holonischen Steuerungsarchitektur

Die Zusammenfassung und Analyse zur Befähigung von autonomen Cyber Physischen Matrixproduktionssystemen (CPMPS) hat Multi-Agenten-Systeme (MAS) und Holone als geeignete Grundlage zur Umsetzung einer Steuerungsarchitektur von verteilten Systemen identifiziert. Agentensysteme können als „standalone agents“ oder „embedded agents“ zur Simulation und Steuerung komplexer Automatisierungssysteme genutzt werden. Holonische und agentenbasierte Steuerungsarchitekturen können durch die Aspekte serviceorientierter Architekturen in ihrer Skalierbarkeit und Interoperabilität erweitert werden. Die Kombination von MAS und serviceorientierten Architekturen (SoA) stellt eine vielversprechende Grundlage für die Erforschung einer geeigneten Steuerungsarchitektur dar. Durch die Entwicklung einer holonischen Steuerungsarchitektur mit agentenbasierter Produktionssteuerung und serviceorientierter Optimierung soll ein CPMPS mit mehreren Ebenen autonom optimiert und rekonfiguriert werden können.

Ein auftragsgesteuertes CPMPS soll seine Entscheidungen auf Basis der zustandsaktuellen statischen und dynamischen Auftragsbearbeitungsoptionen treffen können. Die autonome Optimierung und Rekonfiguration des modular hierarchischen CPPS benötigt eine komplexitätsbeherrschende Steuerungsarchitektur, die das System unter Berücksichtigung der ebenen- und system-spezifischen Freiheitsgrade des Systems selbstorganisiert steuert. Dabei wird die Automatisierungsebene in übergeordnete komplexe Informationsverarbeitungs- und Entscheidungsprozesse für die Abbildung der modular hierarchische Struktur eines CPMPS integriert.

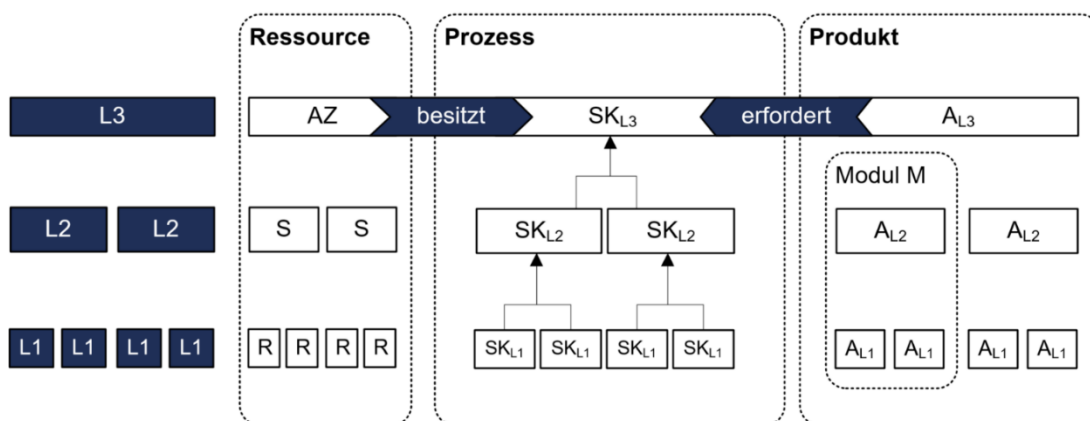


Abbildung 3.1: PPR-Beziehung in modular hierarchischen Produktionssystemen mit RAMI4.0 Hierarchieebenen

Das modular hierarchische System wird für die weitere Modellierung fähigkeitsbasiert abstrahiert und ist in Abbildung 3.1 beispielhaft verdeutlicht. Ein Auftrag des Produktes P erfordert eine spezifische (Prozess-) Fähigkeit, welche als Skill auf einer Ressource bereitgestellt wird. Eine entsprechende Zerlegung ermöglicht die Beschreibung der einzelnen Systeme nach dem PPR-Prinzip

und stellt die Grundlage für die Allokation von Aufträgen innerhalb des Systems dar. Ein Modulauftrag A_{L2} stellt einen Teilauftrag des Produktauftrags A_{L3} dar. Der Modulauftrag kann wiederum eigenständig betrachtet werden. Die Module M sind ebenfalls hierarchisch aufgebaut und besitzen Teilaufträge A_{L1} . Sie ähneln den Produkten P und können individuell behandelt und durch das Produktionssystem transportiert werden. Das Produktionssystem und das Produkt werden so in ihrer verschachtelten Struktur informationstechnisch abstrahiert.

Das modular hierarchische CPMPS wird im Folgenden mit einer agentenbasierten Steuerungsarchitektur als selbstoptimierendes System mit einer dynamischen Ablaufplanung für die flexible Ausführung von Auftragsfolgen modelliert. Dabei werden Mechanismen für eine autonome Entscheidungsfindung innerhalb des modular hierarchischen Systems entwickelt. Eine agentenbasierte Produktionssteuerung mit der Fähigkeit ein erweitertes FJSP zu optimieren, soll die Entscheidungsfindung des verschachtelten CPMPSoS definieren. Dies übernimmt die Funktion des Adaptionalgorithmus und ist in Abbildung 3.2 rechts dargestellt.

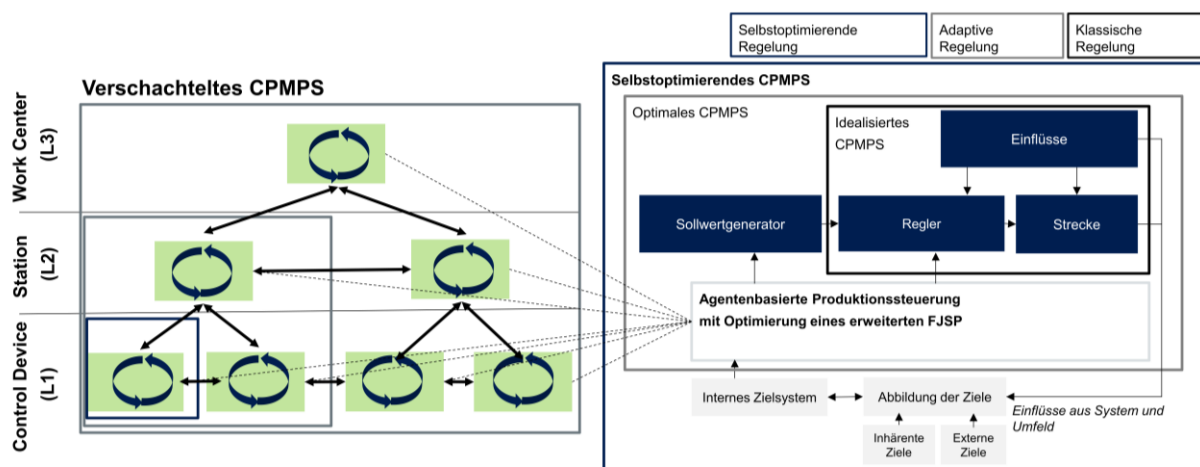


Abbildung 3.2: Entwurf eines selbstoptimierenden adaptiven Regelkreises zur Steuerung verschachtelter CPMPS

Eine ganzheitliche Anpassung gelingt allerdings nur, wenn die Entscheidungen des Systems global optimiert werden können. Ein vielversprechender Ansatz dazu könnte die Kombination eines MAS mit einer Metaheuristik sein. Dazu wird eine agentenbasierte Modellierung für die autonome Steuerung eines CPMPS auf Basis der DACS Methode (Development of Agent Control Systems) vorgenommen [Bu04; LF15]. Die Methode ermöglicht eine modellbasierte Umsetzung von Steuerungsprozessen mit MAS. Die DACS-Methode zeichnet sich durch eine Vielzahl an erfolgreichen und dokumentierten Anwendungen innerhalb der Wissenschaft und Industrie aus. Eine Analyse von fünf Agentenentwicklungsmethoden wurde nach Bewertungskriterien von Dam & Winikoff im Jahr 2004 und zusätzlich bezüglich der Skalierbarkeit und Reife der Entwicklungsmethoden bewertet. Insbesondere aufgrund ihres effektiven Komplexitätsmanagements, ihrer Einfachheit und ihrer Neutralität in Bezug auf Implementierungstechnologien kann die DACS Methode empfohlen werden [DW04; Za19]. Diese Methode wurde grundsätzlich für die Gestaltung von Entscheidungs- und Steuerungssystemen im Produktionsumfeld entworfen [VGL15; LVG18].

Kombiniert wird sie in dieser Arbeit mit Teilen der Methode ADMARMS (Axiomatic Design of a Multi-Agent Reconfigurable Mechatronic System), die speziell für die Entwicklung rekonfigurierbarer Systeme entworfen wurde und Aspekte des Systems Engineering aufgreift [FR15].

Für die autonome Produktionssteuerung eines CPMPs wird daher ein Entscheidungsmodell mit den notwendigen Steuerungsentscheidungen auf Basis eines autonomen komplexitätsbeherrschenden Steuerungsprozesses entwickelt (Abschnitt 3.1). Im Anschluss werden die Abhängigkeiten identifiziert und zu Agenten in einem Agentenmodell (Abschnitt 3.2) gekapselt. Die Interaktionen des Agentenmodells werden im Interaktionsmodell (Abschnitt 3.3) zur Definition des Systemverhaltens spezifiziert. Ergänzend zum ursprünglichen Vorgehen der Methode wird ein Optimierungsmodell zur globalen Optimierung eines erweiterten FJSP für das CPMPs formalisiert (Abschnitt 3.4).

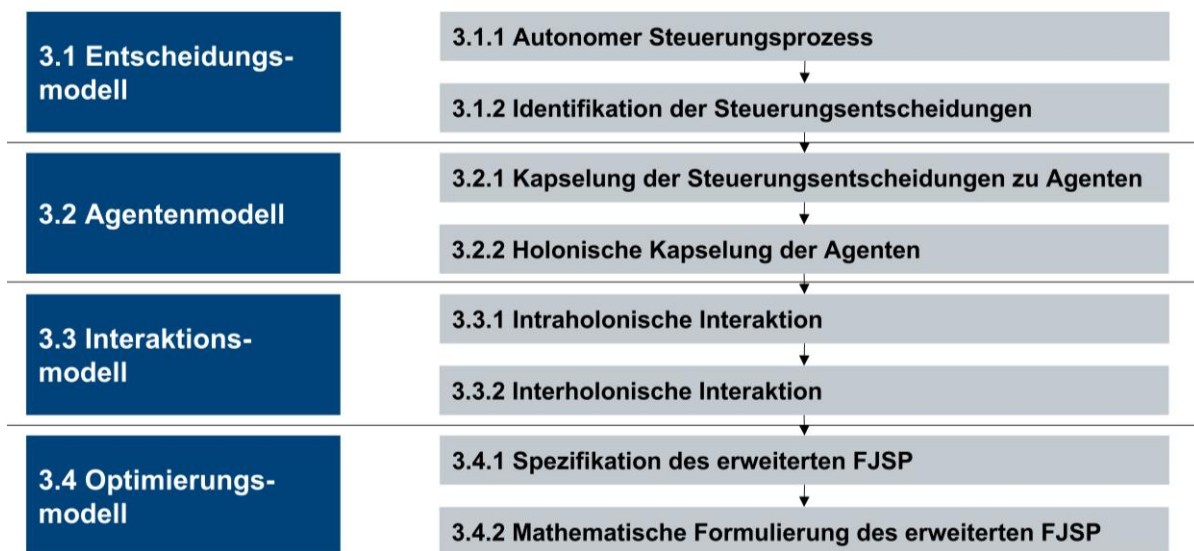


Abbildung 3.3: Struktur des Kapitels 3

3.1 Entscheidungsmodell

Die Literatur hat bisher keine allgemein gültigen übergeordneten Steuerungsentscheidungen für CPMPs definiert. Um die Steuerungsentscheidungen zu identifizieren, wird ein komplexitätsreduzierender Steuerungsprozess entwickelt und darauf aufbauend ein abstrahiertes Entscheidungsmodell erstellt, wie es in Abbildung 3.4 auf Seite 61 in generischer Form dargestellt ist. Grundlegend werden so die erforderlichen Steuerungsentscheidungen für das Produktionssystem ermittelt, um darauf aufbauend ein Agentenmodell zu erstellen.

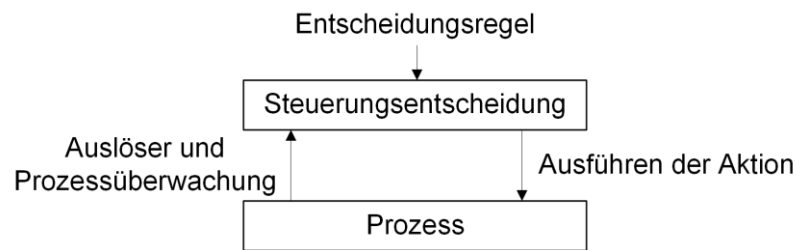


Abbildung 3.4: Abstrahiertes Entscheidungsmodell nach [LK15]

3.1.1 Autonomer Steuerungsprozess

Für die autonome Optimierung und Rekonfiguration eines CPMPs wird ein komplexitätsbeherrschender Steuerungsprozess benötigt. Dabei können grundlegende Design-Prinzipien (DP) zur Entwicklung der Steuerungsarchitektur für die Erfüllung der Anforderungen (A1-A3) Selbstorganisation, Anpassungsfähigkeit und Skalierbarkeit in dem Mehrebenensystem berücksichtigt werden.

DP1 Loose Coupling & Cohesion

Das auftragsgesteuerte CPMP-SoS benötigt die Möglichkeit zwischen verschiedenen Steuerungsarten zu variieren. Die Interaktion der Systeme des modular hierarchischen SoS soll zum einen entsprechend der hierarchischen Ebenen ermöglicht werden und zum anderen zustandsaktuelle Interaktionsbeziehungen eingehen können.

DP2 Divide & Conquer

Das globale Optimierungsproblem eines CPMPs soll mittels Divide & Conquer in mehrere Teilprobleme entsprechend der modular hierarchischen Produktionssystem- und Produktstruktur zerlegt werden können.

DP3 Top-Down & Bottom-Up

Die Produktionsentitäten und Produkte benötigen eine Top-Down und Bottom-Up Beziehung gemäß der Fertigungshierarchie. Eine Aggregation der Fähigkeiten kann nach dem PPR-Prinzip für eine ebenen- und systemspezifische Strukturierung der Interaktion erfolgen.

DP4 Monitoring & Reasoning

Die Informationen und Daten der Produktionsentitäten und Produkte des CPMPs werden zustandsaktuell überwacht. Auf Basis der statischen und dynamischen Informationen zur Auftragsbearbeitung können lokale oder globale Entscheidungen abgeleitet werden.

Zunächst wird ein Prozess zur autonomen Entscheidungsfindung auf Basis der Design Prinzipien entworfen. Das autonome CPMPs soll als auftragsgesteuertes, autonom optimierendes System über mehrere Ebenen befähigt werden.

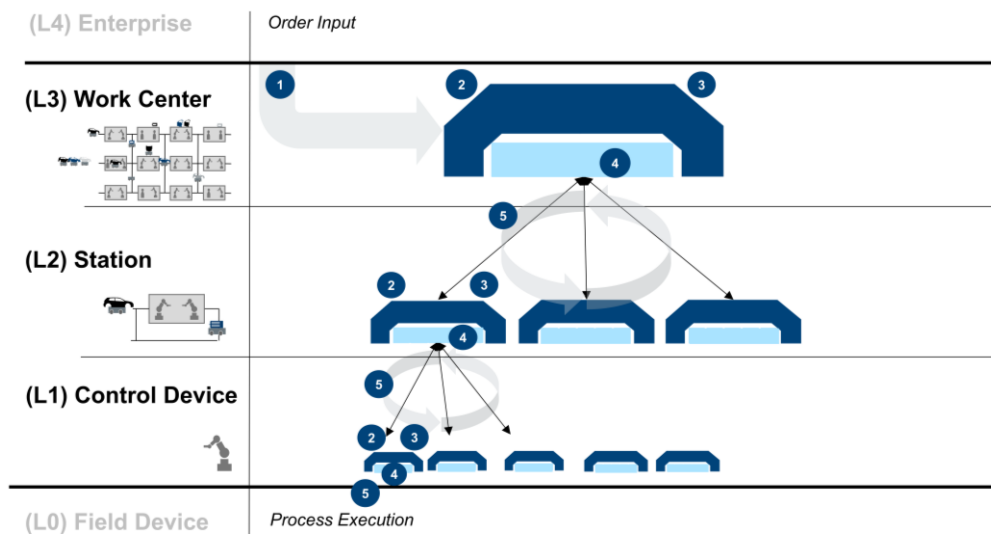


Abbildung 3.5: Autonomer ebenenübergreifender Steuerungsprozess für CPMPs

In Abbildung 3.5 ist der autonome hierarchische Steuerungsprozess dargestellt, der ein fähigkeitsbasiertes PPR Matching zur flexiblen Fertigung in vernetzten Systemen nutzt sowie die Aufgaben der Produktionssteuerung. Zur ebenen- und systemübergreifenden Angebotserstellung werden die Design Prinzipien angewendet. Das CPMPs erhält Aufträge aus einem Auftragspool und organisiert die effiziente Bearbeitung von der Auftragsplanung bis zur Auftragsausführung in den Ebenen L1 bis L3.

1. Bildung von Auftragsgruppen:

Der Ausgangspunkt sind die erzeugten Aufträge des Produktionsprogramms, die in Losgrößen gruppiert als Volumentakt in das System eingelastet werden. Die Freigabe in den Gesamtsystemraum erfolgt auf Basis von Freigaberegeln des internen Zielsystems. Die Auftragsgruppen werden modular hierarchisch zerlegt in entsprechende Jobs und Operationen, die über ebenen- und systemspezifisches Matching auf den Stationen und Ressourcen eingeplant werden sollen.

2. Erstellung systemspezifischer Angebote zur Auftragsbearbeitung

Für die Auftragsgruppen werden die Bearbeitungszeiten und Freiheitsgrade für die Auftragsausführung systemspezifisch identifiziert. Auf dieser Basis werden mögliche Reihenfolgen angefragt und individuell durch die Systeme der darunterliegenden Ebene erstellt und zu systemspezifischen Angeboten aggregiert.

3. Koordination systemspezifischer Angebote zur Auftragsbearbeitung

Für die Auswahl werden die systemspezifischen Angebote hierarchisch koordiniert. Dabei werden die verschiedenen systemspezifischen Angebote der Auftragsgruppen abgeglichen und zu einem gesamt optimalen Ablauf für die Ebene koordiniert.

4. Systemübergreifende Bestätigung der Auftragspläne:

Es erfolgt eine Feinterminierung der Auftragsbearbeitung unter Berücksichtigung der ausgewählten Angebote. Diese werden dann zur Bearbeitung in den Systemen der Ebene unter Berücksichtigung der konkreten Start- und Endzeitpunkte freigegeben.

5. Auftragsausführung und selbstorganisierte Anpassung:

In den Systemen wird die Ausführung der Auftragsfolgen mit den Planvorgaben abgeglichen und situativ angepasst. Dies stellt sicher, dass im Falle von Änderungen die weitere Verarbeitung gewährleistet ist und ein Stillstand vermieden wird. Dadurch kann das System ungeplante Ereignisse (z. B. Ressourcenstillstände) zustandsaktuell berücksichtigen und selbstorganisiert Maßnahmen einleiten.

Der Plan wird für einen definierten Zeitraum zur Ausführung festgelegt und bei Bedarf mit Maßnahmen der operativen Produktionssteuerung optimiert. Dies beinhaltet die Optimierung und Rekonfiguration der Auftrags- und Ressourcenreihenfolge unter Berücksichtigung der Vorrangbeziehungen und Veränderungsmöglichkeiten. Der zentral koordinierte Auftragsplan wird hierarchisch freigegeben und dient den Systemen auf verschiedenen Ebenen zur Orientierung bei der selbstorganisierten Ausführung. Dies entspricht einem Scheduling mit prädiktiv reaktivem Verhalten.

3.1.2 Identifikation der Steuerungsentscheidungen

Im folgenden Abschnitt werden Entscheidungen der Ebenen L1-L3 für die Steuerung des Produktionsflusses behandelt. Dabei werden die ebenen- und systemspezifischen Veränderungsmöglichkeiten für die Entscheidungsfindung strukturiert. Die Entscheidungen E1-E13 sind nachfolgend beschrieben. Eine Übersicht ist in Tabelle 3.1 aufgeführt.

E1 betrifft die Freigabe von Aufträgen für das Gesamtsystem des CPMPS. In dieser Entscheidung wird festgelegt, welches Volumen von Produkten P mit dem jeweiligen Auftrag A_{L3} zu welchem Zeitpunkt freigegeben wird. Die Auftragsfreigabe bedient sich an dem zuvor festgelegten Produktionsprogramm, welches durch die Auftragserzeugung als Teil der Produktionsplanung dem CPMPS zur Verfügung gestellt wird. Die Freigabe erfolgt in Auftragsgruppen (Losen) und erzeugt einen Volumentak [Bo16]. Diese Entscheidung wird durch den aktuellen Bestand im System oder durch eine festgelegte Freigabestrategie oder -methode und parametrierbare Freigaberegeln ausgelöst. Der Entscheidungsraum beinhaltet die Anzahl und Art der Aufträge im Auftragspool sowie deren Freigabezeitpunkte, was einen erheblichen Einfluss auf die Leistung des Systems hat, da dies die Bestandskontrolle und Durchlaufzeiten beeinflusst. Die Aufträge werden dann entsprechend ihrer Produktstruktur hierarchisch für eine Koordination auf den verschiedenen Ebenen zerlegt.

E2 betrifft die Zuordnung des Auftragsvolumens mit den Modulaufträgen zu den Stationen und die Festlegung des Bearbeitungszeitpunkts. Diese Entscheidung wird mithilfe des Fähigkeitsabgleichs auf Stationsebene getroffen. Der Entscheidungsraum umfasst die Verteilung der Modulaufträge auf die Ressourcen unter Berücksichtigung der Anpassungs-, Transport- und Ausführungszeiten. Die Optimierung dieser Zuordnung kann nach verschiedenen Zielgrößen erfolgen, die auf L3-Ebene zentral festgelegt werden. Die Entscheidung **E2** benötigt die Ergebnisse der Entscheidung **E3** aufgrund der bottom-up Aggregation. Die Entscheidung **E3** ist ähnlich zu **E2** und plant nach dem rekursiven Prinzip zentral die Zuordnung der Teilaufträge A_{L1} zu den der Station S untergeordneten Ressourcen R unter Berücksichtigung von ressourcenindividuellen Rüst- und Auftragsausführungszeiten.

E4 ist die Entscheidung auf der L1-Ebene, die über die kurzfristige Anpassung der Ausführung der Reihenfolge für Teilaufträge A_{L1} entscheidet. Obwohl die Zeitpunkte für A_{L1} bereits durch das Scheduling der Auftragsplanung geplant sind, erfordert die Komplexität und die durch eine mögliche Dynamik abweichende Auftragssequenz eine dezentrale Anpassung zur Auftragsausführung. Bei der Reihenfolgebildung der Ressourcenbelegung und Bearbeitungsreihenfolge der Aufträge A_{L1} kann die Ressource selbstorganisiert der globalen Reihenfolge abweichen.

E5 betrifft die Auswahl der kurzfristigen Anpassung der Belegung der nächsten Ressource R nach Ausführung des Auftrags A_{L1} . Dies entspricht der Maßnahme der Kapazitätssteuerung in der operativen Produktionssteuerung. Wenn das Modul die Ressource verlässt, wird die nächste Ressource angesteuert oder die Station verlassen. Diese Entscheidung ist eng mit **E6** verbunden, die Maßnahmen im Falle von identifizierten Abweichungen vom Normalzustand einleitet, um den Produktionsfluss zu optimieren. Wenn es zu einer dezentralen Störung beim Erreichen des nächsten Ziels kommt, kann von der initialen Vorgabe abgewichen und eine alternative Bearbeitungsreihenfolge vorgenommen werden. Dazu werden die Maßnahmen der operativen Produktionssteuerung zur Anpassung von Auftragsfreigabe, Reihenfolgebildung und Kapazität gewählt.

Die folgenden Entscheidungen betreffen die Organisation des Materialflusses und der Routingflexibilität, die einen wesentlichen Einfluss auf den Produktionsfluss im CPMPs haben. Diese Entscheidungen ähneln denen zur Organisation der Produktionsressourcen, erfordern jedoch eine separate Betrachtung aufgrund der Mobilität der Transportressourcen und der daraus resultierenden geringen Stabilität der Transportpläne. **E7** entscheidet unmittelbar über die Wahl des nächsten FTF und hängt von der Auftragsallokation ab, die in **E5** bzw. **E6** durchgeführt wurde. Der Entscheidungsraum umfasst die verfügbaren FTFs mit ihren Attributen wie Status, Koordinaten und Geschwindigkeiten. **E8** ähnelt **E7**, betrifft jedoch die Wahl des nächsten FTFs innerhalb einer Station S zur nächsten Ressource R . **E9** ist ähnlich wie **E8**, betrifft jedoch die Wahl des nächsten FTFs außerhalb der Station S innerhalb des Arbeitszentrums AZ bzw. Work Centers WC zur nächsten Station S . Diese Entscheidungen erfolgen jeweils, wenn ein Produkt P im Ein- oder Ausgangspuffer der Station ankommt.

E10 ist die Entscheidung über die Route, die das FTF zum nächsten Ziel nimmt. Hierbei werden sowohl Transportaufträge zwischen den Prozessmodulen als auch zwischen den Ressourcenmodulen auf verschiedenen Ebenen berücksichtigt. Der Entscheidungsraum ist zweidimensional. Diese Entscheidung wird durch den Transportbefehl für ein Produkt P oder Modul M ausgelöst. **E11** ist spezifisch für die Transportroute des FTF_{AZ} zum nächsten Ziel. Die Parameter für diese Entscheidung umfassen alle Fahrbahnen des Produktionssystems. Der Auslöser für diese Entscheidung ist ebenfalls der Transportbefehl für ein Produkt P oder Modul M innerhalb des Arbeitszentrums AZ bzw. Work Center WC. Beide Entscheidungen resultieren aus der Transportplanung.

E12 und **E13** betreffen die Auswahl eines Transportauftrags für den Transport der Produkte und Module. Wenn ein FTF belegt ist, kann es auf eine Warteliste gesetzt werden, um die kontinuierliche Bearbeitung in der Ressource oder Station sicherzustellen, ohne auf verfügbare FTFs oder FTF_{WCH} warten zu müssen. Wenn ein Transportauftrag abgeschlossen ist, muss die Entscheidung getroffen werden, ob ein weiterer Transportauftrag angenommen wird. Das veränderungsfähige verschachtelte CPMPS wird in seine ebenen- und systemspezifischen Steuerungsentscheidungen zerlegt und ist Grundlage für die weitere Agentensystementwicklung.

Tabelle 3.1: Identifizierte Entscheidungen und Steuerungsaufgaben

ID	Ebene	Steuerungsaufgabe	Auslöser	Entscheidungsraum
E1	L3	Freigabe Auftragsvolumen AL_3	Erreichung des Parameters einer Freigaberegeln	Anzahl und Zeitpunkt der Menge an freizugebenden Aufträgen AL_3
E2	L3	Allokation (Scheduling) Jobs AL_2 & Station S	Auftragsfreigabe AL_3 , Optimierungspotenzial erkannt	Alle Stationen S mit Fähigkeit zur Fertigung des Modulauftrags AL_2 für Produkt P und Produktauftrag AL_3
E3	L2	Allokation (Scheduling) Operation AL_1 & Ressource R	E2 abgeschlossen, E4 abgeschlossen	Alle Ressourcen R mit Fähigkeit zur Fertigung des Teilauftrags AL_1 für Modul M und Modulauftrag AL_2
E4	L1	Reihenfolgebildung (Dispatching) AL_1	Eingang Produkt P oder Auftrag AL_1 beendet	Auswahl des nächsten Auftrags AL_1 für Modul M auf Ressource R
E5	L1	Kapazitätsanpassung (Dispatching) R	AL_3 für Modul M auf R beendet oder Ressourcenstörung R	Auswahl der nächsten Ressource R zur Bearbeitung von Folgeauftrag AL_1 (Dispatching)
E6	L3, L2, L1	Optimierungsmaßnahme einleiten	Ressourcenstörung R, Verspätung P oder M für Ausführung AL_1 oder AL_2 , Prozessabweichung erkannt	Ebenenspezifische Reihenfolgebildung (E4), Kapazitätsanpassung (E5) oder Auftragsfreigabe (E1)
E7	L2	Allokation (Dispatching) AL_1 & TR	E5 bzw. E6 abgeschlossen	Alle FTF mit der Fähigkeit zum Transport des Bauteils zur Folgeressource

ID	Ebene	Steuerungsaufgabe	Auslöser	Entscheidungsraum
E8	L2	Transport-ressourcenkapazitätsanpassung	Eingang Produkt P in $B_{in, S}$	Wahl des nächsten FTF für den Transport in Station S zur nächsten Ressource R
E9	L1	Transport-ressourcenkapazitätsanpassung	Eingang Produkt P in $B_{out, S}$	Wahl des nächsten FTF für den Transport zur nächsten Station S
E10	L1	Routenauswahl	Eingang Transportauftrag	Wahl der Route zur Folgeressource (frei)
E11	L2	Routenauswahl	Eingang Transportauftrag	Wahl der Route zur Folgestation (Bahn)
E12	L3	Transportauftragsreihenfolgebildung	Transportauftrag beendet	Wahl des nächsten Transportauftrags aus Warteliste des Arbeitszentrums AZ
E13	L2	Transportauftragsreihenfolgebildung	Transportauftrag beendet	Wahl des nächsten Transportauftrags aus Warteliste der Station S

Eine Übersicht der Entscheidungen sowie Informationen zur sequenziellen Ausführung ist im Entscheidungsmodell in Abbildung 3.6 dargestellt. Ebenfalls sind dort Informationen zu den Ereignissen, Informationsquellen sowie dem Typ der Verbindungen zwischen Entscheidungen und Ereignissen enthalten. Zusätzlich werden die Entscheidungen hinsichtlich der oben ausgeführten Ebenen unterschieden. Dieses initiale Entscheidungsmodell zeigt die zu treffenden Entscheidungen, welche benötigt werden, um eine grundlegende Ausführung der Produktion für das vorhandene Produktionssystem durchführen zu können.

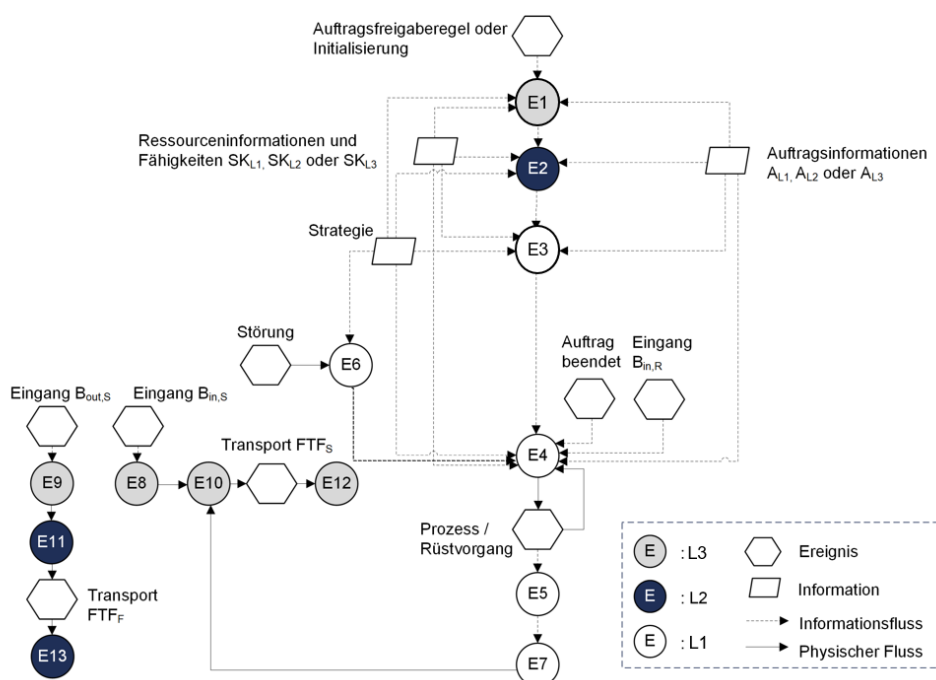


Abbildung 3.6: Grundlegendes Entscheidungsmodell der Probleminstanz

Das Ergebnis der Analyse der notwendigen Steuerungsentscheidungen ist das ebenenübergreifende Entscheidungsmodell mit Informationsbedarfen und Abhängigkeiten. In dem Modell wurden die Steuerungsaufgaben auf Basis der Entscheidungscluster nach Mayer et al. mit den grundlegenden Aufgaben der Produktionssteuerung nach Wiendahl abgeglichen. Darauffolgend konnten Abhängigkeiten zwischen einzelnen Entscheidungen, welche über die zuvor erwähnten lokalen Entscheidungsregeln hinausgehen, identifiziert werden. Dieses Entscheidungsmodell dient als Grundlage für die Erstellung des Agentenmodells im folgenden Abschnitt.

3.2 Agentenmodell

Das Konzept der Holone ermöglicht den Einsatz von Agenten in semi-heterarchischen Architekturen für eine heterarchische als auch eine hierarchische Steuerung. Agenten bieten eine Möglichkeit, Entscheidungen auf verschiedenen Hierarchieebenen miteinander zu verknüpfen.

Zur Erstellung des Agentenmodells werden Regeln zur Gruppierung angewandt. Die erste Regel zielt auf die Gruppierung von Agenten ab, die gemeinsame Steuerungsschnittstellen haben. Dies verhindert, dass verschiedene Agenten widersprüchliche Entscheidungen treffen, die zu Konflikten führen könnten. Die zweite Regel erweitert die erste, indem sie verlangt, dass Zustandsänderungen derselben Systemkomponenten zusammengefasst werden, um redundante Entscheidungen zu vermeiden. Dies hilft, intensive Interaktionen und den Kommunikationsaufwand zwischen Agenten zu reduzieren. Die dritte Regel zielt darauf ab, Entscheidungen zusammenzuführen, wenn sie stark voneinander abhängig sind. Dadurch wird vermieden, dass ein Agent zu stark in die Problemlösung eines anderen parallel arbeitenden Agenten eingreift [LK15]. Die Entscheidungsaufgaben innerhalb des CPMPS werden nach den DACS-Prinzipien gruppiert, um die Umsetzung agentenbasiert zu ermöglichen. Anschließend kann das Entscheidungsmodell gegebenenfalls angepasst werden. In diesem Abschnitt liegt der Fokus auf der Gruppierung und Definition der Agenten. Dabei werden verschiedene Regeln angewandt, um die modulare hierarchische Struktur des Produktionssystems in der Steuerungsstruktur abzubilden. In diesem Schritt werden ebenfalls Prinzipien der Referenzarchitektur ADMARMS nach [FR15] verwendet. Hierbei werden durch die Abhängigkeiten der Entscheidungen auch Beziehungen zwischen den Agenten und Holonen deutlich, welche im späteren Verlauf im Interaktionsmodell ausgestaltet werden. Ergebnis der Kapselung ist ein Agentenmodell, welches die Agenten für eine hybride Steuerungsarchitektur strukturiert.

3.2.1 Kapselung der Steuerungsentscheidungen zu Agenten

Die zuvor identifizierten Entscheidungen und Abhängigkeiten werden entsprechend der Produktionsstruktur und den Produktionselementen modular hierarchisch gekapselt. Hierbei erfolgen grundsätzlich zwei Kapselungsschritte. Zunächst werden abhängige Steuerungsentscheidungen in Agenten gekapselt. Dabei kann sich an [MCE21] orientiert werden. Anschließend werden die

Agenten selbst in Holone gekapselt. Dies ermöglicht eine modulare und hierarchische Organisation der Steuerung, die den Anforderungen des CPMPs gerecht wird.

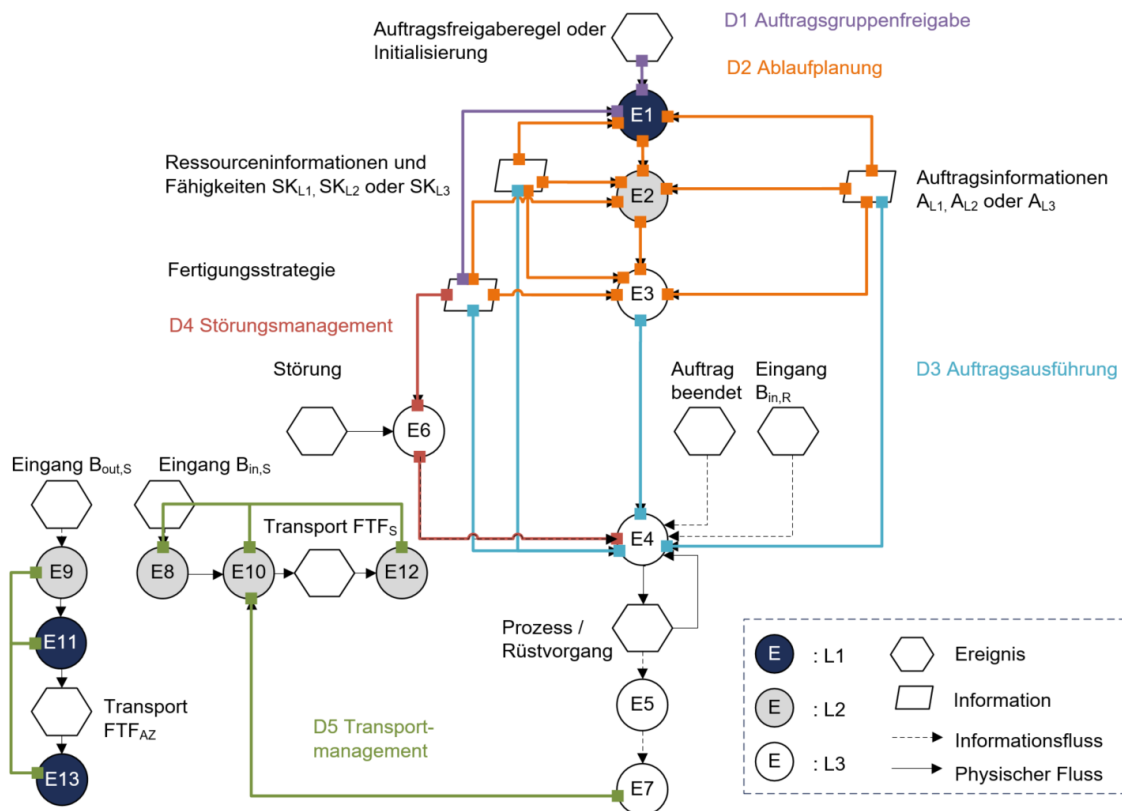


Abbildung 3.7: Identifizierte Abhängigkeiten der Entscheidungen

Aufbauend auf den Steuerungsentscheidungen werden deren Abhängigkeiten (engl. Dependencies) **D** identifiziert und in Agenten gekapselt. Durch die funktionale Dekomposition in Auftragsplanungs- und Auftragsausführungsaufgaben ergeben sich Abhängigkeiten und ein Koordinationsbedarf zwischen diesen.

D1 beschreibt die Abhängigkeit der Gruppierung und Freigabe der Aufträge E1 unter Berücksichtigung des internen Zielsystems des Systembereichs. Diese Abhängigkeit hat eine regulative Funktion auf den Bestand und leitet die Generierung neuer Soll-Werte durch die Auftragsplanung für das Gesamtsystem ein. Die Abhängigkeiten werden in einem Produktionsemitäsholon gekapselt (PEH).

D2 kapselt die Aufgaben des Holons zur Ablaufplanung der Subholone. Dazu werden die unterschiedlichen systemspezifischen Anpassungszeiten (Transport-, Rüst- und Prozesszeiten) für die jeweiligen Aufträge und deren Ausführung innerhalb des betrachteten Planungshorizontes der Produktionsemitäten auf unterschiedlichen Ebenen (Stationen S und Ressourcen R) berücksichtigt. Die Prozesszeiten der Ressourcen und Stationen für die Auftragsbearbeitung sind nur durch eine hierarchische Planung zu ermitteln. Soll Entscheidung E2 getroffen werden, so sind hierfür die Informationen aus E3 ebenfalls nötig. Die hierarchische verteilte Belegungs- und Reihenfolgeplanung nutzt die Struktur des Produktionssystems und führt eine Planung mit den prognosti-

zierten Auftragsbearbeitungszeiten aus der jeweils darunterliegenden Ebene aus. Dies hat Einfluss auf die Auslastung der Ressourcen und beeinflusst weitere Zielgrößen, da die Optimierungsalgorithmen nach einer festgelegten Zielgröße optimieren. Die Abhängigkeiten werden in einem Auftragsplanungsagenten (APA) gekapselt.

D3 ist die Abhängigkeit der Entscheidungen zur regelnden Auftrags- bzw. Prozessausführung. Neben den ereignisbedingten Abhängigkeiten, wie der Beendigung eines Auftrags und dem Eingang eines Produktes im Eingangspuffer, besteht die Abhängigkeit in der Auswertung der Zustandsinformationen und den verfügbaren Fähigkeiten zur Auftragsausführung. Für die Wahl des nächsten Auftrags sind Information des Ergebnisses des Ablaufplans der Ressourcen sowie der Auftrags- und Ressourceninformationen zur fähigkeitsbasierten Prozessausführung notwendig. In Verbindung mit der jeweils hinterlegten Strategie zur Verfolgung des internen Ziels des Systems wird der nächste verfügbare Auftrag gewählt und der Prozess zur Ausführung freigegeben (E4). D3 koordiniert als Auftragsausführungsagent (AA) die internen Abläufe des Holons wie die Ressourcenauftragsfolge und den Bestand des Eingangs- und Ausgangspuffers.

D4 ist die Abhängigkeit zur Maßnahmeneinleitung im Falle von Störungen eines Systems zur Alternativbelegung E6 mit den Entscheidungen E3, E4 und E5. Die Entscheidungen sind abhängig vom internen Ziel und der Priorisierung der logistischen Zielgrößen. Hierzu kann über die Striktheit der Planausführung die dezentrale Entscheidungskompetenz bei der Auftrags- und Prozessfreigabe beeinflusst werden. Diese Abhängigkeit betrifft die Auslastung der Ressourcen und die Durchlaufzeiten der Produkte. Ebenfalls ist der Bestand der zugehörigen Puffer von der gewählten Konfiguration abhängig. D4 kapselt die abhängigen Entscheidungen des Störungs- oder Abweichmanagements.

D5 betrifft die Abhängigkeiten der Entscheidungen der Transportsteuerung E8 bis E12. Bei der Wahl des FTF nach dem Eintreffen eines Auftrags (E8 und E9) stehen mehrere Stationen S und Ressourcen R zur Auswahl. Sobald ein Produkt in den Ausgangspuffer einer Station oder Ressource gelangt, wird ein Transportauftrag für ein FTF benötigt. Dabei beeinflusst dies direkt die tatsächlichen Transportzeiten und die transportbedingte Auslastung der FTF sowie die Pufferstände der jeweils ausführenden Entität. Die Entscheidung, welches FTF für welchen Transportauftrag gewählt wird, hat direkten Einfluss auf die Wahl der jeweiligen Route (E10 und E11), da sie vom Ziel abhängig ist. Diese identifizierten Abhängigkeiten signalisieren den Kommunikationsbedarf, da es in diesem Fall dazu kommen kann, dass nicht nur die lokalen Entscheidungsregeln, sondern die Informationen anderer Entscheidungen Einfluss auf die eigene Entscheidung haben. Die identifizierten Abhängigkeiten sind in Abbildung 3.7 zu sehen. Die FTF unterscheiden sich durch ihren Einsatz als fahrbahngebundenes FTF der L2-Ebene oder als freies FTF im Sinne einer Handhabungseinheit auf L3-Ebene. Die Puffer der Produktionsentitäten zur Materialablage stellen eine Entkopplung der beiden technischen Ebenen dar. Die FTF unterscheiden sich durch ihren Einsatz als fahrbahngebundenes FTF der L2-Ebene oder als freies FTF im Sinne einer Handhabungseinheit auf L3-Ebene. Die Puffer der Produktionsentitäten zur Materialablage stellen eine Entkopplung der beiden technischen Ebenen dar. Die lokalen Entscheidungsregeln dieses

Clusters können durch ihre Verbindung mit den Puffern die Entscheidungen zur Bestandsführung des Systems beeinflussen.

Zur Unterstützung der Entscheidungsfindung werden die statischen und ereignisorientierten Produktions-, Prozess- und Ressourceninformationen zentral gepflegt. Die ereignisorientierte Dokumentation ermöglicht den Agenten strukturierten Zugriff auf zustandsaktuelle Informationen zu Prozess, Produkt und Ressource. Die Steuerungsentscheidungen basieren auf der Interaktion von Produkt, Prozess und Ressource. Aufgrund der intensiven Datenhandlungsaufgaben wird ein Datenagent eingeführt, der die Pflege und Strukturierung der Daten der Assets (z.B. über eine VWS) übernimmt. Es werden prozess-, produkt- und ressourcenbezogene Auftragsinformationen gepflegt.

Prozessbezogene Auftragsinformationen zur Ausführung von Prozessschritten. Dabei werden die Prozesszeiten hierarchisch aggregiert. Die entitätsbezogenen Prozessfähigkeiten werden mit den granularen Prozessbedarfen zur Auftragsbearbeitung gematcht und liefern ressourcenspezifische Ausführungsmöglichkeiten für einen Auftragsstyp.

Produktbezogene Auftragsinformationen zur hierarchischen Strukturierung von Produkt, Modul und Teil. Diese werden Bottom-Up aggregiert zur produktbezogenen Auftragsbearbeitung. Der Vorranggraph mit möglichen Produktionsauftragsfolgen wird über den Datenagent gepflegt.

Ressourcenbezogene Auftragsinformationen der jeweiligen ebenenspezifischen Entität. Die Fähigkeiten werden entitätsbezogen aggregiert. Die Fähigkeiten einer untergeordneten Entität definieren die Fähigkeit einer hierarchisch höheren Entität.

Tabelle 3.2: Definition der Agenten und Beschreibung der Aufgaben

Abhängigkeit	Definition der Agenten	Beschreibung der Aufgaben
D1	Produktionsentitätsholon (PEH)	Das Holon kapselt die Funktionen der Subagenten und aggregiert die Informationen des Systembereichs. Er erzeugt die Aufträge für den Systembereich und übernimmt die bestandsregulierende Auftragsfreigabe sowie die Verfügbarkeit der Fähigkeiten zur Auftragsbearbeitung der Produktionsentitäten im Systembereich.
D2	Auftragsplanungsagent (APA)	Dieser Agent führt die fähigkeitsbasierte hierarchische Ablaufplanung für die Aufträge der nächsttieferen Ebene aus
D3	Produktionsentitätsauftragsausführungsagent (PAA)	Dieser Agent führt die geplanten Aufträge aus und gibt Updates über Auftragsbearbeitungszustand und Prozessergebnis an den Datenagenten.
D4	Transportentitätsauftragsausführungsagent (TAA)	Der Transportauftragsausführungsagent ist mit den FTF verbunden und wählt die Route für den Transportauftrag

D5	Korrekturagent (KA)	Dieser Agent erkennt Abweichungen in der Auftragsausführung und reagiert auf diese mit geeigneten Maßnahmen und leitet diese an den Ausführungsagenten weiter.
D6	Datenagent (DA)	Der Datenagent verwaltet die auftragsrelevanten Prozess-, Produkt- und Ressourceninformationen

3.2.2 Holonische Kapselung der Agenten

Die Agenten werden zur Bearbeitung von Steuerungsentscheidungen rekursiv holonisch gekapselt gemäß der Steuerungshierarchieebenen.

Das Cluster **C1** kapselt die Agenten der L3-Ebene im **Work Center Holon (WCH)** mit dem Planungshorizont für den Gesamtauftrag. Die Auftragsplanung wird durch die Bildung der Auftragsgruppen angestoßen. Unter Berücksichtigung der fähigkeitsbasierten aggregierten Informationen über Bedarfe und Kapazitäten sollen die Entscheidungen zur verteilten Planung für Soll-Vorgaben mit mittelfristigem Planungshorizont koordiniert und der Gesamtsystembestand Top-Down überwacht werden.

Das **Stationsholon (SH)** organisiert die Auftragsbearbeitung der Modulaufträge auf L2-Ebene. Als Teil der verteilten hierarchischen Auftragsplanung besteht ein hierarchischer Koordinationsbedarf in die übergeordnete L3-Ebene, aber auch in die untergeordnete L1-Ebene. Die Bearbeitungsoptionen zur Fertigung von Modulaufträgen werden zur Auftragsplanung Top-Down vom WCH an das SH angefragt. Der Entscheidungsraum wird vom C1 Cluster getrennt, und es besteht die Möglichkeit einer rekursiven und subsidiären Entscheidungsfindung zur Optimierung der Auftragsplanung und Auftragsausführung innerhalb des Systembereichs. Dabei wird innerhalb der Stationsholone eine geeignete Ressource ausgewählt unter Berücksichtigung der Veränderungsfähigkeiten der gekapselten Transport- und Fertigungsressourcenholone und ggf. eine Anpassung des globalen Ablaufplans.

C3 ist das **Control Device Holon (CDH)** und führt die Aufträge auf den Fertigungs- und Transportressourcen aus. Es wird auch die Abkürzung RH für Ressourcenholone oder TH für Transportholone verwendet. Die Agenten des RH besitzen eine stärkere Autonomie und führen die Entscheidungen mit lokal kurzfristigem Entscheidungshorizont aus. Dazu gehört die dezentrale Verfügbarkeitsprüfung und Auftragsauswahl. Die Einleitung von regulativen Maßnahmen im Falle einer Abweichung wird hier angestoßen. Dies kann eine dezentrale Reihenfolgeoptimierung (E4) sein, welche die Ausführung eines alternativen Prozessauftrags zur Folge hätte. Die Einleitung einer regulativen Maßnahme im Fall einer Abweichung findet hier statt (E6). Ebenso der Bedarf für ein notwendiges FTF (E7) wird von der prozessausführenden Ressource unter Berücksichtigung der aktuellen Prozesszeit angestoßen. Entschieden wird dies übergeordnet koordinativ durch das PMH. Innerhalb dieses Clusters werden die kurzfristigen Entscheidungen auf L1-Ebene ausgeführt. Entscheidungen müssen in diesem Fall in kurzfristigen Zeitabständen getroffen werden können. Die Entscheidung zur Ausführung wird reaktiv auf L1-Ebene unter Berücksichtigung zu

erwartender Auftragsbedarfe der Entitäten ausgeführt. Eine zusammenfassende Übersicht der identifizierten Holone ist in nachfolgender Tabelle 3.3 dargestellt.

Tabelle 3.3: Holonische Gruppierung der Agenten

Cluster	Holon	Beschreibungen
C1	Work Center Holon (WCH)	Dieses Holon startet die zentrale selbstoptimierende Auftragsplanung des CPMPs mit Auftragsfreigabe für $L3$ und startet die Angebotsanfrage für die Auftragsplanung der Auftragsgruppen für die Prozessmodule auf $L2$.
C2	Station Holon (SH)	Das Holon führt innerhalb der Prozessmodule die hierarchische Ablaufplanung für die Aufträge A_{L1} für Ressourcen R durch und koordiniert ggf. Entscheidungen für Ressourcenholone.
C3	Control Device Holon (CDH) / Ressourcenholon (RH) / Transportholon (TH)	Das Holon führt die geplanten Aufträge als Prozesse auf den Fertigungs- oder Transportressourcen als 1:1 Beziehung aus und reagiert auf Störungen mit lokalen Maßnahmen.

Die Agenten und Holone ermöglichen die Entscheidungsfindung zur Auftragsplanung und Auftragsausführung innerhalb des Produktionssystems auf unterschiedlichen Hierarchieebenen. Die Beziehungen der Holone werden nachfolgend erläutert.

Die Holarchie mit Verantwortungshoheiten von Holonen über andere Holonen realisiert die komplexitätsreduzierende Auftragsplanung mithilfe rekursiver und subsidiärer Verhaltensmuster. Diese Holarchie umfasst alle darunterliegenden Holone und deren Agenten innerhalb des Systemraums. Zwischen den Ressourcen existieren ebenfalls Hierarchien. Das TH besitzt hauptsächlich hierarchische Beziehungen zu den anderen RH und wird von parallelen Entitäten derselben Ebene gesteuert. Er ist für den Transport der Produkte P , Module M und den Zwischenerzeugnissen zwischen den Ressourcen zur richtigen Zeit über die kürzesten Routen zu den richtigen Zielen zuständig. Ebenfalls sorgt er für eine hohe Auslastung der FTF, da er Transportaufträge der Warteliste auswählt. Dementsprechend sind die Holone der folgenden Ebene die SH. Ein TH existiert für alle FTF auf $L1$ -Ebene und wird als Teil eines SH oder WCH gekapselt.

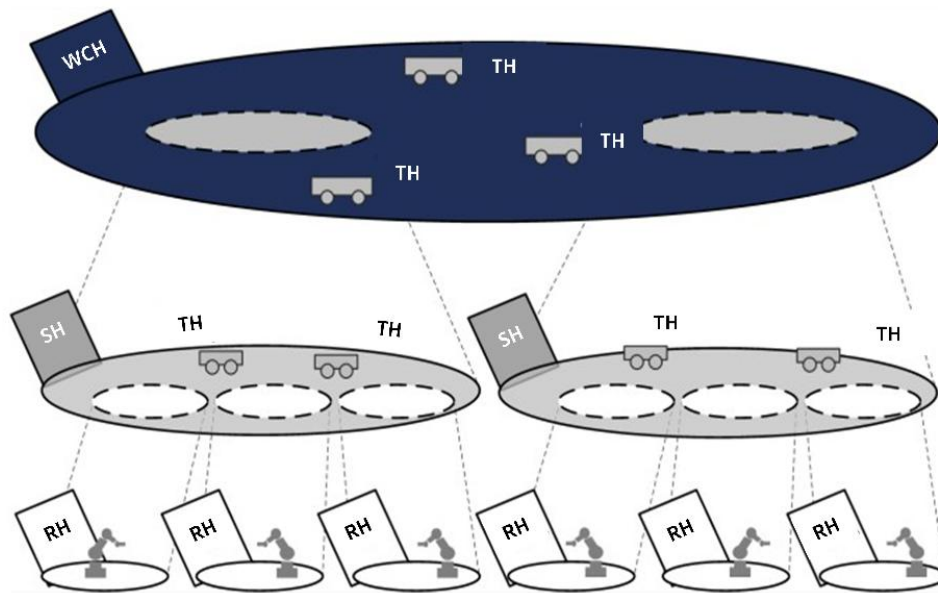


Abbildung 3.8: Holonisches Agentensystem und Holarchie

Der SH beinhaltet entsprechend der verfolgten Holarchie die CDH zur Ausführung der Fertigungs- und Transportaufträge. Jeder SH beinhaltet seine ihm zugeordneten Ressourcen R und besitzt die dazugehörige Verantwortungshoheit. Durch die Belegungs- und Reihenfolgeplanung (E3) beeinflusst er direkt die Auftragsreihenfolge und Zuordnung der ihm zugeordneten Ressourcen. Der RH selbst ist das Holon zur Auftragsausführung auf Prozessebene. Diese holonische Steuerungsarchitektur auf Basis der gruppierten Entscheidungen und unter Rücksichtnahme auf die Designprinzipien für rekonfigurierbare und agentengesteuerte Produktionssysteme ist in Abbildung 3.8 dargestellt. Die einzelnen Ebenen entsprechen den Ebenen L1, L2 und L3 und zeigen die Verantwortungsbereiche der jeweiligen Holone. Das WCH ist für die oberste L3-Ebene zuständig, überlässt die Verantwortung für Teilbereiche hingegen subsidiär dem SH aus L2-Ebene, welche wiederum Verantwortung an CDH bzw. RH auf L1-Ebene übergeben. So kann die Anforderung nach Rekursivität und Subsidiarität erfüllt werden.

Das holonische Agentenmodell repräsentiert ein modular hierarchisches System mit fraktalem Design. Die Holone der jeweiligen Ebene koordinieren die Auftragsbearbeitung selbstorganisiert und besitzen einen selbstähnlichen inneren Aufbau. Die Holone operieren als unabhängige Teilsysteme mit dezentraler Entscheidungskompetenz.

3.3 Interaktionsmodell

Dieser Abschnitt modelliert die generische Interaktion des holonischen Agentensystems. Hierbei werden die Beziehungen der Agenten und Holone aus Abschnitt 3.2.2 berücksichtigt. Ergebnis

dieses Abschnittes ist ein generisches Entwurfsmuster, welches die Entscheidungsfindung innerhalb des komplexen Systems für den ebenen- und systemübergreifenden Einsatz ermöglicht. Die Interaktionen der internen Holonen und Agenten und die ebenenspezifische Ausgestaltung dieser werden darauf aufbauend als Teil der Implementierung in Abschnitt 4.2 erläutert. Die grundlegende interholonische Interaktion ist durch die Holarchie bereits definiert. Hierbei ist zu beachten, dass durch die Holarchie ein übergeordnetes Holon die Verantwortung für darunterliegende Agenten oder Holone trägt und durch Festlegung des Handlungsrahmens die Handlungen maßgeblich beeinflusst, ohne konkrete Verhandlungen zu vollziehen. Innerhalb des Produktionssystems wird ein Bedarf in Form von Aufträgen erzeugt, der mit der Kapazität der Entitäten des Produktionssystems abgearbeitet werden soll. Im Folgenden werden die Interaktionen zur generischen ebenenübergreifenden Entscheidungsfindung definiert.

Die Interaktion definiert das Steuerungsverhalten der hierarchisch verteilten Systeme. Auf Grundlage der identifizierten Steuerungsentscheidungen und Abhängigkeiten wird die Interaktion der Agenten definiert. Um eine skalierbare Steuerung zu ermöglichen, wird ein generisches Interaktionsmodell für eine rekursive Auftragsplanung und subsidiäre Auftragsausführung definiert. Das Entwurfsmuster von Ressourcenagenten in Kombination mit Feldsteuerungsagenten und dem *Delegate MAS* kann als Orientierung für eine holonische Steuerung über mehrere Systemhierarchien mit verschachtelten Entitäten verwendet werden. Im Beispiel des Ressourcenagenten ist dies die Verknüpfung mit der Feldsteuerungsebene. Für die vorliegende Arbeit wird das Prinzip für einen generischen Ebenenübergreifend erweitert.

3.3.1 Intraholonische Interaktion

Die Interaktion kann für jedes System des CPMPs nach dem selbstähnlichen fraktalen Prinzip verwendet werden. Die Interaktion ist in „Control“, „Command“ und „Inform“ unterteilt. Die horizontale und vertikale Kommunikation innerhalb des Systems wird über den Datenagent realisiert, der als Kommunikationsbroker zwischen den Agenten agiert und die Informationen strukturiert. Das Produktionsentitätsholon überwacht den Systembestand und die in seinem Systembereich verfügbaren Fähigkeiten der verschachtelten Entitäten. Der Auftragsausführungsagent ordnet eine fähigkeitsbasierte Auftragsplanung an bei entsprechender Anfrage von einem hierarchisch höheren Holon. Der Korrekturagent unterscheidet zwischen einer selbstoptimierenden Planung oder einer adaptiven Ausführung innerhalb des Systembereichs. Ausgelöst werden kann dies z. B. durch die Störung der Ausführung auf einer untergeordneten Entität und einem damit verbundenen Ausfall einer Fähigkeit bzw. eines Skills oder einer Prozessabweichung. Extern würde z. B. eine Anfrage zur Optimierung gestellt werden. Bei der Entscheidung zur globalen Optimierung werden andere Systeme miteinbezogen, bei der adaptiven Anpassung wird innerhalb eines kurzfristigen Entscheidungshorizontes eine kurzfristigere Optimierung vorgenommen. Das kapselnde Produktionsentitätsholon besitzt einen Ausführungsagenten mit entsprechendem Wissen zur Ausführung des Auftrags auf den Entitäten innerhalb des Systemraums. Die Auftragsfolge des Systembereichs wird entweder durch den Plan des Auftragsplanungsagenten bestimmt

oder durch lokale Entscheidungsregeln angepasst. Der Korrekturagent erkennt den Anpassungsbedarf im Störfall und informiert den Ausführungsagenten über den Bedarf einer globalen oder lokalen Anpassungsmaßnahme. Die Transportaufträge werden reaktiv in Abhängigkeit der zustandsaktuellen Fertigstellungszeitpunkte der Fertigungsaufträge durch die Produktionssubholone heterarchisch angefordert. Das Interaktionsmuster beschreibt den grundlegenden intraholonischen Steuerungsablauf zur selbstoptimierenden Auftragsplanung und -ausführung. Das Entwurfsmuster für die Optimierung modular hierarchischer CPMPs kombiniert Ansätze der Entwurfspattern für eine Ressourcenallokation und Steuerung mit einem Service Interface Layer. Der SIL ist angelehnt an die Verwendung in HMS nach [Ga16] und wird für die vorausschauende Auftragsallokation und Reihenfolgeoptimierung innerhalb eines Systembereichs verwendet.

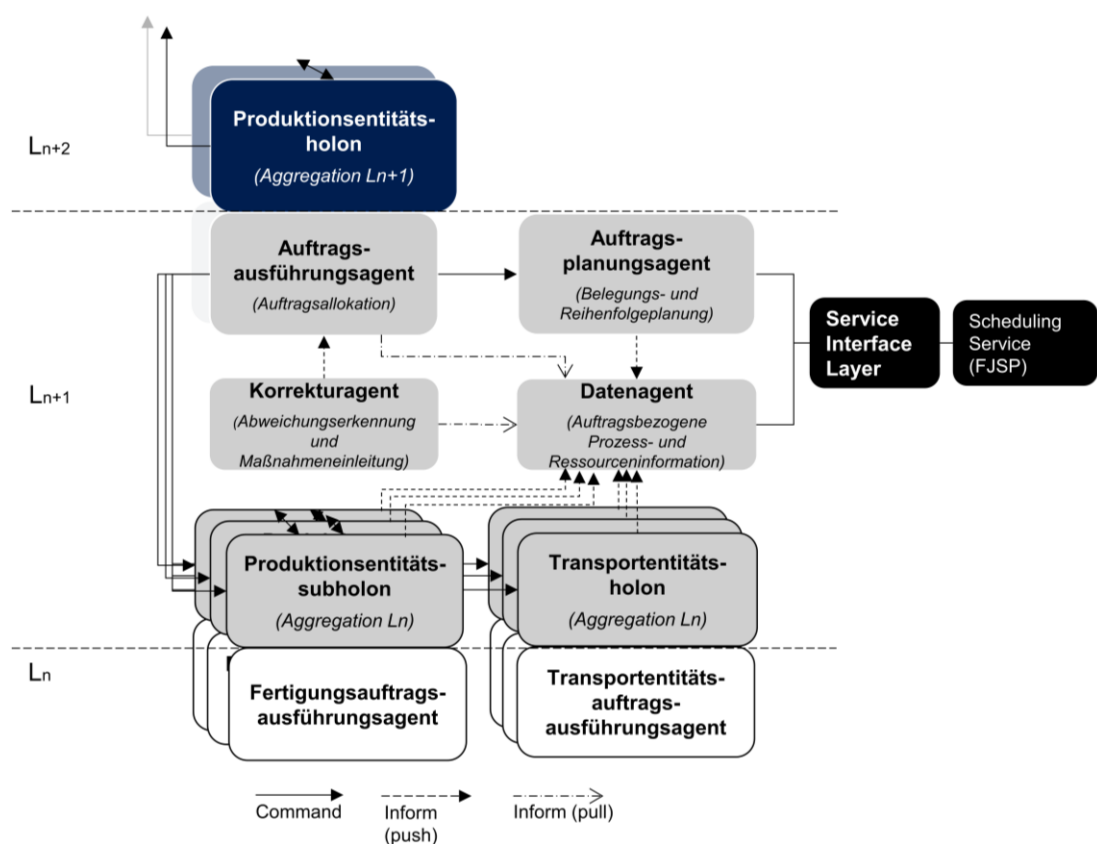


Abbildung 3.9: Intraholonisches Interaktionsmuster zur generischen Auftragsplanung und Auftragsausführung

3.3.2 Interholonische Interaktion

Die Interaktion der Holone definiert das Steuerungsverhalten des Gesamtsystems über die Ebenen. Dazu wird eine interholonische Interaktion der komplexitätsreduzierenden Auftragsplanung und zustandsaktuellen Steuerung entworfen. Diese legt den Grad der Autonomie der Holone und der erlaubten Anpassungsfähigkeit fest. Die Interaktionen I werden nachfolgend beschrieben.

I0 ist eine Interaktion des WCH mit sich selbst. Auf Basis eines Ereignisses, das ein Auftragsfreigabekriterium erfüllt, gibt der WCH die Aufträge AL_3 frei. Mit **I1** werden die für die Auftragsbearbeitung benötigten Fähigkeiten auf Basis der Auftragsinformationen abgeleitet. **I2** beschreibt das Abgleichen der zur Verfügung stehenden Fähigkeiten SKL_2 der SH und die darauf basierende gleichzeitige Abfrage zu Prozesszeiten für die Teilaufträge AL_2 der Aufträge AL_3 . Dadurch wird das komplexe Optimierungsproblem in mehrere Teilprobleme aufgeteilt. Die Station generiert nach interner Berechnung ein Angebot für diese Aufträge AL_2 auf Basis der untergeordneten Ressourcen, welche zuvor über **I1** ihre Ressourceninformation zur Auftragsbearbeitung über den Datenagenten mitgeteilt hat. Das Angebot ist zu diesem Zeitpunkt aggregiert, da noch kein bestätigtes Angebot über den tatsächlichen Start- und Endzeitpunkt der Aufträge beim SH vorliegt. Dies ist abhängig von den Angeboten der anderen Stationen und wird über den WCH koordiniert..

Nach der internen aggregierten Ablaufplanung innerhalb des SH wird mit **I4** ein Angebot für die angefragten Angebote der Auftragsgruppen mit den Aufträgen AL_2 abgegeben. In **I5** wird dieses nach interner Optimierung mithilfe einer Ablaufplanung des WCH entweder angenommen oder abgelehnt. Ergebnis des Plans ist die entsprechende Allokation auf Stationen für die Ausführung der jeweiligen Aufträge. Die Bestätigung der Allokationen und Ausführungszeitpunkte erfolgt parallel für mehrere Stationen und Aufträge. Dazu werden in **I6** die entitätsindividuellen Fähigkeiten als Skills mit ihren Prozesszeiten zur Auftragsbearbeitung berücksichtigt. Da das zuvor getätigte Angebot nur eine aggregierte Bearbeitungszeit des Loses ist, werden in **I7** die Aufträge AL_2 nach der Bestätigung des WCH mit dem tatsächlichen Zustand der Station und deren gekapselten Ressourcen mit den Start- und Endzeiten angepasst. Dabei müssen die technischen Restriktionen der Aufträge AL_1 beachtet werden, die in den Aufträgen AL_2 enthalten sind. Zwar ist die Reihenfolge der Aufträge AL_2 unter Beachtung der technischen Vorranggraphen durch die Reihenfolgebildung innerhalb des Planungshorizontes des WCH berücksichtigt, allerdings erfordert die indessen nicht mehr aggregierte Planung des SH eine Absprache mit den anderen SH. Dafür existiert **I9**, welche parallele SH über die bestätigten Start- und Endzeitpunkte der subsidiären Auftragsplanung aus **I8** informiert, sodass die parallelen SH nun einen Abgleich mit den eigenen individuellen Start- und Endzeiten der Gruppen an Aufträgen AL_2 durchführen können. Die horizontale Interaktion ermöglicht die Optimierung des zuvor lediglich aggregierten Ablaufplans.

Nach erfolgreicher Optimierung aller Stationen werden die Aufträge an die Ressourcen zur Bearbeitung gegeben mit **I10**. Die Ausführung der einzelnen Aufträge AL_3 beginnt, sobald ein Auftrag abgearbeitet ist und kein weiterer Auftrag auf der Ressource für dieses Produkt geplant ist. Der RH initiiert mit **I11** die Weitergabe zur nächsten Ressource. Der TH plant mit **I12** die Route und führt diese mit **I14** aus. Falls Transportaufträge auf die Warteliste gesetzt wurden, werden diese nach Beendigung des aktuellen Transportauftrags evaluiert und ggf. akzeptiert. Eine Störungsmeldung der ausführenden Holone wird mit **I13** hierarchisch weitergegeben an das Stationsholon. Zusätzlich enthalten ist eine zentrale Auftragsdatenverwaltung, welche die benötigten Auftragsinformation beinhaltet und Fähigkeiten ableitet.

Das Interaktionsmodell in Abbildung 3.10 zeigt den Ablauf des Steuerungsprozesses für die Holararchie auf Basis des Entscheidungsmodells und des holonischen Agentenmodells. Das Interaktionsmodell folgt dem grundlegenden Prinzip der hierarchischen Abfrage von Fähigkeiten und Angeboten zu Auftragsgruppen. Außerdem erfolgt eine horizontale Abstimmung zwischen Entitäten sowie eine lokale Ausführung der Aufträge auf unterster Ebene. Die fähigkeitsbasierte Interaktion wurde hiermit modelliert und es könnten in Zukunft weitere Ebenen und Systeme hinzugefügt werden, welche dem Prinzip von rekursiver Auftragsplanung und subsidiärer Auftragsausführung folgen. Eine fähigkeitsbasierte Koordination von Produkt, Prozess und Entität wird system- und ebenenspezifisch ermöglicht.

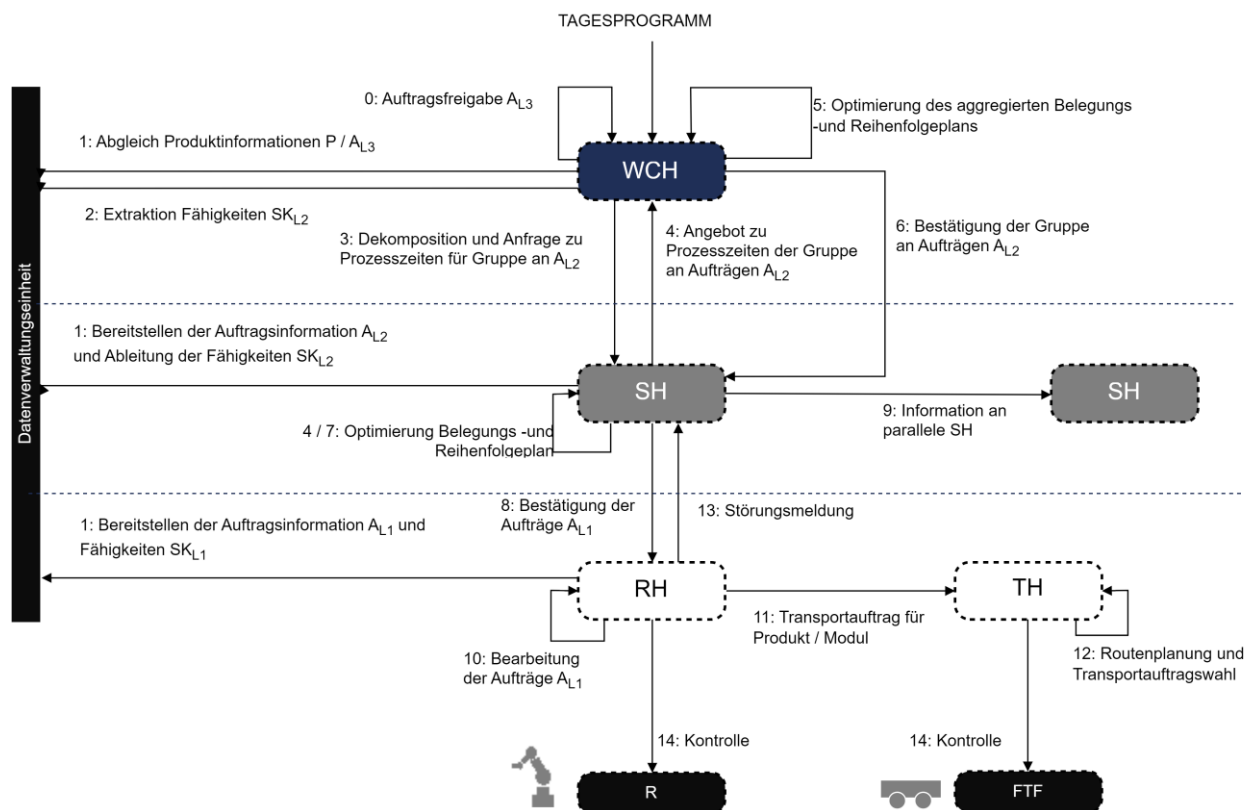


Abbildung 3.10: Interholonisches Interaktionsmodell der Auftragsplanung und Auftragsausführung

3.4 Optimierungsmodell

Das Optimierungsproblem eines veränderungsfähigen CPMPs kann, wie in Abschnitt 2.3.1 vorgestellt, mit einem erweiterten Flexible Job Shop Problem (FJSP) beschrieben werden. Für ein auftragsgesteuertes CPMPs soll eine Optimierung entsprechend der modular hierarchischen Struktur vorgenommen werden können. Dazu wird im Folgenden ein Optimierungsmodell derart formuliert, sodass eine system- und ebenenspezifische Ablaufplanung unter Berücksichtigung der Vorranggraphbeziehungen mit Produkt-, Prozess-, Ressourcen- und Routingflexibilität für mehrstufige Auftragsstrukturen vorgenommen werden kann. Das Problem wird so formuliert, dass es

als ebenenübergreifendes Modell eingesetzt werden kann. Grundsätzlich wird in Job Shop Scheduling Problemen zwischen Jobs und Arbeitsgängen unterschieden. Für die Betrachtung in einem auftragsgesteuerten CPMPS werden die Bezeichnungen entsprechend der Ebenen L3-L1 als Orders (Produktaufträge), Jobs (Modulaufträge) und Operationen/Arbeitsgänge (Prozessaufträge) definiert, die unterschiedliche ebenenspezifische Vorrangbeziehungen besitzen.

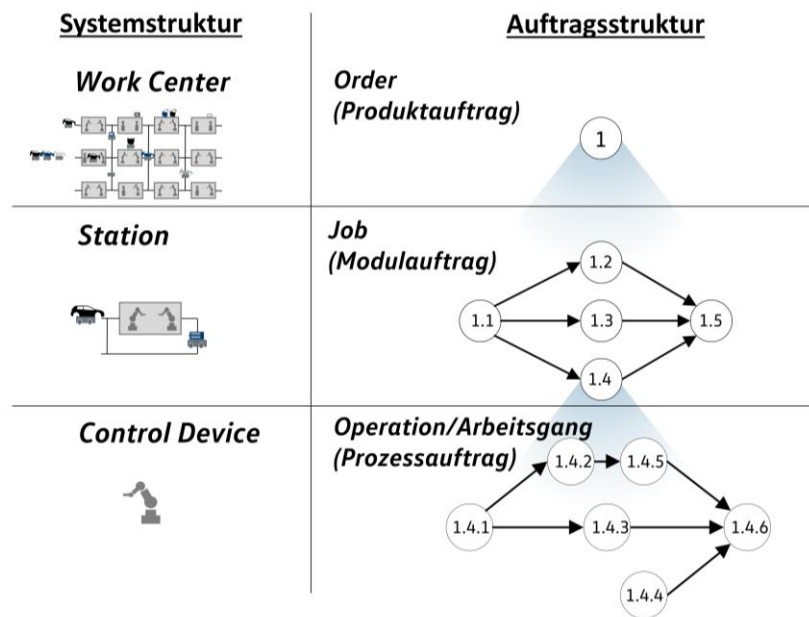


Abbildung 3.11: System- und Auftragsstruktur für die Reihenfolgeoptimierung

3.4.1 Spezifikation des erweiterten FJSP

Für die ebenen- und systemspezifische Auftragsplanung mit Berücksichtigung der operativen Veränderungsfähigkeiten wird ein erweitertes FJSP modelliert. In der Abbildung 3.12 wird das Prinzip der flexiblen Vorranggraphbeziehungen für erweiterte FJSP-SF für diese Arbeit verdeutlicht.

Die linke Darstellung in Abbildung 3.12 stellt den Sachverhalt des FJSP dar, der eine starre Sequenz von fünf aufeinander folgenden Arbeitsgängen desselben Auftrags definiert. In der rechten Darstellung der Abbildung 3.12 ist die Sequenz durch technische Vorranggraphbeziehungen festgelegt, die eine unterschiedliche Reihenfolge der Arbeitsgänge bei der Abarbeitung zulassen. Unter Beachtung der Vorranggraphbeziehungen ergeben sich bei der Aneinanderreihung der Arbeitsgänge 2, 3 und 4 nun sechs verschiedene Sequenzen zur Abarbeitung der Arbeitsgänge für einen Auftrag.

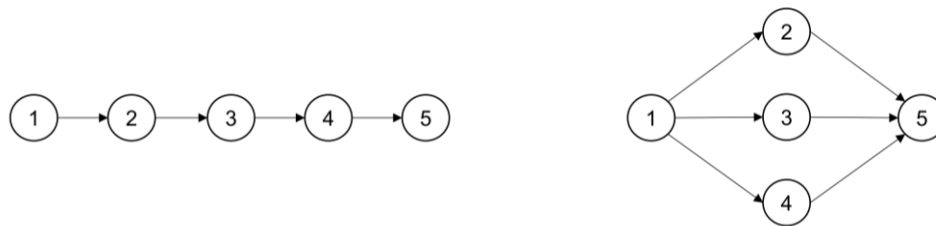


Abbildung 3.12: Starre Sequenz eines FJSP (links) und Vorranggraph des FJSP mit flexiblen Ablaufpfaden (rechts)

Für die Reihenfolgebildung in modularen Produktionssystemen kann weiterhin zwischen zwei Arten von Arbeitsgängen unterschieden werden. Die Montagearbeitsgänge bezeichnen die Prozessschritte, die das Primärwerkstück bearbeiten. Als Komponentearbeitsgänge werden Prozessschritte bezeichnet, die Zwischenerzeugnisse herstellen, die später mit dem Primärwerkstück verbunden werden. Für die Montagearbeitsgänge gilt die gängige Annahme in Scheduling-Problemen, dass zu jedem Zeitpunkt nur ein Kernarbeitsgang desselben Jobs in Bearbeitung sein darf [JP19]. Komponentearbeitsgänge hingegen können parallel zu Montagearbeitsgängen desselben Jobs auf einer anderen Ressource bearbeitet werden, solange dies nicht durch eine Vorrangbeziehung beschränkt wird.



Abbildung 3.13: Unzulässige (links) und zulässige (rechts) Vorrangbeziehung von Komponenten- zu Montagearbeitsgang

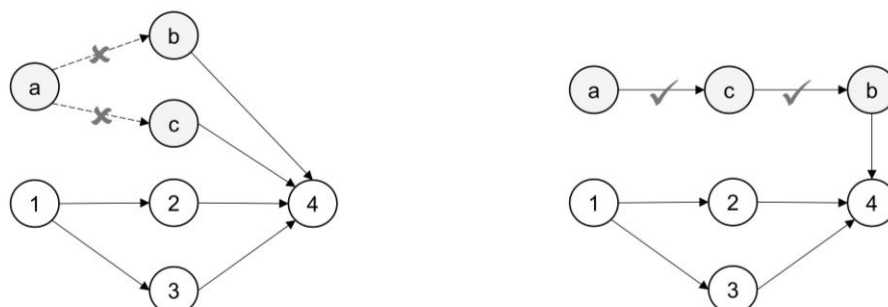


Abbildung 3.14: Unzulässige (links) und zulässige (rechts) Vorrangbeziehung zwischen Komponentearbeitsgängen

Abbildung 3.13 und Abbildung 3.14 veranschaulichen das Prinzip der Montage- und Komponentenarbeitsgänge und die zulässigen und unzulässigen Vorrangbeziehungen zwischen den Arbeitsgängen. Die Montagearbeitsgänge sind in weiß mit einer Zahl gekennzeichnet und die Sekundärarbeitsgänge in grau mit Buchstaben versehen. Ein Komponentenarbeitsgang ist per Definition nicht auf den Abschluss eines Montagearbeitsgangs angewiesen (siehe Abbildung 3.13 links). Ein Montagearbeitsgang kann allerdings erst nach Abschluss eines Komponentenarbeitsgangs begonnen werden (siehe Abbildung 3.13 rechts). Außerdem können Vorrangbeziehungen zwischen Komponentenarbeitsgängen existieren, allerdings sind dabei nur lineare Abfolgen vorgesehen (siehe Abbildung 3.14 rechts).

Unter der Voraussetzung, dass immer nur ein Primärarbeitsgang eines Jobs gleichzeitig in Bearbeitung sein darf, wird eine definierte Reihenfolge der Arbeitsgänge gesucht. Dies wird als Auftrags-Sequenz oder Auftragsreihenfolge definiert. Vorgabe für die Sequenzbildung ist die Berücksichtigung der Vorranggraphbeziehung des jeweiligen Produktes und das Matching mit Prozessfähigkeiten der Ressourcenmodule. Hinzu kommt die Dauer für die Ausführung der Prozessfähigkeit auf einer Ressource. Die Arbeitsgänge, die auf derselben Ressource bearbeitet werden können, werden als Ressourcen-Sequenz oder Ressourcenreihenfolge definiert. Das erweiterte FJSP wird in drei Teilprobleme zur Sequenzierung der Prozessschritte auf den Ressourcen eingeteilt. Das erste Teilproblem des erweiterten FJSP ist die mögliche Reihenfolgebildung der Arbeitsgänge für jeden Auftrag. Das zweite Teilproblem ordnet die Arbeitsgänge der Menge an Ressourcen zu, die in der Lage sind, die Arbeitsgänge mit den geeigneten Fähigkeiten (Capabilities) durchzuführen. Das dritte Teilproblem behandelt die Festlegung der Reihenfolge der Auftragsbearbeitung auf den möglichen Ressourcen unter Berücksichtigung der jeweiligen ressourcenindividuellen Skills mit ihren Prozess-, Rüst- und Transportzeiten.

Die folgenden Annahmen charakterisieren das Ablaufplanungsproblem erweiterter FJSP für die weitere Bearbeitung:

1. Der Ablaufplan erfordert ununterbrochene Bearbeitung für jeden Arbeitsgang (keine Unterbrechung erlaubt)
2. Alle Jobs sind zu Beginn des Planungshorizonts verfügbar
3. Die Bearbeitungszeiten der Arbeitsgänge auf den Ressourcen sind bekannt und können je nach Ressource variieren
4. Rüstzeiten sind abhängig von der Reihenfolge der auf derselben Ressource bearbeiteten Arbeitsgänge und ebenfalls bekannt
5. Es gibt Transportzeiten zwischen der Bearbeitung von Arbeitsgängen desselben Jobs auf verschiedenen Ressourcen, und diese Zeiten sind eindeutig für alle Ressourcenpaare festgelegt, unabhängig vom transportierten Job
6. Ressourcen oder Stationen können zu Beginn des Planungshorizonts für eine vordefinierte Zeit blockiert sein, bevor sie zur Bearbeitung von Arbeitsgängen wieder verwendet werden können

3.4.2 Mathematische Formulierung des erweiterten FJSP

Im Folgenden wird ein MILP-Modell vorgestellt, welches die Veränderungsfähigkeiten eines CPMPS in Bezug auf die Bearbeitungsreihenfolgen mithilfe eines erweiterten FJSP mit Sequenzflexibilität repräsentiert. Dabei werden die Flexibilitätsarten Ressourcen-, Produkt-, Prozess- und Routing- bzw. Sequenzflexibilität berücksichtigt. Ziel ist die Formulierung des Problems für einen Einsatz auf den verschiedenen Ebenen und in den holonisch gekapselten Systembereichen. Daher wird zunächst die grundsätzliche Wahl des Modelldesigns in Bezug auf die Entscheidungsvariablen erläutert. Es werden bei Job Shop Problemen hauptsächlich drei Arten der Modellierung von sequenzbestimmenden Binärvariablen für die Einplanung von Arbeitsgängen auf Ressourcen unterschieden. Diese wurden jeweils zuerst von Wagner [Wa59], Bowman [Bo59] und Manne [Ma60] formuliert, wie in Demir und Kürşat İşleyen zu finden ist [DK13]. Die ursprünglich von Wagner [Wa59] eingeführte Formulierung trifft die Annahme, dass eine fixe Anzahl an Positionen auf jeder Ressource verfügbar ist. Ein Arbeitsgang O_{kj} ist dann der Position l auf Ressource m zugewiesen, wenn $X_{kjl m} = 1$ gilt. Die Bezeichnung der Indizes in den Entscheidungsvariablen wurde der in dieser Arbeit verwendeten Notation angepasst, wie in Tabelle 3.4 dargestellt. Die von Bowman formulierten zeitindizierten Binärvariablen sagen aus, dass ein Arbeitsgang O_{kj} auf Ressource m zum Zeitpunkt t eingeplant ist, wenn $X_{k j m t} = 1$ gilt. Dieser Ansatz setzt zwei Annahmen voraus. Zum einen muss ein Planungshorizont $\mathcal{T}$ in diskrete Zeitpunkte $t \in \mathcal{T}$ unterteilt werden, zum anderen wird die Länge des Planungshorizonts als endlich und bekannt vorausgesetzt, da die Menge $\mathcal{T}$ zum Beginn der Planung definiert werden muss.

Die dritte Art der Variablen, die erstmals durch Manne beschrieben werden, formulieren die Vorgänger-Nachfolger-Beziehungen auf einer Ressource. Gilt $X_{k j h i m} = 1$, dann folgt Arbeitsgang O_{hi} direkt auf den Arbeitsgang O_{kj} auf Ressource m . Ein Vorteil dieser Formulierung besteht darin, dass sie keine Einschränkungen bezüglich der Zeitskala zur Folge hat und sie keine weiteren Anforderungen an eine Kenntnis der genauen Problemsituation stellt. Ein weiteres Argument zur Bevorzugung der Formulierung von Manne stellen die Ergebnisse einer Studie von Demir und Kürşat İşleyen dar [DK13]. Dort werden Versuche mit Modellen für das FJSP vorgestellt, die auf den drei verschiedenen Variablenformulierungen basieren. Dabei wird herausgestellt, dass das Modell von Özgüven, welches auf der Formulierung von Manne basiert, am besten bezüglich der Rechenzeit performt [DK13].

Aus den vorstehenden Gründen wird für das folgende Modell die Formulierung von Manne verwendet. Tabelle 4.1 enthält eine Übersicht über die verwendete Notation des Modells und klassifiziert die Symbole in Indexmengen, Parameter und Entscheidungsvariablen.

Tabelle 3.4: Notation zur Beschreibung des Optimierungsmodells

Indizes und Mengen	
$j \in \mathcal{J}$	Jobs $\mathcal{J} = \{1, \dots, \mathcal{J} \}$
$k \in \mathcal{K}$	Arbeitsgangtypen $\mathcal{K} = \{1, \dots, \mathcal{K} \}$
$m \in \mathcal{M}$	Ressourcen $\mathcal{M} = \{1, \dots, \mathcal{M} \}$
$m \in \mathcal{M}_k$	Ressourcen, die Arbeitsgänge vom Typ k bearbeiten können, $\mathcal{M}_k \subseteq \mathcal{M}$
Parameter	
p_{kjm}	Dauer des Arbeitsgangs bei Bearbeitung durch Ressource m
st_{khm}	Rüstzeit von Ressource m , wenn von Arbeitsgangtyp k auf h umgerüstet wird ($k \neq h$)
$tt_{mm'}$	Dauer des Transports zwischen den Standorten von Ressource m und Ressource m' ($m \neq m'$)
Parameter	
pc_{jkh}	$\left\{ \begin{array}{l} 1, \text{ wenn für Job } j \text{ der Arbeitsgang vom Typ } h \text{ erst gestartet werden darf, wenn der vom Typ } k \text{ beendet ist} \\ 0, \text{ sonst} \end{array} \right.$
Θ	hinreichend große Zahl
d_m	Freigabezeitpunkt von Ressource m
a_{kj}	$\left\{ \begin{array}{l} 1, \text{ wenn Arbeitsgang } O_{kj} \text{ ein Primärarbeitsgang ist} \\ 0, \text{ sonst} \end{array} \right.$
Entscheidungsvariablen	
C_{max}	Makespan
S_{kj}	Beginn der Bearbeitung von Arbeitsgang O_{kj}

$$\begin{aligned}
X_{kjhm} & \begin{cases} 1, \text{ wenn Arbeitsgang } O_{kj} \text{ direkt vor Arbeitsgang } O_{k'lr} \text{ auf Ressource } m \text{ bearbeitet wird} \\ 0, \text{ sonst} \end{cases} \\
Y_{jhk} & \begin{cases} 1, \text{ wenn für Job } j \text{ der Arbeitsgang vom Typ } h \text{ erst gestartet wird, wenn der vom Typ } k \\ \text{beendet ist} \\ 0, \text{ sonst} \end{cases} \\
G_{kjm} & \begin{cases} 1, \text{ wenn Arbeitsgang } O_{kj} \text{ durch Ressource } m \text{ ausgeführt wird} \\ 0, \text{ sonst} \end{cases}
\end{aligned}$$

Die Zielfunktion (1) minimiert die Makespan C_{max} des Ablaufplans. Der Betrag der Makespan ist definiert als der Fertigstellungszeitpunkt des zuletzt fertiggestellten Arbeitsgangs [JP19].

$$\text{Min. } C_{max} \quad (1)$$

Nebenbedingungen (2) stellen sicher, dass die Variable für die Makespan dem Fertigstellungszeitpunkt des zuletzt fertiggestellten Arbeitsgangs entspricht.

$$S_{kj} + \sum_{m \in \mathcal{M}_k} G_{kjm} \cdot p_{kjm} \leq C_{max} \quad \forall j \in \mathcal{J}; k \in \mathcal{K}: O_{kj} \in \Omega \quad (2)$$

Wie in Abschnitt 3.4.1 beschrieben, kann eine Ressource maximal einen Arbeitsgang zu jedem Zeitpunkt bearbeiten. Dazu wird eine Bearbeitungssequenz für jede Ressource gebildet. Wenn ein Arbeitsgang auf einer Ressource m bearbeitet wird, muss er auf dieser genau einen Vorgänger (3) und einen Nachfolger (4) haben. Damit dies für jeden Arbeitsgang gelten kann, ohne dass ein Zyklus entsteht, wird ein fiktiver Arbeitsgang O_{00} eingefügt, der jeweils den ersten und letzten Arbeitsgang auf einer Ressource symbolisiert. Es werden dazu die Mengen $\mathcal{J}^0 := \mathcal{J} \cup \{0\}$ und $\mathcal{K}^0 := \mathcal{K} \cup \{0\}$ eingeführt. Diese Formulierung ist angelehnt an ein MILP-Modell zur Maschinen- und Personalplanung in [SSZ21].

$$\sum_{i \in \mathcal{J}^0} \sum_{\substack{h \in \mathcal{K}^0: \\ O_{hi} \in \Omega \wedge O_{hi} \neq O_{kj} \wedge m \in \mathcal{M}_\ell}} X_{hikjm} = G_{kjm} \quad \forall j \in \mathcal{J}; k \in \mathcal{K}: O_{kj} \in \Omega; m \in \mathcal{M}_\ell \quad (3)$$

$$\sum_{i \in \mathcal{J}^0} \sum_{\substack{h \in \mathcal{K}^0: \\ O_{hi} \in \Omega \wedge O_{hi} \neq O_{kj} \wedge m \in \mathcal{M}_\ell}} X_{kjhim} = G_{kjm} \quad \forall j \in \mathcal{J}; k \in \mathcal{K}: O_{kj} \in \Omega; m \in \mathcal{M}_\ell \quad (4)$$

Nebenbedingungen (5) und (6) sorgen dafür, dass der fiktive Arbeitsgang auf jeder Ressource m jeweils höchstens einen Vorgänger und Nachfolger hat. Er ist aber nicht gezwungen, einen Vorgänger oder Nachfolger zu haben, da eine Ressource auch ungenutzt bleiben kann.

$$\sum_{j \in \mathcal{J}} \sum_{\substack{k \in \mathcal{K}: \\ O_{kj} \in \Omega \wedge m \in \mathcal{M}_\ell}} X_{kj00m} \leq 1 \quad m \in \mathcal{M}_\ell \quad (5)$$

$$\sum_{j \in \mathcal{J}} \sum_{\substack{k \in \mathcal{K}: \\ O_{kj} \in \Omega \wedge m \in \mathcal{M}_\ell}} X_{00kjm} \leq 1 \quad m \in \mathcal{M}_\ell \quad (6)$$

Die Bedingung, dass jeder echte Arbeitsgang von genau einer Ressource ausgeführt wird, ist durch Nebenbedingungen (7) erfüllt.

$$\sum_{m \in \mathcal{M}_\ell} G_{kjm} = 1 \quad \forall j \in \mathcal{J}; k \in \mathcal{K}: O_{kj} \in \Omega \quad (7)$$

Die korrekte Zuordnung von Startzeitpunkten für nacheinander auf einer Ressource eingeplante Arbeitsgänge wird durch Nebenbedingungen (8) gewährleistet. Gilt $X_{kjhim} = 1$, darf O_{hi} erst starten, wenn O_{kj} abgeschlossen ist.

$$S_{kj} + p_{kjm} \leq S_{hi} + (1 - X_{kj,hi,m}) \cdot \quad (8)$$

$$\forall k, h \in \mathcal{K}; j, i \in \mathcal{J}: O_{kj}, O_{hi} \in \Omega \wedge O_{kj} \neq O_{hi}; m \in \mathcal{M}_\ell \cap \mathcal{M}_h$$

Durch die Nebenbedingungen (3) - (7) werden effektiv Rundreisen zwischen den einer Ressource zugeordneten Arbeitsgängen gebildet, die bei dem fiktiven Arbeitsgang O_{00} beginnen und enden. Nebenbedingungen (8) verhindern die Bildung von Kurzzyklen bei der Festlegung der Rundreisen (für ähnlichen Sachverhalt siehe [SSZ21]). Wie in Abschnitt 3.4.1 beschrieben, darf zu jedem Zeitpunkt nur höchstens ein Primärarbeitsgang eines Jobs in Bearbeitung sein, weswegen für diesen eine Reihenfolge festgelegt werden muss.

Analog zu Nebenbedingungen (3) und (4) für die Ressourcen-Sequenzen wird von Nebenbedingungen (9) und (10) sichergestellt, dass alle Montagearbeitsgänge eines Jobs einen Vorgänger und einen Nachfolger in ihrer Job-Sequenz haben. Die Sequenz eines Jobs j beginnt und endet mit dem fiktiven Arbeitsgang O_{0j} .

$$\sum_{\substack{h \in \mathcal{K}^0: \\ O_{hj} \in \Omega \wedge h \neq k \wedge a_{hj}=1}} Y_{jkh} = 1 \quad \forall j \in \mathcal{J}; k \in \mathcal{K}^0: O_{kj} \in (\Omega \cup O_{0j}) \wedge a_{kj} = 1 \quad (9)$$

$$\sum_{\substack{h \in \mathcal{K}^0: \\ O_{hj} \in \Omega \wedge h \neq k \wedge a_{hj}=1}} Y_{jkh} = 1 \quad \forall j \in \mathcal{J}; k \in \mathcal{K}^0: O_{kj} \in (\Omega \cup O_{0j}) \wedge a_{kj} = 1 \quad (10)$$

Nebenbedingungen (11) definieren die frühesten Startzeitpunkte für die aufeinanderfolgenden Montagearbeitsgänge in der Job-Sequenz jedes Jobs und verhindert die Bildung von Kurzzyklen.

$$S_{kj} + \sum_{m \in \mathcal{M}_k} G_{kjm} \cdot p_{kjm} \leq S_{hj} + (1 - Y_{jkh}) \cdot \Theta$$

$$\forall j \in \mathcal{J}; k, h \in \mathcal{K}: O_{kj}, O_{hj} \in \Omega \wedge k \neq h \wedge a_{kj} = 1 \quad (11)$$

Nebenbedingungen (12) stellen die Einhaltung der geltenden Vorrangbeziehungen aus dem Vorranggraphen eines Jobs sicher, sowohl für die Produkt- als auch die Modularbeitsgänge.

$$S_{kj} + \sum_{m \in \mathcal{M}_k} G_{kjm} \cdot p_{kjm} \leq S_{hj} + (1 - p_{c_{jkh}}) \cdot \Theta$$

$$\forall j \in \mathcal{J}; k, h \in \mathcal{K}: O_{kj}, O_{hj} \in \Omega \wedge k \neq h \quad (12)$$

Wenn durch die Reihenfolge der Arbeitsgänge auf einer Ressource eine Umrüstung notwendig wird, sorgen die Nebenbedingungen (13) dafür, dass der folgende Arbeitsgang erst starten kann, wenn nach dem Endzeitpunkt des vorangegangenen Arbeitsgangs die reihenfolgespezifische Rüstzeit verstrichen ist.

$$S_{kj} + p_{kjm} + st_{khi} \leq S_{hi} + (1 - X_{kj,hi,m}) \cdot$$

$$\forall j, i \in \mathcal{J}; k, h \in \mathcal{K}: O_{kj}, O_{hi} \in \Omega \wedge k \neq h; m \in \mathcal{M}_k \cap \mathcal{M}_h \quad (13)$$

Durch Nebenbedingungen (14) wird der initiale Rüstzustand und der Freigabezeitpunkt bei der Terminierung des ersten Arbeitsgangs, der auf einer Ressource bearbeitet wird, berücksichtigt. Die initiale Umrüstung aus dem Ruhezustand ($k = 0$) wird durch st_{0km} abgebildet.

$$d_m + st_{0km} \leq S_{kj} + (1 - X_{00,kj,m}) \cdot$$

$$\forall j \in \mathcal{J}; k \in \mathcal{K}: O_{kj} \in \Omega; m \in \mathcal{M}_k \quad (14)$$

Die Nebenbedingungen (15) und (16) sorgen für die korrekte Berücksichtigung von Transportzeiten. Unter der Bedingung, dass zwei aufeinanderfolgende Arbeitsgänge eines Jobs auf unterschiedlichen Ressourcen bearbeitet werden, darf die Bearbeitung eines Arbeitsgangs erst starten, wenn nach dem Fertigstellungszeitpunkt des vorherigen Arbeitsgangs die nötige Transportzeit verstrichen ist.

$$S_{kj} + p_{kjm} + tt_{mm'} \leq S_{hj} + (2 - G_{kjm} - G_{hjm'}) \cdot \Theta + (1 - Y_{jkh})$$

$$\forall j \in \mathcal{J}; k, h \in \mathcal{K}: O_{kj}, O_{hj} \in \Omega \wedge k \neq h; m \in \mathcal{M}_{\hat{k}}; m' \in \mathcal{M}_{\hat{h}}: m \neq m' \quad (15)$$

Um einen Primärarbeitsgang starten zu können, müssen zunächst alle vorgelagerten Sekundärarbeitsgänge abgeschlossen sein und es muss für selbige die Transportzeit von der jeweiligen ausführenden Ressource hin zur ausführenden Ressource des Primärarbeitsgangs verstrichen sein. Nebenbedingungen (16) stellen auch die korrekte Einbeziehung von Transportzeiten zwischen zwei Sekundärarbeitsgängen sicher.

$$S_{kj} + p_{kjm} + tt_{mm'} \leq S_{hj} + (2 - G_{kjm} - G_{hjm'}) \cdot \Theta + (1 - pc_{jkh}) \cdot$$

$$\forall j \in \mathcal{J}; k, h \in \mathcal{K}: O_{kj}, O_{hj} \in \Omega \wedge (a_{kj} \neq 1); m \in \mathcal{M}_{\hat{k}}; m' \in \mathcal{M}_{\hat{h}}: m \neq m' \quad (16)$$

Die Nebenbedingungen (17) – (21) definieren den gültigen Wertebereich der Entscheidungsvariablen.

$$C_{max} \geq 0 \quad (17)$$

$$S_{kj} \geq 0 \quad \forall j \in \mathcal{J}; k \in \mathcal{K}: O_{kj} \in \Omega \quad (18)$$

$$X_{kjhim} \in \{0,1\} \quad \forall j, i \in \mathcal{J}^0; k, h \in \mathcal{K}^0: O_{kj}, O_{hi} \in \Omega; m \in \mathcal{M}_{\hat{k}} \cap \mathcal{M}_{\hat{h}} \quad (19)$$

$$Y_{jkh} \in \{0,1\} \quad \forall j \in \mathcal{J}; k, h \in \mathcal{K}^0: O_{kj}, O_{hj} \in (\Omega \cup O_{0j}) \wedge a_{kj} \quad (20)$$

$$G_{kjm} \in \{0,1\} \quad \forall j \in \mathcal{J}; k \in \mathcal{K}: O_{kj} \in \Omega; m \in \mathcal{M}_{\hat{k}} \quad (21)$$

4 Implementierung der Steuerungsarchitektur als Simulation

Das holonische Agentensystem wird in diesem Abschnitt durch Kombination einer Diskreten Event Simulation und eines simulierten Cyber Physischen Matrixproduktionssystems als agentenbasierte Modellierung und Simulation implementiert. Dabei werden die autonomen Control Devices der Fertigungs- und Transportressourcen mithilfe der DES repräsentiert und die holonischen Agenten zur Entscheidungsfindung innerhalb eines Multi-Agenten-Systems modular hierarchisch gekapselt.

ABMS sind in der Lage, natürliche Verhalten nachzubilden sowie flexibel anpassbar zu sein, insbesondere da das Verhalten des Gesamtsystems sich emergent aus den Interaktionen einzelner Agenten zusammensetzt, welche via „Plug-and-Play“ neue Eigenschaften erhalten und damit das übergeordnete System beeinflussen können [CH12]. Nachteil einer agentenbasierten Simulation ist dabei unter Umständen die schwerere Kontrolle des Gesamtsystems. Während individuelle Ziele der Agenten programmiert werden können, ist der aus der Zusammenarbeit der Agenten entstehende Effekt nicht zwingend vordefiniert. Ebenfalls sind die Anwendung und die Portabilität der MAS nicht ausreichend erforscht, sodass sie häufig in geschlossenen Softwareumgebungen wie beispielsweise MATLAB ausgeführt werden. Dahingehend ist die Skalierbarkeit hinsichtlich der Problemkomplexität nicht völlig deterministisch abzuschätzen [KPK18].

Grundsätzlich kann zwischen vier unterschiedliche Simulationsarten unterschieden werden. Dazu gehören die System Dynamics (SD), Dynamic Systems (DS), die ereignisorientierte Simulation (engl.: discrete event simulation - DES) und der agentenbasierten Modellierung und Simulation (ABMS). SD nutzen ein System aus Differenzialgleichungen, welches das System durch Rückführungsschleifen und Verzögerungen simuliert und ähnlich dem zweiten Typen der Dynamic Systems (DS) kontinuierliche Systemverhalten abbildet. Drittens existiert die am weitesten verbreitete Technik der DES, welche auf Basis des Prinzips der passiven Entitäten Systeme simuliert. Dabei wandern die passiven Entitäten anhand diskreter Zeitschritte und Übergänge durch das System unter Berücksichtigung von Warteschlangen und Verzögerungen. Den vierten Typen bildet die agentenbasierte Modellierung und Simulation (ABMS), bei welcher in der Grundform aktive und dezentrale Komponenten vorgesehen sind, welche in ihrer Summe das Verhalten des globalen Systems ausmachen. Zusätzlich dazu können alle vier Simulationen koexistieren, insbesondere bei integrierten und gekoppelten Systemen, in denen sowohl kontinuierliche Prozesse (SD) sowie diskrete Zeitschritte (DES/ABMS) betrachtet werden. Die allgemeine Eignung der Simulationstechniken für Produktionssysteme ist in Abbildung 4.1 dargestellt und zeigt die grundsätzliche Detailtiefe der Simulationen sowie die Dynamik des Systems.

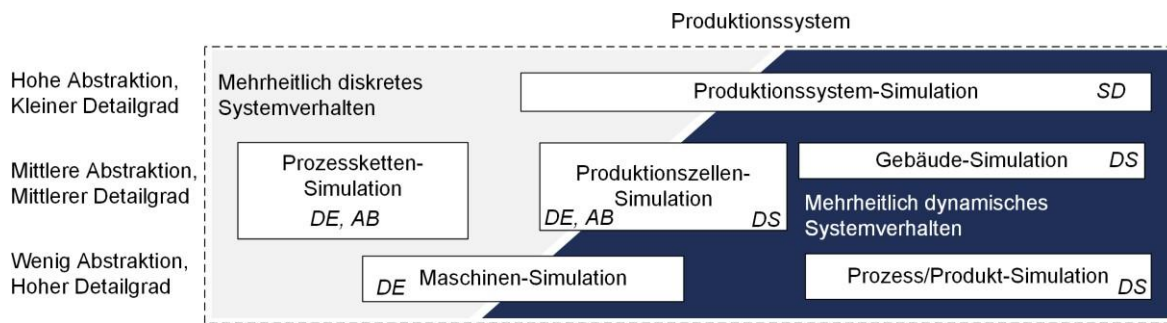


Abbildung 4.1: Anwendungsgebiete der Simulationsarten nach [Sc17]

Die agentenbasierte Simulation soll die Repräsentation eines Systems ermöglichen. Sie erweitert dabei eine klassische DES, welche den Fokus auf Prozesse und Aktivitäten setzt, um welches das System gebaut ist und nicht die Entitäten des Systems selbst. Die ABMS ergänzt damit die Möglichkeit, das Verhalten von Entitäten im Sinne verteilter intelligenter Systeme abzubilden. Die DES eignet sich, um das Prinzip der ausgelagerten agentenbasierten Entscheidungsfindung für eine Anwendung in der Praxis mit dezentralen intelligenten Systemen vorzubereiten. Insbesondere beim Blick auf die Umsetzung des hybriden Steuerungsparadigmas wird deutlich, dass die ABMS mit einer DES die unterschiedlichen Hierarchien und Planungshorizonte abbilden kann.

Im folgenden Kapitel wird die Implementierung der Modelle aus Kapitel 3 beschrieben. Dazu wird in Abschnitt 4.1 der Aufbau des agentenbasierten Simulationsframeworks mit seinen Systemkomponenten vorgestellt. In Abschnitt 4.2 wird die ebenenspezifische Implementierung des holonischen Agentenmodells mit seinen Interaktionen detailliert, bevor in Abschnitt 4.3 die Implementierung des Optimierungsmodells mit einem Lösungsverfahren gezeigt wird.

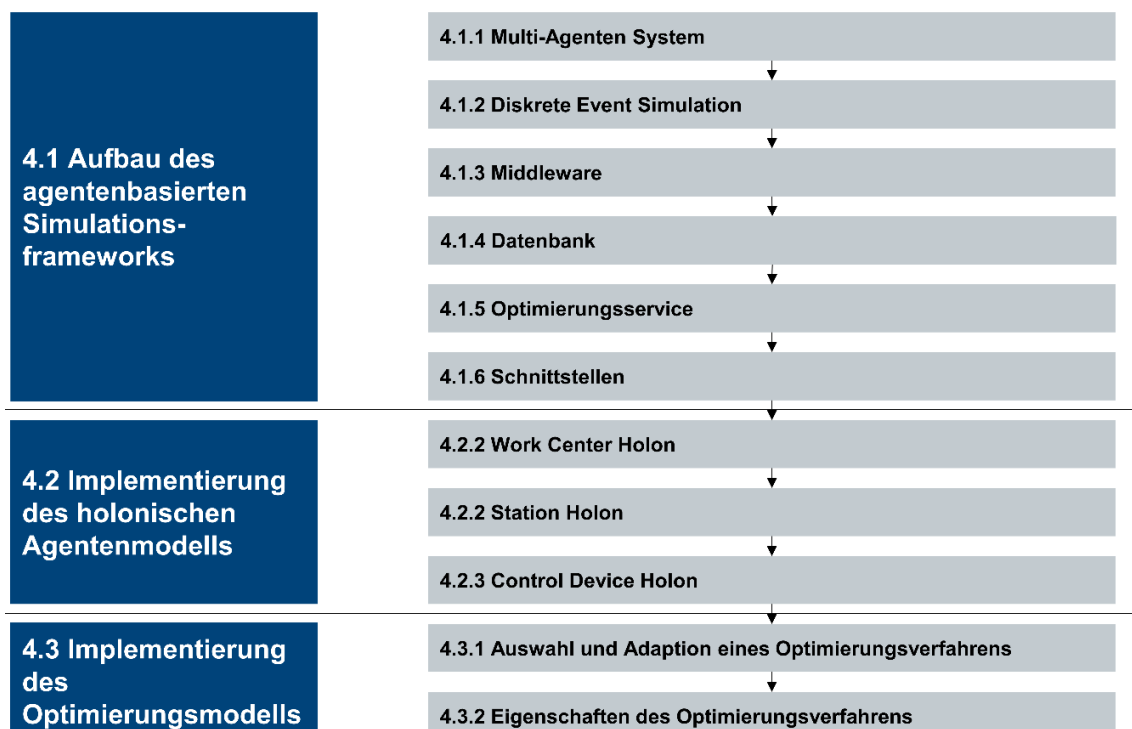


Abbildung 4.2: Struktur des Kapitels 4

4.1 Aufbau des agentenbasierten Simulationsframeworks

Ein Framework mit agentenbasierter serviceorientierter Architektur in Kombination mit einer Materialflusssimulation wurde in [Ma19b] zur Algorithmenvvalidierung im Kontext modularer Produktionsumgebungen verwendet. Mayer gliedert das Framework in Simulation und Entscheidungssystem, welches einer Entscheidungsfindung nach holonischem Architekturdesign entspricht. Ergänzt wird das Agentensystem mit einer Datenbank für standardisierten Zugriff auf Daten und Informationen und einer Middleware für interoperable Datenbereitstellung und -integration. Dieser Aufbau wird zur Implementierung der holonischen Steuerungsarchitektur adaptiert. In Abbildung 4.3 wird aufgezeigt, wie das Simulationsframework von [MA19b] in eine RAMI4.0-fähige Architektur für die Implementierung abstrahiert wird.

Simulationsframework Mayer		Modellierung I4.0-fähige Architektur		Implementierung
Systemdesign	Komponenten	RAMI4.0	Systemmodell	Systemstruktur
Entscheidungssystem	Python Scheduling MQTT - Broker RPi 3B+ MySQL DB	Geschäftsprozess und Organisation Funktion	Scheduling Modul Agentensystem	Service MAS Middleware DB
		Information Kommunikation	Datalayer Middleware	
Simulationsmodell	DES CPPS (Station und FTF)	Integration Asset	CPS Simulation	DES CPS (Control Device)

Abbildung 4.3: Abstraktion und Implementierung der Steuerungsarchitektur nach Abstraktion von [Ma19b]

Die wesentlichen Komponenten des agentenbasierten Simulationsframeworks sind das Multi-Agenten-System (Abschnitt 4.1.1), die Diskrete-Event-Simulation (Abschnitt 4.1.2), die Communication Middleware (Abschnitt 4.1.3), die Datenbank (Abschnitt 4.1.4) und der Optimierungsservice (Abschnitt 4.1.5). Die Systeme werden über geeignete Nachrichtenformate (Abschnitt 4.1.6) miteinander verbunden. Die Systeme können entsprechend der Einteilung in die Architekturlayer nach RAMI4.0 in Planung und Ausführung unterschieden werden, die über die Kommunikations- und Informationsebene verbunden werden.

Zur Umsetzung der Steuerungsarchitektur wird eine Systemstruktur erstellt, die in Abbildung 4.4 dargestellt ist. Die Systemstruktur zeigt die Implementierung der holonischen Steuerungsarchitektur als RAMI4.0-konforme Architektur mit den ausgewählten Implementierungstechnologien, Schnittstellen und Nachrichtenformaten. In den nächsten Abschnitten werden die verwendeten Komponenten zur Implementierung und die Realisierung der Interaktion vorgestellt.

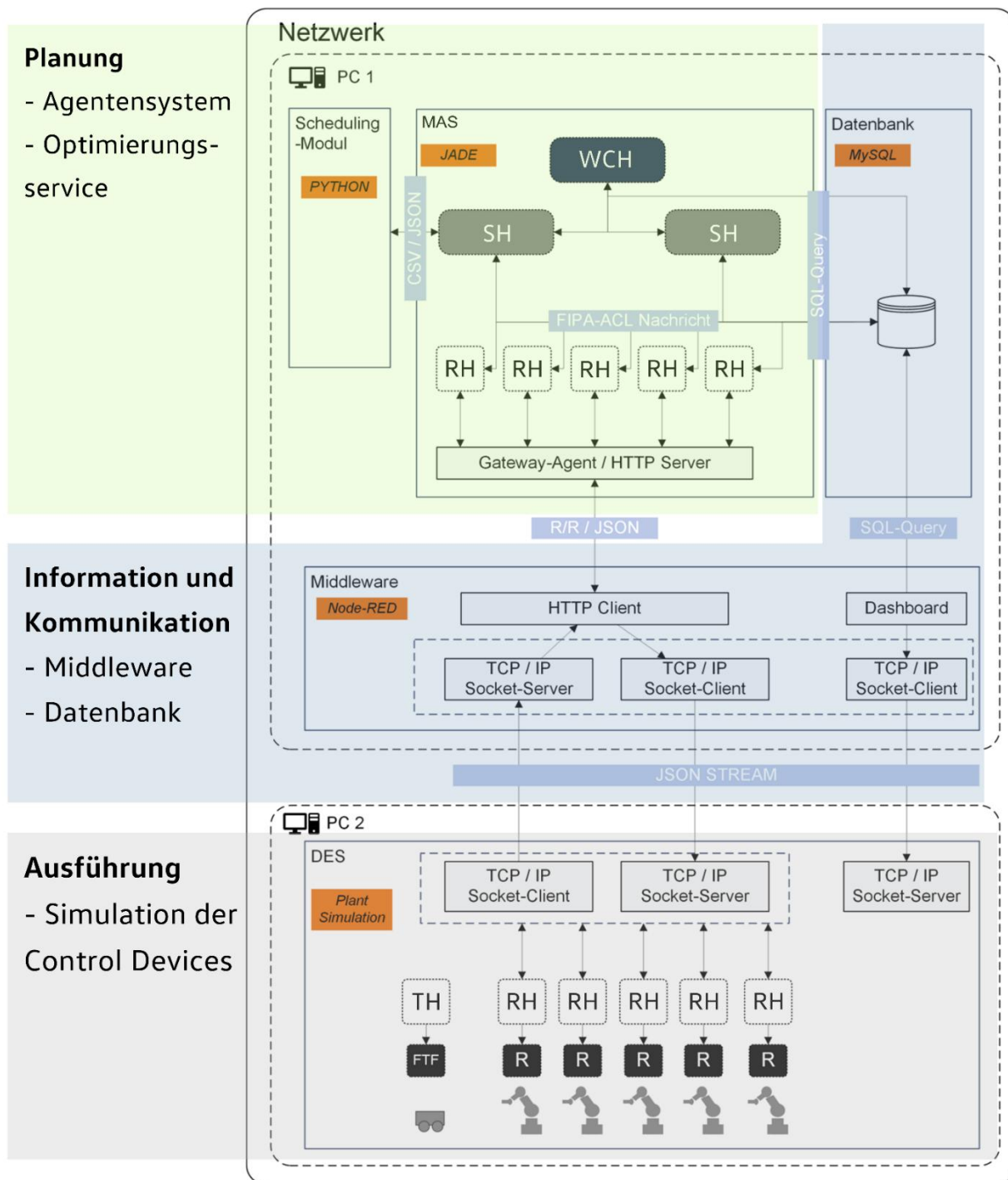


Abbildung 4.4: Implementierte Steuerungsarchitektur mit verwendeten Systemkomponenten und Technologien

4.1.1 Multi-Agenten-System

Die Realisierung des Agentenmodells zur übergeordneten Entscheidungsfindung kann durch den Einsatz von Agentenplattformen erfolgen. Agentenplattformen integrieren grundlegende nachrichtenbasierte Kommunikationsmöglichkeiten, Vorlagen für verhaltensbasierte Funktionen und Schnittstellen zu externen Komponenten [Ab17]. Die Auswahl einer Agentenplattform hängt von

verschiedenen Kriterien ab, darunter Programmiersprachen, Skalierbarkeit, Robustheit, Performanz und Funktionsumfang einschließlich der Protokollunterstützung. Ein Vergleich von Plattformen nach besagt, dass mehr als 80% der Plattformen hauptsächlich auf JAVA basieren. Dies liegt unter anderem an der Interoperabilität durch die Implementierung mittels virtueller Maschinen auf verschiedenen Betriebssystemen [KB15; CF21].

JADE (Java Agent DEvelopment Framework) wird aufgrund seiner hohen Performanz, Skalierbarkeit und Stabilität als geeignete Plattform ausgewählt. Insbesondere die asynchrone Nachrichtenübertragung in JADE ermöglicht eine überdurchschnittlich schnelle Kommunikation. Die Unterstützung des FIPA-ACL Standards und die Popularität von JADE als Plattform für die Entwicklung agentenbasierter Systeme tragen dazu bei, dass der Zugang und die Anwendung auch für weniger erfahrene Nutzer gelingt [KB15]. Besonderer Fokus liegt ebenfalls auf den Verhaltensweisen der Agenten. Die JADE-Plattform implementiert dazu unterschiedliche Ausprägungen der Verhaltensweisen. Die wichtigsten Vertreter der einfachen Verhaltensweisen sind das OneShotBehaviour, TickerBehaviour, WakerBehaviour und das CyclicBehaviour. Das OneShotBehaviour dient dem einmaligen Ausführen einer Aktion und steht konträr zum CyclicBehaviour, welches zyklisch mehrfach ausgeführt wird. Das TickerBehaviour kann ebenfalls mehrfach ausgeführt werden und orientiert sich dabei an Ticks, welche diskreten Eventschritten innerhalb des Systems entsprechen. Das WakerBehaviour wird einmalig aktiviert, nachdem eine definierte Zeit vergangen ist [BCG10].

Aufgrund dieser Überlegungen wird JADE als Entwicklungsplattform für die Implementierung der holonischen Agenten der Control Device-, Station- und Work Station-Ebene ausgewählt. In Abbildung 4.5 ist die Grundarchitektur von JADE dargestellt.

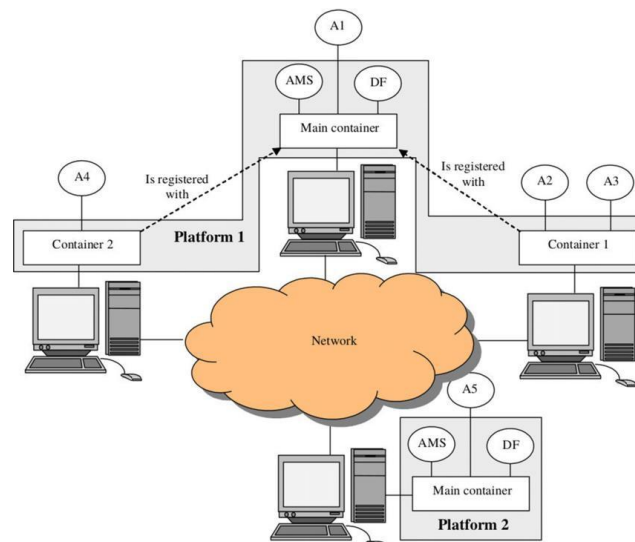


Abbildung 4.5: Architektur der JADE-Plattform nach [Me12]

Obwohl JADE nach außen hin als eine umfassende Plattform erscheint, kann sie aufgrund der Implementierung und Nutzung mehrerer Container selbst als verteiltes System fungieren. Die

Architektur von JADE ermöglicht die Verteilung von Containern und damit von Agenten auf mehrere Computer. Ein Container in der JADE-Architektur enthält Agenten, die JAVA-Prozesse innerhalb des Containers repräsentieren. Ein besonderer Container ist der Hauptcontainer, bei dem sich alle nachträglich erstellten Container registrieren. Dieser Hauptcontainer stellt das Agent-Management System und den Directory Facilitator bereit. Das Agent-Management System fungiert als zentraler Punkt für die Interaktion der Agenten untereinander, da die Agenten hier durch Registrierung einen eindeutigen Agentennamen erhalten. Der Directory Facilitator dient dem Registrieren von individuellen Leistungen, die ein Agent bereitstellen kann und fungiert somit als Katalog von Fähigkeiten [BCG10].

In untenstehender Abbildung ist ein Ausschnitt der Agenten und der FIPA-ACL basierten Kommunikation dargestellt. Die Implementierung der einzelnen Holone ist in Abschnitt 4.2 detailliert beschrieben.

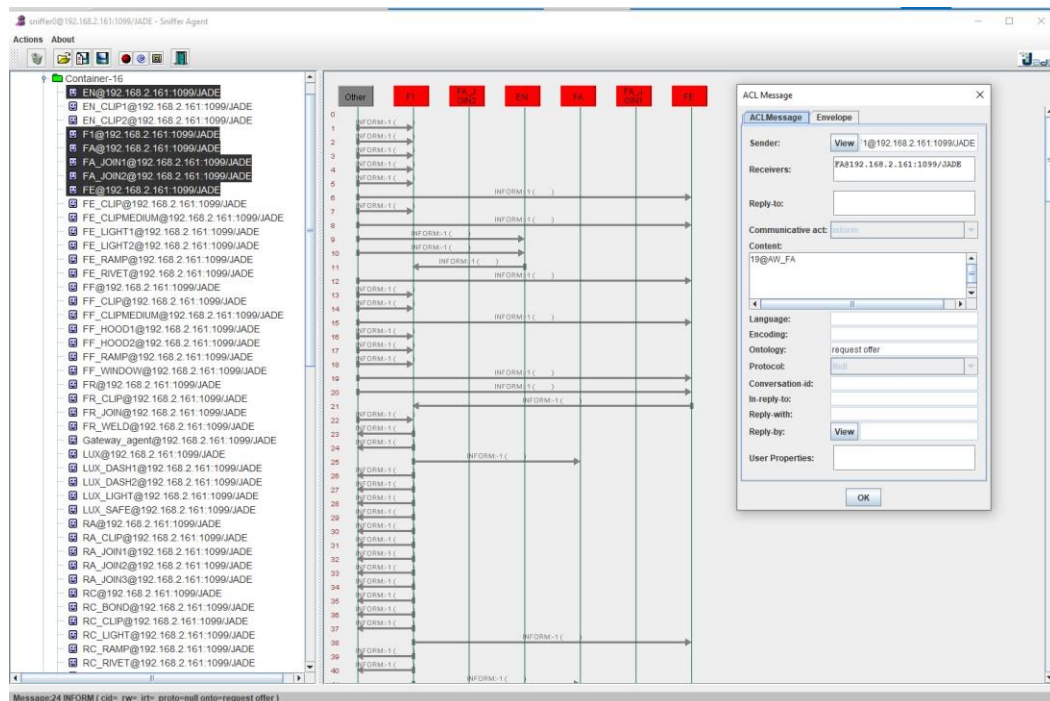


Abbildung 4.6: Übersicht der Agenten, der Agenteninteraktion und einer FIPA-ACL Nachricht

4.1.2 Diskrete Event Simulation

Für die Simulation der Ressourcen wird die Diskrete-Event-Simulation (DES) Software Tecnomatix Plant Simulation verwendet. Diese Software wurde speziell für Automobilhersteller entwickelt und bietet verschiedene Merkmale und Baukästen, die sich auf den Anwendungsfall dieser wissenschaftlichen Arbeit, nämlich die Automobilproduktion, konzentrieren. Die Auswahl dieser Software ermöglicht eine optimale Anbindung an parallele Simulationen und an die Technologien, die im Forschungsumfeld dieser Arbeit relevant sind. Dies gewährleistet die Erfüllung der Anforderungen an die systemübergreifende Interoperabilität. Diese Software wurde bereits in einer früheren Anwendung als Multi-Methoden-Simulation durch [ME19; Ma19b] eingesetzt,

was ihre grundsätzliche Eignung unterstreicht. [Bu17] haben ebenfalls eine agentenbasierte Simulation in Plant Simulation realisiert. Die vollständige implementierte Plant-Simulation ist in zweidimensionaler und dreidimensionaler Form in untenstehenden Abbildungen dargestellt.

Die DES repräsentiert die Control Devices der Ausführungsebene des Architekturmodells und beinhaltet Agenten mit kurzfristigem Planungshorizont. Eine umfassende Übersicht über mehr als 16 verschiedene Diskrete-Event-Simulationen findet sich im Vergleich von [Di16]. Das CPMPs stellt die dynamischen PPR-Daten bereit, welche die Agenten für eine zustandsaktuelle Entscheidungsfindung berücksichtigen.

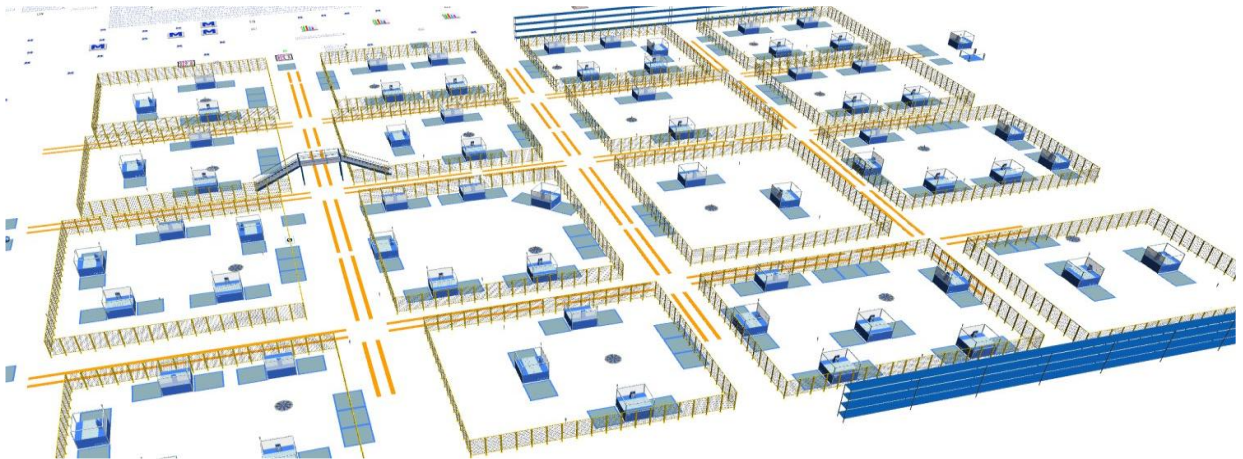


Abbildung 4.7: Dreidimensionale Darstellung der technischen Umgebung in Tecnomatix Plant Simulation

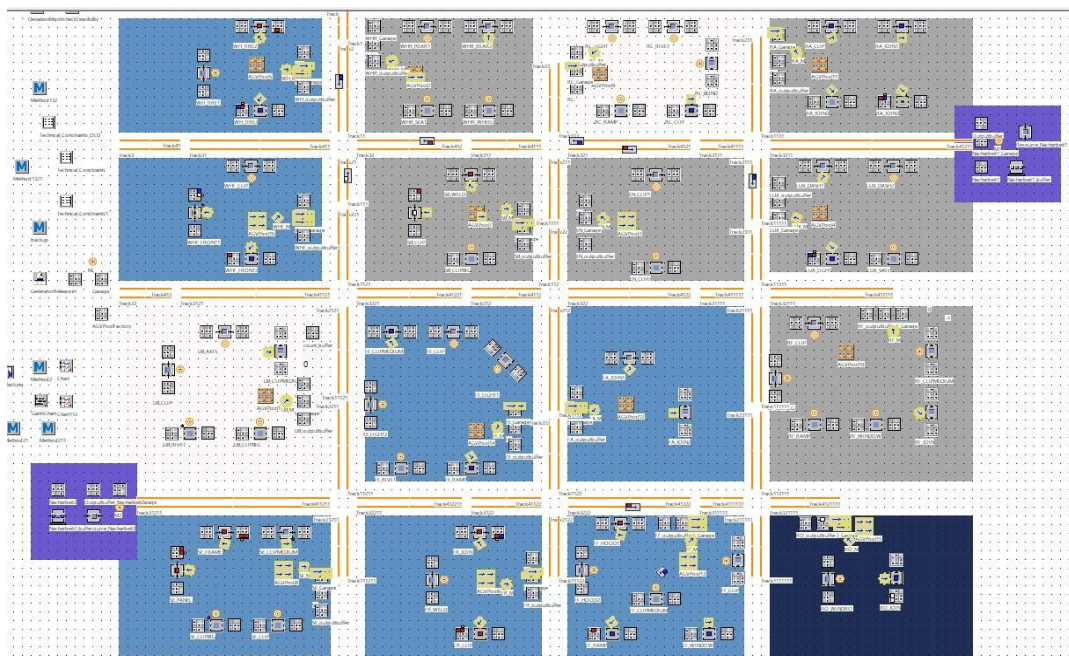


Abbildung 4.8: Zweidimensionale Darstellung der technischen Umgebung in Plant Simulation mit Nacharbeitsstationen (lila)

4.1.3 Middleware

In I4.0-fähigen Architekturen wird die Kommunikation und der Datenaustausch zwischen Komponenten für Interoperabilität und Kompatibilität oft mithilfe einer Middleware realisiert. Durch Datenadapter können strukturierte Eingangsdaten eines Absenders in die gewünschte Datenstruktur eines anderen Empfängers transformiert werden. Die Kommunikation zwischen Discrete-Event-Simulation (DES) und Multi-Agenten-System (MAS) erfolgt über eine standardisierte Schnittstelle mit bidirektionalem Datenaustausch, wobei die Anforderungen an Nachrichtengröße und Geschwindigkeit des Datenaustauschs berücksichtigt werden müssen. Mayer verwendet eine Socket-Schnittstelle und setzt auf eine TCP/IP-basierte Client-Server-Verbindung zwischen der DES und dem Agentensystem. Über diese Schnittstelle werden Updates der Simulation, wie die Anfrage des nächsten Auftrags, des nächsten Prozessschritts, Transport-Updates und Maschinenupdates, ereignisorientiert und strukturiert gesendet.

Von [Ng17] werden drei verschiedene Arten von Middleware unterschieden: Servicebasierte, aktorbasierte und Cloud-basierte Middleware. Für eine ausführliche Analyse dieser Typen wird auf die entsprechende Veröffentlichung verwiesen. Die servicebasierte Middleware, obwohl leistungsstark, erfordert einen erheblichen Implementierungsaufwand und unterstützt keine direkte Gerät-zu-Gerät-Kommunikation. Eine Cloud-basierte Middleware bietet die Möglichkeit, Funktionen in der Cloud anzubinden und über Schnittstellen zugänglich zu machen, setzt jedoch voraus, dass Schnittstellen und Geräte im Voraus definiert sind. Die aktorbasierte Middleware hingegen konzentriert sich auf Wiederverwendbarkeit, Zugänglichkeit und Plug-and-Play. Diese Middleware kann auf jeder Ebene zur Integration eingesetzt werden und ermöglicht eine dezentrale Gestaltung. Sowohl die servicebasierte als auch die aktorbasierte Middleware entsprechen den Prinzipien des Internet der Dinge (IoT) und erleichtern die Integration unterschiedlicher Geräte, indem sie diese abstrahieren und Programmierschnittstellen bereitstellen. Für die Verbindung der DES und des MAS wird eine aktorbasierte Middleware verwendet aufgrund ihrer leichtgewichtigen Implementierungseigenschaften. In diesem Kontext schlagen [Ng17] drei mögliche Systeme vor: Calvin, Node-RED und Ptolemy. Aufgrund der Unterstützung von einschlägigen IoT-Protokollen, der benutzerfreundlichen Programmierung und der einfachen Implementierung von Visualisierungen und Benutzereingaben, wird Node-RED ausgewählt [Ng17].

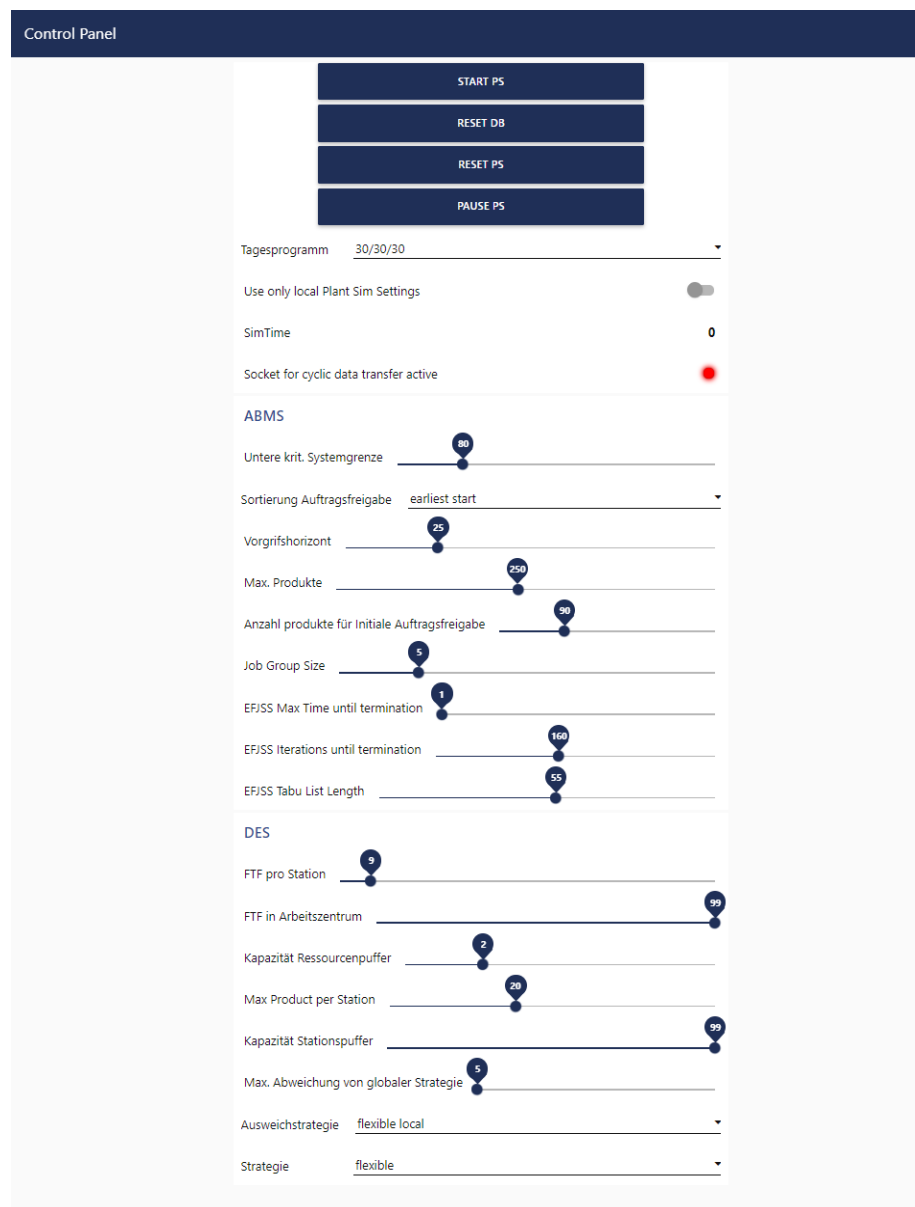


Abbildung 4.9: Benutzeroberfläche der Middleware

4.1.4 Datenbank

Das Informations- und Datenmodell wird mithilfe eines UML-Klassendiagramms visualisiert und abstrahiert die Datenstruktur des Produktionssystems. Dadurch wird die Synchronisation von Simulations- und Agentensystem ermöglicht. Zudem können die auf diesem Datenmodell basierenden Informationen in einer Datenbank gespeichert werden. Eine Übersicht des Informationsmodells findet sich im Anhang in Abbildung A.1.

Aufgrund der signifikanten Bedeutung hochwertiger Daten im Kontext von Industrie 4.0 und dem Internet der Dinge (IoT) benötigt das Gesamtsystem eine Methode für die konsistente Datenspeicherung. Besonders bei der Generierung von strukturierten und unstrukturierten Daten in großer Menge sind systematische und effiziente Lösungen zur Speicherung erforderlich [Az20]. Hierbei

wird zwischen relationalen, nicht-relationalen (NoSQL) und grafischen Datenbanken unterschieden. Grafenbasierte Datenbanken sind für den vorliegenden Anwendungsfall irrelevant, während relationale und nicht-relationale Datenbanken für die konsistente Datenhaltung von ABMS (Agentenbasierte Modellierung und Simulation) und DES (Diskrete-Event-Simulation) geeignet sind.

Die Anforderungen an diese Datenbanken im I4.0- und IoT-Kontext umfassen Speicherkosten, Flexibilität, Skalierbarkeit und Datenabfragegeschwindigkeit [Az20]. Relationale Datenbanken, basierend auf SQL-Abfragesprache, zeichnen sich durch Stabilität und Effizienz in einer statischen Umgebung aus. Nicht-relationale Datenbanken (NoSQL) bieten hingegen Vorteile in Bezug auf Skalierbarkeit und Flexibilität. Die Wahl zwischen Flexibilität und Robustheit im Sinne des IoT hängt vom Anwendungsfall und den Präferenzen des Nutzers ab [RB16]. Aufgrund der Anforderungen an die zentrale Middleware und der Notwendigkeit von Stabilität für das konsistente Datenmanagement wird eine MySQL-Datenbank verwendet. In untenstehender Abbildung ist das Fähigkeitsschema exemplarisch dargestellt. Weitere Details zur Implementierung der Ressourcen-, Auftrags- und Fähigkeitsinformationen innerhalb der Datenbank sind im Anhang dargestellt.

id	setup	setup_time	factor	resource	costs
CLIP_3x2	CLIP_3x2	0	1	FF_CLIPMEDIUM	12
CLIP_3x2	CLIP_3x2	0	0.3	LUX_DASH2	12
CLIP_3x2	CLIP_3x2	0	0.75	LUX_LIGHT	12
CLIP_3x2	CLIP_3x2	0	1	RF_CLIP	12
CLIP_3x2	CLIP_3x2	0	0.3	UB_AXIS	12
CLIP_3x2	CLIP_3x2	0	0.3	UB_CLIP	12
CLIP_3x2	CLIP_3x2	0	0.3	UB_CLIPMEDIUM	12
CLIP_3x2	CLIP_3x2	0	0.3	WHR_REAR2	12
CLIP_3x2	CLIP_3x2x1	3	0.5	EN_CLIP1	12
CLIP_3x2	CLIP_3x2x1	3	0.5	EN_CLIP2	12
CLIP_3x2	CLIP_4x1	3	0.3	LUX_DASH2	12
CLIP_3x2	CLIP_4x1	2	0.3	UB_AXIS	12
CLIP_3x2	CLIP_4x1	2	0.3	UB_CLIP	12
CLIP_3x2	CLIP_4x1	2	0.3	UB_CLIPMEDIUM	12
CLIP_3x2	CLIP_4x1x1	3	0.5	EN_CLIP1	12
CLIP_3x2	CLIP_4x1x1	3	0.5	EN_CLIP2	12
CLIP_3x2	CLIP_4x2	2	0.3	UB_AXIS	12
CLIP_3x2	CLIP_4x2	2	0.3	UB_CLIP	12
CLIP_3x2	CLIP_4x2	2	0.3	UB_CLIPMEDIUM	12
CLIP_3x2	CLIP_4x2x1	3	0.3	LUX_DASH2	12
CLIP_3x2	CLIP_4x2x1	3	0.3	WHR_REAR2	12
CLIP_3x2	CLIP_6x1	3	0.3	LUX_DASH2	12
CLIP_3x2	CLIP_6x1	3	0.75	LUX_LIGHT	12
CLIP_3x2	CLIP_6x1	2	0.3	UB_AXIS	12
CLIP_3x2	CLIP_6x1	2	0.3	UB_CLIP	12
CLIP_3x2	CLIP_6x1	2	0.3	UB_CLIPMEDIUM	12
CLIP_3x2	CLIP_6x2	2	0.3	UB_AXIS	12
CLIP_3x2	CLIP_6x2	2	0.3	UB_CLIP	12
CLIP_3x2	CLIP_6x2	2	0.3	UB_CLIPMEDIUM	12
CLIP_3x2	DASHBOARD	3	0.3	LUX_DASH2	12
CLIP_3x2	JOIN_6x4x1	1	0.3	WHR_REAR2	12
CLIP_3x2	LIGHT_4x1	3	0.75	LUX_LIGHT	12
CLIP_3x2	RAMP_4x1	1	0.3	WHR_REAR2	12
CLIP_3x2	RIVET_1x1	3	0.3	LUX_DASH2	12
CLIP_3x2	RIVET_1x1	3	0.75	LUX_LIGHT	12
CLIP_3x2	RIVET_2x1	3	0.75	LUX_LIGHT	12
CLIP_3x2	WHEELHOUSE_REAR	1	0.3	WHR_REAR2	12
CLIP_3x2x1	CLIP_2x1x1	2	0.3	UB_CLIPBIG	12

Abbildung 4.10: Auszug des Fähigkeiten-Schemas aus der MySQL-Datenbank

4.1.5 Optimierungsservice

Ein serviceorientiertes Optimierungsmodul wird agentenbasiert zur systemspezifischen Auftragsplanung für die Ressourcen- und Auftragsreihenfolgeoptimierung angefragt, um das NP-schwere Optimierungsmodell zu lösen. Es existieren mehrere Maschinen, welche nicht baugleich sein müssen, aber durch ein fähigkeitsbasiertes Matching verschiedene Arbeitsgänge ausführen können. Dies steht in engem Zusammenhang mit der fähigkeitsbasierten Modellierung zur Steuerung produkt- und prozessflexibler Systeme. Die exakte Lösung durch Anwendung eines Solvers auf eine Modellimplementierung ist ungeeignet, da die Anzahl der zu bestimmenden Entscheidungsvariablen für praxisrelevante Instanzen zu groß ist. Daher fällt die Wahl auf die Anwendung einer Metaheuristik. Das Lösungsverfahren und das Verhalten des Verfahrens werden in Abschnitt 4.3 ausführlicher erläutert. Für das systemspezifische Scheduling von Aufträgen wird ein Request über den Service Interface Layer gestellt. Ein Beispiel für den Nachrichtenaustausch via JSON ist in untenstehender Abbildung dargestellt.

4.1.6 Schnittstellen

Die Grundlagen der Kommunikation im vorliegenden Simulationsframework werden durch Nachrichten und die genutzten Formate gelegt. Während die systeminterne Kommunikation innerhalb der DES und ABMS durch die dortige Programmiersprache direkt interpretierbar ist, ist die Kommunikation über die Schnittstellen (DES zu Middleware, Middleware zu MAS, MAS zu Ablaufplanungsmodul) zu definieren. Die effiziente Kommunikation beinhaltet die Serialisierung, die eine Nachricht in ein interpretierbares strukturiertes Format umwandelt. Bei der Auswahl des Nachrichtenformats ist zu beachten, dass die Nachricht menschenlesbar sein sollte, um die Nachverfolgbarkeit der Befehle für Fehlerbehebung zu ermöglichen. JSON wurde aufgrund seiner Geschwindigkeit, weitverbreiteten Nutzung und Verfügbarkeit von Software für die Serialisierung ausgewählt.

Als zentrale Schnittstelle verbindet Node-Red die beiden Teilkomponenten der ABMS. Auf ABMS-Seite stellt die JADE-Plattform in ihren grundlegenden Bestandteilen bereits die Kompatibilität mit HTTP-Protokollen zur Verfügung. Die Kommunikation der JADE-Plattform nach außen kann mithilfe des robusten HTTP-Protokolls erfolgen und wird durch die Vorlage eines HTTP-Servieragenten unterstützt [BCG10]. Die interne Kommunikation der JADE-Plattform und enthaltenen Agenten erfolgt dahingegen durch direkte, asynchrone Nachrichten und der Ausführung von JAVA-Prozessen und Funktionen. Die DES besitzt ebenfalls mehrere mögliche Schnittstellen wie die dateibasierte Kommunikation, OPC-UA und die Socket-Schnittstelle. Während zentrale Datenbanken und dateibasierte Kommunikation dem Gedanken der verteilten Systeme widersprechen, eignen sich OPC-UA und die Socket Schnittstelle für die Kommunikation. Sowohl OPC-UA als auch die Socket Schnittstelle basieren auf dem TCP-Transportprotokoll. Da die Implementierung von OPC-UA aufgrund der Verwendung von Sicherheitszertifikaten nicht trivial ist, wird die

leichtgewichtige Implementierung mittels TCP-Sockets gewählt. Die Verwendung eines P/S Musters für die Kommunikation zu externen Komponenten ist nicht notwendig, da die derzeitige Implementierung noch keine IIoT-Geräte berücksichtigt.

Da die lizenzpflichtige Software Plant Simulation bereits vorhanden ist, allerdings an einen spezifischen Rechner PC 2 gebunden ist, sieht die Gesamtarchitektur einen zweigeteilten Ansatz vor. Der Lizenzrechner erlaubt keine Installation von Open-Source-Software, sodass die Agentenplattform sowie die Middleware hierbei auf einem zweiten Rechner PC 1 installiert wird. Zusätzlich wird die relationale MySQL Datenbank auf dem Rechner aufgrund derselben Restriktionen installiert. Die grundlegende Schnittstelle zwischen PC 1 und PC 2 ist durch eine TCP/IP Socket-schnittstelle definiert. Mit einer Client-Server-Architektur kann eine schnelle Übertragung von Daten von PC 1 an PC 2 erfolgen. Da es sich um eine Socketverbindung handelt, werden die Daten serialisiert und mithilfe eines Buffers gesendet und vom Empfänger empfangen. Da die Übertragungsgröße der Pakete begrenzt ist, werden einzelne Pakete gesendet und der vollständige Transport des JSON-Streams mit einem definierten Zeichen bestätigt. Hier handelt es sich nur um eine direkte Verbindung ohne Festlegung komplexer Protokolle, sodass programmiert werden kann, wie mit Nachrichten umzugehen ist.

Eine weitere Verbindungsebene ist auf dem Rechner PC 1 über eine HTTP-Client-Serververbindung implementiert. Im Rahmen der Agentenplattform ist der HTTP-Server bereits in einem Gateway-Agenten (GA) integriert, der die Interkommunikation zwischen den Systemen koordiniert [Me09; Vo20]. Die Middleware, welche die Datensätze und Anfragen der Plant Simulation in HTTP-Requests umwandelt, stellt eine Verbindung zu diesem GA her. Der GA wiederum transformiert die empfangenen Anfragen in FIPA-ACL-Nachrichten, die als Kommunikationsmittel für die Agenten in der JADE-Plattform fungieren.

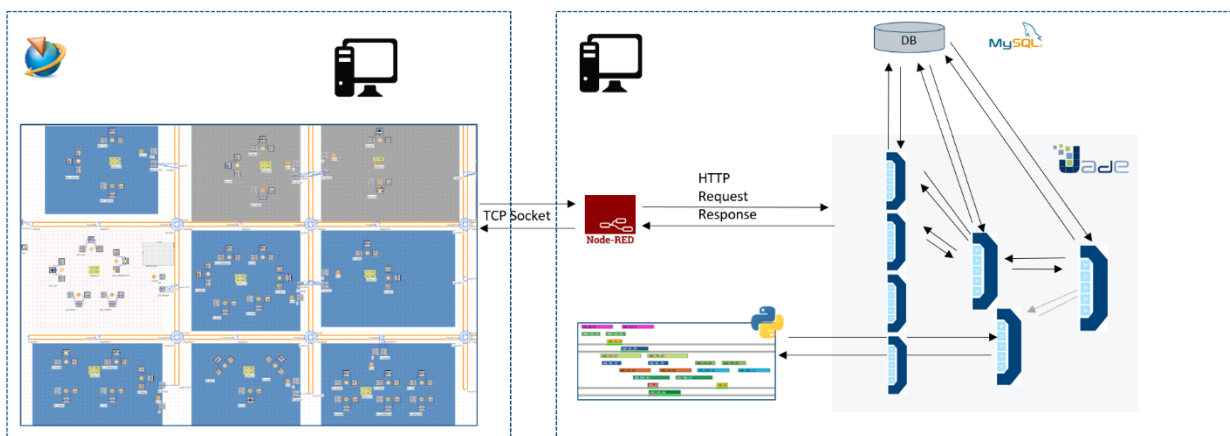


Abbildung 4.11: Übersicht der Komponenten des Simulationsframeworks

4.2 Implementierung des holonischen Agentenmodells

In diesem Abschnitt wird das interne Verhalten der Holone sowie die Interaktion der Holone untereinander implementiert. Es bestehen ebenenspezifische Unterschiede, die entweder zu einer

Selbstoptimierung des Gesamtsystems oder zur Anpassung der Simulation führen. Die nachfolgenden UML-Diagramme markieren in Rot die Möglichkeiten zur Auswahl verschiedener Alternativen oder Parameter für Steuerungsentscheidungen. Diese repräsentieren die Optionen zur Parametrisierung der Fertigungsstrategien als Teil des agentenbasierten Simulationsframeworks. Die Transportsteuerung wird als Funktion in Plant Simulation ausgeführt.

4.2.1 Work Center Holon

Das übergeordnete System auf der höchsten Hierarchieebene plant die Fertigung der Produkte und besitzt den langfristigen Entscheidungshorizont. Das Work Center Holon (WCH) besitzt die planerische Verantwortung für die Koordination des Gesamtsystems. Gemäß den identifizierten Entscheidungen aus Abschnitt 4.4.1 trifft das WCH anhand des Systembestands eine Entscheidung zur Auftragsfreigabe der Produktaufträge des Tagesprogramms. Grundsätzliche Optionen für die Freigabe sind die sofortige Freigabe, die Freigabe nach dem geplanten Starttermin oder eine bestandsorientierte Freigabe in Abhängigkeit des Bestandes des Produktionssystems oder der Auslastung der Arbeitssysteme [Lö16]. Im weiteren Verlauf wird die Freigabe für die Reihenfolgeoptimierung durch eine bestandsorientierte Auftragsfreigaberegeln festgelegt. Durch Sortierungen nach frei wählbaren Terminen kann eine parametrisierte Menge an Produkten freigegeben werden. Dies kann als eine Vereinfachung des Vorgriffshorizonts für den Volumentakt betrachtet werden, der Aufträge vor dem geplanten Starttermin bereits für das System freigibt. Diese Freigabe stellt eine ereignisorientierte und indirekt bestandssteuernde Auftragsfreigabe dar. Der Detaillierungsgrad der Auftragsfreigabe entspricht der L3-Ebene und betrifft durch die Dekomposition ebenfalls direkt die Freigabe auf L2-Ebene.

Nachdem der WCH im "LISTENING"-Zustand die Rückmeldungen seiner Sub-Holone (SH) erhalten hat, übernimmt er den Zustand "ORDER_RELEASE", wenn der Systembestand unter einen festgelegten Wert fällt. In diesem Zustand entnimmt der WCH eine frei-parametrisierbare Anzahl neuer Aufträge aus der Datenbank, wobei Sortierungsalgorithmen und Informationen zur Anzahl der Aufträge (AL1) aus der Fertigungsstrategie angewendet werden.

Bei keinerlei Statusänderungen und beim Empfang von Herzschlag-Nachrichten der Sub-Holone antwortet der WCH mit dem Zustand "ACKNOWLEDGE_NO_ACTION", um den Sub-Holonen mitzuteilen, dass keine zusätzlichen Aufträge entnommen werden sollen.

Da eine globale Optimierung für Belegungs- und Reihenfolgeplanung mehrere Aufträge und die Abstimmung zwischen diesen erfordert, wird eine größere Anzahl von Aufträgen aus der Datenbank zur Freigabe ermittelt. Gemäß der Entscheidung E1 erfolgt die erste Planung mit Losgrößen, bei der mehrere Aufträge zu Losen gebündelt werden (Zustand "JOB_GROUPING"). Die einzelnen Lose müssen aufgrund technischer Restriktionen einheitlich sein, während die Lose selbst sich vollständig unterscheiden können. Zusätzlich müssen die identifizierten Produktlose in Teilaufträge (A_{L2}) aufgeteilt werden, die innerhalb ihrer Lose bleiben. Diese Lose werden auf höherer aggregierter Ebene zur Komplexitätsreduktion geplant, während auf unterer Ebene die Aufträge individuell behandelt werden.

Nach dem Wechsel in den Zustand "REQUESTING_OFFERS" erfolgt die Abfrage der vorhandenen Fähigkeiten der eigenen Stationen (SHs) sowie die Anfrage nach Angeboten für die Fertigung der Module der Auftragsgruppen (A_{L2}). Nach Erhalt der Angebote bildet das Angebot die Grundlage für die Prozesszeit, die einen entscheidenden Einfluss auf die Gesamtbearbeitungszeit und somit die Belegungs- und Reihenfolgeplanung hat. Nach Erhalt aller Angebote wird im Zustand "SCHEDULING" unter Verwendung eines in Abschnitt 4.3 implementierten Optimierungsverfahrens ein abschließendes Scheduling durchgeführt. Nach einer vordefinierten Optimierungszeit, die während der Ausführung des Scheduling-Moduls begrenzt werden kann, werden die zugeordneten Aufträge im Zustand "CONFIRMATION" an alle berücksichtigten Stationsholone weitergeleitet.

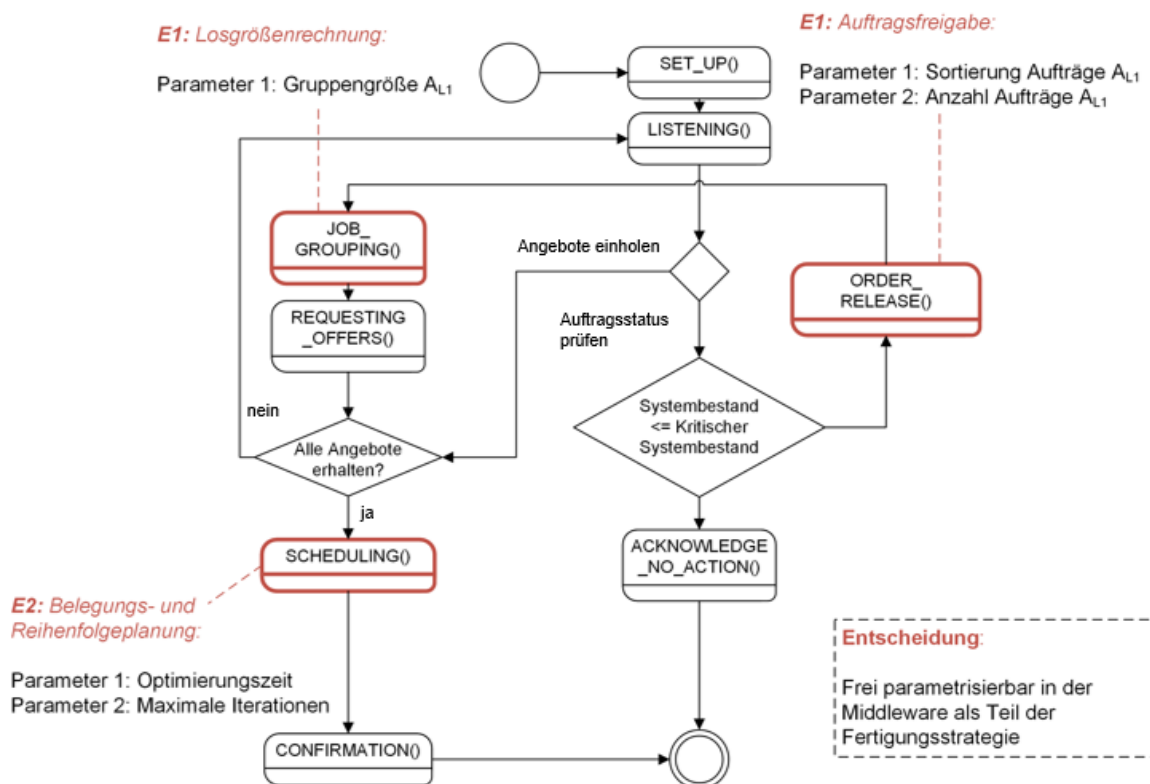


Abbildung 4.12: UML-Zustandsdiagramm Work Center Holon

4.2.2 Station Holon

Das Stationsholon (SH) stellt aufgrund seiner zentralen Rolle zwischen einer untergeordneten und übergeordneten Hierarchie den komplexesten Agenten dar. Nach der Initialisierung im Zustand "SET_UP" empfängt das SH im "LISTENING"-Zustand zunächst Nachrichten von seinen enthaltenen RH, die entweder Updates ohne Änderungen oder mit Änderungen wie z. B. Störungsinformation sein können. Im Falle einer Störungsnachricht wird im "EVALUATE"-Zustand geprüft, ob es sich um eine langfristige oder kurzfristige Störung handelt. Bei einer langfristigen Störung

werden die Ressourcen und alle darin enthaltenen Fähigkeiten (SK_{L3}) im Folgezustand "SKILL_CHECK" überprüft und, nach Abgleich mit dem generischen Fahrzeugbauplan, aktualisiert. Falls eine fehlende Fähigkeit dazu führt, dass eine Fähigkeit SK_{L2} nicht mehr ausführbar ist, wird diese aus der Datenbank entfernt. Jegliche Zustandsänderungen werden dem WCH im Nachhinein als Teil des Zustands "INFORM_UP" mitgeteilt.

Nach der Freigabe durch den WCH kann der SH im "LISTENING"-Zustand Anfragen zu Angeboten verschiedener Losgrößen für Aufträge A_{L2} oder direkte Befehle erhalten. Dies wird mittels der FIPA-ACL Ontologie mitgeteilt. Hierbei handelt es sich um eine erste Grobplanung, da der genaue Zeitpunkt für die Einplanung erst nach der Ablaufplanung des WCH festgelegt wird. Daher wird ein aggregiertes Angebot abgegeben, wobei bei der Belegung der Ressourcen eine einheitliche Grenze angenommen wird.

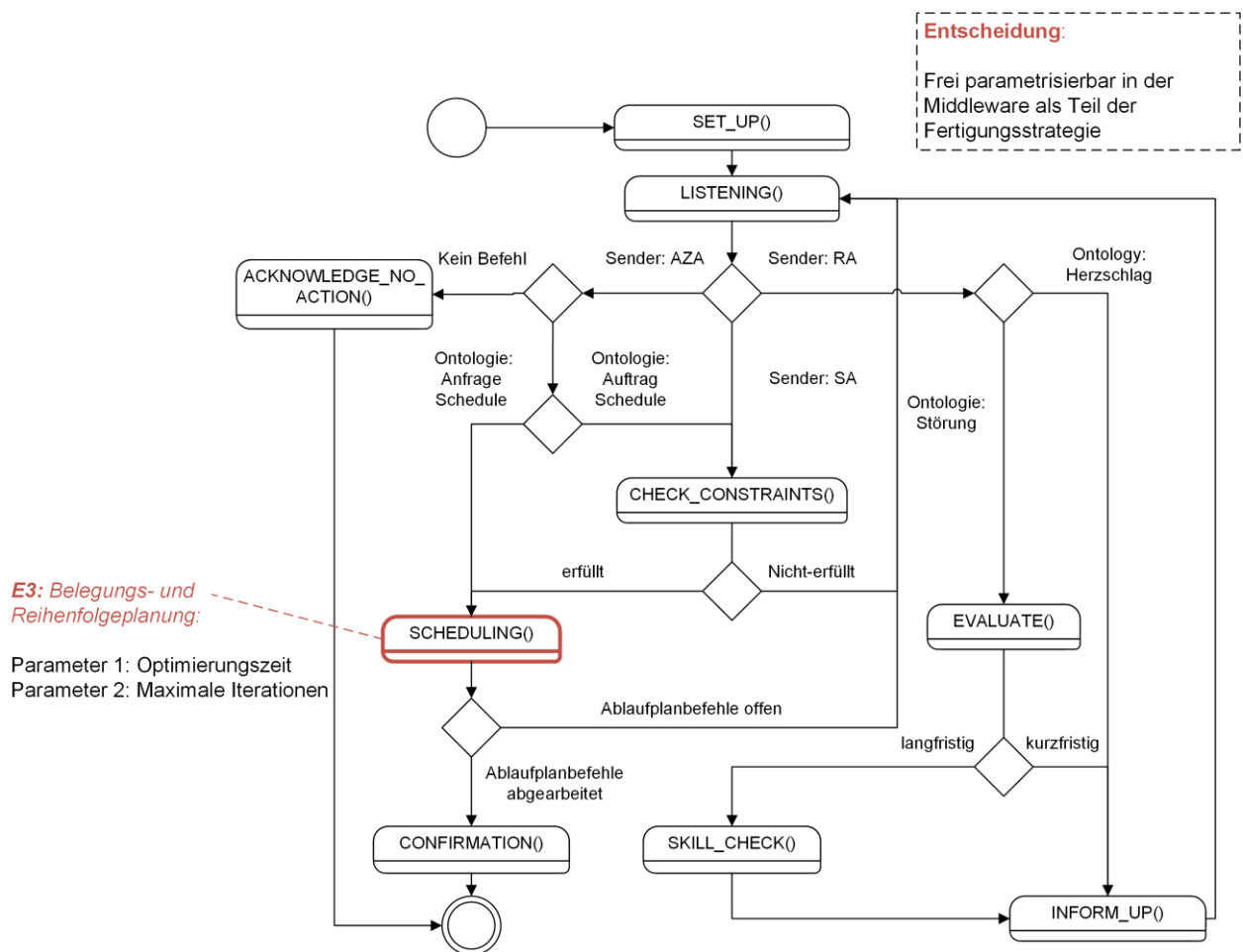


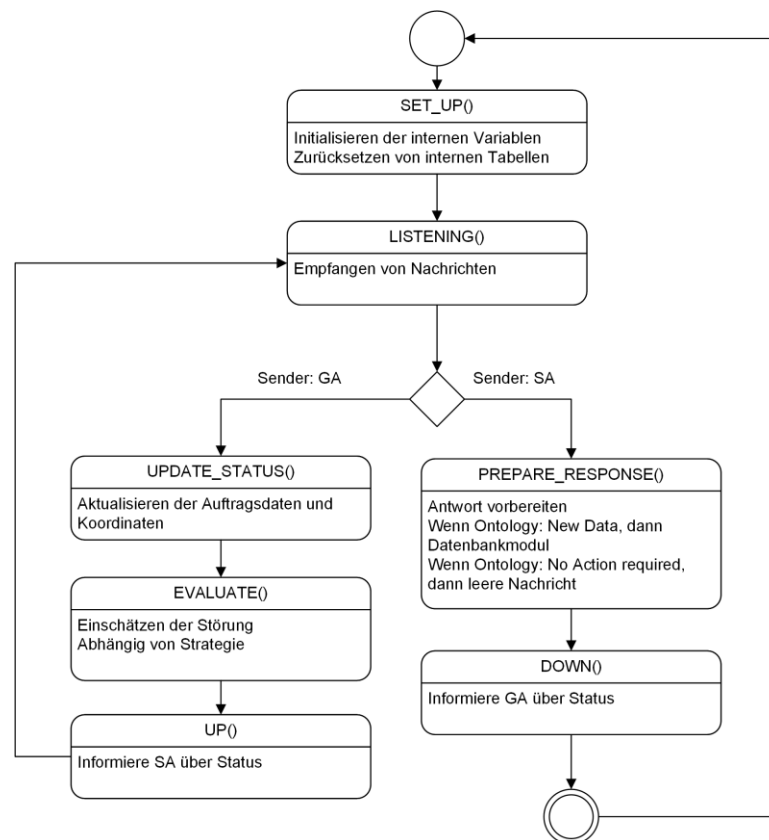
Abbildung 4.13: UML-Zustandsdiagramm Station Holon

Das abgegebene Angebot wird vom WCH für die Koordination der Auftragsgruppen berücksichtigt. Bei Auswahl des Angebots erhält der SH eine Bestätigung für die finale Ablaufplanung, signalisiert durch die FIPA-ACL Ontologie. Nun führt der SH sofort eine Ablaufplanung für das Auf-

tragspaket durch, sofern alle technischen Restriktionen für den Auftragstyp erfüllt sind. Falls Vorgängerbeziehungen der Auftragsgruppe nicht erfüllt sind, geht der SH zurück in den "LISTENING"-Zustand. Die vom WCH zuvor eingeplanten Ablaufpläne müssen ebenfalls vorhanden sein. Im Zustand "CONFIRMATION" werden die betroffenen Ressourcen benachrichtigt, und alle betroffenen Aufträge (A_{L2} und A_{L3}) in der Datenbank werden dem betreffenden Zeitpunkt zugeordnet. Wenn der WCH eine Nachricht ohne Befehl gesendet hat, werden die Ressourcen direkt darüber benachrichtigt.

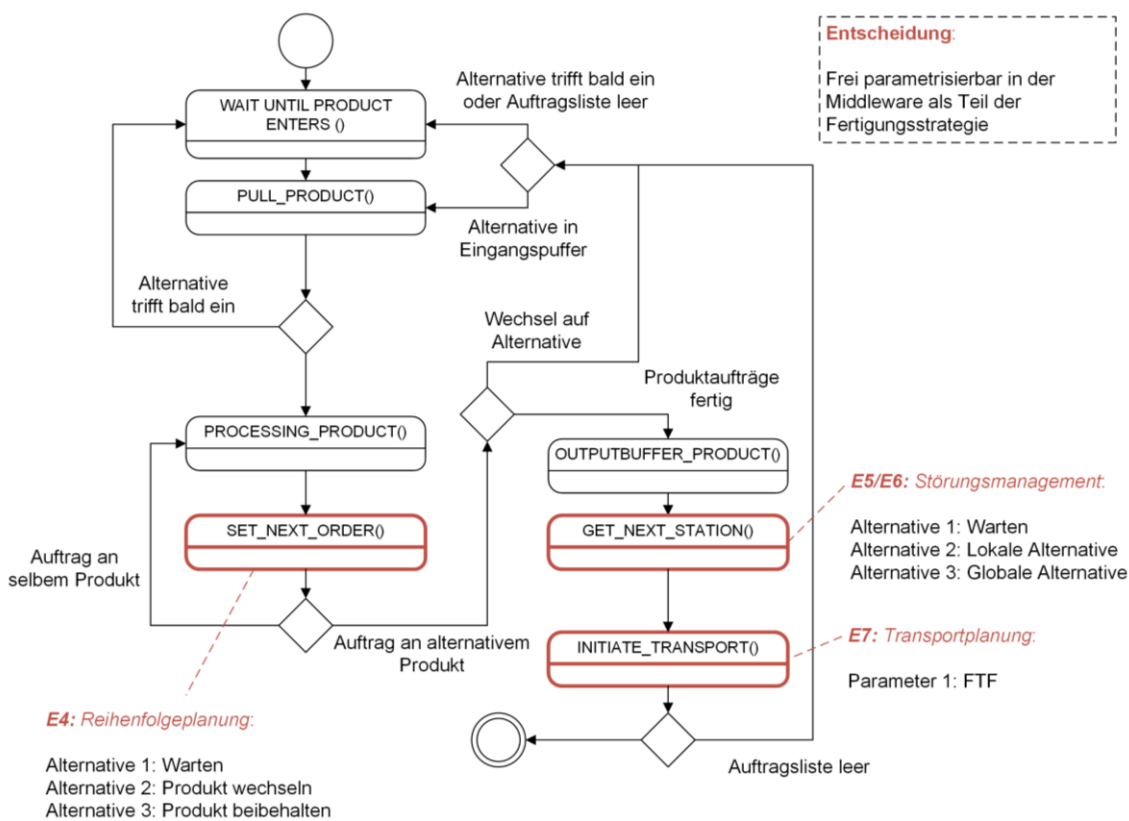
4.2.3 Control Device Holon

Da das Control Device Holon (CDH) in regelmäßigen Abständen Aufträge erhält und direkt mit dem Produkt in Kontakt steht, fungiert es sowohl als Ressourcen Holon des MAS (CDH_{MAS}) als auch als Control Device Holon in der DES (CDH_{DES}). Das nachfolgende Zustandsdiagramm, das den RH_{MAS} betrifft, veranschaulicht die grundlegende und sehr einfache Funktionsweise. Im "SET_UP"-Zustand werden alle Variablen des Agenten, wie offene Aufträge und Anweisungen, zurückgesetzt. Der "LISTENING"-Zustand entspricht ebenfalls dem von SH und WCH. Beim Empfang einer Nachricht des WCH werden die Inhalte des Zustands im JSON-Format überprüft und aktualisiert. Falls eine Störung gemeldet wird, werden entsprechende Maßnahmen je nach Schweregrad abgeleitet. Bei schwerwiegender Störung wird eine Herzschlag-Nachricht an den SA gesendet, andernfalls erfolgt eine Mitteilung über den Ausfall der jeweiligen Fähigkeiten SKL_3 . Im Falle einer Nachricht des SH wird der Inhalt überprüft, und bei der Benachrichtigung über neue Datensätze werden neue Aufträge über den Data Agent aus der Datenbank extrahiert, um die Antwort für den GA zur Übermittlung an den CDH_{DES} vorzubereiten.

Abbildung 4.14: UML-Zustandsdiagramm RH_{MAS}

Die ausführende Instanz der Control Devices in der DES übernimmt die Ausführung der Ressourcenaufträge auf der untersten Hierarchieebene. Die Auftragsreihenfolgen werden aus dem MAS als Auftragspläne zur Ausführung an den RH_{DES} weitergegeben, der eine Reihenfolgebildung und Kapazitätsanpassung mithilfe einfacher Prioritätsregeln vornehmen kann.

Der Auftrag wird über den Zustand „PULL_PRODUCT“ ausgeführt und das Produkt somit bearbeitet, wenn das Produkt im Eingangspuffer der Ressource verfügbar ist und weitere Nebenbedingungen erfüllt sind. Zu den Nebenbedingungen zählen der geplante nächste Auftrag A_{L3} und die geschätzte Ankunftszeit. Hierbei wird die zuvor festgelegte Fertigungsstrategie berücksichtigt. Abhängig von der Abweichung der tatsächlichen Ankunftszeit des nächsten geplanten Auftrags und der Rüstzeit kann das neu eingetroffene Produkt und der damit verbundene Auftrag priorisiert werden. Dazu erfolgt ein Abgleich der entsprechenden Fähigkeiten und Skills zur Auftragsbearbeitung, um einen eventuellen Rüstvorgang einzuleiten. Anschließend wird der Prozess im Zustand "PROCESSING_PRODUCT" ausgeführt.

Abbildung 4.15: UML-Zustandsdiagramm RH_{Des}

Nach der Ausführung wird, abhängig von der gewählten Fertigungsstrategie, die Entscheidung für den nächsten Auftrag im Zustand "SET_NEXT_ORDER" evaluiert. Dabei wird geprüft, ob ein nach globaler Planung vorgelagertes Produkt mit dem entsprechenden Auftrag demnächst eintreffen könnte oder bereits im Eingangspuffer vorhanden ist. Ebenso wird überprüft, ob die Prozesszeit der verbleibenden Aufträge des derzeit gerüsteten Produkts kürzer ist und fortgesetzt werden kann. Nach der Entscheidung wird, je nach Vorhandensein einer Alternative, der nächste Schritt am gerüsteten Produkt vollzogen oder ein neues Produkt zur Auftragsausführung ausgewählt. Wenn dies aufgrund der Fertigstellung des vorhandenen Produkts geschieht, wird das Produkt in den Ausgangspuffer gelegt und ein Transport über TH initiiert. Falls die Auftragsliste leer ist, wechselt das Holon in den Wartezustand oder wählt direkt ein ankommendes Produkt zur Bearbeitung auf der Ressource, wie im Zustandsdiagramm in Abbildung 4.15 dargestellt.

Die Ermittlung der Distanzen für den Transport erfolgt durch das TH auf L1-Ebene auf Basis der euklidischen Distanz zwischen dem Startpunkt und dem verfügbaren FTF innerhalb der Station. Für die L2-Ebene erfolgt dies durch den TH auf Basis der Manhattan-Metrik der Fahrstraßen. Im Gegensatz zur euklidischen Distanz berechnet die Manhattan-Metrik die Distanz auf Grundlage der Summe der absoluten Distanzen der Einzelkoordinaten, die die schachbrettmusterartigen Fahrbahnen des Produktionssystems berücksichtigt. Nach erfolgreicher Registrierung eines Transportauftrags werden Start- und Zielkoordinaten im Zustand "SET_ROUTE_FTF" des FTF festgelegt.

Die Transportholone sind bereits in Plant Simulation integriert, wodurch auf eine detaillierte Darstellung in Form eines UML-Zustandsdiagramms verzichtet wird. Zu Beginn befindet sich das FTF_S/FTF_{AZ} im wartenden Zustand. Bei erfolgreicher Zuweisung eines Transportauftrags durch den RH_{DES} erfolgt die Optimierung der Route auf Grundlage der zu fahrenden Distanz. Falls alle FTF im FTF-Pool belegt sind, wird der Transportauftrag durch den RH_{DES} in eine Warteliste eingetragen. Diese Warteliste wird bei Erreichen des Ziels durch ein FTF überprüft, und bei Verfügbarkeit wird der Transportauftrag dem aufrufenden FTF zugeordnet.

4.3 Implementierung des Optimierungsmodells

Die Lösung des Optimierungsmodells aus 3.2.2 wird durch den Auftragsplanungsagenten unter Berücksichtigung der infrage kommenden Aufträge und Produktionsentitäten über den Service Interface Layer angefragt. Dabei werden ebenfalls die Vorrangbeziehungen mithilfe des Data Agents extrahiert und zur Generierung gültiger Ablauffolgen für den betrachteten Systemraum bereitgestellt. In den folgenden Abschnitten wird die Integration eines Verfahrens zur Lösung des Optimierungsproblems des erweiterten FJSP mit veränderbaren Auftrags- und Ressourcenfolgen beschrieben und bezüglich des Lösungsverhaltens eingeordnet. Aufgrund der Lösungskomplexität und zu erwartenden Problemgröße wird eine Metaheuristik ausgewählt. Zusätzliche verfahrensspezifische Formeln und Notationen werden nicht als Teil der Implementierung vorgestellt sondern können in der verwiesenen Literatur eingesehen werden.

4.3.1 Auswahl und Adaption eines Optimierungsverfahrens

Metaheuristiken sind von einer großen Anzahl an Parametern geprägt, und die Suche nach den optimalen Werten für diese Parameter gestaltet sich als eine nicht-triviale Aufgabe [SG13]. Insbesondere die Dauer der Suche und die Qualität der gefundenen Lösungen können von der richtigen Einstellung dieser Parameter abhängen. Die zur Implementierung der Ablaufplanung relevanten Parameter der Steuerungsarchitektur sind die für die Tabu-Search charakteristischen Begrenzungen der Optimierungszeit oder der maximalen Anzahl an Iterationen.

Das Tabu-Search Verfahren ist eine Metaheuristik für die Lokale Suche. Dauzère und Paulli argumentieren, dass Tabu-Search für Job Shop Probleme besser geeignet ist als beispielsweise Simulated Annealing, da es nach einer festgelegten Zeit mithilfe von definierten Abbruchkriterien gestoppt werden kann [DP97]. Es sei dazu vermerkt, dass der Stand der Forschung seit der Publikation deutlich fortgeschritten und das hier betrachtete Optimierungsproblem gegenüber dem klassischen Job Shop Problem verallgemeinert ist. Es sollte weiterhin beachtet werden, dass das vorzeitige Beenden der Suche zu einer deutlich schlechteren Lösung führen kann als die beste mögliche Lösung, die bei vollständiger Suche erzielt werden könnte. Es gibt eine Reihe von Parametern, deren Werte die Leistung der Suche und die Qualität der gefundenen Lösungen maßgeblich beeinflussen können.

Im Folgenden wird eine Metaheuristik von Dauzère und Paulli zur Lösung des MILP aus Abschnitt 3.4.2 erläutert. Dazu kann das MILP in die Notation von Dauzère und Paulli überführt werden [DP97]. Dieser ursprüngliche Ablauf des Verfahrens wird um die Rüst- und Transportzeiten und neben der Veränderbarkeit der Ressourcensequenzen auch um die Veränderbarkeit der Auftragssequenzen erweitert. Um eine Lokale Suche durchführen zu können, wird zunächst eine gültige Ausgangskonfiguration benötigt, von der aus die Suche gestartet werden kann. Zur Erzeugung einer solchen Anfangslösung wird die natürliche Struktur des Problems in drei Teilprobleme aufgeteilt. Im ersten Schritt wird für jeden Auftrag eine zulässige Auftragsreihenfolge festgelegt, wodurch das verbleibende Problem auf ein FJSP reduziert wird. Durch die Zuweisung von Arbeitsgängen zu Ressourcen wird das Problem weiter auf ein klassisches JSP reduziert. Schließlich müssen unter Berücksichtigung der Auftragsreihenfolgen noch die Ressourcensequenzen bestimmt werden. Die Schritte zwei und drei dieser Konstruktionsheuristik orientieren sich an der Methode, die in [DP97] beschrieben ist.

1. Ermittlung von gültigen Auftragsreihenfolgen:

Der erste Schritt zur Erstellung einer Ausgangskonfiguration besteht darin, gültige Auftragsreihenfolgen unter Berücksichtigung der Vorrangbeziehungen zu ermitteln. Zu diesem Zeitpunkt können die Transportzeiten noch nicht berücksichtigt werden, da den Arbeitsgängen noch keine Ressourcen zugeordnet sind.

2. Zuweisung von Aufträgen zu Produktionsentitäten:

Um die Ressourcenbelegung durchzuführen, werden die Arbeitsgänge zunächst nach der Anzahl der Ressourcen sortiert, die in der Lage sind, einen bestimmten Arbeitsgang durchzuführen. In dieser Reihenfolge wird für jeden Arbeitsgang eine Ressource ausgewählt. Während der Zuweisung wird die Bearbeitungszeit für alle bisher zugewiesenen Arbeitsgänge aufsummiert. Ein Arbeitsgang wird dann der Ressource zugeordnet, deren Gesamtbelastung nach Hinzufügen der ressourcenspezifischen Bearbeitungszeit des Arbeitsgangs am geringsten ist.

3. Ermittlung von Bearbeitungssequenzen:

Um zulässige Bearbeitungssequenzen zu erstellen, werden die Aufträge einzeln hinzugefügt, wobei die Arbeitsgänge des Auftrags hinten an die Ressourcensequenzen angefügt werden. Die Aufträge werden nach der Summe der Bearbeitungszeiten ihrer Arbeitsgänge sortiert. Wenn mehrere Arbeitsgänge eines Auftrags derselben Ressource zugeordnet sind, werden sie in der Reihenfolge eingeplant, die durch die Job-Reihenfolge vorgegeben ist.

Die beschriebene Konstruktionsheuristik führt immer zu einer gültigen Ausgangskonfiguration. Allerdings entstehen durch dieses Vorgehen zunächst große Lücken im Ablaufplan. Die Verbesserungsheuristik beginnt daher mit einer Lösung, die weit vom globalen Optimum entfernt ist. Eine alternative Heuristik zur Erzeugung der Anfangslösung könnte möglicherweise die Leistung der Lokalen Suche verbessern.

Das Verfahren ist in Abbildung 4.16 dargestellt und beginnt mit der Initialisierung verschiedener Mengen, wie z. B. Aufträge, Ressourcen, Vorrangbeziehungen unter Verwendung der ressourcenindividuellen skillbasierten Prozesszeiten. Anschließend wird die Erzeugung einer ersten Lösung mit der im Anhang einsehbaren Konstruktionsheuristik vorbereitet. Beide Nachbarschaftsoperatoren werden gleichberechtigt eingesetzt, um alle erreichbaren Nachbarkonfigurationen in beiden Nachbarschaften der Konfiguration X zu ermitteln. Basierend auf den unteren Schranken für den Zielfunktionswert wird der beste tabu-inaktive Schritt ausgewählt, um eine Nachbarkonfiguration zu erreichen. Danach wird der durchgeführte Schritt zur Tabu-Liste hinzugefügt, wobei der älteste Schritt entfernt wird, wenn die Tabu-Liste ihre maximale Länge erreicht hat. Wenn die neue Konfiguration X besser ist als die bisher beste gefundene Konfiguration X^{best} , wird X^{best} aktualisiert. Wenn das Abbruchkriterium in Form einer maximalen Anzahl von Iterationen ($maxIter$) erreicht ist, wird das Verfahren beendet, sobald der Zähler i die maximale Anzahl erreicht hat. Ein optionales zweites Abbruchkriterium ist die maximale Dauer des Verbesserungsprozesses ($maxTime$), das nach jedem Schritt gegen die fortlaufende Zeitmessung (t^{imp}) geprüft wird, die nach der Generierung der Anfangslösung beginnt. Solange keines der Abbruchkriterien erfüllt ist, wird die Lokale Suche durchgeführt, indem die Nachbarschaften der neuen Konfiguration durchsucht werden.

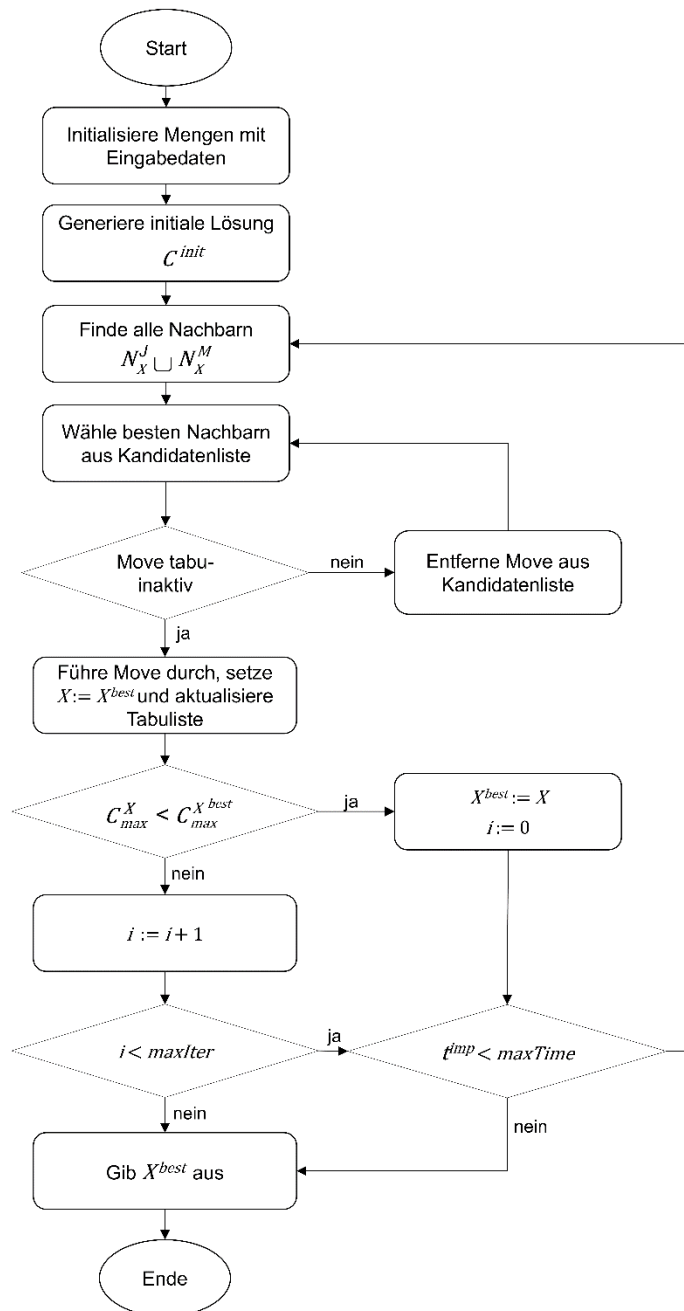


Abbildung 4.16: Top-Level Algorithmus des Tabu-Search Verfahrens

4.3.2 Eigenschaften des Optimierungsverfahrens

Das Lösungsverfahren zur Ablaufoptimierung wurde in Python (Version 3.7.0) implementiert. Für eine skalierbare Ablaufplanung werden die Eigenschaften und das Verhalten des Verfahrens ermittelt. Die Berechnungen zur Charakterisierung des Lösungsverhaltens zur Vorbereitung des anwendungsorientierten Einsatzes wurden auf einem Rechner mit einer Intel Core i7-7700T CPU (Takt: 2,90 GHz) und 32 GB RAM (Takt: 2400 MHz) durchgeführt. Dieses Setup wurde auf bis zu zehn weiteren Rechnern mit identischer Hardware und Software parallel genutzt. Bei der Zeitmessung wird die Gesamtzeit und die Optimierungszeit unterschieden. Die Gesamtzeit t^{cpu} ist

definiert als die Zeit, die von Beginn der Vorbereitung der initialen Lösungsgenerierung an verstreicht. Die Optimierungszeit t^{imp} wird erst beginnend mit Ende der initialen Lösungsgenerierung gemessen.

Aufgrund der fähigkeitsbasierten Zuweisung der Ressourcen, die für die Ausführung eines Arbeitsganges durch ihre Fähigkeit infrage kommen (Menge $\mathcal{M}_0$), ergeben sich nur bestimmte Kombinationen der Faktoren $|\mathcal{M}|$ und $|\mathcal{O}|$ als sinnvolle Instanzen. Andernfalls könnten entweder nicht alle Ressourcen ausgelastet werden (zum Beispiel bei $10m - 5o$) oder einzelne Ressourcen könnten die Varianz der zu bearbeitenden Arbeitsgänge nicht realistisch abdecken (zum Beispiel bei $3m - 15o$). Insgesamt ergeben sich zehn untersuchte Kombinationen von Ressourcen- und Arbeitsganganzahlen:

$3m - 5o, 4m - 5o, 5m - 10o, 6m - 10o, 5m - 15o, 6m - 15o, 8m - 10o, 10m - 10o, 8m - 15o, 10m - 15o$

Es werden Analysen vorgenommen mit bis zu 10 Ressourcen und bis zu 100 Aufträgen mit 5 bis 15 Arbeitsgängen pro Job $|\mathcal{O}_j|$. Einheitlich wurde für die Verbesserungszeit bei jeder Problem Instanz eine Grenze von $\text{maxTime} = 3600$ s (eine Stunde) festgelegt, da große Problem Instanzen mit vielen Arbeitsgängen zu langen Rechenzeiten führen. Zusätzlich wurde das Abbruchkriterium $\text{maxIter} = 300$ festgelegt, um die Suche vor dem Zeitlimit zu beenden, wenn keine weitere Verbesserung der Lösung in Aussicht ist. Die Tabulistenlänge und die Variation der Listenlänge wurden gemäß der Ausführungen in der Ursprungslösung auf die Ressourcen- und Job-Moves angewendet [DP97].

In den Berechnungen zur Charakterisierung des Verfahrens wurden für jede Problem Instanz drei Optimierungsläufe durchgeführt, um die stochastischen Einflüsse auf das Verfahren zu berücksichtigen. Diese Suchläufe wurden mit den in Tabelle 4.1 angegebenen Parametern durchgeführt.

Tabelle 4.1: Parameter und Belegung für die Berechnungen zur Charakterisierung des Verfahrens

Parameter	Belegung
maxTime	3600s
maxIter	300

Einfluss der Problem dimensionen

Die Komplexität der Ablaufplanungsprobleme wird von der Größe des Lösungsraumes bestimmt. Die Abhängigkeiten der Problem dimensionen bestimmen die Eigenschaften des Optimierungsverhaltens. Mithilfe verschiedener Betrachtungen werden die Abhängigkeiten zwischen der Gesamtanzahl der Jobs $|\mathcal{J}|$, der Gesamt arbeitsganganzahl $|\mathcal{O}|$, der Anzahl der Ressourcen $|\mathcal{M}|$ und der benötigten Verbesserungszeit t^{imp} verdeutlicht. Auch die Arbeitsganganzahl pro Job $|\mathcal{O}_j|$ wird berücksichtigt. Die Diagramme in Abbildung 4.17 zeigen erste Ergebnisse, aufgeteilt nach Anzahl der Ressourcen $|\mathcal{M}|$ für die Arbeitsganganzahlen $|\mathcal{O}_j| = 5$ (schwarz), $|\mathcal{O}_j| = 10$ (blau) und $|\mathcal{O}_j| = 15$ (rot).

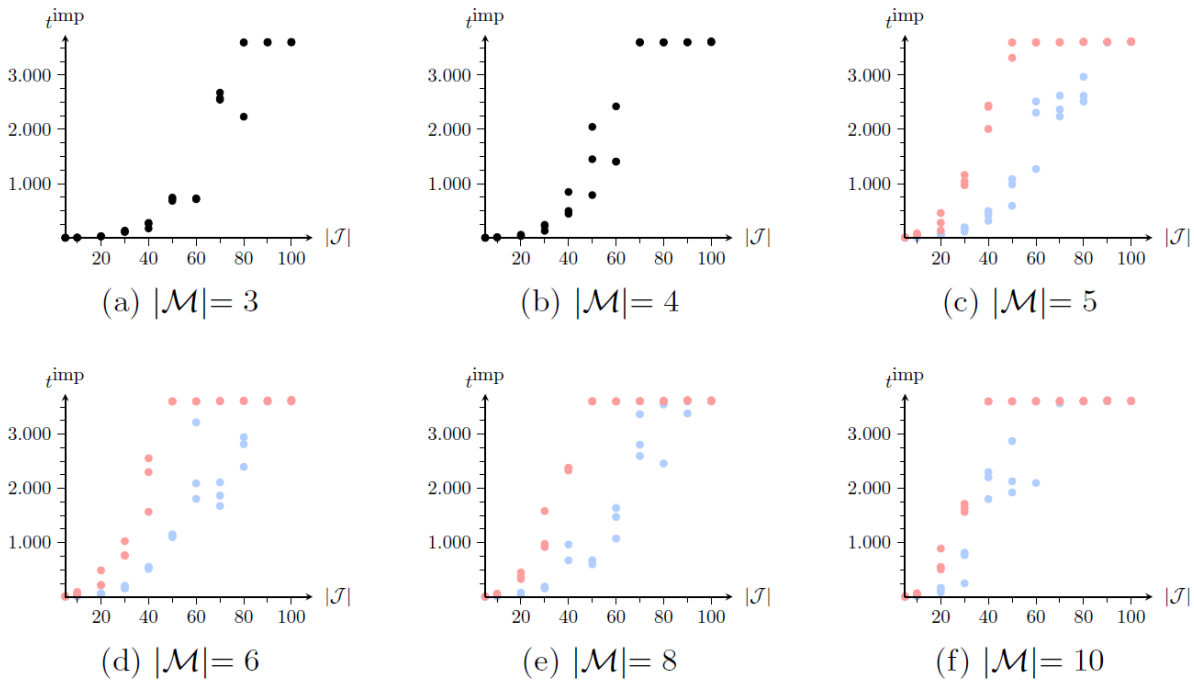


Abbildung 4.17: Nach Ressourcenzahl $|\mathcal{M}|$ gruppierte Plots der Verbesserungszeit t^{imp} in Abhängigkeit zur Anzahl der Jobs $|\mathcal{J}|$

Auf Basis der Abbildung 4.17 werden folgende Vermutungen zum Einfluss der Problemdimensionen aufgestellt:

1. Mit einer zunehmenden Anzahl von Jobs $|\mathcal{J}|$ in der gelösten Problemistanz steigt auch die Zeit für Verbesserungen an. Das Verhältnis scheint nicht-linear zu sein und lässt zumindest eine polynomiale Zunahme vermuten.
2. Eine höhere Anzahl von Arbeitsgängen pro Job $|\mathcal{O}_j|$ führt zu einem steileren Anstieg der benötigten Verbesserungszeit, sodass letztendlich bei gleicher Anzahl von Aufgaben eine längere Verbesserungszeit erforderlich ist.
3. Die Zeit für Verbesserungen scheint offensichtlich von der Anzahl der Ressourcen abhängig zu sein. Obwohl diese Beobachtung in der Grafik nicht deutlich erkennbar ist, wird der Effekt sichtbar, wenn bestimmte Punkte in den Diagrammen festgelegt werden, beispielsweise diejenigen mit $|\mathcal{J}| = 40$ in Abbildung (a) und (b).

Diese Vermutungen können mithilfe von Korrelationsanalysen überprüft und mit Diagrammen visuell nachvollzogen werden. Der häufig verwendete Pearson-Korrelationskoeffizient ist in diesem Fall nicht anwendbar, da die Daten nicht normalverteilt sind und eine nicht-lineare Beziehung zwischen $|\mathcal{J}|$ und t^{imp} vermutet wird. Der Spearman-Korrelationskoeffizient hingegen erfordert weder Normalverteilung noch lineare Zusammenhänge, da er die Korrelation auf der Grundlage der Ränge der Größen berechnet. Daher kann er stattdessen in diesem Kontext verwendet werden [SBS18]. Die Analysen zeigen, dass es eine starke positive Beziehung zwischen der Gruppierung von Arbeitsgängen $|\mathcal{O}|$ und Arbeitsgängen pro Job $|\mathcal{O}_j|$ gibt. Abbildung 4.18 und

Abbildung 4.19 zeigen die Diagramme der angesprochenen Zusammenhänge. Die Korrelationsanalysen sind im Anhang einsehbar.

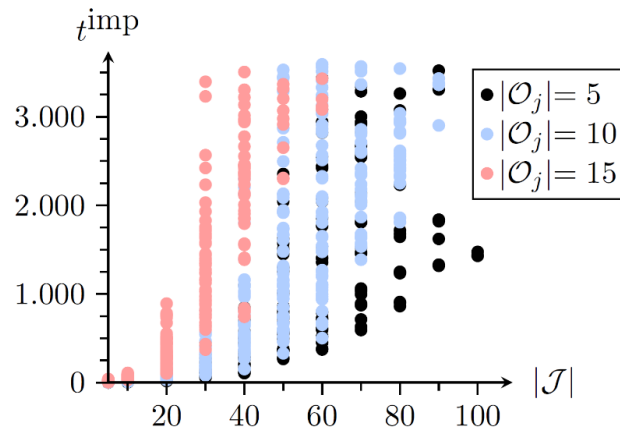


Abbildung 4.18: Verbesserungszeit t^{imp} des Verfahrens in Abhängigkeit zur Anzahl der Jobs

Weiterhin wird die Vermutung überprüft, wie die Arbeitsganganzahl $|\mathcal{O}|$ und t^{imp} zusammenhängen, indem ein weiterer Vergleich durchgeführt wird. Die Gesamtanzahl der Arbeitsgänge lässt sich aus der Multiplikation von $|\mathcal{J}|$ und $|\mathcal{O}_j|$ berechnen. Daher sind beide Größen in diesem Indikator zusammengefasst. Vorausgesetzt, die Spearman-Korrelation ist in diesem Fall aussagekräftig, können die Ergebnisse so interpretiert werden, dass durch die kombinierte Betrachtung von $|\mathcal{J}|$ und $|\mathcal{O}_j|$ durch $|\mathcal{O}|$ ein größerer Anteil der Varianz der Verbesserungszeit erklärt wird. Damit wäre gezeigt, dass auch die Arbeitsgangzahl pro Job von Bedeutung für die Verbesserungszeit ist. Die daraufhin berechnete Korrelation zwischen $|\mathcal{O}|$ und t^{imp} ist größer als die zwischen $|\mathcal{J}|$ und t^{imp} . Die kombinierte Betrachtung von $|\mathcal{J}|$ und $|\mathcal{O}|$ zeigt, dass die Arbeitsgangzahl pro Job für die Verbesserungszeit t^{imp} von Bedeutung ist. Betrachtet man die Gesamtanzahl der Arbeitsgänge, so kann festgestellt werden, dass für Arbeitsgänge $|\mathcal{O}| > 100$ Optimierungszeiten $t^{\text{imp}} > 60\text{s}$ unabhängig der Ressourcenanzahl benötigt werden. Es lässt sich zusammenfassend festhalten, dass ein direkter Zusammenhang von $|\mathcal{O}|$ und damit auch von $|\mathcal{J}|$ und $|\mathcal{O}|$ mit der Verbesserungszeit t^{imp} besteht.

Die Analysen können keine klare Verbindung zwischen der Gesamtzahl der Ressourcen und der Zeit zur Verbesserung der Lösungen zeigen. Durch Variation der Ressourcenanzahl $|\mathcal{M}|$ konnten keine signifikanten Unterschiede festgestellt werden. Abbildung 4.19 lässt jedoch vermuten, dass es innerhalb bestimmter Gruppen, die durch die Ressourcenanzahl einer Probleminstance repräsentiert werden, weniger Unterschiede in den Verbesserungszeiten gibt.

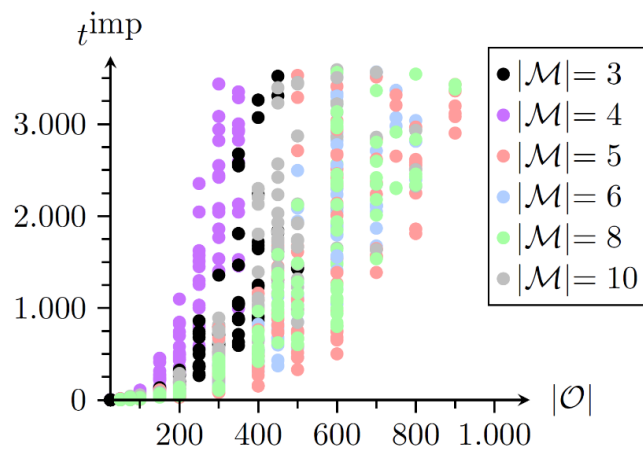


Abbildung 4.19: Verbesserungszeit t^{imp} des Verfahrens in Abhängigkeit zur Gesamtzahl der Arbeitsgänge

Abschließend kann festgestellt werden, dass es eine nicht-lineare Abhängigkeit zwischen der Gesamtanzahl der Arbeitsgänge $|O|$ und der Jobanzahl $|J|$ mit der Verbesserungszeit t^{imp} gibt. Es konnte kein Zusammenhang zwischen der Anzahl der Ressourcen $|M|$ und der Verbesserungszeit t^{imp} nachgewiesen werden. Allerdings könnte die Anzahl der Arbeitsgänge pro Job $|O_j|$, die auf einer Ressource durchgeführt werden können, einen Einfluss auf die Verbesserungszeit haben. Damit hätte zumindest das Verhältnis von Jobanzahl zu Gesamtanzahl der Arbeitsgänge sowie die Verteilung der Prozessfähigkeiten der Ressourcen einen Einfluss auf die Verbesserungszeit.

Konvergenzverhalten des Verfahrens

Das Konvergenzverhalten spielt eine entscheidende Rolle bei der Anwendung des Verfahrens. Ziel ist es, die bisher beste gefundene Lösung stetig zu verbessern. Nach einer nicht vorhersehbaren Anzahl von Schritten tritt jedoch keine weitere Verbesserung mehr auf, entweder weil die globale optimale Lösung erreicht wurde, das Verfahren in einem tiefen lokalen Optimum stecken bleibt oder die aktuelle Konfiguration sich aufgrund beschränkter Auswahlmöglichkeiten nicht mehr ausreichend verändern kann, um neue Lösungen zu finden. Daher konvergiert ein Suchlauf gegen die bisher beste gefundene Lösung. Weitere Iterationen nach der letzten Verbesserung verlängern den Suchlauf unnötig. In der praktischen Anwendung eines Ablaufplanungsverfahrens ist die Abwägung zwischen schneller Lösungsfindung und hoher Lösungsqualität wichtig. Daher sollte die Anzahl nutzloser Iterationen minimiert werden, um längere Lösungszeiten zu vermeiden.

Für die Anwendung des Verfahrens sollte ein geeignetes Abbruchkriterium maxIter gewählt werden können, um die Suche zu beenden, wenn eine Fortführung nicht mehr lohnenswert erscheint. Abbildung 4.20 veranschaulicht die relative Verringerung des Makespan im Verlauf des Verfahrens für einige der größten Instanzen, die im Rahmen der Performanceanalyse für 10 Ressourcen und 15 Arbeitsgängen mit unterschiedlicher Jobanzahl gelöst wurden. Es zeigt die Suchverläufe von elf Instanzen mit unterschiedlich vielen Jobs.

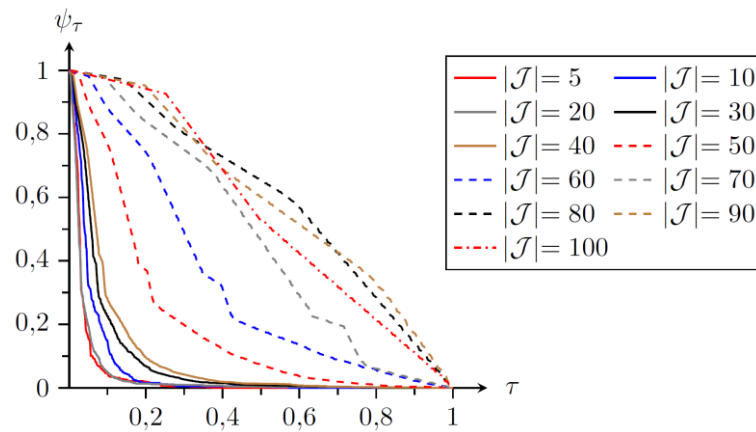


Abbildung 4.20: Relative Verbesserung der Makespan im Verlauf der Suche für eine ausgewählte Problem- instanz mit variabler Auftragsmenge

Die horizontale Achse des Diagramms repräsentiert die aktuelle Iteration, geteilt durch die Gesamtanzahl der durchgeführten Iterationen, auch als relative Suchdauer τ bezeichnet. Die vertikale Achse zeigt die zur aktuellen Suchdauer τ noch verbleibende Makespan-Reduktion im Verhältnis zur Differenz zwischen der initialen und der besten gefundenen Lösung des Suchlaufs, welche mit

$$\psi_{\tau} = \frac{C_{max}^{\tau} - C_{max}^{best}}{C_{max}^{init} - C_{max}^{best}}$$

berechnet wird. Ein Wert von $\psi_{\tau} = 1$ ist dann gleichbedeutend mit $C_{max}^{\tau} = C_{max}^{init}$ und ein Wert von $\psi_{\tau} = 0$ gleichbedeutend mit $C_{max}^{\tau} = C_{max}^{best}$.

Die Grafik verdeutlicht, dass bei einer geringen Anzahl von Jobs ($5 \leq |J| \leq 40$) die Funktion einen konvexen Verlauf aufweist und zu Beginn eine deutlich negative Steigung zeigt. Dies deutet darauf hin, dass in den ersten Iterationen der Suche eine merkliche Reduktion der Makespan erzielt wird. Etwa ein Zwanzigstel bis Zehntel der durchgeführten Iterationen führen zu einer starken Abflachung der Kurven. Dies weist darauf hin, dass das Potenzial besteht, strenge obere Schranken für das Abbruchkriterium *maxIter* festzulegen, was zu einer Verringerung der Lösungszeit führen würde. Bei Probleminstanzen mit mehr als 40 Jobs ($|J| > 40$) weichen die Kurven von einem konvexen Verlauf ab. Dies ist darauf zurückzuführen, dass der Suchlauf für diese Problem- instanzen aufgrund des Erreichens des Zeitlimits abgebrochen wurde. Diese Beobachtung passt zur Analyse der Problemdimensionen, die besagt, dass die Optimierungszeit sich nicht-linear zur Anzahl der Jobs und Arbeitsganganzahl verhält. Insbesondere für $|J| > 70$ befindet sich die Suche noch in einem frühen Stadium, in dem wahrscheinlich weitere Verbesserungen der Makespan möglich wären, wenn dem Verfahren mehr Zeit für die Lösung gewährt würde.

Die nachfolgende Analyse erfolgt aufgrund des konvexen Verlaufs mit starker negativer Steigung zu Beginn der Suche. In diesem Kontext wird untersucht, welche Reduktion der Lösungszeit möglich ist und welche Kompromisse bei der Reduzierung der Makespan dafür in Kauf genommen

werden müssen. Abbildung 4.21 präsentiert ermittelte Werte für die Gap und die relative Suchdauer bei vorzeitigem Abbruch für ein beliebiges $maxIter$ -Wert < 300 im Vergleich zum vollständigen Suchlauf mit $maxIter = 300$. Diese Werte sind Durchschnittswerte über alle untersuchten Suchläufe (alle Instanzen dreimal gelöst), die aufgrund des Abbruchkriteriums $maxIter$ beendet wurden. Die Darstellung zeigt, dass bei einer alternativen Auswahl des Abbruchkriteriums, beispielsweise $maxIter = 100$, durchschnittlich eine Lücke von 2,8% zur besten gefundenen Lösung bei $maxIter = 300$ besteht. Gleichzeitig würde sich die relative Suchdauer verkürzen, sodass das Verfahren nach nur 49,2% der Iterationen im Vergleich zum Suchlauf bei $maxIter = 300$ endet. Da die Dauer jeder Iteration nicht konstant ist, aufgrund unterschiedlicher potenzieller Moves in verschiedenen Konfigurationen, ist der Vergleich nicht exakt. Dennoch kann die Reduzierung der Iterationen hier als eine Annäherung an die eingesparte Rechenzeit bei vorzeitigem Abbruch betrachtet werden.

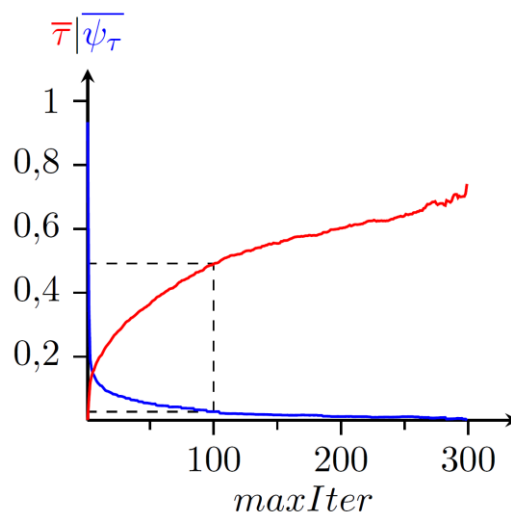


Abbildung 4.21: Durchschnittliche Gap und relative Suchdauer bei Abbruch durch strikteres Abbruchkriterium $maxIter$

Tabelle 4.2 zeigt die Auswirkungen von einigen markanten Werten von $maxIter$ auf $\bar{\psi}_{\tau}$ und $\bar{\tau}$. Beispielsweise lässt sich ablesen, dass für $maxIter = 50$ eine ungefähre Reduktion der Rechenzeit $1 - \bar{\tau}$ von 63,47% bzw. 66,47% erreicht werden kann, wenn ein Gap von 5,41% akzeptiert wird. Die Tabelle 4.2 gibt Aufschluss darüber, wie viele der durchgeführten Suchläufe durch ein strengeres $maxIter$ vorzeitig abgebrochen worden wären.

Die Bestimmung eines guten Kompromisses zwischen Lösungszeit und Lösungsgüte sollte immer unter Berücksichtigung der individuellen Anwendungssituation erfolgen. In dynamischen Situationen, die eine schnellstmögliche Optimierung erfordern, bietet sich der Kompromiss zugunsten der Planungszeit an. In Situationen, in denen ausreichend Zeit zur Verfügung steht, kann die Lösungsgüte priorisiert werden. Die dargestellten Beobachtungen und Analysen dienen als Grundlage für den Einsatz des Verfahrens in dynamischen Fertigungssituationen.

Tabelle 4.2: Auswirkungen weniger strikter *maxIter* auf $\overline{\psi_\tau}$ und $\overline{\tau}$

<i>maxIter</i>	Anzahl Suchläufe	$\overline{\psi_\tau}$	$\overline{\tau}$
5	977	0,1480	0,1521
10	953	0,1140	0,1972
20	904	0,0859	0,2600
50	793	0,0541	0,3653
100	609	0,0280	0,4902
150	418	0,0179	0,5541
200	265	0,0131	0,6020

5 Inbetriebnahme und Validierung

In diesem Kapitel wird die autonome Steuerung des auftragsgesteuerten CPMPS in mittels Simulationsexperimenten in Betrieb genommen und allgemein evaluiert. Zunächst wird der Anwendungsfall in Abschnitt 5.1 vorgestellt, bevor die Simulationsexperimente und Evaluation in Abschnitt 5.2 und 5.3 folgen. Es werden drei Modelle eines Noppenstein-Autos der Marke LEGO® als Produkte P betrachtet [LEGO19; LEGO20]. Insbesondere durch den modularen Aufbau und die Anzahl der Einzelteile ergibt sich eine komplexe Verbaureihenfolge. Die drei Modelle AR (Auto rot), AB (Auto blau) und AW (Auto weiß) sind in nachfolgender Abbildung 5.1 abgebildet.



Abbildung 5.1: Modulare Variante in Rot (AR), Blau (AB), Weiß (AW) aus [LEGO19; LEGO20]

5.1 Vorstellung des Anwendungsfalls

Die Fahrzeuge stellen einen abstrahierten Anwendungsfall mit modular hierarchischer Bauteilstruktur dar. Die Produkte entstammen einer Produktfamilie und besitzen demnach gleiche Bauteile und benötigen sowohl gemeinsame als auch spezifische Fähigkeiten zur Auftragsbearbeitung. Die variantenspezifischen Auftragsstrukturen eines Produktes ergeben sich aus der Zusammensetzung und Aggregation der Teilaufträge. Die nachfolgende Abbildung 5.2 stellt die Implikationen für Produkt, Prozess und Ressourcen sowie die Verantwortlichkeiten in der Agentenstruktur vor.

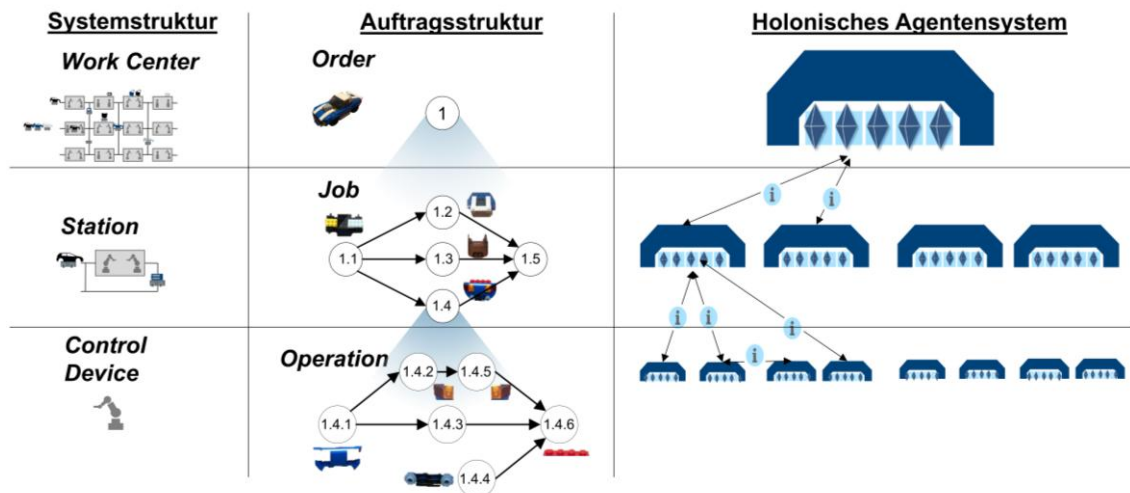


Abbildung 5.2: System- und Auftragsstruktur des Anwendungsfalls mit Holarchie

Die Zerlegung und ebenenspezifische Kapselung ergibt die beispielhafte Darstellung in Abbildung 5.3. Zu sehen ist der initiale Auftrag A_{L3} bestehend aus den Aufträgen A_{L2} und A_{L1} . Die bauteilorientierte Zerlegung wird mit den benötigten Systemfähigkeiten zu einer fähigkeitsbasierten Architektur kombiniert.

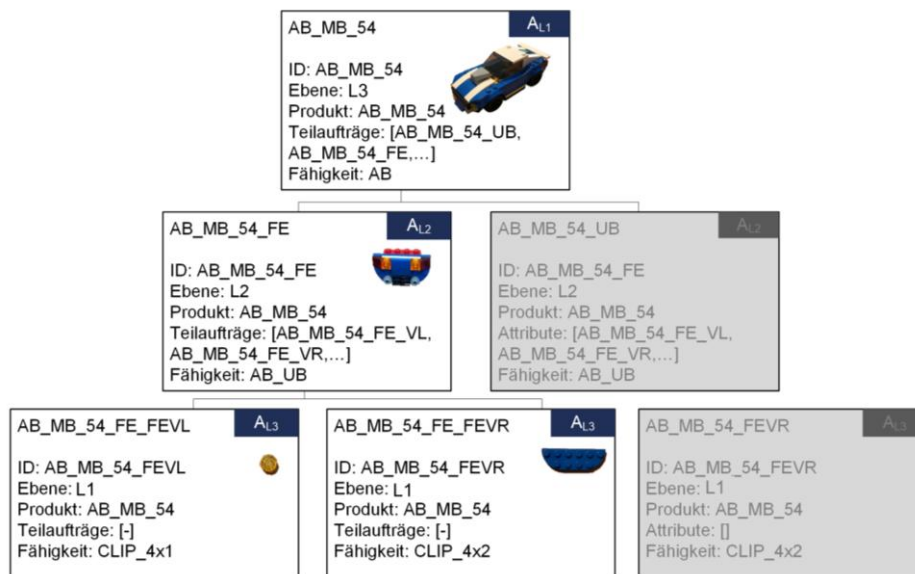


Abbildung 5.3: Beispielhafte Auftragsstruktur Auto blau (AB)

In der vorliegenden Produktionsumgebung stellt die Herstellung von Modulen M für ein Produkt P einen besonderen Fall dar. Gemäß den extrahierten Anforderungen ist es bei individualisierbaren und variantenspezifischen Produkten sinnvoll, einzelne Module zu fertigen, um die gleichzeitige Durchführung von Primärarbeitsgängen an Modulen desselben Produktes zu ermöglichen. Selbst in einer starr verketteten Fertigungsumgebung werden Fahrzeuge nicht vollständig seriell hergestellt, da bestimmte Module, wie beispielsweise Seitenteile, parallel zur Linie zugeführt

werden [He12]. Die identifizierten Modultypen sind in technischen Vorranggraphen für verschiedene Varianten dargestellt (siehe Abbildung 5.4). Insgesamt ergeben sich für die betrachteten Varianten 16 Baugruppen, von denen zwei bis drei eigenständige Module je nach Variante vorhanden sind. Die Module und Baugruppen setzen sich aus ein bis zwölf Arbeitsgängen zusammen. Jeder der A_{L2} -Aufträge verfügt ebenfalls über Teilaufträge mit technischen Vorrangbeziehungen.

Der Fokus liegt auf der selbstorganisierten Auftragsreihenfolgeoptimierung unter Berücksichtigung des Vorranggraphprinzips. Die Instruktionen zur Fertigung der Bauteile sind durch die Anleitung des Herstellers bereits vorgegeben, werden allerdings hinsichtlich ihrer technischen Vorrangbeziehungen geprüft. In diesem Zuge werden die Vorgänger eines Montageschrittes in der Anleitung geprüft und die Abhängigkeiten der Verbaureihenfolge identifiziert. Ist das Fertigen eines Bauteils nur nach Abschluss eines vorangegangenen Prozessschrittes eines vorgelagerten Bauteils ausführbar, herrscht hier eine technische Vorrangbeziehung. Ist dies nicht der Fall, so kann das Bauteil bereits montiert werden. Eine Auswahl für die technischen Vorgängerbeziehungen der Bauteile lässt sich der Tabelle A.1 im Anhang entnehmen. Die Abbildung 5.4 enthält die technischen Vorrangbeziehungen der L2-Ebene.

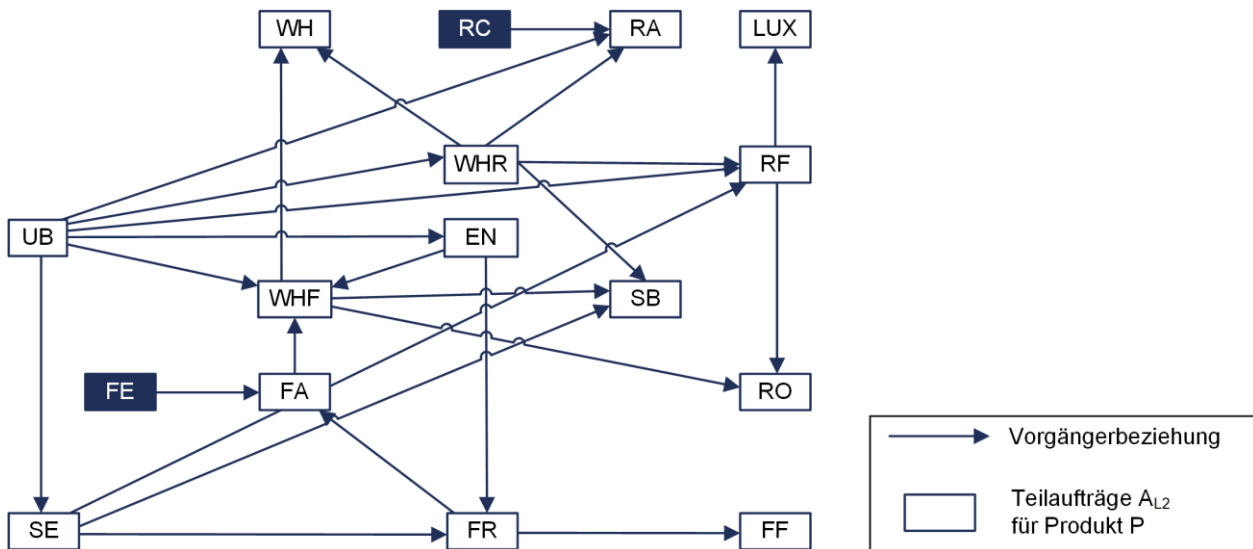


Abbildung 5.4: Vorrangbeziehungen der Aufträge A_{L2} der Variante AB

Eine technische Vorrangbeziehung ist dabei durch eine gerichtete Kante dargestellt. Beispielhaft besitzt der Arbeitsgang RA die technischen Vorgänger RC, UB und WHR. Dies bedeutet, dass die Heckmontage (RA) nicht ohne die Heckklappe (RC), den Unterboden (UB) und das Radhaus hinten (WHR) erfolgen kann.

Der Auftrag wird, wie in Abschnitt 2 beschrieben, auf Control Device Ebene fähigkeitsbasiert als ressourcenspezifischer Skill ausgeführt. Die Ausarbeitung der ausführenden Ressourcen richtet sich nach den Anforderungen des Produktionsbeispiels und den Anforderungen, welche durch die Produkte und Prozesse an das Produktionssystem getragen werden. Orientiert an den Einteilungen der Bauteile zu Baugruppen und dementsprechend der Aufträge A_{L3} zu den Aufträgen A_{L2} sowie der zugrunde liegenden Aufträge A_{L1} mit den grundlegenden Prozessfähigkeiten werden

die initialen Stationen des Produktionssystems entwickelt. Auf Basis der identifizierten 14 bis 16 Module werden 16 Stationen mit insgesamt 63 Ressourcen gebildet. Diese Stationen sind in der Lage, die entsprechenden Aufträge A_{L2} zu bearbeiten. Beispielhafte Stationen sind demnach die Unterbodenstation UB oder die Frontendstation FE. Diese Stationen sind technische Einheiten für die Endmontage und sind aus Sicht des Arbeitszentrums für die Bearbeitung von A_{L2} Aufträgen zuständig. Zur Bereitstellung der dazu benötigten Fähigkeit SK_{L2} müssen sie alle enthaltenen Teilaufträge A_{L1} , welche zu dem jeweiligen A_{L2} Auftrag gehören, abarbeiten können.

Für die Umsetzung der Teilaufträge stehen der Station Ressourcen zur Verfügung. Hierzu wurden die benötigten Fähigkeiten der L1-Ebene auf eine Menge von drei bis acht Ressourcen aufgeteilt. Für einen initialen Ausgleich der möglichen Belastungsfälle wurde ebenfalls berücksichtigt, dass bei einem gleichmäßigen Produktmix die Gesamtzeit, die eine Ressource für alle drei Varianten geblockt wäre, gleichmäßig verteilt ist. Dafür werden die benötigten Fähigkeiten SK_{L1} den Ressourcen zugeordnet, dabei besitzen unterschiedliche Ressourcen individuelle Skills einer Fähigkeit.

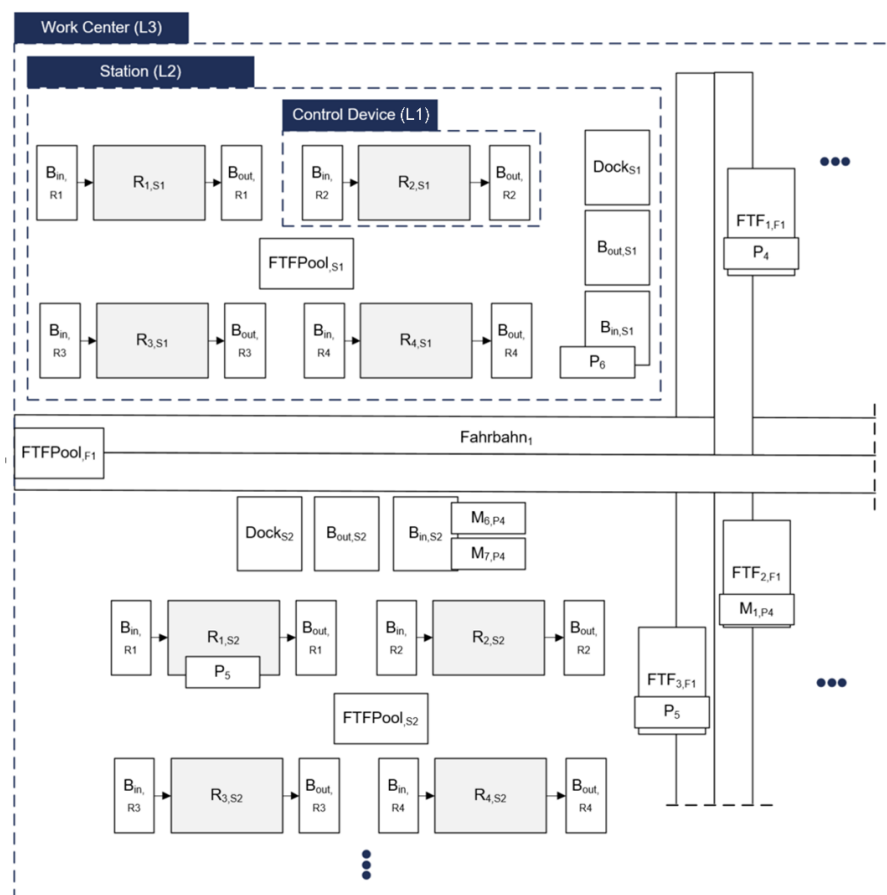


Abbildung 5.5: Abstraktion der Entitäten und Ebenen des CPMPs

Die zweidimensionale Planungsansicht einer einzelnen Station mit drei Ressourcen ist in nachfolgender Abbildung dargestellt.

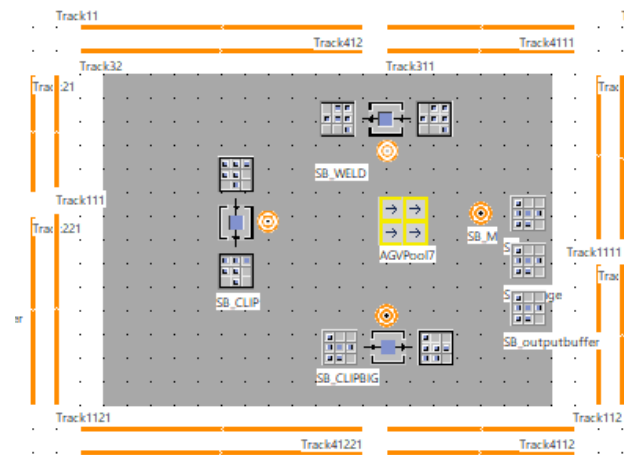


Abbildung 5.6: Ausschnitt einer Station des CPMPs in der DES Plant Simulation

In dieser Abbildung sind beispielhaft die Ressourcen `SB_WELD`, `SB_CLIP` sowie `SB_CLIPBG` der Station `SB` abgebildet. Die Stationen besitzen jeweils Eingangs- und Ausgangspuffer. Die Station ist ebenfalls über diese Puffer, sowie einen FTF_{AZ} -Parkplatz und die Fahrbahnen als Teil des Arbeitszentrums mit den anderen Stationen verbunden.

5.2 Simulationsexperimente

Zur Validierung des Systems für eine dynamische und skalierbare Optimierung werden in diesem Abschnitt exemplarisch Variationen der Parameter getestet. Die betrachteten logistischen Zielgrößen sind die Durchlaufzeit (DLZ) der variantenspezifischen Produkte, die Auslastung der Ressourcen und die Transport- und Lagerzeiten. Die Auftragsfreigabe erfolgt durch das WCH über fixierte Auftragsfreigaberegeln nach Analyse des Work-In-Progress und wird zwischen den Simulationsexperimenten nicht verändert. Vorausgehend wird exemplarisch der Einfluss der Komplexitätsreduktion für eine skalierbare Auftragsplanung aufgezeigt.

Die UML-Zustandsdiagramme aus Abschnitt 4.2 beschreiben die ebenenspezifische Umsetzung der Holone. Das Steuerungsverhalten des Gesamtsystems wird nun mit verschiedenen Variationen getestet. Dabei soll die Evaluation bezüglich der Anforderungen Selbstorganisation, Anpassungsfähigkeit und Skalierbarkeit vorbereitet werden. In dem Simulationsframework ist die Variation der Entscheidung `E4` zur Reihenfolgebildung der Aufträge auf der Ressource und `E5` zur Kapazitätsanpassung im Störfall mit Auswahl alternativer Ressourcen des RH_{DES} , `E3` zur Ablaufplanung des Stationsholons sowie `E1` und `E2` des WCH, parametrierbar. Die Parameter zur Auftragsplanung des SH und WCH werden als Teil von `E2` bzw. `E3` ebenenspezifisch unter Berücksichtigung der Eigenschaften und Verhalten des Verfahrens vorgenommen. Auf Control Device Ebene werden die Auftragspläne durch die Entscheidungen `E4` und `E5` als Teil der Simulation des RH_{DES} mit unterschiedlichen Strategien autonom ausgeführt. Die unterschiedlichen Strategien für die Simulationsexperimente werden in strikt, lokal und global unterschieden und orientieren sich an der Einteilung des Zusammenwirkens von SoS aus Abschnitt 2.1:

- **Strikt:** Diese Strategie ist die Umsetzung des „*gelenkten Verhaltens*“ und befolgt strikt die globale Reihenfolge und Belegung, welche durch den WCH und SH vorgegeben ist. Die Produkte P und Module M werden in der geplanten Reihenfolge ausgeführt, ungeachtet von Kapazitätsengpässen und Maschinenstörungen.
- **Lokal:** Diese Strategie entspricht dem „*anerkannten Verhalten*“ und ermöglicht eine Auswahl des nächsten Auftrags oder ermöglicht im Falle einer Störung die Auswahl einer alternativen Ressource innerhalb derselben Station. Wenn die nächste Ressource für ein Produkt P oder Modul M ausgewählt wird, werden der Status der Ressource und die Kapazität des zugehörigen Puffers überprüft. Im Falle einer Störung werden die technischen Vorranggraphen für den zu bearbeitenden Auftrag A_{L2} in der jeweiligen Station überprüft. Falls keine nachfolgenden Schritte durchführbar sind, wird nach Ressourcen mit der benötigten Fähigkeit SK_{L1} gesucht, die den Schritt durchführen können.
- **Global:** Die Strategie erweitert das „*anerkannte Verhalten*“ der Strategie „Lokal“ um Aspekte des „*kollaborativen*“ und „*virtuellen Verhaltens*“ durch die Möglichkeit des Vorziehens des nächsten Modulauftrags A_{L2} . Hierbei wird zusätzlich zur lokalen Strategie die Belegung der Stationen über das WCH und die technische Vorrangbeziehung des Moduls zum ursprünglich geplanten nächsten Modul geprüft und bei Durchführbarkeit als Alternative gewählt.

In den nachfolgenden Simulationsexperimenten liegt der Fokus auf der Variation der lokalen Strategien und Auftragsgruppengrößen für die Auftragsplanung, wobei die Auswirkungen auf logistische Zielgrößen ganzheitlich präsentiert werden. Das Ziel ist die Nachvollziehbarkeit des Einflusses der lokalen Abweichung von der globalen Planung und Evaluation des Gesamtsystemverhaltens. Zu den betrachteten Parametern gehören Entscheidungsvariablen für die holonischen Agenten, wie Puffergrößen $B_{in, R}$, $B_{out, R}$, $B_{in, S}$, $B_{out, S}$ oder die Anzahl verfügbarer FTFs und FTF_{WCH} . Die Parameter der Simulationsexperimente sind in Tabelle 5.1 detailliert aufgeführt. Die Eingangs- und Ausgangspuffer $B_{in, R} / B_{out, R}$ wurden mit einer Kapazität von 2 ausgestattet, um den Ressourcen die Auswahl zwischen alternativen Produkten und Modulen zu ermöglichen. Dies ist insbesondere relevant, wenn nach der Bearbeitung eines Auftrags eine alternative Option verfügbar ist, die gemäß der globalen Planung bereits hätte bearbeitet werden sollen – eine Situation, die in realen Produktionssystemen einem priorisierten Eilauftrag ähnelt. Um Deadlocks zu vermeiden, wurde die maximale Anzahl zu bearbeitender Modulaufträge in einer Station auf 20 Module begrenzt. Die fixierte untere Systemgrenze des Gesamtsystems wurde an der Anzahl der Ressourcen orientiert und auf 80 festgelegt. Für den Hochlauf des Systems werden initial 110 Aufträge freigegeben. Wenn der Systembestand unter einen Schwellenwert von 80 fällt, wird der Entscheidungshorizont zur Auftragsplanung auf höchster Ebene auf 30 neue Aufträge aus dem Auftragspool eingestellt, wodurch der initiale Systembestand wiederhergestellt wird. Um die Simulationsexperimente durchzuführen, werden drei Szenarien mit unterschiedlichen Zufallsgrößen für Ressourcenausfälle und die Zusammensetzung des Produktmixes im Auftragspool festgelegt. Die drei Szenarien werden als agentenbasierte Simulation mit dynamischer Auftragsplanung mit prädiktiv reaktivem Verhalten ausgeführt. Die Auftragsplanung wird für die Initialplanung mit

110 Aufträgen und die Rekonfiguration/Neuplanung mit 30 Aufträgen parametrisiert. Die Lösungszeiten wurden aufgrund der für den Anwendungsfall relevanten Problemdimensionen und Eigenschaften des implementierten Optimierungsverfahrens aus Abschnitt 4.3.2 ausgewählt.

Tabelle 5.1: Fixierte Parameter der Simulationsexperimente

Parameter	Wert
Kapazität _{max} $B_{in, R} / B_{out, R}$ [P]	2
Kapazität _{max} S [P]	20
Anzahl FTFS	9
Anzahl FTFWCH	45
Untere kritische Systemgrenze [P]	80
Initiale Freigabe [A_{L3}]	110
Vorgriffshorizont [A_{L3}]	30
Ø Verfügbarkeit Ressource [%]	95
Optimierungszeit Work Center Holon [s]	15
Optimierungszeit Station Holon [s]	5

Das Optimierungsproblem wird dazu modular hierarchisch zerlegt, das Prinzip ist in Abbildung 5.7 dargestellt. Dadurch verhalten sich Produktaufträge (Orders) wie Jobs auf L3-Ebene und Jobs wie Arbeitsgänge auf L2-Ebene. Die Auftragsgruppen werden in 5er oder 10er Losgrößen zerlegt. Durch die Zerlegung kann eine hierarchische systemspezifische Belegungs- und Reihenfolgeoptimierung unter Berücksichtigung der Eigenschaften des Optimierungsverfahrens durchgeführt werden. Für den Fall der Neuplanung wird für die Auftragsplanung der L1-Ebene durch die Stationsholone eine Lösungszeit von 5s zur Koordination der Gruppen mit den L1-Orders bzw. Prozessaufträgen (Operationen) auf den Ressourcen gewählt. Zur Neuplanung auf L2-Ebene werden 15s zur Koordination der L2-Orders bzw. Modulaufträge (Jobs) auf den Stationen parametrisiert. Für die Initialplanung der 110 L3-Orders (Produktaufträge) kann allerdings trotz Zerlegung eine kumulierte Optimierungszeit von bis zu 3600s anfallen.

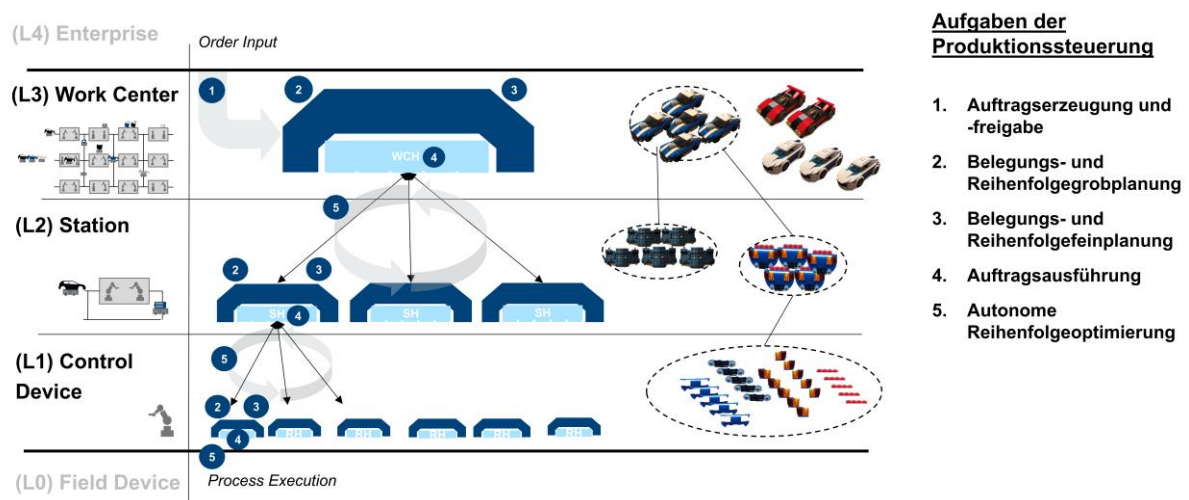


Abbildung 5.7: Exemplarische Zerlegung in Auftragsgruppen unter Berücksichtigung des autonomen Steuerungsprozesses

Durch die Zerlegung des Optimierungsproblems gemäß der Ebenen und Systeme kann die Komplexität des Gesamtsystems reduziert werden für ein hierarchisches Scheduling. Exemplarisch sind Verläufe der Neuplanung der WCH und SH mit den parametrisierten Lösungszeiten in Abschnitt A.VIII im Anhang einsehbar.

Das Systemverhalten wird mithilfe der Betrachtung unterschiedlicher Zielgrößen analysiert. Dabei wird die durchschnittliche Durchlaufzeit (DLZ) pro Produktauftrag und die Gesamtbearbeitungszeit des Auftragspools betrachtet. Außerdem werden die durchschnittlichen Durchlaufzeitanteile pro Produktauftrag an der Gesamtbearbeitungsdauer pro Produkt separat untersucht. Die Durchlaufzeitanteile pro Produktauftrag gliedern sich in Transportzeiten (TZ), Lagerzeiten (LZ) und Prozesszeiten (PZ). Diese sollen genutzt werden, um die Veränderungen zwischen den Durchläufen und das Optimierungsverhalten zu evaluieren. Die Ergebnisse der 18 Simulationenläufe sind in Tabelle 5.2 aufgelistet.

Tabelle 5.2: Parametervariationen der Simulationsexperimente

Strategie	Szenario	Gruppen- größe	Ø Auslastung	Gesamt-bearbei- tungs-zeit [s]	Ø TZ	Ø LZ	Ø PZ	Ø DLZ [s]
global	SZ1	5	0.288	6696.12	0.143	0.563	0.293	1832.748
global	SZ1	10	0.272	6933.64	0.118	0.602	0.280	1934.876
global	SZ2	5	0.283	6839.59	0.150	0.554	0.297	1871.589
global	SZ2	10	0.269	7083.88	0.120	0.604	0.276	1963.610
global	SZ3	5	0.290	6700.26	0.154	0.542	0.304	1806.256
global	SZ3	10	0.266	7093.16	0.147	0.569	0.284	1935.386
lokal	SZ1	5	0.278	7161.59	0.120	0.603	0.277	1942.128
lokal	SZ1	10	0.262	7367.84	0.115	0.619	0.267	2063.251
lokal	SZ2	5	0.272	7205.12	0.147	0.565	0.288	1940.894

lokal	SZ2	10	0.257	7396.87	0.117	0.610	0.273	1984.844
lokal	SZ3	5	0.270	7255.01	0.141	0.579	0.280	1968.842
lokal	SZ3	10	0.253	7491.66	0.113	0.618	0.269	2016.091
strikt	SZ1	5	0.254	7586.28	0.111	0.628	0.261	2096.350
strikt	SZ1	10	0.249	7824.54	0.109	0.635	0.256	2193.704
strikt	SZ2	5	0.258	7572.13	0.115	0.620	0.264	2072.717
strikt	SZ2	10	0.248	7867.39	0.109	0.636	0.256	2158.692
strikt	SZ3	5	0.260	7546.76	0.112	0.626	0.262	2106.686
strikt	SZ3	10	0.256	7623.29	0.109	0.636	0.255	2108.445

Wie in der obigen Tabelle zu sehen ist, sind Unterschiede in der durchschnittlichen DLZ pro Produkt und Gesamtbearbeitungszeit zur Abarbeitung des Auftragspools zu erkennen. Sowohl die durchschnittliche DLZ als auch die Gesamtbearbeitungszeit (*engl. Makespan*) sind in den Fällen, in denen eine globale Ausweichstrategie genutzt wird, grundsätzlich geringer. Zusätzlich ist in der Tabelle zu erkennen, dass bei der Wahl gleicher Szenarien Durchläufe auf Basis einer kleineren Losgröße ebenfalls bessere Werte erreichen. Der Vorgriffshorizont von 30 Aufträgen A_{L3} bleibt gleich, jedoch sind die tatsächlichen Gruppengrößen, deren Belegung und Reihenfolge auf den Stationen optimiert werden, kleiner entsprechend der gewählten Auftragsgruppengröße. Exemplarisch kann der Verlauf der Bestände für die Simulationsdurchläufe im Anhang in Abbildung A.8 eingesehen werden. Ebenfalls enthält der Anhang die Abbildung A.9 mit der beispielhaften Darstellung des variantenspezifischen Bestands der Unterboden-Station, welcher das Prinzip der überlappenden Fertigung als Teil der initialen Ablaufplanung verdeutlicht, aber für diese Arbeit nicht umgesetzt. Die Verläufe der DLZ sind in Abbildung 5.8 präsentiert. Die Ausweichstrategien sind in unterschiedlichen Farben gekennzeichnet mit rot für „strikt“, grün für „lokal“ und blau für „global“.

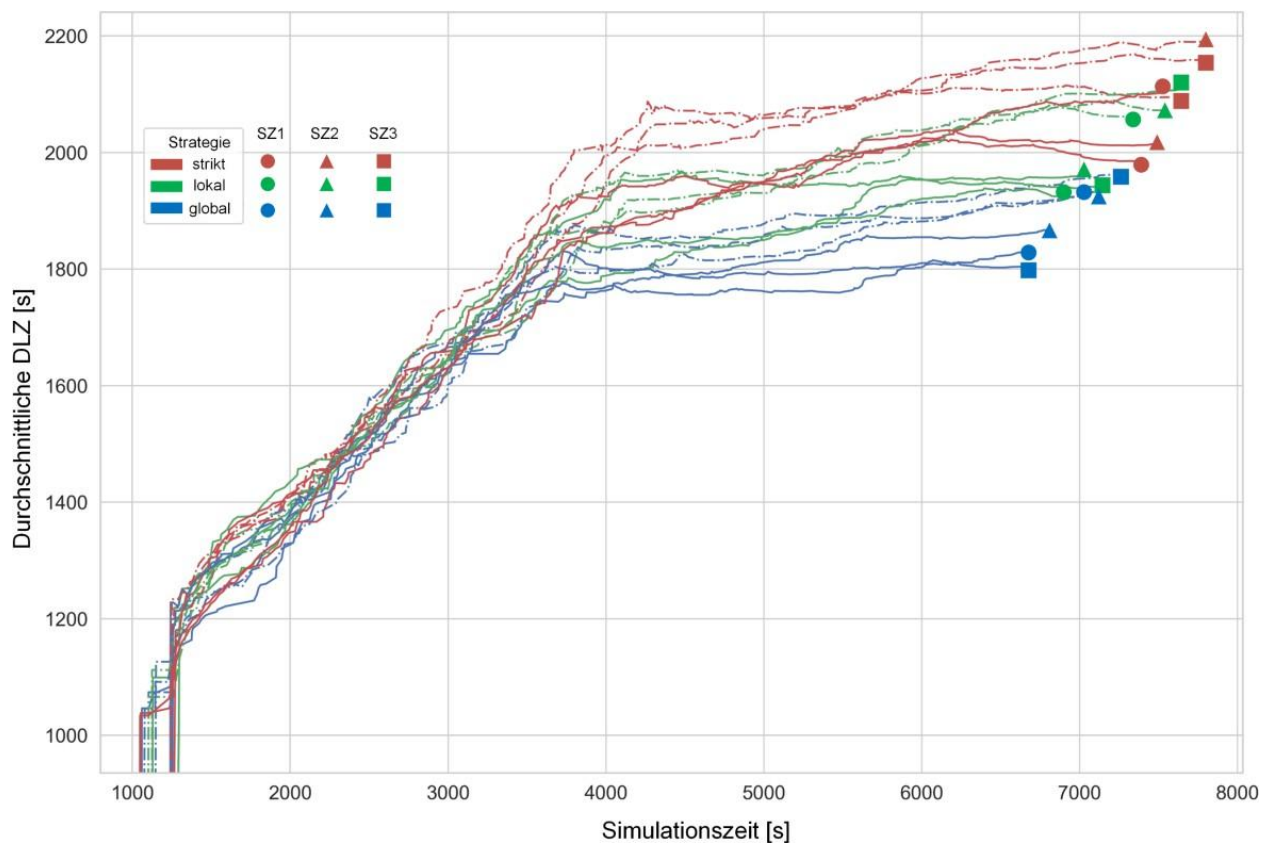


Abbildung 5.8: Durchschnittliche DLZ des Produktionssystems

Verschiedene Linientypen unterscheiden zwischen den gewählten Gruppengrößen bei der Auftragsplanung. Eine gestrichelte Linie repräsentiert Auftragsgruppen der Größe = 5; die durchgezogene Linie Auftragsgruppen der Größe = 10. Die Kurven der Simulationsläufe sind mit dem jeweiligen Szenario gekennzeichnet.

Betrachtet man die durchschnittliche DLZ der verschiedenen Strategien, so wird offensichtlich, dass durch autonome Anpassung das Systemverhalten optimiert werden kann. Es zeigt sich, dass das System unter Ausnutzen des Loose Coupling & Cohesion auf Gesamtsystemebene eine autonome Reihenfolgeoptimierung durch heterarchische fähigkeitsbasierte Interaktion umsetzen kann. Die durchschnittliche Durchlaufzeit der Aufträge wird durch die globale Anpassungsstrategie am meisten verringert im Vergleich. Dies kann darauf zurückgeführt werden, dass die Planung großer Losgrößen zu Engpässen an einzelnen Stationen führt. Die Reihenfolge wird mithilfe der globalen Ausweichstrategie während der Auftragsausführung individuell variiert mit einem begrenzten Vorschauhorizont an unmittelbar verfügbaren Aufträgen.

Die Analyse von Lager-, Transport- und Prozesszeiten zeigt, dass der anteilige Transportzeitbedarf pro Produkt bei der globalen Ausweichstrategie höher ist. Die Verteilung des Transportzeitanteils pro Produkt für die drei unterschiedlichen Strategien ist in Abbildung 5.9 anhand von Dichteverteilungsfunktionen dargestellt.

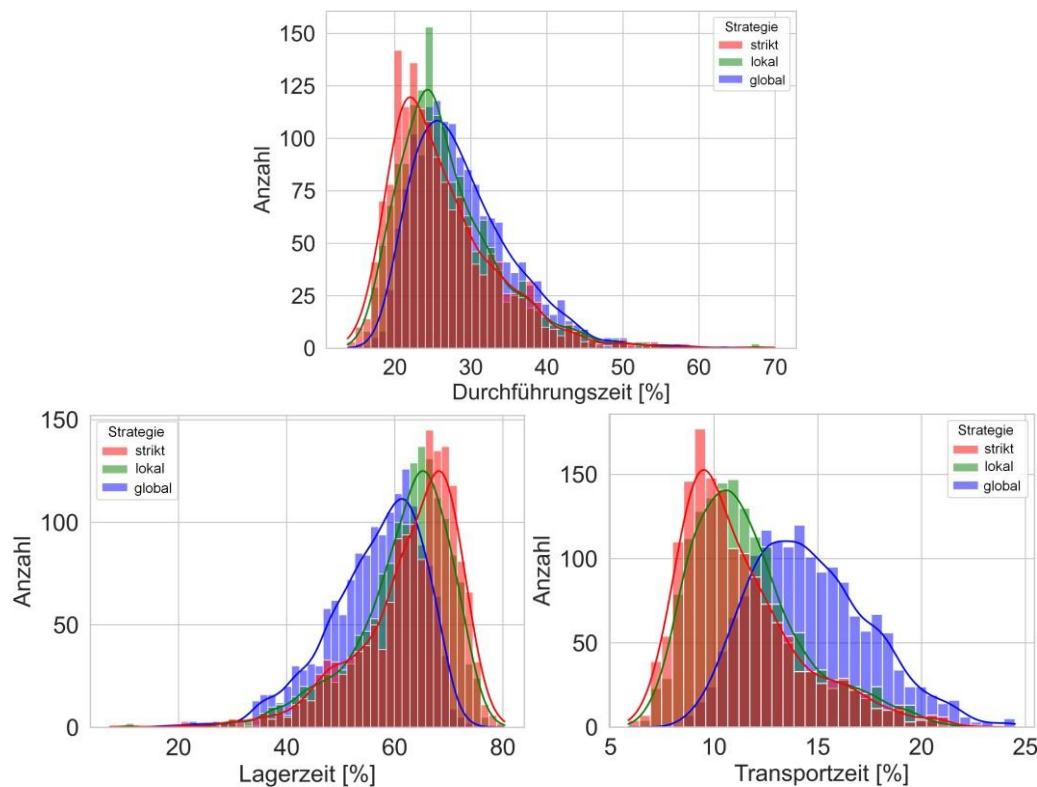


Abbildung 5.9: Zwischenzeitanteile pro Produkt nach Strategie

Die Diagramme zeigen einen durchschnittlichen Anstieg der Transportzeit pro Produkt. Dies ist auf die selbstorganisierte Anpassungsfähigkeit des Systems im Störfall zurückzuführen, bei der auch Ressourcen anderer Stationen zur Auftragsbearbeitung ausgewählt werden können. Die Auswahl alternativer Ressourcen für die Auftragsbearbeitung der Szenarien der lokalen Strategie besitzt keine signifikant höhere Transportzeit als die strikte Strategie. Dies ist darauf zurückzuführen, dass die Produkte bei Ausweichen auf die neue Ressource innerhalb der Station ebenfalls weitere Aufträge abarbeiten, da sie Teil des A_{L2} Auftrages, welcher bereits initial der Station zugeordnet und als Transportaufwand berücksichtigt wurde.

Das Systemverhalten kann darüber hinaus anhand der Ressourcenauslastung analysiert werden. In Tabelle 5.2 wurde ein grundlegender Zusammenhang zwischen Ausweichstrategie und Auslastung festgestellt, welcher in Abbildung 5.10 visualisiert wird. Die Auslastung der Ressourcen, die nach Stationen sortiert und nach dem Namensschema $\langle \text{Station} \rangle_ \langle \text{Bezeichnung} \rangle$ benannt sind, wird dargestellt. Die globale Ausweichstrategie (blau) weist im Allgemeinen eine höhere Auslastung der Ressourcen auf, jedoch ist diese ungleichmäßig verteilt. Einige Ressourcen haben eine Auslastung von weniger als 5% und gehören zu den Stationen FE, RC und RF, die für die Produktion der Module Vorderwagen, Heckabschluss und eines weiteren Heckmoduls ausgelegt sind. Zu beobachten ist insgesamt eine niedrige Auslastung einzelner Ressourcen des Produktionssystems bei gleichzeitig hohen Lagerzeiten der Produkte. Dies ist darauf zurückzuführen, dass die operativ-steuernden Aspekte fokussiert wurden und die gestalterisch-planerischen Aspekte wie z. B. eine Layoutoptimierung in dieser Arbeit vernachlässigt wurden.

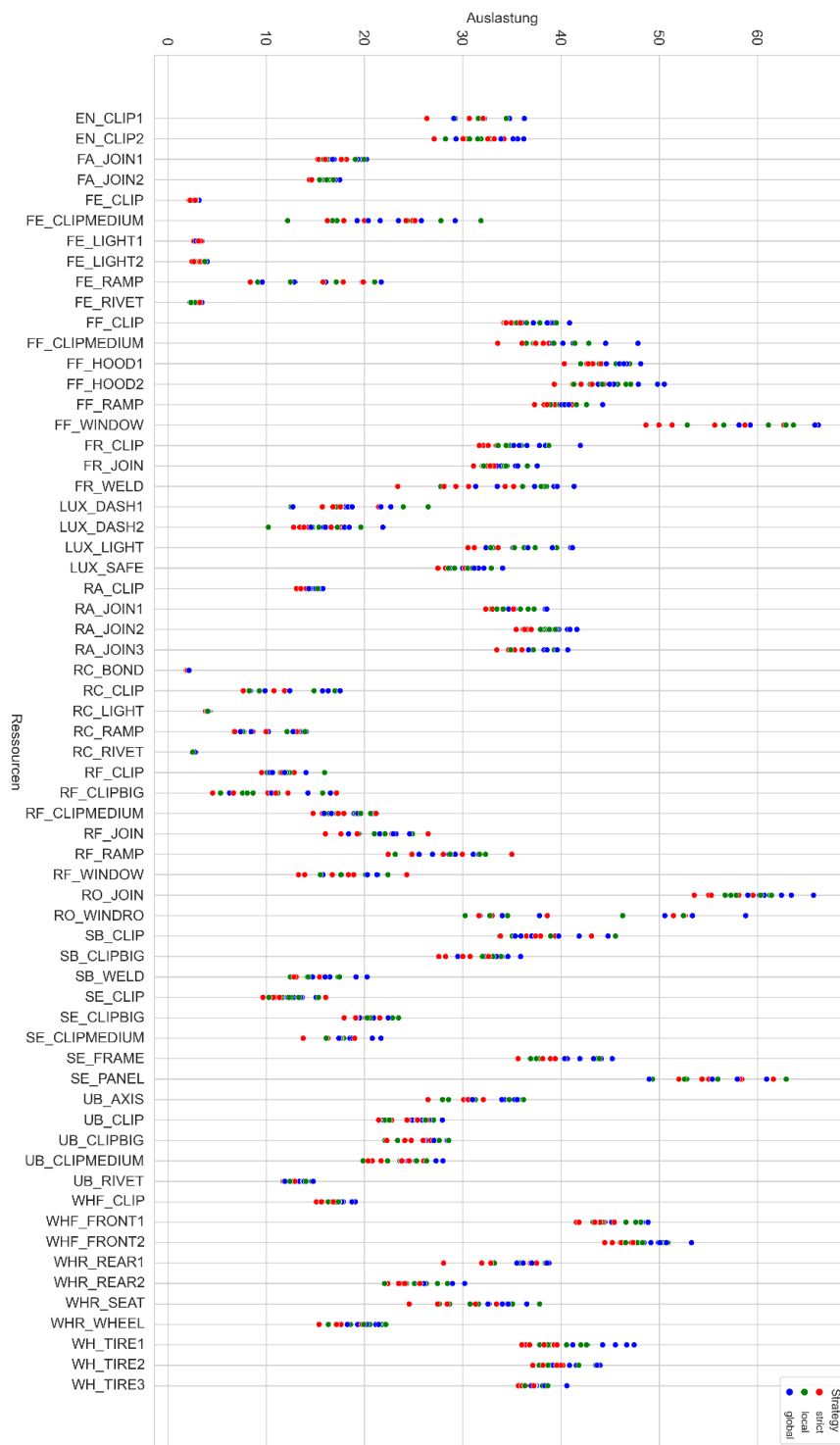


Abbildung 5.10: Durchschnittliche Auslastungen der Ressourcen pro Simulationslauf

Abschließend werden die Simulationsexperimente zusammengefasst und eingeordnet. In den Experimenten wird die Auftragsplanung mit der Auftragsausführung zu einem prädiktiv-reaktiven Systemverhalten kombiniert. Bei der Einordnung wird zunächst auf die Optimierung der prädiktiven Auftragsplanung eingegangen, bevor die reaktive Auftragsausführung mit den unterschiedlichen Ausweichstrategien behandelt wird.

Durch die ebenen- und systemspezifische Zerlegung kann ein heuristisches Optimierungsverfahren (Tabu-Search Verfahren) für Problemdimensionen von CPMPS mit industrieller Systemgröße skalierbar angewendet werden. In [Bu20] wurden bereits bei vergleichsweise kleinen Problemdimensionen eines erweiterten FJSP keine optimalen Ergebnisse mit einem exakten Lösungsverfahren (MILP) erreicht. Das Tabu-Search Verfahren wurde ausgewählt, da es sich, aufgrund der Möglichkeit durch festgelegte Abbruchkriterien nach einer bestimmten Zeit beendet zu werden, besonders gut für Job Shop Probleme eignet [DP97]. Es ist jedoch in der Exploration globaler Optima und in sehr großen Suchräumen eingeschränkt. Durch die modular hierarchische Zerlegung kann eine verteilte lokale Suche erfolgen und liefert deterministische Auftragspläne als Ausgangsbasis für die Simulation des CPMPS mit ungeplanten Ausfällen. Die Modellierung und Implementierung des Optimierungsproblems kann für eine Verbesserung der Lösungsgüte und Lösungszeit in Zukunft noch weiter untersucht werden. Die bisherigen Anstrengungen zu Auswahl und Einsatz des Optimierungsverfahrens für die Auftragsplanung zielten darauf ab, Lösungen in einer angemessenen Zeit für ein CPMPS mit industrieller Systemgröße zu generieren.

Während der reaktiven Auftragsausführung in der Simulation wird die prädiktive Auftragsplanung im dynamischen Betrieb von dem holonischen Agentensystem selbstorganisiert angepasst. Das Ergebnis sind unterschiedliche Ablaufkurven, welche bezüglich der Kennzahlen Durchlaufzeit und Bearbeitungszeitanteil verglichen werden (Abbildungen 5.8 und 5.9). Das implementierte Prioritätsregelverfahren ermöglicht die Parametrierung von Entscheidungsregeln in den Strategien „strikt“, „lokal“ und „global“. Diese optimieren den zuvor generierten Auftragsplan nach der Durchlaufzeit mittels einer Anpassung der Auftragsfolge. Die Strategien unterscheiden sich voneinander bei der Suche nach alternativen Ressourcen zur Auftragsausführung in ihrem Suchradius. Die Entitätsholone entscheiden kollaborativ und ad hoc über Bearbeitungsalternativen auf Basis der unmittelbar verfügbarer bzw. bestätigten Aufträge in ihrem jeweiligen Verantwortungsbereich. Mit den Strategien werden ebenenspezifische Anpassungen realisiert, die den Weg für eine subsidiäre Entscheidungsfindung ebnen. Die globale Strategie liefert dabei die besten Ergebnisse hinsichtlich Durchlaufzeit und Durchführungszeitanteil der Ressourcen. Der Suchradius der globalen Strategie nach alternativen Ressourcen innerhalb des Gesamtsystems ist hier maximal. Die lokale Strategie beschränkt sich bei der Suche nach alternativen Bearbeitungsmöglichkeiten auf Ressourcen auf L1-Ebene innerhalb einer Station, wohingegen die globale Strategie auch die L2-Ebene zur Abfrage der Stationen nach verfügbaren Ressourcen miteinbezieht. Mit der lokalen Strategie wird die Durchlaufzeit verbessert, allerdings erfolgt die deutlichste Verbesserung mit der globalen Strategie. Dies kann mit der geringeren Anzahl an Ausweichalternativen im Vergleich mit der globalen Strategie begründet werden, welche durch den eingeschränkten Suchradius bedingt ist. Darüber hinaus werden auch Wechsel- und Folgewirkungen von Anpassungen der Auftragsfolge innerhalb des Systems noch nicht umfassend berücksichtigt. Z.B. können umrüst-, oder transportbedingte Wartezeiten in der weiteren Auftragsausführung entstehen, die aktuell nicht weiter analysiert werden. Hier bietet sich für zukünftige Untersuchungen der Einsatz von Reinforcement Learning zur Weiterentwicklung hinsichtlich eines vorausschau-

enden Prioritätsregelverfahrens an. Über eine große Bandbreite an Simulationsdurchläufen können Abhängigkeiten der Entscheidungen innerhalb des Systems erlernt und für die Optimierung berücksichtigt werden.

Mit den durchgeführten Simulationsexperimenten wurde der Nachweis über eine erfolgreiche Inbetriebnahme der Steuerungsarchitektur erbracht. Erstmals konnten dynamische CPMPS als modular hierarchische Mehrebenensysteme autonom optimiert werden. Aufbauend darauf kann die agentenbasierte Steuerung mit dem prädiktiv-reaktiven Systemverhalten mit der globalen Strategie für eine subsidiäre Entscheidungsfindung weiterentwickelt werden. Dafür könnten zum einen ergänzende Untersuchungen im aktuellen Framework mit anderer Parametrierung durchgeführt werden. Ebenfalls können auch alternative Optimierungsverfahren für die prädiktive Auftragsplanung oder reaktive Auftragsausführung mit Berücksichtigung weiterer Aspekte (u.a. Layoutoptimierung, Transportoptimierung) modelliert und implementiert werden. Darüber hinaus könnte grundsätzlich untersucht werden, welchen Einfluss die Lösungsgüte eines prädiktiven Auftragsplans in einem dynamischen CPMPS mit geringer Lösungsstabilität besitzt.

5.3 Evaluation

Nach der Inbetriebnahme mittels der Simulationsexperimente in 5.1 wird abschließend die Erfüllung der Anforderungen aus Abschnitt 2.5 bewertet. Die Steuerungsarchitektur wurde als modular hierarchisches CPMPSoS modelliert und als agentenbasierte Simulation implementiert. Die Umsetzung des komplexitätsbeherrschenden Steuerungsprozesses mit dem agentenbasierten Simulationsframework wird nachfolgend zusammenfassend evaluiert.

A1 Selbstorganisation

Die Steuerungsarchitektur kann das Gesamtsystem mithilfe des hybriden Interaktionsmusters über mehrere Ebenen agentenbasiert steuern. Dies wurde durch die Kombination von einer Agentenplattform (JADE) und einer DES (Plant Simulation) agentenbasiert umgesetzt. Nach Bedarf kann das CPMPS selbstorganisiert zustandsaktuell zwischen hierarchischer oder heterarchischer Steuerung wechseln. Die Steuerung unterscheidet zwischen einer selbstoptimierenden prädiktiven Auftragsplanung und einer adaptiven, reaktiven Auftragsausführung zur autonomen Steuerung des CPMPS.

Die Umsetzung des Interaktionsmusters zur agentenbasierten Entscheidungsfindung wurde in dem agentenbasierten Simulationsframework unter Verwendung des FIPA-ACL Standards im MAS hybrid und innerhalb von Plant Simulation mit SimTalk heterarchisch umgesetzt. Mithilfe der Entscheidungsregeln wird selbstorganisiert zwischen hierarchischer Koordination und heterarchischer Kollaboration der Systementitäten gewechselt. Die ermittelte Auftragsfolge des MAS wird auf TCP/IP Basis zur Ausführung auf Control Device Ebene über einen Gateway-Agent an die DES weitergegeben. Das holonische Agentensystem ist in der Lage Abweichungen zu erkennen. Die Ressourcenholone optimieren die Bearbeitungsreihenfolgen daraufhin selbstorganisiert und

auftragsindividuell unter Berücksichtigung der benötigten Fähigkeiten für die Auftragsbearbeitung. Dadurch können die Anforderungen A1.1 und A1.2 wie folgt zusammenfassend bewertet werden.

A1.1 Hierarchische fähigkeitsbasierte Optimierung und Rekonfiguration

Mit dem inter- und intraholonischen Interaktionsmuster wird eine hierarchische Optimierung zur Auftragsbearbeitung zwischen gekapselten holonisch gekapselten Systemräumen ermöglicht. Die Interaktion zwischen Systemräumen unterschiedlicher Hierarchieebenen wird fähigkeitsbasiert hierarchisch koordiniert und liefert die Grundlage für eine globale Optimierung.

A1.2 Heterarchische fähigkeitsbasierte Optimierung und Rekonfiguration

Mit dem inter- und intraholonischen Interaktionsmuster kann eine heterarchische Kollaboration zwischen den gekapselten Systemräumen gleicher Hierarchieebenen ermöglicht werden. Die heterarchische Interaktion wird im MAS während der prädiktiven Auftragsplanung zwischen den Holonen zur Feinterminierung der Ressourcenfolge unter Berücksichtigung der bestätigten Start- und Endzeitpunkte im MAS verwendet. Außerdem wird die heterarchische Interaktion der Agenten auf Ressourcen- und Stationsebene in der DES zur heterarchisch gesteuerten Optimierung der Auftragsausführung der Holone auf L1- und L2-Ebene angewendet.

A2 Anpassungsfähigkeit

Das System ist in der Lage sich an Veränderungen anzupassen. In den Simulationsläufen werden zufällige Ausfälle von Ressourcen simuliert, die Abweichungen vom Auftragsplan verursachen. In den Simulationsläufen wird die Optimierung des Gesamtsystems mittels bestandsorientierter Auftragsfreigabe auf L3-Ebene durch das WCH initiiert. Während der simulierten Auftragsausführung wird das Anpassungsverhalten durch den parametrierbaren Korrekturagenten innerhalb der CDH_{DES} systemspezifisch reaktiv bestimmt. In Simulationsläufen mit unterschiedlichen Szenarien konnte eine Anpassung durch die Prioritätsregeln erfolgen. Die autonome Optimierung und Rekonfiguration des Ablaufes wird durch die selbstorganisierte Entscheidungsfindung innerhalb der Steuerungsarchitektur ermöglicht und erfolgt auf Basis der durch die Datenbank zentral bereitgestellten Auftragspläne und PPR-Informationen. Die Entscheidungsregeln sind über eine Node-Red Middleware durch den Anwender parametrierbar und bieten die Möglichkeit das Systemverhalten zu analysieren. Das System passt sich an die verschiedenen Entscheidungsvorgaben an, wie in den Simulationsexperimenten und Auswertungen der Bearbeitungszeiten und Ressourcenauslastungen gezeigt werden konnte und in A2.1 und A2.2 nachfolgend zusammengefasst wird.

A2.1 System- und ebenenübergreifende Optimierung und Rekonfiguration

Durch Einsatz der entwickelten Interaktionsmuster wird eine agentenbasierte ebenen- und systemübergreifende Optimierung eines erweiterten FJSP mit einem Tabu-Search Verfahren ermöglicht. Die Veränderungsfähigkeiten der Systeme der darunterliegenden Ebene können ebenenübergreifend koordiniert werden.

A2.2 System- und ebenenspezifische Optimierung und Rekonfiguration

Bei der lokalen Reihenfolgeoptimierung während der Auftragsausführung entscheiden die Transport- und Fertigungsressourcenholone zustandsaktuell auf Basis von Prioritätsregeln über die Anpassung der Auftragsreihenfolge oder Ressourcenkonfiguration unter Berücksichtigung der system- und ebenenspezifischen Veränderungsfähigkeiten.

A3 Skalierbarkeit

Mithilfe der Steuerungsarchitektur kann eine skalierbare Auftragsplanung und Auftragsausführung mit einer Optimierung umgesetzt werden. Dazu wurden die Interaktionsmodelle ebenen- und systemspezifisch implementiert und parametrisiert. Das Entwurfsmuster zur komplexitätsreduzierenden subsidiären und rekursiven Optimierung der holonischen Agenten wurde in den Agenten der Agentenplattform und der DES umgesetzt. Das entwickelte interholonische Interaktionsmuster zerlegt die Auftragsbedarfe für eine rekursive Angebotserstellung nach dem Top-Down Prinzip. Die system- und ebenenspezifische Angebotserstellung erfolgt fähigkeitsbasiert nach dem Bottom-Up Prinzip und wird subsidiär durch die Holone koordiniert. Die Auftragsplanung wird mit einem Tabu-Search Verfahren gelöst und als ebenen- und systemunabhängiger Optimierungsservice gekapselt. Die Entscheidung über den Einsatz des Optimierungsservice erfolgt agentenbasiert auf Basis der inter- und intraholonischen Interaktion und der Entscheidungsregeln. Durch die ebenen- und systemspezifische hierarchische Auftragsplanung konnte die Lösungszeit innerhalb weniger Sekunden akzeptable Lösungen für eine Neuplanung generieren. Die Lösungszeit für die Allokation der Auftragsgruppen wurde in Abhängigkeit des Konvergenzverhaltens ebenenspezifisch festgelegt. Ein Trade-Off von Lösungszeit und Lösungsgüte wurde in Abhängigkeit des Konvergenzverhaltens ebenenspezifisch festgelegt. Die subsidiäre Optimierung bei der Auftragsausführung wird heterarchisch durch die Ressourcenholone auf L1-Ebene verhandelt. Nachfolgend ist die Erfüllung der Anforderungen A3.1 und A3.2 zusammengefasst.

A3.1 Rekursive Optimierung und Rekonfiguration

Das inter- und intraholonische Interaktionsmodell konnte das NP-schwere Optimierungsproblem des CPMPS modular hierarchisch zerlegen. Das formulierte Optimierungsmodell wurde mit einer Metaheuristik gelöst, welche über den Service Interface Layer system- und ebenenspezifisch eingesetzt wird. Durch die modular hierarchische Zerlegung des Systems konnte die Metaheuristik innerhalb von Sekunden optimierte Auftragspläne bereitstellen.

A3.2 Subsidiäre Optimierung und Rekonfiguration

Die Funktion einer subsidiären Optimierung wird durch das interholonische Interaktionsmuster ermöglicht. Durch die Implementierung als agentenbasierte Simulation wird eine fähigkeitsbasierte Bottom-Up Angebotserstellung mit entitätsspezifischen Belegungs- und Reihenfolgeoptionen umgesetzt. Die Auswahl eines geeigneten Auftragsausführungsangebots einer Ressource oder Station erfolgt bottom-up durch das Stationsholon bzw. das Work Center Holon. Dadurch werden lokal mögliche Ablaufoptionen zu einem global optimierten Gesamtablauf zusammengefügt.

Wiederverwendbarkeit der entwickelten Lösungen

Abschließend lässt sich zusammenfassen, dass die entwickelte Steuerungsarchitektur die Anforderungen Selbstorganisation, Anpassungsfähigkeit und Skalierbarkeit erfüllt. Die in Abschnitt 3 erarbeiteten Agenten- und Interaktionsmodelle für eine ganzheitliche autonome Entscheidungsfindung in CPMPs können als Grundlage für die weitere Forschung verwendet werden. Die inter- und intraholonischen Interaktionsmuster in Abschnitt 3.3 sind entwickelt worden, um eine autonome Auftragsplanung und -ausführung in CPMPs skalierbar zu ermöglichen. Eine beispielhafte Implementierung der holonischen Agenten, des Optimierungsmodells sowie des Service Interface Layers für den abstrahierten Anwendungsfall wurde in einem privaten Repository in GitHub¹ veröffentlicht und kann auf Anfrage zur Verfügung gestellt werden. Der Code dient ausschließlich illustrativen Zwecken und enthält keine unternehmensspezifischen oder vertraulichen Inhalte. Darüber hinaus kann das agentenbasierte Simulationsframework als Referenz für zukünftige Forschungsvorhaben genutzt werden. Die Komponenten können individuell weiterentwickelt werden und bezüglich ihrer Performanz untersucht werden. Die entwickelte Steuerungsarchitektur trägt zur Schließung der in [PI22] identifizierten Forschungslücke bezüglich der autonomen Optimierung und Rekonfiguration von CPMPs bei.

¹ ([SimonKoma/CPMPS_Holon_MAS: Holonic interaction structure for dynamic scheduling and control of cyber physical matrix production systems \(github.com\)](https://github.com/SimonKoma/CPMPS_Holon_MAS))

6 Zusammenfassung

In der vorliegenden Forschungsarbeit wurde eine holonische Steuerungsarchitektur für CPMPSoS entwickelt und validiert. In Kapitel 2 wurden die Grundlagen und Ansätze zur Erforschung einer autonomen Steuerungsarchitektur unter Berücksichtigung des aktuellen Forschungsstands im Bereich der autonomen Optimierung und Rekonfiguration von CPMPs erarbeitet. Dabei wurden die relevanten Bereiche für CPMPs als veränderungsfähige CPPS betrachtet und vor allem die Steuerungsarchitekturen und die Methoden der Produktionsplanung und -steuerung für die Befähigung autonomer CPMPs analysiert. Das Ergebnis der Analyse ist die Identifikation des Forschungsbedarfs zur Entwicklung einer geeigneten Steuerungsarchitektur mit komplexitätsbeherrschendem Verhalten unter ganzheitlicher Berücksichtigung der verschiedenen Veränderungsfähigkeiten und zustandsaktueller Ereignisse über mehrere Ebenen. Abschnitt 2.6 fasst die gewonnenen Erkenntnisse in den Anforderungen A1-A3 für die weitere Modellierung einer geeigneten Steuerungsarchitektur für Mehrebenensysteme mit systemindividuellen Freiheitsgraden zusammen. Die Anforderungen gliedern sich in Selbstorganisation (A1) Anpassungsfähigkeit (A2) und Skalierbarkeit (A3) für die autonome Steuerung von CPMPSoS definiert.

In Kapitel 3 wurde auf der Basis der Erkenntnisse und Anforderungen aus Kapitel 2 eine holonische Steuerungsarchitektur als agentenbasiertes modular hierarchisches CPMPSoS modelliert, welches über eine agentenbasierte Entscheidungsfindung und serviceorientierte Optimierung verfügt. Die agentenbasierte Steuerungsarchitektur wurde entwickelt, um die ganzheitliche und autonome Optimierung und Rekonfiguration eines CPMPs mit seinen system- und ebenenspezifischen Veränderungsfähigkeiten zu ermöglichen. Hierzu wurde nach der agentenbasierten Modellierungsmethode DACS eine selbstorganisierte Entscheidungsfindung mit hybridem Steuerungsparadigma für einen komplexitätsbeherrschenden Steuerungsprozess (Abschnitt 3.1.1) umgesetzt. Dazu wurde ein Entscheidungsmodell (Abschnitt 3.1), ein Agentenmodell (Abschnitt 3.2) und ein Interaktionsmodell (Abschnitt 3.3) entwickelt. Erweitert wurde die DACS-Methode mit einem MILP-Optimierungsmodell (Abschnitt 3.4) für ein erweitertes FJSP mit integrierter Prozess- und Montageplanung. Das Modell der holonischen serviceorientierten Steuerungsarchitektur wurde in Abschnitt 4 als agentenbasiertes Simulationsframework mit RAMI4.0-konformer Architektur implementiert und umfasst ein holonisches Agentensystem in JADE (Abschnitt 4.1.1), eine Tecnomatix DES mit simulierten Fertigungs- und Transportressourcen (Abschnitt 4.1.2), eine Node-Red Middleware zur Integration der simulierten Ressourcen (Abschnitt 4.1.3), eine SQL-Datenbank für die Verwaltung statischer und dynamischer Daten (Abschnitt 4.1.4) sowie ein serviceorientiertes Optimierungsmodul in Python (Abschnitt 4.1.5). Die Implementierung des holonischen Agentenmodells und des Optimierungsmodells wurde in Abschnitt 4.2 und Abschnitt 4.3 vorgestellt.

Die Inbetriebnahme und Validierung der holonischen Steuerungsarchitektur erfolgte in Abschnitt 5 mittels Simulationsexperimenten für einen Anwendungsfall mit Noppensteinautos. Dazu wur-

den zwei Optimierungsverfahren zur Umsetzung eines prädiktiv-reaktives Systemverhalten kombiniert mit verschiedenen Parametern. Zur Lösung eines erweiterten FJSP-Optimierungsmodells wurde ein Tabu Search-Verfahren integriert, welches in Kombination mit dem holonischen Agentensystem eine skalierbare Auftragsplanung (Scheduling) mit adaptiver Auftragsausführung (Dispatching) ermöglicht. In Simulationsexperimenten in Abschnitt 5.2 werden verschiedene Strategien zur Produktionssteuerung mit der Steuerungsarchitektur getestet und bezüglich der eingesetzten Optimierungsstrategien der Auftragsplanung und Auftragsausführung eingeordnet. Abschließend werden diese in Abschnitt 5.3 bezüglich der Anforderungen aus Abschnitt 2.6 anfordereevaluiert. Es kann gezeigt werden, dass die Steuerungsarchitektur die formulierten Anforderungen A1-A3 zur autonomen Optimierung und Rekonfiguration in einem simulierten CPMPs mit mehreren Ebenen erfüllen kann.

Auf dieser Grundlage wird in den weiteren beiden Abschnitten ein Fazit gezogen und anschließend ein Ausblick für die weitere Forschung gegeben.

6.1 Fazit

Die in Abschnitt 1.2 formulierte Zielsetzung dieser Arbeit, die Erstellung einer Steuerungsarchitektur für autonom gesteuerte CPMPs, konnte erreicht werden. Für die ganzheitliche autonome Steuerung eines CPMPs mit mehreren Ebenen werden aufgrund der system- und ebenenspezifischen Veränderungsfähigkeiten strukturierende Maßnahmen zur Beherrschung der Komplexität benötigt, die bisher in der Forschung zu CPMPs nicht behandelt wurden. In dieser Forschungsarbeit wurde das CPMPs daher als modular hierarchisches SoS strukturiert und es konnte gezeigt werden, dass eine holonische, serviceorientierte Steuerungsarchitektur, bestehend aus einer agentenbasierten Produktionssteuerung und serviceorientierter Auftragsplanung, geeignet ist, um die formulierten Anforderungen an eine autonome Optimierung und Rekonfiguration über mehrere Ebenen zu erfüllen. Durch die Berücksichtigung von Ansätzen zur Erforschung von CPMPs im speziellen, aber auch holonischer Steuerungsarchitekturen im allgemeinen, wurde eine Steuerungsarchitektur modelliert und simulationsbasiert validiert. Die Kombination von einem MAS und einem Tabu-Search Verfahren zur autonomen Auftragsplanung und -ausführung wurde in einer holonischen serviceorientierten Steuerungsarchitektur umgesetzt und konnte als geeigneter Ansatz für eine autonome Optimierung von CPMPs mit industrieller Systemgröße in dynamischen Simulationsläufen validiert werden. Die Steuerungsarchitektur ist in der Lage, die operativen Veränderungsfähigkeiten Umrüstbarkeit, Rekonfiguration und Flexibilität für die autonome Steuerung ganzheitlich zu berücksichtigen.

Die Entscheidungsfindung des CPMPs konnte nach Ansätzen des Systems Engineering als modular hierarchisches System of Systems mithilfe holonischer Agenten und entsprechender Interaktionsmuster strukturiert werden. Diese Interaktionsmuster ermöglichen die Umsetzung eines hybriden Steuerungsparadigmas für eine ganzheitlich agentenbasierte Auftragsplanung und Auftragsausführung des Produktions- und Materialflusses in einem CPMPs. Das System besitzt damit

die grundlegende Fähigkeit, die Vorteile beider Steuerungsparadigmen selbstorganisiert zur Optimierung und Rekonfiguration eines integrierten Gesamtsystems einzusetzen. Die aktuelle Implementierung der modellierten Steuerungsarchitektur setzt eine rekursive Auftragsplanung mithilfe eines MAS in einer Agentenplattform (JADE) und einer in einem Optimierungsservice gekapselten Metaheuristik (Tabu-Search Verfahren) um. Eine selbstorganisierte Auftragsausführung und Reihenfolgeoptimierung erfolgt subsidiär innerhalb der agentenbasierten DES auf Ressourcenebene und führt u. a. zu einer Verbesserung der Gesamtbearbeitungszeit, wie in Simulationsexperimenten gezeigt werden konnte. Die Implementierung des holonischen Agentenmodells und des Optimierungsmodells als agentenbasiertes Simulationsframework konnte die Machbarkeit bestätigen.

Mit dieser Forschungsarbeit wurde der Stand der Forschung für die autonome Steuerung von CPMPs erweitert. Durch die Entwicklung einer holonischen serviceorientierten Steuerungsarchitektur kann ein CPMP mit Berücksichtigung mehrerer Ebenen autonom optimiert und rekonfiguriert werden. Unter Berücksichtigung zustandsaktueller Auftrags- und Ressourceninformationen wird selbstorganisiert über eine Optimierung des Auftragsplans oder eine Rekonfiguration entschieden. Neben den Anforderungen Selbstorganisation und Anpassungsfähigkeit konnte auch die Skalierbarkeit grundsätzlich erfüllt werden. Die Steuerungsarchitektur konnte u.a. das agentenbasierte Framework nach [He02] mit der Eigenschaft einer fähigkeitsbasierten hybriden Interaktion und einem Service Interface Layer für die Befähigung autonomer CPMPs weiterentwickeln. Das Ergebnis ist die autonome Steuerung eines CPMPs mit integrierter Auftragsplanung und Auftragsausführung unter Berücksichtigung seiner verschiedenen und mitunter verschachtelten ebenen- und systemspezifischen Veränderungsfähigkeiten.

Auf dieser Grundlage werden abschließend die Forschungsfragen beantwortet.

Zu Forschungsfrage 1:

Ein CPMP besteht aus mehreren Ebenen mit verteilten Produktionsressourcen. Wie können die Abläufe in einem industriellen modular hierarchischen Produktionssystem unter Berücksichtigung zustandsaktueller Ereignisse autonom rekonfiguriert werden?

In dieser Forschungsarbeit kann festgestellt werden, dass eine autonome Steuerung von CPMPs über mehrere Systemebenen durch die Modellierung als holonisches Agentensystem realisiert werden kann. Die Aufgaben der Produktionssteuerung eines CPMPs können im Sinne eines System of Systems modular hierarchisch ausgeübt werden. Die Agenten übernehmen die Steuerungsaufgaben Auftragsfreigabe, Auftragsplanung, Auftragsausführung und Abweichungsmanagement und werden entsprechend der Systemhierarchie des CPMPs holonisch gekapselt. Die in dieser Arbeit entwickelten Interaktionsmuster zwischen den Agenten und Holonen ermöglichen eine autonome Rekonfiguration unter Berücksichtigung zustandsaktueller Vorranggraphinformationen für eine ganzheitliche Auftragsbearbeitung.

Die Steuerungsarchitektur koordiniert die Auftragsbearbeitung fähigkeitsbasiert mittels PPR Matching und ermöglicht zustandsaktuelle Anpassungen. Die Steuerungsarchitektur setzt eine dynamische Reihenfolgeoptimierung mit prädiktiv reaktivem Verhalten um. Dies wird durch einen selbstorganisierten Wechsel zwischen hierarchischer oder heterarchischer Steuerung ermöglicht.

Zu Forschungsfrage 2:

CPMPS besitzen charakteristische Veränderungsfähigkeiten mit system- und ebenenspezifischen Freiheitsgraden. Wie können CPMPS ganzheitlich optimiert werden und wie kann die Komplexität beherrscht werden?

Das holonische Agentensystem berücksichtigt die system- und ebenenspezifischen Veränderungsfähigkeiten des CPMPS für eine integrierte Auftragsplanung und -ausführung. So konnten die operativ-steuernden Veränderungsfähigkeiten Umrüstbarkeit, Rekonfigurierbarkeit sowie die Flexibilitätsarten Ressourcen-, Produkt-, Prozess- und Routing- bzw. Sequenzflexibilität berücksichtigt werden. Durch die Zerlegung des CPMPS in ein System of Systems können die systemindividuellen Freiheitsgrade auf unterschiedlichen Ebenen berücksichtigt werden. Für die Auftragsplanung von CPMPS über verschiedene Ebenen kann ein erweitertes FJSP mit integrierter Prozess- und Montageplanung für Produkt-, Prozess-, Ressourcen und Routingflexibilität modelliert und optimiert werden. Für eine skalierbare Auftragsplanung kann das erweiterte FJSP durch die Kombination eines MAS und einer Metaheuristik (Tabu-Search Verfahren) ebenen- und systemspezifisch gelöst werden. Die Entscheidungskomplexität des Systems wird mit inter- und intraholonischen Interaktionsmodellen modular hierarchisch zerlegt. Durch die Zerlegung in ebenen- und systemindividuelle Optimierungsprobleme können diese rekursiv und subsidiär optimiert werden. Dadurch kann das NP-schwere Optimierungsproblem des simulierten CPMPS global innerhalb von Sekunden optimiert werden. Bei der Auftragsausführung in einer dynamischen DES mit zufällig verteilten Störungen wird eine selbstorganisierte Anpassung kurzfristig mit dem Prioritätsregelverfahren vorgenommen und kann zu einer Verbesserung der Gesamtbearbeitungszeit im Vergleich zur strikten Planerfüllung während des dynamischen Betriebs führen.

Es kann daher abschließend festgehalten werden, dass die entwickelte Steuerungsarchitektur ein CPMPS unter Berücksichtigung mehrerer Ebenen mithilfe eines holonischen Agentensystems und serviceorientierter Optimierung ganzheitlich autonom steuern kann. Damit wurde eine Grundlage geschaffen, um das Systemverhalten von CPMPS als System of Systems weiter zu erforschen. Nach der Beantwortung der Forschungsfragen wird im folgenden Abschnitt ein Ausblick für die weitere Forschung gegeben.

6.2 Ausblick

Im Rahmen dieser Forschungsarbeit wurde eine holonische serviceorientierte Steuerungsarchitektur für veränderungsfähige CPPS mit drei System- und Produktebenen entwickelt. Während

der Bearbeitung wurden verschiedene Herausforderungen identifiziert, die über die Zielsetzung der Arbeit hinausgehen und Anregung für künftige Forschungsvorhaben bieten können. Im folgenden Abschnitt werden Möglichkeiten zum Einsatz des Systems, aber vor allem auch Optionen zur Weiterentwicklung des Entscheidungsverhaltens und weiterer Implementierungsmöglichkeiten gegeben. Dadurch soll der Reifegrad der modellierten Steuerungsarchitektur für den Einsatz in realen Produktionsumgebungen mit mehreren Ebenen weiter vorbereitet werden. In dieser Arbeit wurde sich auf die Ebenen L1 - L3 mit ihren operativ-steuernden Veränderungsfähigkeiten für die ebenen- und systemübergreifende autonome Entscheidungsfindung fokussiert. In nachfolgenden Arbeiten können weitere Ebenen und bisher nicht betrachtete Veränderungsfähigkeiten für die gestalterisch-planerische und operativ-steuernde Anwendung miteinbezogen werden.

Unter gestalterisch-planerischen Gesichtspunkten sollte die Rekonfiguration in Bezug auf die Wandlungsfähigkeit unter dem Aspekt der Layoutoptimierung auf Gesamtsystem- oder Stationsebene für die Auslegung von CPMPS in weiterführenden Simulationsstudien integriert werden. Das implementierte Simulationsframework kann aber bereits jetzt schon zur Analyse des Systemverhaltens von CPMPS genutzt werden. Das Framework besitzt eine höhere Granularität im Vergleich zu anderen CPMPS-Projekten mit Simulationsbezug. Das Steuerungsverhalten könnte mit der Integration von simulierten oder realen Prozessgeräten weiter validiert werden. Dies kann perspektivisch dabei helfen, netzwerkübergreifend Produktions- und Materialflüsse zu simulieren und manuell oder automatisiert zu optimieren. Durch die Variation der lokalen und globalen Entscheidungsparameter kann das komplexe Systemverhalten des CPMPS unter operativ-steuernden und strategisch-planerischen Gesichtspunkten analysiert werden. Es sei vermerkt, dass die Auslastung der Ressourcen des simulierten CPMPS verhältnismäßig gering ist. Die entwickelte agentenbasierte Simulation bietet die Möglichkeit, die Auswirkungen der Verteilung von Prozessfähigkeiten für die Auftragsbearbeitung auf den Ressourcen und die schwankende Anzahl von Ressourcen, Puffergrößen und FTF-Anzahlen szenarienbasiert zu analysieren. Dies könnte es einem Anwender ermöglichen, eine ineffiziente Auslastung durch Änderungen am Simulationsmodell zu prüfen.

Unter operativ-steuernden Gesichtspunkten kann das Entscheidungsverhalten der Agenten weiter abgestimmt werden. Die ganzheitliche Verbesserung des Produktionsflusses unter Berücksichtigung der operativen Veränderungsfähigkeiten Umrüstbarkeit, Flexibilität und Rekonfigurierbarkeit kann durch den Einsatz von unterschiedlichen Methoden, Technologien oder Systemkomponenten optimiert und für einen realen Einsatz erprobt werden. Die holonische Steuerungsarchitektur kann vor allem als Grundlage genutzt werden, um die Umsetzung einer zustandsaktuellen und vorausschauenden ganzheitlichen Produktionsplanung und -steuerung für CPMPS zu erweitern. Die in dieser Arbeit entwickelten Interaktionsmodelle können für weitere Anwendungen als Entwurfsmuster dienen. Die aktuell implementierten Informationsmodelle und Kommunikationsprotokolle sind für den Einsatz in der industriellen Automatisierung zu prüfen. Orientierung können Standards und Technologien (z. B. in VDI/VDE2193-1 und VDI/VDE2193-2) sowie

Forschungsarbeiten zur Modellierung und dem Matching von Skills und Fähigkeiten für die Anwendung in der flexiblen Fertigung bieten [He21; Vo23]. Ebenso kann durch die Anwendung effizienter Kommunikationsprotokolle und Informationsmodelle die Interaktion bezüglich der Skalierbarkeit weiter optimiert werden [Ge20; Ge23]. Darüber hinaus ist die Integration von echtzeitfähigen Controllern zu erforschen. Zur Steuerung bis auf L0-Ebene ist dazu „harte Echtzeit“ zu berücksichtigen. Eine 1:1 Kommunikationsbeziehung zwischen Holon und Produktionsentität sollte implementiert werden und den aktuell implementierten Gateway-Agenten, der einen interoperablen synchronisierten Datenaustausch zwischen MAS und DES umsetzt, ablösen. In einem nächsten Schritt könnte die Steuerung der Prozessgeräte vollständig über die Agenten der Agentenplattform implementiert werden. Dazu kann auch die Verwendung von alternativen Systemkomponenten für die Umsetzung der holonischen Agenten, wie z. B. Janus, in Kombination mit anderen Implementierungstechnologien für die entsprechenden Architekturlayer des Simulationsframeworks geprüft werden. Ein Zwischenschritt zur Umsetzung in die Praxis könnte die Nutzung der Simulationsumgebung als Emulationsumgebung sein, um das Systemverhalten unter Echtzeitbedingungen zu untersuchen. Dazu sollte das Kommunikationsverhalten im Kontext Ausfallsicherheit oder Error-Handling für das Abweichungsmanagement betrachtet werden. Es kann außerdem erforscht werden, wie die Entscheidungsfindung in Bezug auf die Verteilung der Rechenkapazität innerhalb des Netzwerkes organisiert werden sollte. Dazu können Analysen bezüglich der Eignung von Cloud-Konzepten in Kombination mit anlagennaher Prozesssteuerung in holonischen Steuerungsarchitekturen weiter vertieft werden [DCT21].

Das agentenbasierte Entscheidungsverhalten zur ganzheitlichen autonomen Optimierung und Rekonfiguration könnte ebenfalls in weiteren Forschungsvorhaben weiterentwickelt werden. In Anlehnung an die Kategorien eines SoS konnten unterschiedliche Systemverhalten durch die Verfolgung von verschiedenen Strategien in den Simulationsexperimenten grundsätzlich umgesetzt werden. Der autonome Steuerungsprozess für die modular hierarchische Zerlegung ermöglicht eine skalierbare dynamische Optimierung mit gültigen Auftragsplänen. Um Aussagen über die Leistungsfähigkeit des Systems treffen zu können, sollten die Entscheidungen und ihre Abhängigkeiten untereinander in Bezug auf die Erreichung globaler Zielwerte weiter analysiert werden. Das rekursive und subsidiäre Optimierungsverhalten wurde bisher vor allem als Teil der Auftragsplanung implementiert. Das subsidiäre Entscheidungsverhalten zur Optimierung als Teil der Auftragsausführung beschränkt sich aktuell aufgrund der simulationsbasierten Umsetzung auf L1-Ebene und könnte auf die L2- und L3-Ebene ausgeweitet werden. Dazu wird eine Implementierung außerhalb einer Simulationsumgebung empfohlen, um verschiedene Zeithorizonte in „weicher“ Echtzeit berücksichtigen zu können. So könnte z. B. die Entscheidung zur Anpassung gestuft hierarchisch freigegeben werden, wie es bei [He02] erfolgt. Auch kann die derzeitige Umsetzung der Auftragsfreigabe von Produktaufträgen auf L3-Ebene feingranularer mit einer ebenen- und systemspezifischen Freigabe von Teilaufträgen auf L2- und L1-Ebene bezüglich der Auswirkungen auf den Produktions- und Materialfluss des Gesamtsystems geprüft werden. Ebenso könnten alternative Freigabemechanismen untersucht werden. Zur Reduktion der Komplexität werden die Aufträge aktuell hierarchisch sortenrein aggregiert. Die Planung der aggregierten Auftragsgruppen auf L2- und L3-Ebene führt zu einer Unschärfe, da die Überlappung der einzelnen Aufträge

nicht berücksichtigt wird. Dies begünstigt zwar eine skalierbare Planung, verhält sich jedoch nachteilig in Bezug auf die Termintreue. Es ist zu analysieren, wie der Kompromiss zwischen der Aggregation und der Granularität aussehen sollte, um eine bestmögliche Lösungsgüte in kürzestmöglicher Zeit zu erreichen. In einem komplexen und dynamischen System ist die Lösungsstabilität gering und bedarf ständiger Anpassungen im Sinne eines prädiktiv-reaktiven Systemverhaltens. Aufgrund der Wechselwirkungen mit nach- und vorgelagerten Prozessen ist eine Planung des Produktions- und Materialflusses aber zwingend erforderlich. Es ist zu prüfen, welcher Vorschauhorizont und welche Lösungsgüte für den dynamischen Betrieb den bestmöglichen Kompromiss bietet. Dies gilt sowohl für den aktuellen Betrachtungsraum als auch für Bereiche und Aufgaben außerhalb des betrachteten Systemraums, wie z. B. der Materialbereitstellung. Hier deutet sich an, dass die Bewertung nach Effizienz- und Resilienzkriterien erfolgen sollte, die mit den Anforderungen nach Selbstorganisation, Anpassungsfähigkeit und Skalierbarkeit in dieser Arbeit bereits angestoßen wurde.

In weiteren Vorhaben sollten für die ganzheitliche Bewertung auch weitere Kennzahlen berücksichtigt werden. Aktuell strebt das holonische Agentensystem lediglich die Minimierung der Gesamtbearbeitungszeit an. Die Wechselwirkungen mit weiteren Zielgrößen wie FTF-Anzahlen, Pufferkapazitäten oder Ressourcenanzahl können weiter analysiert werden. Die Modellierung des erweiterten FJSP könnte durch die Integration weiterer Abhängigkeiten und Flexibilitätsarten eine detailliertere Auftragsplanung vornehmen. Orientierung dazu bietet die Übersicht von [Li22] mit weiteren Ansätzen zur Integration der Produktionsplanung, Instandhaltungsplanung oder Transportplanung. Auch könnten in Zukunft Materialbereitstellungsstrategien berücksichtigt werden. Die lokalen Entscheidungsregeln, die mit Prioritätsregeln umgesetzt wurden, können durch den Einsatz von Methoden des Maschinellen Lernens oder des Reinforcement Learning [PYR21] zu vorausschauenden Prioritätsregeln erweitert werden und die Steuerungsarchitektur mit lern- und wissensbasierten Ansätzen ergänzen. In zukünftigen Forschungsvorhaben könnten weitere alternative Optimierungsverfahren zur Auftragsplanung oder -ausführung über den Service Interface Layer integriert und die Auswirkungen auf das Entscheidungsverhalten des Systems analysiert werden.

Die adaptierte Metaheuristik besitzt in dieser Arbeit eine untergeordnete Rolle, nichtsdestotrotz können weitere Forschungsmöglichkeiten in diesem Zusammenhang genannt werden. Generell wurde ein existierendes Verfahren zur Lösung des Optimierungsmodells eines erweiterten FJSP adaptiert und implementiert. Unter Berücksichtigung der Eigenschaften und des Verhaltens wurde die Lösungszeit für eine ebenen- und systemspezifische Anwendung parametrisiert. In Zukunft könnten noch weitere Alternativen zur Parametrierung der Lösungszeit geprüft werden. Für eine verbesserte Performance des Verfahrens könnten die Abbruchkriterien in Abhängigkeit der unteren Schranken und Anzahl der Iterationen genauer untersucht werden. Die aktuelle Parametrierung erfolgte nach Beobachtung des Optimierungsverhaltens ebenenspezifisch. Es könnte analysiert werden, wie das Systemverhalten durch eine Verschiebung zugunsten einer besseren Lösungsgüte oder einer kürzeren Lösungszeit beeinflusst wird. Die Erkenntnis, dass zu Beginn des Verfahrens eine deutliche Verbesserung möglich ist, könnte genutzt werden, um bei

einem Abfallen der Verbesserungen einen Mechanismus zur Diversifikation auszulösen. Dies könnte die Effizienz des Suchverfahrens erhöhen. Die Erarbeitung effizienter Heuristiken für die Erzeugung von Startlösungen und die Entwicklung besserer unterer Schranken für die Berechnung der optimalen Makespan sollten ebenfalls in Betracht gezogen werden, um die Qualität der Lösungen und die Rechenzeit weiter zu verbessern. Der Einfluss der Problemdimensionen kann in weiteren Analysen vertieft werden. Vor allem der Einfluss einer größeren Anzahl von Ressourcen und die Verteilung der Fähigkeiten zur Auftragsbearbeitung könnte weiter analysiert werden.

Die modellierte Steuerungsarchitektur und das implementierte agentenbasierte Simulationsframework können als Grundlage für die weitere Forschung zur autonomen Steuerung von CPMPS genutzt werden. Das holonische Agentensystem kann die operativ steuernden Veränderungsfähigkeiten Umrüstbarkeit, Flexibilität und Rekonfigurierbarkeit in einem CPMPS zur autonomen Optimierung und Rekonfiguration berücksichtigen. Auf dieser Basis können weitere Ebenen implementiert und das Kommunikations- und Interaktionsdesign zur Vorbereitung eines Einsatzes in realen Produktionsumgebungen weiter ausgestaltet werden.

Anhang

A.I. UML-Klassendiagramm

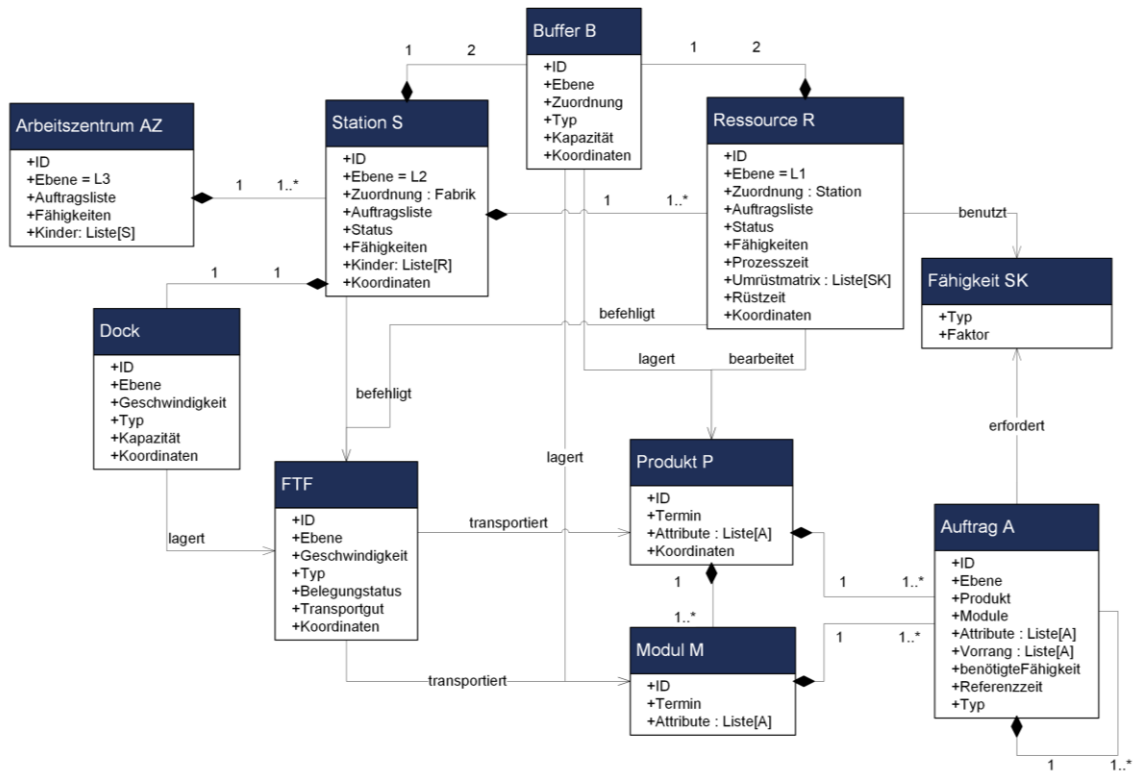


Abbildung A.1: UML-Klassendiagramm des implementierten Informationsmodells

A.II. Technische Vorrangbeziehungen

Die nachfolgende Tabelle A.1 zeigt einen Ausschnitt der technischen Vorrangbeziehungen für die AB-Variante des Anwendungsfalls.

Tabelle A.1: Auswahl der technischen Vorrangbeziehungen für Variante AB

ID	Variante	Ebene	Fähigkeit	Elternteil	Vorgänger	Referenzzeit	Auftragsklasse
AB	AB	L3	AB		-	-	0
AB_EN	AB	L2	AB_EN	AB	AB_UB	-	1
AB_EN_EN1	AB	L1	CLIP_3x2	AB_EN	-	18	1
AB_FA	AB	L2	AB_FA	AB	AB_FE	-	1
AB_FA	AB	L2	AB_FA	AB	AB_FR	-	1
AB_FA_MAFE	AB	L3	JOIN_4x4x2	AB_FA	-	16	1
AB_FE	AB	L1	AB_FE	AB	-	-	0
AB_FE_BU	AB	L1	CLIP_2x1	AB_FE	-	3	1
AB_FE_CBU1	AB	L1	CLIP_1x1	AB_FE	AB_FE_BU	1	1
AB_FE_CBU2	AB	L1	CLIP_1x1	AB_FE	AB_FE_BU	1	1
AB_FE_GRL	AB	L1	GRILL_2x1	AB_FE	AB_FE_UCG1	3	1
AB_FE_GRR	AB	L1	GRILL_2x1	AB_FE	AB_FE_UCG2	3	1
AB_FE_LL	AB	L1	LIGHT_1x1	AB_FE	-	4	1
AB_FE_LR	AB	L1	LIGHT_1x1	AB_FE	-	4	1
AB_FE_NUM	AB	L1	FOIL	AB_FE	AB_FE_BU	4	1
AB_FE_UCG1	AB	L1	CLIP_2x1	AB_FE	-	3	1
AB_FE_UCG2	AB	L1	CLIP_2x1	AB_FE	-	3	1
AB_FF	AB	L2	AB_FF	AB	AB_FR	-	1
AB_FF_CONF	AB	L1	CLIP_3x2	AB_FF	-	5	1
AB_FF_GRL	AB	L1	GRILL	AB_FF	AB_FF_CONF	7	1
AB_FF_HDL	AB	L1	CLIP_4x1_RAMP	AB_FF	-	4	1
AB_FF_HDR	AB	L1	CLIP_4x1_RAMP	AB_FF	-	4	1
AB_FF_HOOD	AB	L1	CLIP_2x2_RAMP	AB_FF	AB_FF_CONF	4	1
AB_FR	AB	L2	AB_FR	AB	AB_EN	-	1
AB_FR	AB	L2	AB_FR	AB	AB_SE	-	1
AB_FR_FEPR	AB	L1	CLIP_2x2	AB_FR	AB_FR_FEPR1	3	1
AB_FR_FEPR	AB	L1	JOIN_2x2x2	AB_FR	AB_FR_FEPR2	3	1
AB_FR_FRRL	AB	L1	RIVET_1x1	AB_FR	AB_FR_FEBON	2	1
AB_FR_FRRL	AB	L1	RIVET_1x1	AB_FR	AB_FR_FEPR1	2	1
AB_FR_FRRL	AB	L1	RIVET_1x1	AB_FR	AB_FR_FEPR3	2	1
AB_FR_FRRR	AB	L3	RIVET_1x1	AB_FR	AB_FR_FEPR3	2	1
AB_LUX	AB	L1	AB_LUX	AB	AB_RF	0	1

Die weiteren Beziehungen der übrigen Modelle sind den beigefügten Datenbanken und Excel-Dateien zu entnehmen. Die IDs stellen die generischen Aufträge dar und beschreiben die Montage eines Bauteils. Der Elternteil beschreibt den generischen darüber.

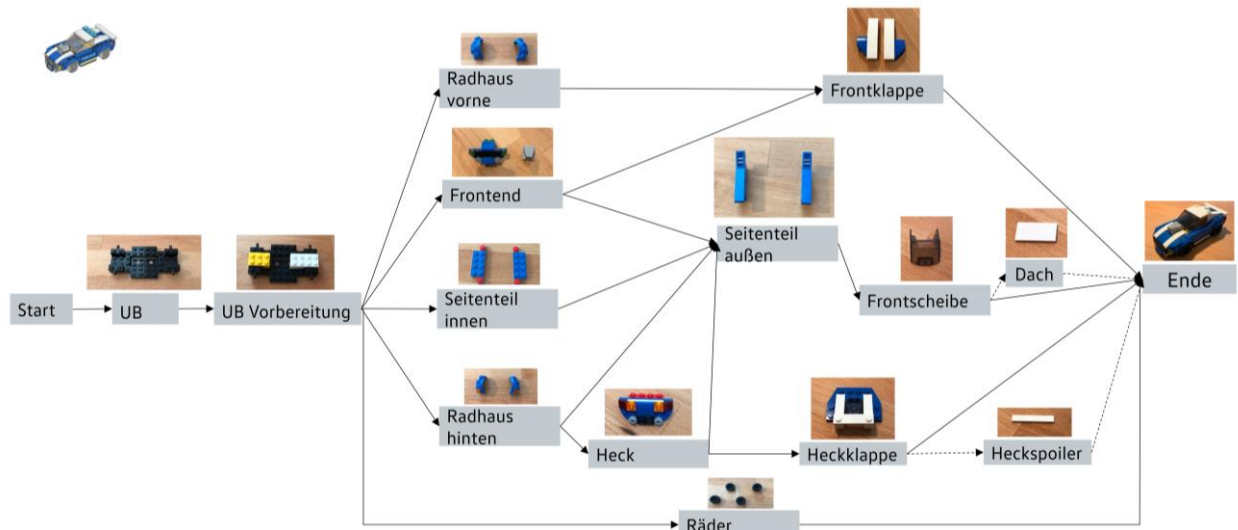


Abbildung A.2: Visualisierung Vorranggraph Produkt Auto blau (AB)

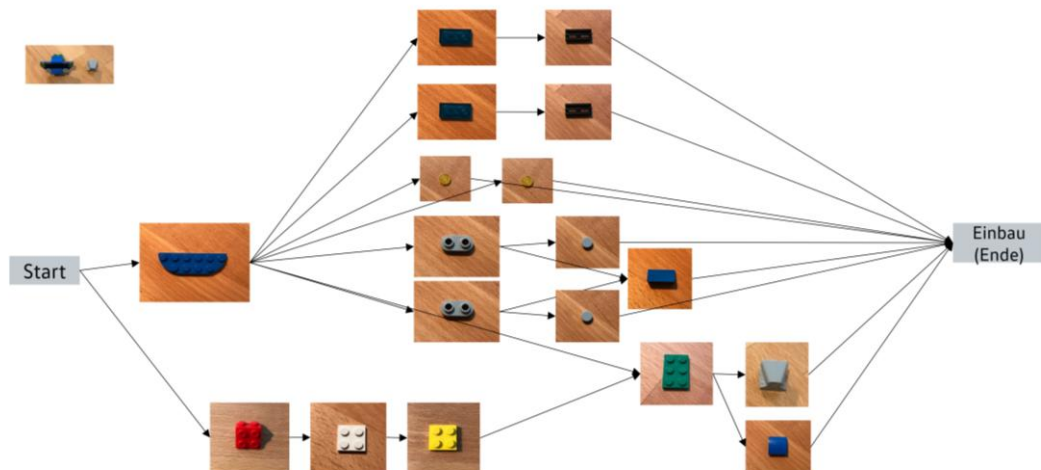


Abbildung A.3: Visualisierung Vorranggraph des Moduls Frontend (FE) für Produkt Auto blau (AB)

A.III. Holonische Kapselung des Entscheidungsmodells

Die nachfolgende Abbildung stellt die gekapselten Holone graphisch dar.

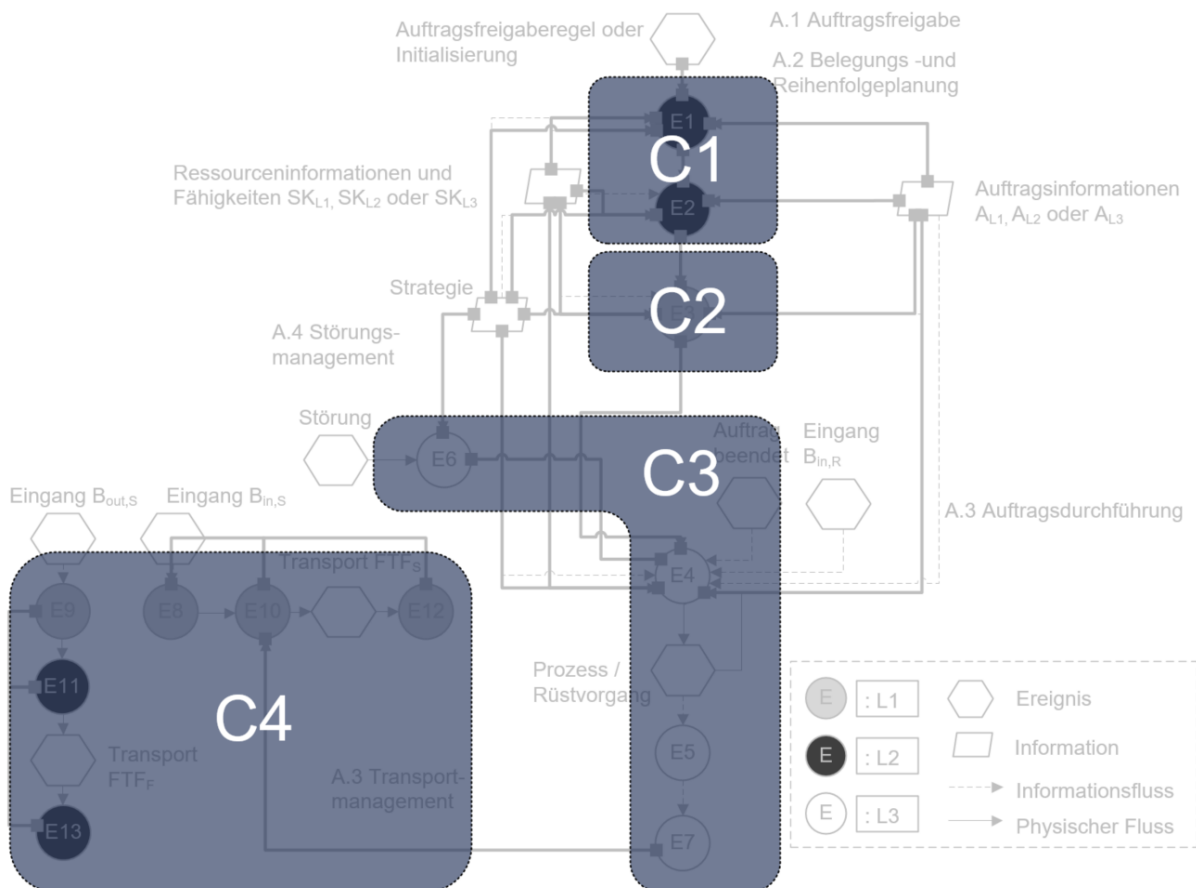


Abbildung A.4: Gruppierung der Steuerungsentscheidungen zu Clustern

A.IV. Detailansicht DES

Die DES spiegelt die technische Agentenumgebung des Simulationsframework wider und stellt die detaillierte Produktionsumgebung dar. In ihr existieren Produktionsentitäten wie Ressourcen, Puffer, FTF und Produkte.

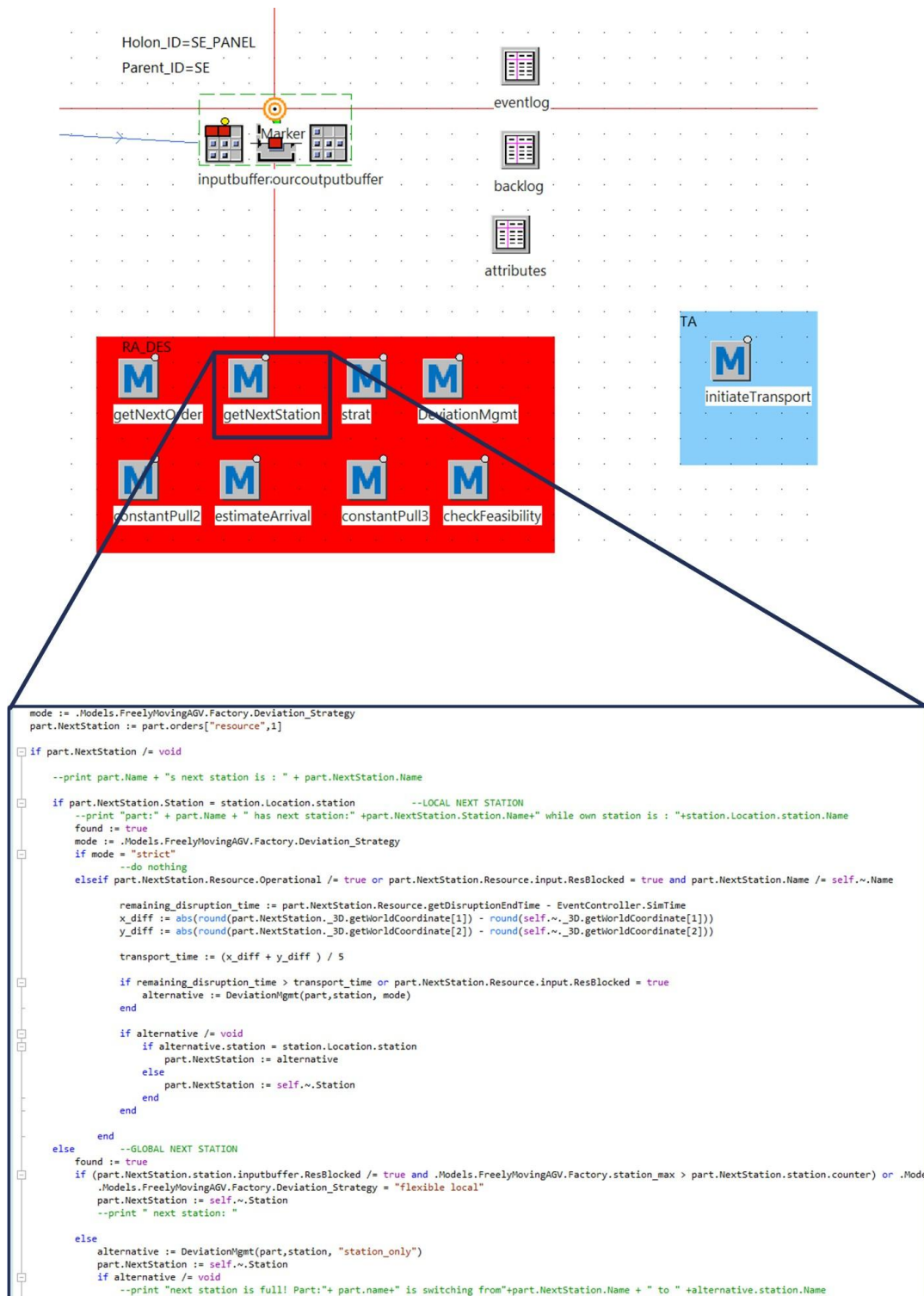


Abbildung A.5: Darstellung der Ressource / RH_{DES} mit Ausschnitt der Funktion „getNextStation“

A.V. Übersicht Datenbank

Weitere Datenbankauszüge zu den fähigkeitsbasierten PPR-Schemata.

id	level	status	parent_id	X_Coordinate	Y_Coordinate	current_state	current_order	current_tool	Input_Capacity	Output_Capacity
EN	L2	operational	F1	69	-64	NULL	NULL	NULL	NULL	NULL
EN_CLIP1	L3	operational	EN	77	-62	Waiting	AR_LF_45_AR_EN_EN2	CLIP_2x2	2	2
EN_CLIP2	L3	operational	EN	77	-70	Working	AR_ZW_23_AR_EN_EN2	CLIP_2x2	1	2
F1	L2	operational	F1	0	0	NULL	NULL	NULL	NULL	NULL
FA	L2	operational	F1	68	-86	NULL	NULL	NULL	NULL	NULL
FA_JOIN1	L3	operational	FA	74	-83	Waiting	AB_TP_198_AB_FA_MAFE	JOIN_4x4x2	2	2
FA_JOIN2	L3	operational	FA	79	-88	Working	AB_GA_292_AB_FA_MAFE	JOIN_4x4x2	2	2
FE	L2	operational	F1	61	-90	NULL	NULL	NULL	NULL	NULL
FE_CLIP	L3	operational	FE	54	-80	Waiting	AB_KE_58_AB_FE_CBU2	CLIP_1x1	2	2
FE_CLIPMEDIUM	L3	operational	FE	47	-80	Waiting	AB_QW_197_AB_SB_FRSECONR	CLIP_4x1	2	2
FE_LIGHT1	L3	operational	FE	59	-83	Waiting	AB_NZ_44_AB_FE_LL	LIGHT_1x1	2	2
FE_LIGHT2	L3	operational	FE	46	-87	Waiting	AW_DG_269_AW_FE_LFER	LIGHT_1x1	2	2
FE_RAMP	L3	operational	FE	56	-93	Waiting	AB_QW_197_AB_SB_SECONR	CLIP_2x1...	2	2
FE_RIVET	L3	operational	FE	48	-93	Waiting	AW_SI_182_AW_FE_GRL	GRILL_2x1	2	2
FF	L2	operational	F1	80	-101	NULL	NULL	NULL	NULL	NULL
FF_CLIP	L3	operational	FF	85	-105	Working	AW_XS_71_AW_FF_CLFF	CLIP_2x1	1	2
FF_CLIPMEDIUM	L3	operational	FF	77	-108	Waiting	AW_YE_211_AW_FF_FFOR	CLIP_2x2	2	2
FF_HOOD1	L3	operational	FF	73	-102	Setting-up	AW_DI_231_AW_FF_FF	HOOD_4x2	2	2
FF_HOOD2	L3	operational	FF	70	-106	Working	AW_YE_211_AW_FF_FFFR	HOOD_2x1	2	2
FF_RAMP	L3	operational	FF	72	-112	Setting-up	AB_IL_265_AB_FF_HDR	CLIP_4x1...	2	2
FF_WINDOW	L3	operational	FF	83	-112	Waiting	AW_YE_211_AW_FF_WVS	FRONT_WI...	2	2
FR	L2	operational	F1	63	-105	NULL	NULL	NULL	NULL	NULL
FR_CLIP	L3	operational	FR	57	-112	Working	AR_OB_162_AR_FR_FECON	CLIP_4x2	0	2
FR_JOIN	L3	operational	FR	57	-103	Waiting	AR_YI_158_AR_FR_FEASSEM	JOIN_2x1	1	2
FR_WELD	L3	operational	FR	53	-107	Working	AR_YI_158_AR_FR_FESUPP	WELD_3x2	2	2
LUX	L2	operational	F1	91	-64	NULL	NULL	NULL	NULL	NULL
LUX_DASH1	L3	operational	LUX	96	-61	Waiting	AW_YE_211_AW_LUX_DB	DASHBOARD	2	2
LUX_DASH2	L3	operational	LUX	104	-61	Waiting	AW_OB_1_AW_LUX_DB	DASHBOARD	2	2
LUX_LIGHT	L3	operational	LUX	97	-69	Setting-up	AR_IF_203_AR_LUX_SPEC1	RIVET_2x1	0	2
LUX_SAFE	L3	operational	LUX	104	-69	Working	AR_TQ_47_AR_LUX_SPEC2	JOIN_SAFE	0	2
RA	L2	operational	F1	91	-48	NULL	NULL	NULL	NULL	NULL
RA_CLIP	L3	operational	RA	97	-48	Waiting	AB_QW_240_AB_RA_CLIP	CLIP_1x1	2	2

Abbildung A.6: Auszug des Ressourcen-Schemas aus der MySQL-Datenbank

id	level	status	parent_id	X_Coordinate	Y_Coordinate	current_state	current_order	current_tool	Input_Capacity	Output_Capacity
EN	L2	operational	F1	69	-64	NULL	NULL	NULL	NULL	NULL
EN_CLIP1	L3	operational	EN	77	-62	Waiting	AR_LF_45_AR_EN_EN2	CLIP_2x2	2	2
EN_CLIP2	L3	operational	EN	77	-70	Working	AR_ZW_23_AR_EN_EN2	CLIP_2x2	1	2
F1	L2	operational	F1	0	0	NULL	NULL	NULL	NULL	NULL
FA	L2	operational	F1	68	-86	NULL	NULL	NULL	NULL	NULL
FA_JOIN1	L3	operational	FA	74	-83	Waiting	AB_TP_198_AB_FA_MAFE	JOIN_4x4x2	2	2
FA_JOIN2	L3	operational	FA	79	-88	Working	AB_GA_292_AB_FA_MAFE	JOIN_4x4x2	2	2
FE	L2	operational	F1	61	-90	NULL	NULL	NULL	NULL	NULL
FE_CLIP	L3	operational	FE	54	-80	Waiting	AB_KE_58_AB_FE_CBU2	CLIP_1x1	2	2
FE_CLIPMEDIUM	L3	operational	FE	47	-80	Waiting	AB_QW_197_AB_SB_FRSECONR	CLIP_4x1	2	2
FE_LIGHT1	L3	operational	FE	59	-83	Waiting	AB_NZ_44_AB_FE_LL	LIGHT_1x1	2	2
FE_LIGHT2	L3	operational	FE	46	-87	Waiting	AW_DG_269_AW_FE_LFER	LIGHT_1x1	2	2
FE_RAMP	L3	operational	FE	56	-93	Waiting	AB_QW_197_AB_SB_SECONR	CLIP_2x1...	2	2
FE_RIVET	L3	operational	FE	48	-93	Waiting	AW_SI_182_AW_FE_GRL	GRILL_2x1	2	2
FF	L2	operational	F1	80	-101	NULL	NULL	NULL	NULL	NULL
FF_CLIP	L3	operational	FF	85	-105	Working	AW_XS_71_AW_FF_CLFF	CLIP_2x1	1	2
FF_CLIPMEDIUM	L3	operational	FF	77	-108	Waiting	AW_YE_211_AW_FF_FFOR	CLIP_2x2	2	2
FF_HOOD1	L3	operational	FF	73	-102	Setting-up	AW_DI_231_AW_FF_FF	HOOD_4x2	2	2
FF_HOOD2	L3	operational	FF	70	-106	Working	AW_YE_211_AW_FF_FFFR	HOOD_2x1	2	2
FF_RAMP	L3	operational	FF	72	-112	Setting-up	AB_IL_265_AB_FF_HDR	CLIP_4x1...	2	2
FF_WINDOW	L3	operational	FF	83	-112	Waiting	AW_YE_211_AW_FF_WVS	FRONT_WI...	2	2
FR	L2	operational	F1	63	-105	NULL	NULL	NULL	NULL	NULL
FR_CLIP	L3	operational	FR	57	-112	Working	AR_OB_162_AR_FR_FECON	CLIP_4x2	0	2
FR_JOIN	L3	operational	FR	57	-103	Waiting	AR_YI_158_AR_FR_FEASSEM	JOIN_2x1	1	2
FR_WELD	L3	operational	FR	53	-107	Working	AR_YI_158_AR_FR_FESUPP	WELD_3x2	2	2
LUX	L2	operational	F1	91	-64	NULL	NULL	NULL	NULL	NULL
LUX_DASH1	L3	operational	LUX	96	-61	Waiting	AW_YE_211_AW_LUX_DB	DASHBOARD	2	2
LUX_DASH2	L3	operational	LUX	104	-61	Waiting	AW_OB_1_AW_LUX_DB	DASHBOARD	2	2
LUX_LIGHT	L3	operational	LUX	97	-69	Setting-up	AR_IF_203_AR_LUX_SPEC1	RIVET_2x1	0	2
LUX_SAFE	L3	operational	LUX	104	-69	Working	AR_TQ_47_AR_LUX_SPEC2	JOIN_SAFE	0	2
RA	L2	operational	F1	91	-48	NULL	NULL	NULL	NULL	NULL
RA_CLIP	L3	operational	RA	97	-48	Waiting	AB_QW_240_AB_RA_CLIP	CLIP_1x1	2	2

Abbildung A.7: Auszug des Auftrags-Schemas aus der MySQL-Datenbank

A.VI. Weitere Zielgrößen der Simulationsexperimente

Nachfolgende Darstellung zeigt den Bestandsverlauf des Gesamtsystems als Teil der Simulationsexperimente für eine untere kritische Systemgrenze von 80 Produkten und einem Vorgriffshorizont des WCH von 30 Produkten.

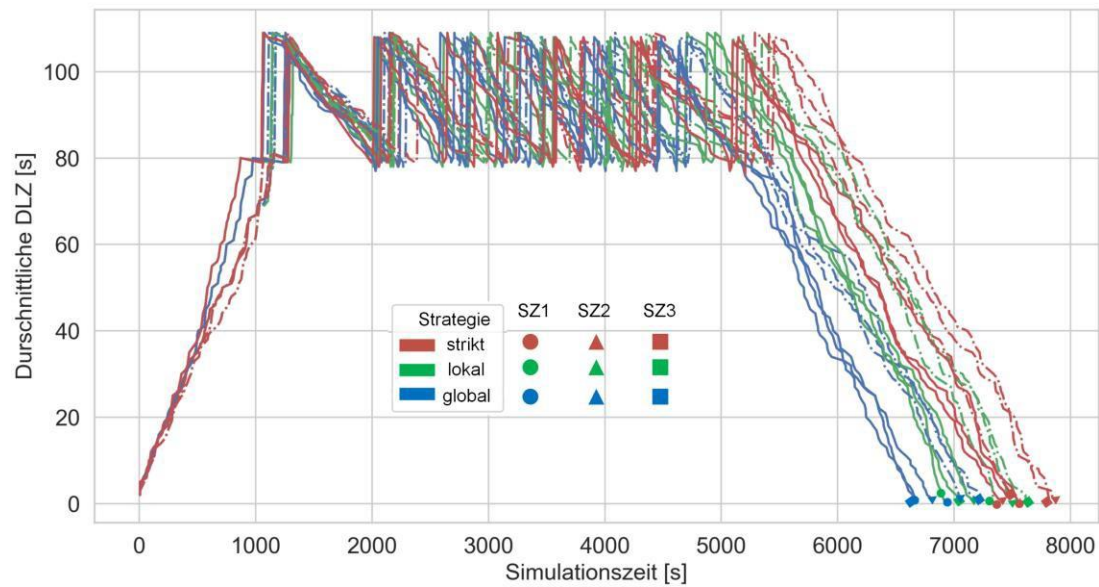


Abbildung A.8: Bestandsverlauf des Systems über alle Simulationsexperimente

Nachfolgende Darstellung präsentiert den beispielhaften variantenspezifischen Verlauf der ersten 20 Minuten der Simulation der Unterboden-Station und verdeutlicht das Prinzip der überlappenden Fertigung. Im weiteren Verlauf wird dieses Prinzip durch die sofortige Auftragsfreigabe nach Auftragsplanung vernachlässigt.

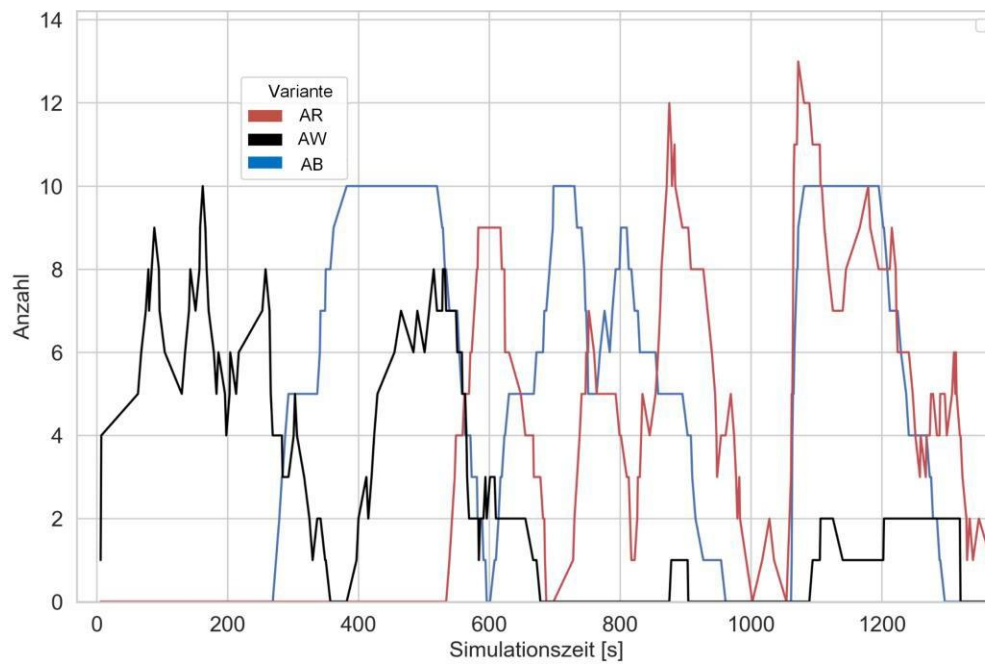


Abbildung A.9: Variantenspezifischer Bestand der UB-Station

Nachfolgende Grafik präsentiert einen Verlauf der Anteile an Störungs-, Produktions-, Rüst- und Wartezeit einer beispielhaften Ressource für die Radmontage.

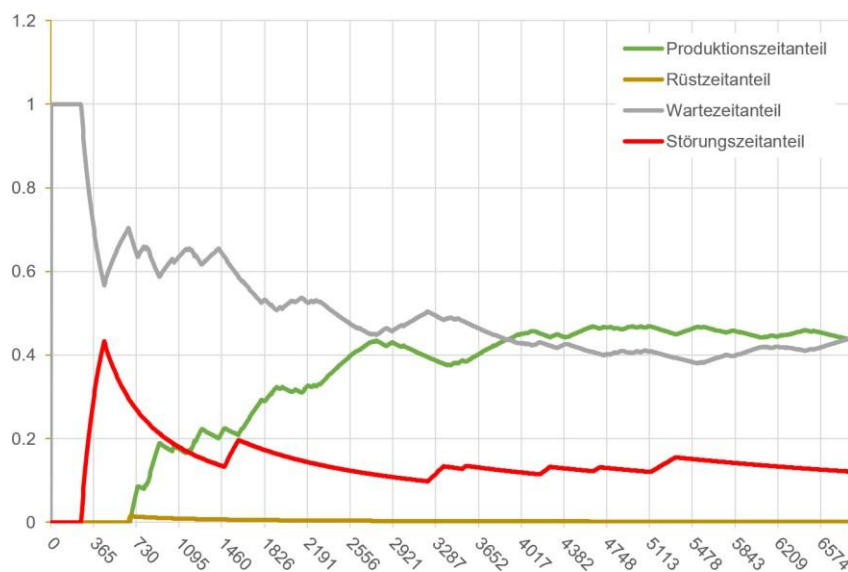


Abbildung A.10: Zeitanteile der Ressource WH_TIRE2 in SZ2 mit globaler Ausweichstrategie

A.VII. Korrelationsanalysen zum Optimierungsverfahren

Die Korrelationsanalysen zeigen den Zusammenhang der Problemdimensionen und den Einfluss dieser.

Einfluss der Jobanzahl

Tabelle A.2: Spearman-Korrelationskoeffizient ρ für Zusammenhang zwischen $|\mathcal{J}|$ und t^{imp}

$ \mathcal{M} - \mathcal{O}_j $	ρ	p-Wert
$3m - 5j$	0,9799	$1,3300 \cdot 10^{-17}$
$4m - 5j$	0,9712	$2,7005 \cdot 10^{-13}$
$5m - 10j$	0,9853	$1,0603 \cdot 10^{-20}$
$5m - 15j$	0,9852	$4,0405 \cdot 10^{-12}$
$6m - 10j$	0,9724	$2,5734 \cdot 10^{-17}$
$6m - 15j$	0,9820	$8,6482 \cdot 10^{-11}$
$8m - 10j$	0,9755	$5,8291 \cdot 10^{-14}$
$8m - 15j$	0,9820	$8,6482 \cdot 10^{-11}$
$10m - 10j$	0,9515	$1,1861 \cdot 10^{-10}$
$10m - 15j$	0,9716	$1,3812 \cdot 10^{-07}$

Tabelle A.2 präsentiert die Spearman-Korrelationskoeffizienten für verschiedene Instanzen, bei denen $|\mathcal{M}|$ und $|\mathcal{O}|$ konstant sind und lediglich die Anzahl der Jobs variiert. Resultate, bei denen das Zeitlimit für die Berechnung erreicht wurde, wurden ausgeschlossen. In solchen Fällen wiesen die Ergebnisse denselben Wert für die Verbesserungszeit bei unterschiedlichen Job-Anzahlen auf, was die Korrelation beeinträchtigen könnte. Die p-Werte in der Tabelle dienen als Maß für die Signifikanz der Korrelation und geben die Wahrscheinlichkeit an, dass die beobachtete Korrelation nicht durch einen tatsächlichen Zusammenhang, sondern durch zufällige Varianz in den Daten erklärt werden kann (vgl. z. B. [Ak18]). Diese Werte, ebenso wie die in den folgenden Abschnitten, wurden mithilfe der Funktion `cor.test()` in der Statistiksoftware R ermittelt [R 20].

Einfluss der Arbeitsganganzahl

Ein Korrelationstest zur Untersuchung des Einflusses der Arbeitsganganzahl pro Job auf die Verbesserungszeit ist nicht angemessen durchführbar, da nur drei unterschiedliche Werte für $|\mathcal{O}_j|$ vorliegen. Stattdessen wird im Folgenden die These, dass $|\mathcal{O}_j|$ mit t^{imp} in Zusammenhang steht, durch die Durchführung eines weiteren Vergleichs untersucht.

Tatsächlich ist auch die berechnete Korrelation über die Gesamtheit der Daten für den Zusammenhang zwischen $|\mathcal{O}|$ und t^{imp} größer ($\rho^{\text{TS}} = 0,9303$ mit p-Wert $< 0,0001$) als für das Verhältnis von $|\mathcal{J}|$ und t^{imp} ($\rho^{\text{TS}} = 0,9029$ mit p-Wert $< 0,0001$).

Einfluss der Ressourcenanzahl

Bei näherer Untersuchung kann gezeigt werden, dass die durchschnittliche Anzahl der Arbeitsgänge pro Job, die von einer Ressource aufgrund ihrer Fähigkeiten bearbeitet werden kann, positiv mit der Verbesserungszeit korreliert und mit

$$\eta = \sum_{j \in \mathcal{J}} \frac{\sum_{o \in \mathcal{O}_j} \frac{|\mathcal{M}_o|}{|\mathcal{M}|}}{|\mathcal{J}|}$$

berechnet wird.

Tabelle A.3: Spearman-Korrelationskoeffizienten für Zusammenhang zwischen η und t^{imp}

$ \mathcal{M} - \mathcal{J} $	#	ρ	p-Wert
$5m - 5j$	30	0,5815	$7,51 \cdot 10^{-04}$
$5m - 10j$	30	0,8665	$6,07 \cdot 10^{-10}$
$5m - 20j$	30	0,8434	$4,88 \cdot 10^{-09}$
$5m - 30j$	30	0,8665	$6,07 \cdot 10^{-10}$
$5m - 40j$	27	0,7273	$1,73 \cdot 10^{-05}$
$5m - 50j$	18	0,4741	$4,68 \cdot 10^{-02}$
$6m - 5j$	30	0,6355	$1,61 \cdot 10^{-04}$
$6m - 10j$	30	0,8665	$6,07 \cdot 10^{-10}$
$6m - 20j$	30	0,8665	$6,07 \cdot 10^{-10}$
$6m - 30j$	30	0,8588	$1,27 \cdot 10^{-09}$
$6m - 40j$	30	0,8665	$6,07 \cdot 10^{-10}$
$6m - 50j$	18	0,6465	$3,74 \cdot 10^{-03}$
$8m - 5j$	25	0,6907	$1,32 \cdot 10^{-04}$
$8m - 10j$	25	0,8492	$7,99 \cdot 10^{-08}$
$8m - 20j$	25	0,8266	$3,55 \cdot 10^{-07}$
$8m - 30j$	25	0,8492	$7,99 \cdot 10^{-08}$
$8m - 40j$	13	0,8180	$1,05 \cdot 10^{-06}$
$8m - 50j$	30	0,7319	$4,45 \cdot 10^{-03}$
$10m - 5j$	30	0,6586	$7,59 \cdot 10^{-05}$
$10m - 10j$	30	0,7202	$7,23 \cdot 10^{-06}$
$10m - 20j$	30	0,8434	$4,88 \cdot 10^{-09}$
$10m - 30j$	30	0,8665	$6,07 \cdot 10^{-10}$
$10m - 40j$	17	0,5590	$1,97 \cdot 10^{-02}$

Die Tabelle A.3 zeigt die Korrelationen bei variierender Anzahl von Arbeitsgängen pro Job, bei gleichzeitig konstanter Ressourcen- und Jobanzahl. Auch hier wurden Datenpunkte entfernt, wenn der Suchlauf aufgrund des Zeitlimits terminiert wurde. Die Anzahl der Suchläufe, für die die Bedingungen zutreffen, ist in der Spalte # angegeben.

Die Varianz der beschriebenen Größe zwischen den betrachteten Problem instanzen ist allerdings nicht sehr hoch, weswegen die Korrelation hier nicht so eindrücklich dargestellt werden kann wie für $|\mathcal{O}|$.

A.VIII. Exemplarische Optimierungsverläufe

Für die Anwendung des implementierten Optimierungsverfahrens aus Abschnitt 4.3 zur Auftragsplanung wird unter Berücksichtigung der Eigenschaften des Verfahrens zur Problemgröße und Konvergenz aus Abschnitt 4.3.2 eine Lösungszeit parametrisiert. Dabei ist zu beachten, dass die freigegebenen Orders (Produktaufträge) auf L3-Ebene zu Gruppen mit 5 oder 10 Aufträgen gekapselt werden. Die Produktaufträge besitzen jeweils Modulaufträge auf L2-Ebene. Die Beziehung zwischen Produkt- und Modulaufträgen entspricht durch die Aggregation und Kapselung der von Jobs und Arbeitsgängen auf L3- und L2-Ebene.

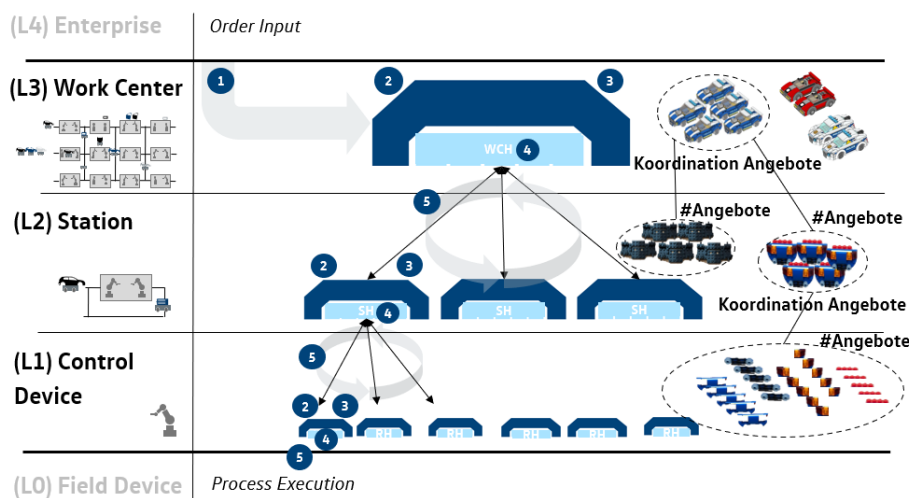


Abbildung A.11: Zerlegung der Aufträge und Systeme

Die Verringerung der Lösungskomplexität wird exemplarisch erläutert (vergleiche Abbildung 4.18 in Abschnitt 4.3.2).

Initialplanung: Für die Initialplanung während des Hochlaufs werden die 110 L3-Orders (Orders/Produktaufträge) zu 22 (11) Auftragsgruppen mit jeweils 5 (10) L3-Orders zusammengefasst. Die Anzahl der einzuplanenden L2-Orders auf 16 Stationen beträgt 320 (160). Die Anzahl der zu koordinierenden L1-Orders innerhalb einer Station variiert je nach Anzahl der L3-Auftragsgruppengröße und Anzahl der Prozessaufträge pro Job zwischen 25-150 Operationen. Für die Initialplanung kann aufgrund der Anzahl der L2-Orders bereits für bis zu 10 Entitäten eine Optimierungszeit von über 3600s anfallen.

Neuplanung: Für die Neuplanung von 30 Aufträgen würden ohne Zerlegung und Kapselung 480 Jobs auf 16 Stationen koordiniert werden müssen. Durch Zerlegung und Kapselung in die Auftragsgruppen entspricht dies 96 (48) Jobs mit je 5 (10) Jobs pro Station auf L2-Ebene. Für die Ablaufplanung von 96 (48) Auftragsgruppen auf 16 Stationen wird eine Optimierungszeit von 17s für die Koordination von 10 Ressourcen und bis zu 150 Arbeitsgänge gewählt. Auf Control Device

Ebene besitzt jede L2-Order (Modulaufträge) 5 - 15 L1-Orders (Prozessaufträge), die innerhalb einer Station auf 3-8 Control Devices ausgeführt werden.

Für die Neuplanung werden exemplarisch Verläufe der WCH und SH innerhalb der parametrisierten Lösungszeiten gezeigt.

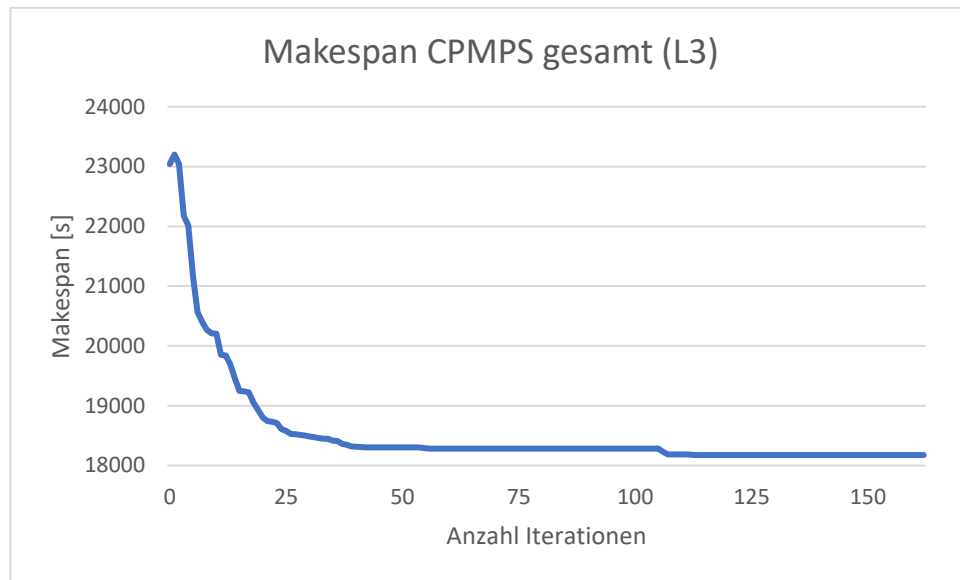


Abbildung A.12: Exemplarische Makespanoptimierung WCH mit Lösungszeit 15s

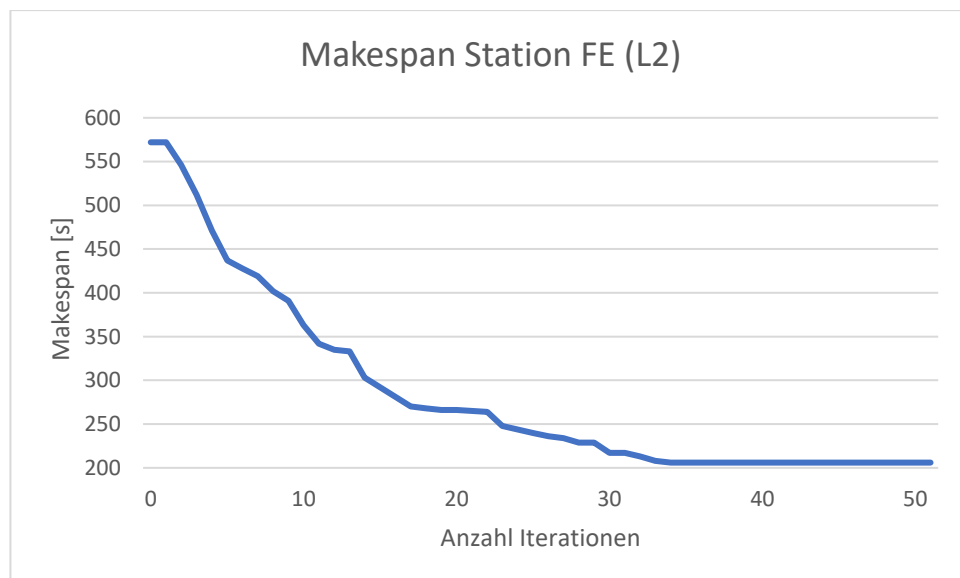


Abbildung A.13: Exemplarische Makespanoptimierung SH auf L2-Ebene mit Lösungszeit 5s

Literaturverzeichnis

- [Ab17] Abar, S. et al.: Agent Based Modelling and Simulation tools: A review of the state-of-art software. *Computer Science Review* 24, S. 13–33, 2017.
- [Ak18] Akoglu, H.: User's guide to correlation coefficients. *Turkish journal of emergency medicine* 3/18, S. 91–93, 2018.
- [Ar21] Arlinghaus, J.: Resilienz ist wichtiger als Effizienz. *atp magazin* 11, S. 18, 2021.
- [Au16] Audi AG: *Die Modulare Montage*, Ingolstadt, 2016.
- [Az20] Azad, P. et al.: The role of structured and unstructured data managing mechanisms in the Internet of things. *Cluster Computing* 2/23, S. 1185–1198, 2020.
- [Ba13] Barbosa, J. et al.: Structural Self-organized Holonic Multi-Agent Manufacturing Systems. In (Mařík, V.; Lastra, J. L. M.; Skobelev, P. Hrsg.): *Industrial Applications of Holonic and Multi-Agent Systems*. Springer Berlin Heidelberg, Berlin, Heidelberg, S. 59–70, 2013.
- [Ba14] Bauernhansl, T.: Die Vierte Industrielle Revolution – Der Weg in ein wertschaffendes Produktionsparadigma. In (Bauernhansl, T.; Hompel, M. ten; Vogel-Heuser, B. Hrsg.): *Industrie 4.0 in Produktion, Automatisierung und Logistik*. Springer Fachmedien Wiesbaden, Wiesbaden, S. 5–35, 2014.
- [Ba15a] Barbosa, J. et al.: Improving the ADACOR2 supervisor holon scheduling mechanism with genetic algorithms. *AIP Publishing LLC*, S. 140008, 2015.
- [Ba15b] Bauernhansl, T.: Automotive industry without conveyer belt and cycle - research campus ARENA2036. In: Michael Bargende, Hans-Christian Reuss und Jochen Wiedemann (Hg.): *15. Internationales Stuttgarter Symposium. Automobil- und Motorentechnik*. Wiesbaden: Springer Vieweg, 2015.
- [Ba20a] Bayha, A.; Bock, J.; Boss, B.; Diedrich, C.; Malakuti, S.: Describing Capabilities of Industrie 4.0 Components. Bundesministerium für Wirtschaft und Energie. Berlin (Joint White Paper between Plattform Industrie 4.0, VDI GMA 7.20, BaSys 4.2), 2020.
- [Ba20b] Bauernhansl, T.: Entwicklung, Aufbau und Demonstration Einer Wandlungsfähigen (Fahrzeug-) Forschungsproduktion. Unter Mitarbeit von Manuel Fechter und Thomas Dietz. Berlin, Heidelberg: Springer Berlin / Heidelberg (Arena2036 Ser), 2020.

- [BCG10] Bellifemine, F. L.; Caire, G.; Greenwood, D.: Developing multi-agent systems with JADE. John Wiley, Chichester, England, Hoboken, NJ, 2010.
- [Be22] Bergweiler, S. et al.: Production Level 4 – Der Weg zur zukunftssicheren und verlässlichen Produktion - Whitepaper SmartFactory-KL. https://smartfactory.de/wp-content/uploads/2022/05/SF_BROSCHU%CC%88RE_052022_A4_WEB-1.pdf, 2022.
- [BFR15] Birgin, E. G.; Ferreira, J. E.; Ronconi, D. P.: List scheduling and beam search methods for the flexible job shop scheduling problem with sequencing flexibility. In: *European Journal of Operational Research* 247 (2), S. 421–440, 2015.
- [BG08] Botti, V.; Giret, A.: ANEMONA. Springer London, London, 2008.
- [BJW04] Bussmann, S.; Jennings, N. R.; Wooldridge, M.: Agent-Based Production Control. In (Ishida, T. et al. Hrsg.): Multiagent Systems for Manufacturing Control. Springer Berlin Heidelberg, Berlin, Heidelberg, S. 9–52, 2004.
- [BLT10] Bratukhin, A.; Lüder, A.; Treytl, A.: Applications of Agent Systems in Intelligent Manufacturing. In (Kühnle, H. Hrsg.): Distributed Manufacturing: Paradigm, Concepts, Solutions and Examples. Springer London, London, S. 113–138, 2010.
- [Bo16] Bochmann, L. S. et al.: Innovative Konzepte einer sich selbstorganisierenden Fahrzeugmontage am Beispiel des Forschungsprojekts SMART FACE. In (Roth, A. Hrsg.): Einführung und Umsetzung von Industrie 4.0. Springer, S. 191, 2016.
- [Bo17] Bochmann, L.S. et al.: Human-robot Collaboration in Decentralized Manufacturing Systems: An Approach for Simulation-based Evaluation of Future Intelligent Production. In: *Procedia CIRP* 62, S. 624–629, 2017.
- [Bo18] Bochmann, L. S.: Entwicklung und Bewertung eines flexiblen und dezentral gesteuerten Fertigungssystems für variantenreiche Produkte, 2018.
- [Bo19] Borangiu, T. et al. Hrsg.: Service Orientation in Holonic and Multi-Agent Manufacturing. Springer International Publishing, Cham, 2019.
- [Bo59] Bowman, E. H.: The Schedule-Sequencing Problem. *Operations Research* 5/7, S. 621–624, 1959.
- [Bö06] Böcker, J. et al.: Self-Optimization as a Framework for Advanced Control Systems. In: S. 4671–4675, 2006.
- [BPH82] Blackstone, J. H.; Philipps, D. T.; Hogg, G. L.: A state-of-the-art survey of dispatching rules for manufacturing job shop operations. *International Journal of Production Research* 1/20, S. 27–45, 1982.

- [Br10] Broy, Manfred: Cyber-Physical Systems. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.
- [Br15] Braunisch, D.: Multiagentensysteme im Bereich der rückführenden Logistik Entwurf einer Systemarchitektur zur Steigerung der Prozesseffizienz durch dynamische Disposition der Sekundärrohstofflogistik, 2015.
- [Br18] Brecher, C. et al.: Multi-Level Modeling Framework for Machine as a Service Applications Based on Product Process Resource Models: Proceedings of the 2nd International Symposium on Computer Science and Intelligent Control. ACM, New York, NY, USA, S. 1–9, 2018.
- [Bu17] Buth, L. et al.: Introducing agent-based simulation of manufacturing systems to industrial discrete-event simulation tools: 2017 IEEE 15th International Conference on Industrial Informatics (INDIN). IEEE, S. 1141–1146, 2017.
- [Bu20] Burggräf, P. et al.: Modeling and evaluating agile assembly systems using mixed-integer linear programming. Procedia CIRP 93, S. 1073–1078, 2020.
- [Bu22] Buckhorst, A. F. et al.: A Holonic Control System Approach for Line-less Mobile Assembly System Operations. Procedia CIRP 107, S. 1449–1454, 2022.
- [Ca11] Castellini, P. et al.: Towards the integration of process and quality control using multi-agent technology: IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society, S. 421–426, 2011.
- [CF21] Cardoso, R. C.; Ferrando, A.: A Review of Agent-Based Programming for Multi-Agent Systems. In: *Computers* 10 (2), S. 16, 2021
- [CH12] Crooks, A.; Heppenstall, A. J.: Introduction to Agent-Based Modelling: Agent-Based Models of Geographical Systems, S. 85–105, 2012.
- [Ch94] Christensen, J.: Holonic Manufacturing Systems: Initial Architecture and Standards Directions, 1994.
- [CM00] Chirn, J.-L.; McFarlane, D. C.: A holonic component-based approach to reconfigurable manufacturing control architecture: Proceedings 11th International Workshop on Database and Expert Systems Applications, S. 219–223, 2000.
- [Co14] Colombo, A. W. et al.: Industrial Cloud-Based Cyber-Physical Systems. Springer International Publishing, Cham, 2014.
- [Co15] Colombo, A. W. et al.: Chapter 4 - Industrial Agents in the Era of Service-Oriented Architectures and Cloud-Based Industrial Infrastructures. In (Leitão, P.; Karnouskos, S. Hrsg.): Industrial Agents. Morgan Kaufmann, Boston, S. 67–87, 2015.

- [Co17] Colombo, Armando W.; Karnouskos, Stamatis; Kaynak, Okyay; Shi, Yang; Yin, Shen: Industrial Cyberphysical Systems: A Backbone of the Fourth Industrial Revolution. In: *EEE Ind. Electron. Mag.* 11 (1), S. 6–16, 2017.
- [Cr19] Cruz Salazar, L. A. et al.: Cyber-physical production systems architecture based on multi-agent’s design pattern—comparison of selected approaches mapping four agent patterns. *The International Journal of Advanced Manufacturing Technology* 9/105, S. 4005–4034, 2019.
- [Da09] Dangelmaier, W.: *Theorie der Produktionsplanung und -steuerung*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [Da96] Das, S. K.: The measurement of flexibility in manufacturing systems. *International Journal of Flexible Manufacturing Systems* 1/8, S. 67–93, 1996.
- [DCT21] Derigent, W.; Cardin, O.; Trentesaux, D.: Industry 4.0: contributions of holonic manufacturing control architectures and future challenges. *Journal of Intelligent Manufacturing* 7/32, S. 1797–1818, 2021.
- [Di16] Dias, L. M. S. et al.: Discrete simulation software ranking — A top list of the world-wide most popular and used tools: 2016 Winter Simulation Conference (WSC). IEEE, S. 1060–1071, 2016.
- [DIN SPEC 91345]:2016, DIN SPEC 91345. Referenzarchitekturmodell Industrie 4.0 (RAMI4.0).
- [DK13] Demir, Y.; Kürşat İşleyen, S.: Evaluation of mathematical models for flexible job-shop scheduling problems. *Applied Mathematical Modelling* 3/37, S. 977–988, 2013.
- [Do15] Domschke, W. et al.: *Einführung in Operations Research*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.
- [DP97] Dautère-Pérès, S.; Paulli, J.: An integrated approach for modeling and solving the general multiprocessor job-shop scheduling problem using tabu search. *Annals of Operations Research* 0/70, S. 281–306, 1997.
- [DS19] Dorigo, M.; Stützle, T.: Ant Colony Optimization: Overview and Recent Advances. In: Michel Gendreau und Jean-Yves Potvin (Hg.): *Handbook of Metaheuristics*, Bd. 272. Cham: Springer International Publishing (272), S. 311–351, 2019.
- [DW04] Dam, K. H.; Winikoff, M.: Comparing Agent-Oriented Methodologies. In (Kanade, T. et al. Hrsg.): *Agent-Oriented Information Systems*. Springer Berlin Heidelberg, Berlin, Heidelberg, S. 78–93, 2004.

- [EL19] Echsler Minguillon, F.; Lanza, G.: Coupling of centralized and decentralized scheduling for robust production in agile production systems. In: *Procedia CIRP* 79, S. 385–390, 2019.
- [EH10] Elger, J.; Haußner, C.: Entwicklungen in der Automatisierungstechnik. In (Günthner, W.; Hompel, M. ten Hrsg.): *Internet der Dinge in der Intralogistik*. Springer Berlin Heidelberg, Berlin, Heidelberg, S. 23–27, 2010.
- [Eh11] Ehm, H. et al.: Towards a supply chain simulation reference model for the semiconductor industry: Proceedings of the 2011 Winter Simulation Conference (WSC), S. 2119–2130, 2011.
- [ELM05] ElMaraghy, H. A.: Flexible and reconfigurable manufacturing systems paradigms. *International Journal of Flexible Manufacturing Systems* 4/17, S. 261–276, 2005.
- [Er05] Erl, T.: *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall PTR, USA, 2005.
- [Er21] Ertel, W.: Lernen durch Verstärkung (Reinforcement Learning). In (Ertel, W. Hrsg.): *Grundkurs Künstliche Intelligenz*. Springer Fachmedien Wiesbaden, Wiesbaden, S. 351–377, 2021.
- [ES20] Echsler Minguillon, F.; Stricker, N.: Robust predictive–reactive scheduling and its effect on machine disturbance mitigation. *CIRP Annals* 1/69, S. 401–404, 2020.
- [FB15] Foith-Förster, P.; Bauernhansl, T.: Changeable and reconfigurable assembly systems – A structure planning approach in automotive manufacturing. In: Michael Barge, Hans-Christian Reuss und Jochen Wiedemann (Hg.): *15. Internationales Stuttgarter Symposium. Automobil- und Motorentechnik*. Wiesbaden: Springer Vieweg (SpringerLink Bücher), S. 1173–1192, 2015.
- [Fe06] Ferrarini, L. et al.: Control Architecture for Reconfigurable Manufacturing Systems: the PABADIS'PROMISE approach: 2006 IEEE Conference on Emerging Technologies and Factory Automation. *IEEE*, S. 545–552, 2006.
- [FEB17] Foith-Förster, P.; Eising, J.-H.; Bauernhansl, T.: Effiziente Montagesysteme ohne Band und Takt. Online verfügbar unter <https://publica.fraunhofer.de/handle/publica/247946>, 2017.
- [FI02] Foundation for Intelligent Physical Agents: FIPA Abstract Architecture Specification. <http://www.fipa.org/specs/fipa00001/index.html>, Stand: 2023-12-23.
- [Fi19] Filz, M.-A. et al.: Analyzing different material supply strategies in matrix-structured manufacturing systems. *Procedia CIRP* 81, S. 1004–1009, 2019.

- [FMW08] Fleischmann, B.; Meyr, H.; Wagner, M.: Advanced Planning. In (Stadtler, H.; Kilger, C. Hrsg.): Supply Chain Management and Advanced Planning. Springer Berlin Heidelberg, Berlin, Heidelberg, S. 81–106, 2008.
- [Fo] Fowler, M.: Microservices. a definition of this new architectural term. <https://martinfowler.com/articles/microservices.html>, Stand: 02.01.2024.
- [FR15] Farid, A. M.; Ribeiro, L.: An Axiomatic Design of a Multiagent Reconfigurable Mechatronic System Architecture. IEEE Transactions on Industrial Informatics 5/11, S. 1142–1155, 2015.
- [FSS03] Fischer, K.; Schillo, M.; Siekmann, J.: Holonic Multiagent Systems: A Foundation for the Organisation of Multiagent Systems. In (Goos, G. et al. Hrsg.): Holonic and Multi-Agent Systems for Manufacturing. Springer Berlin Heidelberg, Berlin, Heidelberg, S. 71–80, 2003.
- [FWF20] Fries, C.; Wiendahl, H.-H.; Foith-Förster, P.: Planung zukünftiger Automobilproduktionen. In (Bauernhansl, T.; Fechter, M.; Dietz, T. Hrsg.): Entwicklung, Aufbau und Demonstration einer wandlungsfähigen (Fahrzeug-) Forschungsproduktion. Springer Berlin Heidelberg, Berlin, Heidelberg, S. 19–43, 2020.
- [Ga16] Gamboa Quintanilla, F. et al.: A modeling framework for manufacturing services in Service-oriented Holonic Manufacturing Systems. Engineering Applications of Artificial Intelligence 55, S. 26–36, 2016.
- [Ga18] Gan, K. S. et al.: A FIPA-ACL Ontology in Enhancing Interoperability Multi-agent Communication. In (Rayner, A. et al. Hrsg.): Computational Science and Technology. Springer Singapore, Singapore, S. 151–160, 2018.
- [Ge20] Gehlhoff, F.; Fay A.: Agent-based decentralised architecture for multi-stage and integrated scheduling. 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vienna, Austria, 2020, pp. 1443-1446, 2020.
- [Ge23] Gehlhoff, F.: Agent-based Decentralised Architecture for Integrated Process Planning and Scheduling of Transport and Production Processes. Dissertation, Hamburg, 2023.
- [GES05] Gausemeier, J.; Ebbesmeyer, P.; Steffen, D.: Selbstoptimierung als Perspektive für die Gestaltung intelligenter Fertigungssysteme. In: Zeitschrift für wirtschaftlichen Fabrikbetrieb 100 (9), S. 495–501, 2005.
- [GJS76] Garey, M. R.; Johnson, D. S.; Sethi, R.: The Complexity of Flowshop and Jobshop Scheduling. Mathematics of Operations Research 2/1, S. 117–129, 1976.

- [Gö18] Göppert, A. et al.: Frei verkettete Montagesysteme. In: *Zeitschrift für wirtschaftlichen Fabrikbetrieb* 113 (3), S. 151–155, 2018.
- [Gö21] Göppert, A. et al.: Agile Hybrid Assembly Systems: Bridging the Gap Between Line and Matrix Configurations. In (Weißgraeber, P.; Heieck, F.; Ackermann, C. Hrsg.): *Advances in Automotive Production Technology – Theory and Application*. Springer Berlin Heidelberg, Berlin, Heidelberg, S. 3–11, 2021.
- [Gö22] Göppert, A.: Artificial intelligence in online scheduling of dynamically interconnected assembly systems. RWTH Aachen University, 2022.
- [Gr14] Greschke, P. et al.: Matrix Structures for High Volumes and Flexibility in Production Systems. *Procedia CIRP* 17, S. 160–165, 2014.
- [Gr19] Grunert, D. et al.: Service-oriented Information Model for the Model-Driven Control of Dynamically Interconnected Assembly Systems. In (Schüppstuhl, T.; Tracht, K.; Roßmann, J. Hrsg.): *Tagungsband des 4. Kongresses Montage Handhabung Industrieroboter*. Springer Berlin Heidelberg, Berlin, Heidelberg, S. 23–33, 2019.
- [GRS20] Göppert, A.; Rachner, J.; Schmitt, R. H.: Automated scenario analysis of reinforcement learning controlled line-less assembly systems. *Procedia CIRP* 93, S. 1091–1096, 2020.
- [GW18] Gronau, N.; Weber, E.: Reihenfolgeplanung im Zeitalter von Industrie 4.0. In: *prod* (1), 2018.
- [Ha10] Hansen, P. et al.: Variable Neighborhood Search. In (Gendreau, M.; Potvin, J.-Y. Hrsg.): *Handbook of Metaheuristics*. Springer US, Boston, MA, S. 61–86, 2010.
- [Ha19] Haberfellner, R. et al.: *Systems Engineering*. Springer International Publishing, Cham, 2019.
- [HBS19] Hüttemann, G.; Buckhorst, A. F.; Schmitt, R. H.: Modelling and Assessing Line-less Mobile Assembly Systems. *Procedia CIRP* 81, S. 724–729, 2019.
- [He02] Heragu, S. S. et al.: Intelligent agent based framework for manufacturing systems control. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 5/32, S. 560–573, 2002.
- [He03a] Hernández Morales, R.: *Systematik der Wandlungsfähigkeit in der Fabrikplanung*. Zugl.: Hannover, Univ., Diss., 2002. VDI-Verl., Düsseldorf, 2003.
- [He03b] Heragu, S. et al.: A hybrid scheduling and control system architecture for warehouse management. In: *IEEE Trans. Robot. Automat.* 19 (6), S. 991–1001, 2003.

- [He12] Herlyn, W.: PPS im Automobilbau. Produktionsprogrammplanung und -steuerung von Fahrzeugen und Aggregaten Mit 190 Bildern. Hanser, München, 2012.
- [He19a] Hees, F. et al.: PROJEKTATLAS KOMPETENZ MONTAGE KOLLABORATIV UND WANDLUNGSFÄHIG, 2019.
- [He19b] Hermann, J. et al.: Self-description of Cyber-Physical Production Modules for a product-driven manufacturing system. In: *Procedia Manufacturing* 38, S. 291–298, 2019.
- [He21] Hermann, J.: Dynamische Generierung alternativer Fertigungsfolgen im Kontext von Production as a Service. Dissertation, Kaiserslautern, 2021.
- [Hi00] Hilmer, S.: Objektorientierte Multiagenten-Systeme für die verteilte Produktionsplanung. Shaker, Aachen, 2000.
- [HL96] Hung, Y.-F.; Leachman, R. C.: A production planning methodology for semiconductor manufacturing based on iterative simulation and linear programming calculations. *IEEE Transactions on Semiconductor manufacturing* 2/9, S. 257–269, 1996.
- [Ho92] Holland, J. H.: *Adaptation in Natural and Artificial Systems*. The MIT Press, 1992.
- [HRR20] Hermann, J.; Rübel, P.; Ruskowski, M.: Development of a system architecture enabling a dynamic generation of process chains for Production as a Service: 2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS), S. 15–20, 2020.
- [IB13] Ilie, S.; Bădică, C.: Multi-agent approach to distributed ant colony optimization. In: *Science of Computer Programming* 78 (6), S. 762–774, 2013.
- [JP19] Jaehn, F.; Pesch, E.: *Ablaufplanung*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2019.
- [Ka13] Kagermann, H. et al.: Recommendations for implementing the strategic initiative INDUSTRIE 4.0: Securing the future of German manufacturing industry; final report of the Industrie 4.0 Working Group, 2013.
- [Ka20] Katti, B.: *Ontology-Based Approach to Decentralized Production Control in the Context of Cloud Manufacturing Execution Systems*, 2020.
- [Ka23] Kaiser, J. et al.: A review of reference architectures for digital manufacturing: Classification, applicability and open issues. *Computers in Industry* 149, S. 103923, 2023.
- [KB15] Kravari, K.; Bassiliades, N.: A Survey of Agent Platforms. *Journal of Artificial Societies and Social Simulation* 1/18, 2015.

- [KCF18] Kupzik, D.; Coutandin, S.; Fleischer, J.: Vorrichtungsfreies Fügen*/Fixtureless joining in car body assembly. In: *wt* 108 (10), S. 703–707, 2018.
- [Ke00] Kempf, K. et al.: Evaluation and comparison of production schedules. *Computers in Industry* 2-3/42, S. 203–220, 2000.
- [Ke21] Kern, W.: *Modulare Produktion*. Springer Fachmedien Wiesbaden, Wiesbaden, 2021.
- [KHM18] Kolberg, D.; Hermann, J.; Mohr, F.: SmartFactoryKL Systemarchitektur für Industrie 4.0-Produktionsanlagen: Whitepaper SF-1.2: 04/2018, 2018, url: https://smartfactory.de/wp-content/uploads/2018/04/SF_WhitePaper_SystemArchitecture_1-2_DE_XS.pdf, Stand: 14.03.2022.
- [KLB17] Kern, W.; Lämmermann, H.; Bauernhansl, T.: An Integrated Logistics Concept for a Modular Assembly System. In: *Procedia Manufacturing* 11, S. 957–964, 2017
- [Ko11] Kossiakoff, A. et al.: *Systems Engineering Principles and Practice*. Wiley, 2011.
- [Ko21] Komesker, S et al.: Structured Information Processing as Enabler of Versatile, Flexible Manufacturing Concepts. In (Weißgraeber, Heieck et al. (Hrsg.) - *Advances in Automotive Production Technology*. Springer International Publishing. S. 108-116, 2021
- [Ko22] Komesker, S et al.: Enabling a Multi-Agent System for Resilient Production Flow in Modular Production Systems. *Procedia CIRP* 107, 991-998, 2022
- [Ko23] Komesker, S. et al.: Multi-agent Interaction Structure for Enabling Subsidiary Planning and Control in Modular Production Systems. In (Kiefl, Wulle et al. (Hrsg.) – *Advances in Automotive Production Technology*. Springer International Publishing. S. 354-364, 2023.
- [Ko99] Koren, Y.; Heisel, U.; Jovane, F.; Moriwaki, T.; Pritschow, G.; Ulsoy, G.; van Brussel, H.: Reconfigurable Manufacturing Systems. In: *CIRP Annals* 48 (2), S. 527–540, 1999.
- [Kö23] Köcher, A. et al.: A reference model for common understanding of capabilities and skills in manufacturing. *at - Automatisierungstechnik* 2/71, S. 94–104, 2023.
- [KPK18] Kamdar, R.; Paliwal, P.; Kumar, Y.: A State of Art Review on Various Aspects of Multi-Agent System. *Journal of Circuits, Systems and Computers* 11/27, S. 1830006, 2018.
- [KRB16] Kern, W.; Rusitschka, F.; Bauernhansl, T.: Planning of Workstations in a Modular Automotive Assembly System. *Procedia CIRP* 57, S. 327–332, 2016.

- [KS10] Koren, Y.; Shpitalni, M.: Design of reconfigurable manufacturing systems. *Journal of Manufacturing Systems* 4/29, S. 130–141, 2010.
- [Ku19] Kusiak, A.: Service manufacturing: Basic concepts and technologies. *Journal of Manufacturing Systems* 52, S. 198–204, 2019.
- [LCR05] Leitaó, P.; Colombo, A. W.; Restivo, F. J.: ADACOR: A Collaborative Production Automation and Control Architecture. *IEEE Intelligent Systems* 1/20, S. 58–66, 2005.
- [Le09] Leitão, P.: Agent-based distributed manufacturing control: A state-of-the-art survey. *Engineering Applications of Artificial Intelligence* 7/22, S. 979–991, 2009.
- [Le15] Leitão, P. et al.: Multi-Agent System for Integrating Quality and Process Control in a Home Appliance Production Line. In (Leitão, P.; Karnouskos, S. Hrsg.): *Industrial Agents*. Morgan Kaufmann, Boston, S. 287–300, 2015.
- [Le16] Leitaó, P. et al.: Smart Agents in Industrial Cyber–Physical Systems. *Proceedings of the IEEE* 5/104, S. 1086–1101, 2016.
- [LEGO19] Bauanleitungen für 60239, Streifenwagen, <https://www.lego.com/de-de/service/buildinginstructions/60239>, zuletzt geprüft am: 09.12.2021.
- [LEGO20] Bauanleitungen für 60242, Festnahme auf der Autobahn, <https://www.lego.com/de-de/service/buildinginstructions/60242>, zuletzt geprüft am: 09.12.2021.
- [LF15] Lüder, A.; Foehr, M.: Identification and Implementation of Agents for Factory Automation Exploiting Mechatronical Concepts for Production System Structuring. In (Leitão, P.; Karnouskos, S. Hrsg.): *Industrial agents. Emerging applications of software agents in industry*. Elsevier, Amsterdam, Netherlands, S. 171–190, 2015.
- [LHS19] Lettmann, P.; Hüttemann, G.; Schmitt, R.: Produkttrouten in frei verketteten Montagesystemen. *Zeitschrift für wirtschaftlichen Fabrikbetrieb* 9/114, S. 517–520, 2019.
- [Li10] Lietaer, B. et al.: Is our monetary structure a systemic cause for financial instability? Evidence and remedies from nature. *Journal of Futures Studies* 14, 2010.
- [Li22] Li, X. et al.: Survey of integrated flexible job shop scheduling problems. *Computers & Industrial Engineering* 174, S. 108786, 2022.
- [LK15] Leitão, P.; Karnouskos, S. Hrsg.: *Industrial Agents*. Morgan Kaufmann, Boston, 2015.
- [LM13] Laguna, M.; Martí, R.: Heuristics. In (Gass, S. I.; Fu, M. C. Hrsg.): *Encyclopedia of Operations Research and Management Science*. Springer US, Boston, MA, S. 695–703, 2013.

- [Lo11] Lorenzer, T.: Wandelbarkeit in der Serienfertigung durch rekonfigurierbare Werkzeugmaschinen, 2011.
- [Lö16] Lödding, H.: Verfahren der Fertigungssteuerung. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016.
- [LR02] Leitão, P.; Restivo, F.: Holonic adaptive production control systems. IEEE 2002 28th Annual Conference of the Industrial Electronics Society. IECON 02 4, 2968–2973 vol.4, 2002.
- [LR03] Leitão, P.; Restivo, F.: Identification of ADACOR Holons for Manufacturing Control. IFAC Proceedings Volumes 3/36, S. 109–114, 2003.
- [LR08] Leitão, P.; Restivo, F.: Implementation of a Holonic Control System in a Flexible Manufacturing System. In: *IEEE Trans. Syst., Man, Cybern. C* 38 (5), S. 699–709, 2008.
- [LS15] Lee, E. A.; Seshia, S. A.: Introduction to embedded systems. A cyber-physical systems approach. LeeSeshia.org, Lulu, 2015.
- [LVG18] Lüder, A.; Vogel-Heuser, B.; Göhner, P.: Elektronische Datenverarbeitung – Agentenbasiertes Steuern. In (Grote, K.-H.; Bender, B.; Göhlich, D. Hrsg.): *Dubbel: Taschenbuch für den Maschinenbau*. Springer Berlin Heidelberg, Berlin, Heidelberg, S. 153–170, 2018.
- [Ma06] MacKenzie, C. et al.: Reference model for service oriented architecture 1.0. Public Rev. Draft 2, 2006.
- [Ma19a] Maoudj, A. et al.: Distributed multi-agent scheduling and control system for robotic flexible assembly cells. *Journal of Intelligent Manufacturing* 4/30, S. 1629–1644, 2019.
- [Ma19b] Mayer, S. et al.: Standardized Framework for Evaluating Centralized and Decentralized Control Systems in Modular Assembly Systems. 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), S. 113–119, 2019.
- [Ma21a] May, M. C. et al.: Product Generation Module: Automated Production Planning for optimized workload and increased efficiency in Matrix Production Systems. *Procedia CIRP* 96, S. 45–50, 2021.
- [Ma21b] May, M. C. et al.: Decentralized Multi-Agent Production Control through Economic Model Bidding for Matrix Production Systems. *Procedia CIRP* 96, S. 3–8, 2021.
- [Ma23] Mathews, J. B. et al.: Industrial applications of a modular software architecture for line-less assembly systems based on interoperable digital twins. *Frontiers in Mechanical Engineering* 9, 2023.

- [Ma60] Manne, A. S.: On the Job-Shop Scheduling Problem. *Operations Research* 2/8, S. 219–223, 1960.
- [Ma96] Maier, M. W.: Architecting Principles for Systems-of-Systems. *INCOSE International Symp (INCOSE International Symposium)* 6/1, S. 565–573, 1996 .
- [Ma98] Maier, M. W.: Architecting principles for systems-of-systems. *Systems Engineering* 4/1, S. 267–284, 1998.
- [MCE21] Mayer, S.; Classen, T.; Endisch, C.: Modular production control using deep reinforcement learning: proximal policy optimization. *Journal of Intelligent Manufacturing* 8/32, S. 2335–2351, 2021.
- [Me09] Mendes, J. M. et al.: Service-Oriented Agents for Collaborative Industrial Automation and Production Systems. In (Mařík, V.; Strasser, T.; Zoitl, A. Hrsg.): *Holonic and Multi-Agent Systems for Manufacturing. 4th International Conference on Industrial Applications of Holonic and Multi-Agent Systems, HoloMAS 2009, Linz, Austria, August 31 - September 2, 2009, Proceedings*. Springer Berlin Heidelberg, Berlin, Heidelberg, S. 13–24, 2009.
- [Me12] Metawei, M. A. et al.: Load balancing in distributed multi-agent computing systems. In: *Ain Shams Engineering Journal* 3 (3), S. 237–249, 2012.
- [ME19] Mayer, S.; Endisch, C.: Adaptive Production Control in a Modular Assembly System Based on Partial Look-Ahead Scheduling. *2019 IEEE International Conference on Mechatronics (ICM)*, S. 293–300, 2019.
- [MFM13] Mönch, L.; Fowler, J. W.; Mason, S. J.: *Production Planning and Control for Semiconductor Wafer Fabrication Facilities*. Springer New York, New York, NY, 2013.
- [MM05] Marik, V.; McFarlane, D.: Industrial Adoption of Agent-Based Technologies. *IEEE Intelligent Systems* 1/20, S. 27–35, 2005.
- [Mö06] Mönch, L.: Autonome und kooperative Steuerung komplexer Produktionsprozesse mit Multi-Agenten-Systemen. In: *Wirtsch. Inform.* 48 (2), S. 107–119., 2006.
- [MSZ03] Mönch, L.; Stehli, M.; Zimmermann, J.: FABMAS: An Agent-Based System for Production Control of Semiconductor Manufacturing Processes. In (Mařík, V.; McFarlane, D.; Valckenaers, P. Hrsg.): *Holonic and Multi-Agent Systems for Manufacturing*. Springer Berlin Heidelberg, Berlin, Heidelberg, S. 258–267, 2003.
- [MVK06] Monostori, L.; Váncza, J.; Kumara, S.: Agent-Based Systems for Manufacturing. *CIRP Annals* 2/55, S. 697–720, 2006.

- [MW03] McKay, K. N.; Wiers, V. C. S.: Planning, scheduling and dispatching tasks in production control. *Cognition, Technology & Work* 5, S. 82–93, 2003.
- [Ng17] Ngu, A. H. H. et al.: IoT Middleware: A Survey on Issues and Enabling Technologies. *IEEE Internet of Things Journal* 4, S. 1–20, 2017.
- [Ni23] Nielsen, C. P. et al.: Control system architecture for matrix-structured manufacturing systems. *Computers in Industry* 146, S. 103851, 2023.
- [Ny08a] Nyhuis, P. Hrsg.: Beiträge zu einer Theorie der Logistik. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.
- [Ny08b] Nyhuis, P.: Produktionskennlinien — Grundlagen und Anwendungsmöglichkeiten. In (Nyhuis, P. Hrsg.): Beiträge zu einer Theorie der Logistik. Springer Berlin Heidelberg, Berlin, Heidelberg, S. 185–218, 2008.
- [NY22] Nielsen, C. P.; Yu, F.: Product Design for Matrix-Structured Manufacturing Systems. *Procedia CIRP* 109, S. 407–412, 2022.
- [OBF06] Onori, M.; Barata, J.; Frei, R.: Evolvable Assembly Systems Basic Principles: Information Technology For Balanced Manufacturing Systems. Springer US, Boston, MA, S. 317–328, 2006.
- [ÖÖY10] Özgüven, C.; Özbakır, L.; Yavuz, Y.: Mathematical models for job-shop scheduling problems with routing and process plan flexibility. In: *Applied Mathematical Modelling* 34 (6), S. 1539–1548, 2010.
- [OP09] Ouelhadj, D.; Petrovic, S.: A survey of dynamic scheduling in manufacturing systems. *Journal of Scheduling* 4/12, S. 417–431, 2009.
- [Os12] Ostgathe, Martin: System zur produktbasierten Steuerung von Abläufen in der auftragsbezogenen Fertigung und Montage, 2012.
- [Pe14] Peschl, Michael: An architecture for flexible manufacturing systems based on task-driven agents, 2014.
- [Pe16] Petrović, M. et al.: Integration of process planning and scheduling using chaotic particle swarm optimization algorithm. In: *Expert Systems with Applications* 64, S. 569–588, 2016.
- [Pf15] Pfrommer, J. et al.: Plug & produce by modelling skills and service-oriented orchestration of reconfigurable manufacturing systems. *at - Automatisierungstechnik* 10/63, S. 790–800, 2015.
- [Pi16] Pinedo, M. L. (2016): Scheduling. Cham: Springer International Publishing, 2016

- [PI22] Plattform Industrie 4.0: Umsetzung von cyber-physischen Matrixproduktionssystemen. Forschungsbeirat Industrie 4.0/acatech – Deutsche Akademie der Technikwissenschaften, 2022.
- [PLK05] Peschke, J.; Luder, A.; Kuhnle, H.: The PABADIS’PROMISE architecture - a new approach for flexible manufacturing systems: 2005 IEEE Conference on Emerging Technologies and Factory Automation, 6 p-496, 2005.
- [PYR21] Popper, J.; Yfantis, V.; Ruskowski, M.: Simultaneous Production and AGV Scheduling using Multi-Agent Deep Reinforcement Learning. *Procedia CIRP* 104, S. 1523–1528, 2021.
- [R 20] R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing.
- [RB16] Rautmare, S.; Bhalerao, D. M.: MySQL and NoSQL database comparison for IoT application: 2016 IEEE International Conference on Advances in Computer Applications (ICACA). IEEE, S. 235–238, 2016.
- [RB18] Ribeiro, L.; Bjorkman, M.: Transitioning From Standard Automation Solutions to Cyber-Physical Production Systems: An Assessment of Critical Conceptual and Technical Challenges. In: *IEEE Systems Journal* 12 (4), S. 3816–3827, 2018.
- [RBC08] Ribeiro, L.; Barata, J.; Colombo, A.: MAS and SOA: A Case Study Exploring Principles and Technologies to Support Self-Properties in Assembly Systems: 2008 Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops. IEEE, S. 192–197, 2008 - 2008.
- [Re17] Reinhart, G. Hrsg.: Handbuch Industrie 4.0. Geschäftsmodelle, Prozesse, Technik. Carl Hanser Verlag; Ciando, München, 2017.
- [RN03] Russell, S. J.; Norvig, P.: Artificial Intelligence. A Modern Approach Contributing Writers John F. Canny, Douglas D. Edwards, Jitendra M. Malik [et autres]. Pearson Education, New Jersey, 2003.
- [Ro09] Ropohl, G.: Allgemeine Technologie eine Systemtheorie der Technik. KIT Scientific Publishing, 2009.
- [RH18] Ribeiro, L.; Hochwallner, M.: On the Design Complexity of Cyberphysical Production Systems. In: *Complexity* 2018, S. 1–13, 2018.
- [Ru21] Ruskowski, M.: Komplexe Produktionssysteme beherrschbar machen - SmartFactory-KL zeigt Demonstrator LIVE und digital, Kaiserslautern, 2021.

- [Ru22] Ruskowski, M.: "Weg zur Smart Factory wird Schritt für Schritt begangen". <https://www.automobil-produktion.de/produktion/smart-factory/weg-zur-smart-factory-wird-schritt-fuer-schritt-begangen-986.html>, Stand: 16.01.2024.
- [RZL21] Reich, J.; Zentarra, L.; Langer, J.: Industrie 4.0 und das Konzept der Verwaltungsschale – Eine kritische Auseinandersetzung. *HMD Praxis der Wirtschaftsinformatik* 3/58, S. 661–675, 2021.
- [SAV12] Schuh, G.; Arnoscht, J.; Völker, M.: Product Design Leverage on the Changeability of Production Systems. *Procedia CIRP* 3, S. 305–310, 2012.
- [SBS18] Schober, P.; Boer, C.; Schwarte, L. A.: Correlation Coefficients: Appropriate Use and Interpretation. *Anesthesia and analgesia* 5/126, S. 1763–1768, 2018.
- [SC01] Sage, A. P.; Cuppan, C. D.: On the Systems Engineering and Management of Systems of Systems and Federations of Systems. *Information Knowledge Systems Management* 2, S. 325–345, 2001.
- [Sc96] Schwaninger, M.: Systemtheorie. In: Kern, W. (Hrsg.): *Handwörterbuch der Produktionswirtschaft*, Stuttgart: Schäffer-Poeschel, S. 1946–1960; 1996.
- [Sc15] Schönemann, M. et al.: Simulation of matrix-structured manufacturing systems. *Journal of Manufacturing Systems* 37, S. 104–112, 2015.
- [Sc17] Schönemann, M.: *Multiscale Simulation Approach for Battery Production Systems*. Springer Cham, 2017
- [Sc19a] Schönemann, M. et al.: Multiscale simulation approach for production systems. *The International Journal of Advanced Manufacturing Technology* 5-8/102, S. 1373–1389, 2019.
- [Sc19b] Schulz, T.: 4. Referenzarchitektur und Industrie-4.0-Komponente. In (Axel Mangelsdorf; Petra Weiler Hrsg.): *Normen und Standards für die digitale Transformation Werkzeuge, Praxisbeispiele und Entscheidungshilfen für innovative Unternehmen, Normungsorganisationen und politische Entscheidungsträger*. De Gruyter Oldenbourg, Berlin, Boston, S. 37–52, 2019.
- [Sc21] Schmidtke Nils, Rettmann Alina, Behrendt Fabian: Matrix Production Systems - Requirements and Influences on Logistics Planning for Decentralized Production Structures. In (Bui, T. Hrsg.): *Proceedings of the 54th Hawaii International Conference on System Sciences*. Hawaii International Conference on System Sciences, S. 1655–1674, 2021.
- [SD08] Sivanandam, S. N.; Deepa, S. N.: *Introduction to Genetic Algorithms*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

- [SF07] Scholz-Reiter, B.; Freitag, M.: Autonomous Processes in Assembly Systems. *CIRP Annals* 2/56, S. 712–729, 2007.
- [SG13] Sörensen, K.; Glover, F. W.: Metaheuristics. In (Gass, S. I.; Fu, M. C. Hrsg.): *Encyclopedia of Operations Research and Management Science*. Springer US, Boston, MA, S. 960–970, 2013.
- [SHS09] Sergienko, I. V.; Huliannytskyi, L. F.; Sirenko, S. I.: Classification of applied methods of combinatorial optimization. In: *Cybern Syst Anal* 45 (5), S. 732–741, 2009.
- [Sh10] Shao, X. et al.: Metaheuristic approaches to sequencing mixed-model fabrication/assembly systems with two objectives. In: *Int J Adv Manuf Technol* 48 (9-12), S. 1159–1171, 2010.
- [SK99] Saygin, C.; Kilic, S. E.: Integrating Flexible Process Plans with Scheduling in Flexible Manufacturing Systems. In: *The International Journal of Advanced Manufacturing Technology* 15 (4), S. 268–280, 1999.
- [SL09] Shoham, Y.; Leyton-Brown, K.: Multiagent systems. Algorithmic, game-theoretic, and logical foundations. Cambridge: Cambridge University Press, 2009.
- [Sp13] Spath, D. Hrsg.: *Produktionsarbeit der Zukunft - Industrie 4.0. Studie*. Fraunhofer-Verl., Stuttgart, 2013.
- [SS12] Schuh, G.; Stich, V.: *Produktionsplanung und -steuerung 1*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [SS14] Schuh, G.; Schmidt, C.: *Produktionsmanagement*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.
- [SSZ21] Seifi, C.; Schulze, M.; Zimmermann, J.: A new mathematical formulation for a potash-mine shift scheduling problem with a simultaneous assignment of machines and workers. *European Journal of Operational Research* 1/292, S. 27–42, 2021.
- [St21] Stricker, N. et al.: Self-adjusting multi-objective scheduling based on Monte Carlo Tree Search for matrix production assembly systems. *CIRP Annals* 1/70, S. 381–384, 2021.
- [Te98] Telsang, M. T.: *Industrial engineering and production management*. S. Chand Publishing, 1998.
- [TFB20] Trierweiler, M.; Foith-Förster, P.; Bauernhansl, T.: Changeability of Matrix Assembly Systems. *Procedia CIRP* 93, S. 1127–1132, 2020.

- [Tr09] Trentesaux, D.: Distributed control of production systems. In: Engineering Applications of Artificial Intelligence 22 (7), S. 971–978., 2009.
- [Tr19] Trunzer, E. et al.: System architectures for Industrie 4.0 applications. Production Engineering 3/13, S. 247–257, 2019.
- [TS97] Turner, L.; Sandberg, A.: Enriching Production: Perspectives on Volvo's Uddevalla Plant as an Alternative to Lean Production. Industrial and Labor Relations Review 2/50, S. 359, 1997.
- [Ue92] Ueda, K.: A concept for bionic manufacturing systems based on DNA-type information: Human Aspects in Computer Integrated Manufacturing. Elsevier, S. 853–863, 1992.
- [Un15] Unland, R.: Chapter2 - Industrial Agents. In (Leitão, P.; Karnouskos, S. Hrsg.): Industrial Agents. Morgan Kaufmann, Boston, S. 23–44, 2015.
- [Va19a] Valckenaers, P.: ARTI Reference Architecture – PROSA Revisited. In (Borangiu, T. et al. Hrsg.): Service Orientation in Holonic and Multi-Agent Manufacturing. Springer International Publishing, Cham, S. 1–19, 2019.
- [Va19b] Vathoopan, M. et al.: Planning and Engineering Component-Based Automation Systems in AutomationML. Unpublished, 2019.
- [vB90] van Brussel, H.: Planning and Scheduling of Assembly Systems. CIRP Annals 2/39, S. 637–644, 1990.
- [vBV98] van Brussel, H. et al.: Reference architecture for holonic manufacturing systems: PROSA. Computers in Industry 3/37, S. 255–274, 1998.
- [Ve06] Verstraete, P. et al.: On Applying the PROSA Reference Architecture in Multiagent Manufacturing Control Applications: Multiagent Systems and Software Architecture, 2006.
- [VGL15] Vogel-Heuser, B.; Göhner, P.; Lüder, A.: Agent-Based Control of Production Systems - and its Architectural Challenges. In (Leitão, P.; Karnouskos, S. Hrsg.): Industrial Agents. Morgan Kaufmann, Boston, S. 153–170, 2015.
- [VHL03] Vieira, G. E.; Herrmann, J. W.; Lin, E.: Rescheduling Manufacturing Systems: A Framework of Strategies, Policies, and Methods. Journal of Scheduling 1/6, S. 39–62, 2003.
- [VLL15] Vogel-Heuser, B.; Lee, J.; Leitão, P.: Agents enabling cyber-physical production systems. at - Automatisierungstechnik 10/63, S. 777–789, 2015.

- [Vo09] Vogel-Heuser, B. et al.: Global Information Architecture for Industrial Automation. Automatisierungstechnische Praxis (atp) 1, S. 108–115, 2009.
- [Vo18] Vogel-Heuser, B. et al.: Agentenmuster für flexible und rekonfigurierbare Industrie 4.0/CPS- Automatisierungsbzw. Energiesysteme: Automation 2018. VDI Verlag, S. 1119–1130, 2018.
- [Vo20] Vogel-Heuser, B. et al.: Multi-agent systems to enable Industry 4.0. at - Automatisierungstechnik 6/68, S. 445–458, 2020.
- [Vo21] Volkmann, M. et al.: Integration of a feasibility and context check into an OPC UA skill. In: IFAC-PapersOnLine 54 (1), S. 276–281, 2021.
- [Vo23] Volkmann, M.: Modellierung von Dienstleistungen, Capabilities und Skills zur Einbindung von Lohnfertigern in eine marktplatzbasierte Fertigung. Dissertation, Kaiserslautern, 2023.
- [VV06] Verstraete, P.; Valckenaers, P.: TOWARDS COOPERATING PLANNING AND MANUFACTURING EXECUTION SYSTEMS. IFAC Proceedings Volumes 3/39, S. 393–398, 2006.
- [VvB15] Valckenaers, P.; van Brussel, H.: Holonic Manufacturing Systems. In (Valckenaers, P.; van Brussel, H. Hrsg.): Design for the Unexpected. From holonic manufacturing systems towards a humane mechatronics society. Butterworth-Heinemann is an imprint of Elsevier, Oxford, UK, S. 41–76, 2015.
- [Wa18] Waschneck, B. et al.: Optimization of global production scheduling with deep reinforcement learning. In: Procedia CIRP 72, S. 1264–1269, 2018.
- [Wa20] Waschneck, B.: Autonome Entscheidungsfindung in der Produktionssteuerung komplexer Werkstattfertigungen. Universität Stuttgart, 2020.
- [Wa59] Wagner, H. M.: An integer linear-programming model for machine scheduling. Naval Research Logistics Quarterly 2/6, S. 131–140, 1959.
- [Wa92] Warnecke, H.-J.: Die Fraktale Fabrik. Springer Berlin Heidelberg, Berlin, Heidelberg, 1992.
- [We00] Westkämper, E. et al.: Ansätze zur Wandlungsfähigkeit von Produktionsunternehmen: Ein Bezugsrahmen für die Unternehmensentwicklung im turbulenten Umfeld. Werkstattstechnik - Springer 1/2/90 (2000), S. 22–26, 2000.
- [We01] Weiß, G.: Agentenorientiertes Software Engineering. Informatik-Spektrum 2/24, S. 98–101, 2001.

- [We15] Weilkiens, T. et al.: Model-Based System Architecture. Wiley, 2015.
- [WF17] Wassermann, E.; Fay, A.: Interoperability rules for heterogenous multi-agent systems: Levels of conceptual interoperability model applied for multi-agent systems: 2017 IEEE 15th International Conference on Industrial Informatics (INDIN). IEEE, S. 89–95, 2017.
- [Wi07] Wiendahl, H.-P. et al.: Changeable Manufacturing - Classification, Design and Operation. CIRP Annals 2/56, S. 783–809, 2007.
- [Wi09] Wiendahl, H.-P.: Veränderungsfähigkeit von Produktionsunternehmen. Zeitschrift für wirtschaftlichen Fabrikbetrieb 1-2/104, S. 32–37, 2009.
- [Wi14] Wiendahl, H.-P.: Betriebsorganisation für Ingenieure. Hanser, München, 2014.
- [Wi97] Wiendahl, H.-P.: Fertigungsregelung. Logistische Beherrschung von Fertigungsabläufen auf Basis des Trichtermodells ; 42 Tabellen. München, Wien: Hanser, 1997
- [WJ95] Wooldridge, Michael; Jennings, Nicholas R.: Intelligent agents: theory and practice. In: The Knowledge Engineering Review 10 (2), S. 115–152, 1995.
- [WR17] Weyer, J.; Roos, M.: Agentenbasierte Modellierung und Simulation. TATuP - Zeitschrift für Technikfolgenabschätzung in Theorie und Praxis 3/26, S. 11–16, 2017.
- [WS20] Wu, K.; Shen, Y.: A unified view on planning, scheduling and dispatching for a factory. Advanced Engineering Informatics 46, S. 101188, 2020.
- [WW20] Wiendahl, Hans-Peter; Wiendahl, Hans-Hermann: Betriebsorganisation für Ingenieure. 9., vollständig überarbeitete Auflage. München: Hanser, 2020.
- [WWB98] Westkämper, E.; Wiendahl, H.-H.; Balve, P.: Dezentralisierung und Autonomie in der Produktion. Zeitschrift für wirtschaftlichen Fabrikbetrieb 9/93, S. 407–410, 1998.
- [Yf22a] Yfantis, V et al.: An Improved Optimization Model for Scheduling of an Industrial Formulation Plant based on Integer Linear Programming. In: Yoshiyuki Yamashita und Manabu Kano (Hg.): 14th International Symposium on Process Systems Engineering, Bd. 49: Elsevier (Computer Aided Chemical Engineering), S. 487–492, 2022.
- [Yf22b] Yfantis, V. et al.: Hierarchical Distributed Model Predictive Control based on Dual Decomposition and Quadratic Approximation. In: 2022 30th Mediterranean Conference on Control and Automation (MED). Mediterranean Control Association. Piscataway, NJ: IEEE, S. 914-919, 2022

- [YH18] Ye, X.; Hong, S. H.: An AutomationML/OPC UA-based Industry 4.0 Solution for a Manufacturing System: 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), S. 543–550, 2018.
- [Za19] Zawisza, J.: Entwicklung und Integration interdependenter Agentensysteme zur dezentralen Produktionsplanung und -steuerung. Otto von Guericke University Library, Magdeburg, Germany, 2019.
- [ZE20] Zanjirani, M. D.; Esmaelian, M.: An integrated approach based on Fuzzy Inference System for scheduling and process planning through multiple objectives. In: Journal of Industrial & Management Optimization 16 (3), S. 1235–1259, 2020.
- [Zh20] Zhang, Sicheng; Li, Xiang; Zhang, Bowen; Wang, Shouyang: Multi-objective optimisation in flexible assembly job shop scheduling using a distributed ant colony system. In: European Journal of Operational Research 283 (2), S. 441–460, 2020.
- [Zi08] Zimmermann, J.: Adaptive Multi-Agenten-Systeme zur Steuerung komplexer Produktionssysteme, Hagen, 2008.
- [Zi19] Zimmermann, P. et al.: Skill-based Engineering and Control on Field-Device-Level with OPC UA: 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). IEEE, S. 1101–1108, 2019.
- [ZMV05] Zaeh, M. F.; Moeller, N.; Vogl, W.: Changeable Agile Reconfigurable and Virtual Production Symbiosis of Changeable and Virtual Production – The Emperor ’ s New Clothes or Key Factor for Future Success ?, 2005.

Betreute studentische Arbeiten

- [He20] Henke, L.: Wege zur Vision der autonomen Produktionssysteme. Masterarbeit, Magdeburg, 2020.
- [Ho20] Hofer, D.: Erarbeitung eines Konzeptes für die Informationsflusssteuerung von flexiblen Fertigungsinseln in der Automobilindustrie. Studienarbeit, Braunschweig, 2020.
- [Mu21] Mueller, L.: Ablaufplanung in Job Shop-Umgebungen unter Berücksichtigung flexibler Vorrangbeziehungen und paralleler Arbeitsgänge. Masterarbeit, Hildesheim, 2021.
- [Ba21] Bartels, J. : Agentenbasierte Modellierung und Analyse von Fertigungsstrategien für ein modulares Produktionssystem. Masterarbeit. Braunschweig, 2021.
- [Sch22] Schwartz, J.: Datenbasierte Entscheidungsunterstützung eines Multiagentensystems für die Prozesssteuerung von Feldgeräten in einem modularen Produktionsumfeld. Bachelorarbeit, Hildesheim, 2022.

Lebenslauf

Persönliche Daten

Name **Simon Komesker**

Staatsangehörigkeit **deutsch**

Bildung

03/2019-03-2025 **TU Kaiserslautern / RPTU Kaiserslautern-Landau, Kaiserslautern (Deutschland)**

Dr.-Ing. Industriepromotion Produktionssteuerung

09/2017-01/2018 **Tokyo University of Science (TUS), Tokio (Japan)**

Auslandssemester: Supply Chain Management; Masterarbeit

04/2016-04/2018 **Leibniz Universität Hannover, Hannover (Deutschland)**

M.Sc. Wirtschaftsingenieurwesen (Produktionstechnik)

08/2014-03/2015 **Universidade de São Paulo (USP), São Carlos (Brasilien)**

Auslandssemester: Produktionstechnik; Bachelorarbeit

10/2011-03/2016 **Leibniz Universität Hannover, Hannover (Deutschland)**

B.Sc. Wirtschaftsingenieurwesen (Produktionstechnik)

09/2004-08/2011 **Gymnasium Melle, Melle (Deutschland)**

Abitur

Berufliche Erfahrung

Seit 01/2022 **Volkswagen AG, Wolfsburg (Deutschland)**

Planungsingenieur Digitales Produktionsmanagement / Shopfloor IT

03/2019 – 01/2022 **Volkswagen AG, Wolfsburg (Deutschland)**

Doktorand Digitales Produktionsmanagement

05/2018 - 02/2019 **ABB Automation AG, Friedberg (Deutschland) / Zürich (Schweiz)**

Global Trainee Procurement & Logistics

11/2015 – 09/2016 **Bentley Motors Ltd., Crewe (Großbritannien)**

Praktikant Produktionsplanung Vorserie

04/2015 – 10/2015 **Volkswagen Slovakia a.s., Bratislava (Slowakei)**

Praktikant Karosseriebau Anlauf C-SUV