

Otto-von-Guericke-Universität Magdeburg



OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

MB

FAKULTÄT FÜR
MASCHINENBAU

Entwicklung und Integration interdependenter Agentensysteme zur dezentralen Produktionsplanung und -steuerung

Dissertation

zur Erlangung des akademischen Grades

Doktoringenieur

(Dr.-Ing.)

von Dipl.-Wirtsch.-Ing. Jacek Zawisza
geboren am 15.04.1983 in Katowice, Polen

genehmigt durch die Fakultät für Maschinenbau
der Otto-von-Guericke-Universität Magdeburg

Gutachter: Prof. Dr.-Ing. Arndt Lüder
Prof. Dr. Frank Werner

Promotionskolloquium am 13.12.2018

Kurzfassung

Zunehmender globaler Wettbewerb und hohe Nachfrage nach kundenindividualisierten Produkten erhöhen rasant die Komplexität sowohl der Produkte selbst als auch der Produktionsprozesse in der Fertigungsindustrie. Um den daraus resultierenden Anforderungen gerecht werden zu können, bedarf es neuer Ansätze für die Produktion von morgen. Die Vision der Industrie 4.0 möchte diesen Herausforderungen begegnen, indem Produktionsprozesse zunehmend vernetzt und Flexibilität, Autonomie und Robustheit in der Fertigung gesteigert werden. Derzeit ist jedoch unklar, wie sich die Komponenten in einem Industrie 4.0-Produktionssystem definieren, um in der Realität umgesetzt zu werden, und welche Informationen sie enthalten müssen. Zudem existiert eine Reihe von potenziellen Umsetzungstechnologien für die Realisierung der Vision von Industrie 4.0. Agentenbasierte Steuerungssysteme haben sich in diesem Zusammenhang zu einem vielversprechenden Ansatz entwickelt, um die zunehmend komplexen Produktionssteuerungssysteme zu dezentralisieren und zu vernetzen. In einer Reihe von Forschungs- und Industrieprojekten haben Agentensysteme bereits ihre Nützlichkeit für viele Anwendungsfälle unter Beweis gestellt, indem sie Teilsysteme komplexer Produktionssysteme abbilden und steuern. Es fehlt derzeit jedoch noch an geeigneten Methoden, um die heute getrennt voneinander agierenden Steuerungssysteme in ein interdependentes, ganzheitliches Agentensystem zu integrieren, das eine gemeinsame Zielsetzung und ein globales Optimum verfolgt. Die vorliegende Arbeit versucht diese Forschungslücke zu schließen, indem eine agentenbasierte Entwicklungsmethode erarbeitet wird, mit der die Entwicklung individueller Agentensysteme und ihre schrittweise Integration ermöglicht werden soll. Auf dem Weg dahin werden Beispiele und Anwendungsfälle aus der Fertigungsindustrie und insbesondere aus dem Umfeld der Automobilfertigung herangezogen und konkrete Agentensysteme entworfen, programmiert und simuliert. Die erarbeitete Entwicklungsmethode zur Integration agentenbasierter Steuerungssysteme wird ebenfalls anhand von Anwendungsfällen demonstriert. Weiterhin werden Wege aufgezeigt, wie Unternehmen damit die Migration von der derzeit zentral organisierten Produktion hin zu einem dezentral organisierten Produktionsnetzwerk aus intelligenten und integrierten Agentensystemen beschreiten können.

Abstract

Continuously growing demands of global competition and customer-individualized products are rapidly increasing the complexity of products and production processes in the manufacturing industry. To meet the resulting challenges, new approaches are needed for tomorrow's production. The Industrie 4.0 vision is to meet these challenges by increasingly networking production processes and boosting flexibility, autonomy and robustness in manufacturing. However, it is currently unclear how the components in an industry 4.0 production system define themselves in order to be implemented in real-life applications and what information they must contain. In addition, there is a range of potential implementation technologies for realizing the vision of Industrie 4.0. Agent-based control systems have become a promising approach for decentralising and networking the increasingly complex production control systems. In a number of research and industrial projects, agent systems have already proven their usefulness for many applications by mapping and controlling subsystems of complex production systems. However, there is still a lack of suitable methods for integrating today's separate control systems into an interdependent, holistic agent system that pursues a common goal and a global optimum. This work tries to close this research gap by developing an agent-based development method that enables the development of individual agent systems and their gradual integration. Along the way, examples and applications from the manufacturing industry and in particular from the field of automotive production are used and concrete agent systems are designed, programmed and simulated. The elaborated development method for the integration of agent-based control systems is demonstrated on the basis of use cases and ways are shown how companies can use the method to migrate from the currently centrally organized production to a decentralized production network of intelligent and integrated agent systems.

Inhaltsverzeichnis

KURZFASSUNG	III
ABSTRACT	V
INHALTSVERZEICHNIS	VII
ABBILDUNGSVERZEICHNIS	XII
TABELLENVERZEICHNIS	XV
ABKÜRZUNGEN	XVI
1 EINLEITUNG	1
1.1 Ausgangssituation und Problemstellung	1
1.2 Motivation und Zielsetzung	4
1.3 Aufbau der Arbeit	8
2 PRODUKTIONSPLANUNG UND -STEUERUNG IM AUTOMOBILBAU	11
2.1 Grundlagen der Automobilfertigung	11
2.1.1 Fertigungs- und Kundenauftragsprozess.....	11
2.1.2 Arbeitsorganisation in der Automobilfertigung	13
2.2 Grundlagen der Produktionsplanung und -steuerung	17
2.2.1 Aufgaben der PPS und Umsetzung in der Automobilindustrie	17
2.2.2 Aufgaben der Fertigungssteuerung	21
2.2.3 Automatisierungstechnische Sicht auf die PPS	23
2.3 Produktionssteuerung der Zukunft	25
2.3.1 Zunehmende Dynamität und Folgen für die Produktion	26
2.3.2 Anforderungen an moderne Produktionssteuerungssysteme	30
3 PARADIGMEN DER PRODUKTIONSSTEUERUNG	37
3.1 Status quo Industrie 4.0	37
3.2 Steuerungsparadigmen und -architekturen	38
3.2.1 Zentrale Steuerungsarchitekturen	39
3.2.2 Hierarchische Steuerungsarchitekturen.....	39
3.2.3 Dezentrales Steuerungsparadigma	41

3.2.4	Heterarchische Steuerungsarchitekturen	43
3.2.5	Hybride Steuerungsarchitekturen und Holonic Manufacturing Systems	43
3.3	Software-Design-Paradigmen zur Umsetzung von Industrie 4.0	45
3.3.1	Speicherprogrammierbare Steuerungen nach IEC 61131	45
3.3.2	Verteilte Automatisierung nach IEC 61499	47
3.3.3	Serviceorientierte Architekturen	52
3.3.4	Agentensysteme	56
3.4	Bewertung der Paradigmen zur Umsetzung von Industrie 4.0	63
4	STRUKTUR UND DESIGN INDUSTRIELLER AGENTENSYSTEME	67
4.1	Agentenbasierte Steuerungsarchitekturen	67
4.1.1	PROSA	67
4.1.2	PABADIS und PABADIS’PROMISE	68
4.1.3	ADACOR	68
4.1.4	GRACE	69
4.1.5	Agenten auf Feldsteuerungsebene	70
4.1.6	Allgemeingültige Entwurfsmuster	71
4.2	Entwicklungsmethoden für Agentensysteme	73
4.2.1	Gaia (2000)	74
4.2.2	MaSE (2001)	76
4.2.3	Tropos (2002)	77
4.2.4	Prometheus (2002)	78
4.2.5	DACS (2004)	80
4.2.6	ADMARMS (2015)	82
4.2.7	Bewertung und Auswahl einer Entwicklungsmethode	83
5	STEUERUNGSAUFGABEN IN PRODUKTIONSSYSTEMEN	87
5.1	Generische Fertigungssystemstruktur	87
5.2	Steuerungsaufgaben und -informationen	90
6	AUSWAHL VON ANWENDUNGSFÄLLEN UND VORGEHEN	95
6.1	Anforderungen und Optionsraum	95

6.1.1	Herleitung von Anforderungen an Anwendungsfälle	95
6.1.2	Optionsraum	96
6.2	Auswahl geeigneter Anwendungsfälle	98
6.3	Weiteres Vorgehen.....	100
6.3.1	Auswahl geeigneter Agentenplattformen.....	101
6.3.2	Evaluation der Anwendungsfälle in Simulationsstudien.....	103
7	ANWENDUNGSFALL 1: KAROSSERIEFOLGESTEUERUNG	107
7.1	Ausgangssituation und Zielsetzung	107
7.1.1	Problemstellung	107
7.1.2	Produktionssteuerung nach dem Perlenkettenprinzip.....	110
7.1.3	Scheduling-Verfahren zur Reihenfolgebildung im Fahrzeugbau	112
7.1.4	Agentensysteme für die Reihenfolgesteuerung	115
7.1.5	Zielsetzung	117
7.2	Entwicklung des Agentensystems.....	117
7.2.1	Entscheidungsmodell.....	118
7.2.2	Agentenmodell.....	122
7.2.3	Interaktionsmodell.....	125
7.3	Programmierung des Agentensystems	131
7.3.1	Auftragsfreigabe.....	131
7.3.2	Allokation und Reservierung von Ressourcen.....	136
7.3.3	Initiierung und Überwachung von Produktionsaufträgen	140
7.4	Simulationsexperimente	141
7.4.1	Design der Simulationsexperimente	141
7.4.2	Simulationsergebnisse und kritische Beurteilung	145
8	ANWENDUNGSFALL 2: ROUTENZUGSTEUERUNG	149
8.1	Ausgangssituation und Zielsetzung	149
8.1.1	Problemstellung	149
8.1.2	Tourenplanung von Routenzügen	152
8.1.3	Agentensysteme in der Intralogistik.....	155
8.1.4	Zielsetzung	157

8.2	Entwicklung des Agentensystems	158
8.2.1	Entscheidungsmodell.....	159
8.2.2	Agentenmodell.....	164
8.2.3	Interaktionsmodell	167
8.3	Programmierung des Agentensystems.....	171
8.3.1	Umsetzung von Lageragent und Directory Facilitator.....	171
8.3.2	Umsetzung von Routenzugagenten.....	172
8.3.3	Umsetzung von Bedarfsortagenten.....	173
8.3.4	Systemarchitektur und detaillierte Kommunikationsprozesse.....	174
8.4	Simulationsexperimente.....	177
8.4.1	Design, Ausführung und Analyse der Simulationsexperimente	177
8.4.2	Simulationsergebnisse und kritische Beurteilung	182
9	AOSE-METHODE FÜR INTERDEPENDENTE AGENTENSYSTEME	185
9.1	Anforderungen und Zielsetzung der Integration	185
9.2	Herausforderungen beim Design interdependenter Systeme	187
9.2.1	Schneeballeffekte in Agentensystemen	188
9.2.2	Integration von Zielstrukturen	189
9.3	Erweiterung der DACS-Methode.....	190
9.3.1	Phase 1: Konzeption individueller Agentensysteme.....	191
9.3.2	Phase 2: Integration von Agentensystemen.....	192
9.3.3	Zusammenführung der Phasen zu einer erweiterten AOSE-Methode.....	194
10	INTEGRATION DER ANWENDUNGSFÄLLE	197
10.1	Vorbetrachtungen.....	197
10.2	Entwicklung des integrierten Agentensystems.....	199
10.2.1	Übergreifendes Entscheidungsmodell	200
10.2.2	Übergreifendes Agentenmodell	203
10.2.3	Übergreifendes Interaktionsmodell	205
10.3	Ergebnisse und kritische Beurteilung.....	209
10.3.1	Ergebnisse	209
10.3.2	Kritische Beurteilung.....	210

11	ZUSAMMENFASSUNG UND AUSBLICK.....	213
11.1	Beantwortung der Forschungsfragen.....	213
11.2	Ausblick.....	217
	LITERATURVERZEICHNIS	219
	ANHANG.....	251

Abbildungsverzeichnis

Abbildung 1-1: RAMI 4.0 (links) und Industrie 4.0-Komponente (rechts) i.A.a. [ZVE15]	2
Abbildung 1-2: Aufbau der Arbeit	9
Abbildung 2-1: Kundenauftrags- & Fertigungsprozess Automobilbau i.A.a. [Kro09, Mei09].....	12
Abbildung 2-2: Organisationstypen der Produktion i.A.a. [GüT12].....	14
Abbildung 2-3: Globale Automobilproduktion von 1900 bis 2017 i.A.a. [Bus07, OIC18].....	15
Abbildung 2-4: Kausalkette der 3M und Einfluss der PPS	16
Abbildung 2-5: Aufgabensicht des Aachener PPS-Modells [ScS12].....	18
Abbildung 2-6: Hierarchie der PPS und Anknüpfung an den Kundenauftragsprozess.....	20
Abbildung 2-7: Ziele der Produktionslogistik i.A.a. [Wie14].....	22
Abbildung 2-8: Modell zur Fertigungssteuerung i.A.a. [Löd08]	23
Abbildung 2-9: Automatisierungspyramide i.A.a. [Hei15, HLG15, Sie16].....	24
Abbildung 2-10: Entwicklung von komplizierten zu komplexen Systemen i.A.a. [GBS17].....	26
Abbildung 2-11: Abgrenzung Flexibilität und Wandlungsfähigkeit i.A.a. [ZMV05, VDI17]	32
Abbildung 3-1: Zentrale Steuerungsarchitektur i.A.a. [Unl15b]	39
Abbildung 3-2: (Modifizierte) Hierarchische Steuerungsarchitektur i.A.a. [Unl15b]	40
Abbildung 3-3: Heterarchische Steuerungsarchitektur i.A.a. [Unl15b]	43
Abbildung 3-4: Holonische Steuerungsarchitektur i.A.a. [Unl15b].....	44
Abbildung 3-5: Basis-FB (links) und zusammengesetzter FB (rechts) i.A.a. [DIN14b].....	48
Abbildung 3-6: IEC 61499 Anwendungs- und Gerätemodell i.A.a. [For15]	49
Abbildung 3-7: Grundlegender Aufbau Serviceorientierter Architekturen i.A.a. [Mab08].....	52
Abbildung 3-8: Einfache SOA-Diensthierarchie i.A.a. [FeL10].....	53
Abbildung 3-9: Vier Grundtypen von Agenten i.A.a. [GiH07, RuN10]	58
Abbildung 3-10: Layout von Production 2000+ i.A.a. [Bus12].....	59
Abbildung 4-1: Entwurfsmuster für die Ressourcenallokation i.A.a. [LCZ17]	72
Abbildung 4-2: Entwurfsmuster für den Ressourcenzugriff i.A.a. [LCZ17]	73
Abbildung 4-3: Gaia-Modelle und ihre Beziehungen im Entwurfsprozess i.A.a. [ZRW03].....	74
Abbildung 4-4: Phasen der MaSE-Methode i.A.a. [DWS01]	76
Abbildung 4-5: Phasen des Tropos-Modells i.A.a. [BGG02]	78

Abbildung 4-6: Phasen der Prometheus-Methode und ihre Beziehungen i.A.a. [PaW02]	79
Abbildung 4-7: Phasen der DACS-Methode i.A.a. [LVG14]	80
Abbildung 4-8: Phasen der ADMARMS-Methode i.A.a. [FaR15].....	82
Abbildung 6-1: Typische Steuerungsaufgaben in der Automobilproduktion i.A.a. [ZaL17]	97
Abbildung 6-2: Einordnung der Anwendungsfälle in den Gesamtprozess i.A.a. [ZaL17].....	100
Abbildung 6-3: Ablauf von Simulationsstudien i.A.a. [Law08]	105
Abbildung 6-4: Design und Analyse von Simulationsexperimenten i.A.a. [LHM12].....	106
Abbildung 7-1: Anwendungsfall 1 – Hallenlayout Karosseriebau	108
Abbildung 7-2: Häufigkeitsverteilung der Durchlaufzeit von Aufträgen im Karosseriebau	109
Abbildung 7-3: Umlaufbestand der drei Produktvarianten A, B und C.....	110
Abbildung 7-4: Ursachen einer instabilen Auftragsfolge i.A.a. [GMG07]	111
Abbildung 7-5: Materialfluss im Karosseriebau	118
Abbildung 7-6: Prozessschritte zur Karosseriefolgsteuerung	118
Abbildung 7-7: Auftragsfreigabe Karosseriebau im Status quo i.A.a. [Bec16].....	119
Abbildung 7-8: Steuerungsentscheidungen Karosseriefolgsteuerung	120
Abbildung 7-9: Entscheidungsmodell Karosseriefolge	122
Abbildung 7-10: Entscheidungscluster und Agenten Karosseriefolgsteuerung	124
Abbildung 7-11: Kommunikationsdiagramm für die Ressourcenallokation.....	130
Abbildung 7-12: Pull-Prinzip zur Initialisierung des Auftragsfreigabeprozesses	132
Abbildung 7-13: Beispielhafte Anwendung des Adaptionalgorithmus.....	134
Abbildung 7-14: Störungsfälle und Umgangsstrategien	135
Abbildung 7-15: Agentenkommunikation zur Ressourcenallokation in der KFS	137
Abbildung 7-16: Reservierungsmechanismus.....	138
Abbildung 7-17: Engpassvermeidung durch Berücksichtigung der Auftragsreihenfolge.....	138
Abbildung 7-18: Sequenzdiagramm Initiierung und Überwachung der Produktion.....	140
Abbildung 7-19: Zufallsfunktion für TBF als Exponentialverteilung	145
Abbildung 7-20: Zufallsfunktion für TTR als Erlang-Verteilung mit $n = 2$	145
Abbildung 8-1: Anwendungsfall 2 – Layout Routenzugsystem.....	150
Abbildung 8-2: Prozessschritte Materialbelieferung mit Routenzügen	159

Abbildung 8-3: Tourenplanung und Transportsteuerung i.A.a [Hab16].....	160
Abbildung 8-4: Entscheidungen für die Materialbelieferung mit Routenzügen	162
Abbildung 8-5: Entscheidungsmodell Routenzugsteuerung	163
Abbildung 8-6: Entscheidungscluster und Agenten Routenzugsteuerung	165
Abbildung 8-7: Kommunikationsdiagramm der Routenzugsteuerung	171
Abbildung 8-8: Zustandsdiagramm RZA (links) und BOA (rechts) i.A.a [Hab16]	174
Abbildung 8-9: Ein- und Ausgangsdaten im Simulationsmodell.....	175
Abbildung 8-10: Agentenkommunikation zur Materialbelieferung in der RZS i.A.a. [Hab16]....	176
Abbildung 8-11: Effektanalyse der Zielfunktionsfaktoren i.A.a. [Hab16].....	181
Abbildung 8-12: Gegenüberstellung Fahrzeit/Auftrag und Aufträge/Tour	182
Abbildung 8-13: Gegenüberstellung Terminabweichung	182
Abbildung 9-1: Grundsätzliche Arten von Zielstrukturen.....	189
Abbildung 9-2: Individuelle Agentensysteme als Ergebnisse der Phase 1.....	191
Abbildung 9-3: Erweiterte AOSE-Methode für interdependente Agentensysteme	196
Abbildung 10-1: Möglicher Informationsaustausch zwischen den Agentensystemen	198
Abbildung 10-2: Entscheidungsmodelle der individuellen Agentensysteme.....	200
Abbildung 10-3: Übergreifendes Entscheidungsmodell des integrierten Agentensystems.....	202
Abbildung 10-4: Übergreifendes Agentenmodell des integrierten Agentensystems.....	204
Abbildung 10-5: Übergreifendes Interaktionsmodell des integrierten Agentensystems.....	208
Abbildung 10-6: Ergebnisse der Integration der Anwendungsfälle	210

Tabellenverzeichnis

Tabelle 2-1: Anforderungen an Veränderbarkeit moderner Produktionssysteme	35
Tabelle 4-1: Evaluation von AOSE-Methoden	84
Tabelle 5-1: Generische Fertigungssystemstruktur und Praxisbeispiele i.A.a. [ZHR16]	88
Tabelle 5-2: Steuerungsaufgaben und -informationen in der GFSS i.A.a. [ZHR16].....	91
Tabelle 6-1: Auswahlkriterien für Agentenplattformen i.A.a. [VDI12].....	101
Tabelle 7-1: Variantenflexibilität der Ressourcen.....	109
Tabelle 7-2: Steuerungsentscheidungen Karosseriefolgesteuerung	120
Tabelle 7-3: Abhängigkeiten von Entscheidungen der Karosseriefolgesteuerung.....	121
Tabelle 7-4: Agentenmodell mit zugehörigen Steuerungsaufgaben und -informationen.....	125
Tabelle 7-5: Klassifikation der Abhängigkeiten in der Karosseriefolgesteuerung	128
Tabelle 7-6: Eigenschaften der Fertigungslinien 1 und 2	133
Tabelle 7-7: Kennzahlen für die Evaluation der Leistung des Agentensystems	143
Tabelle 7-8: Leistungsvergleich beider Produktionssteuerungssysteme	146
Tabelle 8-1: Anwendungsfall 2 – Kennzahlen im Status quo i.A.a. [Hab16].....	151
Tabelle 8-2: Steuerungsentscheidungen Routenzugsteuerung i.A.a. [Hab16]	161
Tabelle 8-3: Abhängigkeiten von Entscheidungen der Routenzugsteuerung i.A.a. [Hab16]....	164
Tabelle 8-4: Agentenmodell mit zugehörigen Steuerungsaufgaben und -informationen.....	166
Tabelle 8-5: Klassifikation der Abhängigkeiten in der Karosseriefolgesteuerung	169
Tabelle 8-6: Kennzahlen für die Evaluation der Leistung des Agentensystems	179
Tabelle 8-7: Klassifikation der Modellvariablen.....	180
Tabelle 8-8: Gegenüberstellung Agentensystem und Status quo	183
Tabelle 10-1: Zusätzliche Koordinationsentscheidung im integrierten Agentensystem	201
Tabelle 10-2: Abhängigkeiten von Entscheidungen zwischen den Systemen.....	201
Tabelle 10-3: Steuerungsaufgaben und -informationen im übergreifenden Agentenmodell	205
Tabelle 10-4: Klassifikation der Koordinationsabhängigkeiten im integrierten System	206

Abkürzungen

ACL	Agent Communication Language
ADACOR	Adaptive Holonic Control Architecture for Distributed Manufacturing Systems
ADMARMS	Axiomatic Design of a Multi-Agent Reconfigurable Mechatronic System
AE	Arbeitseinheit
AOP	Agentenorientierte Programmierung
AOSE	Agentenorientierte Softwareentwicklung
API	Application Programming Interface
AS	Arbeitsstation
BDE	Betriebsdatenerfassung
BDI	Belief Desire Intention
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CAP	Computer Aided Planning
CAQ	Computer Aided Quality Assurance
CASE	Computer Aided Software Engineering
CDA	Continuous Double Auction
CFP	Call for Proposition
CIM	Computer-Integrated Manufacturing
CNC	Computer Numerical Control
CNP	Contract Net Protocol
CPS	Cyber-physisches System
CPPS	Cyber-physisches Produktionssystem
CSV	Comma Separated Values
DACS	Designing Agent-based Control Systems
DIN	Deutsches Institut für Normung
DPWS	Devices Profile for Web Services
ERP	Enterprise Resource Planning
ESB	Enterprise Service Bus
FA	Fertigungsabschnitt
FB	Funktionsbaustein
FCFS	First Come First Serve
FFS	Flexibles Fertigungssystem
FG	Funktionsgruppe
FIFO	First in First out
FIPA	Foundation of Intelligent Physical Agents
FL	Fertigungslinie

FTF	Fahrerlose Transport-Fahrzeuge
FTS	Fahrerloses Transportsystem
GFSS	Generische Fertigungssystemstruktur
GRACE	Process and Quality Control using Multiagent Technology
HDT	Hand-Daten-Terminal
HMI	Human-Machine-Interface
HMS	Holonic Manufacturing System
HTTP	Hypertext Transfer Protocol
IEC	International Electrotechnical Commission
I4.0	Industrie 4.0
i.A.a.	in Anlehnung an
IIoT	Industrial Internet of Things
IoT	Internet of Things
ISA	International Society of Automation
IT	Informationstechnik
JADE	Java Agent Development Framework
JIS	Just in Sequence
JIT	Just in Time
JVM	Java Virtual Machine
KAP	Kundenauftragsprozess
KE	Konstruktionselement
KFS	Karosseriefolgesteuerung
KI	Künstliche Intelligenz
MAS	Mult-Agent System
MaSE	Multiagent Systems Engineering
MDE	Maschinendatenerfassung
MES	Manufacturing Execution System
MTBF	Mean Time Between Failures
MTTR	Mean Time To Repair
OLE	Object Linking and Embedding
OOP	Objektorientierte Programmierung
OPC UA	OLE for Process Control Unified Architecture
PABADIS	Plant Automation based on Distributed Systems
PABADIS'	PABADIS based Product Oriented Manufacturing Systems for Re-Configurable En-
PROMISE	terprises
PK	Prozesskomponente
PLC	Programmable Logic Control (dt. <i>SPS</i>)
PPS	Produktionsplanung und -steuerung

PROSA	Product-Resource-Order-Staff-Architecture
QR-Code	Quick-Response-Code
QS	Qualitätssicherung
RAMI 4.0	Referenzarchitektur Modell Industrie 4.0
RAP	Requesting Action Protocol
RZS	Routenzugsteuerung
SCADA	Supervisory Control And Data Acquisition
SK	Steuerungskomponente
SOA	Serviceorientierte Architektur
SOAP	Simple Object Access Protocol
SPS	Speicherprogrammierbare Steuerung (engl. <i>PLC</i>)
SQL	Structured Query Language
TBF	Time Before Failure
TPM	Total Productive Maintenance
TPS	Toyota Production System
TTR	Time To Repair
UML	Unified Modelling Language
VKI	Verteilte Künstliche Intelligenz
VRP	Vehicle Routing Problem
WIP	Work in Progress (dt. <i>Umlaufbestand</i>)
WS	Webservice
WZ	Wunschzeit

PROSA:

OH	Order Holon
PH	Product Holon
RH	Resource Holon
SH	Staff Holon

PABADIS' PROMISE:

AB	Ability Broker
IC	Information Collector
OA	Order Agent
OAS	Order Agent Supervisor
PDR	Product Data Repository
RA	Resource Agent
RAS	Resource Agent Supervisor

ADACOR:

OH	Operational Holon
PH	Product Holon
SH	Supervisor Holon
TH	Task Holon

GRACE:

IMA	Independent Meta Agent
PA	Product Agent
PTA	Product Type Agent
QCA	Quality Control Agent
RA	Resource Agent

Anwendungsfall 1: Karosseriefolgesteuerung

AA	Auftragsagent
DF	Directory Facilitator
EEOP	Estimated End of Production
EOP	End of Production
ESOP	Estimated Start of Production
MA	Mediatoragent
PA	Produktagent
RA	Ressourcenagent
SA	Scheduling-Agent
SMA	Shop-Management-Agent
SOP	Start of Production

Anwendungsfall 2: Routenzugsteuerung

BOA	Bedarfsortagent
DF	Directory Facilitator
DF	Directory Facilitator
LA	Lageragent
MnA	Main-Agent
RZA	Routenzugagent
WZ	Wunschzeit

*“It is not the strongest or the most intelligent who will survive,
but those who can best manage change.”*

– Charles Darwin

1 Einleitung

1.1 Ausgangssituation und Problemstellung

Fortschritte in Technologie und Wissenschaft ermöglichen es unserer Gesellschaft, die Lebensbedingungen kontinuierlich und mit zunehmender Geschwindigkeit zu verbessern sowie grundlegende zivilisatorische Herausforderungen zu bewältigen. Mit den neuen Möglichkeiten und Chancen steigen gleichzeitig auch die Anforderungen an neue Ansätze und Lösungswege, die den veränderten Rahmenbedingungen standhalten müssen.

Auf den weltweiten Märkten sind diese Veränderungen insbesondere im Hinblick auf Wettbewerb, Technologie und Kunden deutlich zu erkennen. Unternehmen sehen sich in nahezu allen Branchen einem globalisierten und in sich verstärkten Wettbewerb, sowohl mit bekannten als auch mit neuen Konkurrenten, ausgesetzt. Die Entwicklung von Technologiemarkten vollzieht sich insbesondere im Bereich der digitalen Datenverarbeitung schneller als in nahezu allen anderen Bereichen und bietet neue Möglichkeiten zur Steuerung von Produktionsprozessen und zur Optimierung des Ressourceneinsatzes [Wag18]. Die Nutzung dieser Möglichkeiten ist dringend erforderlich, da steigende Kundenanforderungen und immer kürzere Produktlebenszyklen den Wettbewerbsdruck zusätzlich erhöhen [BoG08]. Der zunehmende Wunsch der Kunden nach individualisierten Produkten lässt die Anzahl von Produktderivaten und -varianten rasant steigen. Trotz wachsender Produktionszahlen führt dies zu einer Reduzierung der Stückzahlen je Produktvariante und damit letztlich zu einem Kostenanstieg [Jan12]. Durch zunehmende Volatilität der Absatzmärkte sowie komplexere und damit weniger transparente Lieferketten wird die Erstellung zuverlässiger Nachfrageprognosen weiter erschwert [Gün10]. Kurzfristige Anpassungen bestehender Produktionspläne werden damit zum Alltag und verursachen hohen Koordinations- und Steuerungsaufwand [DOS13].

Als Resultat erlebt die Fertigungsindustrie einen fundamentalen Wandel von einem Verkäufer- hin zu einem Käufermarkt. Der Erfolg von Unternehmen ist folglich zunehmend abhängig von der Flexibilität, Agilität und Autonomie der zugrundeliegenden Produktionssysteme [EIM09].

Die gestiegenen Anforderungen zwingen die Beteiligten neue Wege zu gehen und die Anstrengungen zu erhöhen. Historisch führte dies nicht selten zu gravierenden Umbrüchen und zu neuen Entwicklungsstufen der Menschheit. Aus industrieller Sicht haben in den letzten 250 Jahren drei solcher Umbrüche stattgefunden und wir befinden uns nun an der Schwelle zum vierten. Dieser

Umbruch, plakativ die vierte industrielle Revolution oder kurz Industrie 4.0 genannt, soll Produktionssysteme intelligenter machen. Die heute schon oft vollautomatische Ausführung von Abläufen soll um die Fähigkeit erweitert werden autonom zu handeln und situationsabhängig Entscheidungen zu treffen [BIT14]. Unter dem Begriff Industrie 4.0 werden vier Kernaspekte zusammengefasst: 1) horizontale Integration von Unternehmen über Wertschöpfungsnetzwerke hinweg, 2) vertikale Integration von Produktionsschritten und deren Steuerung in diesen Netzwerken, 3) digitale Durchgängigkeit des Engineerings über den kompletten Produktlebenszyklus hinweg sowie 4) verbesserte Interaktion mit dem Menschen [KWH13].

Erste Ergebnisse auf dem Weg zur Realisierung dieser Ziele wurden bereits mit der Definition des Referenzarchitekturmodells Industrie 4.0 (RAMI 4.0), der virtuellen Repräsentation sowie der dazugehörigen Verwaltungsschale erreicht [ZVE15] (vgl. Abbildung 1-1) und in der DIN SPEC 91345 spezifiziert [DIN16]. Eine Konkretisierung wurde durch die Beschreibung der Struktur der Verwaltungsschale ebenfalls bereits erzielt [Pla16a]. Ein wichtiges Erfolgskriterium für deren flächendeckende Umsetzung ist die Verankerung weiterer Standards und Normen über Unternehmens- und Landesgrenzen hinweg [BVZ15]. Daher findet eine Koordination der Standardisierungsbestrebungen sowohl auf nationaler als auch auf internationaler Ebene statt [Pla16b].

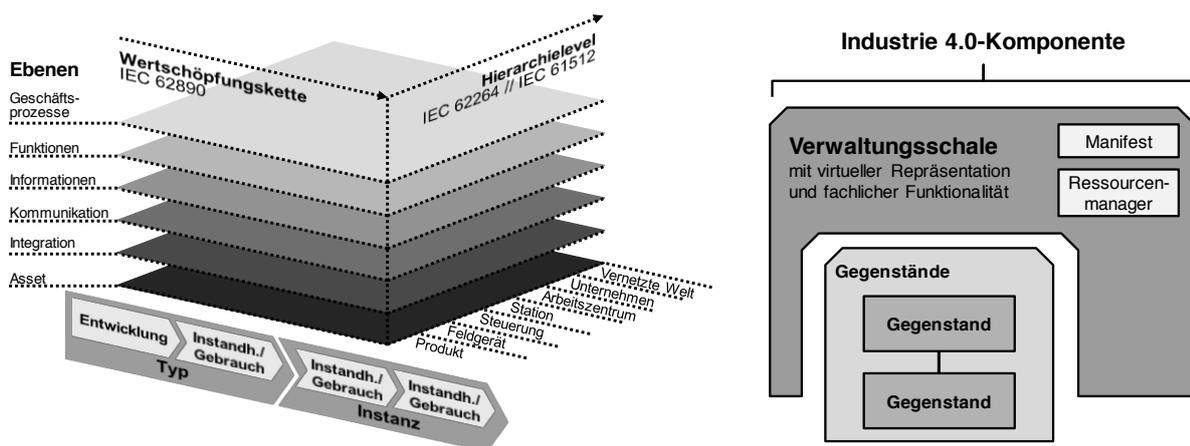


Abbildung 1-1: RAMI 4.0 (links) und Industrie 4.0-Komponente (rechts) i.A.a. [ZVE15]

Dies sind jedoch nur erste Schritte um die gestiegenen Anforderungen an Flexibilität und Veränderbarkeit von Produktionssystemen zu erfüllen. Für Hochlohnstandorte wie Deutschland ist die Positionierung als Leitanbieter und Leitmarkt für Industrie 4.0-Lösungen dabei von besonderer Bedeutung um die eigene Wettbewerbsfähigkeit zu stärken [BVZ15]. Produktionssysteme in Industrieländern stehen daher vor einem mehr oder weniger radikalem Umbau, bei dem vor allem derzeit vorangetriebene Modularisierungskonzepte auf Basis cyber-physischer Systeme in die Realität umgesetzt werden sollen [VBH17a].

Eine wesentliche Herausforderung stellt dabei die große Menge an teilweise interdependenten Steuerungsentscheidungen dar, durch die moderne Produktionssysteme charakterisiert sind. Deren Abhängigkeiten werden in der Entwicklungsphase – und oft auch darüber hinaus – nicht ausreichend berücksichtigt. Die Steuerungsentscheidungen müssen jedoch in einer verteilten, aber

konsistenten und aufeinander abgestimmten Form getroffen werden, um ein erwünschtes Systemverhalten zu erreichen [TeT09, KWH13].

In vielen Fällen werden derzeit eingesetzte Steuerungssysteme jedoch stark isoliert voneinander entwickelt und anschließend über Schnittstellen mit partizipierenden oder übergeordneten Systemen verbunden. Einzelne betrachtet stellt jedes System eine eigene, in sich vollständige Einheit dar. Gleichzeitig sind solche Systeme jedoch in vielen Fällen auch Teil eines übergeordneten Systems (z.B. Staplerleitsystem oder Depalettersystem) [Weg04, SaP11], die als sogenannte Holons¹ bezeichnet werden können [Sch13] (vgl. Kapitel 3.2).

Die Leistung des Gesamtsystems steht damit in direktem Zusammenhang mit der Leistung der darin enthaltenen isolierten Subsysteme, die sich gegenseitig beeinflussen und konfliktäre Ziel-funktionen haben können. Bei Entwicklungs- und Optimierungsaktivitäten sowie im eigentlichen Betrieb stehen jedoch meist diese isolierten Subsysteme im Fokus. Sie haben eigene Ziele, die separat gemessen werden. Folglich können im besten Fall lokale Optima erreicht werden, die nicht zwangsläufig mit dem globalen Optimum des Gesamtsystems identisch sind.

Die Optimierung von Subsystemen zu Lasten eines übergeordneten Systems widerspricht dem Prinzip der Gewinnmaximierung, ist jedoch aufgrund der Komplexität moderner Produktionssysteme Normalität in vielen Unternehmen. Aus wirtschaftlicher Perspektive ist es sinnvoller, die Ziele eines Subsystems gegenüber den Zielen des übergeordneten Systems niedriger zu gewichten und Subsysteme so aufeinander abzustimmen, dass die von Stakeholdern definierte Zielkonfiguration bestmöglich erreicht wird. Durch mangelnde Transparenz der gegenseitigen Abhängigkeiten, sich addierende und neutralisierende Effekte sowie nicht darauf ausgelegte Entwicklungsmethoden wird die Optimierung in Richtung des globalen Optimums jedoch oft vernachlässigt. Erweitert man den Betrachtungsbereich über lokale Produktionssysteme hinaus auf gesamte Produktionsnetzwerke, steigt die Komplexität überproportional und die Transparenz sinkt weiter.

Zusammenfassend ergeben sich für die Fertigungsindustrie zwei Herausforderungen. Erstens müssen Produktionssysteme zunehmend flexibler, agiler und autonomer werden, um mit dem schwankenden Bedarf, den immer individuelleren Kundenanforderungen und den schärferen Wettbewerbsbedingungen mithalten zu können. Zweitens bedarf es neuer Lösungen, um die zunehmend steigende Komplexität dieser Produktionssysteme zu beherrschen und ihre Bestandteile in den Phasen der Entwicklung, des Betriebs und der Optimierung so zu koordinieren, dass das Gesamtsystem nahe dem globalen Optimum arbeitet.

¹ Zusammensetzung des Wortes „Holos“ (griech. ganz, Gesamtheit) und dem Suffix „on“ (griech. Partikel, Teil).

1.2 Motivation und Zielsetzung

Die aufgezeigten Herausforderungen sind sowohl in der Forschung als auch in der Industrie aufgegriffen worden und erste Ergebnisse wurden bereits erzielt (vgl. RAMI 4.0 etc.). Die zugrundeliegenden Umsetzungstechnologien stützen sich dabei auf sehr unterschiedliche Ansätze.

Ein aus der klassischen Steuerungs- und Automatisierungstechnik stammender Ansatz ist der Standard zur verteilten Automatisierung nach IEC 61499. Hier kommen sogenannte Funktionsbausteine (FB) zum Einsatz, die einzelne Funktionen wiederverwendbar kapseln und sie Entwicklern über Bibliotheken zur Verfügung stellen [ZoL14].

Ein geschäftsprozessorientierter Ansatz ist das aus der Informatik stammende Architekturmuster der serviceorientierten Architekturen (SOA). Zielsetzung von SOA ist es, starre, monolithische Unternehmensapplikationen durch Kapselung, lose Kopplung und Verteilung von Services zu ersetzen. Diese Services sollen durch andere Applikationen jederzeit aufrufbar sein, sodass eine maximale Wiederverwendbarkeit gewährleistet ist [Hom10].

Microservices stellen eine Weiterentwicklung dieser Architektur dar. Sie unterscheiden sich vor allem dadurch, dass Dienste unabhängig voneinander implementiert und betrieben werden können. Dies ist möglich, da sie im Gegensatz zur SOA nicht auf einem gemeinsamen Enterprise Service Bus (ESB) ablaufen und nicht auf dieselben Datenbanken zugreifen müssen [New15].

Ein aus dem Gebiet der Verteilten Künstlichen Intelligenz stammender Ansatz ist die Agentenorientierte Softwareentwicklung (AOSE). Sie fußt auf der Dekomposition von komplexen Problemen in einzelne autonome Abläufe und deren Abbildung in Softwareagenten. Agenten sind Softwarekomponenten, die über einheitliche Schnittstellen miteinander kommunizieren. Sie können unabhängig voneinander entwickelt und leicht zur Laufzeit getauscht werden [FeL10]. Interagieren mehrere Agenten miteinander um gemeinsam Aufgaben zu erfüllen, spricht man im Englischen von Multi-Agent Systems (MAS) oder zu Deutsch von Agentensystemen [VDI10].

Im Gegensatz zu gängigen Steuerungsarchitekturen verfolgt Industrie 4.0 den Leitgedanken eines hierarchielosen Steuerungsprinzips, in dem autonome Einheiten miteinander kooperieren. Agententechnologie ist dafür entwickelt worden ebensolche dezentralen Steuerungsarchitekturen zu realisieren [LCL10]. Zu den Stärken von Agentensystemen gehören deren Modularität, Anpassungsfähigkeit, Flexibilität, Robustheit, Wiederverwendbarkeit und Skalierbarkeit. Diese Stärken können insbesondere in Domänen ausgespielt werden, in denen dezentralisierte Entscheidungsfindung und Steuerung notwendig sind. Das sind allem voran Produktionsplanung und -steuerung, Logistik, Steuerung von Lieferketten und Verkehr, Smart Grids im Energiesektor, Gebäude- und Heimautomatisierung, militärische Verteidigung, humanitäre Hilfe, Netzwerksicherheit sowie unbemannte Luftfahrt [Unl15b].

Durch ihre inhärenten Eigenschaften gehören Agentensysteme zu den vielversprechendsten Ansätzen zur Realisierung von flexiblen Produktionssystemen und sie sind darüber hinaus in vielen

Anwendungsbereichen applizierbar [BoG08, MZK12, Göh13]. Ihr Potenzial wurde in zahlreichen Forschungsprojekten nachgewiesen und ist gut dokumentiert, z.B. im Rahmen der Konferenzserien HoloMAS und MATES. In entsprechenden VDI-Richtlinien wird darüber hinaus festgehalten, dass eine Vielzahl von typischen Herausforderungen im Produktionsumfeld isoliert sehr gut mit Agentensystemen gelöst werden kann [VDI12].

Die zahlreichen Stärken von Agentensystemen, ihre erwiesenen Potenziale und ihre universelle Anwendbarkeit für weit verbreitete Steuerungsprobleme geben ihnen aus Sicht des Autors einen Vorsprung gegenüber anderen Ansätzen. Eine ausführliche Abhandlung der Vor- und Nachteile der unterschiedlichen Ansätze in Kapitel 3 stützt diese These. Folglich ist die agentenorientierte Softwareentwicklung die maßgebliche Technologie und der Kern der Betrachtungen in dieser Arbeit.

In der industriellen Praxis sind Produktionssysteme in den meisten Fällen keine abgeschlossenen, in sich konsistenten Einheiten, sondern ein komplexes Netzwerk aus miteinander vernetzten und verschachtelten Subsystemen. Die Lösung der individuellen Probleme auf Ebene der Subsysteme mithilfe von agentenbasierten Steuerungsarchitekturen reicht daher nicht aus, denn die resultierenden Steuerungsarchitekturen beziehen sich ausschließlich auf eine begrenzte Menge an Steuerungsentscheidungen innerhalb der Subsysteme [TeT09, KWH13]. Analog dazu findet das Entwickeln, Betreiben und Optimieren oft unter dem Aspekt der expliziten Zielfunktionen eines spezifischen Subsystems statt und kann daher zwangsweise nur zu Lösungen führen, die sich höchstens im Bereich des lokalen Optimums dieses Subsystems befinden. Es existieren zwar Ansätze zur analytischen Leistungsbewertung von Subsystemen und ihrer Steuerung, die allerdings bei einer hohen Systemkomplexität oftmals überfordert sind und in Synchronisationsproblemen resultieren [Gut02].

Damit wird der Interdependenz der Subsysteme untereinander jedoch nicht ausreichend Rechnung getragen und das Erreichen eines globalen Optimums des Gesamtsystems gerät außer Reichweite. Dies ist insbesondere zutreffend, wenn die Steuerungs- und Optimierungsziele der Subsysteme im Konflikt zueinander stehen und keine datenbasierte Mediation der Ziele stattfindet. Folglich bleiben isoliert entwickelte und implementierte Steuerungsarchitekturen hinter den Möglichkeiten eines integrierten Ansatzes zurück und lassen wirtschaftliche Potenziale für Unternehmen ungenutzt.

Das Ziel von Industrie 4.0 ist es diese Potenziale zu heben. Es ist bisher jedoch unklar wie eine Menge von autonomen Subsystemen so miteinander vernetzt und integriert werden kann [MVK06], dass ein interagierendes „System of Systems“ entsteht, in dem die gegenseitigen Interdependenzen ausreichend Berücksichtigung finden [SaP11]. Eine Entwicklungsmethode, die dieses Problem angeht, muss dabei nicht nur eine geeignete Steuerungsarchitektur für die isolierte Entwicklung von Subsystemen zur Verfügung stellen. Sie muss auch sicherstellen, dass die Kommunikation der Subsysteme untereinander möglich ist, eine systemweite Datendurchgängigkeit

besteht und die Mediation bzw. situationsbedingte Priorisierung der Zielfunktionen unterstützt wird. Auf Grundlage dieser Annahmen wird das primäre Ziel dieser Arbeit wie folgt definiert:

Zielsetzung:

Das Ziel dieser Arbeit ist die Erarbeitung einer Entwicklungsmethode, die eine schrittweise Entwicklung und Integration agentenbasierter Steuerungssysteme ermöglicht und damit den Weg für umfassendere, vertikal und horizontal integrierte Steuerungsarchitekturen bereitet.

Um dieses Ziel zu erreichen, müssen im Rahmen von Vorarbeiten die notwendigen Grundsteine gelegt werden. Dafür wurden vier Forschungsfragen identifiziert, die im Rahmen dieser Arbeit sukzessiv beantwortet werden. Dabei wird auf den bisherigen Forschungsergebnissen wie dem RAMI 4.0, der Industrie 4.0-Komponente (kurz I4.0-Komponente) und der Verwaltungsschale aufgesetzt und es werden, wo nötig, neue Konzepte und Methoden ergänzt.

Bevor mit der Integration von Subsystemen begonnen werden kann, ist es erforderlich eine Steuerungsarchitektur zu definieren, welche die Subsysteme begründet. Die erste Forschungsfrage bezieht sich daher auf die Struktur von I4.0-Komponenten:

Forschungsfrage 1:

Was sind relevante Ebenen für Industrie 4.0-Komponenten von Fertigungssystemen und wie definieren sie sich?

Zur Gewährleistung der systemweiten Datenkonsistenz steht im nächsten Schritt die vollständige Erfassung und transparente Darstellung der in den jeweiligen über- und untergeordneten Steuerungssystemen enthaltenen Steuerungsentscheidungen und -informationen und deren strukturierte, überschneidungsfreie Darstellung im Fokus:

Forschungsfrage 2:

Welche Informationen müssen auf den verschiedenen Ebenen der virtuellen Repräsentation während der Inbetriebnahme- und Nutzungsphase zur bestmöglichen Ausführung komplexer, interdependenter Fertigungsprozesse verfügbar sein?²

Mit der Beantwortung dieser beiden Fragen wird die Grundlage geschaffen, um im Hauptteil der Arbeit zu untersuchen wie dezentrale Steuerungsansätze die Flexibilität, Agilität und Autonomie von Produktionssystemen positiv beeinflussen können:

² Für Betrachtungen der Entwicklungs- und End-of-Life-Phase siehe z.B. [HHL16] und [SLH16].

Forschungsfrage 3:

Wie können agentenbasierte Steuerungsarchitekturen die Leistungsfähigkeit komplexer Produktionssysteme verbessern, um den gegenwärtigen Herausforderungen hinsichtlich Flexibilität, Agilität und Autonomie gerecht zu werden?

Zur Beantwortung dieser Frage werden zwei Anwendungsfälle aus der Domäne der Automobilfertigung herangezogen und geeignete Agentensysteme konzipiert, entwickelt und anhand von Simulationen getestet. Die Erkenntnisse aus den Anwendungsfällen stellen die Grundlage zur Beantwortung der finalen Forschungsfrage dar, die die Entwicklung einer Methode zur Integration voneinander abhängiger Systeme zum Ziel hat. Insbesondere steht dabei die Integration von Systemen mit konfliktären Zielfunktionen im Vordergrund, die zur Erreichung eines globalen Optimums aufeinander abgestimmt werden müssen:

Forschungsfrage 4:

Wie können voneinander abhängige agentenbasierte Steuerungssysteme miteinander verbunden werden, um umfassendere, vertikal und horizontal integrierte Steuerungsarchitekturen und Datendurchgängigkeit zu realisieren?

Eine der Hauptherausforderungen von agentenbasierten Entwicklungsmethoden besteht in der Notwendigkeit etwaige Agentensysteme an die Anforderungen der jeweiligen Anwendungsdomäne anzupassen [FeL10]. Da die Betrachtung aller Domänen, für die der Einsatz von Agentensystemen in Frage kommt, den Rahmen dieser Arbeit deutlich überschreiten würde, bedarf es im vorliegenden Fall einer Konkretisierung des Betrachtungsumfangs. Diese sollte zwei Kriterien erfüllen um den Mehrwert der vorliegenden Arbeit zu erhöhen. Einerseits sollte das betrachtete Feld der Domäne einen hohen Einfluss auf die Produktivität von Unternehmen in der Fertigungsindustrie haben und andererseits sollte das Anwendungsgebiet anschaulich, repräsentativ und auf andere Branchen übertragbar sein.

Diesen Anforderungen wird durch folgende Fokussierung Rechnung getragen: Erstens wird der Schwerpunkt der Betrachtungen hinsichtlich der Unternehmensfunktion auf die Aufgaben der Produktionsplanung und -steuerung (PPS) sowie Logistik gelegt. Die Herausforderungen in diesen Bereichen betreffen große Netzwerke mit einer Vielzahl an Teilnehmer-Knotenpunkten und Informationen. Damit haben sie einen großen Einfluss auf den Unternehmenserfolg. Zweitens richtet sich der Fokus der Ausführungen hinsichtlich der betrachteten Branche auf Beispiele aus dem Bereich der Automobilproduktion. Automobilhersteller und deren Lieferanten versammeln unter sich einen großen Teil des Bruttoinlandsproduktes führender Industrienationen, weshalb der Automobilbau eine Leitindustrie für viele andere Fertigungsbranchen ist [Dru66]. Sie verfügt über einen hohen Komplexitätsgrad, der eine Integration von Subsystemen rechtfertigt, ist aber

gleichzeitig anschaulich und auf andere Branchen übertragbar. Um die Übertragbarkeit zu verdeutlichen, werden an ausgewählten Stellen Beispiele aus anderen Bereichen der Fertigungsindustrie herangezogen. Die in dieser Arbeit vorgestellten Prinzipien, Konzepte und Vorgehensweisen können in den meisten Fällen ohne wesentliche Anpassungen auf sie übertragen werden.

Zur Beantwortung der definierten Forschungsfragen in dem wie beschrieben abgegrenzten Bereich wurde die im folgenden Abschnitt beschriebene Struktur gewählt.

1.3 Aufbau der Arbeit

Das Hauptziel der vorliegenden Arbeit ist die im vorangehenden Abschnitt dargelegte Erarbeitung einer Entwicklungsmethode, die eine schrittweise Integration agentenbasierter Steuerungssysteme ermöglicht und den Weg für umfassendere, vertikal und horizontal integrierte Steuerungsarchitekturen bereitet. Mit diesem Ziel und den sich daraus ergebenden Forschungsfragen wurde die vorliegende Arbeit wie folgt strukturiert (vgl. Abbildung 1-2).

Zunächst wird in Kapitel 2 der Stand der Technik in der Automobilindustrie erörtert. Dabei werden Grundlagen der Automobilfertigung wie grundsätzliche Produktionsprozesse und Organisationsprinzipien erläutert sowie der Status quo in der Produktionsplanung und -steuerung im Allgemeinen beschrieben. Darauf aufbauend werden die Herausforderungen konkretisiert, die aus den Defiziten des derzeitigen Zustands resultieren.

Im darauffolgenden Kapitel 3 wird der Stand der Forschung von möglichen Lösungsansätzen für die beschriebenen Herausforderungen erfasst. Dabei wird auf die hinter Industrie 4.0 stehenden Steuerungsparadigmen und -architekturen eingegangen sowie Software-Design-Paradigmen vorgestellt, die zu ihrer Umsetzung verwendet werden können. Ferner werden auch konkret implementierte Beispiele aus Forschungs- und Industrieprojekten vorgestellt. Die Ansätze werden schließlich gegeneinander abgewogen und bewertet, was zur Auswahl von agentenbasierten Systemen als Leittechnologie für den weiteren Teil der Arbeit führt.

In Kapitel 4 werden die verschiedenen Möglichkeiten der architektonischen Gestaltung von Agentensystemen sowie ihre jeweiligen Stärken und Schwächen diskutiert. Auf dieser Grundlage werden Entwurfsmuster für wiederkehrende Steuerungsprobleme abgeleitet. Des Weiteren werden verfügbare Methoden für die Entwicklung agentenbasierter Steuerungssysteme beschrieben und bewertet. Die Ausführungen münden in der Auswahl einer konkreten Entwicklungsmethode für die Verwendung im weiteren Teil dieser Arbeit, womit die Basis für die Beantwortung der Forschungsfragen gelegt wird.

Zur Beantwortung der ersten beiden Forschungsfragen wird in Kapitel 5 eine generische Fertigungssystemstruktur für Betriebe der Fertigungsindustrie definiert und mit Beispielen aus unterschiedlichen Industrien und Branchen unterlegt. Damit entsteht ein Rahmenwerk, anhand dessen die Zuordnung von Steuerungsaufgaben und -informationen vorgenommen werden kann. Die

Systematik vereinfacht zudem die spätere Analyse der Entscheidungsaufgaben in den konkreten Anwendungsfällen (vgl. Kapitel 7, 8 und 10).

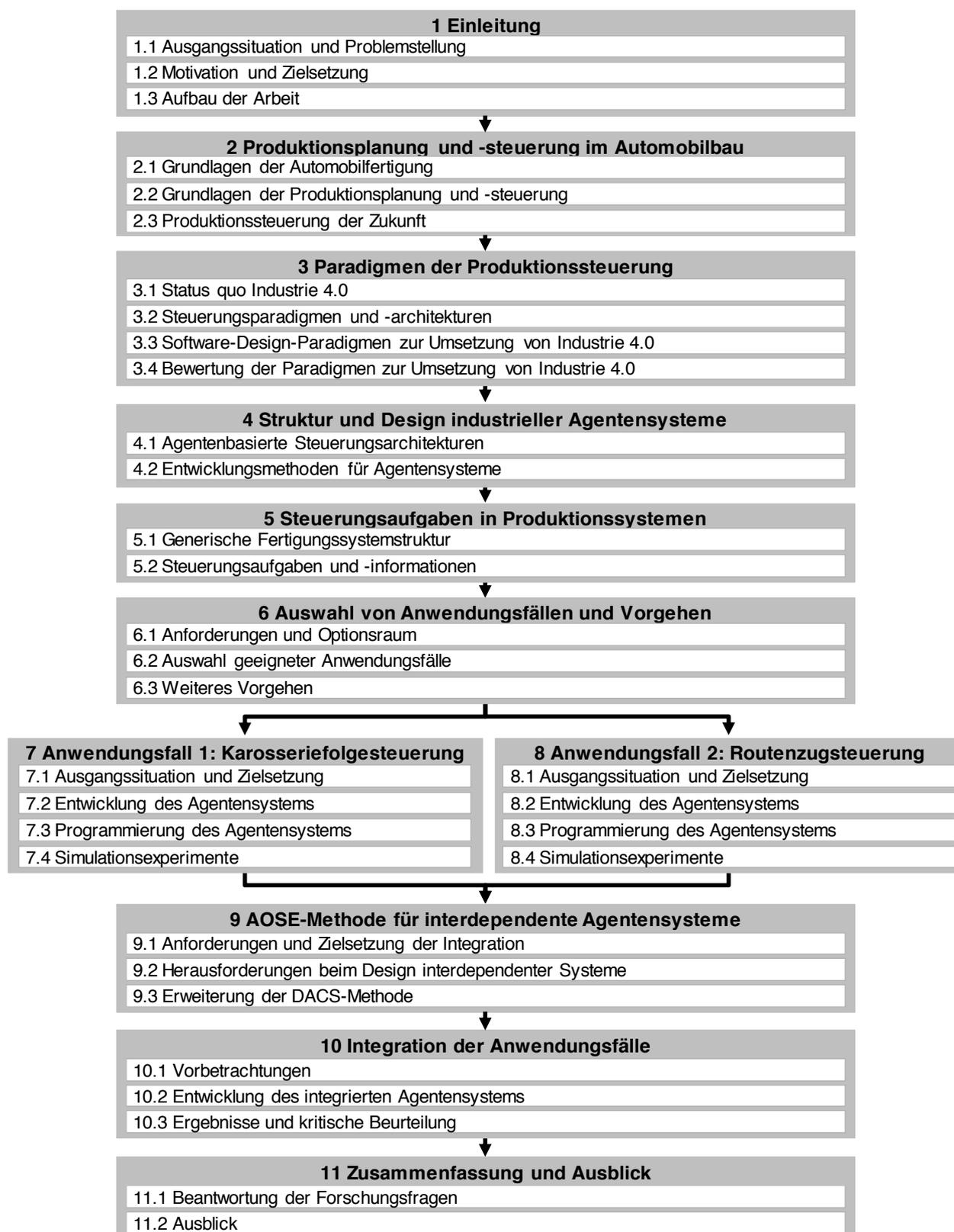


Abbildung 1-2: Aufbau der Arbeit

Als Grundgerüst für die Beantwortung der weiteren Forschungsfragen werden in Kapitel 6 verschiedene Steuerungsaufgaben in der Automobilfertigung analysiert und aus ihrer Gesamtheit

zwei interdependente Anwendungsfälle für die Verwendung im weiteren Teil der Arbeit ausgewählt. Zudem wird das Vorgehen für die Simulation und Parametrisierung der Anwendungsfälle festgelegt.

Die Anwendungsfälle werden in den Kapiteln 7 und 8 zunächst individuell betrachtet und später integriert. Die beiden Kapitel folgen daher der gleichen Struktur: Zunächst wird der Status quo und die damit zusammenhängenden Herausforderungen vorgestellt sowie eine Zielsetzung abgeleitet. Darauf aufbauend werden die dazugehörigen Agentensysteme nach der ausgewählten Entwicklungsmethode konzipiert und in Programmcode umgesetzt. Schließlich werden auf dieser Grundlage Simulationsexperimente entworfen und durchgeführt, mit denen die Leistungsfähigkeit der Agentensysteme gemessen und bewertet wird.

Auf den Erkenntnissen der Anwendungsfälle aufbauend erfolgt in Kapitel 9 die Entwicklung einer AOSE-Methode für die Integration voneinander abhängiger Agentensysteme. Dabei wird aufgezeigt, welche Limitationen die bisherigen Entwicklungsmethoden bei der Implementierung interdependenter Agentensysteme haben und es werden Schritte zu ihrer Erweiterung vorgestellt. Dafür werden zunächst die Anforderungen und die Zielsetzung diskutiert, aber auch Herausforderungen bei der Entwicklung interdependenter Systeme aufgezeigt. Unter Berücksichtigung dieser Vorgaben erfolgt auf Basis des zuvor ausgewählten Entwicklungsvorgehens die Erarbeitung einer umfangreichen ASOE-Methode, die eine schrittweise Implementierung von interdependenten Agentensystemen ermöglicht.

Im anschließenden Kapitel 10 wird die entwickelte AOSE-Methode auf die zuvor beschriebenen Anwendungsfälle angewendet und ein Konzept für ein integriertes Agentensystem entworfen. Durch diese Erprobung werden die Praxistauglichkeit der Methode aber auch ihre Limitationen aufgezeigt, die in weiteren Forschungsfragen münden.

Die Arbeit schließt in Kapitel 11 mit einer zusammenfassenden kritischen Betrachtung der Ergebnisse sowie einem Ausblick auf noch offene Fragen und zukünftige Forschungsgebiete.

2 Produktionsplanung und -steuerung im Automobilbau

Das Automobil ist eines der komplexesten in Massenfertigung hergestellten Produkte unserer Zeit [MaW04]. Ein durchschnittliches Fahrzeug besteht aus 20.000 bis 30.000 Einzelteilen, die zur richtigen Zeit am richtigen Ort und in der richtigen Qualität bereitgestellt werden müssen [KuT08, Toy18]. Bei der Fertigung müssen individuelle und kumulierte Qualitätsmerkmale wie z.B. Toleranzketten eingehalten sowie sicherheitsrelevante Bauteile, insbesondere in Karosserie und Antriebsstrang, dokumentiert und ihre Funktionalität sichergestellt werden [Boh98, Wit11]. Die Koordination der dafür notwendigen Teileströme ist komplex und stellt eine Herausforderung für Automobilhersteller aller Klassen dar [GBS17]. Um den Stand der Technik in diesem Bereich zu erfassen, wird im folgenden Abschnitt 2.1 zunächst aufgezeigt wie die Automobilindustrie diese Herausforderungen heute bewältigt. Darauf aufbauend wird in Abschnitt 2.2 diskutiert, welche Rolle die Produktionsplanung und -steuerung dabei einnimmt. Schließlich wird in Abschnitt 2.3 beleuchtet, welche Hürden dabei in Zukunft überwunden werden müssen.

2.1 Grundlagen der Automobilfertigung

Die zur Steuerung der Automobilfertigung benötigten Grundlagen werden unter drei Aspekten diskutiert: In Abschnitt 2.1.1 werden zunächst die grundsätzlichen Fertigungsprozesse und -technologien vorgestellt, die in der Fahrzeugherstellung zum Tragen kommen. Darauf aufbauend werden in Abschnitt 2.1.2 die arbeitsorganisatorischen Prinzipien und Methoden erläutert, welche die Kosten der Automobilherstellung dramatisch reduziert und Personenkraftwagen dadurch der breiten Öffentlichkeit zugänglich gemacht haben. Im darauffolgenden Abschnitt 2.1.3 werden die resultierenden Anforderungen an Produktionssysteme wie z.B. Flexibilität und Robustheit beschrieben sowie wesentliche Begriffe für die Verwendung in dieser Arbeit definiert und abgegrenzt.

2.1.1 Fertigungs- und Kundenauftragsprozess

In die Automobilproduktion fließen zahlreiche Fertigungstechnologien ein, die alle Verfahren der sechs Hauptgruppen aus DIN 8580 abdecken [DIN03]. Auch wenn Trends wie Leichtbau und additive Fertigungsverfahren in den letzten Jahren zunehmend an Bedeutung gewinnen [Lin16], sind die grundlegenden Prinzipien der verwendeten Fertigungsverfahren trotz der über 100-jährigen Geschichte des Automobils immer noch dieselben. Die wesentlichen Fertigungsschritte und der Bestelldurchlauf in der Automobilindustrie sind in Abbildung 2-1 dargestellt.

Der **Fertigungsprozess** setzt sich aus den vier Gewerken Presswerk, Karosseriebau, Lackiererei und Montage zusammen. Bevor mit dem Fügen der Karosserie begonnen werden kann, müssen im **Presswerk** Karosserieteile wie Unterboden, Seitenteile, Kotflügel, Türen und Klappen gefertigt werden. Dafür werden Technologien wie Stanzen, Tiefziehen und Warmumformung angewendet. Die Belieferung mit Rohmaterial, vor allem Stahl- und Aluminium-Coils, erfolgt meist über externe Lieferanten.

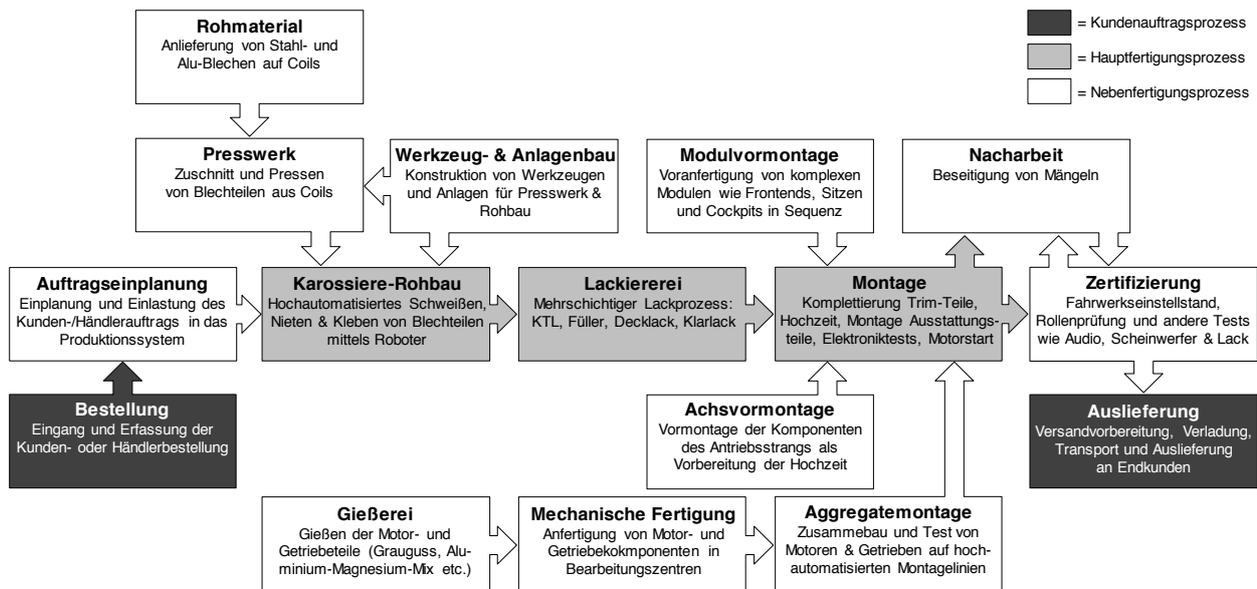


Abbildung 2-1: Kundenauftrags- & Fertigungsprozess Automobilbau i.A.a. [Kro09, Mei09]

Im **Karosseriebau**, auch Rohbau genannt, erfolgt das Fügen der Bauteile aus dem Presswerk mithilfe von Widerstandspunktschweißen, MIG- und MAG-Schweißen, Laserschweißen sowie in zunehmendem Maße durch Kleben [BHS13]. Der Werkzeug- und Anlagenbau, der an vielen Produktionsstätten ebenfalls angesiedelt ist, beliefert das Presswerk und den Karosseriebau mit speziell angefertigten Werkzeugen und Anlagen. Nach dem Ausschweißen und dem Einbau von Türen und Klappen erfolgt im Finish-Bereich des Rohbaus eine Qualitätsprüfung sowie die Weiterleitung an die **Lackiererei**. Hier wird in einem vierstufigen Prozess die kundenspezifische Lackierung aufgebracht. Nach dem Reinigen der Karosserie erfolgt zunächst das Aufbringen der Kathodischen Tauchlackierung (KTL) in großen Bädern. Anschließend wird ein Füller zum Ausgleichen der Oberflächen angewendet sowie nacheinander der Deck- und Klarlack aufgebracht. Trocknungsvorgänge zwischen den Schritten sorgen für eine ausreichende Härte des Lackes [VDA14]. Zeitgleich mit dem Einlauf der lackierten Karosserie in die Montagehalle werden bereits zahlreiche vorangestellte Prozesse abgeschlossen. Dazu zählt einerseits die **Aggregatefertigung**, in der für jedes Fahrzeug Motor und Getriebe hergestellt werden. Die Teilschritte dafür sind das Gießen der Motor- und Getriebeteile, die mechanische Fertigung der benötigten Komponenten in Bearbeitungszentren sowie die Montage und das Testen der Aggregate. Zudem erfolgt auch eine **Achsvormontage**, bei welcher der gesamte Antriebsstrang zusammengebaut wird, sowie die **Modulvormontage** großer Komponenten wie Frontend, Sitze und Cockpit. In einem nächsten Schritt findet in der **Montage** die sogenannte Hochzeit statt, d.h. die Verbindung von Fahrgestell und Karosserie. Anschließend werden Trimmteile wie Unterboden und Seitenverkleidung eingefügt, Ausstattungsteile wie Schiebedach und Navigationssysteme eingebaut, die Steuerungssoftware aufgespielt und der erste Motorstart durchgeführt. In der nun folgenden **Abnahme** werden diverse Qualitätssicherungsmaßnahmen wie Dichtigkeitsprüfung, Rütteltest und Funktionstest durchgeführt. Schließlich erfolgt die **Versandvorbereitung**, Verladung und der Transport des Fahrzeugs bis zum Endkunden bzw. bestellenden Händler [Kro09].

Der **Kundenauftragsprozess (KAP)** erstreckt sich von der Bestellung über die Fertigung bis zur Auslieferung eines Fahrzeugs an den Endkunden [Her05]. Der Startpunkt für die Fertigung nach dem Build-to-Order-Prinzip³ ist die Bestellung eines Fahrzeugs durch einen Kunden oder Händler [PaG08]. Die eingehende Bestellung wird erfasst und in die Auftragseinplanung überführt. Hier wird der Auftrag in einem mehrstufigen Prozess einem Werk zugeordnet und schließlich in die Produktion eingelastet (vgl. Abschnitt 2.2). Mit der Auftragsfreigabe erfolgt im Karosseriebau die physische Kennzeichnung eines Karosserieteils (z.B. Querträger vorne) mit einer eindeutigen Auftragsnummer [Her05]. Ab hier sind die Gewerke Karosseriebau, Lackiererei und Montage über einen einheitlichen Materialfluss miteinander verbunden und der Auftragsdurchlauf durch die Gewerke beginnt. Als Fördertechnik werden dafür z.B. Elektrohängebahnen und Fließbänder verwendet. Ein mobiler Datenspeicher begleitet das Fahrzeug über den gesamten Fertigungsprozess und ermöglicht damit das Erfassen von Statusmeldungen sowie steuerungstechnische Eingriffe in den Produktionsablauf [Her05].

Während die angewendeten Verfahren vom Grundsatz gleichgeblieben sind, hat sich mit der technischen Weiterentwicklung der **Automatisierungsgrad** drastisch erhöht. Mit Ausnahme der Montage verfügen die Gewerke heute häufig über Automatisierungsgrade über 90% [Her12]. In den modernsten Werken werden sogar noch höhere Automatisierungsgrade von bis zu 98% erreicht [Aud18]. Diese Werte konnten nur mithilfe von Arbeitsteilung und ausgeklügelten Organisationsformen realisiert werden. Die dahinterstehenden arbeitsorganisatorischen Prinzipien haben der Automobilindustrie zum Durchbruch verholfen und werden im Folgenden adressiert.

2.1.2 Arbeitsorganisation in der Automobilfertigung

Die Organisationsprinzipien der Produktion lassen sich grundsätzlich in die fünf in Abbildung 2-2 dargestellten Formen einteilen.

Fast dreißig Jahre nach der Erfindung des ersten Automobils mit Verbrennungsmotor durch Carl Benz und Gottlieb Daimler in 1885 waren Autos immer noch größtenteils in Werkstattproduktion gefertigte Unikate, was die Stückzahl stark begrenzte [Cla05]. Erst 1913, mit der Umsetzung der Fließfertigung im Highland Parks-Automobilwerk, führte Henry Ford den Automobilbau aus der traditionellen handwerklichen Herstellung in das Zeitalter der Massenfertigung [WJR92].

Dafür verband Ford zwei innovative Ansätze in Produkt und Produktion miteinander. Einerseits entwickelte er das durch Ransom Eli Olds begründete Konzept der **Fließbandproduktion** konsequent weiter und implementierte einen kontinuierlichen, mechanisch gekoppelten Materialfluss durch die gesamte Fabrik [Her12]. Andererseits trieb er die **Produktstandardisierung** auf die Spitze indem mit dem Modell T nur ein einziges, einheitliches Produkt angeboten wurde, das

³ Das Build-to-Order-Prinzip findet vor allem im europäischen Raum Anwendung. Im amerikanischen Raum hingegen wird das hybride Build-to-Order-Prinzip verwendet, da die Kunden die Fahrzeuge hier ohne Wartezeit beziehen möchten. Die Konfiguration und Bestellung der Fahrzeuge erfolgt daher auf Basis von Prognosen durch Händler [Han06].

selbst wiederum aus hochstandardisierten Bauteilen bestand. Durch die Kombination der beiden Ansätze konnte der von Taylor gepriesene hohe Grad der Arbeitsteilung und die damit einhergehende Trennung von dispositiven und ausführenden Tätigkeiten erreicht werden [GrK94, Sys06]. Geschichtlich gesehen entstand damit das erste formalisierte Produktionssystem der Welt – die **Massenproduktion**. In der Zeit von 1908 bis 1924 konnte Ford so den Preis des Modell T von 850 USD auf 260 USD senken [Cla05]. Damit wurden Autos erstmals für die breite Masse erschwinglich und das Konzept, das später von Alfred Sloan von GM weiterentwickelt wurde, hat der Automobilindustrie dazu verholfen zur größten produzierenden Industrie der Welt aufzusteigen [WJR07].

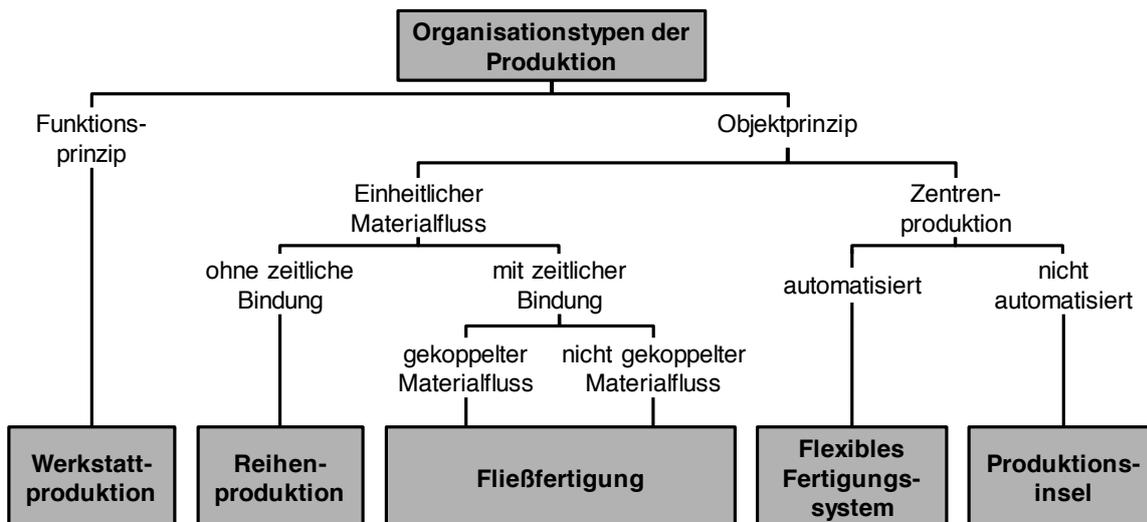


Abbildung 2-2: Organisationstypen der Produktion i.A.a. [Güt12]

Allerdings ist die Massenfertigung in vielen Bereichen auch mit Schwächen verbunden. Aufgrund der fehlenden Qualifikation der Arbeiter bedurfte es für viele Tätigkeiten wie Planung, Wartung und Qualitätskontrolle zusätzlicher Spezialisten [WJR07]. Die **Monotonie** der hochgradig arbeitsteiligen Aufgaben führte zu geringer Motivation und Eigenverantwortung sowie zu hohen Krankenständen, die durch Aufbau von Überkapazitäten kompensiert werden mussten [Uli05, WJR07]. Aufgrund mangelnder Qualitätskontrollen am Arbeitsplatz wurden Fehler erst am Ende der Linie behoben, was gegenüber der sofortigen Nachbesserung am Entstehungsort mit einem wesentlich höheren Aufwand verbunden ist [Ulr06]. Zudem benötigen größere Produktänderungen in dem sehr starren System, im Gegensatz zur Handwerksfertigung, oft mehrere Jahre und hohen Aufwand, sodass keine ausreichende Flexibilität gewährleistet war [WJR07].

Bis Ende der 1950er-Jahre verlangsamte sich daher das Wachstum in der Automobilindustrie und konnte erst durch die Einführung eines neuen Produktionsparadigmas, das später unter der Bezeichnung Lean Production bekannt wurde, wieder an Schwung gewinnen (vgl. Abbildung 2-3).

In der Tat sind die mit der Einführung der Lean Production einhergehenden Veränderungen so grundlegend, dass eines der bekanntesten Literaturstücke zu dem Thema in der deutschen Übersetzung „Die zweite Revolution in der Autoindustrie“ betitelt ist [WJR92].

Die Prinzipien der **Lean Production** wurden bereits in den 1950er-Jahren bei dem japanischen Automobilhersteller Toyota entwickelt und bis Mitte der 60er-Jahre unternehmensübergreifend implementiert [KIS14]. Die Begründer des Konzeptes, Eiji Toyoda und Taiichi Ohno, erkannten die Defizite der Massenproduktion und entwickelten als Gegenpol das Toyota Produktionssystem (TPS), um trotz der schwierigen wirtschaftlichen Lage nach dem zweiten Weltkrieg kosteneffizient zu fertigen. Es hat allerdings bis in die 1990er-Jahre gedauert bis dieses Konzept, vor allem dank dem International Motor Vehicle Programm, Einzug in die westliche Welt erhalten hat [WJR07].

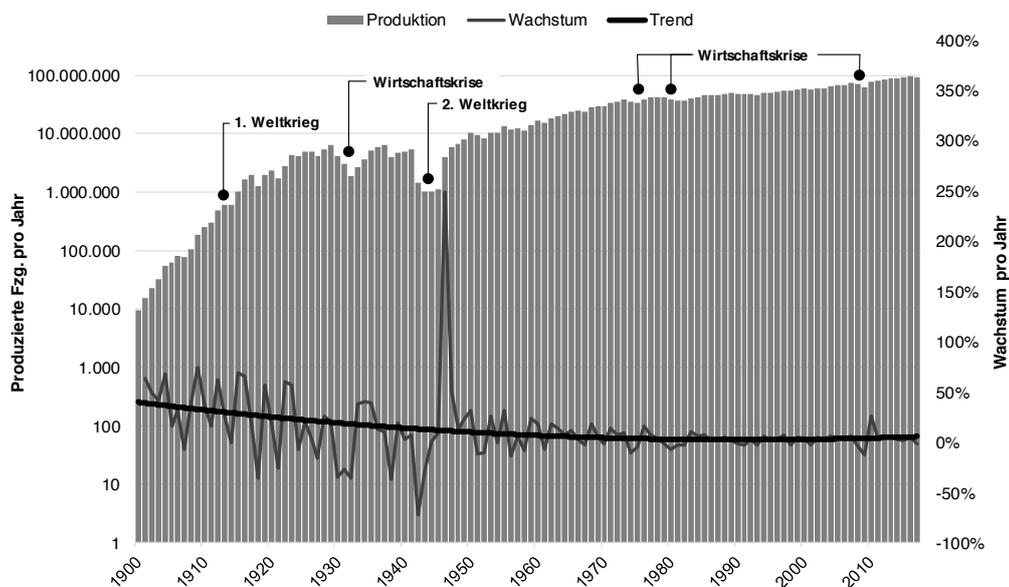


Abbildung 2-3: Globale Automobilproduktion von 1900 bis 2017 i.A.a. [Bus07, OIC18]

Der durchschlagende Erfolg von Lean Production führte zur Übertragung des Konzeptes auf andere Branchen und Anwendungsgebiete unter der Bezeichnung Lean Management. Beispiele für erfolgreiche Implementierungen existieren u.a. im Bereich Bürowesen, Gesundheitswesen, Start-ups und in der Softwareentwicklung [Rie11]. Letzteres wird auch als agile Softwareentwicklung bezeichnet und das dahinterstehende „agile manifesto“ wurde 2001 durch Softwareingenieure entwickelt [Bee01], die von dem Lean Production-Konzept aus Japan inspiriert wurden [Duh16].

Ein Kernmerkmal von Lean Production ist das „ganzheitlich integrative Aufgabenverständnis bei der Leistungserstellung“ sowie die Minimierung der strikten Arbeitsteilung der Massenproduktion, weswegen es auch als **post-tayloristisches Paradigma** bezeichnet wird [GrK94]. Die prägenden Eigenschaften des Konzeptes fußen auf der „Effizienz der Prozesse, weitgehende[r] Fehlerfreiheit der Produkte und Präzision bei Planung und Synchronisation parallel auszuführender Aufgaben“ [Ohn93].

Dabei werden drei grundsätzliche Quellen für eine ineffektive und ineffiziente Produktion unterschieden, die im Rahmen des TPS als **3M** bezeichnet werden: 1) Ungleichmäßigkeit bzw. Unregelmäßigkeit der Produktion (Mura), 2) Überlastung von Menschen und Maschinen (Muri) und 3) Verschwendung (Muda) [Ohn93].

Während Muda, auch als sieben Arten der Verschwendung⁴ bezeichnet, in der Fachliteratur spätestens seit Anfang der 1990er-Jahre stark im Fokus der Management-Literatur war, fanden die anderen beiden, Mura und Muri, vergleichsweise wenig Beachtung.

Dabei wird die **Kausalkette** zwischen den 3M (vgl. Abbildung 2-4) oft vernachlässigt und der Fokus auf Muda, also Verschwendung auf dem Shopfloor, gelegt. Muda ist jedoch häufig nur ein Puffer, der Mura kompensiert und damit die Unregelmäßigkeit in der Produktion verdeckt [Lik04]. Ferner entsteht Muda oft erst aus der Überlastung von Menschen und Maschinen, die wiederum aus der Ungleichmäßigkeit der Produktion selbst herrührt. Neue Chancen liegen daher in der Reduzierung von Mura durch Erhöhung von **Flexibilität und Robustheit von Produktionssystemen**. Die Stellhebel dafür liegen im Einflussbereich der PPS, die insbesondere vor dem Hintergrund der steigenden Variantenvielfalt zunehmend an Bedeutung gewinnt.

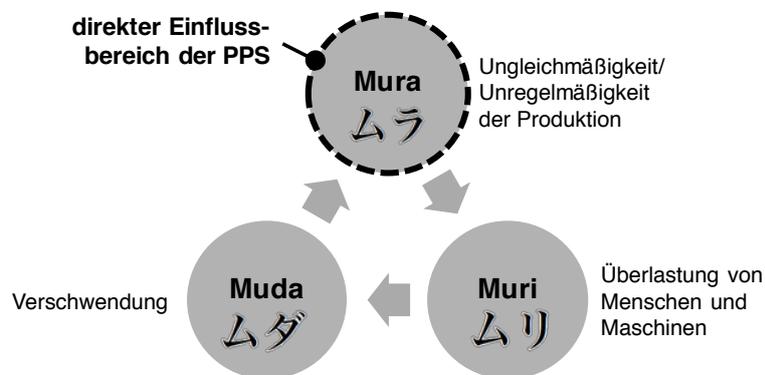


Abbildung 2-4: Kausalkette der 3M und Einfluss der PPS

Aufgrund des hohen Personalisierungsgrades und der damit einhergehenden Produktvarianz, sehen sich Automobilhersteller mit einer rasant steigenden Anzahl theoretischer **Konfigurationen** konfrontiert. BMW bietet z.B. bis zu 10^{17} Konfigurationsmöglichkeiten für das 7er-Modell und Daimler-Benz für die E-Klasse sogar bis zu 10^{24} Varianten an [PiH04, ESE13].

Die steigende Variantenvielfalt hat Auswirkungen auf nahezu alle Bereiche der Produktion: Sie erschwert die Arbeitsplanerstellung sowie die Konstruktion, Herstellung und Beschaffung von Werkzeugen. Sie führt zu komplexeren Stücklisten, aufwändigerer Fertigungs- und Montagesteuerung, kleineren Losgrößen, zusätzlichen Bereitstellungsflächen und Laufwegen für Werker sowie erhöhtem Risiko der Teilverwechslung [Ihm06]. Weitere Aspekte wie die steigende Relevanz von Umweltaspekten, die Zunahme an Elektronikkomponenten sowie anhaltender Kostendruck verschärfen die Situation weiter [GSW17]. Diese Entwicklungen führen zu neuen Anforderungen an Produktionssysteme und die PPS. Im Folgenden sollen daher zunächst die Grundlagen der PPS sowie ihr Einflussbereich umschrieben werden, damit im Anschluss die steuerungstechnischen Herausforderungen besser identifiziert werden können.

⁴ Überproduktion, Wartezeiten, unnötige Transportzeiten, ungünstige Herstellungsprozesse, überhöhte Lagerhaltung/Bestände, unnötige Bewegungen, Fehler bzw. Nacharbeit [TaM06]; später ergänzt um schlechte Ergonomie und unzureichende Kommunikation [KIS14].

2.2 Grundlagen der Produktionsplanung und -steuerung

Die bisherigen Ausführungen machen deutlich, dass die PPS einen erheblichen Einfluss auf wesentliche, zunehmend geforderte Eigenschaften von Produktionssystemen hat. Um den Einfluss der PPS auf die Fertigung ganzheitlich darzustellen, wird im Folgenden ein Vorgehen vom Groben zum Feinen gewählt, sodass ausgehend von den abstrakten Modellen der PPS eine schrittweise Konkretisierung bis hin zur praktischen Ausführung im Betrieb erfolgt.

Diesem Vorgehend folgend werden in Abschnitt 2.2.1 zunächst die zugrundeliegenden theoretischen Modelle, Prinzipien und Aufgaben der PPS erläutert sowie auf ihre Umsetzung in der Automobilindustrie eingegangen. Im darauffolgenden Abschnitt 2.2.2 wird der Fokus auf die Ausführung der Aufgaben im Rahmen der Fertigungssteuerung gelegt sowie ihre Ziele und die Stell- und Regelgrößen definiert. Da im Rahmen dieser Arbeit insbesondere der Steuerungsaspekt im Vordergrund steht wird anschließend in Abschnitt 2.2.3 auf die automatisierungstechnische Umsetzung der PPS eingegangen.

2.2.1 Aufgaben der PPS und Umsetzung in der Automobilindustrie

Die Funktion der PPS ist nach Wiendahl die laufende Vorausplanung des Produktionsprogramms für mehrere Planungsperioden, die Ableitung der resultierenden Material- und Ressourcenbedarfe sowie die bestmögliche Realisierung des Produktionsprogramms trotz unvermeidlicher Störungen [Wie14]. Sie ist notwendig um eine effiziente und schnelle Produktion zu erreichen und dabei gleichzeitig kurze Lieferfristen zu erzielen [Gün17]. Die einzelnen Aufgaben der PPS wurden von zahlreichen Autoren in teilweise ähnlicher und teilweise abweichender Form beschrieben (vgl. z.B. [Ada88, Höc98, BCG10]). Eines der in diesem Zusammenhang am häufigsten zitierten und am weitesten verbreiteten Modelle ist das 1998 von Luczak et al. publizierte Aachener PPS-Modell [LES98]. Aufgrund seiner Bekanntheit hat es sich als Referenzmodell etabliert, weshalb es für diese Arbeit als Rahmen dienen soll.

Das **Aachener PPS-Modell** besteht in seiner ursprünglichen Fassung aus Kern- und Querschnittsaufgaben. Aufgrund der zunehmend sinkenden Wertschöpfungstiefe sind Unternehmen jedoch immer stärker in Wertschöpfungsnetzwerke eingebunden. Als Konsequenz wurde das Modell durch Schuh und Stich aufgegriffen und um Netzwerkaufgaben erweitert [ScS12]. Die diesen drei Kategorien untergeordneten Aufgaben sind in Abbildung 2-5 dargestellt.

Da im Rahmen dieser Arbeit insbesondere auf die steuerungstechnischen Aspekte eingegangen wird, erfolgt im Folgenden eine kurze Beschreibung der Kern- und Querschnittsaufgaben nach [ScS12].

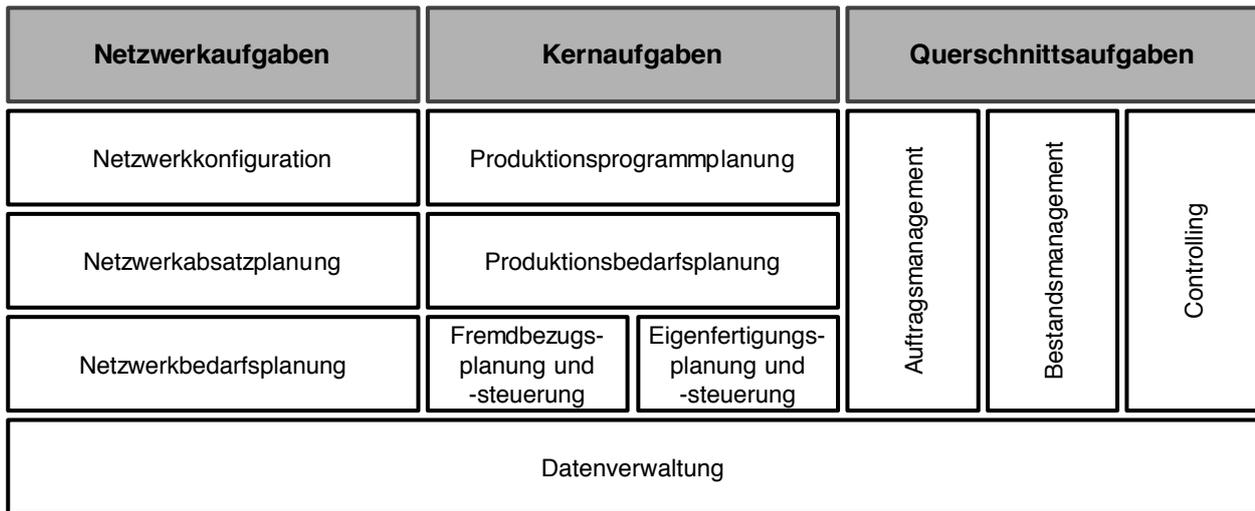


Abbildung 2-5: Aufgabensicht des Aachener PPS-Modells [ScS12]

Kernaufgaben der PPS

Die Kernaufgaben haben zum Ziel die Fertigstellung eines Auftrags voranzutreiben. Im Rahmen der **Produktionsprogrammplanung** wird daher für einen fixen Planungszeitraum Art, Menge und Zeit der herzustellenden Produkte definiert. Als Unteraufgabe erfolgt die Absatzplanung, bei der die Absetzbarkeit der Produkte in das Programm einfließt. In den darauffolgenden zwei Schritten *Primärbedarfs-* und *Ressourcengrobplanung* wird die Realisierbarkeit des Absatzplans geprüft und festgehalten.

Die **Produktionsbedarfsplanung** setzt auf dem Produktionsprogramm auf und konkretisiert mithilfe der Primärbedarfe die resultierenden mittelfristigen Ressourcenbedarfe. Dazu erfolgt zunächst eine *Brutto-* und später eine *Nettosekundärbedarfsermittlung* unter Abzug von Lagerbeständen, Reservierungen, Umlauf- und Sicherheitsbeständen etc. Ferner wird in der *Beschaffungsartzuordnung* ermittelt, ob der Bedarf durch Eigen- oder Fremdfertigung gedeckt werden soll und es wird im Rahmen der *Durchlaufterminierung* ein zeitlicher Zusammenhang zwischen den Fertigungsaufträgen hergestellt. Durch die *Kapazitätsbedarfsermittlung*, die an dieser Stelle das erste Mal von begrenzten Kapazitäten ausgeht, erfolgt eine weitere Konkretisierung des Bedarfs. Dieser wird schließlich in der *Kapazitätsabstimmung* mit dem Kapazitätsangebot abgeglichen.

Die in Eigenfertigung zu erzeugenden Produkte fallen in den Bereich der **Eigenfertigungsplanung und -steuerung**. Hier werden die Bedarfe zunächst der *Losgrößenrechnung* zugeführt, die wirtschaftlich optimale Fertigungslose ermittelt. Anschließend werden im Rahmen der *Feinterminierung* die Start- und Endtermine der Arbeitsgänge in jedem Fertigungsbereich bestimmt. Während dies unter der Annahme unbegrenzter Kapazitäten geschieht, konkretisiert die *Ressourcenfeinplanung* die Ergebnisse durch Einbezug der tatsächlichen Ressourcenbelastung. Im Anschluss findet die *Reihenfolgeplanung* statt, in der die Abarbeitungsreihenfolge der Aufträge an den Ressourcen determiniert wird. Nach einer *Verfügbarkeitsprüfung* findet schließlich die *Auftragsfreigabe* statt, in der die Bereitstellung der notwendigen Ressourcen veranlasst wird.

Die in Fremdfertigung hergestellten Bedarfe durchlaufen im Rahmen der **Fremdbezugsplanung und -steuerung** einen anderen Prozess als die Eigenfertigung. Zunächst wird in der *Bestellrechnung* unter Berücksichtigung von Beschaffungskosten wie Bestell-, Transport-, Versicherungs- und Verpackungskosten eine wirtschaftliche Bestellmenge ermittelt. Darauffolgend werden im Rahmen der *Angebotseinholung/-bewertung* erstmalig auftretende und noch nicht zugeordnete Bedarfe bei mehreren Lieferanten angefragt und zurückkommende Angebote bewertet. Auf dieser Basis findet im nächsten Schritt die *Lieferantenauswahl* statt, bei der Kriterien wie Qualität, Liefertermintreue, Preise und Lieferkonditionen bewertet werden und die in der Nominierung von Lieferanten mündet. Schließlich werden mit der Bestellfreigabe die tatsächlichen Bestellungen an die Lieferanten geschickt.

Querschnittsaufgaben der PPS

Die Querschnittsaufgaben dienen der Integration und Optimierung der PPS. Im **Auftragsmanagement** werden alle Aufgaben zusammengefasst, die für die Abwicklung eines Auftrags von Bedeutung sind. Zunächst erfolgt, ausgelöst durch eine Kundenanfrage, die *Angebotsbearbeitung*. Dafür werden alle betroffenen Bereiche koordiniert und Informationen bereitgestellt, um ein zeitlich und kostenseitig definiertes Angebot abzugeben und eine Kundenbestellung aufzunehmen. In der anschließenden *Auftragsbearbeitung* erfolgt die Überprüfung der Bestellung, die Anforderung kundenspezifischer Umfänge und die Bereitstellung von Daten für die Fachabteilungen. Zudem wird ein grober Produktionsdurchlauf festgelegt. In der anschließenden *Auftragskoordination* werden alle auftragsbezogenen Prozessüberwachungsaufgaben zusammengefasst. Zu den bedeutendsten zählen die Produktionsauftrags-, Ressourcen- und Bestellüberwachung.

Im **Bestandsmanagement** wird das Ziel verfolgt zur richtigen Zeit die richtigen Bestände in der geforderten Menge bereitzustellen. Dafür werden in der *Bestandsplanung und -analyse* angemessene Dispositionsstrategien und -parameter festgelegt sowie Rationalisierungspotenziale ausgeschöpft, um gleichzeitig Lagerbestände gering zu halten und Fehlmengen zu vermeiden. In der *Bestandsführung* werden operativ alle Zu- und Abgänge der Lager erfasst und geprüft. In der *Chargenverwaltung* werden schließlich alle Informationen zu der unter praktisch gleichen Umständen erzeugten Gesamtheit von Materialien gesammelt und für spätere Zwecke bereitgestellt.

Aufgabe des **Controllings** ist es Informationen zur wirtschaftlichen Steuerung der PPS bereitzustellen. Dafür werden im Rahmen der *Informationsaufbereitung* Kennzahlen identifiziert, erhoben und aufbereitet, um über einen Soll-Ist-Vergleich Probleme zu erkennen. Durch die *Maßnahmenableitung* sollen diese schließlich mit angemessenen Mitteln bewältigt werden.

Ein zentrales Instrument, das allen Netzwerk-, Kern- und Querschnittsaufgaben zur Verfügung steht ist schließlich die **Datenverwaltung**. Hier erfolgt die Speicherung und Pflege sämtlicher Daten. Dies beinhaltet einerseits die *Stammdatenpflege*, die z.B. Material-, Ressourcen-, Kunden- und Lieferantenstammdaten umfasst, aber auch Stücklisten und Arbeitspläne beinhaltet.

Das zweite Element der Datenverwaltung ist die *Bewegungsdatenpflege*, in der Daten zu Lagerbeständen, Produktionsauftragsdaten und Betriebsdaten erfasst werden.

Umsetzung der PPS in der Automobilindustrie

In der Automobilindustrie werden die Aufgaben der PPS nach einem ähnlichen Muster wie im Aachener PPS-Modell umgesetzt, denn auch hier muss zur Sicherstellung kurzer Lieferfristen der Bedarf schon vor Bestelleingang grob bekannt sein [KRK02, Klu18]. Die dafür notwendigen Bedarfsprognosen entstehen im Rahmen einer hierarchischen Produktionsplanungskaskade, in der externe und interne Lieferanten die Planung, Steuerung und Vorfertigung von Teilen einleiten [Her05, Sck15]. Beginnend bei der Absatzprognose wird der Planungshorizont schrittweise reduziert und der Detaillierungsgrad der Planung gleichzeitig erhöht (vgl. Abbildung 2-6). Die dafür wesentlichen Schritte werden im Folgenden kurz vorgestellt. Für eine ausführliche Betrachtung der Prozesse in der Automobilindustrie sei an dieser Stelle u.a. auf die Werke von [Her05], [Her12] und [Gün17] verwiesen.

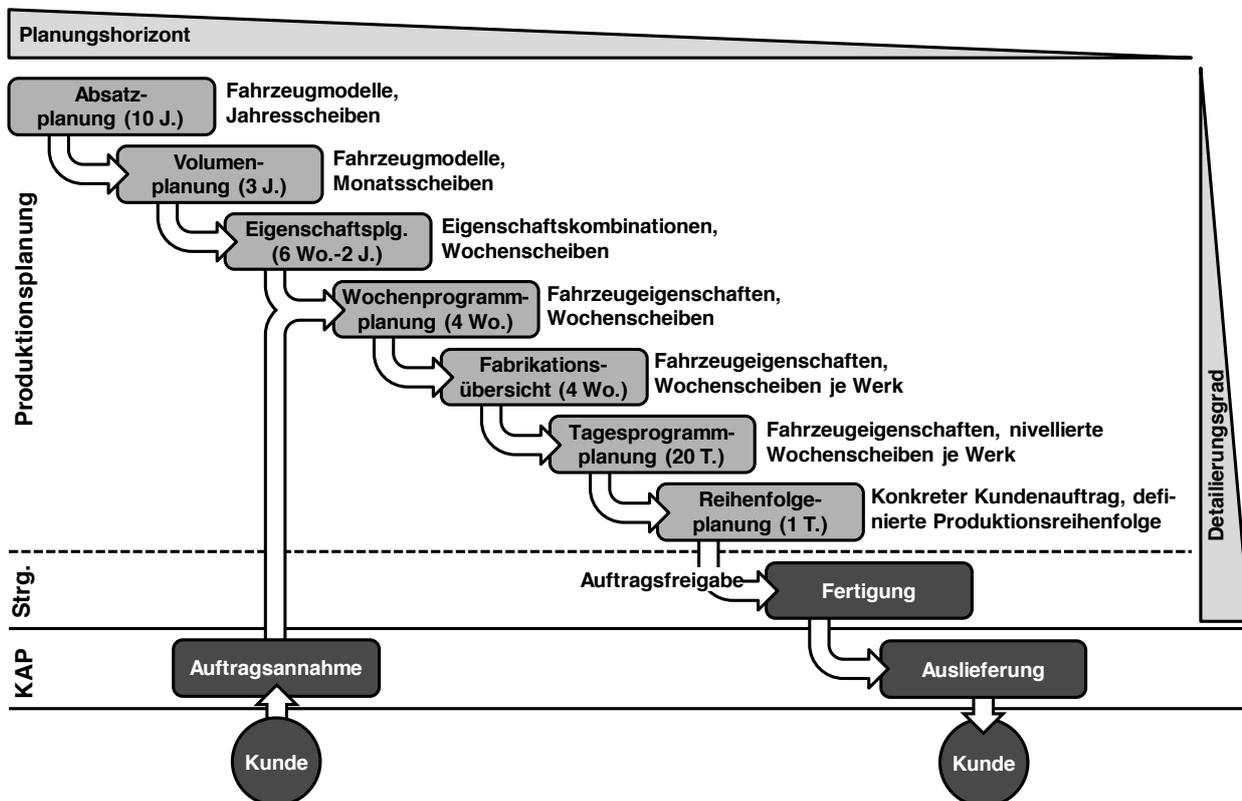


Abbildung 2-6: Hierarchie der PPS und Anknüpfung an den Kundenauftragsprozess

Der Planungsprozess beginnt mit der langfristigen **Absatzplanung**, in der auf Basis der Annahmen der Vertriebsregionen der Bedarf an Fahrzeugmodellen ermittelt und an die gegenwärtigen Marktbedingungen angepasst wird. Im nächsten Schritt erfolgt die **Volumenplanung** bei der die Menge der geforderten Fahrzeugmodelle auf die verfügbaren Plankapazitäten heruntergebrochen wird. Der Beschaffung fällt die Aufgabe zu, die notwendigen Kapazitäten bei Lieferanten frühzeitig abzusichern. Auf Seiten der Produktion und Logistik wird der Bedarf mit den geplanten

Kapazitäten in den verfügbaren Werken grob abgeglichen [Her05]. In der darauffolgenden **Eigenschaftsplanung** erfolgt eine Konkretisierung des Bedarfs mit Eigenschaftskombinationen der Fahrzeuge sowie ein genauer Abgleich mit vorhandenen Kapazitäten. Für Langläuferteile mit langen Werkzeugerstellzeiten erfolgt die Planung für bis zu zwei Jahre im Voraus. Alle anderen Teile werden in einem kurzen Planungshorizont von etwa sechs Wochen berücksichtigt [SSS08]. Aufgrund der hohen Individualisierungsmöglichkeiten ist in der Automobilindustrie fast jeder Auftrag ein Unikat. Mit Ausnahme des Presswerks wird daher üblicherweise mit einer Losgröße von eins gerechnet, sodass die **Losgrößenrechnung** für die Gewerke Rohbau, Lackiererei und Montage entfällt [Her12]. Die nächste Konkretisierung des Produktionsprogramms erfolgt im Rahmen der **Wochenprogrammplanung**. Hier fließen das erste Mal echte Kundenaufträge in die Planung ein und es erfolgt mit der Zuweisung der Aufträge zu definierten Wochen eine Entscheidung über den *Zeitraum* in dem ein Auftrag gefertigt werden soll [Boy05]. Da die Aufträge über festgelegte Fahrzeugmengen und -eigenschaften verfügen, kommt es bei Überschreitung der Kapazitäten einzelner Eigenschaften zu einem Überlauf von Aufträgen in die nächste Woche [Klu18]. Dies wird in grober Form schon bei der Auftragsannahme durch den Händler geprüft, sodass ein ungefährender Liefertermin für die Ausstattungskombination gegeben werden kann. Mit der Erstellung der **Fabrikationsunterlage** erfolgt die Verteilung des Wochenprogramms auf die verfügbaren Werke, sodass ein mit realen Kundenaufträgen gefülltes werkspezifisches Produktionsprogramm für eine Woche entsteht. Im nächsten Schritt, der **Tagesprogrammplanung**, wird diese Entscheidung weiter eingegrenzt. Dabei werden konkrete Restriktionen der Fertigung des jeweiligen Werks einbezogen und die Aufträge so verteilt, dass ein gleichmäßig nivelliertes Tagesprogramm entsteht. Damit soll einerseits eine gleichmäßige Belastung der Produktion gewährleistet werden. Andererseits erfolgt hierdurch eine Komplexitätsreduzierung, welche die im nächsten Schritt erfolgende **Reihenfolgeplanung** vereinfacht. Die unterschiedlichen Planungsansätze zur Bildung eines homogenen Produktionsprogramms wie z.B. Mixed-Model-Sequencing, Car-Sequencing und Level-Scheduling werden in Kapitel 7 ausführlicher besprochen. Mit der Bildung der Reihenfolge ist der planerische Teil der PPS abgeschlossen, sodass im nächsten Schritt die Steuerung des physischen Auftrags entlang der Produktion im Fokus steht.

Das Aachener PPS-Modell konzentriert sich vor allem auf die planerisch-gestalterischen Aufgaben der PPS und weniger auf den ausführenden Teil der eigentlichen Fertigungssteuerung. Da dieser jedoch insbesondere aus automatisierungstechnischer Sicht relevant ist, wird im nächsten Abschnitt ausführlicher auf die Aufgaben der Fertigungssteuerung eingegangen.

2.2.2 Aufgaben der Fertigungssteuerung

Aufgabe der Fertigungssteuerung ist nach Wiendahl die Umsetzung der Vorgaben der Produktionsplanung trotz, häufig unvermeidbarer, Störungen [Wie14]. Damit ist die Fertigungssteuerung ein besonders wichtiges Element zur Optimierung der **logistischen Zielgrößen** Durchlaufzeit,

Termintreue, Bestände und Auslastung (vgl. Abbildung 2-7). Lödning unterscheidet weiter zwischen internen und externen logistischen Zielen und führt dabei weitere Größen wie Servicegrad und Kaufpreis für externe Güter ein [Löd08].

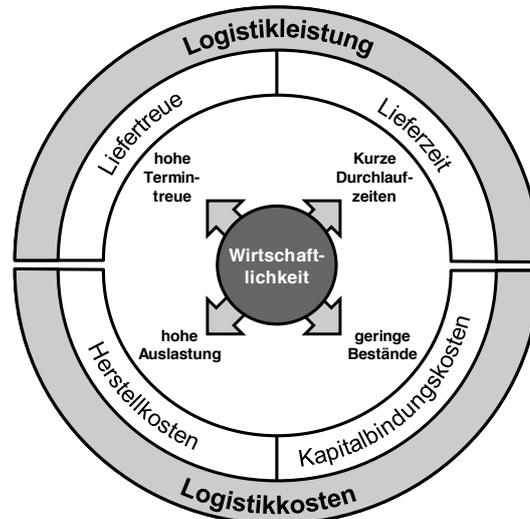


Abbildung 2-7: Ziele der Produktionslogistik i.A.a. [Wie14]

Die Fertigungssteuerung verfügt über drei wesentliche Stellgrößen mit denen sie die Produktion hinsichtlich der logistischen Zielgrößen beeinflussen kann. Diese sind: Beeinflussung des Ist- und Plan-**Zugangs**, des Ist- und Plan-**Abgangs** sowie der Ist- und Plan-**Reihenfolge** [Löd08]. Die drei Kernaufgaben der Fertigungssteuerung Auftragsfreigabe, Kapazitätssteuerung und Reihenfolgebildung haben zum Ziel diese Stellgrößen so zu beeinflussen, dass ein möglichst wirtschaftliche Ergebnis dabei herauskommt [Löd08]. Mit der **Auftragsfreigabe**, im Automobilbau häufig auch Einlastung oder Bandaufgabe genannt, wird die physische Erstellung des Auftrags in der Fabrik angestoßen. Diese richtet sich nach der geplanten Reihenfolge aus der Produktionsplanung. Im Rahmen der **Reihenfolgebildung** kann die Planung jedoch an die akute Kapazitätssituation in der Fabrik angepasst werden. Bei Störungen können Aufträge gesperrt werden und es wird durch Hinzunahme anderer Aufträge, für die keine Sperrung vorliegt, eine neue Auftragsreihenfolge erstellt. Gründe für Störungen können z.B. Anlagendefekte, Lieferengpässe oder Qualitätsprobleme sein. Unter **Kapazitätssteuerung** ist sowohl der Einsatz technischer Kapazitäten als auch die Anwendung organisatorischer Maßnahmen zu verstehen. Dies beinhaltet z.B. die Veranlassung von Überstunden oder Sonderschichten zur kurzfristigen Erhöhung der Produktionskapazität oder, im umgekehrten Fall, die Veranlassung von Kurzarbeit [Löd08]. Der Zusammenhang zwischen den Stell-, Regel- und Zielgrößen wurde durch Lödning in dem Modell zur Fertigungssteuerung zusammengefasst (vgl. Abbildung 2-8) [Löd08].

Die Zielgrößen können, wie wir in späteren Kapiteln sehen werden, oft weiter unterteilt bzw. konkretisiert werden, um z.B. die Leistung/Kosten oder Aspekte davon in abgeschlossenen Bereichen (z.B. Auslastung Cockpitvormontage) oder einzelnen Anlagen (z.B. Schweißpunkte pro

Stunde) zu messen. Zur Konkretisierung der Auslastung können z.B. Kennzahlen verwendet werden wie Ausbringung/Durchsatz, Verfügbarkeit/OEE, Nutzungsgrad der technischen Kapazität etc.

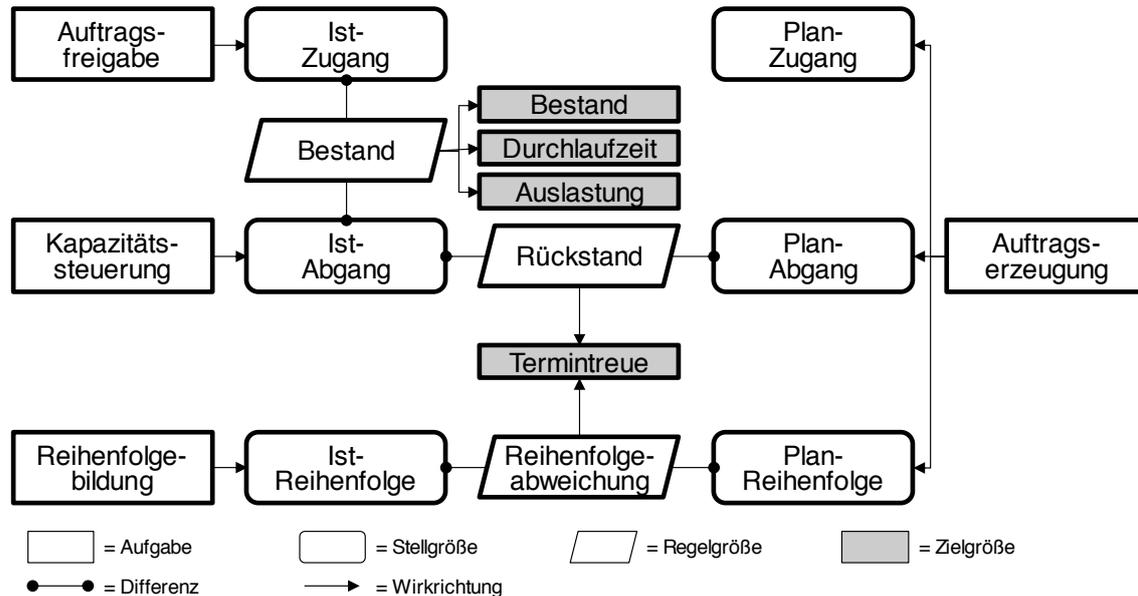


Abbildung 2-8: Modell zur Fertigungssteuerung i.A.a. [Löd08]

In der Automobilindustrie werden zur Bewertung der **Reihenfolgestabilität** häufig weitere Kennzahlen hinzugezogen. Dies ist notwendig, da sich Aufträge trotz der über das Fließbandprinzip gekoppelten Produktionsprozesse gegenseitig überholen können, insbesondere bei parallelen Produktionslinien. Um Aussagen über die Reihenfolgestabilität treffen zu können, werden bei nach dem **Perlenkettenprinzip** gesteuerten Fertigungen daher Kennzahlen wie die Perlenkettengüte (siehe Kapitel 7) erhoben, die Rückschlüsse auf die Einhaltung der Planreihenfolge zulassen. Die Reihenfolge kann dabei sowohl physisch und als auch virtuell verändert werden. Für eine physische Änderung der Reihenfolge muss nicht nur die steuerungstechnische Entscheidung für eine Resequenzierung getroffen werden, sondern die Entscheidung muss in einem weiteren Schritt auch operativ umgesetzt werden. Die dafür notwendige Ansteuerung von Anlagen und Geräten findet mithilfe von Automatisierungstechnik statt, die Thema des nächsten Abschnitts ist.

2.2.3 Automatisierungstechnische Sicht auf die PPS

Damit die in modernen Fabriken vorherrschenden Automatisierungsgrade erreicht werden können bedarf es zahlreicher Systeme. Unter **Automatisierung** ist im Rahmen dieser Arbeit die selbstständige Erfüllung von vorgegebenen Aufgaben und Funktionen zu verstehen [Lit13]. Wie die Aufgaben der PPS selbst, können auch die Systeme, die diese Aufgaben erfüllen nach dem zeitlichen Horizont ihres Einflusses unterteilt werden. Während langfristige Planungsaufgaben wie die Absatzplanung einen Zeithorizont von mehreren Jahren umfassen, bezieht sich z.B. die

Reihenfolgeplanung auf einzelne Tage oder Schichten und die automatisierungstechnische Ausführung sogar nur auf Millisekunden [Kle06].

Um die Komplexität von Automatisierungssystemen zu handhaben existieren Modelle, die Systeme in verschiedene Ebenen strukturieren und sie dadurch übersichtlicher und verständlicher machen. Im Bereich der Automatisierungstechnik ist die **Automatisierungspyramide** das am weitesten verbreitete Modell [Hei15]. Die Automatisierungspyramide gliedert sich in sechs Ebenen, die sich von der Unternehmensleitebene bis zum eigentlichen Produktionsprozess erstrecken (vgl. Abbildung 2-9).

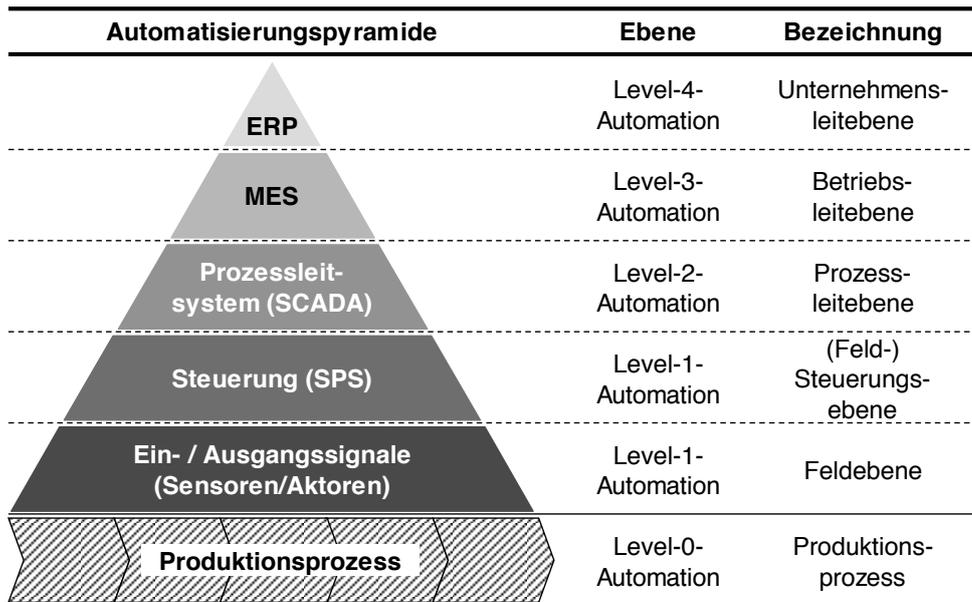


Abbildung 2-9: Automatisierungspyramide i.A.a. [Hei15, HLG15, Sie16]

Auf der **Unternehmensleitebene** (engl. *Enterprise Resource Planning* bzw. ERP) erfolgen langfristige strategische Tätigkeiten wie Auftragsabwicklung, Bestandsführung, Disposition, Beschaffung, Vertrieb, eine grobe Produktions-, Kapazitäts- und Investitionsplanung sowie Personalwesen und betriebliches Rechnungswesen. Diese Ebene bildet die in Abschnitt 2.2.1 beschriebenen Aufgaben der PPS ab. In der darunterliegenden **Betriebsleitebene** (manchmal auch Produktionsleitebene, engl. *Manufacturing Execution System* bzw. MES) finden kurz- bis mittelfristige Tätigkeiten wie die Verwaltung und Abarbeitung von Lieferaufträgen, Produktionsplanung, Bestandsführung, Terminüberwachung, Betriebsdatenverwaltung, Kostenanalyse, Personaleinsatzplanung sowie Mensch-Maschine-Kommunikation (engl. *Human-Machine-Interface* bzw. HMI) auf einem hohen Aggregationsniveau statt. Auf dieser und den darunterliegenden Ebenen finden die in Abschnitt 2.2.2 aufgezeigten Aufgaben der Fertigungssteuerung statt. Der **Prozessleitebene** (engl. *Supervisory Control And Data Acquisition* bzw. SCADA) kommen operative Funktionen wie komplexe Regelungen und Optimierung sowie die Koordination bei Störungen oder Umstellungen zu. Die HMI hat hier daher eine größere Detaillierungstiefe. Ferner werden hier auch Überwachungsfunktionen wie Qualitätskontrolle und Wartung ausgeführt sowie Rezepturen und Prüfauf-

träge verwaltet und Maschinendaten protokolliert. Die **Feldsteuerungsebene** übernimmt die Aufgabe der Regelung, Steuerung und Überwachung von Produktionsprozessen, sodass hier über HMI eine direkte Ansteuerung von Anlagen möglich ist. Die sich darunter befindende **Feldebene** enthält alle Sensoren und Aktoren zur Erfüllung des eigentlichen Produktionsprozesses. Hier findet das Messen und Stellen von Reglern mithilfe von Speicherprogrammierbaren Steuerungen (SPS, engl. *Programmable Logic Control* bzw. PLC) sowie die damit zusammenhängende Signalverarbeitung und Datenerfassung statt [Hei15, HLG15, Sie16].

Die Automatisierungspyramide wird durch eine Reihe von nationalen und internationalen Normen sowie weiteren Definitionen ergänzt und ausdetailliert. Zu den wichtigsten Normen zählen die ISA 88 und die ISA 95.

Die ursprünglich aus dem Bereich der chargenorientierten Fahrweise stammende **ISA 88** stellt mehrere Modelle auf der Prozessleitebene zur Verfügung. Dazu gehört u.a. ein physisches Modell eines Unternehmens mit den darin enthaltenen technischen Einrichtungen, ein Modell der Ablaufsteuerung mit Rezepten für die Durchführung von Prozessen, ein Prozessmodell zur Realisierung der Rezepte sowie ein Kontroll- und Aktivitätsmodell auf Anlagenebene [ISA10]. Aufgrund ihrer Bedeutung für die Automatisierungstechnik wurde der Standard durch die IEC als Norm IEC 61512 und in Deutschland im Rahmen der DIN EN 61512 veröffentlicht [DIN00].

Die **ISA 95** geht einen Schritt weiter als die ISA 88 und hat zum Ziel die Interoperabilität zwischen der Produktion und der Unternehmensführung herzustellen [AKS07]. Dazu stellt sie ein hierarchisches und ein funktionales Unternehmensmodell sowie ein Modell der Betriebsmittel-Hierarchie bereit. Auf dieser Basis wird ein Modell für den Informationsaustausch zwischen den Ebenen abgeleitet, mit dem die Interoperabilität sichergestellt werden soll [DIN14a]. Die Norm wurde unter der Bezeichnung IEC 62264 in englischer und als DIN EN 62264 in deutscher Sprache unter dem Titel „Integration von Unternehmens-EDV und Leitsystemen“ veröffentlicht [DIN14a].

Der Funktionsumfang auf den einzelnen Ebenen der Automatisierungspyramide kann darüber hinaus durch Integration weiterer Softwaremodule wie der **Betriebs- und Maschinendatenerfassung** bzw. dem **Asset Performance Management** erweitert werden. Da eine ausführliche Untersuchung aller Umfänge im Rahmen dieser Arbeit nicht möglich ist, wird hierfür auf die Werke [WeB06, Kle06, SGS16] verwiesen.

Nachdem mit diesen Ausführungen ein Überblick über den aktuellen Stand der Technik gegeben wurde, soll im nächsten Abschnitt aufgezeigt werden wie sich die Produktionssteuerung verändern muss, um den zukünftigen Rahmenbedingungen zu genügen.

2.3 Produktionssteuerung der Zukunft

Unter dem Begriff Industrie 4.0 finden derzeit zahlreiche Initiativen statt, mit denen die Produktion auf neue Herausforderungen ausgereicht werden soll. Ein wesentlicher Teil dieser Bemühungen konzentriert sich dabei auf die PPS, da hier große Potenziale vermutet werden. In Abschnitt

2.3.1 wird daher zunächst aufgezeigt, welche Entwicklungen eine Neuausrichtung der PPS notwendig machen und welche Schwierigkeiten damit einhergehen. Darauf aufbauend werden in Abschnitt 2.3.2 Anforderungen an die Produktionssteuerung der Zukunft definiert und damit die Grundlage für die im Kapitel 3 stattfindende Diskussion von Lösungsansätzen gegeben.

2.3.1 Zunehmende Dynaxität und Folgen für die Produktion

Durch parallele Entwicklungen wie der **Personalisierung** und Anpassung von Produkten an **regionale Bedürfnisse** sowie der **Globalisierung** von Material- und Informationsflüssen steigt die Komplexität und Dynamik im Produktionsumfeld. In diesem Zusammenhang spricht man auch von **Dynaxität** [Mai04]. Dabei befinden wir uns derzeit an der Schwelle von komplizierten zu komplexen Systemen (vgl. Abbildung 2-10). Dies stellt eine besondere Herausforderung dar, denn durch den Sprung von der Kompliziertheit zur Komplexität können Produkte und Prozesse nicht mehr exakt beschrieben und Entwicklungen nicht genau prognostiziert werden [Bau14]. Dies bedeutet, dass Systeme stärker mit **nicht vorhersehbaren** und teilweise **nicht beherrschbaren Störungen** umgehen müssen. Bekannte Methoden wie Lean Production aus dem Umfeld von komplizierten Systemen reichen nicht mehr aus, um den Herausforderungen in komplexen Systemen gerecht zu werden. Es bedarf neuer Herangehensweisen und der Weiterentwicklung heutiger Produktionssysteme, um die Vielzahl unvorhersehbarer Ereignisse zu handhaben, die mit der neuen Dynaxität einhergehen.

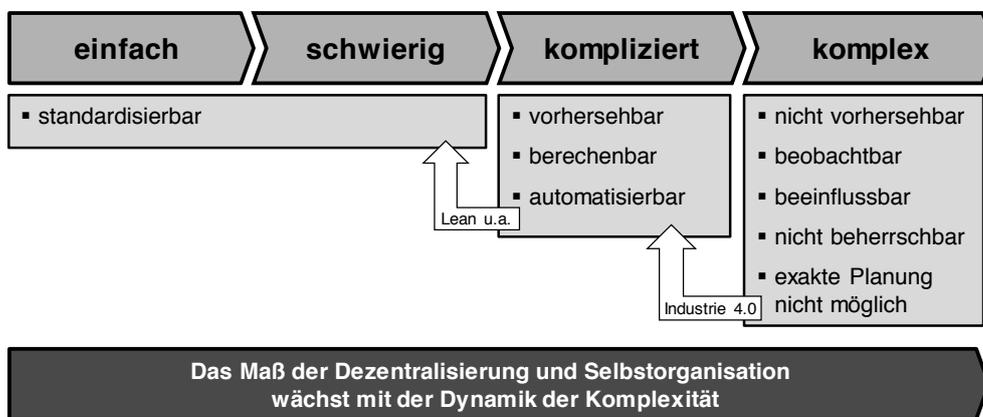


Abbildung 2-10: Entwicklung von komplizierten zu komplexen Systemen i.A.a. [GBS17]

Vor diesem Hintergrund werden die zu lösenden Optimierungsprobleme der PPS größer und schwieriger, weswegen unter dem Stichwort Industrie 4.0 mehr Dezentralisierung und Selbstorganisation von Produktionssystemen verlangt wird [GBS17].

Dörmer gliedert die Planungsprobleme in der variantenreichen Fließproduktion, zu der die Automobilindustrie gehört, in zwölf Kategorien. Diese erstrecken sich von der strategischen bis zur operativen Planung [Dör13]. Anhand dieser Systematik können u.a. folgende konkrete Herausforderungen abgeleitet werden, deren Lösung durch die zunehmende Dynaxität deutlich erschwert wird:

- Bei der Bildung von **Wochen- und Tagesprogrammen** müssen immer mehr Eigenschaftskombinationen berücksichtigt werden.
- Bei der **Reihenfolgebildung** fließen mehr Karosserievarianten, Ausstattungskombinationen sowie technische und vertriebsseitige Restriktionen ein; dies erschwert auch die Bildung optimaler **Farblöcke** für den Lackierprozess sowie die Bestimmung des **Personalbedarfs** und der **Austattung der Linien**.
- Im Rahmen der **Reihenfolgesteuerung** muss die **Einlastung von Aufträgen** in der richtigen Reihenfolge und zur rechten Zeit geschehen, um Maschinen und Arbeiter bestmöglich auszulasten und mithilfe von **Resequenzierung** während der Laufzeit effizient auf Störungen und Verwirbelungen⁵ zu reagieren.
- Eine erhöhte Anzahl an individuellen Teilen erschwert die **Materialbedarfs- und Beschaffungsplanung**. Die richtigen Teile müssen zur richtigen Zeit und in der richtigen Menge bestellt und im Rahmen der **Abrufsteuerung** (z.B. über JIT- oder JIS-Abrufe) bezogen werden, damit sie für den Einbau in kundenindividuelle Aufträge verfügbar sind.
- Damit werden auch die Herausforderungen in der **Lagersteuerung** größer, die das Ziel verfolgt möglichst wenig Bestände aufzubauen, da Lagerplatz nur beschränkt verfügbar ist und Bestände mit Kapitalbindungskosten verbunden sind.
- Die **Distributions- und Transportplanung** muss so getaktet werden, dass das richtige Material zur richtigen Zeit am richtigen Ort ist, um in das richtige Fahrzeug eingebaut zu werden; dies führt zur Komplexitätserhöhung in der **Materialflusssteuerung** und der Steuerung von Leitsystemen für die **Materialandienung** wie Gabelstapler, Routenzüge oder autonomen Fahrerlosen Transportsystemen (FTS).
- Die **Distribution** der fertigen Produkte, z.B. per LKW, Bahn oder Schiff, wird erschwert, da die Auslieferungsplanung zusätzliche Faktoren für die Ermittlung der optionalen Beladungsreihenfolge und zur Maximierung der Termintreue berücksichtigen muss.

Obwohl die obenstehende Liste keine erschöpfende Übersicht der Herausforderungen darstellt, machen die Beispiele deutlich, dass es mit immer **größerem Aufwand** verbunden sein wird eine angemessene Lösung für drängende Optimierungsprobleme zu finden. Dies ist jedoch nicht die einzige Herausforderung, die sich durch die geänderten Rahmenbedingungen ergibt. Die Vielzahl der ständig auf das Produktionssystem einwirkenden Faktoren führt des Weiteren dazu, dass die ermittelten Lösungen schnell ihre **Gültigkeit verlieren**, da während der Ermittlung der Lösungen bereits neue Information wie Anlagenstörungen, Fehlmaterial oder Personalausfälle hinzukommen [SaK99]. Diese **geringe Lösungsstabilität** ist u.a. in der hohen Anzahl unvorhersehbarer Ereignisse, der Komplexität von multikriteriellen Optimierungen und dem Einfluss von menschlichen Entscheidungsträgern begründet [PML18].

⁵ Verwirbelungen sind ungeplante Veränderungen der Soll-Reihenfolge hin zur Ist-Reihenfolge [Mei09].

Für die Lösung solcher Planungsprobleme (engl. *scheduling*) existieren in der Literatur zahlreiche Lösungsansätze. Exakte Verfahren versuchen Optimierungsprobleme mathematisch zu lösen, benötigen aber gerade bei großen Problem sehr viel Rechenzeit und eignen sich oft nicht für die Praxis. Heuristiken wiederum sind auf konkrete Problemfälle zugeschnittene Lösungsverfahren, die aufgrund von Vereinfachungen oft in der Lage sind gute Lösungen in einer angemessenen Zeit zu ermitteln. Zu bekannten Heuristiken gehören Ansätze wie z.B. Constraint Propagation Techniken, Simulated Annealing und genetische Algorithmen. Es können zur Lösungsermittlung aber auch andere Ansätze wie die Anwendung von Fuzzy Logic oder neuronalen Netzwerken verwendet werden. Die Nutzung konventioneller Ansätze in realen Applikationen ist vor dem Hintergrund der zunehmenden Dynaxität jedoch mit immer größeren Herausforderungen verbunden [SWH06].

Dies liegt an der Schwierigkeit der zur Lösung der Scheduling-Probleme erforderlichen **Rechenoperationen**. Zur Bewertung der Schwierigkeit der Probleme kann die aus der Komplexitätstheorie stammende NP-Schwere herangezogen werden [Bru07]. Die Klasse P steht dabei für Entscheidungsprobleme deren Lösung effizient überprüfbar ist, d.h. Lösungen, die in Polynomialzeit nachvollzogen werden können. Im Gegensatz hierzu stehen **NP-vollständige** und **NP-schwere** Probleme [GaJ79]. Sie sind nichtdeterministisch und können mit heutigen Mitteln vermutlich nicht effizient gelöst werden. Kann nachgewiesen werden, dass ein Problem NP-vollständig oder sogar NP-schwer ist, ist mit hoher Wahrscheinlichkeit anzunehmen, dass kein effizienter Algorithmus zur Ermittlung einer Lösung existiert [BrK12].

Ein prominentes Beispiel dafür ist die **Reihenfolgebildung** in der Automobilproduktion. Hier müssen einerseits hunderte oder sogar tausende Produktionsaufträge mit jeweils bis zu 10^{27} Eigenschaftskombinationen berücksichtigt werden. Andererseits bilden die Ausprägungen und die Auslastung der Produktion harte Restriktionskriterien wie z.B. Verbaureihenfolge, Sperrungen und Störungen, die unter allen Umständen eingehalten werden müssen. Dies macht die Reihenfolgebildung zu einem schwer zu lösenden Optimierungsproblem [SCN08]. Um ein Optimierungsproblem zu bewerten sollte es zunächst klassifiziert werden. In der Produktionswirtschaft haben sich dafür vier wichtige Organisationstypen herausgebildet. Diese sind das Einmaschinenmodell (engl. *single machine*), das Parallelmaschinenmodell (engl. *parallel machines*), die Fließproduktion (engl. *flow shop*) sowie die Werkstattfertigung (engl. *job shop* und *open shop*) [Ise16]. Die Reihenfolgebildung in der Automobilproduktion ist grundsätzlich als **Flow Shop** zu klassifizieren [Bru07]. Immer häufiger ist jedoch festzustellen, dass Produktionssysteme einen gemischten Charakter haben, der zwischen Flow-Shop und parallelen Maschinen liegt. Diese Form wird im Speziellen als **flexibler oder hybrider Flow-Shop** bezeichnet [JRC09]. Ein gewöhnliches Flow-Shop-Modell enthält einen mehrstufigen Produktionsprozess, bei dem die Aufträge alle Produktionsstufen in der gleichen Reihenfolge durchlaufen müssen. Ein flexibles Flow-Shop-Modell ist

eine Verallgemeinerung des klassischen Flow-Shop-Modells und stellt die Situation in der Fertigung realistischer dar. Es geht davon aus, dass mindestens eine Produktionsstufe mehrere Maschinen enthält, eine Maschine höchstens einen Auftrag und ein Auftrag von höchstens einer Maschine gleichzeitig bearbeitet werden kann. Darüber hinaus ist eine Fertigung im Voraus nicht zulässig, sodass das Optimierungsproblem darin besteht, die Aufträge den Maschinen in jeder Phase so zuzuordnen und die einer Maschine zugeordneten Aufträge so zu sequenzieren, dass die zu optimierenden Kriterien maximal verbessert werden [JRC09]. Da Flow-Shop-Probleme mit mehr als zwei Maschinen bereits NP-schwer sind, sind für andere PPS-Aufgaben in der Automobilfertigung ähnliche Herausforderungen wie bei der Reihenfolgebildung anzutreffen [DSV93]. Da die Scheduling-Theorie durch eine praktisch unerschöpfliche Anzahl an Optimierungsproblemen charakterisiert ist [Bru07], wird auf eine ausführliche Problembeschreibung an dieser Stelle verzichtet. Stattdessen werden in den Kapiteln 7 und 8 **konkrete Anwendungsfälle** skizziert und die dahinterliegenden Optimierungsprobleme und Ansätze im Detail beleuchtet.

Da nicht vorhersehbare Ereignisse per Definition nicht von vornherein eingeplant werden können, muss die Lösung folglich darin bestehen bestmöglich auf solche Ereignisse vorbereitet zu sein. Dafür sollten effektive Strategien vorhanden sein, um rechtzeitig optimale Handlungen einzuleiten, die entweder reaktiv oder unter Umständen sogar proaktiv ausgeführt werden können. **Ashby's „Law of requisite variety“**⁶ folgend, bedarf die zunehmende Komplexität eines Systems einer mindestens ebenso großen Zunahme der Wirk-, Handlungs- und Kommunikationsmöglichkeiten des Systems, um die darin auftretenden Störungen angemessen handhaben zu können [Ash56].⁷ Gleichzeitig sind viele der Scheduling-Probleme in der Fertigungssteuerung NP-schwer [GaJ79, BrK12] und damit durch Algorithmen nicht in angemessener Zeit lösbar. Aufgrund von Störungen im Fertigungsablauf werden darüber hinaus permanent Änderungen notwendig und Anpassungen von Prozessen und Abläufen sind die Folge.

Das Lösen komplexer Optimierungsprobleme durch zentrale Instanzen erfordert es jeden Auftrag, jede Entität und jeden Prozess im System aufs Neue zu kalkulieren. Unter den Echtzeitanforderungen im Produktionsumfeld ist dies auch mit den schnellsten Supercomputern der Welt derzeit nicht möglich. Forschungen im Bereich des **Quantum Computing** erwecken die Hoffnung, dass durch Ausnutzung des Superpositionsprinzips und des Phänomens der Quantenverschränkung, zukünftig zumindest Lösungen für NP-vollständige Optimierungsprobleme auch in Polynomialzeit ermittelt werden können [KLB18]. Da es sich, trotz erster Erfolge mit Rechenleistungen von bis zu 20 Qubits [Cas17], hierbei aktuell um ein weitestgehend **theoretisches Konzept** handelt, bedarf es anderer oder zusätzlicher Ansätze.

⁶ Dt. „Gesetz von der erforderlichen Varietät“.

⁷ Ashby schreibt weiter: „[...] variety can destroy variety.“

Innovationen entstehen oft aus der Übertragung bekannter Prinzipien auf neue Domänen. Daher kann auf der Suche nach neuen Ansätzen für die Produktionssteuerung z.B. auf die Erfolgskriterien der diskutierten Formen der Arbeitsorganisation wie der Lean Production und der agilen Softwareentwicklung zurückgegriffen werden. Die grundlegenden Prinzipien, die diesen Ansätzen zum Durchbruch verholfen haben, sind das Delegieren von Verantwortung an den Ort des Geschehens und die Befähigung der Arbeiter mit der größten Nähe zum Problem auf Basis dieser Verantwortung zu handeln [Duh16].

Diese **Dezentralisierung** von Entscheidungen ist auch auf die Steuerung von Automatisierungssystemen übertragbar. Eine Neuberechnung aller Produktionsprozesse aufgrund einer Störung an einer bestimmten Stelle im Produktionssystem erscheint unverhältnismäßig und ist mit überproportional langer Rechenzeit verbunden. Da währenddessen neue Ereignisse die Rahmenbedingungen verändern, sind die ermittelten Lösungen zum Zeitpunkt ihrer Finalisierung oft bereits veraltet. Die Agilität eines Gesamtsystems kann potenziell stärker erhöht werden, wenn die Entitäten des Systems selbst mit ausreichend **Entscheidungskompetenz** ausgestattet werden, um Entscheidungen am Ort des Geschehens und mit den dort verfügbaren Informationen in Echtzeit treffen zu können und somit flexibel auf unvorhersehbare Ereignisse zu reagieren.

Agentensysteme machen genau dies, indem sie die Entscheidungen auf viele Agenten am Ort des Geschehens verteilen, anstatt sie zentral über eine monolithische Software zu treffen. Bevor im Kapitel 3 der Fokus auf diese Lösungstechnologie gelegt wird, bedarf es zunächst einer ausführlichen Betrachtung der Anforderungen, die durch sie erfüllt werden müssen. Im nächsten Abschnitt wird daher der Versuch unternommen die wesentlichen Anforderungen an moderne Produktionssysteme zusammenzufassen.

2.3.2 Anforderungen an moderne Produktionssteuerungssysteme

Die im Zusammenhang mit den im vorherigen Abschnitt genannten Veränderungen führen zu zahlreichen Anforderungen an Produktionssysteme. Diese sind im Allgemeinen jedoch sehr breit gefasst und teilweise diffus [Sch13]. Sie verfolgen in ihrem Wesen jedoch ein ähnliches Ziel: Die Produktion soll flexibel auf Störungen und Ausfälle reagieren können, hochgradig wandlungsfähig und rekonfigurierbar sein sowie sich agil den kontinuierlich ändernden Rahmenbedingungen anpassen. Ferner wird Robustheit der Prozesse und Resilienz ganzer Unternehmen gefordert und Produktionssysteme sollen trotz einer hohen Variantenvielfalt eine sehr hohe Produktivität aufweisen [Bau12, KWH13].

Die oben verwendeten Begriffe sind sowohl in der Wissenschaft als auch in der Praxis vielfältig und werden mitunter synonym gebraucht [Ses03]. Daher erfolgt für die Verwendung in dieser Arbeit im Weiteren die Definition von sieben in der Literatur häufig genannten Anforderungen. Diese werden anhand von vier Kriterien kategorisiert: Zeithorizont, Auslöser der Veränderung,

Einfluss auf die Struktur des Produktionssystems und Einfluss auf die Ebenen des Unternehmens.

Flexibilität

Flexibilität ist ein in der Literatur vielfach diskutierter Begriff, der im Allgemeinen die Fähigkeit eines Produktionssystems beschreibt sich an veränderte Umweltbedingungen anpassen zu können [Elb87, Ses03, DOS13]. Eine Konkretisierung wurde u.a. durch Eversheim vorgenommen:

„Unter Flexibilität soll [...] die Fähigkeit verstanden werden, auf Änderungen der äußeren Rahmenbedingungen ohne deutliche Einbußen an Produktivität und Wirtschaftlichkeit reagieren zu können. Flexibilität kann als eine erleichterte Anpassung des Kapazitätsangebots nach Art und Menge verstanden werden.“ [Eve96]

Zahlreiche weitere Werke führen ähnliche Definitionen auf, beschränken sich dabei aber nicht nur auf die Kapazität, sondern führen andere Flexibilitätsarten ein. Häufige Ausprägungen sind auch Produktmix-, Maschinen-, Materialfluss-, Bearbeitungs(folgen)-, Erweiterungs-, Produktmengen-, Durchlauf- und Durchsatzflexibilität. Je nach Quelle lässt sich über ein Dutzend unterschiedlicher Flexibilitätsarten identifizieren [TeK93, REF90, Hal99, Han01, Löd08].

Vor dem Hintergrund von Industrie 4.0 und den damit an moderne Produktionssysteme gestellten Anforderungen sind unter Flexibilität im Voraus geplante und definierte Prozesse bzw. Systeme zu verstehen, die innerhalb von vorgedachten aber begrenzten Korridoren ein möglichst breites Spektrum von Anforderungen abdecken [BVZ15]. Dabei hat die Flexibilität eine statische Ausprägung, da sie nur Reaktionen auf erwartete Veränderungen in vorgedachten Grenzen beinhaltet [Wie02, Pac12]. Während manche Autoren Flexibilität als taktische Fähigkeit ansehen [WEN07], begreifen sie andere, insbesondere im Automatisierungstechnischen Zusammenhang, als operative Eigenschaft [Unl15a].

Wandlungsfähigkeit

In Abgrenzung zur Flexibilität hat die Wandlungsfähigkeit (engl. *adaptability*) eines Produktionssystems eine langfristige, taktisch-strategische Ausprägung und einen dynamischen Charakter. Sie wird nach Westkämper et al. wie folgt definiert:

„Ein System wird als wandlungsfähig bezeichnet, wenn es aus sich selbst heraus über gezielt einsetzbare Prozess- und Strukturvariabilität sowie Verhaltensvariabilität verfügt. Wandlungsfähige Systeme sind in der Lage, neben reaktiven Anpassungen auch antizipative Eingriffe vorzunehmen. Diese Aktivitäten können auf Systemveränderungen wie auch auf Umfeldveränderungen hinwirken.“ [WZB00]

In großangelegten Studien wie z.B. von Toni und Tonchia [ToT98] und umfangreichen Literaturanalysen wie z.B. von Narain et al. [NYS00] wird der Begriff ausführlich untersucht. Den Definiti-

onen ist gemein, dass Wandlungsfähigkeit im Gegensatz zur Flexibilität mit strukturellen Veränderungen des Produktionssystems einhergeht [Wie02, RBE02]. Der Zusammenhang zwischen Flexibilität und Wandlungsfähigkeit ist in Abbildung 2-11 dargestellt.

Ferner wird unterschieden zwischen Wandlungsfähigkeit, die sowohl das technische als auch das soziale Systeme eines Unternehmens beeinflussen kann, und Wandelbarkeit (engl. *changeability*), die sich nur auf technische Systeme bezieht. Flexibilität wird in diesem Zusammenhang als eine Teilmenge der Wandlungsfähigkeit angesehen [WZB00]. Ähnlich wie bei der Flexibilität sind die grundsätzlichen Freiräume der Wandlungsfähigkeit, innerhalb derer Veränderungen stattfinden können, auch hier bereits vorgedacht [Nyh08]. Die reaktiven und antizipativen bzw. proaktiven Elemente sind des Weiteren nicht alleine der Wandlungsfähigkeit vorbehalten. Im Sinne der operativen Flexibilität ist auch innerhalb der Flexibilitätskorridore sowohl reaktives als auch proaktives Handeln möglich und erwünscht. Es bezieht sich allerdings auf einen kürzeren Zeithorizont und kleinere Dimensionen [Her03, McB03, DOS13].

Mit Bezug auf die Verwendung in dieser Arbeit ist unter Wandlungsfähigkeit daher ein Verschieben der Grenzen der Flexibilitätskorridore zu verstehen, das im Gegensatz zur Flexibilität mit strukturellen Veränderungen des Produktionssystems einhergeht und sich daher vor allem auf unerwartete Umwelteinflüsse bezieht [BVZ15, VDI17].

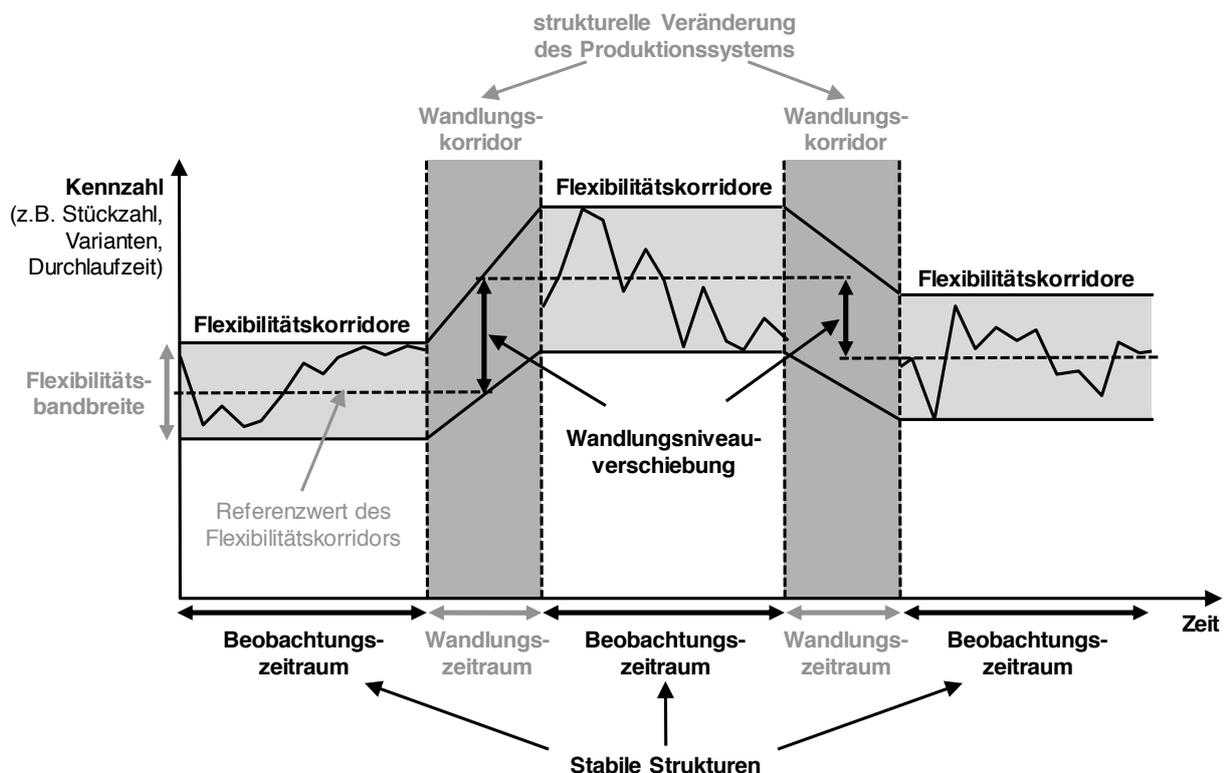


Abbildung 2-11: Abgrenzung Flexibilität und Wandlungsfähigkeit i.A.a. [ZMV05, VDI17]

Robustheit

Der Begriff Robustheit (engl. *robustness*) beschreibt die auf ein Gesamtsystem bezogene Fehlertoleranz [Sch13] und die Fähigkeit eines Systems auch unter Einfluss von Störungen und Unterbrechungen die geforderte Leistung zu erbringen [CTM07]. Da es sich dabei um eine operative

Eigenschaft handelt, müssen Reaktionen auf erwartete und unerwartete Störungen mit einer unveränderten Struktur des Produktionssystems bewältigt werden.

Gribble beschreibt Robustheit im Kontext komplexer Systeme daher als:

„[...] the ability of a system to continue to operate correctly across a wide range of operational conditions, and to fail gracefully outside of that range.“ [Gri01]

Dieser Definition folgend leitet der VDA Merkmale von robusten Produktionsprozessen ab. Dazu zählen u.a. die Unempfindlichkeit gegenüber Störgrößen, die Einhaltung der Produktqualität sowie die Gewährleistung des Betriebs über eine vorgegebene Lebensdauer bei gleichzeitiger Einhaltung des geplanten wirtschaftlichen Aufwands [VDA07].

Robustheit kann durch Adaptionsmechanismen erreicht werden, mit deren Hilfe Variationen der Umfeldbedingungen und Teilausfälle des Systems kompensiert werden [Ay06].

Adaptivität

Auf den Bedürfnissen der Robustheit aufbauend wird Adaptivität (engl. *adaptivity*) als Fähigkeit eines Systems bezeichnet sich ohne strukturelle Änderungen an unerwartete Konditionen anpassen zu können. Im Deutschen wird daher oft auch der Begriff Anpassungsfähigkeit gebraucht [Kle12]. Änderungen des Systemverhaltens können dabei die Faktoren Zeit, Kapazität und/oder Quantität betreffen [SJF11]. Unter Adaptivität wird ferner die selbständige Anpassung und Optimierung von Fertigungsprozessen und Prozessketten verstanden, wobei die Selbstständigkeit der Anpassungen betont wird [IPT16]. Von der Flexibilität grenzt sich die Adaptivität dadurch ab, dass sie sich nur auf das Systemverhalten bezieht und vorwiegend Änderungen durch externe Einflüsse auf ein Produktionssystem berücksichtigt. Daher ist sie dem taktischen Bereich zuzuordnen [Sch13].

Agilität

Agilität (engl. *agility*) kann als eine höhere Form von Flexibilität bezeichnet werden, denn sie stellt die allgemeine Fähigkeit dar unerwarteten Störungen zu begegnen. Damit grenzt sie sich von Flexibilität ab, die den Umgang mit Änderungen in einem erwarteten Rahmen betrifft [Zob05]. Zobel definiert Agilität wie folgt:

„Agilität [ist] die Fähigkeit zum profitablen Wandel bei Eintreten unerwarteter bzw. nicht vorhersehbarer Ereignisse [...]. Darüber hinaus ist die (proaktive) Option zur Initiierung von Wandel oder die Veränderung geltender Wettbewerbsspielregeln als ein weiterer wichtiger Bestandteil der Agilität anzusehen.“ [Zob05].

Hinsichtlich des zeitlichen Bezugsrahmens gibt es in der Literatur zwei Standpunkte. Einerseits wird Agilität im Sinne des „Agile Manufacturing“ ein strategischer Charakter zugeschrieben. Demzufolge ist Agilität die Handlungsfähigkeit des gesamten Unternehmens, d.h. der Systeme, Strategien, Technologien und Mitarbeiter, auf veränderte Rahmenbedingungen wie z.B. Sprünge in

der Kundennachfrage einzugehen [Gun99, Gun01, WEN07]. Andererseits wird Agilität als operative Eigenschaft im Sinne der Fehlertoleranz und Robustheit verstanden [LeR08].

Die Sichtweisen teilen die Gemeinsamkeit, dass sie beide die Eigenschaften Antizipation und Proaktivität als Teil von Agilität ansehen [Ter15]. Ereignisse müssen daher vorhergesehen und Handlungen ex ante eingeleitet werden, um ein agiles Verhalten zu erreichen.

Da die vorliegende Arbeit einen steuerungstechnischen Fokus hat, wird unter dem Begriff Agilität im Weiteren die vorausschauende und proaktive Handlungsfähigkeit technischer Systeme im Umgang mit unerwarteten Änderungen auf operativer Ebene verstanden. Zu den grundlegenden Ansätzen, die dies unterstützen, gehört u.a. die Rekonfigurierbarkeit.

Rekonfigurierbarkeit

Rekonfigurierbarkeit (engl. *reconfigurability*) ermöglicht die agile Anpassung der Struktur von technischen Systemen aufgrund unerwarteter Einflüsse [Ter15] und hat daher einen taktischen Charakter [Sch13]. Farid und McFarlane definieren Rekonfigurierbarkeit als:

“[...] ability to add, remove and/or rearrange in a timely and cost-effective manner the components and functions of a system which result in a desired set of reconfigurations.”

[FaM07]

Dabei ist nicht nur das Hinzufügen oder Entfernen von Kapazitäten gemeint, sondern auch der Wechsel zwischen definierten Teilefamilien und Baugruppen [WEN07]. Um dies umzusetzen werden autonome und standardisierte Funktionseinheiten gefordert wie sie z.B. Plug & Produce-Module darstellen [Nyh08]. Gleichzeitig wird hierfür eine gewisse Flexibilität und Skalierbarkeit der physischen Komponenten vorausgesetzt. Rekonfiguration kann sich aber auch auf logische Komponenten wie „rerouting“, „rescheduling“ und „replanning“ beziehen, die im Zusammenhang mit steuerungstechnischen Aspekten eine Rolle spielen [EIM05]. Die Zeit zur Berechnung einer neuen Konfiguration hängt dabei von der Größe und Flexibilität des Produktionssystems ab [VGL15].

Resilienz

Unter dem Begriff Resilienz (engl. *resilience*) wird in Zusammenhang mit Produktion teilweise eine Vielzahl von Eigenschaften verstanden. Kagermann et al. definieren Resilienz wie folgt:

„Resilienz bedeutet Widerstandsfähigkeit, aber auch Agilität, Adaptivität, Redundanz, Dezentralität und Lernfähigkeit.“ [KWH12]

Darunter verstehen die Autoren, dass ein breites Spektrum an kundenindividuellen Produkten bei schwankender Nachfrage produziert werden kann und dass durch eine situative Anpassung der Produktion eine optimale Kapazitätsauslastung erreicht wird [KWH12]. Bauernhansl erweitert die besagte Widerstandsfähigkeit um den Begriff der Regenerationsfähigkeit gegenüber Störungen und nennt z.B. Wirtschaftskrisen und Infrastrukturausfälle als Ursachen. Als Werkzeuge zum Umgang mit diesen Herausforderungen sieht der Autor bessere und frühzeitigere Prognosen auf

Basis von Smart Data. Damit soll eine unmittelbare Reaktion möglich sein, sodass sich z.B. Wertschöpfungsnetze kurzfristig neu bilden und Krisen besser und schneller bewältigt werden können [BHV14].

Allgemeiner gefasst sehen Scharte und Thoma Resilienz als:

„[...] Fähigkeit, tatsächliche oder potenziell widrige Ereignisse abzuwehren, sich darauf vorzubereiten, sie einzukalkulieren, sie zu verkräften, sich davon zu erholen und sich ihnen immer erfolgreicher anzupassen. Widrige Ereignisse sind menschlich, technisch sowie natürlich verursachte Katastrophen oder Veränderungsprozesse, die katastrophale Folgen haben.“ [ScT16]

Im Rahmen dieser Arbeit soll Resilienz dieser Definition folgend als Fähigkeit von technischen Systemen verstanden werden, bei Störungen bzw. Teilausfällen durch Einsatz der zuvor genannten Eigenschaften wie Flexibilität, Agilität und Wandelbarkeit widerstandsfähig gegenüber widrigen Ereignissen zu sein und wesentliche Systemfunktionen aufrechtzuerhalten.

Zusammenfassung

Die beschriebenen Anforderungen an Produktionssysteme sind in Tabelle 2-1 anhand der vier Kriterien Zeithorizont, Auslöser der Veränderung, Einfluss auf die Struktur des Produktionssystems und Ebene des Unternehmens zusammengefasst.

Der direkte Einflussbereich der Steuerungstechnik bezieht sich vor allem auf die operative Planung und Ausführung der Fertigungsaufgaben, sodass nicht alle der gestellten Anforderungen durch sie beeinflusst werden können. Die Steuerungstechnik stellt allerdings auch Mittel zur Verfügung, um taktische und strategische Anforderungen zu erfüllen. Plug & Produce-Funktionen können z.B. helfen die Rekonfigurierbarkeit von Produktionssystemen in Echtzeit zu realisieren, was wiederum Auswirkungen auf die Adaptivität und Wandlungsfähigkeit hat.

Anforderung	Zeithorizont	Veränderung	Struktur	Ebene
Robustheit	operativ	(un)erwartet	konstant	Gesamtsystem
Flexibilität	operativ-taktisch	erwartet	konstant	Bereich
Adaptivität	taktisch	unerwartet	konstant	Gesamtsystem
Rekonfigurierbarkeit	taktisch	unerwartet	verändert	Anlage/Gruppe
Agilität	operativ-strategisch	unerwartet	verändert	Bereich- Unternehmen
Wandlungsfähigkeit	taktisch-strategisch	unerwartet	verändert	Generalstruktur
Resilienz	taktisch-strategisch	unerwartet	konstant/ verändert	Unternehmen

Tabelle 2-1: Anforderungen an Veränderbarkeit moderner Produktionssysteme

Agentensysteme können prinzipiell auf allen Ebenen der Automatisierungspyramide eingesetzt werden, der Nutzen ist auf den Ebenen jedoch unterschiedlich zu bewerten. Im nächsten Kapitel sollen daher industrielle Agentensysteme und weitere Lösungsansätze als mögliche Technologie

untersucht werden, mit der die an moderne Produktionssteuerungssysteme gestellten Anforderungen erfüllt werden können und es soll definiert werden, wo ihr Einsatz am sinnvollsten erscheint.

3 Paradigmen der Produktionssteuerung

Die diskutierten Herausforderungen stellen Forschung und Industrie vor die Aufgabe die notwendigen Technologien zur Umsetzung der Vision Industrie 4.0 zu entwickeln. Obwohl erste Schritte bereits unternommen sind, fehlen wesentliche Elemente zur Realisierung einer vernetzten, intelligenten Produktion. Im folgenden Abschnitt 3.1 wird daher zunächst ein kurzer Überblick über Industrie 4.0 und die dafür geforderte Steuerungsarchitektur gegeben. Darauf aufbauend werden in Abschnitt 3.2 grundlegende Steuerungsparadigmen und -architekturen vorgestellt und ihre Zweckdienlichkeit diskutiert. Im anschließenden Abschnitt 3.3 werden drei Software-Design-Paradigmen vorgestellt sowie ihre Stärken und Schwächen untersucht. Schließlich werden die Paradigmen in Abschnitt 3.4 gegenübergestellt und es wird ein Ansatz für die weiteren Betrachtungen im Rahmen dieser Arbeit ausgewählt.

3.1 Status quo Industrie 4.0

Hinter der Vision von Industrie 4.0 verbirgt sich das ambitionierte Ziel die Industrie branchenübergreifend zu digitalisieren und sie damit in ein neues Zeitalter zu bringen. Der Begriff Industrie 4.0 ist 2011 im Rahmen des Zukunftsprojekts der Bundesregierung und der damit verbundenen High-tech-Strategie in Deutschland entstanden [BMB16]. In anderen Ländern werden mit anderen Initiativen ähnliche Ziele verfolgt. Im angelsächsischen Raum wird dabei oft vom **Industrial Internet** bzw. **Industrial Internet of Things (IIoT)** gesprochen [KAG16]. In Frankreich wurde die Initiative „Industrie du Futur“ [AIF18] und in China „Made in China 2025“ aufgesetzt [The17].

Die Bestrebungen die Produktion mithilfe von Informationstechnik (IT) zu vernetzen sind jedoch nicht neu. Im 1973 von Harrington veröffentlichten Werk [Har73] wird das erste Mal von **Computer Integrated Manufacturing (CIM)** gesprochen. Dahinter verbirgt sich die Zielsetzung die verschiedenen computergestützten Systeme wie CAD, CAP, CNC, CAM, CAQ, PPS und BDE innerhalb von Unternehmen zu integrieren. Dadurch sollte die Leistungserstellung von Unternehmen ganzheitlich betrachtet und durch integrierte IT-Systeme unterstützt werden [Sod17]. Die Bestrebungen führten allerdings nicht zu den erhofften Ergebnissen. Eine zum damaligen Zeitpunkt nicht überwindbare technische Hürde lag im Mangel an Datensystemen, Sensorik und Datenübertragungstechnik mit ausreichender Leistungsfähigkeit und angemessenen Preisen. Dies machte die Produktion gleichzeitig teuer und schwer beherrschbar und führte zum Scheitern von CIM [DOS13, Sod17].

Weitere Konzepte wie der 1992 durch Warnecke eingeführten Begriff „**Fraktale Fabrik**“ hatte zum Ziel die bereits automatisierte Produktion in kleine, selbständige Einheiten zu zerteilen, die miteinander kooperieren [War92]. Das Konzept der **Flexiblen Fertigungssysteme (FFS)** wiederum sollte die Flexibilität der Produktion auf Ebene der Maschinen erhöhen [Eve89].

Vor diesem Hintergrund stellt sich die Frage wie sich Industrie 4.0 von den bisherigen Initiativen unterscheidet. Zwei konzeptionelle Unterschiede stehen dabei besonders hervor. Einerseits die

steuerungstechnische Ausgestaltung der zugrundeliegenden Systeme und andererseits die Rolle des Menschen. Während CIM auf zentral gesteuerte, hoch integrierte Systeme auf Basis von Leitrechnern setzte, wird im Rahmen von Industrie 4.0 auf **dezentrale Steuerung** kooperierender **cyber-physischer Produktionssysteme** (CPPS) gesetzt. Diese Produktionssysteme bestehen dabei aus einer Vielzahl von selbstständigen, aber interagierenden cyber-physischen Systemen (CPS), die sich **selbst organisieren** und auf Basis von Webdiensten **cloudbasiert kommunizieren**. Darüber hinaus wird im Gegensatz zu CIM nicht das Ziel verfolgt menschliche Arbeitskräfte vollständig überflüssig zu machen, sondern sie über **Augmentation** und **Vernetzung** intelligent und nutzenstiftend in das CPPS einzubinden [Bau13].

Der Begriff Industrie 4.0 hat seit seiner Entstehung zu hohen, möglicherweise sogar **überzogenen Erwartungen** geführt und einen regelrechten Hype ausgelöst [Rot16]. Die Erwartungen konnten bisher allerdings nicht überall erfüllt werden – eine Entwicklung, die dem **typischen Verlauf aufkommender Technologien** entspricht und z.B. im Gartner Hype Cycle regelmäßig verfolgt wird [Gar17]. Die einkehrende Ernüchterung basiert auf dem Mangel an Umsetzungstechnologien zur Realisierung der erhofften wirtschaftlichen Potenziale. Viele der **hardwareseitigen Hemmnisse**, die z.B. die Realisierung von CIM in den 80er-Jahren verhindert haben, existieren heute nicht mehr. Rechenleistung und Speicherplatz sind in ausreichendem Maß und zu günstigen Preisen verfügbar [Bra14]. Durch die Reduzierung der Größe von Rechen- und Speichermodulen stellt die Implementierung komplexer eingebetteter Systeme keine Herausforderung mehr dar [Wan17]. Die verbleibenden Herausforderungen liegen daher vor allem im Bereich der **Steuerungsarchitektur** [VDE13]. Die Automatisierungstechnologie muss nun also, ähnlich wie die IT-Landschaft durch den Internet-Boom der 90er-Jahre, ebenfalls eine Entwicklung nachvollziehen, bei der die zugrundeliegenden Technologien vorangebracht, Standards definiert und Entwurfspattern abgeleitet werden [EIH10]. In der Deutschen Normungsroadmap Industrie 4.0 [DIN18] wird der Weg zur weiteren Standardisierung in diesem Bereich gezeichnet und eine Dokumentation bereits erarbeiteter Ergebnisse kann z.B. im VDI-Forum Industrie 4.0 (<https://www.vdi.de/industrie40/>) abgerufen werden. Der Ansatz, den Industrie 4.0 dabei verfolgt ist die **Verteilung von Steuerungsentscheidungen** auf viele miteinander vernetzte Entitäten. Das zugrundeliegende Steuerungsprinzip auf Basis von **dezentral verteilten Systemen** stellt einen Paradigmenwechsel dar, auf den im nächsten Abschnitt eingegangen wird.

3.2 Steuerungsparadigmen und -architekturen

Die Leistung eines Produktionssystems wird in hohem Maße durch die zugrundeliegende Art der Steuerung des Systems determiniert. Die Auswahl einer adäquaten Steuerungsarchitektur ist daher von hoher Bedeutung und sollte eine Reihe von Faktoren wie die Größe, Komplexität und den Flexibilitätsbedarf des zu steuernden Systems berücksichtigen. Zur Definition des Begriffs Steuerungsarchitektur kann der Begriff der Systemarchitekturen herangezogen werden. Eine **Systemarchitektur** definiert die Komponenten eines Systems, die Anordnung dieser Komponenten in

diesem System sowie die Interaktionen zwischen den Komponenten. Grundsätzlich können drei Typen unterschieden werden: zentrale, dezentrale und hybride Systemarchitekturen [StT17]. In Analogie dazu ist die **Steuerungsarchitektur eines Produktionssystems** eine Menge an (softwarebasierten) Steuerungselementen sowie ihren Beziehungen und Eigenschaften, die notwendig sind um Automatisierungsfunktionen zu realisieren [VGL15].

3.2.1 Zentrale Steuerungsarchitekturen

Dem zentralen Steuerungsparadigma folgend stellt die zentrale Steuerungsarchitektur die einfachste Art der Steuerung dar (vgl. Abbildung 3-1). Sie wird charakterisiert durch einen zentralen Rechner, der alle Planungs- und Informationsverarbeitungsprozesse vornimmt sowie die Aktivitäten des gesamten Produktionssystems bestimmt. Alle anderen Entitäten im System sind passiv und können nicht selbstständig agieren oder Entscheidungen treffen [Unl15b].

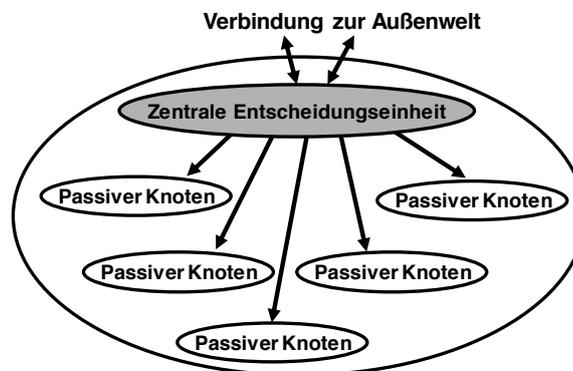


Abbildung 3-1: Zentrale Steuerungsarchitektur i.A.a. [Unl15b]

Durch die zentrale Verfügbarkeit globaler Informationen wird die ganzheitliche Optimierung zentral organisierter Fertigungssysteme vereinfacht. Allerdings neigen zentrale Steuerungsarchitekturen auch zu geringer Reaktionsfähigkeit, Verlässlichkeit, Modifizierbarkeit und Erweiterbarkeit [VaB15]. Das macht sie vor allem für kleine und einfache Systeme mit statischen Aufgaben nützlich. Bei steigender Größe und Komplexität des Produktionssystems nimmt die Reaktionszeit zentraler Steuerungssysteme jedoch drastisch zu. Bei einem Ausfall der zentralen Steuerungseinheit kommt es darüber hinaus zu einem Stillstand des Gesamtsystems. Um dies zu verhindern sollte dieses Steuerungsparadigma daher nur in vergleichsweise einfachen Fällen angewendet werden bzw. in Systemen, die einen solchen Ausfall verkraften können [Unl15b].

3.2.2 Hierarchische Steuerungsarchitekturen

Die Defizite der zentralen Steuerung machten die Entwicklung neuer Ansätze notwendig. Getrieben durch das CIM-Paradigma in den frühen 70er-Jahren wurde der erste Schritt in Richtung von **verteilten Steuerungssystemen** gemacht [Unl15b]. Ein **verteiltes System** kann grundsätzlich definiert werden als ein System, „[...] das auf einer Menge von Rechnern ausgeführt wird, die nicht über einen gemeinsamen Speicher verfügen, und das sich dem Benutzer wie ein einzelner

Rechner darstellt“ [Hei15]. In Rahmen dieser Arbeit soll unter einem **verteilten Steuerungssystem** in Anlehnung an [StT17] eine Menge autonomer Steuerungselemente verstanden werden, die gegenüber ihren Nutzern als einziges, kohärentes System erscheint.

Die Distribution von Entscheidungsproblemen in verteilten Steuerungssystemen ermöglicht es die zentrale Entscheidungsfindung durch einen hierarchisch organisierten Entscheidungsbaum mit unterschiedlichen Entscheidungsebenen zu ersetzen. In einer solchen **hierarchischen Steuerungsarchitektur** (vgl. Abbildung 3-2) übernehmen höhere Entscheidungsebenen die Verantwortung für Entscheidungen mit strategischem Charakter, während die unteren Ebenen auf vergleichsweise einfache, lokale Entscheidungen konzentriert sind [Unl15b]. Es existieren strikte Master-Slave-Beziehungen zwischen den Ebenen [VaB15], d.h. die Entitäten unterschiedlicher Ebenen stehen in einer **asymmetrischen Beziehung** zueinander bezüglich ihrer Entscheidungsrechte, ihres Informationszustands und des Zeitpunktes der Entscheidungsfindung [Mön06].

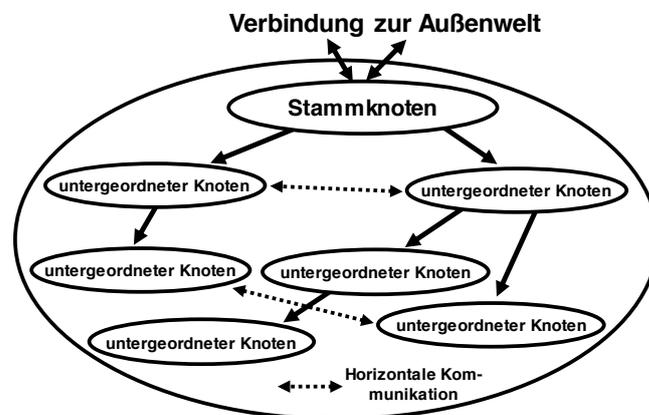


Abbildung 3-2: (Modifizierte) Hierarchische Steuerungsarchitektur i.A.a. [Unl15b]

Durch den Einsatz hierarchischer Steuerungsarchitekturen können auch in großen Systemen oft Lösungen nahe dem Optimum ermittelt und die Robustheit, Vorhersagbarkeit und Effizienz gesteigert werden. Vorteilhaft sind hierarchische Architekturen vor allem in **statischen Umgebungen** mit einer geringen Produktvielfalt, seltenen Produktionsänderungen und sehr wenigen Ausfällen von Systemkomponenten [Unl15b]. Durch Modifizierung der hierarchischen Steuerungsarchitektur kann ferner die horizontale Kommunikation zwischen Entitäten derselben Ebene ermöglicht werden. Dies ermöglicht zwar nicht das horizontale Treffen von Entscheidungen, beschleunigt jedoch die Reaktionszeiten im Vergleich zur herkömmlichen hierarchischen Kommunikation, die eine Eskalation von Entscheidungen über alle Ebenen erfordert [Unl15b].

Aufgrund ihrer zahlreichen Vorteile ist die hierarchische Steuerungsarchitektur **in nahezu allen modernen Produktionssystemen vertreten** und wird durch die **Verteilung auf eine Menge von Steuerungseinheiten** wie SPSen realisiert [Rib15]. Trotz ihrer Popularität ist die hierarchische Steuerungsarchitektur mit einer Reihe von Nachteilen verbunden. Durch die starre Struktur ist es schwer ungeplante Änderungen vorzunehmen und die starke Kopplung zwischen den Mo-

dulen beeinflusst die Modifizierbarkeit, Erweiterbarkeit und Fehlertoleranz des Produktionssystems negativ. Darüber hinaus ist die Reaktionsfähigkeit auf unvorhergesehene Störungen nur begrenzt möglich, da Entitäten der niedrigen Ebenen für die Entscheidungsfindung höhere Ebenen hinzuziehen müssen. Als Konsequenz führt die geringe Kommunikationsgeschwindigkeit dazu, dass globale Entscheidungen oft auf Basis von veralteten Informationen getroffen werden [VaB15]. Da die Steuerungseinheiten über vordefinierte Pfade miteinander verbunden sind, stellt jede Steuerungseinheit im System zudem einen Single Point-of-Failure dar und macht das System somit anfällig gegenüber Ausfällen [Rib15]. Die potenziell größte Schwäche hierarchischer Steuerungsarchitekturen ist jedoch die zugrundeliegende Annahme der **genauen Planbarkeit**, der **deterministischen Abarbeitung** und der **zentralen Steuerung fester, im Voraus bekannter Prozesse** [Bes14]. Diese vereinfachenden Annahmen verlieren aufgrund fast nicht prognostizierbarer Absatzmärkte, sinkender Transparenz von Lieferketten, zunehmender Variantenvielfalt etc. immer mehr ihre Gültigkeit. Folglich ist die schnelle Anpassung an wechselnde Umweltbedingungen erforderlich, sodass Reaktionsfähigkeit, Flexibilität, Wandelbarkeit und ähnliche Eigenschaften an Bedeutung gewinnen. Der damit verbundene Paradigmenwechsel führt in Forschung und Industrie zur zunehmenden Dezentralisierung der Steuerung, z.B. durch dezentrale Verarbeitung von Sensorsignalen und der Gestaltung von leicht wiederverwendbaren, austauschbaren Softwarekomponenten [GCK10].

3.2.3 Dezentrales Steuerungsparadigma

Der Einsatz von dezentralen Steuerungskonzepten ist mit einer Reihe von Vor- und Nachteilen verbunden. So ist die **logistische Leistungsfähigkeit** dezentraler Systeme nicht per se höher als die von zentralen Systemen. Aufwändige Simulationen haben gezeigt, dass die logistischen Leistungsparameter bei der Anwendung beider Ansätze zu ähnlichen Ergebnissen führen oder dass zentral gesteuerte Systeme sogar bessere Ergebnisse erzielen, nämlich dann, wenn globale Optimierung möglich und sinnvoll ist. Ferner unterliegen dezentrale Systeme dem Risiko möglicher **Aufschaukelungseffekte**, die nicht im Voraus beschrieben werden können [Nop10].

Häufig macht es jedoch mehr Sinn **vernünftige Entscheidungen in begrenzter Zeit dezentral zu treffen** als vermeintlich optimale Lösungen über ein zentrales System zu spät [DOS13]. Dies ist insbesondere in Systemen der Fall, wo ein hohes Ausmaß an Volatilität vorherrscht und eine deterministische Planung aufgrund von unvorhersehbaren Ereignissen sowie der geringen Stabilität optimaler Lösungen nicht zielführend ist. Folglich müssen Steuerungssysteme entworfen werden, die mit unerwarteten Situationen umgehen können [VaB15]. Anstatt wie in heutigen Systemen üblich eine fest vorgeschriebene Arbeitsanweisung zu erhalten, werden dezentrale Systeme mit einer Art **Zielvereinbarung** ausgestattet, die jede Entität unter Berücksichtigung der Umgebungseinflüsse zu erreichen hat [DOS13]. Viele Produktions- und Transportaufgaben können auf diese Weise dezentral schneller gelöst werden und benötigen keine Intervention eines zentralen Systems (z.B. kürzester Weg zum Ziel in einer Flughafen-Gepäckförderanlage)

[KrW13]. Weitere Vorteile dezentral organisierter Systeme sind der hohe Grad an Skalierbarkeit, die schnelle Verfügbarkeit neu hinzugefügter Elemente und die Möglichkeit Veränderungen im Gesamtsystem effizient umzusetzen.

Der entscheidende Vorteil von dezentralen Systemen ist jedoch, dass sie im Sinne einer **kollektiven Intelligenz** genutzt werden können, um komplexe aber zerlegbare Probleme selbstständig zu lösen [Bos14]. Voraussetzung dafür ist, dass die einzelnen Elemente des verteilten Systems über eine gewisse Intelligenz verfügen und miteinander kommunizieren können. Dabei erscheint es kontraintuitiv, dass die individuelle Intelligenz der Elemente nicht hoch sein muss, um komplexe Probleme zu lösen und ein nach außen intelligent wirkendes System zu erschaffen. Solche Phänomene treten in der Natur häufig auf. Ein bekanntes Beispiel sind **Ameisenkolonien**, die trotz der geringen Intelligenz der Individuen komplexe Probleme lösen können wie z.B. die Suche nach dem kürzesten Weg zur nächsten Nahrungsquelle. Die in solchen Systemen entstehende Schwarmintelligenz ist auf zwei Konzepte zurückzuführen: Stigmergie und Emergenz [Unl15a].

Stigmergie ist eine Form der indirekten Koordination in dezentralen Systemen und dient der Selbstorganisation [MaO08]. Die Kommunikation erfolgt dabei nicht auf direktem Weg zwischen den Individuen, sondern **indirekt** durch **Veränderung der lokalen Umgebung** [Lew13]. Im Falle von Ameisen werden durch das Zurücklassen von Pheromonen entlang des Weges zu einer Nahrungsquelle Informationen mit anderen Ameisen geteilt. Das Konzept der Selbstorganisation ist auch bei anderen Insekten wie Bienen und Termiten, aber auch bei Tieren wie Fischen, Vögeln und Schafen zu beobachten [Mil10].

Als **Emergenz** wird das Phänomen bezeichnet, dass durch die lokale Interaktion von einfachen Elementen auf niedrigen Ebenen eines Systems ein komplexes Verhalten auf höheren Ebenen entsteht [Woo09]. Emergenz wird oft durch Stigmergie ausgelöst, z.B. führt die Veränderung der Umgebung durch Pheromone zu Folgeaktivitäten anderer Ameisen, die zusammengenommen in einem kohärenten, scheinbar systematischen Verhalten resultieren [Lew13]. Ebenso wird die Funktion des menschlichen Gehirns durch das Zusammenspiel von für sich genommen einfachen Strukturen erreicht. Marvin Minsky stelle in dem KI-Klassiker „**Society of Mind**“ die Theorie auf, dass der menschliche Geist und andere natürlich entstandenen kognitiven Systeme auf einer großen Gemeinschaft von individuell simplen Prozessen basieren [Mil88]. Während individuelle Neuronen über keine Intelligenz verfügen, entsteht durch das Zusammenwirken von Milliarden von ihnen ein komplexes Verhalten, das wir als Intelligenz bezeichnen.

Zur Umsetzung von Steuerungen, die dem dezentralen Paradigma folgen können u.a. heterarchische Steuerungsarchitekturen eingesetzt werden. Es existieren aber zahlreiche weitere Architekturen, die sich z.B. durch den **Grad der Dezentralisierung** voneinander unterscheiden und mit individuellen Stärken und Schwächen verbunden sind.

3.2.4 Heterarchische Steuerungsarchitekturen

Die heterarchische Steuerungsarchitektur stellt die **maximal dezentralisierte Steuerungsform** dar. Sie ist charakterisiert durch eine flache Struktur aus verteilten, lokal autonomen Einheiten, die unmittelbar miteinander kooperieren können (vgl. Abbildung 3-3) [VaB15].

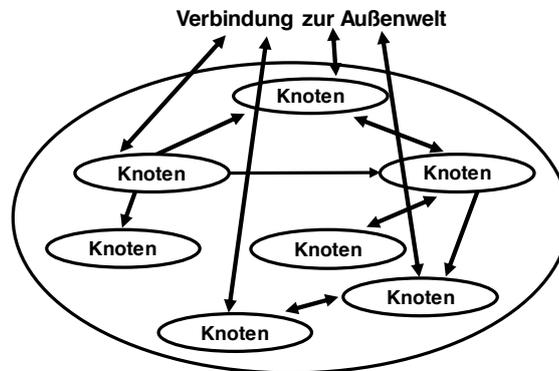


Abbildung 3-3: Heterarchische Steuerungsarchitektur i.A.a. [Unl15b]

Im Gegensatz zur hierarchischen Steuerung, die auf einer strikten Entscheidungsabhängigkeit zwischen hierarchisch organisierten Entscheidungseinheiten beruht, setzt die heterarchische Architektur auf **Zusammenarbeit zwischen den Entitäten**. In einer vollständig heterarchischen Struktur ist die Master-Slave-Beziehung zwischen den Einheiten aufgelöst und jede Einheit erfüllt gleichzeitig beide Rollen. Dadurch verschwinden jegliche Hierarchien und alle Entitäten befinden sich auf der gleichen Ebene [Unl15b]. Globale Informationen sind in dieser Architektur minimal oder fehlen vollständig. Folglich verfügt das System über eine höhere Modularität, reduzierte Koppelung zwischen den Modulen sowie eine höhere Robustheit gegenüber Störungen. Aufgrund der geringen Vorhersagbarkeit des Systemverhaltens ist es allerdings schwierig vordefinierte Produktionspläne exakt einzuhalten und eine globale Optimierung ist nicht möglich [VaB15].

3.2.5 Hybride Steuerungsarchitekturen und Holonic Manufacturing Systems

Die Erkenntnis, dass weder zentrale, noch hierarchische, noch heterarchische Steuerungssysteme den Herausforderungen im zunehmend komplexen Produktionsumfeld gewachsen sind, führte zur Entwicklung von hybriden Steuerungsarchitekturen [VaB15]. Diese erweitern den heterarchischen Steuerungsansatz um hierarchische Elemente und versuchen so die Stärken zentraler und dezentraler Strukturen zu vereinen [Unl15b]. Sie werden daher auch als **semi-heterarchische Architekturen** bezeichnet. Aus diesem Gedanken heraus entstand im Rahmen eines Projektes, das durch das Intelligent Manufacturing Systems (IMS) Consortium in den frühen 90er-Jahren initiiert wurde, das sogenannte Holonic Manufacturing System (HMS) [BoG08]. HMS (dt. *Holonische Produktionssteuerungssysteme*) können durch den rekursiven Aufbau von Holons **hierarchische Elemente** in den Steuerungsprozess einbeziehen und unterscheiden sich dadurch von rein heterarchisch organisierten Steuerungsarchitekturen (vgl. Abbildung 3-4). Sie

verfügen für gewöhnlich über hohe Intelligenz auf höheren Ebenen und niedrigere Intelligenz auf unteren Ebenen [Unl15b].

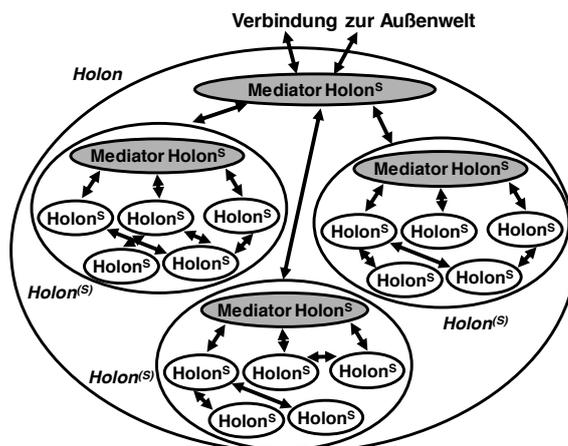


Abbildung 3-4: Holonische Steuerungsarchitektur i.A.a. [Unl15b]

Die Grundlagen des HMS-Konzeptes gehen auf die Pionierarbeit von **Arthur Koestler** zur Modellierung von biologischen und sozialen Systemen in den späten 60er-Jahren zurück. Der Begriff "Holon" ist eine von Koestler eingeführte Verbindung aus dem griechischen Wort „holos“ (gr. *das Ganze*) und dem Suffix „on“ (gr. *der Teil*) und umschreibt den Zustand in dem etwas simultan ein **Ganzes und ein Teil eines anderen Ganzen** sein kann [MMV03]. Im Kontext von Produktionssystemen wird **Holon** daher definiert als:

„[...] autonomer, kooperativer Baustein eines Produktionssystems für die Transformation, den Transport, die Speicherung und/oder Validierung von Informationen und physischen Objekten.“ [VaB15]

Ein Beispiel aus der natürlichen Welt sind Körperzellen, die für sich selbst genommen ein Ganzes darstellen, gleichzeitig aber auch Teil eines anderen Ganzen, nämlich eines Organs, sind. Ein Organ wiederum ist Teil eines Körpers usw. Auf diese Weise entsteht eine Hierarchie von Holons, die als sogenannte **Holarchie** bezeichnet wird. Diese lose Hierarchie unterscheidet sich von herkömmlichen Hierarchien durch folgende drei Eigenschaften [VaB15]:

- Holons können verschiedenen Hierarchien zugehörig sein
- Holons können temporäre Hierarchien bilden und wieder auflösen
- Holons sind nicht von der korrekten Funktion anderer Holons abhängig, um ihre Aufgaben zu erfüllen.

Diese Eigenschaften holonischer Architekturen ermöglichen es zur Laufzeit dynamisch und situationsabhängig zwischen **unterschiedlichen Steuerungsparadigmen** zu wechseln [Mön06]. Dadurch kombiniert die semi-heterarchische Architektur von HMS die Robustheit und das Echtzeitverhalten heterarchischer Systeme mit der Optimierbarkeit hierarchischer Systeme, sodass diese Art der Architektur die **Beste Wahl für viele Fertigungssysteme** zu sein scheint [Unl15b].

Obwohl die holonische Steuerungsarchitektur zahlreiche Stärken aufweisen kann, ist ihre Implementierung mit technischen Hürden verbunden. Auf Basis der beschriebenen Evaluierung der grundlegenden Steuerungsarchitekturen muss daher die Frage geklärt werden wie und mit welchen Technologien funktionsfähige Steuerungssysteme für den Einsatz in realen Fabriken umgesetzt werden können. Die Paradigmen zur softwaretechnischen Umsetzung der beschriebenen Systeme sind folglich Gegenstand des nächsten Abschnitts.

3.3 Software-Design-Paradigmen zur Umsetzung von Industrie 4.0

Für die Realisierung der für Industrie 4.0 notwendigen dezentralen Steuerungsarchitektur kommen zahlreiche Design-Paradigmen für Softwareapplikationen in Frage. Zu den drei am besten erforschten und von der Fachwelt als aussichtsreiche Kandidaten angesehenen Technologien gehören die verteilte Automatisierung nach IEC 61449, Serviceorientierte Architekturen sowie Agentensysteme [FeL10]. Die aktuell am weitesten verbreitete Technologie ist hingegen die Steuerung nach IEC 61131. Die Grundlagen der Technologien sowie ihre Stärken und Schwächen werden im Folgenden diskutiert.

3.3.1 Speicherprogrammierbare Steuerungen nach IEC 61131

Überblick IEC 61131

Die derzeit am weitesten verbreitete Hardware-Plattform der Steuerungstechnik ist die Speicherprogrammierbare Steuerung, kurz SPS. Im klassischen Aufbau besteht eine SPS aus einer CPU sowie zentralen Eingabe- und Ausgabe-Baugruppen [WeZ15]. Die Norm IEC 61131 ist ein weltweit gültiger Programmierstandard für diesen Steuerungstyp [DIN04]. Darin werden Grundbegriffe im SPS-Zusammenhang definiert, Betriebsmittelanforderungen und Prüfungen festgelegt sowie Programmiersprachen spezifiziert, die hierarchische Steuerungsarchitekturen ermöglichen [DIN04, DIN08, DIN14c].

Eigenschaften von IEC 61131

Kernelement der IEC 61131-Architektur sind sogenannte **Funktionsbausteine (FB)** und ein zyklisches Ausführungsmodell mit dem zentrale Tasks ausgeführt werden können [DIN04]. Dies ermöglicht es Programme für industrielle Automatisierungsaufgaben wie z.B. die Drehzahlregelung eines Motors **wiederverwendbar zu kapseln**, sodass sie von Entwicklern jederzeit aufgerufen und verwendet werden können ohne die dahinterliegende interne Funktionsweise im Detail zu kennen oder zu verstehen. Im Programmiersprachenteil IEC 61131-3 bietet der Standard dafür sowohl textuelle als auch grafische Sprachen. Bei der Anweisungsliste werden einzelne Anweisungen listenartig untereinander aufgeführt. Mit strukturiertem Text wird eine Pascal-ähnliche Hochsprache zur Verfügung gestellt, die leistungsfähige Schleifenprogrammierung sowie mathematische Funktionen und Iterationen ermöglicht. Die Ablaufsprache definiert eine Reihe von Schritten und Übergangsbedingungen, die miteinander verbunden werden können, um überge-

ordnete Abläufe zu definieren. Die Funktionsbausteinsprache spiegelt funktionsorientierte logische Ablaufketten wieder, die aus Netzwerken logischer Bausteine besteht. Der Kontaktplan orientiert sich schließlich am Stromlaufplan elektrischer Schaltungen [Fri09]. Eine Übersicht der Programmiersprachen nach IEC 61131-3 ist in Anhang 1 gegeben.

Entwicklungs- und Implementierungsstand von IEC 61131

Der Einsatz von SPSen auf Basis von IEC 61131 ist in der Automatisierungstechnik etabliert und ein Großteil der heutzutage betriebenen Fabriken ist mit Steuerungstechnik ausgestattet, die dem Standard IEC 61131 folgt [LVM09]. Der Standard wurde notwendig, weil aufgrund der marktbeherrschenden Stellung des Marktführers das von einem einzigen Unternehmen definierte Vorgehen zu einem Quasistandard wurde, was die Interoperabilität erschwerte und zu Insellösungen führte [Wec01]. Der Einsatz von objektorientierten Ansätzen mit IEC 61131 kündigt sich zwar an, trotzdem ist die Standardisierung von SPS-Programmen noch nicht vollständig in der Industrie angekommen, sodass die Entwicklung solcher Systeme teilweise hochgradig individuell abläuft [FeL10].

Stärken und Schwächen von IEC 61131

Die Stärken der Speicherprogrammierbaren Steuerung nach IEC 61131 sind vielfältig. Zum einen erlauben sie die Abarbeitung von Programmen mit festen, garantierten Zykluszeiten und decken alle Anforderungen für das Ansteuern mechanischer Komponenten sowie für das Treffen zeitkritischer Entscheidungen im Millisekundenbereich ab. Zum anderen ist der Standard weit verbreitet und in der Automatisierungstechnik etabliert [NeK10].

Eine wesentliche Schwäche des Standards ist jedoch, dass eine Verteilung der Steuerungsarchitektur auf Basis von IEC 61131 letztlich zu **großen monolithischen Softwarestrukturen** führt. Da die Norm herstellerspezifische Sprachelemente zulässt, sind IEC-konforme Anwendungsprogramme nicht zwischen SPSen verschiedener Hersteller austauschbar [Wec01]. Folglich enthalten die Programme meist herstellerspezifische Lösungen, sodass solche Systeme durch eine **geringe Interoperabilität** gekennzeichnet sind [ZoS15]. Diese Einschränkungen machen den Standard zunehmend inkompatibel mit der steigenden Komplexität von Produktionssystemen sowie der aus den Marktbedingungen resultierenden Anforderungen an Flexibilität und Robustheit. Die Sprachen von IEC 61131 sind ferner nicht dafür vorgesehen dezentrale Steuerungsarchitekturen zu entwickeln und zu betreiben, was eine zusätzliche Herausforderung für Entwickler darstellt. Um dies zu umgehen kann z.B. die Umsetzung der steuerungstechnischen Ebene in Hochsprachen stattfinden. Dies ist jedoch mit hohem Kommunikationsaufwand verbunden und kann zudem einen Single Point of Failure in das System einbringen, wenn z.B. ein PC die Laufzeitumgebung für Agenten stellt. Die Verarbeitung und Interpretation komplexer Daten wie z.B. die Interpretation von XML-codierten Nachrichten ist mit großem Aufwand verbunden und die Implementierung von Datenbanken und Algorithmen stößt zudem schnell an ihre Grenzen [NeK10].

Fazit zu IEC 61131

Speicherprogrammierbare Steuerungen nach dem IEC 61131-Standard sind weit verbreitet und haben den heutigen Grad an Automatisierung in der Fertigungsindustrie erst ermöglicht. Sie sind in der Lage komplexe hierarchische Steuerungssysteme abzubilden und erlauben eine zuverlässige Abarbeitung vordefinierter Fertigungsschritte. Es ist jedoch fraglich, ob der Standard der zunehmenden Komplexität zukünftiger Produktionssteuerungssysteme und der damit einhergehenden Anforderungen an Flexibilität, Agilität und Robustheit gewachsen ist. Da die Realisierung dieser Anforderungen auf Basis von IEC 61131 mit einem erheblichen Zusatzaufwand verbunden wäre, werden im Folgenden drei Ansätze diskutiert, die im Kontext der Dezentralisierung und Verteilung entstanden sind und sich potenziell besser zur Erreichung der gesteckten Ziele eignen.

3.3.2 Verteilte Automatisierung nach IEC 61499

Überblick IEC 61499

Aus den Herausforderungen der Speicherprogrammierbaren Steuerung nach IEC 61131 heraus entstand der Standard zur Verteilten Automatisierung nach IEC 61499. Aufgrund seiner zahlreichen Vorteile wird der neue Standard als potenzieller **Nachfolger** von IEC 61131 gehandelt [LVM09]. Im Gegensatz zu SOA und Agentensystemen entstammt der IEC 61499-Standard nicht der Softwaretechnik, sondern der klassischen **Steuerungs- und Automatisierungstechnik** und wurde durch die Arbeitsgruppe TC65/WG6 des IEC etabliert. Hinter dem Standard verbirgt sich die Zielsetzung die **Agilität von Automatisierungssystemen** zu steigern, indem ermöglicht wird Steuerungsalgorithmen aufwandsarm wiederzuverwenden und schnell zu rekonfigurieren [FeL10].

Eigenschaften von IEC 61499

Grundlage des Standards IEC 61499 ist eine generische **Steuerungsarchitektur** zur Realisierung verteilter Automatisierungssysteme. Diese wird ergänzt durch acht unterschiedliche **Referenzmodelle** sowie eine **Entwicklungsmethodik** für das Engineering [DIN14b]. In IEC 61499 wird das aus IEC 61131 bekannte Konzept der Funktionsbausteine so erweitert, dass verteilte Steuerungssysteme mithilfe von **logisch verbundenen Funktionsblöcken** entwickelt werden können, die auf **unterschiedlichen Ressourcen** ablaufen. Dies ermöglicht es das **Systemdesign anwendungszentriert** zu gestalten, sodass Applikationen zunächst für ein Gesamtsystem entwickelt und anschließend auf vorhandene Geräte im System verteilt werden können [ZoL14]. Ein bedeutender Unterschied im Vergleich zu IEC 61131 liegt zudem im Ausführungsmodell, das den beiden Standards jeweils zugrunde liegt. Während IEC 61131 auf einem **zyklischen Ausführungsmodell** basiert, wird in IEC 61499 ein **ereignisbasiertes Ausführungsmodell** eingeführt. Dies ermöglicht es Ereignisse als Mittel zur Koordination der Abläufe im System zu verwenden, anstelle der bisher bekannten zentralen Tasks in IEC 61131 [FeL10]. Damit die Ereignisse zwischen den FBs hin- und her kommuniziert werden können, verfügen FBs über zwei Arten von Ein- und Ausgängen: die üblichen Ein- und Ausgänge für den **Datenaustausch** sowie zusätzliche

Ein- und Ausgänge für die Kommunikation von **Ereignissen** [DIN14b]. Die im Rahmen von IEC 61499 zur Verfügung gestellten Referenzmodelle liefern wertvolle Werkzeuge zur Umsetzung des Standards. Das grundlegendste Modell ist das **Funktionsbausteinmodell**, welches folgende drei Arten von FBs zur Verfügung stellt [DIN14b]:

- **Basisfunktionsbausteine:** Funktionsbausteine, die nicht in weitere Funktionsbausteine zerlegt werden können und über einen Plan zur Ausführungssteuerung ihrer Algorithmen verfügen (vgl. Abbildung 3-5, links).
- **Zusammengesetzte Funktionsbausteine:** Funktionsbausteine, die sich aus anderen Funktionsbausteinen zusammensetzen. Ihre Algorithmen und die Bearbeitungssteuerungen werden vollständig durch FB-Netzwerke beschrieben (vgl. Abbildung 3-5, rechts).
- **Dienstschnittstellen-Funktionsbausteine:** Funktionsbausteine, die anderen Anwendungen Dienste zur Verfügung stellen, die auf einer Abbildung der Telegramme auf die Ereignis- und Daten-Ein- und Ausgänge basieren. Sie verfügen über die gleichen Außenschnittstellen wie Basisfunktionsbausteine.

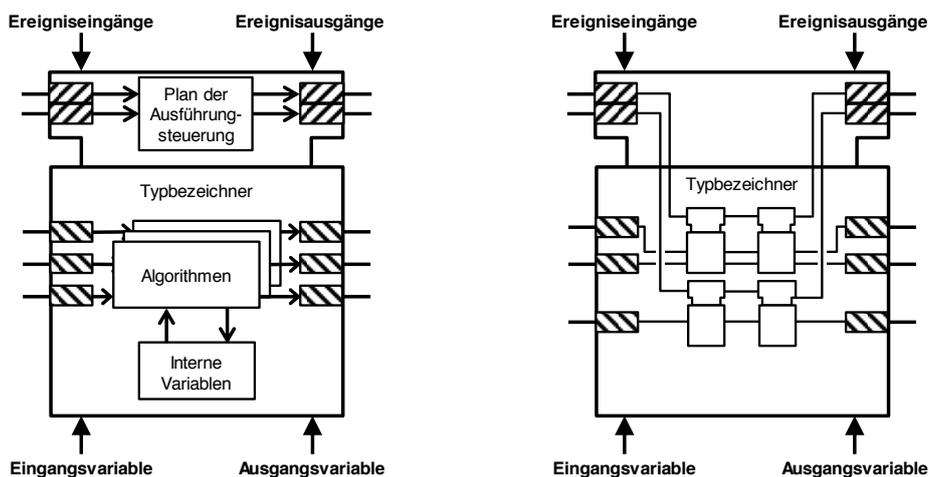


Abbildung 3-5: Basis-FB (links) und zusammengesetzter FB (rechts) i.A.a. [DIN14b]

Die drei grundsätzlichen Funktionsbausteintypen werden durch **Adapterschnittstellen** und **Unteranwendungen** ergänzt. Während erstere dafür verantwortlich sind mehrere Event- und Datenverbindungen zu bündeln, fassen letztere miteinander verbundene Teil-FB und Teilunteranwendungen zusammen [DIN14b].

Anwendungen werden durch das Verbinden von FBs in einem FB-Netzwerk definiert. Während die Knoten des Netzwerks aus FBs oder Unteranwendungen und deren Parametern bestehen, sind die Kanten durch Daten- oder Ereignisverbindungen definiert. Anwendungen fließen in das **Anwendungsmodell** ein, das den Ereignis- und Datenfluss zwischen Instanzen von Funktions-

bausteinen und Unteranwendungen beschreibt. Im **Ressourcenmodell** werden dann eigenständige, in Ressourcen⁸ enthaltene Funktionen verwaltet. Sie fließen wiederum in das **Gerätemodell** ein, das die Ressourcen von Geräten⁹ bündelt und deren Kommunikations- und Prozessschnittstellen verwaltet (vgl. Abbildung 3-6).

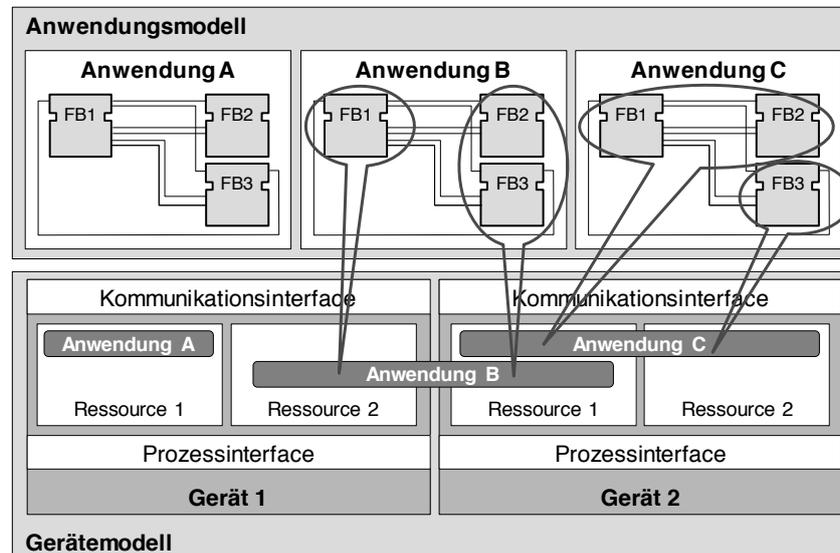


Abbildung 3-6: IEC 61499 Anwendungs- und Gerätemodell i.A.a. [For15]

Im **Systemmodell** werden die Anwendungen sowie die Topologie des gesamten Systems abgebildet. Über das **Verteilungsmodell** kann die Verteilung von Applikationen auf mehrere Geräte gesteuert werden und mithilfe des **Managementmodells** wird die Verwaltung der Anwendungen ermöglicht [DIN14b]. Mithilfe von Softwarewerkzeugschnittstellen und Managementkommandos können diese Anwendungen auch online rekonfiguriert werden [Zoi09]. Im **Betriebszustandsmodell** wird schließlich der Systemlebenszyklus aller im System enthaltenen funktionalen Einheiten wie z.B. Geräten, Ressourcen und Anwendungen mit den Phasen Entwurf, Inbetriebnahme, Betrieb sowie Instandhaltung festgehalten [DIN14b].

Entwicklungs- und Implementierungsstand IEC 61499

Seit der erstmaligen Veröffentlichung des Standards 2006 sind zahlreiche **Forschungs- und Industrieprojekte** entstanden, die das Steuerungsparadigma der Verteilten Automatisierung nach IEC 61499 umsetzen. Ein frühes Forschungsprojekt, das bereits parallel zur Entwicklung des Standards im Jahr 2004 begann, trägt den Titel „µCrons“. Im Rahmen des Projektes wurde das Ziel verfolgt mithilfe von agilen Systemkomponenten die Flexibilität der Produktion zu erhöhen, um die Individualisierung von Produkten „bis [zur] Losgröße 1“ zu erreichen [Pic16]. Das Projekt resultierte in der Veröffentlichung einer IEC 61499-basierten **Laufzeitumgebung** als Open Source Projekt (<http://www.fordiac.org>). Weitere Anwendungen, die ähnliche Ziele verfolgen, sind

⁸ Eine Ressource wird definiert als „[...] funktionale Einheit [...], welche eine eigenständige Kontrolle über ihre Funktion besitzt, die in einem Gerät enthalten ist.“ [DIN14b].

⁹ Ein Gerät wird definiert als „[...] unabhängige physische Einheit, die fähig ist, eine oder mehrere spezifische Funktionen in einem speziellen Kontext auszuführen und die durch ihre Schnittstellen abgegrenzt ist“ [DIN14b].

u.a. das von der Siemens AG ins Leben gerufene Profinet CBA (<http://www.automation.siemens.com/cba>), Isagraf v5.0 (<http://www.isagraf.com>), nxccontrol (<http://www.nxccontrol.com>) und das EnAS-Projekt (Energieautarke Aktoren und Sensoren, <http://www.energieautark.com>). Eines der aktuellsten Beispiele für die Umsetzung von IEC 61499 ist das Forschungsprojekt „Cloud-Based Industrial Control Services“ (CICS). Hier wird durch die Entwicklung eines **cloudbasierten Steuergeräts** versucht klassische SPSen für Anwendungen mit nicht echtzeitrelevanten Reaktionszeiten zu ersetzen, um eine Verteilung und Vernetzung der Steuerungselemente im Sinne von Industrie 4.0 zu erreichen [LaS18]. Für eine umfassendere Übersicht sei auf die Werke [ZoL14] und [ZoS15] verwiesen.

Des Weiteren existieren auch **Modellierungswerkzeuge**, die für den Entwurf von IEC 61499-konformen verteilten Automatisierungssystemen entwickelt wurden und den Entwurfsprozess unterstützen. Dazu gehört das auf Eclipse basierende Open-Source-Framework 4DIAC (www.fordiac.org), ebenso wie das von Rockwell Automation entworfene IEC 61499-basierte Function Block Development Kit (FBDK) (www.holobloc.com) sowie Corfu ESS / Archimedes (seg.ee.upatras.gr/corfu).

Stärken und Schwächen von IEC 61499

Aus den Erfahrungen in Forschungs- und Industrieprojekten lassen sich zahlreiche Stärken und Schwächen von IEC 61499 ableiten. Auf Basis der Verteilten Automatisierung nach IEC 61499 können Automatisierungssysteme realisiert werden, die sich durch mehrere **Vorteile** von konventionellen Automatisierungssystemen abheben. Zum einen entfällt durch das ereignisbasierte Ausführungsmodell der häufige **Leerlauf** der zyklischen Programmbearbeitung, da Bausteine nur aufgerufen werden, wenn entsprechende Ereignisse dies erforderlich machen [SPS14]. Zum anderen ermöglicht der Standard eine **plattformunabhängige, komponentenbasierte Architektur** und begünstigt damit die gegenseitige Vernetzung zwischen Produktionsprozessen. Dies führt in Summe zu einer deutlichen Steigerung der **Interoperabilität** von Komponenten und der **Zuverlässigkeit** von Automatisierungssystemen [Jak17]. Da die Funktionsblöcke zudem nicht in einer sequentiellen Abfolge aktiviert werden müssen, sondern durch Ereignisse aufgerufen werden, wird gleichzeitig auch die **Flexibilität** erhöht [ZoL14]. Ferner wird dank des automatisierungstechnischen Hintergrunds des Standards die **Echtzeitfähigkeit** von Steuerungssystemen gewährleistet, sodass sich der Ansatz sehr gut für Automatisierungssysteme eignet, in denen die Steuerungs- und Entscheidungslogik dezentralisiert und über mehrere (semi-)autonome Hardware- und Software-Einheiten verteilt ist [Unl15b, Jak17].

Gleichzeitig ist der IEC 61499-Standard auch mit einer Reihe von **Nachteilen** verbunden, die zur Folge haben, dass der Standard bis heute eine geringe Verbreitung in der Industrie und eine vergleichsweise kleine Gemeinschaft von Entwicklern aufweist [Jak17]. So fehlt es an methodischer Unterstützung bei der **Konfiguration von Kommunikationsabläufen** sowie bei der **Re-**

konfiguration von Steuerungsanwendungen [SMS07]. Durch das Fehlen einer **formalen Semantik** zur Beschreibung von Steuerungsanwendungen werden zudem die **Validierungs- und Überprüfungsprozesse** bei dem Entwurf von Steuerungssystemen erschwert. Ein Ansatz zur Lösung dieser Schwierigkeiten könnte die Anwendung formaler Modelle und Ansätze aus der Softwareentwicklung sein [IBF09]. Eine **einheitliche Entwicklungsmethode** für IEC 61499-basierte Steuerungssysteme und eine formale Semantik zur Beschreibung von Steuerungsanwendungen auf Basis von Erkenntnissen aus der Softwareentwicklung könnten helfen den Standard weiter durchzusetzen [AnB11]. Zudem sei angemerkt, dass ein effizientes Management heterogener Systeme, bestehend aus einer Vielzahl von Entitäten in hochdynamischen, teilweise unvorhersehbaren Umgebungen wie sie in der Fertigungsindustrie vorzufinden sind, ein **höheres Maß an Intelligenz** erfordert als die in IEC 61499 verwendeten Funktionsblöcke dies derzeit zulassen [MoV15]. Während die Steuerung niedriger Ebenen über gewöhnliche Steuerungsmodule auf SPSen realisiert werden kann, erfordert eine effiziente Lösung hochgradig komplexer Planungs- und Steuerungsaufgaben wie sie auf MES- und ERP-Ebenen üblicherweise existieren, die Verwendung von **Hochsprachen** wie C++ oder Java [Kad15]. Da der IEC 61499-Standard aus der automatisierungstechnischen Perspektive entwickelt wurde, liegen seine Stärken naturgemäß in der Dezentralisierung der unteren Ebenen der Automatisierungspyramide. Rechenintensive Aufgaben wie sie auf den höheren Ebenen erforderlich sind, können jedoch mehr Zeit benötigen als dies die festen Scan-Zyklen von SPSen zulassen. Jede Berechnung, die mehr Zeit als die Scan-Zykluszeit in Anspruch nimmt, kann zu Fehlern und unerwünschtem Verhalten führen [DPD15]. Eine große Herausforderung bei der Realisierung der verteilten Automatisierung nach IEC 61499 stellt demnach **die Synchronisation und Integration der hohen und niedrigen Ebenen** der Automatisierungspyramide dar. Um diese Schwäche auszugleichen gibt es Bestrebungen den IEC 61499-Standard mit anderen Ansätzen wie intelligenten Software-Agenten, Serviceorientierten Architekturen und Semantic Web Technologien verbinden [DPD15, MoV15, ZoS15].

Fazit zu IEC 61499

Aufgrund der inhärenten Unterschiede erscheint eine Trennung zwischen Anwendungen mit und ohne **Echtzeitanforderungen** bei gleichzeitiger Definition einfacher, universeller Programmierschnittstellen sinnvoll. Auf diese Weise könnten die **Vorteile** des Ansatzes der Verteilten Automatisierung nach IEC 61499 bei der Dezentralisierung der unteren Ebenen der Automatisierungspyramide mit den Vorteilen anderer Ansätze auf den höheren Ebenen **kombiniert** werden.

Ein Ansatz, der sich stärker mit den höheren Ebenen der Automatisierungspyramide auseinandersetzt sind Serviceorientierte Architekturen. Sie entspringen im Gegensatz zu IEC 61499 dem Kontext der Softwaretechnik und sind Gegenstand der Betrachtungen des nächsten Abschnitts.

3.3.3 Serviceorientierte Architekturen

Überblick SOA

Das Paradigma der Serviceorientierten Architekturen (SOA) ist ein Konzept zur Entwicklung verteilter Systeme, das ursprünglich für Geschäftssysteme und den elektronischen Geschäftsverkehr entworfen wurde [CKM15]. SOA ist aus dem Bedürfnis entstanden immer komplexere, monolithische Unternehmenssoftwarearchitekturen durch flexible, verteilte Anwendungslandschaften zu ersetzen, deren Elemente mit geringerem Aufwand angepasst und erweitert werden können [FeL10]. Um dies zu erreichen werden einzelne Funktionalitäten bzw. die Steuerungslogik ähnlich wie bei Funktionsblöcken nach IEC 61499 in sogenannte Services wiederverwendbar gekapselt. SOA unterscheidet sich dadurch, dass es keine abstrakten Modelle zur Beschreibung der Systemtopologie erfordert. Stattdessen werden die Services über lose Koppelung miteinander verbunden und orchestriert, sodass komplexe Geschäftsprozesse modelliert werden können.

Eigenschaften von SOA

Der grundlegende Aufbau von Serviceorientierten Architekturen besteht aus drei Rollen, von denen jeder Teilnehmer einer SOA eine oder mehrere ausfüllen kann. Diese Rollen sind: Dienstanbieter, Dienstanutzer und Dienstvermittler (vgl. Abbildung 3-7).

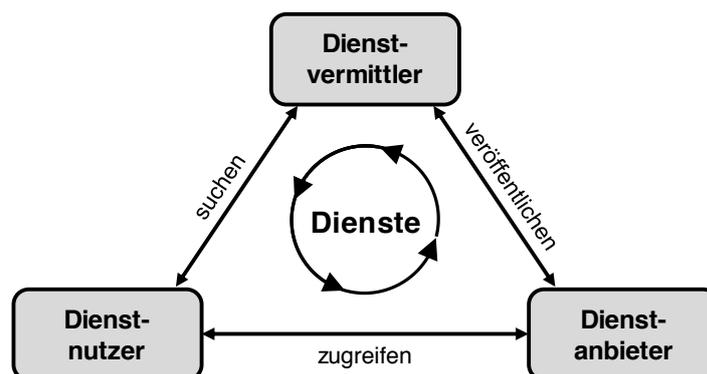


Abbildung 3-7: Grundlegender Aufbau Serviceorientierter Architekturen i.A.a. [Mab08]

Die Interaktion zwischen den drei Parteien laufen dabei nach folgendem Muster ab [Zhu05]:

- *veröffentlichen* (engl. *publish/register*): Der Diensteanbieter veröffentlicht seinen Dienst in einem globalen Dienstverzeichnis
- *suchen* (engl. *find*): Der Dienstanutzer fragt beim Dienstvermittler einen Service an, den der Dienstvermittler in einer Liste aus verfügbaren Services sucht.
- *binden* (engl. *bind*): Wird ein adäquater Service gefunden erfolgt die Mitteilung der (Netzwerk-) Adresse des Dienstanbieters an den Dienstanutzer und der Funktionsaufruf wird an diese Adresse gebunden.
- *ausführen* (engl. *execute*): Der Service wird aufgerufen und erhält Eingangsparameter. Als Antwort werden die geforderten Ausgabeparameter zurückgesendet.

Auf Basis dieser einfachen Konstellation können sehr komplexe Diensthierarchien aufgebaut werden, in denen Dienste miteinander verbunden werden um Aufgaben bedarfsgerecht zu lösen. Eine einfache Diensthierarchie ist in Abbildung 3-8 dargestellt.

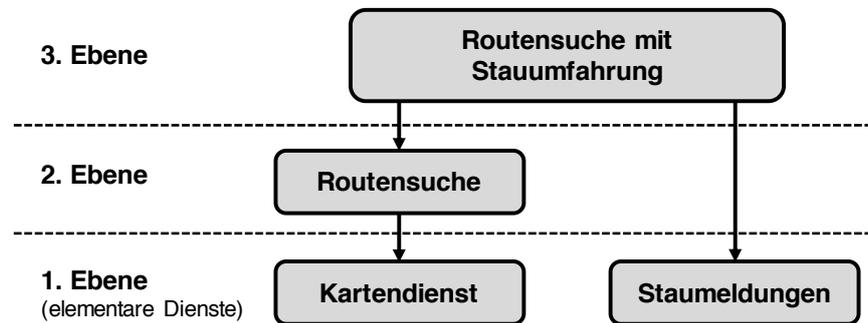


Abbildung 3-8: Einfache SOA-Diensthierarchie i.A.a. [FeL10]

Hier bedient sich ein Dienstanwender auf der obersten Ebene zweier Dienste auf niedrigeren Ebenen (Routensuche und Staumeldung). Während der Staumeldungsdienst ein elementarer Dienst ist, der für sich selbst funktioniert, ist im Routensuchdienst lediglich der Algorithmus zur Berechnung der besten Strecke enthalten. Daher muss der Routensuchdienst selbst auf einen weiteren Dienst, den Kartendienst, zurückgreifen. Die so entstehenden Diensthierarchien können bedarfsgerecht konfiguriert und ebenso schnell wieder aufgelöst werden.

Für die technische Realisierung von SOA können verschiedene Technologien wie z.B. CORBA, Microsoft's COM/DCOM oder Java RMI verwendet werden. Im Automatisierungsumfeld haben Web Services (WS) großen Zuspruch erhalten, da sie eine Vielzahl von Spezifikationen und Standards für die Umsetzung unterschiedlichster Aufgaben enthalten. Als Grundlage für die Kommunikation von WS dient das Simple Object Access Protocol (SOAP), das zum Informationsaustausch in verteilten Systemen konzipiert wurde. Zusammen mit dem Devices Profile for Web Services (DPWS) und einem DPWS-Gateway können so industrielle serviceorientierte Automatisierungssysteme umgesetzt werden, die SPSen, Roboter und fortschrittliche drahtlose Sensoren einbinden [FeL10, CKM15].

Eigenschaften von Microservices

Microservices stellen eine Weiterentwicklung bzw. eine Untermenge der serviceorientierten Architekturen dar. Nach Fowler ist die Microservice-Architektur ein Ansatz zur Entwicklung einer **einzigsten Anwendung** als ein **Paket kleiner Services**, von denen jeder als eigener Prozess abläuft und mithilfe eines leichtgewichtigen Mechanismus, oft eine HTTP-basierte API, mit anderen Systemteilnehmern kommuniziert. Diese Services sind um Geschäftskompetenzen herum gebaut und unabhängig voneinander durch vollautomatisierte Deployment-Mechanismen bereitstellbar. Die Services selbst können dabei in unterschiedlichen Programmiersprachen umgesetzt werden und auf verschiedene Datensicherungsmethoden zurückgreifen [Fow14]. Die Intelligenz wird dabei in die Services integriert anstatt wie in konventionellen Systemen in die Verbindungen zwischen den Services („smart endpoints and dumb pipes“) [Fow15]. Durch die Erstellung multipler

Instanzen von Services werden Single Points of Failure vermieden. Da diese in der Cloud erstellt werden können, ist eine Skalierung insbesondere bei sogenannten *stateless services* (zustandslose Dienste ohne Cache und Datenspeicherung) sehr einfach realisierbar. Mit Umwegen lassen sich so auch sogenannte *stateful services* (zustandsabhängige Dienste) bedarfsgerecht skalieren. Darüber hinaus wird das Deployment komplexer Applikationen auf viele unterschiedliche Geräte vereinfacht [Eva16].

Obwohl Microservices teilweise als selbstständiger Architekturstil angesehen werden, können sie auch als Weiterentwicklung und Teilmenge von SOA angesehen werden. Während SOA vor allem für die Integration großer IT-Landschaften eingesetzt werden, fokussieren Microservices auf die Umsetzungen **komplexer Einzelanwendungen** und verzichten dabei auf die Verwendung eines Enterprise Service Bus. Darüber hinaus kommen in Microservices üblicherweise Webtechnologien und einfache, leichtgewichtige Kommunikationsprotokolle wie HTTP zum Einsatz [Fow14, New15].

Entwicklungs- und Implementierungsstand von SOA

Insbesondere in den 2000er-Jahren wurden zahlreiche SOAs im Rahmen von Forschungs- und Industrieprojekten implementiert. Aufgrund der Ausrichtung auf die **Integration von IT-Landschaften** sind Anwendungsfälle zwar häufig bei fertigen Unternehmen vorzufinden, die meisten Anwendungen beziehen sich jedoch nicht auf die Automatisierung selbst. SOA-Plattformen wie Oracle SOA Suite 11g und IBM WebSphere haben zur Verbreitung des SOA-Paradigmas beigetragen und Anwendungsfälle hervorgebracht [Cri11]. IBM hat das Konzept u.a. an einem Standort zur Halbleiterproduktion umgesetzt als im Rahmen einer Kooperation mit **externen Lieferanten** die Einbindung von IBM-interner Steuerungssoftware auf MES-Ebene notwendig wurde. Zum Schutz des geistigen Eigentums konnten die IBM-internen Funktionen mithilfe von Services gekapselt und den Lieferanten zur Verfügung gestellt werden. Für die Realisierung des Projektes wurde der SOAP-Kommunikationsstandard verwendet [MES08]. Vor dem Hintergrund der effektiven **Integration der IT-Landschaften einzelner Unternehmen** in die gegenseitigen Wertschöpfungsketten ist das Paradigma der SOA auch in der Automobilindustrie auf großes Interesse gestoßen. OEMs wie z.B. Daimler, Audi und Volkswagen haben das SOA-Paradigma umgesetzt, um noch tiefer als bislang in die Prozesse ihrer Lieferanten eingebunden zu werden [Erk11]. Ford hat im Rahmen des eHub-Projektes die SOA zur Sicherstellung der Interoperabilität von Shopfloor-Anwendungen innerhalb und außerhalb des Unternehmens umgesetzt. Auch hier kamen SOAP, XML und andere Web Service Standards zum Einsatz [MES08].

Während in diesen Projekten ein starker Fokus auf die Interoperabilität von Unternehmensanwendungen gelegt wurde, existieren auch Projekte, die Serviceorientierte Architekturen für industrielle **Automatisierungssysteme** anwenden. So wurde im Rahmen des europäischen Forschungsprojektes **SOCRADES** eine SOA mit Web Services in einem dynamischen, modular aufgebauten Montagesystem implementiert, die Funktionen für die Montage, Qualitätssicherung,

Funktionsprüfung, Reparatur und Verpackung beinhaltet [CKM15]. Auch dieses Projekt basiert auf dem SOAP-Protokoll und dem DPWS-Standard und verbindet die höheren Ebenen mithilfe eines Mediators mit dem PROFIBUS-System [BDR09]. Eine aktuellere Implementierung von SOA in Automatisierungsumfeld wurde im **SkillPro**-Projekt (www.skillpro-project.eu) realisiert. In dem Projekt werden die funktionalen Eigenschaften von Industrie 4.0-Komponenten auf Ebene der Fertigungsprozesse betrachtet. Damit wird das Ziel verfolgt die inhärente Flexibilität und Rekonfigurierbarkeit heutiger Produktionssystemkomponenten wie Werkzeugmaschinen, Handling- und Logistiksystemen durch ein ganzheitliches serviceorientiertes Framework nutzbar zu machen [PSA14]. Eine mögliche Realisierung der RAMI 4.0-Verwaltungsschale mithilfe von SOA ist in [UsE15] beschrieben. Eine exemplarische Übertragung wurde im Projekt SmartFactoryKL verwirklicht [MKW17].

Trotz der erfolgreichen Umsetzung von SOA in zahlreichen Projekten ist anzumerken, dass die Bedeutung des SOA-Paradigmas nach 2009 deutlich abgenommen hat. Dies liegt einerseits daran, dass SOA durch die zunehmende Adaption unternehmensseitiger Technologievorgaben, einschränkender Standards und zentraler Führungsstrukturen ihren **leichtgewichtigen Charakter verloren** haben [McL16]. Andererseits zeichnet sich durch die zunehmende Komplexität von Steuerungsaufgaben eine Transition von synchronen zu **asynchronen Systemen** in der Produktion ab. Vor diesem Hintergrund haben sich in der Zwischenzeit **internetbasierte Konzepte** gegenüber den in SOA angewendeten Standards durchgesetzt. So wurde das Kommunikationsprotokoll SOAP durch das organisch gewachsene Konzept der RESTful¹⁰ APIs und OPC Unified Architecture (OPC UA) als Standard abgelöst [CKM15, McL16]. Es ist zu erwarten, dass ähnliche internetbasierte Technologien und Konzepte zunehmend Einzug in die Automatisierung erhalten werden. Des Weiteren zeichnet sich ab, dass Microservices verstärkt an die Stelle von SOA treten und vor allem bei sehr großen, komplexen Applikationen erfolgreich eingesetzt werden. Ein Repräsentant für eine solche Implementierung von Microservices ist der Streamingdienst Netflix. Die Anwendung des Unternehmens verfügt über mehr als hundert separate Services, die in drei weltweit verteilten Regionen gehostet werden und zuverlässig weltweit über 87 Mio. Kunden bedienen [Eva16].

Stärken und Schwächen von SOA

Die Stärken von SOA liegen erwartungsgemäß in den grundlegenden Eigenschaften des Ansatzes – der Kapselung und der losen Kopplung von Services. Dies ermöglicht einerseits die unkomplizierte **Skalierbarkeit** und reduziert andererseits Kosten für Folgeprojekte, da Komponenten **wiederverwendet** werden können [MES08]. Dank der eindeutig definierten Schnittstellen sind die Dienste darüber hinaus **unabhängig von Betriebssystemen und Programmiersprachen**. Sie können daher in den Programmiersprachen entworfen werden, die sich für die Lösung

¹⁰ REpresentational State Transfer

des Problems und die Implementierung der jeweiligen Funktion am besten eignen [FeL10]. Wenn die Services in **getrennten Prozessen** ausgeführt werden, kann darüber hinaus bei Fehlfunktionen eines Dienstes eine Kaskadierung des Fehlers vermieden und ein systemweiter Ausfall verhindert werden [Fow14]. Durch den Einsatz von Webtechnologien ist zudem die Ausführbarkeit über das Internet sowie eine hohe Leistung mit wenigen Kompatibilitätseinschränkungen gewährleistet. Diese Eigenschaften haben SOA zum bevorzugten Ansatz für die Integration von Anwendungen in Unternehmen gemacht [CKM15].

Auf der anderen Seite führen die Stärken des Ansatzes zu Schwächen in anderen Bereichen. So liegt der Fokus von SOA auf der Verfügbarmachung von Diensten über universelle Schnittstellen, nicht jedoch auf der Beschreibung der **Ausführungsdetails**, die für Fertigungsprozesse entscheidend sind. Dienste können ferner zwar miteinander kommunizieren, sie verfügen jedoch nicht über **soziale Fähigkeiten**, um z.B. dynamisch Verträge miteinander auszuhandeln. Die Rekonfiguration des Systems ist zudem oft mit einer Umprogrammierung von Services verbunden [CKM15]. Ferner werden Services nur aufgerufen, wenn ein Ereignis dies erforderlich macht. Sie sind jedoch nicht in der Lage selbstständig und proaktiv Handlungen auszuführen und die Initiative zu ergreifen, wenn sie die Umfeldbedingungen verändern. Das resultierende rein **reaktive Verhalten** von SOA ist für PPS-Aufgaben im turbulenten Fertigungsumfeld nicht optimal [LüF15].

Fazit zu SOA

Das Konzept der Serviceorientierung ist ein aufgrund der zunehmenden Komplexität von Unternehmenssoftware notwendig gewordenes Paradigma. SOAs werden daher vor allem zur Sicherstellung der Interoperabilität von internen und externen IT-Landschaften verwendet, wo sie ihre Stärken ausspielen können. Für die Realisierung komplexer Anwendungen, die robust und leicht anpassbar sind, scheinen hingegen Microservices zukünftig der richtige Weg zu sein. Im Kontext der Automatisierung fehlen SOA darüber hinaus derzeit wesentliche Elemente wie soziale Fähigkeiten und Proaktivität, die für die Realisierung von Industrie 4.0 gefordert werden.

Ein Ansatz, der über diese und weitere Fähigkeiten verfügt ist das Paradigma der Agentensysteme. Sie können für sich genommen oder gemeinsam mit SOA implementiert werden und werden im nächsten Abschnitt behandelt.

3.3.4 Agentensysteme

Überblick Agentensysteme

Jahrzehntelange Erfahrungen aus der Entwicklung großer Softwareanwendungen haben zur Erkenntnis geführt, dass **Interaktion** die für sich genommen vermutlich wichtigste Charakteristik komplexer Softwaresysteme darstellt. Die **Agentenorientierte Programmierung (AOP)** bzw. **Agentenorientierte Software Entwicklung (AOSE)** ist ein Paradigma, das diesen Erkenntnis Rechnung trägt [Woo02]. AOP kann als eine Spezialisierung der Objektorientierten Programmie-

rung (OOP) gesehen werden [Sho90] und sie gehört zu den Verfahren der **Verteilten Künstlichen Intelligenz (VKI)** [Mön06]. Unter Agenten werden je nach Domäne und Autor teilweise unterschiedliche Dinge verstanden [Sho90, BJW01, MVK06, GiH07, BCG08]. Eine oft verwendete Definition ist auf Wooldridge zurückzuführen, der **Agenten** definiert als:

„[...] Computersystem, das in seiner Umwelt eingebettet ist und in dieser Umwelt zu autonomen Handlungen fähig ist, um seine Zielvorgaben zu erreichen.“ [Woo02]

Damit wird unterstrichen, dass Agenten selbstständig in der Lage sind Entscheidungen zu treffen – im Gegensatz zu konventionellen Softwareprogrammen, die angeleitet werden müssen [Klü14]. Diese Definition soll für die Verwendung in der vorliegenden Arbeit um die Definition aus dem Automatisierungstechnischen Bereich nach der VDI/VDE-Norm 2653-1 ergänzt werden:

„Ein [(technischer)] Agent ist eine abgrenzbare (Hardware- oder/und Software-) Einheit mit definierten Zielen. Ein Agent ist bestrebt, diese Ziele durch selbstständiges Verhalten zu erreichen und interagiert dabei mit seiner Umgebung und anderen Agenten. Agenten sind ein Modellierungskonzept zur Lösung von technischen Aufgabenstellungen unabhängig von einer bestimmten Realisierungsform.“ [VDI10]

Unter einem Agentensystem (engl. *Multi-Agent System* bzw. MAS)¹¹ wird folglich ein System verstanden, in dem mehrere Agenten miteinander interagieren, um individuelle oder gemeinsame Ziele zu erreichen. Dies ist insbesondere dann notwendig, wenn das Erreichen dieser Ziele außerhalb der individuellen Fähigkeiten bzw. des individuellen Wissens eines Agenten liegt [BoG08]. Agentensysteme können somit als Problemlösungssysteme bezeichnet werden, die große Aufgaben in Unteraufgaben zerlegen und sie dann gemeinsam lösen [BJW01]. Auch bei der Definition von **Agentensystemen** wird im Rahmen dieser Arbeit aufgrund der Automatisierungstechnischen Ausrichtung auf die Norm VDI/VDE 2653-1 zurückgegriffen:

„Ein Agentensystem besteht aus einer Menge von Agenten, die interagieren, um gemeinsam eine oder mehrere Aufgaben zu erfüllen. Laufzeitumgebungen, Ablaufsysteme und Plattformen für technische Agenten sind eine mögliche Basis zur Realisierung von Agentensystemen, sie sind selbst keine Agentensysteme.“ [VDI10]

Etablierte agentenbasierte Steuerungssysteme benutzen **heterarchische oder holonische Steuerungsarchitekturen** in denen Agenten üblicherweise Ressourcen, Produkte oder Aufgaben repräsentieren. Die Allokation erfolgt durch dynamische **Marktmechanismen**, was zu einfachen, aber fehlertoleranten Systemen führt [BoG08].

Die Anwendungsgebiete von Agentensystemen in der Industrie sind zahlreich. Monostori et al. teilen sie in folgende acht Kategorien ein: 1) Konstruktion, 2) Prozessplanung, 3) Produktionssys-

¹¹ Bei dem Begriff „Multiagentensystem“ handelt es sich um eine unzutreffende Übersetzung [VDI10]. Im Rahmen dieser Arbeit wird stattdessen der deutsche Begriff „Agentensystem“ bzw. die englische Abkürzung MAS verwendet.

templanung und Ressourcenallokation, 4) Produktionsplanung und -steuerung, 5) Prozesssteuerung, Monitoring und Diagnose, 6) Unternehmensorganisation und -integration, 7) verteilte Produktion in Netzwerken sowie 8) Montage- und Lebenszyklus-Management [MVK06].

Eigenschaften von Agentensystemen

Ungeachtet einer fehlenden allgemeingültigen Definition von Agenten existiert in der Fachwelt Einigkeit über die Eigenschaften von Agenten und Agentensystemen. Zu den **Kerneigenschaften von Agenten** gehören nach [GiH07, BCG08, BoG08, Moo13, Unl15a] u.a.:

- **Autonomie:** die Fähigkeit, selbständig zu agieren und Entscheidungen zu treffen, ohne dass externe Eingriffe durch Menschen oder andere Entitäten notwendig wären.
- **Soziales Verhalten:** die Fähigkeit, mit anderen Agenten zu kommunizieren, um individuelle oder gemeinsame Ziele zu erreichen. Dafür können Agenten Wissen über andere Agenten erwerben und sammeln.
- **Reaktivität:** die Fähigkeit, die Umgebung wahrzunehmen und auf Veränderungen zu reagieren (Voraussetzung dafür ist die Situiertheit bzw. Einbettung in eine Umwelt).
- **Proaktivität:** die Fähigkeit, die Initiative zu übernehmen, um Ziele zu erreichen
- **Lernfähigkeit:** die Fähigkeit, aus zuvor getätigten Entscheidungen oder Beobachtungen der Umwelt zu lernen.

Nach [RuN10] können vier Grundtypen von Agenten unterschieden werden (vgl. Abbildung 3-9).

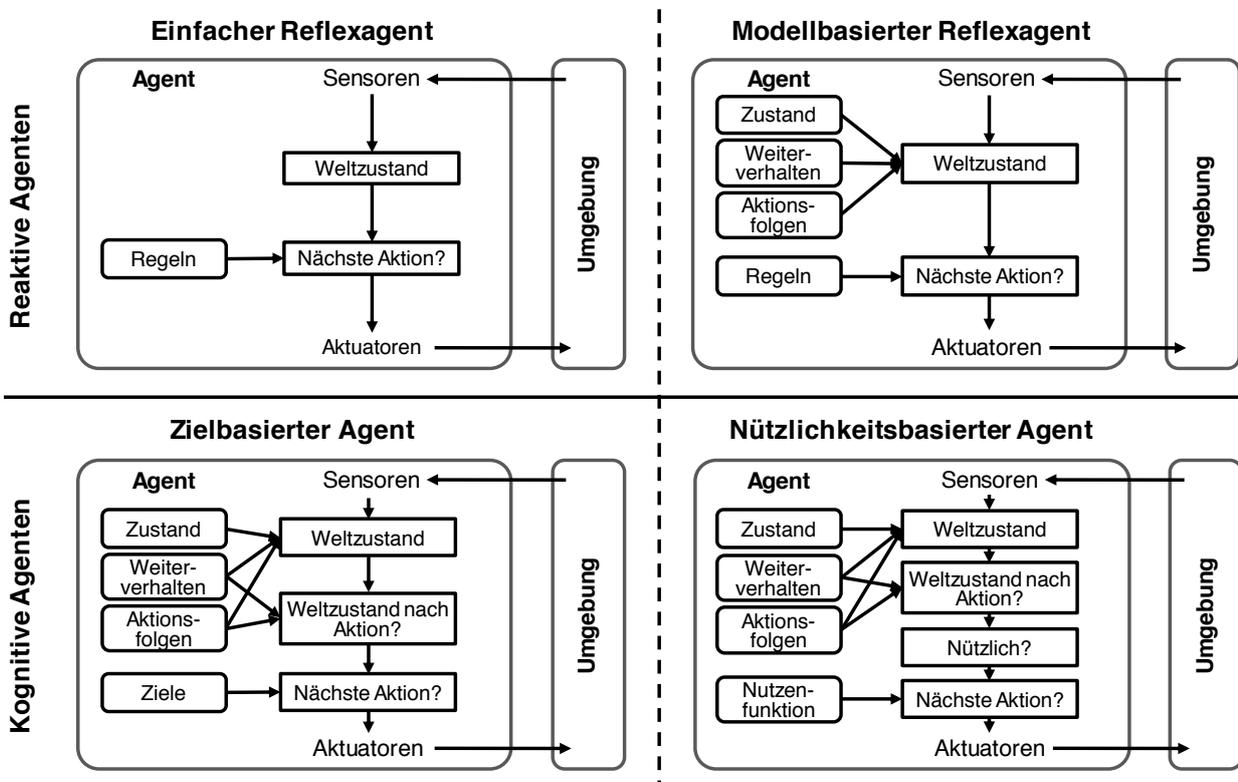


Abbildung 3-9: Vier Grundtypen von Agenten i.A.a. [GiH07, RuN10]

Einfache **Reflexagenten** treffen Entscheidungen auf Basis der Wahrnehmung ihrer Umgebung und können nicht auf vergangene Ereignisse zurückzugreifen. **Modellbasierte Reflexagenten**

hingegen treffen Entscheidungen auf Basis des inneren Zustandsmodells, das historische Daten enthält. **Zielbasierte Agenten** verfügen nicht nur über einen inneren Zustand, sondern ziehen zum Treffen von Entscheidungen auch Zielinformationen heran. **Nutzenbasierte Agenten** schließlich betrachten mithilfe einer Nutzenfunktion darüber hinaus wie sie ihr Ziel erreichen können. Dies ist insbesondere bei der Auswahl mehrerer Lösungen und beim Abwägen zwischen konfliktären Zielgrößen erforderlich. Die ersten beiden Agententypen gehören zur Kategorie der reaktiven Agenten, die letzten beiden werden als kognitive Agenten bezeichnet [GiH07].

Die individuellen Agenten inhärenten Eigenschaften führen zu emergenten **Eigenschaften der Agentensysteme** in die sie eingebettet sind [Bog13]. Dank der Autonomie der Agenten realisieren Agentensysteme dezentrale Steuerungsstrukturen, die sich durch hohe **Flexibilität, Adaptivität, Robustheit** und **Fehlertoleranz** auszeichnen. Da Agenten einem System jederzeit hinzugefügt und wieder daraus entfernt werden können, ist sowohl die **Skalierbarkeit** als auch die **Rekonfigurierbarkeit** zur Laufzeit gewährleistet [Unl15a]. Agentensysteme erfüllen damit alle der in Kapitel 2 definierten Anforderungen an moderne Produktionssysteme.

Entwicklungs- und Implementierungsstand von Agentensystemen

Aufgrund des breiten Anwendungsspektrums von Agentensystemen – von der Modellierung des Sozialverhaltens von Gruppen über die Simulation von Logistikströmen bis zu hin zur Produktionssteuerung – existieren zahlreiche Frameworks und Implementierungen¹². Auch für die in dieser Arbeit betrachteten Felder lassen sich relevante Projekte ausmachen.

Ein frühes Industrieprojekt, das eine agentenbasierte Steuerung umsetzt ist die **Production 2000+** genannte Initiative der früheren Daimler Chrysler AG [Bus12]. Im Rahmen eines Piloten wurde eine agentenbasierte Zylinderkopffertigung realisiert, um die Flexibilität und Robustheit der Produktion zu steigern. Darin werden flexible Maschinen eingesetzt, die gemeinsam alle geforderten Produktvarianten fertigen können und sich bei Bedarf die notwendigen NC-Programme für jede Variante herunterladen. Die Maschinen sind durch ein flexibles Förderbandsystem verbunden, um die Werkstücke in Abhängigkeit eines Bearbeitungsgraphen zwischen allen Maschinen hin- und herbewegen zu können (vgl. Abbildung 3-10).

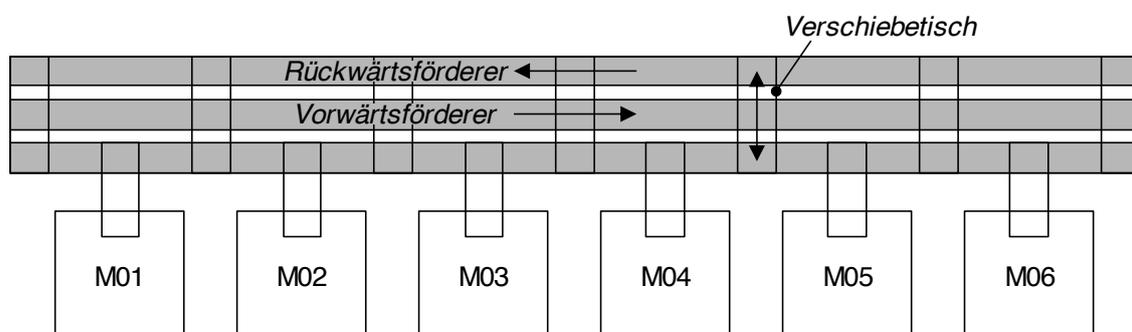


Abbildung 3-10: Layout von Production 2000+ i.A.a. [Bus12]

¹² In [Wik17] erfolgt eine Gegenüberstellung von ca. 90 Agenten-Frameworks.

In der fünf Jahre andauernden Laufzeit des Projektes konnte eine hohe Zuverlässigkeit und eine um 20% erhöhte Produktivität des Systems nachgewiesen werden. Ferner konnten neue Produkte ohne Anpassung der Agenten leicht hinzugefügt werden.

Auf wissenschaftlicher Seite ist das Forschungsprojekt **Cambridge Packing Cell** des Institute for Manufacturing der Cambridge University hervorzuheben, in dem ein holonisches Fertigungssystem für das Packen von hochgradig kundenindividuellen Geschenkboxen entwickelt wurde [FML03]. In dem System, bestehend aus einem Roboter, einem Shuttle-Förderer, einem angeschlossenen Warenhaus sowie Boxen für die Produkte, sind für alle Ressourcen und Produkte Agenten implementiert. Die Agenten orchestrieren alle Aufgaben wie Auslagern, Entpacken, Zwischenlagern oder Sortieren der Güter (Geschenkartikel). Auf Basis der Agentensteuerung konnte eine ausgefeilte Ressourcenallokation umgesetzt werden, die sich in einer hohen Reaktivität und Fehlertoleranz des Systems widerspiegelt. Der effektive Umgang mit Störungen resultiert folglich in einer ebenfalls hohen operativen Performance. Dank der Umsetzung der Plug & Produce-Philosophie können darüber hinaus Rekonfigurationen aufwandsarm ausgeführt werden.

Ein neueres Forschungsprojekt, der Industrie 4.0-Demonstrator **MyJoghurt**, entspringt einer offenen Initiative des Lehrstuhls für Automatisierung und Informationssysteme der Technischen Universität München [PMD17]. Der agentenbasierte Demonstrator vernetzt eine Menge von deutschlandweit verteilten intelligenten Produktionsanlagen und ermöglicht eine verteilte Produktion von kundenspezifisch konfiguriertem Joghurt. Die Produktionsschritte beinhalten die Herstellung von Rohjoghurt, die Beimischung einer Auswahl von Zusätzen (z.B. Früchte, Schokolade etc.), die Herstellung der Verpackung und die Abfüllung des personalisierten Joghurts. Da die Kopplung der Anlagen weitestgehend automatisch erfolgt, kann die Produktion dynamisch rekonfiguriert und skaliert werden kann.

Es lassen sich zahlreiche weitere agentenbasierte Anwendungen finden wie z.B. das Gepäckfördersystem für Flughäfen der Siemens AG, Staplerleitsystem der PSI Logistics AG, das Transportsteuerung von Jung oder der intelligente Container von Lang et al. [GöL10, EHH10, Jun15, LJM11]. Die Aufführung aller Anwendungen würde jedoch den Umfang dieser Arbeit übersteigen, sodass stattdessen ausgewählte Beispiele mit Bezug auf die behandelten Anwendungsfälle in den Kapiteln 7 bis 8 vorgestellt werden. Für weitere Beispiele der Anwendung von Agentensystemen im industriellen Umfeld wird auf das Werk [LeK15] verwiesen.

Stärken und Schwächen von Agentensystemen

Vor dem Hintergrund von Industrie 4.0 werden hierarchielose Steuerungssysteme gefordert, die aus kooperierenden autonomen Einheiten bzw. CPS bestehen. Dank ihrer **dezentralen Steuerungsarchitektur**, sind Agentensystemen für die Umsetzung solcher Systeme prädestiniert [LCL10, Uni15b, VaB15, VGL15].

Die Stärken von Agentensystemen gegenüber anderen Ansätzen resultieren aus den Eigenschaften von Agenten und Agentensystemen. **Produktionsseitige Vorteile** entstehen durch Ausnutzung der Autonomie der einzelnen Systemteilnehmer [FeL10]. So können Agenten sowohl reaktiv als auch proaktiv auf Änderungen ihrer Umwelt reagieren, was zu einem flexiblen, adaptiven und hochgradig robusten Systemverhalten führt [CKM15]. Agenten, die u.a. Ressourcen oder Produkte repräsentieren, können jederzeit zur Laufzeit hinzugefügt oder entfernt werden, ohne dass dafür eine Rekonfiguration notwendig wäre, während dies in konventionellen Systemen einen hohen Aufwand verursachen würde [LeR08]. Der agentenbasierte Ansatz erlaubt des Weiteren die Integration heterogener Automatisierungskomponenten in ein ganzheitliches System, sodass immer die am besten geeigneten Komponenten gewählt werden können und eine effiziente Lösung von Automatisierungsaufgaben möglich ist [LaT10]. Ferner können Agentensysteme mithilfe von mechatronikorientierten Agenten nicht nur auf ERP- und MES-, sondern auch auf der Feldebene implementiert werden [LüF13]. Berücksichtigt man zudem die Möglichkeit Agenten sowohl in lokalen Netzwerken als auch über das Internet zu verteilen, wird klar, dass die Anwendungsmöglichkeiten von Agentensystemen erheblich sind [CKM15].

Aus der **Perspektive der PPS** betrachtet entstehen so zahlreiche neue Möglichkeiten. Das Produktionsspektrum kann aufwandsarm angepasst werden, um die Produktindividualisierung zu erhöhen. Die unvermeidbaren Störungen im Produktionsplan werden durch Eingriffe autonomer Agenten kompensiert. Für die Anpassung der Produktion an den Bedarf braucht es nicht mehr monatelanger Planungs- und Inbetriebnahmephase sowie Produktionsunterbrechungen. Stattdessen können Erweiterungen oder Reduzierungen des Systems ad-hoc durchgeführt werden.

Aus den in mehr als zwei Jahrzehnten gesammelten Erfahrungen mit der Entwicklung und dem Betrieb von Agentensystemen entspringen darüber hinaus zwei weitere Vorteile: Einerseits konnte in Forschungs- und Industrieprojekten bestätigt werden, dass agentenbasierte Systeme im Betrieb technisch zuverlässig beherrschbar sind [LaT10]. Andererseits existieren **zur Unterstützung des Entwicklungsprozesses** bewährte Methoden sowohl für die Softwareentwicklung (AOSE) als auch für die Beschreibung des Verhaltens von Agenten [FeL10, CKM15]. Aufgrund des geringen Umfangs der einzelnen verteilten Komponenten ist die Programmierung, Fehlersuche und Pflege von Agentensystemen darüber hinaus einfacher als in konventionellen Systemen [LeR08].

Dennoch sind Agentensysteme nicht für alle Steuerungsprobleme der beste Ansatz. Das liegt zum einen daran, dass eine dezentrale Lösung **nicht zwangsläufig zu besseren oder schnelleren Lösungen** von Optimierungsproblemen führt als zentrale Ansätze [Mön06, MVK06]. Agentensysteme sind daher zwangsläufig nur eine von mehreren Softwareentwicklungsprinzipien, die in Unternehmen eingesetzt werden können. Sie müssen daher **mit anderen Applikationen integriert werden** können. Zum anderen ist die erstmalige Implementierung von Agentensystemen

mit einem Umdenken hinsichtlich des Steuerungsparadigmas verbunden, was zu erhöhtem Aufwand und Risiko führen kann. Dies wird am Beispiel des Projektes Production 2000+ deutlich. Das Projekt ermöglichte eine deutlich höhere Flexibilität, diese war zu dem Zeitpunkt aber nicht in dem Ausmaß gefordert. Da die an die Anlage gestellten Anforderungen mit konventionellen Lösungen günstiger erreicht werden konnten, wurde das Projekt trotz der erhöhten Flexibilität, Produktivität und Robustheit nach Ende des Produktlebenszyklus der Zylinderköpfe nicht fortgesetzt [Bus12]. Die Anforderungen haben sich seitdem jedoch weiter erhöht, sodass Agententechnologien wieder ins Rampenlicht gerückt sind. Vor der Entscheidung für oder gegen Agentensysteme sollte jedoch, wie bei allen Technologieentscheidungen, eine Potenzialanalyse durchgeführt werden [LaT10].

Bei einer **Agentifizierungspotenzialanalyse** sollten nach [BJW01] mindestens drei Faktoren berücksichtigt werden: 1) Damit die Steuerungslogik verteilt werden kann, muss das System aus mehr als einer Entscheidung bestehen (zwingend). 2) Entscheidungen müssen zumindest teilweise unabhängig sein (zwingend). 3) Um die Stärken von Agentensystemen vollständig zu erschließen ist es vorteilhaft, wenn Entscheidungen zeitlich verteilt anfallen (optional).

Gleichzeitig sollten auch Risiken des Einsatzes von Agentensystemen im Rahmen einer **Risikoanalyse** bewertet werden. Allem voran ist zu beachten, dass die Systemleistung von Agentensystemen hochgradig abhängig von der Definition der Marktregeln ist. Eine ausgiebige Erprobung dieser Regeln kann das Risiko reduzieren und zu einer hohen Robustheit führen. Es kann jedoch kein minimales Leistungsniveau garantiert werden, da das Verhalten des Systems nicht in allen Situationen vorhergesehen werden kann [BoG08]. Bei der Entwicklung agentenbasierter Methoden ist zudem Know-how der Anwendungsdomäne notwendig, um das System an den konkreten Anwendungsfall anzupassen [FeL10]. Zudem können, je nach Größe des Systems, hohe Rechenanforderungen existieren. Zur Sicherung der Interoperabilität zwischen den Agenten des Systems bedarf es einheitlicher Kommunikationsstandards wie z.B. FIPA-ACL¹³, die die interne Kommunikation strukturieren [CKM15]. Für die Kommunikation nach außen existieren derzeit hingegen keine Vorgaben, was die Realisierung einer systemübergreifenden Kommunikation insbesondere in großen IT-Landschaften erschwert. Folglich gibt es hier weiterhin Forschungsbedarf.

Fazit zu Agentensystemen

Die zahlreichen Vorteile von Agentensystemen machen den Ansatz zum bevorzugten Mittel für die Realisierung von dezentralen Strukturen für Industrie 4.0. Dank ihrer Autonomie ermöglichen Agenten ein hohes Niveau an Flexibilität, Agilität und Robustheit. Dabei sind sie gleichzeitig einfacher zu entwickeln als konventionelle Systeme und es kann auf über zwei Jahrzehnte Erfahrungen in der AOSE zurückgegriffen werden. Ergibt eine Potenzialanalyse, dass Agenten für die

¹³ Die Foundation of Intelligent Physical Agents (FIPA) ist ein Standardkomitee der IEEE Computer Society, das sich mit der Erarbeitung von Standards für Agentensysteme beschäftigt. Zu den entwickelten Standards gehört die Agent Communication Language (FIPA-ACL), die eine Nachrichtenstruktur für den Datenaustausch zwischen Agenten spezifiziert (Näheres auf <http://www.fipa.org>)

Lösung einer bestimmten Aufgabe sinnvoll sind, kann ein Agentensystem unter Zuhilfenahme von domänenspezifischen Wissen entwickelt werden. Zum Herunterbrechen von Agentensystemen auf die Feldebene erscheint es sinnvoll mechatronische Agenten zu verwenden oder Agentensysteme mit dem IEC 61499-Ansatz zu verbinden. Hinsichtlich der Integration von Subsystemen in IT-Landschaften existieren jedoch immer noch Forschungslücken, die eine Weiterentwicklung der bestehenden Entwicklungsmethoden notwendig erscheinen lassen. Ein möglicher Ansatz ist die bereits angedeutete Verknüpfung von Agentensystemen und SOA.

Bevor darauf eingegangen werden kann, erfolgt im nächsten Abschnitt eine Gegenüberstellung der in diesem Kapitel vorgestellten Paradigmen sowie die Auswahl eines Ansatzes für die weitere Verwendung im Rahmen dieser Arbeit.

3.4 Bewertung der Paradigmen zur Umsetzung von Industrie 4.0

Die vorangehenden Betrachtungen der Software-Design-Paradigmen IEC 61131, IEC 61499, SOA und Agentensysteme zeigen deren individuelle Stärken und Schwächen. Bei der Gegenüberstellung der Ansätze fällt darüber hinaus auf, dass sie sich in gewissen Situationen gegenseitig ergänzen können.

IEC 61131

Die IEC-Norm 61131 ist unbestreitbar ein De-Facto-Standard der Fertigungsindustrie und hat weltweit einen hohen Automatisierungsgrad von Fabriken ermöglicht. Gleichzeitig führt die Anwendung des Standards zu monolithischen Softwarestrukturen mit geringer Interoperabilität. Da der Standard darüber hinaus für die Entwicklung hierarchisch strukturierter Automatisierungssysteme entwickelt wurde, stellt die Realisierung dezentral organisierter Steuerungssysteme eine erhebliche Herausforderung dar. Dies sind vor dem Hintergrund der zunehmenden Komplexität in Produktionssystemen entscheidende Nachteile. Aus Sicht des Autors haben die anderen drei Ansätze eine höhere Erfolgswahrscheinlichkeit die in Kapitel 2 definierten Anforderungen zu erfüllen.

IEC 61499

IEC 61499 hebt sich durch das ereignisbasierte Ausführungsmodell von konventionellen Ansätzen ab. Im Vergleich zu SOA und Agentensystemen kann mit dem Ansatz jedoch nur ein geringes Maß an Intelligenz umgesetzt werden und es fehlen ausgereifte Entwicklungsmethoden sowie eine formale Semantik für Steuerungsanwendungen. An dieser Stelle sind SOA und Agentensysteme überlegen. Die IEC 61499-Norm kann ihre Stärken jedoch auf den unteren Ebenen der Automatisierungspyramide ausspielen, da sich mit ihr – im Gegensatz zu den anderen Paradigmen – echtzeitfähige Anwendungen realisieren lassen. Da der Fokus dieser Arbeit vor allem auf der MES-Ebene liegt und dort potenziell große Verbesserungen zu erzielen sind, ist eine alleinige Anwendung von IEC 61499 im Rahmen dieser Arbeit nicht zielführend. Die Stärken des Ansatzes können jedoch sehr wohl in Verbindung mit SOA oder Agentensystemen genutzt werden, um die

Dezentralisierung von der MES-Ebene bis auf den Shopfloor zu tragen [Kad15]. Dafür sollten Anwendungen mit und ohne Echtzeitanforderungen unterschieden werden.

SOA

Während der Einsatz von IEC 61499 auf den unteren Ebenen sinnvoll ist, eignet sich SOA stärker zur Sicherung der Interoperabilität von Applikationen auf höheren Ebenen. Ebenso spricht die Wiederverwendbarkeit von Services, die Skalierbarkeit ganzer Systeme sowie die Unabhängigkeit von Programmiersprachen und Betriebssystemen für SOA. Für die Fertigungsebene nachteilig ist jedoch der fehlende Fokus auf die Beschreibung von Ausführungsdetails, die ein elementarer Bestandteil von IEC 61499 ist. Schwerer noch wiegt der Mangel an sozialen Fähigkeiten der Teilnehmer, durch den Proaktivität verhindert wird.

Agentensysteme

Die Schwächen von SOA können agentenbasierte Systeme aufgrund der Autonomie und Intelligenz einzelner Agenten kompensieren. Da sie sowohl reaktives als auch proaktives Verhalten aufweisen, können damit hochgradig adaptive, fehlertolerante und robuste Produktionssteuerungssysteme realisiert werden. Durch ihre sozialen Fähigkeiten können Agenten zudem komplexe Probleme gemeinsam lösen und die Produktion damit flexibel, rekonfigurierbar und skalierbar machen. Langjährige Erfahrungen aus der Entwicklung und dem Betrieb von Agentensystemen in der Automatisierungstechnik haben darüber hinaus vergleichsweise reife Entwicklungsmethoden hervorgebracht.

Zusammenfassende Bewertung der Ansätze

Zusammenfassend betrachtet werden die vergleichsweise wenigen Schwächen der agentenbasierten Softwareentwicklung für die Automatisierungstechnik durch ihre zahlreichen Vorteile mehr als aufgewogen. Die bestehenden Schwachpunkte können darüber hinaus durch die Kombination mit weiteren Ansätzen – z.B. IEC 61499 für die Umsetzung auf dem Shopfloor und SOA für die Interoperabilität innerhalb von sowie zwischen Unternehmens-IT-Landschaften – verringert werden. Ein zusätzliches Argument für die Nutzung von Agentensystemen ist die **derzeitige und prognostizierte Entwicklung** der Technologie. Agentensysteme gehören in das Forschungsfeld der Künstlichen Intelligenz und durchlaufen zurzeit eine ähnliche Entwicklung wie andere Technologien der Domäne (z.B. Machine Learning oder Deep Neural Networks). Ebenso wie diese Technologien wurden auch Agentensysteme zunächst nur sehr langsam adaptiert. Die langsame Durchdringung in industriellen Anwendungen lässt sich auf drei Gründe zurückführen, die heute nicht mehr zutreffen. Erstens mangelte es an der **technischen Realisierbarkeit**, denn Agentensysteme waren noch nicht ausgereift genug und mobile Endgeräte zu teuer und zu leistungsschwach, um den Einsatz zu rechtfertigen. Zweitens bestand in der Industrie ein deutlich **geringerer Flexibilitätsbedarf** als heute, sodass einfachere, monolithische IT-Systeme ausgereicht haben, um die Komplexität zu beherrschen. Drittens war die **Vorhersagbarkeit** von Agentensystemen im Vergleich zur Domäne der Objektorientierten Programmierung viel unzuverlässiger, da

es an Methoden zur Entwicklung, Verifikation und Fehlersuche mangelte. Heute sind diese Faktoren nicht mehr zutreffend. Agentensysteme haben einen **Reifegrad erreicht**, der den Einsatz im industriellen Umfeld ermöglicht [Unl15a]. **Rechenleistung** und **Speicherkapazität** sind preiswert verfügbar [McK15] und neue **Modellierungsmethoden** vereinfachen die Bewertung von Systementwürfen [Kad15]. Diese Entwicklungen haben dazu geführt, dass Agentensysteme die Phase der überzogenen Erwartungen verlassen und das Plateau der Produktivität erreicht haben [Lil12, Unl15a].

Vor dem Hintergrund des Fokus dieser Arbeit auf die Steuerung und Integration komplexer Automatisierungssysteme, erscheint das Designparadigma der **Agentenorientierten Softwareentwicklung** der **vielversprechendste Ansatz** zu sein. Er wird folglich als Grundlage für die weiteren Ausführungen in dieser Arbeit ausgewählt.

Herausforderungen für Agentensysteme

Die Fortschritte bei der Entwicklung und dem Betrieb von Agentensystemen bieten eine vorteilhafte Ausgangslage, um die Steuerung von Automatisierungssystemen grundlegend zu verändern. Damit diese Entwicklung einen tatsächlich disruptiven Charakter annehmen kann, müssen noch weitere Herausforderungen durch Forschung und Wirtschaft gleichermaßen gelöst werden. Die Stärken von Agentensystemen bei der isolierten Steuerung von Subsystemen innerhalb komplexer IT-Landschaften reicht im Zeitalter von vernetzten Unternehmen, Konglomeraten und unternehmensübergreifenden Partnerschaften nicht aus. Vor diesem Hintergrund führt die Optimierung eines **Subsystems** nicht zu einer ausreichenden Verbesserung des **Gesamtsystems**. Ihr volles Potenzial werden Agentensysteme erst dann erreichen, wenn sie systemweit eingesetzt werden und derart miteinander interagieren, dass das Gesamtsystem am optimalen Betriebspunkt arbeitet. Um der **Interdependenz** zwischen den Systemen Rechnung zu tragen, wird eine Entwicklungsmethode benötigt, die eine schrittweise Integration voneinander abhängiger Agentensysteme im Kontext eines größeren Gesamtsystems ermöglicht.

Als erster Schritt für die Entwicklung einer solchen Methode, werden im folgenden Kapitel relevante Architekturen und Modellierungsmethoden von Agentensystemen diskutiert, auf denen der weitere Teil dieser Arbeit aufbaut.

4 Struktur und Design industrieller Agentensysteme

In der über 25-jährigen Geschichte von Agentensystemen in der Fertigungssteuerung sind in zahlreichen Projekten verschiedene Ansätze für die Entwicklung und Ausgestaltung dezentraler agentenbasierter Steuerungsarchitekturen entstanden. Im nachfolgenden Kapitel 4.1 werden zunächst bekannte Architekturstile vorgestellt und allgemeingültige Entwurfsmuster identifiziert. Im Anschluss daran werden in Kapitel 4.2 Entwicklungsmethoden sowie ihre Stärken und Schwächen diskutiert und auf dieser Basis eine Methode ausgewählt, die als Ausgangspunkt für die Weiterentwicklung in den nächsten Kapiteln dienen soll.

4.1 Agentenbasierte Steuerungsarchitekturen

Die strukturelle Ausgestaltung von agentenbasierten Steuerungsansätzen kann auf unterschiedliche Weise umgesetzt werden. In der Literatur wird eine Reihe von Architekturen beschrieben wie z.B. AARIA, MASCADA, metaMORPH, HOLOS/MASSIVE, DEDEMAS, ADRENALIN etc. [BLT10]. Aus dieser Menge heben sich einige Architekturen ab. Die vier Architekturstile PROSA, PABADIS bzw. die Weiterentwicklung PABADIS'PROMISE, ADACOR und GRACE zeichnen sich dadurch aus, dass sie die gesamte Automatisierungspyramide abdecken und einen besonderen Zuspruch in Forschung und Industrie genießen [LüF13]. Darüber hinaus existieren zahlreiche weitere Ansätze zur Implementierung von Agenten auf Feldsteuerungsebene. Jede dieser Steuerungsarchitekturen wird im Folgenden in einem kurzen Beitrag vorgestellt.

4.1.1 PROSA

Die PROSA-Architektur¹⁴ wurde 1998 von Valckenaers et al. veröffentlicht und ist eine weit verbreitete HMS-Referenzarchitektur, in der Holone aus dem HMS-Steuerungsparadigma als Agenten implementiert werden. Sie wird durch die vier Holontypen **Product** (PH), **Resource** (RH), **Order** (OH) und **Staff Holon** (SH) konstituiert. Letzteres ist eine optionale Komponente, die, in Analogie zu Unternehmensorganisationen, der Unterstützung anderer Funktionen dient [VBW98]. Beispielsweise können SHs andere Holons mit Experten-Know-how versorgen [VGH06]. Die anderen drei Holontypen sind die Grundbausteine von PROSA. Ihre Beziehung zueinander ist in Anhang 2 abgebildet. OHs repräsentieren einen Fertigungsauftrag und verwalten die physischen Produkte sowie ihre Statusmodelle und Logistikinformationen. RHs bestehen ebenfalls aus einem physischen und einem informationsverarbeitenden Teil. Sie bieten ihre Kapazitäten und Funktionalitäten anderen Holons an. PHs wiederum stellen Produkt- und Prozesswissen wie z.B. Arbeitspläne und Qualitätsvorschriften zur Verfügung, um die korrekte Ausführung der Fertigungsschritte zu gewährleisten [Mön06, BoG08]. Die Holons arbeiten in zwei voneinander getrennten Holarchien zusammen. In der ersten Holarchie kooperieren alle drei Haupttypen um die **Ressourcenallokation** zu bewerkstelligen. In der zweiten Holarchie arbeiten PHs und RHs zusammen

¹⁴ Product-Resource-Order-Staff-Architecture

um die **Prozesssteuerung** zu realisieren [BoG08]. Die PROSA-Architektur eignet sich z.B. zur Verbindung von Planungs- und Fertigungssystemen in Fabriken [VeV06].

4.1.2 PABADIS und PABADIS’PROMISE

PABADIS’PROMISE¹⁵ setzt auf dem PABADIS¹⁶-Projekt auf, in dem ein Agentensystem zur Steuerung der MES- und Feldebene beschrieben wird. PABADIS’PROMISE erweitert den Ansatz von PABADIS durch Einführung einer agileren Steuerungsarchitektur auf Basis von mobilen Agenten. Maschinen und Geräte werden durch **Resource Agents** (RA) repräsentiert, die die Fähigkeit haben zu kommunizieren und über offene Fertigungsaufträge zu verhandeln [Pes14]. Im Kern der PABADIS’PROMISE-Architektur stehen die drei Elemente **Order Agent** (OA), **Ability Broker** (AB) und **Information Collector** (IC). Als Schnittstelle zwischen der ERP- und der MES-Ebene dienen die **Resource Agent Supervisors** (RAS), die **Order Agent Supervisors** (OAS) und das **Product Data Repository** (PDR). Auf dem Shopfloor kommunizieren die RAs mit den Anlagen und bilden eine Schnittstelle zur Feldsteuerungsebene [PAB08]. Details der PABADIS’PROMISE-Architektur sind in Anhang 3 zu finden.

Im Vergleich zu PROSA erlaubt die PABADIS’PROMISE-Architektur höhere Autonomie und Kooperation durch dynamische Ressourcensteuerung. Eine Hierarchie ist nicht explizit festgelegt, sondern implizit durch eine flexible Auftragsstruktur definiert. Mögliche Hierarchien können als Erweiterung der Architektur beschrieben werden, wie z.B. in der Arbeit von [Rya16]. Die Architektur erlaubt ferner eine verteilte Entscheidungsfindung und bindet Feedback aus der ERP-Ebene über periodische und ereignisbasierte Berichte ein [BLT10]. Die PABADIS’PROMISE-Architektur vereinfacht die Rekonfiguration von Fertigungs- und Materialflusssystemen. Dies kann sowohl durch Hinzufügen oder Entfernen von Komponenten über den Fabriklebenszyklus geschehen (plug & produce), als auch durch dynamische Anpassungen während der Produktion. Die Ressourcen sind folglich nur durch ihre physischen Fähigkeiten und nicht durch Steuerungsprozesse limitiert, was zu einer besseren Ausnutzung der Anlagenflexibilität beiträgt und erlaubt es, kundenspezifische Produkteigenschaften an den spätestmöglichen Punkt in der Produktion zu verlagern [VGL15].

4.1.3 ADACOR

Die ADACOR-Architektur¹⁷ wurde entwickelt, um eine dynamische und agile Adaption an Störungen in der Fertigung zu ermöglichen und die Integration der Wertschöpfungskette voranzutreiben [LeR03, LCR05]. Sie besteht aus vier Arten von Holons: Product, Task, Operational und Supervisor Holons [LeR02]. Ihre Struktur ist in Anhang 4 abgebildet. Produkte werden in der ADACOR-

¹⁵ PABADIS based Product Oriented Manufacturing Systems for Re-Configurable Enterprises

¹⁶ Plant Automation based on Distributed Systems

¹⁷ Adaptive Holonic Control Architecture for Distributed Manufacturing Systems

Architektur durch **Product Holons** (PH) repräsentiert, die produkt- und prozessrelevantes Wissen bündeln und für die Prozessplanung verantwortlich sind. Fertigungsaufträge werden durch **Task Holons** (TH) abgebildet. Sie übernehmen die Steuerung und Überwachung der Fertigungsschritte und speichern dynamisch die dazugehörigen Informationen. Über die **Operational Holons** (OH) werden die physischen Fertigungsressourcen repräsentiert, ihr Verhalten gesteuert und die Produktionspläne optimiert [LeR03]. Während die PHs, THs und OHs der PROSA-Architektur entsprechen, haben die **Supervisor Holons** (SH) eine besondere Rolle, denn die ADACOR-Architektur ist weder vollständig dezentral noch hierarchisch. Stattdessen wechselt sie in Abhängigkeit der aktuellen Anforderungen zwischen beiden Paradigmen. In störungsfreien Situationen (stationärer Zustand) überwacht das SH die Aktivitäten der unteren Holons und integriert die Planungs- und Steuerungsmechanismen um sie zu koordinieren. Bei einer Störung (Übergangszustand) müssen die Holons jedoch ggf. ohne die Hilfe von SHs auskommen. Dadurch unterscheiden sich die SHs von den Staff-Holons der PROSA-Architektur. Identifiziert ein OH eine Störung, wird sein Autonomiefaktor erhöht und die Information, dass es einer Reorganisation bedarf, wird mittels einer pheromonbasierten Technik an andere Holons verbreitet. Dadurch wechseln die Holons in einen Übergangszustand und reorganisieren sich in einer heterarchischen Struktur. In diesem Zustand kommunizieren sie direkt miteinander und erarbeiten so einen alternativen Produktionsplan. Wenn die Störung behoben wurde, lässt die Stärke des Pheromonsignals nach und es wird wieder eine hierarchische Struktur angenommen, die meist der ursprünglichen entspricht. Dies resultiert in dynamischer Autonomie und Informationsverbreitungsmechanismen, wie sie aus ameisenkoloniebasierten Ansätzen bekannt sind [LCR05].

4.1.4 GRACE

Den Prinzipien der ADACOR-Architektur folgend wurde die GRACE-Architektur¹⁸ entwickelt, deren primäres Ziel die Integration der Prozess- und Qualitätssteuerung ist [LRT15]. GRACE enthält fünf Arten von Agenten. **Product Type Agents** (PTA) enthalten Produkt- und Prozesswissen sowie historische Daten zu vergangenen Produktionsaufträgen, mit denen sie neue Aufträge verfolgen und Anpassungen bei deren Ausführung initiieren können. **Product Agents** (PA) verwalten Kundenaufträge und haben einen Prozessplan zur Ausführung der Fertigungsschritte. Sie interagieren mit Resource und Quality Control Agents um Aufträge umzuleiten und sammeln dabei Produktionsdaten. **Resource Agents** (RA) repräsentieren physische Ressourcen und managen die eigentliche Ausführung der Produktionsschritte auf dem Shopfloor. Sie sind verantwortlich für das Ressourcenmonitoring und können Anpassungen bzw. Optimierungen der Ressourcenparameter vornehmen. **Quality Control Agents** (QCA) sind verantwortlich für die Ausführung der Qualitätssicherung (QS). Dafür sind sie den QS-Stationen zugewiesen, managen die Ausführung der QS-Maßnahmen und passen auf Basis von Umfeldinformationen und historischen Daten die

¹⁸ Integration of Process and Quality Control using Multiagent Technology

QS-Algorithmen und -Parameter an. Schließlich führen die **Independent Meta Agents** (IMA) eine Art Hierarchie in das dezentralisierte System ein und erlauben eine umfassende Überwachung sowie eine verbesserte Entscheidungsfindung. Dazu sammeln sie Daten von individuellen Agenten und identifizieren bedeutsame Korrelationen, die sie in die Systemdatenbank überführen [CCF11]. Die Beziehungen der Agenten in der GRACE-Architektur sind in Anhang 5 dargestellt. Der deutlichste Unterschied von GRACE im Vergleich zu den übrigen vorgestellten Architekturen manifestiert sich im QCA, der eine integrierte Steuerung der QS-Prozesse erlaubt. Der QCA ist jedoch lediglich eine Abstraktion der ansonsten üblichen Ressourcenagenten.

4.1.5 Agenten auf Feldsteuerungsebene

Agentensysteme erfüllen alle Anforderungen zur Realisierung von CPPS und können mithilfe unterschiedlicher Architekturen bis auf Feldsteuerungsebene heruntergebrochen werden [VLL15]. Während dafür einerseits die oben dargestellten Architekturen wie GRACE verwendet werden können, existieren andererseits auch weitere Architekturen, die auf viele Anwendungsfälle übertragen werden können.

Ein Beispiel für eine solche Architektur zur Implementierung von Agenten auf Feldebene ist das in [VLL15] diskutierte Modell `agent@PLC`. Die Architektur sieht einen **CPPS-Fabrikagenten** vor, der die Allokation eines Produktionssystems in einem CPPS-Netzwerk übernimmt. Darin enthalten ist ein **Systemagent**, der über Wissen zur Werksstruktur verfügt, ein **Kommunikationsagent**, der für die Nachrichtenübermittlung zwischen den Einheiten des CPPS verantwortlich ist, ein **Prozessagent**, der den Produktionsprozess und notwendige Teilschritte enthält, ein **Whiteboard**, das die zu erfüllenden Aufgaben und ihren Status enthält sowie ein **Automatisierungssteuerungsagent**, der den physischen Teil der Anlage repräsentiert und das Scheduling der notwendigen Teilschritte für den Prozessagenten übernimmt.

Ein weiteres Beispiel einer Architektur zur Realisierung von Agenten auf Feldsteuerungsebene ist das in [FMV18] entwickelte Metamodell, mit dem eine aus einer Gruppierstation mit einem Sortierförderer und zwei Portalmodulen bestehende Verpackungsanlage aus dem Bereich Lebensmittel und Getränke umgesetzt wurde. Darin werden eingehende Transportobjekte vorverarbeitet und nach vordefinierten Mustern arrangiert, bevor sie anschließend palettiert werden. Das entwickelte Metamodell besteht aus drei Hauptkomponenten: Einer **Ressourcenverwaltung**, in der alle Ressourceninformationen des Materialflusssystems und der darin enthaltenen Materialflussmodule enthalten sind, dem **Transportauftrag**, der die vom Agentensystem berechnete und ausgeführte endgültige Zielroute enthält und einem **Datenmodell** für den Palettenspeicher, das allgemeine Informationen für die Durchführung der Steuerungsaufgabe enthält. Während der Ressourcenteil und das Datenmodell statische Informationen darstellen, die sich nach dem Start einer bestimmten Konfiguration des Materialflusssystems nicht ändern, hängt der Teil, der den Auftrag und die zu erfüllende Steuerungsaufgabe beschreibt, von der Position der eingehenden Transportobjekte ab und wird daher als dynamische Information betrachtet.

In der Literatur sind zahlreiche weitere Beispiele für die Realisierung von Agentensystemen auf Feldsteuerungsebene zu finden. In [WSV13] wird z.B. eine kontinuierliche, hydraulische Heizpresse mit einem agentenbasierten Steuerungssystem gesteuert. Für eine Reihe weiterer Beispiele wird auf die Arbeit von [Kad15] verwiesen. Darüber hinaus findet in [NeK10] eine Bewertung unterschiedlicher Umsetzungstechnologien statt. Aufgrund der zahlreichen Beispiele wird im Weiteren davon ausgegangen, dass die Umsetzungstechnologien für die Implementierung von Agentensystemen auf Feldebene vorhanden sind.

Durch die Analyse der unterschiedlichen Steuerungsarchitekturen und Anwendungsfälle werden Gemeinsamkeiten erkennbar und es lassen sich einige typische Muster und Agentenarten identifizieren, die für die Ausführung spezifischer Aufgaben verwendet werden können. Auf zwei relevante Implikationen wird im Folgenden eingegangen.

4.1.6 Allgemeingültige Entwurfsmuster

Die vorgestellten Agentenarchitekturen gehören zu einer großen Menge an Ansätzen, die seit Begründung der agentenorientierten Programmierung entwickelt wurden. Eine Gegenüberstellung der Ansätze in [LüF15] zeigt, dass sich die Rollen der Agenten trotz unterschiedlicher Bezeichnungen teilweise stark ähneln. Es kann daher erwartet werden, dass sich Konventionen und Entwurfsmuster etablieren, durch die industrielle Agentensysteme leichter und zuverlässiger entworfen und betrieben werden können. Solche Entwicklungen sind aus anderen Bereichen der Softwareentwicklung hinlänglich bekannt und werden z.B. im GMA-Fachausschuss 5.15 "Agentensysteme"¹⁹ diskutiert. Um daraus Best Practices zu identifizieren, wurden in der Studie von Lüder et al. 19 unterschiedliche Architekturtypen auf wiederkehrende Muster hin untersucht [LCZ17]. Als Ergebnis konnten Entwurfsmuster für zwei der wichtigsten Aktivitäten, die Ressourcenallokation und den Ressourcenzugriff, identifiziert werden.

Ressourcenallokation

Das erste identifizierte Entwurfsmuster gilt der Allokation von Aufträgen zu Ressourcen, um jegliche wertschöpfenden und unterstützenden Schritte ausführen zu können. Eine solche Zuordnung muss in nahezu jedem Produktionssystem stattfinden. Aufträge können dabei Kundenaufträge repräsentieren, aber auch Instandhaltungs- oder Reparaturaufträge für Anlagen bis hin zu Informationsaufträgen, bei denen z.B. Informationen zu bestimmten Ressourcen gesammelt werden. Das Scheduling dieser Aktivitäten ist ein komplexes mathematisches Optimierungsproblem, dessen Lösung oft viel Rechenzeit benötigt oder nur eine Approximation darstellen kann (vgl. Kapitel 2.3.1 sowie die Anwendungsfälle in Kapitel 7.1.3 und 8.1.2). In verteilten Steuerungssystemen können solche Probleme effektiv gelöst werden [LCZ17]. Das Entwurfsmuster, das sich hierfür durchgesetzt hat, ist in Abbildung 4-1 dargestellt.

¹⁹ Gesellschaft Mess- und Automatisierungstechnik des VDI/VDE

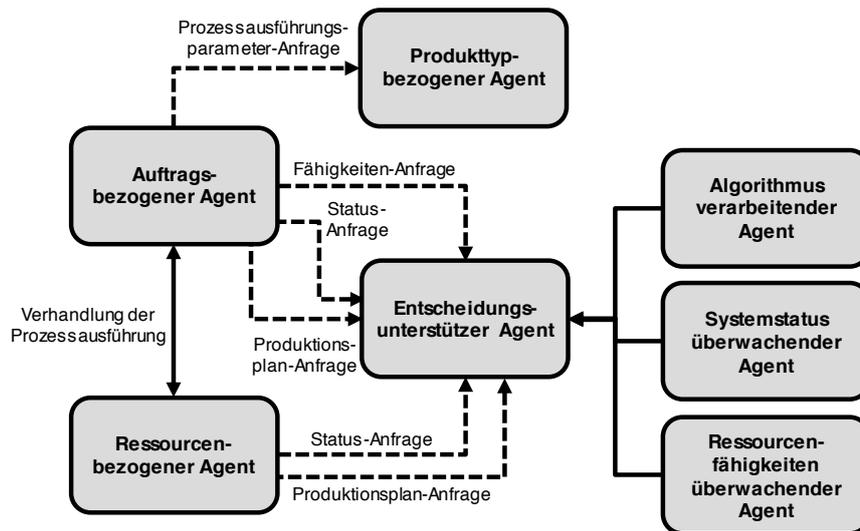


Abbildung 4-1: Entwurfsmuster für die Ressourcenallokation i.A.a. [LCZ17]

Das Entwurfsmuster kann mit zwei bis vier Agenten realisiert werden. Die Minimalausführung besteht aus den Haupttypen **Auftrags-** und **Ressourcenagent**. Beide Agententypen sind für die sie jeweils betreffenden Informationen und Zeitpläne verantwortlich und führen bei Bedarf Verhandlungen auf Basis eines Contract-Net-Protocols (CNP) miteinander. Des Weiteren können **Produktagenten** notwendiges Produkt- und Prozesswissen bereitstellen und/oder **Entscheidungsunterstützungsagenten** mit verschiedenen Algorithmen und Informationen unterstützen [LCZ17]. Ein Implementierungsbeispiel für das Entwurfsmuster zur Ressourcenallokation ist in Anhang 6 beigelegt.

Ressourcenzugriff

Das zweite identifizierte Entwurfsmuster dient dem Zugriff auf Ausführungsfunktionen von Ressourcen. Häufig ist für die Realisierung von wertschöpfenden und/oder unterstützenden Prozessen eine Kombination aus einer Menge an Unterfunktionen notwendig, die durch Kooperation mehrerer Komponenten erreicht wird. Diese Funktionsmengen können in Abhängigkeit der Gegebenheiten in der Produktion unterschiedlich zusammengesetzt sein und werden üblicherweise auf Feldsteuerungsgeräten wie SPS, CNC oder RNC implementiert; im Optimalfall können sie dynamisch den Anforderungen angepasst werden. Da die Feldebene Echtzeitfunktionalität erfordert, müssen MES- und Feldebene getrennt werden. Im Rahmen der Studie von Lüder et al. konnte das in Abbildung 4-2 gezeigte abstrahierte Entwurfsmuster für den Ressourcenzugriff aus der Menge der untersuchten Architekturen destilliert werden [LCZ17].

In dem Entwurfsmuster repräsentieren **Ressourcenagenten** die Fähigkeiten ihrer Ressourcen und bieten diese als Skills im Gesamtsystem an. Dabei werden zwei Typen von Ressourcenagenten unterschieden: einerseits **Feldsteuerungsagenten**, die direkten Zugriff auf die Feldsteuerungsgeräte haben und mit ihnen interagieren und andererseits **Aggregationsagenten**, die Skills bündeln und koordinieren. Ein Implementierungsbeispiel für das Entwurfsmuster zum Ressourcenzugriff ist in Anhang 7 beigelegt.

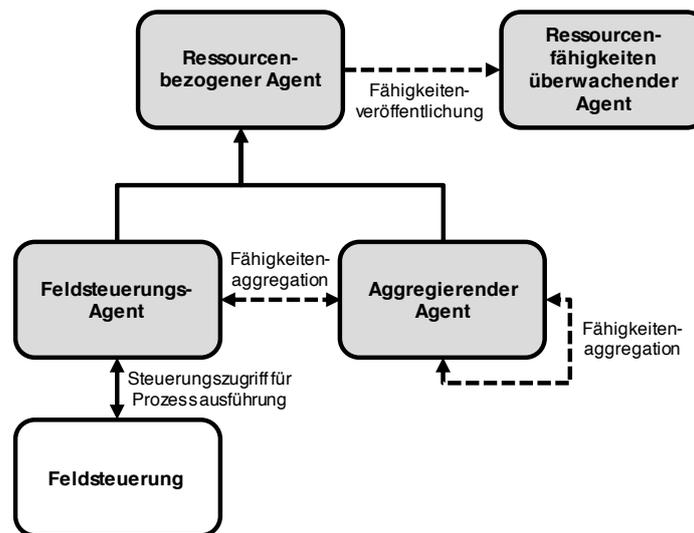


Abbildung 4-2: Entwurfsmuster für den Ressourcenzugriff i.A.a. [LCZ17]

Auf Basis der zwei vorgestellten Entwurfsmuster können zahlreiche Produktionssituationen modelliert werden, sodass beide in den Anwendungsfällen in Kapitel 7.2 und 8.2 zum Einsatz kommen. Entwurfsmuster alleine reichen jedoch nicht aus, um agentenbasierte Steuerungssysteme zu entwickeln. Dafür bedarf es einer tiefgreifenden Analyse und Übersetzung des Produktionssystems in einen agentenbasierten Ansatz. Auch dazu existieren zahlreiche Entwicklungs- bzw. Modellierungsmethoden, die im nächsten Abschnitt vorgestellt werden.

4.2 Entwicklungsmethoden für Agentensysteme

Wie bereits bei den Entwurfsmustern kann auch bei den Entwicklungsmethoden für agentenbasierte Systeme auf eine große Menge von Ansätzen zurückgegriffen werden. Alleine in der VDI-Norm 2653-2 werden über zehn Entwicklungsmethoden aufgezählt und erläutert [VDI12]. Da eine ausführliche Betrachtung aller Ansätze den Rahmen dieser Arbeit übersteigt, wird an dieser Stelle lediglich auf ausgewählte Methoden eingegangen. Für eine detailliertere Betrachtung sei auf die Werke [StS14] und [GiT15] verwiesen, in denen über 30 AOSE-Methoden bewertet werden. Aus der Menge an Ansätzen wurden fünf ausgewählt, die sich durch kontinuierliche Publikationen und signifikanten Einfluss auf das AOSE-Feld auszeichnen. Diese Ansätze sind, gemäß der Reihenfolge ihrer Erstveröffentlichung, Gaia, MaSE, Tropos, Prometheus und DACS. Zudem wurde auch ein neuerer AOSE-Ansatz, die ADMARMS-Methode, aufgenommen. Die Methoden werden in den folgenden Abschnitten 4.2.1 bis 4.2.6 vorgestellt sowie ihre Stärken und Schwächen diskutiert. Als Bewertungskriterien werden die u.a. aus [DaW04, CeZ11, StS14] bekannten und erprobten Merkmale – Konzepte & Eigenschaften, Modellierung & Notation, Prozess sowie Praxistauglichkeit – verwendet. Im Anschluss erfolgt in Abschnitt 4.2.7 eine quantitative Bewertung der Methoden anhand dieser Merkmale sowie die Auswahl eines für diese Arbeit geeigneten Vorgehens.

4.2.1 Gaia (2000)

Übersicht Gaia

Die Gaia-Methode²⁰ wurde erstmals 1999 von Wooldridge et al. veröffentlicht und ist die erste AOSE-Methode, die sowohl umfassend als auch für ein breites Spektrum an Anwendungen einsetzbar ist. Sie behandelt Agentensysteme nicht nur auf der Mikroebene individueller Agenten, sondern auch auf der Makroebene der sozialen Beziehungen zwischen ihnen [WJK99].

Gaia besteht in der ursprünglichen Form aus den zwei Phasen Analyse und Design in denen die fünf Modelle Rollen-, Interaktions-, Agenten-, Service- und Bekanntschaftenmodell Anwendung finden [WJK00]. Die Gaia-Methode beginnt erst nach der Erfassung und Spezifizierung von Anforderungen und schließt die Implementierung nicht ein [CJS04]. Zambonelli et al. haben die Methode 2003 erweitert, um ihren möglichen Einsatzbereich auszudehnen [ZRW03]. In dieser Fassung enthält die Methode weitere Phasen und Modelle (vgl. Abbildung 4-3).

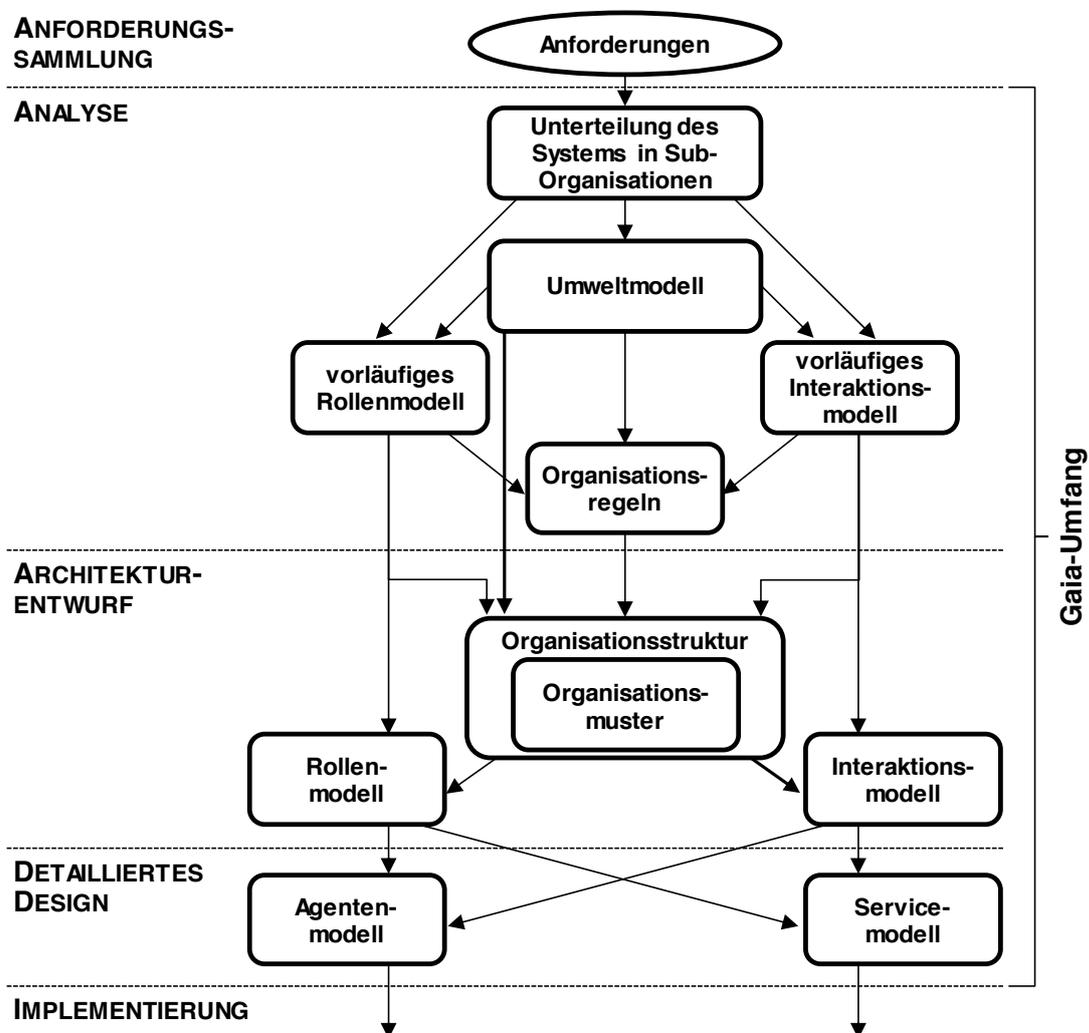


Abbildung 4-3: Gaia-Modelle und ihre Beziehungen im Entwurfsprozess i.A.a. [ZRW03]

²⁰ Der Begriff Gaia entstammt der griechischen Mythologie und steht für „Mutter Erde“. Ferner beschreibt er die Hypothese des Ökologen James Lovelock, der alle lebenden Organismen als Komponenten einer einzigen Entität definiert. Da diese Sichtweise auch auf Agentensysteme zutrifft, ist der Begriff Namensgeber für diese Methode [WJK00].

Die **Analysephase** besteht aus drei Modellen und einer Menge an Organisationsregeln. Im **Umweltmodell** wird eine abstrakte Repräsentation der Umwelt des Agentensystems erstellt. Im **vorläufigen Rollenmodell** werden diejenigen Basisfähigkeiten für die Aufgabenerfüllung in der Organisation identifiziert, die keiner Festlegung auf eine bestimmte Organisationsstruktur bedürfen. Das **vorläufige Interaktionsmodell** spezifiziert die grundlegenden Interaktionen, die für die vorläufigen Rollen benötigt werden. Die beiden vorläufigen Modelle dürfen an dieser Stelle unvollständig bleiben. Als letzter Schritt der Analysephase erfolgt die Erstellung der **Organisationsregeln**, mit denen Einschränkungen von Aktivitäten sowie Protokolle beschrieben werden.

In der folgenden **Architekturentwurfsphase** werden einerseits die Topologie und andererseits das Steuerungssystem der Organisation erarbeitet. Dazu erfolgt die **Fertigstellung** der vorläufigen **Rollen- und Interaktionsmodelle**. In der anschließenden **Detaillierungsphase** erfolgt zunächst die Definition eines **Agentenmodells** und schließlich die Bestimmung des **Servicemodells**. Während im Agentenmodell die notwendigen Agentenklassen und -instanzen durch Aggregation der Rollen identifiziert werden, dient das Servicemodell der Definition von kohärenten Aktivitätsbündeln, in denen Agenten zusammenarbeiten um ihre Rollen zu erfüllen [StS14].

Stärken und Schwächen von Gaia

Die Stärken der Gaia-Methode liegen in ihrem Umfang und der eingängigen Betrachtung von Mikro- und Makroebene. Sie unterstützt nahezu alle agentenbasierten Konzepte (mit Ausnahme der BDI-Notation²¹) und ist daher für viele Anwendungsfälle geeignet [StS14]. Eindeutige Schwächen der ursprünglichen Gaia-Version sind, dass sie sich nur für geschlossene Agentensysteme eignet und unübliche Notationstechniken verwendet werden [CJS04]. Aufgrund dieser Unzulänglichkeiten sind in zahlreichen Werken wie [JPS02, GAP06, MoS06, CeZ09, SpM09, CMO11] Erweiterungen der Methode vorgeschlagen worden, die u.a. Entwicklungswerkzeuge, zusätzliche Modelle, Notationstechniken wie Agent-UML und Anwendungsfälle beinhalten, mit denen diese Schwächen teilweise gemildert werden [StS14]. Die VDI-Norm 2653-2 empfiehlt z.B. bei der Verwendung von Gaia auf die Erweiterung ROADMAP zurückzugreifen [VDI12], da sie die Anwendung von Gaia für offene Systeme erleichtert und die Skalierbarkeit erhöht [JPS02]. Trotz der Vielzahl an Überarbeitungen bleibt Gaia unvollständig und bedarf weiterer Spezifikationen bezüglich des Modellierungsansatzes sowie der Notation. Es existieren derzeit jedoch keine koordinierten Bemühungen diese Unzulänglichkeiten zu beheben [StS14]. In Ergänzung dazu ist die Gaia-Methode vergleichsweise kompliziert und aufwändig, was sie für Entwickler von Automatisierungssystemen unhandlich macht und ihren Einsatz im industriellen Umfeld hemmt. Dies sind Schwächen, die die Gaia-Methode für den Einsatz im weiteren Teil dieser Arbeit unzweckmäßig erscheinen lassen.

²¹ Belief Desire Intention

4.2.2 MaSE (2001)

Übersicht MaSE

Die MaSE-Methode²² wurde 2001 von DeLoach et al. veröffentlicht und ist eine vielseitige Methode zur Entwicklung heterogener Agentensysteme. In Abgrenzung zur Gaia-Methode, die einen agentenorientierten Ansatz verfolgt, gehört MaSE zu den objektorientierten Methoden [VDI12]. MaSE nutzt eine Reihe grafischer Modelle zur Beschreibung von Systemzielen, Verhalten, Agententypen und Kommunikationsschnittstellen für Agenten. Wie die erste Fassung von Gaia gliedert sich auch MaSE in die zwei Phasen Analyse und Design (vgl. Abbildung 4-4) [StS14].

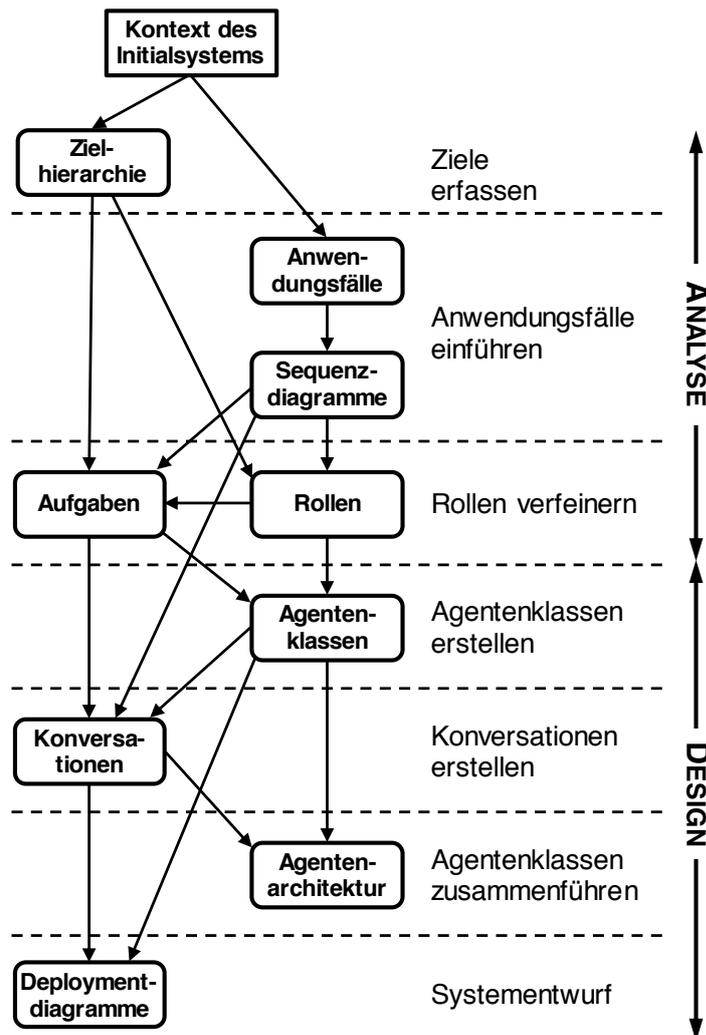


Abbildung 4-4: Phasen der MaSE-Methode i.A.a. [DWS01]

Ziel der **Analysephase** ist es, Rollen zu definieren, die die Aufgaben im System erfüllen können. Dafür erfolgt im ersten Schritt die **Aufnahme der Zielsetzung** aus Systemsicht. Als nächstes werden **Anwendungsfälle** definiert und in **Sequenzdiagrammen** abgebildet. Auf dieser Basis erfolgt eine funktionale Dekomposition, indem eine Menge an **Rollen** und zugehörigen **Aufgaben** definiert wird [StS14].

²² Multiagent Systems Engineering

In der **Designphase** wird die Art und Weise wie das System die definierten Ziele erreichen soll konkretisiert, indem sein Verhalten und seine Architektur festgelegt werden. Dafür werden zunächst **Klassen von Agenten** erstellt, aus denen das System später bestehen soll, und die **Konversationen** zwischen ihnen werden determiniert. Diese Konversationen enthalten Koordinationsprotokolle für Agentenpaare, in denen sowohl das Verhalten des Initiators als auch des Antwortsenders beschrieben wird. Darauf aufbauend erfolgt die **Zusammensetzung von Agentenklassen** in einer Agentenarchitektur, die schließlich in einem finalen Systemdesign mit der physischen Systemarchitektur und der Verteilung von Agentenklassen mündet [StS14].

Stärken und Schwächen von MaSE

MaSE zeichnet sich durch seine universelle Anwendbarkeit in verschiedenen Kontexten aus und unterstützt, mit Ausnahme der BDI-Notation, alle wesentlichen agentenbasierten Konzepte. Eine weitere Stärke ist, dass MaSE im Gegensatz zu Gaia auf die bekannte Unified Modelling Language (UML) als Notationstechnik zurückgreift, was ihre Anwendung für Entwickler erleichtert. MaSE deckt ferner einen Großteil des Entwicklungszyklus ab und es existieren zahlreiche CASE-Tools²³, die die Anwendung erleichtern. Gleiches gilt für die zur Verfügung stehenden Kommunikationsprotokolle [StS14]. Als eine Schwäche ist die Komplexität von MaSE aufzuführen. Zwar ist im Vergleich zu Gaia eine Verbesserung erkennbar, die Methode besteht dennoch aus zahlreichen Schritten und Modellen, die zu einem aufwendigen Entwicklungsprozess mit wenig Spielraum für inkrementelle Verbesserungen führen.

Vor diesem Hintergrund gibt es eine Reihe von Weiterentwicklungen der MaSE-Methode. In [BMG08] wird MaSE erweitert, um die Entwicklung von eingebetteten Echtzeitsystemen zu realisieren. In [DeO10] wurde, unter Beteiligung der ursprünglichen Autoren, eine Weiterentwicklung vorgestellt, die aufbauend auf dem Method-Engineering-Ansatz versucht Designern zu ermöglichen individuelle AOSE-Entwicklungsprozesse zur Verfügung zu stellen. Diese Bemühungen haben bislang jedoch nicht in einer verstärkten Adaption in der Fertigungsindustrie geführt, was sie für die Verwendung in dieser Arbeit ungeeignet macht.

4.2.3 Tropos (2002)

Übersicht Tropos

Die Tropos-Methode²⁴ wurde 2002 durch Bresciani et al. publiziert [BGG02] und verfolgt einen OO-Ansatz [VDI12]. In Abgrenzung zu Gaia und MaSE werden mentale Konzepte wie Ziele und Pläne von Agenten über den gesamten Entwicklungsprozess berücksichtigt, BDI ist jedoch nicht Bestandteil. Tropos setzt bereits früh bei der Anforderungsbestimmung an und berücksichtigt die Umwelt in die das System eingebettet werden soll [BGG02]. Die Tropos-Methode besteht aus den fünf in Abbildung 4-5 dargestellten Phasen.

²³ Computer-Aided Software Engineering

²⁴ Tropos ist vom griechischen "trope" abgeleitet, das „einfach veränderbar“ bzw. „einfach anpassbar“ bedeutet



Abbildung 4-5: Phasen des Tropos-Modells i.A.a. [BGG02]

In der **frühen Anforderungsanalyse** werden die Intentionen der Stakeholder betrachtet und als Ziele modelliert. Die Beteiligten werden als soziale Akteure modelliert, deren Zielerreichung voneinander abhängt. Schließlich werden die funktionalen und nicht-funktionalen Anforderungen an das System abgeleitet. Die **späte Anforderungsanalyse** bringt eine Spezifizierung der Anforderungen hervor und ergänzt einen zusätzlichen Akteur. Dieser Akteur repräsentiert das System sowie die zwischen ihm und den anderen Teilnehmern notwendigen Beziehungen. In der **Architekturdesignphase** wird das Zusammenarbeitsmodell der Komponenten beschrieben. Dabei definiert Tropos Architekturstile für kooperative, dynamische und verteilte Anwendungen, um den Entwicklungsprozess zu unterstützen. In der Phase des **detaillierten Designs** werden die Komponenten des Systems weiter konkretisiert, indem die durch einzelne Agenten zu erfüllenden Ziele mit vordefinierten Designpattern hinterlegt werden. In der **Implementierungsphase** wird schließlich die Programmierung des Systems vorgenommen. Der Übergang zwischen den Entwicklungsstufen wird in Tropos durch spezielle Richtlinien unterstützt [StS14].

Stärken und Schwächen von Tropos

Die Umsetzung mentaler Konzepte geht bei Tropos weiter als bei den bereits vorgestellten Ansätzen, da es sowohl die Modellierung als auch die Notation von BDI zu einem größeren Grad unterstützt. Zudem deckt Tropos einen großen Teil des Entwicklungszyklus ab und ist unter dem Gesichtspunkt der Praxistauglichkeit mit Entwicklungswerkzeugen ausgestattet, mit denen die Spezifizierung, Analyse und Implementierung von Agentensystemen erleichtert wird [StS14].

Da die Methode ihren Ursprung im Bereich der Geschäftsprozesse hat, existieren jedoch nur wenige Anwendungen im Bereich der Automatisierungstechnik. Übertragungen auf die Fertigungsdomäne wie z.B. in [GMN03] und [AMS08] zielen darüber hinaus nicht per se auf die Fertigungssteuerung ab, sondern nehmen Geschäftsziele oder -risiken in den Fokus, sodass eine Adaption im Automatisierungsumfeld unwahrscheinlich ist. Für die Verwendung im Rahmen dieser Arbeit ist diese Methode daher nicht angemessen.

4.2.4 Prometheus (2002)

Übersicht Prometheus

Die Prometheus-Methode²⁵ wurde erstmalig 2002 durch Paghdam und Winikoff veröffentlicht [PaW02] und basiert ebenfalls auf einem OO-Ansatz [VDI12]. Die Methode besteht aus den drei Phasen Systemspezifikation, Architekturdesign und detailliertes Design (vgl. Abbildung 4-6).

²⁵ Die Bezeichnung Prometheus stammt aus der griechischen Mythologie. Dort wird Prometheus als Beschützer der Menschheit bezeichnet, der Zeus das Feuer gestohlen hat, um es den Menschen zu geben.

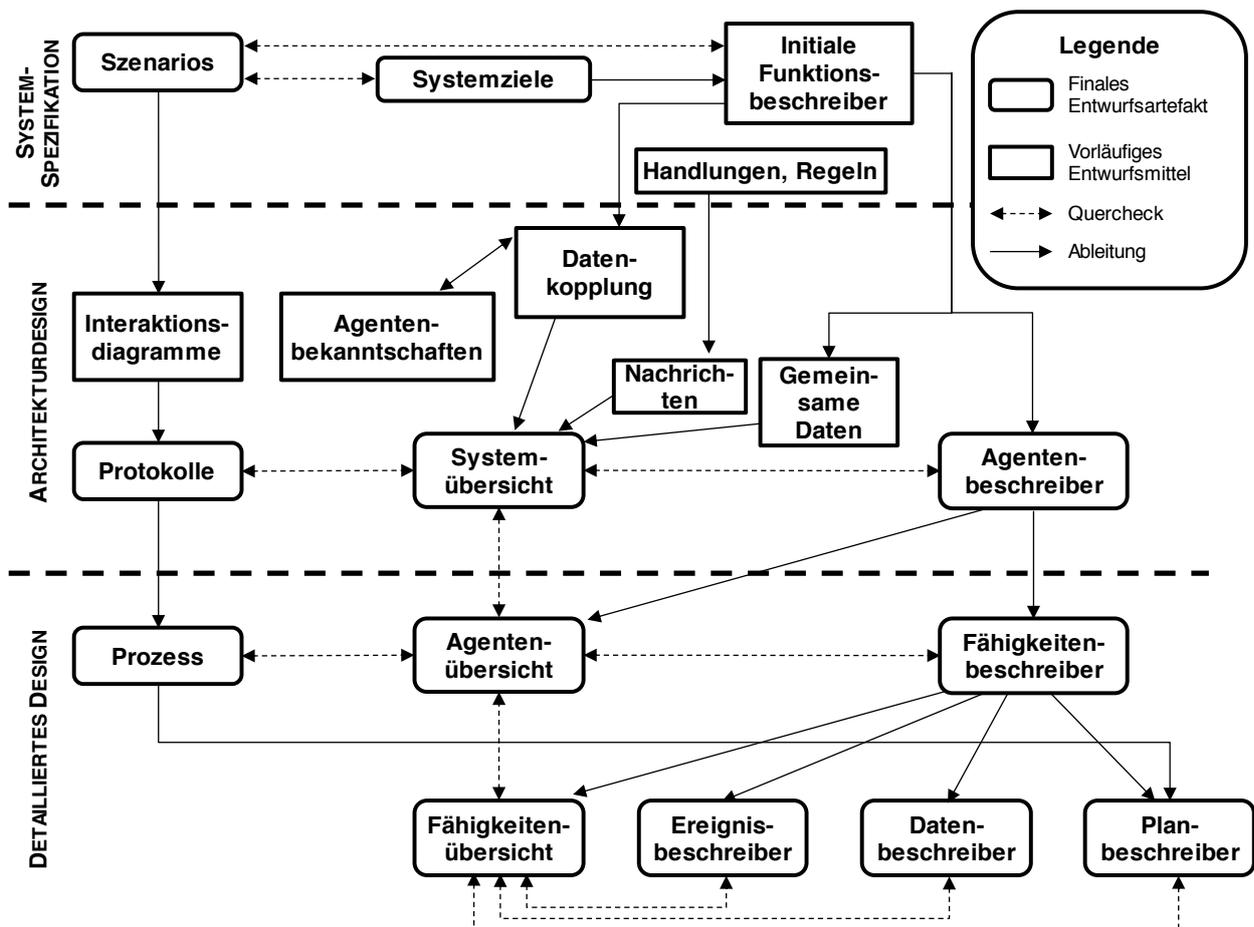


Abbildung 4-6: Phasen der Prometheus-Methode und ihre Beziehungen i.A.a. [PaW02]

In der Phase der **Systemspezifikation** werden mithilfe einer Szenarioanalyse die grundsätzlichen Funktionalitäten des Systems definiert. Dafür müssen zunächst die Systemziele identifiziert und Rollen zugeordnet werden. Darauf aufbauend werden die zur Realisierung der Ziele notwendigen Handlungen und Regeln betrachtet und ebenfalls Rollen zugewiesen [StS14]. In Phase zwei, dem **Architekturdesign**, werden die Agenten und ihre Interaktionen determiniert. Zunächst werden dafür die notwendigen Agententypen identifiziert und zusammen mit ihrem Lebenszyklus beschrieben. Ferner werden die Bekanntschaften zwischen Agenten sowie die Datenkopplung analysiert und auf dieser Basis Interaktionsprotokolle identifiziert. Als letzter Schritt dieser Phase wird eine Systemübersicht erstellt, die alle Agenten, Protokolle, Ereignisse, Handlungen und Daten enthält [StS14]. In der dritten Phase erfolgt schließlich das **detaillierte Design** des Systems. Hierbei werden die internen Eigenschaften der Agenten beschrieben, indem ihre Fähigkeiten, Pläne, internen Ereignisse und Daten spezifiziert werden. Dies geschieht mithilfe von Übersichtsdiagrammen für Agenten und Beschreibungen ihrer Fähigkeiten. Im Anschluss werden Prozessdiagramme und Interaktionsprotokolle erstellt sowie die Fähigkeiten ausdetailliert und umgesetzt [PaW02].

Stärken und Schwächen von Prometheus

Die Prometheus-Methode unterstützt, ähnlich wie Tropos, eine große Bandbreite an Konzepten und schließt auch mentale Konzepte wie BDI-Ansätze mit ein. Eine weitere Stärke ist die klare

Notation, die z.B. Modelle wie das Systemübersichtsdiagramm beinhaltet und damit das Systemverständnis verbessert. Prometheus deckt zudem einen großen Teil des Entwicklungsprozesses ab und geht dabei sogar auf Testing und Debugging ein. Lediglich MaSE geht noch weiter und berücksichtigt als weiteren Schritt das Deployment [DaW04]. Zu den Schwächen von Prometheus gehört die geringe Unterstützung von Entwicklern bei der Auswahl von Interaktionsprotokollen. Darüber hinaus ist die komplexe Terminologie, die u.a. zwischen Regeln, Störungen, Triggern und Ereignissen unterscheidet, für einige Nutzer verwirrend [DaW04]. Unter praktischen Gesichtspunkten ist positiv aufzuführen, dass Prometheus eine solide Unterstützung durch CASE-Werkzeuge bietet, die in Kursen unterrichtet und kontinuierlich weiterentwickelt werden [StS14]. Dies macht Prometheus zu einem geeigneten Kandidaten für die Verwendung in dieser Arbeit.

4.2.5 DACS (2004)

Übersicht DACS

Ein im Jahr 2004 veröffentlichter agentenorientierter Ansatz ist die DACS-Methode²⁶ von Bussmann et al. [BJW04]. Im Gegensatz zu anderen Methoden entspringt DACS direkt dem Anwendungsgebiet der Automatisierungstechnik, denn sie ist im Verlauf des in Kapitel 3.3.4 beschriebenen Daimler-Projektes Produktion 2000+ entwickelt worden. Ein weiterer Unterschied der Methode ist, dass das sonst übliche Rollenmodell durch ein Entscheidungsmodell ersetzt wird, das die notwendigen Steuerungsentscheidungen zusammenfasst und strukturiert [LüF15]. Die DACS-Methode beinhaltet die drei in Abbildung 4-7 dargestellten Phasen.

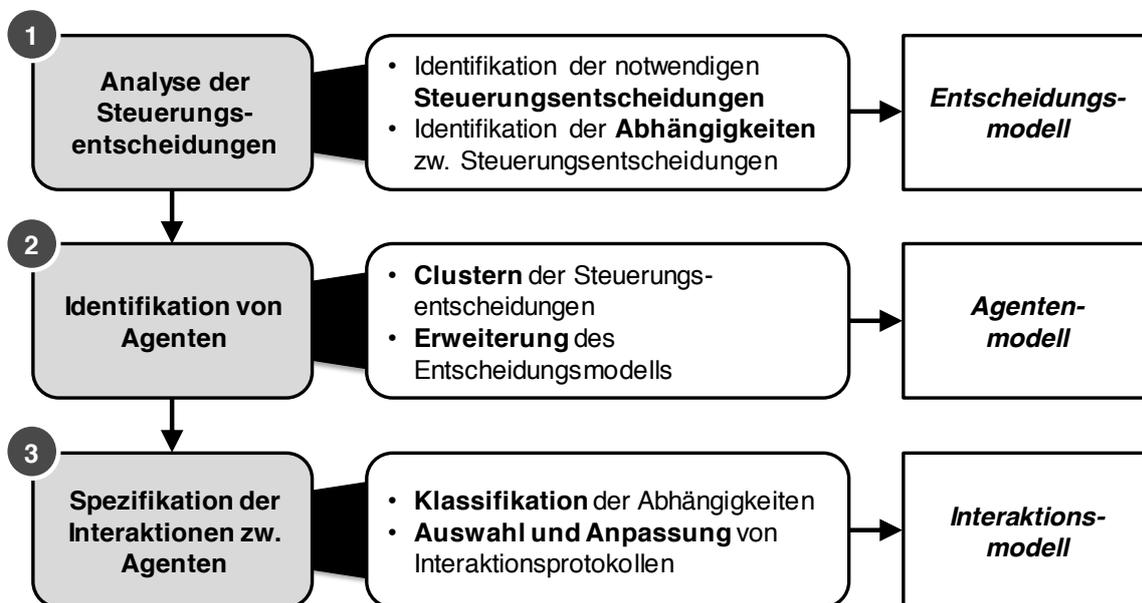


Abbildung 4-7: Phasen der DACS-Methode i.A.a. [LVG14]

In der ersten Phase erfolgt die Analyse und Beschreibung der **Steuerungsentscheidungen** im geplanten Produktionssystem. Dafür wird zunächst die zu steuernde Ebene der Steuerungspyra-

²⁶ DACS steht für Designing Agent-based Control Systems

mide identifiziert und untersucht, welche **Steuerungsentscheidungen** für die geforderten Funktionen notwendig sind. Als nächstes erfolgt die Identifikation von Abhängigkeiten zwischen den Steuerungsentscheidungen und es wird festgelegt, welche Informationen ausgetauscht werden müssen. Die Ergebnisse der ersten Phase führen zu einem **Entscheidungsmodell**. In der zweiten Phase erfolgt die **Identifikation von Agenten** im Produktionssystem, indem zunächst Steuerungsentscheidungen gruppiert und dann einzelnen Agenten zugeordnet werden. Anschließend wird definiert, welche Interaktionen zwischen den Agenten stattfinden müssen. Falls notwendig können Erweiterungen bzw. Anpassungen des Entscheidungsmodells vorgenommen werden. Die zweite Phase resultiert in einem **Agentenmodell**. In der dritten Phase erfolgt schließlich die **Spezifikation der Interaktionen zwischen den Agenten**. Dafür werden die sich aus den Abhängigkeiten ergebenden Interaktionsanforderungen erhoben und klassifiziert sowie mit passenden Interaktionsprotokollen aus einer Bibliothek abgeglichen. In einem letzten Schritt werden die am besten geeigneten Protokolle ausgewählt und bei Bedarf auf den konkreten Anwendungsfall angepasst. Als Ergebnis bringt die dritte Phase ein **Interaktionsmodell** hervor [BJW04, LÜF13].

Stärken und Schwächen von DACS

Im Gegensatz zu anderen Verfahren stellt die DACS-Methode keinen unabhängigen Ansatz im Softwareentwicklungsumfeld dar, sondern bettet sich in etablierte Praktiken der Softwareentwicklung ein [Wey10]. Dadurch ist DACS gut auf reale Probleme anwendbar und eignet sich insbesondere zur Lösung von Steuerungsproblemen in der Fertigung [BoG08]. Sie findet daher Anwendung in zahlreichen Projekten [DHL08, Wey10, VGL15]. Abseits der Fertigungsdomäne existieren ebenfalls Anwendungsgebiete. So kann eine sozio-technische Infrastruktur wie z.B. eine verteilte Stromerzeugung durch Aufnahme nicht-physischer Prozesse ebenso mit der DACS-Methode modelliert werden [DBH06]. Dadurch zeigt sich ihre universelle Anwendbarkeit. Die DACS-Methode selbst legt sich nicht auf spezifische Modellierungs- oder Notationstechniken fest, so dass in der praktischen Anwendung meist die üblichen UML-Methoden verwendet werden [VGL15]. Sie stellt jedoch eine simple Systematik zur Darstellung von Entscheidungen und den zwischen ihnen bestehenden Abhängigkeiten zur Verfügung (vgl. Anhang 9-Anhang 12).

Eine Schwäche der Methode ist die fehlende Spezifizierung der Implementierungsschritte, die zur Auswahl einer Agentenarchitektur und -plattform benötigt wird. Durch den starken Fokus auf Shopfloor-nahe Entscheidungen werden zudem andere Komponenten von Fertigungssystemen außer Acht gelassen [BoG08]. Vor diesem Hintergrund existieren in der Literatur Erweiterungen für die DACS-Methode. In der Arbeit von Lüder und Foehr [LüF15] wird DACS für die Entwicklung von mechatronisch-orientierten Agenten ausgebaut und Vogel-Heuser et al. [VGL15] erweitern die Methode, um die Integration der Feldebene zu verbessern. Durch zusätzliche Erweiterungen könnte sie zudem noch weiter aufgewertet werden. Vor diesem Hintergrund stellt die DACS-Methode ein für die Verwendung in dieser Arbeit geeignetes AOSE-Vorgehen dar.

4.2.6 ADMARMS (2015)

Übersicht ADMARMS

Als letzter Ansatz soll mit ADMARMS²⁷ eine neuere Methode vorgestellt werden, die 2015 durch Farid und Ribeiro publiziert wurde [FaR15]. Die ADMARMS-Methode ist aus dem Mangel an formalen, quantitativen AOSE-Methoden entstanden. Sie basiert auf einer etablierten Entwicklungsmethode, dem sogenannten „axiomatic design for large flexible engineering systems“ und stützt sich auf Entwicklungsprinzipien zur mathematischen Quantifizierung der Rekonfigurierbarkeit von Produktionssystemen. Als zentrales Axiom ist dabei stets die Unabhängigkeit der funktionalen Anforderungen zu bewahren. Darauf basierende Designentscheidungen finden auf einer hohen Ebene statt und können zu Einschränkungen auf tieferen Ebenen führen. Diese Einschränkungen grenzen die Produktionsfreiheitsgrade ein und können z.B. durch die Zuweisung von Prozessen zu Ressourcen entstehen. ADMARMS setzt auf den aus IEC 61499 bekannten und in [ZoP13] dokumentierten Vorgehen auf und beinhaltet fünf Phasen (vgl. Abbildung 4-8) [FaR15].

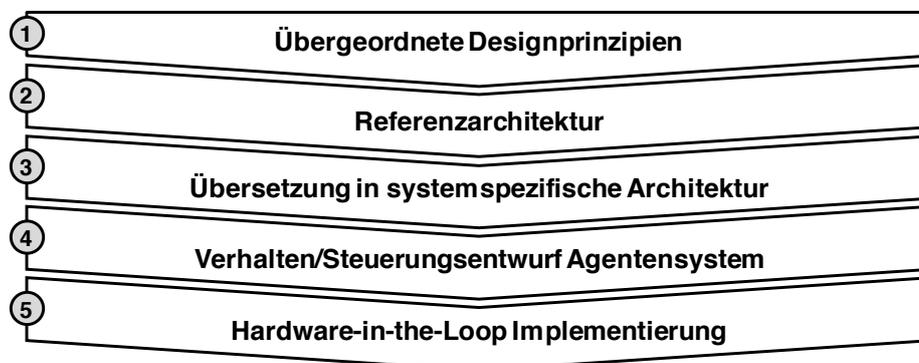


Abbildung 4-8: Phasen der ADMARMS-Methode i.A.a. [FaR15]

Die erste Phase legt zehn **übergeordnete Designprinzipien** fest. Diese Prinzipien geben die grundsätzlichen Richtlinien für das Systemdesign vor und behandeln Themen wie Heterogenität, Verfügbarkeit, Interaktion und Kapselung. Eine detaillierte Aufstellung ist in Anhang 13 beigefügt. In Phase zwei wird eine **Referenzarchitektur** vorgegeben, deren Anwendung für Systementwickler zwingend ist. Diese Referenzarchitektur, die eine Alternative zu den in Kapitel 4.1 vorgestellten Architekturen darstellt, gibt u.a. Agententypen, Produktmodelle und diverse Matrizen als Wissensdatenbank des Produktionssystems vor. In den Matrizen wird einerseits die Anlagenverfügbarkeit nachverfolgt. Andererseits werden darin die Interaktionen zwischen Ressourcenagenten und Produktagenten sowie alle physischen, informations- und energietechnischen Kopplungen zwischen Entitäten des Produktionssystems beschrieben. Die ADMARMS-Referenzarchitektur und das dazugehörige Datenmodell sind in Anhang 14 abgebildet. In der dritten Phase findet auf dieser Grundlage die Übersetzung der Referenzarchitektur in eine **systemspezifische Architektur** statt. Im Vergleich zu anderen Methoden soll dies unkompliziert realisierbar sein. Aufgabe der darauffolgenden vierten Phase ist es, eine formale **agentenbasierte Steuerung** zu entwerfen,

²⁷ Axiomatic Design of a Multi-Agent Reconfigurable Mechatronic System

durch die das gewünschte Verhalten und die geforderte Leistung des Produktionssystems erreicht werden können. Schließlich erfolgt in der fünften und letzten Phase eine **Hardware-in-the-Loop Implementierung**, mit der die Entwicklung der Kommunikation zwischen Agenten und der Hardware-Abstraktionsebene vereinfacht werden soll [FaR15].

Stärken und Schwächen von ADMARMS

Die ADMARMS-Methode ist der ungewöhnlichste der vorgestellten Ansätze. Ihr Mehrwert liegt vor allem in der extremen Systematisierung des Entwicklungsprozesses. Ferner ist ADMARMS durch die stärkste Ausprägung hinsichtlich der universellen Anwendbarkeit der Methode charakterisiert, was die Entwicklung von Agentensystemen zukünftig deutlich vereinfachen könnte. Zudem können mit der Methode potenzielle Designmängel frühzeitig identifizieren werden, noch bevor Daten zur Systemleistung zur Verfügung stehen.

Trotz der Intention der Autoren eine vergleichsweise einfache AOSE-Methode zu entwickeln wirkt das Konzept in Summe sehr abstrakt und schwerfällig, was für Entwickler potenziell abschreckend ist. Für die Übersetzung der Referenzarchitektur in eine systemspezifische Architektur werden des Weiteren keine ausreichenden Werkzeuge zur Verfügung gestellt. Der Fokus auf Rekonfigurierbarkeit lenkt zudem von anderen Aspekten wie z.B. Proaktivität und Autonomie ab. Diese Unzulänglichkeiten könnten der Grund dafür sein, dass die Methode in der Literatur bisher nicht stärker aufgegriffen wurde und dass derzeit keine weiteren Implementierungsfälle aus Forschung und Industrie bekannt sind. Zusammenfassend scheint die ADMARMS-Methode großes Potenzial zu besitzen, sie wirkt aktuell jedoch noch nicht ausgereift genug, weswegen sie sich zum jetzigen Zeitpunkt für die Verwendung in dieser Arbeit nicht eignet.

4.2.7 Bewertung und Auswahl einer Entwicklungsmethode

Der auf die Betrachtung der individuellen AOSE-Methoden folgende Schritt ist ihre gegenüberstellende Bewertung und die Auswahl einer Methode für die Verwendung in dieser Arbeit. Zur Bewertung wurden die Kriterien nach [DaW04] angesetzt und mithilfe der Bewertungsskala aus [CeZ11] in Zahlenwerte transformiert. Die so entstehende Bewertung ist in Tabelle 4-1 zusammengefasst. An den Stellen, an denen Bewertungen mehrerer Autoren für dieselbe Methode eingeflossen sind (vgl. Fußnoten in der Tabelle), wurden Durchschnittswerte gebildet, sodass die Bewertung Punkte von eins bis zehn inklusive Kommazahlen zulässt.

Hinsichtlich der unterstützten **Konzepte und Eigenschaften** sind die vorgestellten Ansätze alle auf einem sehr hohen Niveau. Die einzige Ausnahme stellt die ADMARMS-Methode dar, die aufgrund der strikten Vorgaben weniger Spielraum zulässt als die übrigen Ansätze. Durch zukünftige Arbeiten und Umsetzungen der Methode in Forschungs- und Industrieprojekten könnte diese Schwachstelle behoben werden.

In Bezug auf **Modellierung und Notation** zeigen sich deutlichere Abweichungen zwischen den Methoden. So ist Gaia aufgrund der Verwendung unüblicher Notationstechniken weniger zugänglich als MaSE. Trotzdem weißt MaSE aufgrund der vielen Phasen, Modelle und Verbindungen zwischen den Entitäten eine hohe Komplexität auf, die die Adaption hemmen. Deutlich stärker sind in diesem Bereich die Methoden DACS und Prometheus aufgestellt, die aufgrund der Klarheit der zugrundeliegenden Modelle und Verbindungen die Komplexität reduzieren und mit üblichen Notationstechniken eine steile Lernkurve ermöglichen.

Bei der Abdeckung des **Entwicklungsprozesses** bewegen sich die vorgestellten Methoden im Mittelfeld. Hier geht die MaSE-Methode am weitesten, erfüllt aber dennoch nicht alle Anforderungen gemäß der Kriterien von [DaW04]. Da eine vollständige Abdeckung des Entwicklungsprozesses bis in die Implementierungs- und Instandhaltungsphase zwar wünschenswert, aber nicht zwingend erforderlich ist, wird dies nicht als schwerwiegender Mangel gewertet.

Kriterium	Gaia	MaSE ²⁸	Tropos ²⁸	Prometheus ²⁹	DACS	ADMARMS
Konzepte & Eigenschaften						
• Autonomie	10	10	10	10	10	5
• Reaktivität	8	6	6	8,5	8	10
• Proaktivität	10	10	10	8,5	10	7
• Mentale Konzepte	7	7	7	9	7	7
• Teamwork	10	6	6	3	7	5
• Protokolle	7	8,5	7	8	10	5
Modellierung & Notation						
• Konsistenz	10	9	4	8,7	10	10
• Komplexitätsmanagement	5	6,5	7,5	10	10	3
• Ausdruckskraft	3	7,5	6,5	8	8	8
• Erschließbarkeit	3	6	6	8,9	10	3
Prozess						
• Abdeckung Entwicklungszyklus	7	7,5	6	5	7	7
Praxistauglichkeit						
• Domänenübergreifende Anwendbarkeit	10	10	10	10	10	10
• Skalierbarkeit	7	4	4	6	5	10
• Reife	5	5	4	7,7	5	4
Total	102	103	94	111	117	94

Tabelle 4-1: Evaluation von AOSE-Methoden

Hinsichtlich der **Praxistauglichkeit** ist festzuhalten, dass alle vorgestellten Methoden prinzipiell eine domänenübergreifende Anwendbarkeit zulassen. In Bezug auf Skalierbarkeit und Reife existieren jedoch große Unterschiede. Bei der Skalierbarkeit kann ADMARMS mit seiner Gradlinigkeit

²⁸ Mittelwerte von Dam & Winkow sowie Shehory et al. aus [CeZ11]

²⁹ Werte aus [DaW04]

punkten und sich gegen etabliertere Methoden wie Gaia und Prometheus durchsetzen. Gleichzeitig schneiden Tropos und ADMARMS wegen der geringen Anzahl an Umsetzungen im Fertigungsumfeld bezüglich der Reife schlecht ab. Ein Kriterium bei dem Prometheus besser dasteht.

Als Ergebnis der Gegenüberstellung eignen sich die beiden Methoden DACS und Prometheus am besten für die Verwendung in dieser Arbeit. Vorteilhaft für die DACS-Methode ist die im Steuerungsumfeld nützliche Fokussierung auf Steuerungsentscheidungen, die Klarheit und Einfachheit des Entwicklungsprozesses und der Notationstechniken sowie die Unterstützung der Protokollauswahl mithilfe einer Bibliothek. Darüber hinaus hat sich die Nützlichkeit der DACS-Methode auch in der empirischen Erfahrung des Autors bestätigt. Durch Erweiterung des Ansatzes hinsichtlich Skalierbarkeit und Reifegrad kann die Methode zudem weiter ausgebaut werden. Aus diesen Gründen wird für die weitere Verwendung im Rahmen dieser Arbeit die **DACS-Methode als designierter AOSE-Ansatz** gewählt.

Der erste Schritt bei der Entwicklung von Agentensystemen mit der DACS-Methode ist die Analyse der Steuerungsentscheidungen im betrachteten System. Um diesen Schritt so einfach und umfassend wie möglich zu gestalten, ist es sinnvoll über ein gutes Verständnis der Steuerungsentscheidungen und ihrer hierarchischen Abhängigkeiten zu verfügen. Ziel des nächsten Kapitels ist es daher eine Übersicht von Steuerungsentscheidungen innerhalb einer generischen Fertigungssystemstruktur zu erarbeiten.

5 Steuerungsaufgaben in Produktionssystemen

Das in Kapitel 1 vorgestellte RAMI 4.0 gibt grobe Richtlinien für die in modernen und vernetzten Produktionssystemen zu berücksichtigenden Elemente vor und beschreibt ihre Abhängigkeiten. Kernelement der Architektur sind I4.0-Komponenten, die mehrere Gegenstände eines Systems bündeln und koordiniert ansteuern können. Die Verwaltungsschale dieser I4.0-Komponenten kann durch Agenten realisiert werden [BMK16]. In einem Agentensystem würden Agenten so die virtuelle Repräsentation der Komponenten darstellen und ihre fachlichen Funktionalitäten anderen Agenten zur Verfügung stellen. Dafür benötigen sie jedoch Zugriff auf alle Informationen der Komponente. Die im RAMI 4.0 definierten Ebenen sind auf Basis eines Konsens der Interessen von verschiedenen Branchen der Prozess- und Fertigungsindustrie entstanden, die unterschiedliche Standards, Informations- und Kommunikations- sowie Automatisierungstechnologien verwenden [ZVE15]. Sie bilden daher die prozess- und steuerungstechnisch notwendigen Ebenen eines Produktionssystems nur unzureichend ab. Es ist folglich nicht im Detail geklärt, wie sich Bestandteile von Produktionssystemen in sinnvolle I4.0-Komponenten gruppieren lassen und über welche Informationen die jeweiligen I4.0-Komponenten verfügen müssen. Dies sind jedoch notwendige Voraussetzungen für die Agentifizierung und daher Inhalt der Forschungsfragen 1 und 2. Im Rahmen der Erstellung dieser Arbeit wurden diese Herausforderungen daher unter federführender Beteiligung des Autors in der Veröffentlichung [ZHR16] bearbeitet. Die Ergebnisse der Forschungsarbeit werden im Folgenden zusammengefasst. In Abschnitt 5.1 wird zunächst eine generische Struktur von Fertigungssystemen bereitgestellt, die bestehende Forschungsarbeiten zusammenfasst und auf deren Basis eine sinnvolle Gruppierung von I4.0-Komponenten erfolgen kann. Anschließend wird in Abschnitt 5.2 gezeigt wie diese Struktur genutzt werden kann, um eine Zuordnung der notwendigen Informationsarten auf allen Ebenen zu realisieren.

5.1 Generische Fertigungssystemstruktur

Die Gruppierung von I4.0-Komponenten bedarf einer ausreichend generischen Grundlage. Die Entwicklung einer allgemeingültigen Produktionssystemstruktur für diese Arbeit erscheint wegen des breiten Spektrums an Anwendungsgebieten jedoch nicht zielführend, da sie zulasten des Informationsgehalts und der Nützlichkeit für die verfolgte Zielstellung gehen würde. Daher wurde als Betrachtungsraum die Domäne der Fertigungsindustrie und genauer – aufgrund der hohen Relevanz von Flexibilität – die Variantenfließfertigung gewählt. Zudem wurden vorrangig steuerungsrelevante Informationen betrachtet, die in der Inbetriebnahme- und Nutzungsphase notwendig sind. Für Betrachtungen der Informationen in der Entwicklungs- und End-of-Life-Phase wird auf die Arbeiten von [HHL16] und [SLH16] verwiesen.

Als erster Schritt zur Erstellung einer generischen Fertigungssystemstruktur wurde eine detaillierte Literaturanalyse zur Strukturierung von Fertigungssystemen durchgeführt. Zu diesem Themengebiet existiert eine große Vielzahl an Publikationen, die nicht alle im Rahmen dieser Arbeit

vorgestellt werden können. Stattdessen wird kurz auf die Vertreter von drei wesentlichen Strömungen eingegangen, in die ein Großteil der Veröffentlichungen eingeordnet werden kann.

In der ersten Gruppe von Forschungsarbeiten, zu dessen Vertretern [WEN07], [Sch07] und [WaW14] zählen, werden Fertigungssysteme aus der Sicht **ganzer Unternehmen** bzw. **Fabriken** betrachtet. Die Fertigungssysteme werden in Ebenen wie Unternehmensnetzwerk, Unternehmen, Standort, Segment, Produktionsanlage, Zelle und Station untergliedert. Hier werden vordergründig die verschiedenen Hierarchiestufen von Fertigungsressourcen adressiert. Zu der zweiten Gruppe gehören u.a. die Werke von [PSA14], [Kie07a] und [Mel13]. Ihre Arbeiten beziehen sich auf technologische **Fertigungsfunktionen** und die zu ihrer Ausführung direkt oder indirekt notwendigen **Systemkomponenten**. Die so entstehenden Fertigungssystemhierarchien enthalten Ebenen wie Fertigungszelle, Hauptfunktionsgruppe, Funktionsgruppe und Teilfunktionsgruppe. Die dritte Gruppe von Veröffentlichung legt einen stärkeren Fokus auf die **individuellen Komponenten** von Fertigungssystemen, die sowohl das physische Verhalten als auch deren Integration realisieren. Vertreter dieser Strömung sind [WSW11], [SiW11] und [LMB09]. Im Zentrum dieser Forschungsarbeiten stehen Automatisierungsgeräte und mechatronische Systeme. Dies resultiert in Ebenen wie Funktionseinheiten, Geräten und Gerätefunktionen.

Die gemeinsame Betrachtung der drei Forschungsströmungen ermöglicht die Identifizierung von Hierarchieebenen, die die gesamte Bandbreite vom Unternehmen bis zu einzelnen Automatisierungsgeräten und mechanischen Bauteilen beinhalten. Ergänzend zur Literaturanalyse wurden bestehende Fertigungssysteme in der Automobilindustrie in die Analyse einbezogen und anhand der Kriterien technische Funktionalität, Modularität, Steuerungsarchitektur, menschliche Arbeit, Relevanz in den verschiedenen Phasen des Entwurfsprozesses sowie Relation zur Produktstruktur bewertet. Auf dieser Grundlage wurden weitere Fertigungssysteme aus verschiedenen Industriebereichen analysiert und die Struktur in mehreren Iterationsschleifen verfeinert. Das Ergebnis ist die in Tabelle 5-1 dargestellte generische Fertigungssystemstruktur (GFSS).

Ebene	Presswerk	Karobau	Lackiererei	Montage	Förder- technik	Küvetten- fertigung	Warmband- straße	GF	FM	Ref
9. Fertigungs- netzwerk	Automobilunternehmen mit Zulieferern					Medizintech- nikuntern. mit Zulieferern	Stahlunterneh- men mit Zulie- ferern	x		
8. Fabrik	Werk an Standort					Werk an Standort	Werk an Standort		x	
7. Fertigungs- linie	Pressen- straße	Karosserie- bauscheibe	Lackierlinie	Montagelinie	Förderlinie (z.B. Montage)	Küvetten- fertigung	Warmband- straße	x	x	
6. Fertigungs- abschnitt	Presse	Aufbau	Decklack	Teilmontage	-/-	-/-	Walzstraße	x	x	
5. Arbeits- einheit	Werkzeug- satz	Aufbauteil	Clearcoat- linie	Dachmontage	Förder- strecke	Abfüllanlage	Walz- und Kühlanlage	x	x	
4. Arbeits- station	Werkzeug	Schweißzelle	Roboterzelle	Klebestation	Takt	Füllstation	Walzenstuhl		x	x
3. Funktions- gruppe	Werkzeugteil	Roboter mit Schweißzange	Roboter mit Spritzpistole	Roboter mit Klebepistole	Schubskid- plattform	Positionier-/ Dosiereinheit mit Pipette	Walzengruppe			x
2. Kompo- nente	Wirkfläche	Schweißzange	Spritzpistole	Doppelfass- pumpe	Hubtisch	Dosiereinheit	Antriebsstrang			x
1. Konstrukti- onselement	Werkzeug- bauteil	Schweißkontakt	Glocke	Düse	Karosserie- aufnahme	Pipette	Walze			x

Tabelle 5-1: Generische Fertigungssystemstruktur und Praxisbeispiele i.A.a. [ZHR16]

Die funktionalitätsbezogenen Eigenschaften der Ebenen definieren sich wie folgt: Ein **Konstruktionselement** (KE) ist ein mechanisches, elektrisches oder elektronisches Bauteil, das selbst keine eigene Funktion erbringt. Es ist allerdings für die Funktionalität der Komponente notwendig, in die es integriert ist. Folglich setzt sich eine **Komponente** aus KEs zusammen und kann eine wertschöpfende oder eine die Wertschöpfung unterstützende Funktion erbringen. Komponenten, die für die Ausführung eines Fertigungsschritts notwendig sind, werden auf der nächsten Ebene zu **Funktionsgruppen** (FGs) zusammengefasst. FGs können verschiedene Fertigungstechnologien enthalten und sie in einer festen Kombination nutzbar machen (z.B. Halten und Schweißen). Durch Verknüpfung von Funktionen wird auf der nächsten Ebene der **Arbeitsstationen** (ASs) die Realisierung bestimmter Produkteigenschaften bzw. -qualitäten gewährleistet. Dafür werden eine oder mehrere Funktionsgruppen so integriert, dass sie eine oder mehrere wertschöpfende und/oder unterstützende Funktionen im Rahmen des Fertigungsprozesses erbringen können. Durch die Verknüpfung mehrerer ASs mit einer gewissen Menge an wertschöpfenden und/oder unterstützenden Funktionen entstehen **Arbeitseinheiten** (AEs). Sie bilden eine geschlossene Prozesskette aus sinnvoll miteinander kombinierten zeitlich abhängigen Teilprozessen. Die nächsthöhere Ebene der **Fertigungsabschnitte** (FAs) wird durch eine Verkettung einer Vielzahl von wertschöpfenden und nichtwertschöpfenden Funktionen charakterisiert, bei der eine Menge von AEs über Puffer gekoppelt werden. Weiter oben in der Hierarchie ermöglichen **Fertigungslinien** (FLs) die Abgrenzung zwischen verschiedenen fertigungstechnischen Disziplinen (z.B. Presswerk, Karosseriebau, Lackiererei und Endmontage). Diese können aufgrund unterschiedlicher Anforderungen hinsichtlich der Fertigungstechnologien (Urformen, Umformen, Fügen etc.) oder der Fertigungsmengen notwendig sein. Die obersten beiden Ebenen, **Fabrik** und **Fertigungsnetzwerk**, werden durch die Kombination aller fertigungstechnischen, logistischen und sonstigen unterstützenden Funktionen definiert, die für die Erstellung eines Produktportfolios mit einer festen Menge an Produkteigenschaften benötigt werden. In Abhängigkeit der Herkunft der Inputfaktoren, der geografischen Verteilung und der Besitzverhältnisse können Fabriken und Fertigungsnetzwerke unterschieden werden.

Zur Validierung der erstellten generischen Fertigungssystemstruktur (GFSS) wurde eine Menge studentischer Forschungsarbeiten realisiert, in denen Fertigungssysteme aus verschiedenen Industriebereichen partiell modelliert und bewertet wurden. Der Betrachtungsumfang umfasste u.a. Presswerk, Karosseriebau, Lackiererei, Endmontage und Intralogistik aus dem **Automobilbau**, eine Warmbandstraße und einen Querschneider aus der **Stahlindustrie**, einen Steinbrecher aus der **Bergbauindustrie**, ein Fertigungssystem für Dachstühle aus dem Bereich der **Holzindustrie**, eine Mikroküvettenfertigung aus der **Medizintechnik**, ein **Logistikzentrum** sowie ein Solarpark und eine Gasturbinenfertigung aus dem Bereich der **Stromerzeugung**.

Basierend auf dieser Validierung sind die Autoren von der breiten Anwendbarkeit der entwickelten Struktur überzeugt. Eine Gegenüberstellung mit den Ansätzen von ISA 95 und RAMI 4.0 in Anhang 15 zeigt zudem ihre vergleichsweise hohe Detaillierungstiefe. Im Sinne der ersten Forschungsfrage sind mit der GFSS die relevanten Ebenen für Industrie 4.0-Komponenten definiert, sodass die erarbeitete Struktur zur Beantwortung der zweiten Forschungsfrage genutzt werden kann.

5.2 Steuerungsaufgaben und -informationen

Die Ausführung komplexer Fertigungsprozesse mithilfe von I4.0-Komponenten erfordert ebenenspezifische Informationen, die in der Verwaltungsschale abgebildet werden müssen. Zur Beantwortung der zweiten Forschungsfrage wurden daher im Rahmen der diskutierten Forschungsarbeit [ZHR16] generische Steuerungsaufgaben und -informationen identifiziert und den Hierarchieebenen der oben vorgestellten Fertigungssystemstruktur zugeordnet. Die Betrachtung umfasst die relevanten **Steuerungssysteme aller Hierarchieebenen**, d.h. Systeme für die Ansteuerung von Sensoren/Aktoren und Prozessteuerungen auf Feldebene, Systeme zur Allokation und zum Monitoring von Fertigungsressourcen auf MES-Ebene sowie Warenwirtschafts- und Unternehmensleitsysteme auf ERP-Ebene [Lüd06]. Zudem wurden die unterschiedlichen Aspekte der **Automatisierungspyramide** nach [DLH11] und [VDB13] sowie die applizierbaren **Implementierungstechnologien** für Steuerungssysteme nach [Kie07b], [Lun08], [KiD14], [YVP14] und [VGL15] in den Betrachtungsumfang einbezogen.

Für die Identifizierung der Steuerungsaufgaben und -informationen wurde, in Analogie zur obigen GFSS, zunächst eine **Literaturanalyse** und anschließend eine Untersuchung von **Steuerungssystemen aus der Praxis** durchgeführt. Die Literaturanalyse schließt u.a. die Werke [WeB06], [Löd08], [ZVE10], [Her12], [ScS12] und [Dör13] ein und führte in Kombination mit der Validierung realer Fertigungssysteme zu der in Tabelle 5-2 abgebildeten Zuordnung. In Anlehnung an die Veröffentlichung [ZHR16] des Autors verfügen die Steuerungsaufgaben und -informationen auf den einzelnen Ebenen über die im Folgenden beschriebenen Eigenschaften. Die Beispiele repräsentieren typische Steuerungsumfänge der Automobilindustrie als Vertreter der Variantenfließfertigung und bilden wesentliche Umfänge des Auftragsdurchlaufs ab.

Auf Ebene der **Konstruktionselemente** wird zwischen aktiven und passiven KEs unterschieden. Während aktive KEs einen gewissen Aktivitätsstatus repräsentieren und an Elemente im System kommunizieren können, sind passive KEs dazu nicht in der Lage. Zudem können aktive KEs in Abhängigkeit ihrer Funktionalität verschiedene Zustände annehmen (z.B. Motor ein-/ausgeschaltet) und Regler enthalten (z.B. Zweipunktregler bei Thermostaten). Da eine KE-interne Kommunikation nicht möglich ist, enthalten KEs keine weiteren Steuerungsfunktionalitäten oder -parameter.

Ebene	Steuerungsaufgaben	Steuerungsinformationen
9. Fertigungsnetzwerk	Langfr. Absatzplanung, Volumenplanung Primärbedarf, Produktionsprogrammplanung, Werkbelegung	Produktvolumen, Investitions- u. Kapazitätsplanung, Fabrikkosten
	Volumenplanung Eigenschaften, Brutto-Sekundärbedarfsrechnung, Bedarfs-Kapazitäts-Abgleich u. Engpassmanagement	Einbauraten, interne u. externe Kapazitäten, Kosten u. Preise sowie KPIs, Wirtschaftlichkeitsrechnung
	Produktdatenmanagement, Auftragsmanagement, Baubarkeitsprüfung	Teile-Nr., Teile-Status, Primäreigenschafts-Nr., technische Regeln, Produkthierarchien, Vertriebs-Regeln
8. Fabrik	Netto-Sekundärbedarfsermittlung, Vorschau Lieferabruf	Fahrzeugprogramm, Sperrungen (Technik/Vertrieb), Baubarkeit, Teilebestand /-verfügbarkeit
	Wochen-/Tagesprogrammbildung Feinplanung Materialbeschaffung u. Disposition, Frachtkostenoptimierung, Auftragsfreigabe, Fertigungssequenz der Werke	Gewerkerestriktionen, Produktmix, Linienzuordnung, Kapitalbindungskosten, Fabrik-Bereitschaft
	Fabrik-Monitoring & -Analyse, virt. Resequenzierung	Fertigungsrelevante Auftragsdaten, Fabrik-Kennzahlen
7. Fertigungslinie	Auslastungsvorschau und Taktsimulation	Personalbedarf, Schichtplanung
	Lieferabrufe: intern u. externe Feinabrufe, produktionsasynchrone Abrufe von JIT- und JIS-Umfängen	Plan-Bedarf, Materialbestand, Steuerzeiten für Lieferanten, Fahrzeug- und Auftrags-Status, Auftragsdaten
	Fertigungslinien-Monitoring (ggf. HMI)	Soll- und Ist-Stückzahlen, Anlagenstatus (verfügbar/ge-stört), Pufferstände für Produktvarianten
6. Fertigungsabschnitt	Betriebsmittelsteuerung und -überwachung (HMI)	Taktzeit-Controlling (Bewegungs-, Prozess-, Nebenprozesszeiten), Verfügbarkeit & Nutzungsgrad verketteter Anlagen, autom. TPM-Alarm bei Prozessabweichungen, gegenüberstellender Vergleich Energieverbrauch
	Materialfluss zwischen Arbeitseinheiten, Kommissionierung Sequenz-/Warenkorbumfänge, Tourenplanung und Transportsteuerung, physische Resequenzierung	Fertigungssequenz in Abhängigkeit der Produktvariante, Abweichung Ist-/Soll-Position, Perlenkettengüte
	Materialbestandsführung und Lagerabrufe	Meldungen Warenein-/ausgang, Umlaufbestände, WIP
5. Arbeitseinheit	Anlagenführung und -bedienung (HMI)	Taktzeiten, Variantenkonfiguration, Wartungszyklen, Statusinformationen
	Betriebs-/Maschinendatenerfassung (BDE/MDE)	Ausbringung, Einzel-Verfügbarkeits-KPIs, Störungsauswertung, Energieverbrauch, NC-Daten, Werkzeugdaten
	Visualisierung auf Ebene des ausgeführten Prozesses	Zustands- und Fehlermeldungen, Füllstände, Takt- und Prozesszeiten
4. Arbeitsstation	Prozess-Koordination und -Überwachung, umfasst <ul style="list-style-type: none"> ▪ simult. Zustandserfassung mehrerer Funktionsgr. ▪ simult. Ansteuerung mehrerer Funktionsgruppen 	Zustandsinformationen Steuerbefehle
	Qualitätsprüfung u. -datenerfassung, Bauzustandsdokumentation	Qualitätsdaten
	HMI	Programmabrufe und manuelle Steuerungsbefehle
3. Funktionsgruppe	Produkt-Identifikation	Produkt-ID-Daten: Kenn-Nummer, Auftragsnummer
	Parametrisierung von Prozess-Komponenten	Prozessparameter
	Condition Monitoring und Asset-Management	Interpretierte Sensor-/Aktor-Daten
	Funktionsspezifische HMI	Robotersteuerung: Trajektorien, Start- und Haltepunkte, Anfah- und Brems-Geschwindigkeiten
2. Komponente	Sensor-/Aktor-Steuerung	Sensor-/Aktor-Daten
	Schaltvorgänge	Ein- und Ausschaltbefehle
	Condition Monitoring und Asset-Management	Prozesskomponenten-KPIs
1. Konstruktionselement	Repräsentation eines Aktivitätsstatus, der an umliegende Systeme kommuniziert werden kann, keine komponenteninterne Kommunikation	keine Steuerungsinformationen

Tabelle 5-2: Steuerungsaufgaben und -informationen in der GFSS i.A.a. [ZHR16]

Auf Komponentenebene können **Steuerungskomponenten (SK)** und **Prozesskomponenten (PK)** unterschieden werden. SKs übernehmen die Verarbeitung und Übermittlung von Informationen und ermöglichen damit grundlegende Steuerungsfunktionen für aktive KEs. PKs verfügen über einen vordefinierten Funktionsumfang und verarbeiten zu deren Ausführung Ein- und Ausgangsinformationen. Sie können daher parametrisiert und bedarfsgerecht angepasst werden. Typische Steuerungsaufgaben von PKs sind Sensor-/Aktor-Steuerung, Schaltvorgänge, Condition-Monitoring und Asset-Management.

Die Koordination der Komponenten erfolgt auf Ebene der **Funktionsgruppen**. Sie erbringen feste, parametrierbare Prozesse sowie Überwachungs- und Diagnosefunktionen, die durch eine Mensch-Maschine-Schnittstelle (HMI) angepasst werden können. Prozessparameter wie z.B. das Kleberaufenvolumen einer Klebeapplikation in dm^3/m oder der Anpressdruck eines Sauggreifers in N/mm^2 gehören zu den relevanten Steuerungsinformationen dieser Ebene. Da es produktspezifische Unterschiede gibt, bedarf es zudem gewisser Produktinformationen (z.B. Auftragsnummern) sowie interpretierter Sensor-/Aktordaten (z.B. Schwingungen von Wälzlagern oder Öl-Analysen eines Ziehwerkzeugunterteils).

Die Orchestrierung der FGs erfolgt auf der darüberliegenden Ebene der **Arbeitsstationen**. Sie führen steuerungstechnisch festgelegte, aber parametrierbare und produktspezifische Prozesse zur Umsetzung einer definierten Funktionsmenge aus. Dies bedarf einer simultanen Zustandserfassung und Ansteuerung mehrerer FGs. Das Setzen eines Schweißpunktes im Karosseriebau benötigt z.B. die simultane Bereitstellung von Zustandsinformationen (z.B. Spannelemente geschlossen, Schweißzangen an vorgesehener Stelle) und Steuerungsbefehlen (z.B. Spanner geschlossen halten, Schweißstrom einleiten). Weitere Funktionen schließen die Prüfung und Dokumentation von Qualitätsmerkmalen ein, bei denen Qualitätsdaten wie z.B. Verschraubungs- oder Karosseriegeometriedaten entstehen und in eine Bauzustandsdokumentation einfließen können. Bei Bedarf können Steuerungsbefehle und Programme auf der AS-Ebene (z.B. Roboter bei Störung auf Nullposition setzen) über HMIs auch manuell ausgeführt werden.

Auf Ebene der **Arbeitseinheiten** erfolgt die Anlagenführung und -bedienung mittels HMI sowie die Realisierung der produktseitig definierten Prozessabfolge. Zudem setzen AEs Überwachungsfunktionen wie z.B. das Erfassen von Betriebs- und Maschinendaten um. Aus der Analyse dieser Daten entstehen Kennwerte wie Ausbringung, Verfügbarkeit und Energieverbrauch. Ferner können Zustands- und Fehlermeldungen (z.B. Schweißzange bereit/gestört, Lichtschranke geöffnet/unterbrochen), Füllstände sowie Takt- und Prozesszeiten erhoben und weiterführende Analysen (z.B. Störungsauswertung/Benchmarks) erstellt und visualisiert werden.

Übergeordnete Steuerungsentscheidungen, zu denen die operative Betriebsmittelsteuerung und -überwachung, die Materialbestandsführung und Materialflusssteuerung sowie das Auslösen von Lagerabrufen gehört, werden auf Ebene der **Fertigungsabschnitte** gesteuert. Dabei entstehen Informationen zum Controlling von Taktzeiten, Verfügbarkeiten und Nutzungsgraden verketteter

Anlagen sowie übergeordnete Benchmarks (z.B. Vergleich des Energieverbrauchs). Im Rahmen der Materialflusssteuerung wird der Materialstrom zwischen den AEs sowie die Kommissionierung von Sequenz-/Warenkörben, die Tourenplanung, Transportsteuerung und physische Resequenzierung von Aufträgen koordiniert. Dafür werden Informationen zu Fertigungssequenzen (z.B. Füge-/Montagefolge) sowie Soll- und Ist-Reihenfolgen benötigt. Ferner werden in der Materialbestandsführung und bei Lagerabrufen z.B. Informationen zu Warenein- und -ausgängen, Umlaufbeständen und unfertigen Erzeugnissen generiert.

Auf der nächsthöheren Ebene der **Fertigungslinien** erfolgen operative Planungsfunktionen wie die Auslastungsvorschau und Personalplanung sowie externe Materialabrufe und das Monitoring untergeordneter Ressourcen und Kennzahlen. Für Feinabrufe werden dabei die kurzfristigen Planbedarfe für Material und Komponenten sowie die voraussichtlichen Materialbestände mit einem Vorlauf von ein bis zwei Wochen benötigt. Produktionssynchrone Abrufe (auch JIT-Abrufe genannt) hängen hingegen vom Status des zugrundeliegenden Auftrags, seinen Spezifikationen und den Steuerzeiten der Lieferanten ab. Das FL-Monitoring umfasst Informationen zu Soll- und Ist-Stückzahlen, dem groben Anlagenstatus (verfügbar/gestört) sowie den Pufferständen für Produktvarianten.

Auf Ebene von **Fabriken** erfolgen drei Arten von Steuerungsaufgaben: 1) Taktisch-planerische Aufgaben zur Sicherstellung der Teilverfügbarkeit. Sie haben einen Planungshorizont von Wochen bis hin zu Monaten und benötigen dafür Informationen wie z.B. das Fahrzeugprogramm, technische und vertriebsseitige Sperrungen, Baubarkeitsprüfungen sowie Teilebestände und -verfügbarkeiten. 2) Operativ-vorbereitende bzw. kurzfristige Produktionsplanungsaufgaben zur Realisierung der Produktionsaufträge. Sie haben einen Horizont von Stunden bis hin zu Tagen und beinhalten u.a. das Herunterbrechen des Produktionsprogramms auf Tage und Schichten, die Feinplanung der Materialbeschaffung sowie die Auftragsfreigabe und Sequenzierung. Zur Ausführung dieser Aufgaben muss u.a. auf Informationen zu Gewerkerestriktionen, dem Produktmix und der Linienzuordnung zugegriffen werden. 3) Operativ-überwachende Aufgaben zur Sicherstellung der Produktionseffektivität und -effizienz. Dies beinhaltet fertigungsrelevante Auftragsdaten (z.B. Produktvariante, Ausstattungskombinationen) sowie diverse Fertigungskennzahlen (z.B. Durchlaufzeit, Liefertreue, Ausschuss).

Schließlich erfolgt auf Ebene des **Fertigungsnetzwerks** die Ausführung von strategischen und taktischen Fertigungsaufgaben wie z.B. die Festlegung der Absatz- und Produktionsprogramme sowie der Werkbelegung, die Kapazitätsplanung und Materialbedarfsermittlung, das Produktdaten- und Auftragsmanagement sowie die Baubarkeitsprüfung. Als Basis dafür dienen diverse Informationen, vom Produktvolumen über Einbauraten und Wirtschaftlichkeitsrechnungen bis hin zu Teilenummern sowie technischen und vertriebsseitigen Regeln.

Mithilfe der erarbeiteten Zuordnung von Steuerungsaufgaben und -informationen zu Ebenen der GFSS können die Verwaltungsschalen von I4.0-Komponenten einfacher und genauer definiert

werden. Ferner lassen sich auf dieser Grundlage auch die Interaktionen zwischen I4.0-Komponenten modellieren, was eine wichtige Aufgabe bei der Agentifizierung ist. Auf der Struktur aufbauende Interaktionsbeispiele können in [ZHR16] und [LSH17a] nachgeschlagen werden. Weitere Beispiele sind in Anhang 16 bis 20 beigefügt. Sie beinhalten einerseits einen Feinabruf eines lieferantenspezifischen Nettobedarfs sowie einen JIT-Abbruch beim Montageeinlauf. Andererseits sind auch Interaktionsbeispiele für die Anwendungsfälle aus Kapitel 7 und 8 angelegt. Die darin enthaltenen Steuerungsentscheidungen spielen sich vor allem auf höheren Ebenen des Modells (Ebene 5 bis 8) ab. Für ihre Umsetzung ist die Ausführung von Befehlen bis auf die untersten Ebenen notwendig.

Damit sind nun auch die relevanten Steuerungsaufgaben und -informationen im Sinne der zweiten Forschungsfrage definiert und den Ebenen der generischen Fertigungssystemstruktur zugeordnet. Auf dieser Grundlage können zur Beantwortung der verbleibenden Forschungsfragen im nächsten Schritt der Arbeit Anwendungsfälle ausgewählt werden, um die Leistungsfähigkeit agentenbasierter Steuerungsarchitekturen aufzuzeigen und deren Integration zu realisieren.

6 Auswahl von Anwendungsfällen und Vorgehen

Die in den vorangegangenen Kapiteln erarbeiteten Ergebnisse legen die Grundlage für die Bearbeitung der Forschungsfragen 3 und 4. Diese zielen auf den Mehrwert von Agentensystemen im komplexen Produktionsumfeld und deren Integrierbarkeit ab (vgl. Abschnitt 1.2). Um die Beantwortung der Forschungsfragen möglichst anschaulich und praxisnah zu gestalten, soll ihre Erarbeitung unter Verwendung von zwei konkreten Anwendungsfällen erfolgen. Diese werden in den nachfolgenden Kapiteln 7 und 8 zunächst separat mithilfe der in Kapitel 4.2 ausgewählten DACS-Methode als Agentensysteme umgesetzt und danach bewertet. Im Anschluss wird in Kapitel 9 eine allgemeine AOSE-Methode für die Integration von interdependenten Agentensystemen abgeleitet, sodass die beiden Anwendungsfälle in Kapitel 10 schließlich mithilfe der neu entwickelten Methode integriert werden können.

Um passende Anwendungsfälle auszuwählen werden im folgenden Abschnitt 6.1 die Anforderungen an die Anwendungsfälle vorgestellt und eine grobe Potenzialbewertung der in Frage kommenden Hierarchieebenen vorgenommen. Anschließend werden in Abschnitt 6.2 aus dem resultierenden Optionsraum an typischen Steuerungssystemen in der Automobilproduktion zwei geeignete Anwendungsfälle ausgewählt und eine Vorabprüfung der konkreten Agentifizierungspotenziale durchgeführt. Schließlich erfolgt in Abschnitt 6.3 die Beschreibung des Vorgehens im Rahmen der durchzuführenden Simulationsstudien sowie die Auswahl geeigneter Implementierungstechnologien.

6.1 Anforderungen und Optionsraum

6.1.1 Herleitung von Anforderungen an Anwendungsfälle

Die Auswahl der Anwendungsfälle sollte zur Sicherstellung der Qualität, Übertragbarkeit und Relevanz dieser Arbeit eine Reihe von Anforderungen erfüllen. Vor diesem Hintergrund wurden sechs Anforderungen identifiziert, die im Folgenden vorgestellt werden.

A1: Eigenschaften moderner Produktionssteuerungssysteme. Zunächst sollten die Anwendungsfälle ein inhärentes Potenzial besitzen, sich von konventionellen Produktionssteuerungssystemen deutlich abzuheben, indem sie die in Abschnitt 2.3.2 geforderten Eigenschaften erfüllen. Dies kann sich z.B. in einer höheren Flexibilität der Ressourcen, einer schnelleren Rekonfigurierbarkeit bei neuen Marktbedingungen oder einer größeren Robustheit bei plötzlich auftretenden Störungen manifestieren.

A2: Stell- und Zielgrößen der Fertigungssteuerung. Für die Implementierung neuartiger Systeme in der Industrie bedarf es eines messbaren wirtschaftlichen Vorteils. Daher sollten die Anwendungsfälle einen signifikanten Einfluss auf die wesentlichen Stell- und Zielgrößen der Ferti-

gungssteuerung haben. D.h. sie sollten die Auftragsfreigabe, Kapazitätssteuerung und/oder Reihenfolgebildung direkt oder indirekt beeinflussen können, um die logistischen Zielgrößen Auslastung, Durchlaufzeit, Termintreue und Bestände zu verbessern.

A3: Messbarkeit und Vergleichbarkeit. Die Anwendungsfälle sollten über klar definierte Kennzahlen eine quantitative Messung der Leistung sowie einen gegenüberstellenden Vergleich mit konventionellen Steuerungssystemen zulassen. Diese Kennzahlen können, wie in Abschnitt 5.2 bereits verdeutlicht, sehr unterschiedlich ausfallen, sodass über die logistischen Zielgrößen hinaus auch anwendungsspezifische Messgrößen verwendet werden sollten, die im jeweiligen Umfeld sinnvoll sind.

A4: Hohes Umsetzungspotenzial. Die Anwendungsfälle müssen die drei Kriterien der Agentifizierungspotenzialanalyse aus Abschnitt 3.3.4 erfüllen, damit eine Umsetzung als Agentensystem sinnvoll ist. Darüber hinaus sollten die Anwendungsfälle so gewählt werden, dass eine reale Umsetzung einerseits ein hohes Nutzenpotenzial aufweist und andererseits verhältnismäßig schnell und aufwandsarm möglich ist.

A5: Interdependenz. Da im weiteren Verlauf der Arbeit die Integration der Anwendungsfälle vorgesehen ist, ist das Vorhandensein einer gewissen Abhängigkeit der Steuerungsentscheidungen in der Wertschöpfungskette erwünscht. Solche Interdependenzen sind für die meisten Steuerungssysteme nicht ungewöhnlich und entstehen z.B. durch zeitlich verwobene Materialströme, vordefinierte Füge- und Montagefolgen sowie Toleranzketten.

A6: Hierarchieebenen. Um ein möglichst breites Anwendungsfeld von Agentensystemen und der anschließend zu erarbeitenden AOSE-Methode zu gewährleisten, ist es sinnvoll Anwendungsfälle zu untersuchen, die auf verschiedenen Hierarchieebenen der GFSS ablaufen. Sie haben für gewöhnlich unterschiedliche Erfordernisse, deren Abbildung eine höhere Vielseitigkeit der zugrundeliegenden Agentensysteme und der Entwicklungsmethode erfordert.

6.1.2 Optionsraum

Vor dem Hintergrund der aufgeführten Anforderungen wurde aus der in Abschnitt 5.2 vorgestellten Übersicht eine Auswahl typischer Steuerungsaufgaben in der Automobilproduktion zusammengestellt (vgl. Abbildung 6-1). Die Aufgaben erstrecken sich über die vier Hauptgewerke Presswerk bis Montage und stellen den Optionsraum für die Bestimmung geeigneter Anwendungsfälle dar. Da die Steuerungsaufgaben die komplette Bandbreite von strategisch-taktischen Entscheidungen wie der Wochenprogrammplanung bis hin zu echtzeitrelevanten operativen Entscheidungen wie der Verarbeitung von Sensor- und Aktordaten reichen, wird im Folgenden skizziert, welche Hierarchieebenen sich besonders für die Agentifizierung in dieser Arbeit eignen.

Auf der **Feldebene**, die etwa der Ebene 1 bis 3 bzw. 4. in der GFSS entspricht, ist die Implementierung von Agenten grundsätzlich möglich. Das Aufwand-Nutzen-Verhältnis sollte jedoch vorab

genau geprüft werden, denn aufgrund der Echtzeitanforderungen ist die Umsetzung hier mit einem Zusatzaufwand verbunden, der nicht zwangsweise in jedem Anwendungsfall wirtschaftlich vertretbar ist (vgl. Abschnitt 3.3.2). Auch die Optimierungspotenziale sind auf Feldebene beschränkt. Da die Realisierbarkeit von Agenten auf Feldebene zudem bereits u.a. in den Werken [Wan10], [LVM09], [Göh13] und [VBH17b] nachgewiesen wurde, wird im Rahmen dieser Arbeit auf ihre Betrachtung verzichtet. Sie ist bei vielen Agentensystemen jedoch durchaus sinnvoll und ein logischer nächster Schritt zur Erweiterung höherer Agentensysteme.

Modellebene		Presswerk	Karosseriebau	Lackiererei	Montage
9	Fertigungs-Netzwerk				Wochen-programm-bildung
8	Fabrik	Batch-festlegung	Rohbau-aufgabe	Lackier-reihenfolge	Wochen-programm-bildung
7	Fertigungslinie	Material-bedarfs-ermittlung	Karosseriefolge	Virtuelle Resequenzierung	Tages-programm-bildung
6	Fertigungs-abschnitt	Linien-moni-toring	Linien-moni-toring	Linien-moni-toring	JIT-Abruf
5	Arbeitseinheit	Betriebsmittelsteuerung und -überwachung	Betriebsmittelsteuerung und -überwachung	Betriebsmittelsteuerung und -überwachung	Betriebsmittelsteuerung und -überwachung
4	Arbeitsstation		Kenn-Nr.-Vergabe		Materiallogistik Routenzugsystem
3	Funktionsgruppe				
2	Komponente ▪ Prozesskomp. ▪ Steuerungskomp.	Sensor/Aktor Daten I/O-Befehle	Sensor/Aktor Daten I/O-Befehle	Sensor/Aktor Daten I/O-Befehle	Sensor/Aktor Daten I/O-Befehle
1	Konstruktions-element				

Abbildung 6-1: Typische Steuerungsaufgaben in der Automobilproduktion i.A.a. [ZaL17]

Größere wirtschaftliche Potenziale sind auf höheren Ebenen von Produktionssystemen zu erwarten, da hier mit der Orchestrierung der untergeordneten Steuerungsaufgaben wesentliche systemweite Entscheidungen getroffen werden. Solche Entscheidungen finden auf der **MES-Ebene** statt, die sich etwa von Ebene 4 bis 7 bzw. 8 der GFSS erstreckt. Auf diesen Ebenen können z.B. Störungen im Ressourcenbetrieb, Turbulenzen im Materialfluss und andere Unregelmäßigkeiten erkannt und korrigiert werden. Die hier ablaufenden Prozesse haben ferner direkte, zeitkritische Auswirkungen auf den Shopfloor (z.B. Rekonfiguration der Auftragsfolge bei Störung), sodass eine Implementierung an dieser Stelle besonders sinnvoll erscheint.

Bei den Aufgaben der **ERP-Ebene**, die etwa auf den Ebenen 7 bis 9 stattfinden können, ist eine Differenzierung zwischen Steuerungsentscheidungen mit unmittelbaren und mittelbaren Auswirkungen auf den Shopfloor erforderlich. Während erstere hohes wirtschaftliches Potenzial für die Umsetzung in Agentensystemen enthalten (z.B. JIT-Systeme von Zulieferern), scheinen letztere aufgrund der zeitlich stärker vom Shopfloor getrennten Auswirkungen weniger relevant zu sein. Für die Betrachtungen in dieser Arbeit werden daher die mittleren Hierarchieebenen 5 bis 8 fokussiert. Da eine Verbindung der Unternehmensleitebene (z.B. langfristige Absatzplanung) mit

den Automatisierungssystemen dennoch vorteilhaft sein kann, ist auch deren Integration wünschenswert. Sie wird im Rahmen dieser Arbeit jedoch nicht behandelt. Ein adäquates Mittel für die Vernetzung könnten in Anlehnung an Kapitel 3.3.3 ggf. Serviceorientierte Architekturen wie z.B. Microservices darstellen.

6.2 Auswahl geeigneter Anwendungsfälle

Durch die oben getroffenen Begrenzungen des Betrachtungsraums entfallen einerseits Aufgaben niedriger Ebenen wie z.B. die Sensor-/Aktor-Steuerung und andererseits hochrangige Aufgaben wie die langfristige Absatzplanung. Zur Einhaltung der Anforderungen A1-A6 werden daher Anwendungsfälle der mittleren Ebenen in Betracht gezogen.

Im ersten Gewerk, dem Presswerk, wird größtenteils auftragsunabhängig gearbeitet und es besteht kein gekoppelter Materialfluss zu den folgenden Gewerken. Daher werden die Steuerungsaufgaben des Presswerks zunächst ausgeklammert. Im Karosseriebau hingegen erfolgt die physische Auftragszuweisung sowie die Verbindung der Prozesse über eine Fördertechnik, sodass hier eine Vielzahl potenziell interessanter Steuerungsaufgaben existiert. Die Rohbauauflage sowie die Kennnummernvergabe bestimmen die Auftragsreihenfolge und haben daher eine hohe Bedeutung. Dadurch, dass sie pro Auftrag jedoch nur einmalig ausgeführt werden, sind ihre Handlungsmöglichkeiten stark eingeschränkt. Die Betriebsmittelsteuerung und -überwachung hat dagegen kontinuierlichen Einfluss auf die laufende Produktion und damit grundsätzlich höheres Potenzial. Diese Steuerungsentscheidungen finden aber eher auf niedrigen Hierarchieebenen statt, was sie für diese Arbeit wiederum weniger interessant macht. Die **Steuerung der Karosseriefolge** hingegen erfüllt die wesentlichen Kriterien aus dem vorangehenden Abschnitt und soll daher als **erster Anwendungsfall** herangezogen werden. Die hochrangigen Steuerungsaufgaben in der Lackiererei sind denen des Karosseriebaus sehr ähnlich, sodass als zweiter Anwendungsfall ein System aus der Montage gewählt werden soll. Der Einfluss der werksübergreifenden Wochenprogrammplanung ist durch die Ressourcen der Werke meistens physisch auf die Fertigung bestimmter Modelle beschränkt. Eine Verlagerung von Aufträgen zwischen Werken ist mit heutigen Ressourcen daher kaum realisierbar. JIT-Abrufe sind wichtig um die Kopplung mit internen und externen Zulieferern zu realisieren. Ihre Freiheitsgrade sind derzeit jedoch ebenfalls eingeschränkt, da sie Unternehmensgrenzen überschreiten, was die Implementierung verkompliziert. Von größerem Interesse ist dagegen die Materiallogistik, die über unterschiedliche Transportsysteme realisiert werden kann. Bei der werksinternen Materialbelieferung gehören Routenzugsysteme zu den am weitesten verbreiteten Mitteln. Da auch diese Steuerungsaufgabe die gestellten Anforderungen erfüllt, wird die **werksinterne Materialbelieferung mit Routenzügen** als **zweiter Anwendungsfall** gewählt.

Im Folgenden werden zunächst die individuellen (A1-A4) und anschließend die gemeinsamen Ausprägungen (A5-A6) der Anwendungsfälle hinsichtlich der Anforderungen skizziert. Eine detaillierte Betrachtung der Anwendungsfälle erfolgt in den Kapiteln 7 und 8.

Anwendungsfall 1: Karosseriefolgesteuerung

Aufgabe der Karosseriefolgesteuerung (KFS) ist es Aufträge so zu steuern, dass eine hohe Ressourcenausnutzung, kurze Durchlaufzeiten und geringe Lieferterminabweichungen erreicht werden. Dafür können Karossen vorgezogen oder zurückgehalten sowie Pufferplätzen bzw. Ressourcen zugewiesen werden. Durch eine intelligente KFS kann so u.a. die Flexibilität und Robustheit des Systems verbessert werden (A1). Da durch die Reihenfolgesteuerung der Produktvarianten auch der Materialbestand an den Arbeitsstationen beeinflusst wird, hat sie außerdem Auswirkungen auf die wesentlichen Stell- und Zielgrößen des Systems (A2). Das Gebiet der Reihenfolgesteuerung ist zudem von entscheidender Bedeutung für die Produktion, sodass auch zahlreiche Kennzahlen existieren, mit denen die Systemleistung gemessen werden kann (A3). Dazu gehört u.a. die Perlenkettengüte, auf die in Kapitel 7 im Detail eingegangen wird. Aus demselben Grund hat die Agentifizierung der KFS ein hohes Nutzenpotenzial und sie kann vergleichsweise aufwandsarm implementiert werden, da sie keine physischen Veränderungen benötigt. Der Anwendungsfall erfüllt schließlich auch die Kriterien der Agentifizierungspotenzialanalyse, denn er enthält mehrere Entscheidungen, die voneinander abhängen und die zeitlich verteilt werden können (A4).

Anwendungsfall 2: Routenzugsteuerung für die werksinterne Materialbelieferung

Aufgabe der Routenzugsteuerung (RZS) ist es die an den Montagebändern entstehenden Materialbedarfe zu decken. Damit beeinflusst die RZS sowohl Materialbestände als auch Transport- und Fertigungsressourcen. Die Steuerung dieser Vielzahl von Komponenten birgt auch hier die Chance, Flexibilität, Agilität, Robustheit und Rekonfigurierbarkeit des Systems wesentlich zu verbessern (A1). Die RZS wirkt direkt auf die Kapazitätssteuerung der Routenzüge und der Ressourcen, denn es kann nur mit ausreichend Material gefertigt werden (A2). Auch in diesem gut erforschten Bereich existiert eine Reihe von Kennzahlen, die für die Leistungsmessung des Steuerungssystems verwendet werden kann (A3). Dazu gehören u.a. Fahrzeit, Terminabweichung und Auftragsanzahl pro Fahrt, auf die in Kapitel 8 näher eingegangen wird. Schließlich besteht auch in diesem Fall ein hohes wirtschaftliches Potenzial, das z.B. durch Erhöhung der Routenzugauslastung oder der Liefertreue realisiert werden kann. Gleichzeitig bedarf es keiner physischen Anpassung des Systems und es werden auch hier die Kriterien der Agentifizierungspotenzialanalyse erfüllt, da mehrere, zeitlich verteilbare Entscheidungen getroffen werden müssen (A4).

Einordnung der Anwendungsfälle

Die beiden Anwendungsfälle sind so gewählt, dass sie auch die letzten beiden Anforderungen erfüllen. In Abbildung 6-2 ist erkennbar, dass die Anwendungsfälle aufeinanderfolgende Schritte der Wertschöpfungskette darstellen. Da sie über den Materialfluss miteinander gekoppelt sind, werden sie durch eine gegenseitige Interdependenz charakterisiert (A5). Einerseits beeinflussen die Ausgangsgrößen der KFS die Eingangsgrößen der RZS, denn je nach Auftrag und Produktvariante werden unterschiedliche Materialien in der Montage verschieden schnell verbraucht und

lösen Bedarfe aus. Andererseits wirken sich die Kapazitätsgrenzen und der Systemzustand der RZS auch auf die KFS aus, denn es sollten nur Aufträge übergeben werden, für die ausreichend Material bereitgestellt werden kann. Schließlich erfolgt die Entscheidungsfindung der beiden Anwendungsfälle auf unterschiedlichen Ebenen (A6). Zwar erstrecken sich die beiden Anwendungsfälle theoretisch bis auf Ebene des Shopfloors, die relevanten Steuerungsentscheidungen finden jedoch weiter oben statt. Bei der KFS werden sie auf den Ebenen 7-8 getroffen und bei der RZS auf den Ebenen 6-7. Die Ausführung erfolgt jeweils auf den darunterliegenden Ebenen.

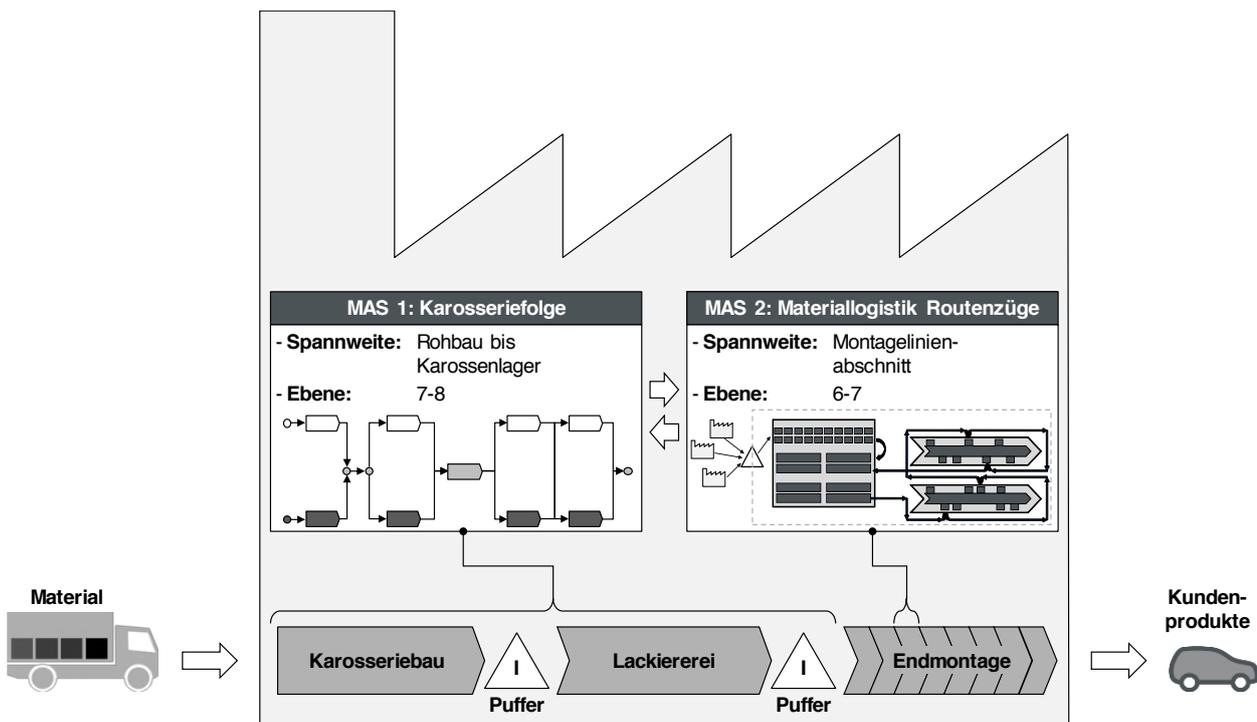


Abbildung 6-2: Einordnung der Anwendungsfälle in den Gesamtprozess i.A.a. [ZaL17]

Nach der nun erfolgten Auswahl und kurzen Vorstellung der Anwendungsfälle wird im nächsten Abschnitt das weitere Vorgehen für die Erreichung der Ziele dieser Arbeit konkretisiert.

6.3 Weiteres Vorgehen

In den Kapiteln 7 und 8 werden die beiden Anwendungsfälle zunächst separat bearbeitet. Dafür erfolgt zunächst eine detaillierte Aufnahme des Status quo der Anwendungsfälle sowie eine Beschreibung der derzeit für sie verwendeten Scheduling-Methoden und existierender Implementierungsbeispiele von Agentensystemen in angrenzenden Bereichen. Als nächstes sollen die Modelle mithilfe der DACS-Methode als Agentensysteme konzipiert werden und mit den geeigneten Umsetzungstechnologien programmiert werden. Da für die Programmierung zahlreiche Agentenplattformen in Frage kommen, werden in Abschnitt 6.3.1 für diese Arbeit sinnvolle Ansätze ausgewählt. Um die Systeme zu testen und deren reale Implementierung in eine Fabrik vorzubereiten, sollen die beiden Agentensysteme anschließend in Simulationsumgebungen erprobt und bewertet werden. Dies ist bei Systemen der vorliegenden Komplexität üblich, da eine analytische Bewertung nicht realisierbar ist [Law15]. Um die Leistung der Agentensysteme zu messen und

mit konventionellen Steuerungssystemen vergleichen zu können, bedarf es eines systematischen Vorgehens für den Ablauf von Simulationsstudien und das Design von Simulationsexperimenten. Die dafür ausgewählten Methoden werden in Abschnitt 6.3.2 vorgestellt.

6.3.1 Auswahl geeigneter Agentenplattformen

Zur Implementierung von Agentensystemen sind grundsätzlich keine Agentenplattformen notwendig [VDI12], denn die entsprechenden Systeme können auch in selbstständig entwickelten Umgebungen ausgeführt werden. Um die Entwicklung zu beschleunigen sowie die Standardisierung von Agentensystemen voranzutreiben, ist jedoch die Verwendung einheitlicher Implementierungstechnologien – im Weiteren auch Frameworks oder Agentenplattformen genannt – unerlässlich. Die Vielzahl existierender Lösungen stellt Entwickler dabei vor die Herausforderung eine passende Plattform auszuwählen. Alleine in [Wik17] erfolgt eine Gegenüberstellung von etwa 90 Agenten-Frameworks. Vor dem Hintergrund der Vielzahl an Möglichkeiten schlägt die VDI-Norm 2653-2 die in Tabelle 6-1 aufgeführten Kriterien zur Bewertung von Agentenplattformen vor [VDI12].

Kriterien	Ausprägungen
1. Allgemein	Hersteller/Service-Provider
2. Erprobtheit	Erste Version, Referenzkunden/-projekte, externe Publikationen, Anzahl akademischer/industrieller Nutzer
3. Pflege & Weiterentwicklung	Letztes Release, Versionen/Weiterentwicklung, Verfügbarkeit, Lizenzbedingungen, Verfügbarkeit des Quellcodes, Kompatibilität
4. Einsatzbereiche	Unterstützte Anwendungsumgebungen, typische Anwendungsfelder, unterstützte Automatisierungsebenen
5. Entwicklerunterstützung	Analysewerkzeuge, Entwicklungswerkzeuge, Entwicklerführung, Implementierungssprache, Komponentenbibliothek, Visualisierung/Debugging, Dokumentation, Benutzerforen
6. Interoperabilität	Offene Schnittstellen, Betriebssysteme, Kommunikation, Andere Agentenplattformen, Vorbereitete Kommunikationsschnittstellen, Integrierbarkeit in bestehende Automatisierungsanwendungen
7. Standards	Unterstützung von Architekturstandards, Ontologieunterstützung, Kommunikationsstandards der Automatisierungstechnik
8. Sonstiges	Unterstützung von mobilen Geräten, Ressourcenbedarf, Echtzeitfähigkeit, Sicherheitsmodell, Autorisierungen und Berechtigungen

Tabelle 6-1: Auswahlkriterien für Agentenplattformen i.A.a. [VDI12]

Die VDI-Kriterien werden weiter konkretisiert, indem ihnen die Anforderungen unterschiedlicher Anwendungsbereiche wie z.B. Produktionsleittechnik, Prozessautomatisierung und Logistik gegenübergestellt werden. Eine Eingrenzung der Menge an Frameworks findet jedoch nicht statt. Hier knüpft die Studie von Kravari und Bassiliades an, in der 24 Agentenplattformen für die Simulation von realen Anwendungsfällen untersucht und bewertet werden [KrB15]. Die Studie klassifiziert Frameworks anhand zahlreicher Kriterien wie Domäne, Programmiersprache, Erlernbarkeit sowie dem aktuellen Entwicklungsstatus und ermöglicht Entwicklern schrittweise eine geeignete

Plattform zu wählen [KrB15]. Als Agentenplattform für die Simulation der hier ausgewählten Anwendungsfälle eignen sich grundsätzlich sowohl in allgemeinen Programmiersprachen verfasste Frameworks bzw. Bibliotheken als auch spezifisch auf Simulation ausgelegte Softwarepakete. **Allgemeine Frameworks** benutzen meist den Anwendern bereits vertraute Programmiersprachen wie Java oder C++. Die so verfassten Modelle bieten eine hohe Flexibilität und können schneller ausgeführt werden als Simulationspakete. Allgemeine Frameworks unterstützen zudem häufig die OOP, wodurch die Analyse der Simulation erleichtert wird. **Simulationspakete** stellen auf der anderen Seite häufig hilfreiche Funktionen zur zeit- und kosteneffizienten Erstellung von Simulationsmodellen zur Verfügung. Die Simulationsumgebungen bestehen aus standardisierten Bausteinen, sodass Modelle einfacher zu modifizieren und zu pflegen sind. Es existieren allerdings nur wenige kommerziell verfügbare Simulationspakete, die sich für die Simulation der ausgewählten Anwendungsfälle eignen [Jun15]. Aufgrund der Vorteile beider Plattfortmtypen soll für die Verwendung in dieser Arbeit von jedem Typ jeweils eine Plattform ausgewählt werden.

Auf Seiten der allgemeinen Frameworks ist **JADE**³⁰ die am weitesten verbreitete Plattform. Sie basiert auf einer Open-Source-Lizenz, ist vollständig in Java umgesetzt und unterstützt verschiedene Betriebssysteme, da sie auf der Java Virtual Machine (JVM) ausgeführt wird [KrB15]. Folglich existieren zahlreiche Implementierungen in Wissenschaft und Industrie. Dank der großen Zahl an Benutzern sind Bibliotheken, Debugging-Werkzeuge und Benutzerforen vorhanden. Zudem erfüllt JADE weitere Anforderungen wie Kompatibilität mit den FIPA-Standards³¹ und ACL³² und kann darüber hinaus mit der Erweiterung JadeX auch BDI-Ansätze realisieren [PoB18]. JADE wird daher als Agentenplattform für die Umsetzung des ersten Anwendungsfalls verwendet.

Auf Seiten der Simulationspakete ist zu berücksichtigen, dass sich die Software für die Simulation eines dynamischen Systems eignen muss, dem ein deterministisches Ereignismodell zugrunde liegt. Zudem sollen interagierende Agenten abgebildet werden, sodass keine rein diskrete Simulation verwendet werden kann. Eines der wenigen kommerziell verfügbaren Simulationspakete, die sich hierfür eignen ist die Simulationssoftware **AnyLogic**. Sie erlaubt die Ausführung systemdynamischer, diskreter ereignis- und agentenbasierter Modellierungen sowie von Mischformen davon [Bor13]. AnyLogic ist anwenderfreundlich, hochgradig anpassbar und eignet sich für den Einsatz in fast jeder Domäne [KrB15]. Folglich wird AnyLogic für die Umsetzung des zweiten Anwendungsfalls ausgewählt.

Obwohl sich die vorgestellten Plattformen für die in dieser Arbeit verfolgte Zielstellung gut eignen, könnten für die Entwicklung und das Deployment von großen, skalierbaren Applikationen andere

³⁰ Java Agent Development Framework

³¹ Foundation for Intelligent Physical Agents

³² Agent Communications Language

Technologien wie z.B. Erlang OTP (www.erlang.org) eine bessere Wahl darstellen. Für diese Anwendungsbereiche wird auf [VaB15] verwiesen.

Als letzter vorbereitender Schritt vor der Realisierung der Agentensysteme werden im nächsten Abschnitt Methoden vorgestellt, die für den Ablauf von Simulationsstudien und das Design von Simulationsexperimenten berücksichtigt werden müssen.

6.3.2 Evaluation der Anwendungsfälle in Simulationsstudien

Grundlagen der Simulation

Bei der Betrachtung von Simulationsstudien ist zwischen den Begriffen Simulation und Simulationsexperiment zu unterscheiden. Eine **Simulation** ist nach der VDI-Norm 3633 definiert als:

„[...] Verfahren zur Nachbildung eines Systems mit seinen dynamischen Prozessen in einem experimentierbaren Modell, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind.“ [VDI13]

Ein **Simulationsexperiment** wird in Analogie nach der VDI-Norm 3633 definiert als:

„[...] gezielte empirische Untersuchung des Modellverhaltens über einen bestimmten Zeithorizont durch wiederholte Simulationsläufe mit systematischen Parametervariationen.“ [VDI13]

Simulationsmodelle können anhand von drei Dimensionen klassifiziert werden [Law15]:

- **Statisch/dynamisch.** Statische Simulationsmodelle bilden Systeme zu einer bestimmten Zeit ab, wohingegen dynamische Modelle die zeitliche Entwicklung der Systeme erfassen.
- **Deterministisch/stochastisch.** In deterministischen Modellen sind die zukünftigen Zustände des Systems eindeutig festgelegt, während sie in stochastischen Modellen einer Wahrscheinlichkeitsverteilung unterliegen, die nicht exakt vorhergesehen werden kann.
- **Kontinuierlich/diskret.** In kontinuierlichen Systemen variieren die Zustandsvariablen permanent mit dem Zeitverlauf. In diskreten Ereignissimulationen, die sich in der Simulationsforschung etabliert haben [Bor13], verändert sich der Systemzustand durch Ereignisse hingegen sprunghaft.

Da weder kontinuierliche noch diskrete Simulationsmodelle ein Gesamtsystem aus interagierenden Subsystemen abbilden können, wurden ergänzend agentenbasierte Simulationen entwickelt [Law15]. Sie stellen eine spezifische Art der ereignisorientierten Simulation dar, die „bottom up“ anstatt wie üblicherweise „top down“ erfolgt [Jun15]. Darin werden aktive Agenten und passive Objekte als Entitäten unterschieden und abstrakte Zustandsvariablen wie z.B. Populationsgrößen individuell simuliert [WGP08].

Stärken und Schwächen von Simulationen

Simulationen erlauben im Vergleich zu realen Systemen eine flexiblere Steuerung der Betriebsbedingungen. In Simulationen kann die Zeit komprimiert oder ausgedehnt werden, um Beobachtungen über sehr kurze oder sehr lange Zeiträume zu realisieren [Law15]. So kann eine Vielzahl von Durchläufen untersucht und damit die statistische Genauigkeit der Simulation und die Validität des zugrundeliegenden Modells erhöht werden [DaC10]. Simulationsstudien werden daher oft verwendet, um die Leistung eines existierenden oder geplanten Systems unter spezifischen Betriebsbedingungen einzuschätzen oder alternative Systemdesigns hinsichtlich der Erfüllung gestellter Anforderungen zu vergleichen [Law15]. Sie eignen sich zudem für die Darstellung komplexer und dynamischer Prozesse sowie zur Identifizierung und Veranschaulichung komplizierter Wirkzusammenhänge [DaC10]. Trotz dieser Vorteile sind analytische Methoden Simulationen vorzuziehen, falls sie verfügbar oder unter verhältnismäßigem Aufwand zu entwickeln sind. Dies wird durch das nichtlineare Verhalten der hier betrachteten logistischen Prozesse jedoch erschwert [Jun15]. Zudem ist zu berücksichtigen, dass Simulationen mit hohem Zeitaufwand und Kosten verbunden sind und die Gefahr besteht, Vertrauen in nicht validierte Modelle zu setzen [Law15]. Für die Zielsetzung der vorliegenden Arbeit stellen Simulationen jedoch ein adäquates Mittel dar. Ihre Durchführung erfolgt nach dem im Folgenden skizzierten methodischen Vorgehen.

Ablauf von Simulationsstudien

Das Vorgehen bei der Durchführung von Simulationsstudien wird in der Literatur sehr unterschiedlich beschrieben. Es existieren sowohl universelle Ansätze wie das Grundmodell von Kulla [Kul87] als auch spezifische Ansätze wie der Ansatz von Gilbert und Troitzsch [GiT11], der Experimente als Sensitivitätsanalyse gestaltet. Das Modell von Kühn gibt eine detaillierte Anleitung zum praktischen Projektablauf. Es eignet sich jedoch nur bedingt für diese Arbeit, da es einerseits starken Fokus auf die praktische Umsetzung der Simulation legt und andererseits wesentliche Schritte wie z.B. eine a priori Validierung von Daten und Annahmen fehlen [Küh06]. Geeigneter ist das Modell nach Law, das den prinzipiellen Ablauf von Simulationsstudien in einer angemessenen Detailtiefe erfasst und implizit die Schritte des Grundmodells von Kulla enthält [Law15]. Zudem ist es universell anwendbar, hat eine solide theoretische Grundlage und enthält wertvolle Hinweise für die praktische Durchführung von Simulationsstudien. Daher dient das Modell nach Law als Vorlage für die in dieser Arbeit durchgeführten Simulationen. Die Schritte des Modells sind in Abbildung 6-3 zusammengefasst.

Im ersten Schritt wird zunächst die Problemstellung konkretisiert und klar formuliert. Als nächstes werden alle notwendigen Daten, Informationen und Algorithmen erfasst sowie Annahmen abgeleitet. Im dritten Schritt werden die verfügbaren Daten und Annahmen auf ihre Gültigkeit geprüft und anschließend – bei erfolgreicher Prüfung – in ein Modell überführt und programmiert. Sollten sich Teile als ungültig erweisen, fließen diese Erkenntnisse in die vorangehenden Schritte ein. Im fünften Schritt wird das nun programmierte Modell ebenfalls validiert und im darauffolgenden

sechsten Schritt werden Simulationsexperimente entworfen, durchgeführt und dokumentiert. Als letztes erfolgt die Dokumentation der Simulationsergebnisse [Law08]. Zur Unterstützung der beiden finalen Schritte wird auf die im nächsten Abschnitt vorgestellte Methodik zurückgegriffen.

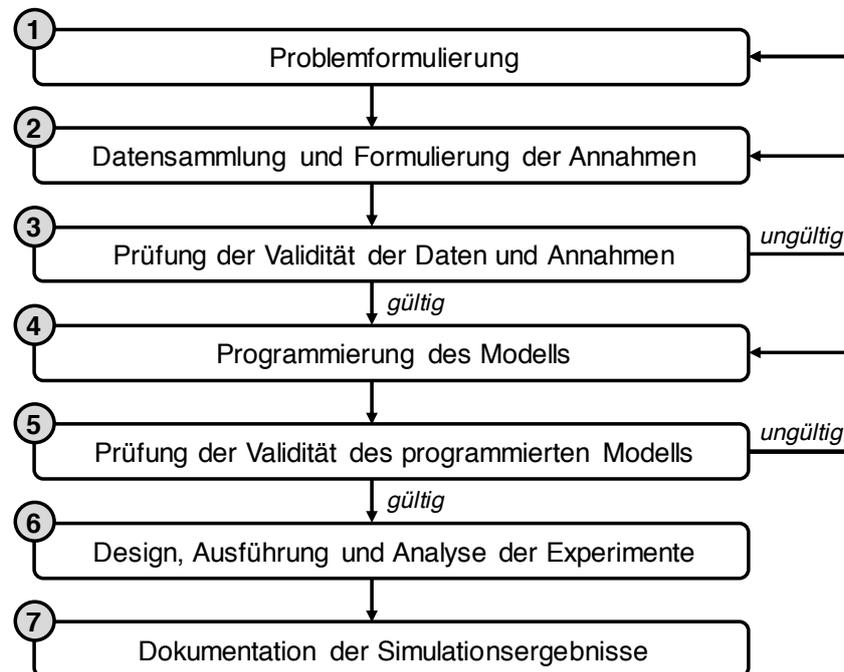


Abbildung 6-3: Ablauf von Simulationsstudien i.A.a. [Law08]

Design und Analyse von Simulationsexperimenten

Bei Modellen mit vielen Einflussparametern stellt die Entwicklung aussagekräftiger Experimente und ihre verständliche Dokumentation eine besondere Herausforderung dar. Um die Wechselwirkungen zwischen untersuchten Parametern systematisch zu identifizieren, wird eine statistische Versuchsplanung empfohlen [SBH10]. Die Methode von Lorscheid et al. erleichtert es, Abhängigkeiten auch bei einer Vielzahl von Faktoren und Faktorstufen systematisch zu identifizieren [LHM12]. Ihre Schritte sind in Abbildung 6-4 dargestellt.

In der **vorexperimentellen Phase** wird im ersten Schritt die Zielsetzung des Simulationsexperimentes formuliert. Typische Beispiele sind die Bestimmung der optimalen Einstellungen oder einflussreicher Variablen eines Systems sowie die Identifikation von Steuerfaktoren zur Begrenzung der Auswirkungen von Störgrößen [WHW00]. Als nächstes werden die Modellvariablen in die drei Kategorien abhängige, unabhängige und steuernde Variablen eingeteilt. Abhängige Variablen werden durch die Werte von unabhängigen Variablen beeinflusst. Steuernde Variablen sind hingegen entweder fix eingestellte Werte oder durch Zufallsvariablen bestimmt [LHM12]. Im dritten Schritt der Methode beginnt die **experimentelle Phase**, in der die zuvor ermittelten Modellvariablen in Faktoren und Zielgrößen überführt werden. Faktoren sind unabhängige bzw. steuernde Variablen, die anhand der vermuteten Bedeutung für die Problemstellung als Einflussgrößen des Modells ausgewählt werden [SBH10, Kle16]. Im Gegensatz dazu sind unter abhängigen Variablen die Zielgrößen des Systems zu verstehen [LHM12]. Für alle Faktoren werden Faktorstufen festgelegt, sodass jeder Faktor auf mindestens zwei Stufen getestet werden kann. Faktoren, die

nicht untersucht werden, sollten möglichst konstant gehalten werden [Kle16]. Im vierten Schritt werden im Rahmen des Designs der Versuchsreihen die zu testenden Faktorstufenkombinationen festgelegt. Bei einer vollfaktoriellen Versuchsreihe werden alle möglichen Kombinationen untersucht. In Schritt fünf erfolgt die Abschätzung der experimentellen Fehlervarianz, um die Einflüsse probabilistischer Elemente zu würdigen. Als nächstes erfolgt die Durchführung des eigentlichen Simulationsexperimentes, die als Ergebnis die gesuchten Zielgrößen für unterschiedliche Faktorstufenkombinationen hervorbringt. Im letzten Schritt werden die Ergebnisse analysiert, um ein möglichst genaues Bild aller Effekte und Wechselwirkungen für die ausgewählten Faktoren im System zu erhalten [LHM12].

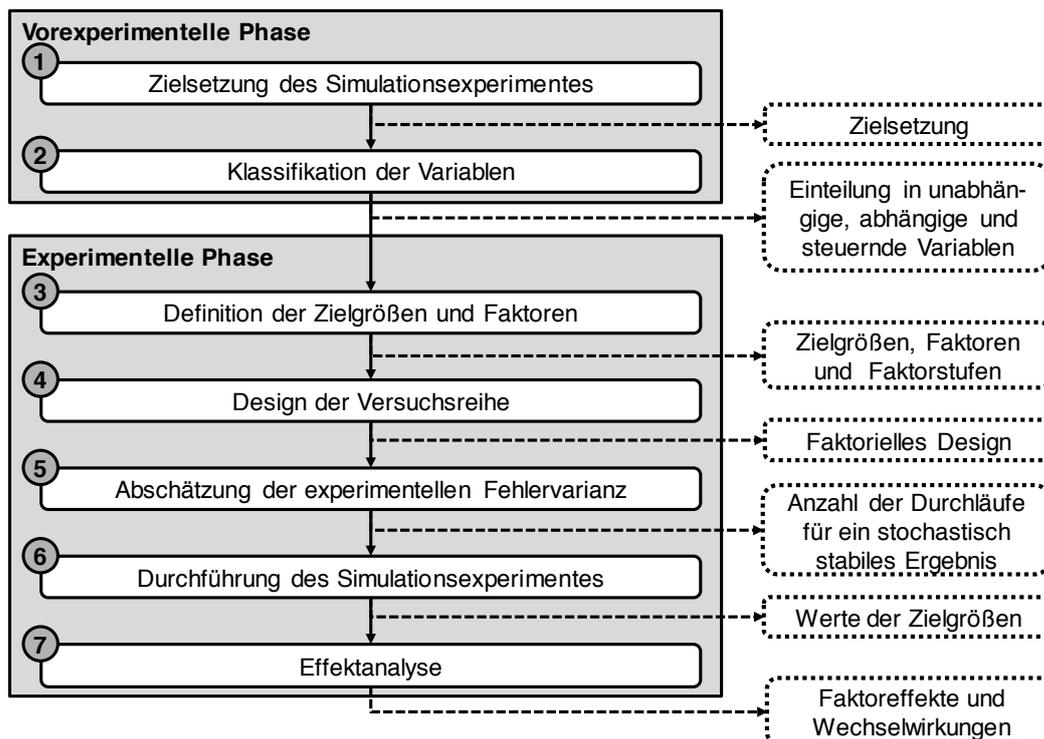


Abbildung 6-4: Design und Analyse von Simulationsexperimenten i.A.a. [LHM12]

Zusammenfassend sind mit dem Ablauf von Simulationsstudien nach Law und der statistischen Versuchsplanung nach Lorscheid et al. die notwendigen Methoden gegeben, um eine qualifizierte Simulation der ausgewählten Anwendungsfälle zu realisieren. Damit sind nun die Vorarbeiten abgeschlossen, sodass in den nächsten Kapiteln die Umsetzung und die Integration der definierten Anwendungsfälle als Agentensysteme vorgenommen wird. Die in den nächsten beiden Kapiteln vorgestellten Anwendungsfälle wurden im Rahmen von zwei separaten Projekten im Rahmen dieser Arbeit durchgeführt. Weitere Details sind in den Arbeiten [LZB17], [ZaL17] und [ZHL17] sowie in den Abschlussarbeiten von [Bec16] und [Hab16] zu finden. Im Folgenden erfolgt eine detailliertere und ganzheitliche Betrachtung der Ergebnisse dieser Arbeiten.

7 Anwendungsfall 1: Karosseriefolgesteuerung

Ziel dieses Kapitels ist die Konzeptionierung, Implementierung, Simulation und Bewertung eines Agentensystems für den ausgewählten Anwendungsfall der Karosseriefolgesteuerung. Als Struktur wird dafür das zuvor beschriebene Vorgehen nach Law in Kombination mit der DACS-Methode genutzt, sodass sich dieses Kapitel wie folgt gliedert. In Abschnitt 7.1 erfolgen die ersten drei Schritte nach Law, d.h. es wird die grundsätzliche Problemstellung aufgezeigt, Daten werden aufgenommen und Annahmen formuliert und validiert. Im daran anknüpfenden Abschnitt 7.2 erfolgt die Konzeption des Agentensystems nach den drei Schritten der DACS-Methode. Im Anschluss daran wird in Abschnitt 7.3 gemäß dem vierten und fünften Schritt nach Law auf die Konzepte und Algorithmen eingegangen, die für die Programmierung und Validierung des Systems relevant sind. In Abschnitt 7.4 werden schließlich die letzten beiden Schritte nach Law ausgeführt, indem zunächst das Design der Simulationsexperimente sowie anschließend die Analyse und Bewertung der Simulationsergebnisse stattfindet.

7.1 Ausgangssituation und Zielsetzung

Als erster Anwendungsfall für diese Arbeit wurde ein moderner Karosseriebau ausgewählt. Eine Vorstellung der grundsätzlichen Problemstellung sowie der wesentlichen Rahmendaten erfolgt in Abschnitt 7.1.1. In den nächsten zwei Abschnitten werden notwendige Daten gesammelt sowie Annahmen formuliert. Dafür wird zunächst in Abschnitt 7.1.2 das zugrundeliegende Steuerungsprinzip der Perlenkette sowie die Ursachen für eine instabile Auftragsfolge diskutiert. Darauf aufbauend werden in Abschnitt 7.1.3 gängige Scheduling-Verfahren zur Reihenfolgebildung untersucht. Zur Validierung der prinzipiellen Eignung des agentenbasierten Lösungsansatzes für das ausgewählte Problem wird in Abschnitt 7.1.4 analysiert, in welchen angrenzenden Bereichen Agentensysteme bereits erfolgreich eingesetzt wurden. Schließlich wird in Abschnitt 7.1.5 die mit dem Anwendungsfall zu erreichende Zielsetzung festgelegt.

7.1.1 Problemstellung

Eine hohe Komplexität von Produkten und Produktionssystemen erhöht die Wahrscheinlichkeit von Störungen während des Herstellungsprozesses. Auch wenn Konzepte wie Total Productive Maintenance (TPM) und Predictive Maintenance in den vergangenen Jahrzehnten dazu beigetragen haben Ausfälle zu reduzieren, müssen Fertigungsunternehmen ständig mit Störungen umgehen. Da Störungen jederzeit in jedem der tausenden von Elementen eines Produktionsnetzwerkes auftreten können, gehören **Anpassungen des Produktionsablaufs** zur Tagesordnung. Durch Umsetzung des Industrie 4.0-Gedanken könnten insbesondere Unternehmen profitieren, die gleichzeitig eine hohe Produkt- und Produktionskomplexität, eine große Produktvielfalt sowie ein hohes Volumen bewältigen müssen. Ein bekannter Vertreter solcher Produktionsbedingungen ist die Automobilindustrie, bei der hochindividuelle, komplexe Produkte auf gemischten Produkti-

onslinien in Serie gefertigt werden. Die dort angewendeten Produktionssteuerungsverfahren werden auch in anderen Branchen wie der Fertigung von Unterhaltungselektronik, Haushaltsgeräten, Möbeln und Bekleidung eingesetzt [SaP01]. Dadurch ist die Übertragbarkeit der hier vorgestellten Konzepte auf diese Bereiche gegeben. Um das Potenzial von Produktionssystemen in solchen dynamischen Umgebungen optimal auszuschöpfen ist eine detaillierte Ablaufplanung erforderlich [BFS09]. Die Ursachen vieler Steuerungsprobleme in diesem Zusammenhang sind mit der **Auftragsfreigabe** sowie der **Ressourcenzuweisung** verbunden. Deshalb wurde als Anwendungsfall ein moderner Karosseriebau gewählt, dessen vereinfachtes Hallenlayout in Abbildung 7-1 dargestellt ist.

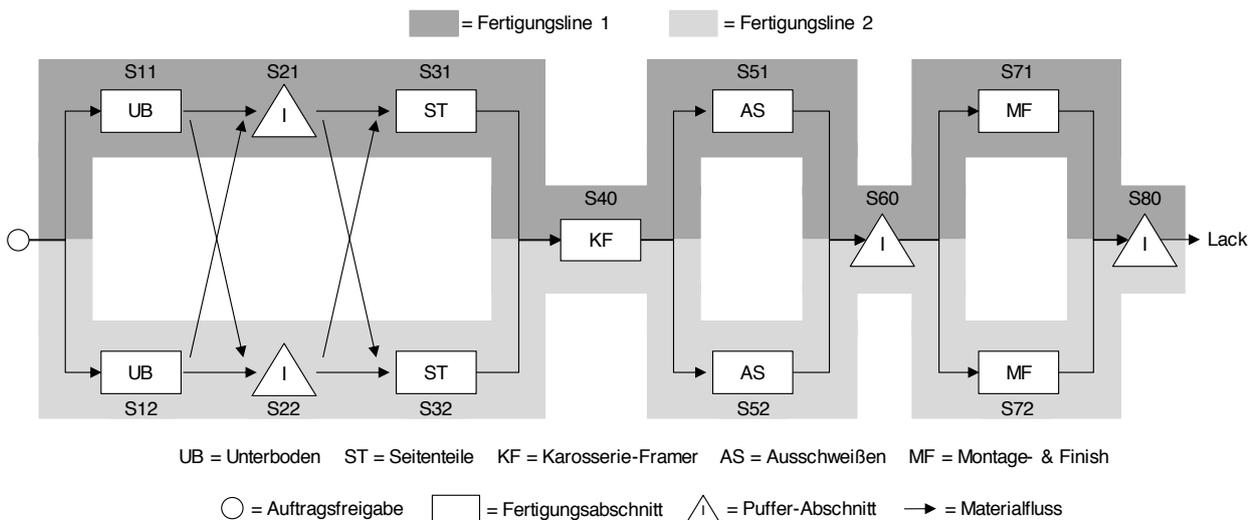


Abbildung 7-1: Anwendungsfall 1 – Hallenlayout Karosseriebau

Der Karosseriebau umfasst die Ebenen 1 bis 7 der GFSS. Aufgrund des MES-Fokus dieser Arbeit werden im Folgenden die Ebenen 5 bis 7 berücksichtigt, d.h. Arbeitseinheiten, Fertigungsabschnitte und Fertigungslinien. Auf oberster Ebene existieren zwei Fertigungslinien, die unterschiedliche Karosserietypen herstellen. Diese beiden Linien bestehen aus 13 verschiedenen Fertigungsabschnitten S11 bis S80. Einige dieser Abschnitte, wie S11 und S12, werden ausschließlich von der Fertigungslinie 1 oder 2 verwendet, andere wie S40 und S60 werden von beiden gemeinsam genutzt und bilden daher einen Engpass.

Auf Ebene 5 besteht jedes Segment aus mehreren separaten Arbeitseinheiten, die nicht im Layout abgebildet sind. Sie führen jeweils eine Reihe von Produktionsschritten durch, die im Fertigungsabschnitt erforderlich sind. Insgesamt kann die Karosserieschweißerei rund 70 verschiedene Produktvarianten herstellen, darunter verschiedene Karosserietypen und marktspezifische Modelle. Alle Varianten basieren jedoch auf drei Haupttypen: einem zweitürigen und einem viertürigen Kompaktwagen sowie einem viertürigen Kompaktvan. Zur Veranschaulichung der grundsätzlichen Effekte von Agentensystemen konzentriert sich diese Arbeit im Weiteren auf diese drei Haupttypen, die als Varianten A, B und C bezeichnet werden. Während die Fertigungslinie 1 die

Varianten A und B herstellen kann, werden auf Fertigungslinie 2 nur die Varianten A und C hergestellt. Die gemeinsam genutzten Ressourcen können alle Varianten gemäß Tabelle 7-1 aufnehmen.

Variante	Fertigungslinie 1				Fertigungslinie 2				Gemeinsame Ressourcen				
	S11	S31	S51	S71	S12	S32	S52	S72	S21	S22	S40	S60	S80
A	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
B	✓	✓	✓	✓					✓	✓	✓	✓	✓
C					✓	✓	✓	✓	✓	✓	✓	✓	✓

Tabelle 7-1: Variantenflexibilität der Ressourcen

Im Rahmen dieser Arbeit wurde der ausgewählte Karosseriebau bei einem großen Automobilhersteller untersucht und Probleme identifiziert, die für ein solches Steuerungsumfeld typisch sind. Dazu zählt einerseits die **ungleichmäßige Verteilung der Durchlaufzeiten** wie in Abbildung 7-2 dargestellt. Hier ist die zu erwartende Glockenkurve stark nach rechts gestreckt, da ein Teil der Aufträge sehr lange Durchlaufzeiten hat. Ein Phänomen, das in der Automobilindustrie als „Schleppe“ bezeichnet wird [Her12]. Nach den Erfahrungen des Autors ist dieser Effekt in den meisten Automobilfabriken vorzufinden, was den Schluss nahelegt, dass effektive Produktionssteuerung einen wesentlichen Hebel zur Verbesserung der Situation darstellen kann.

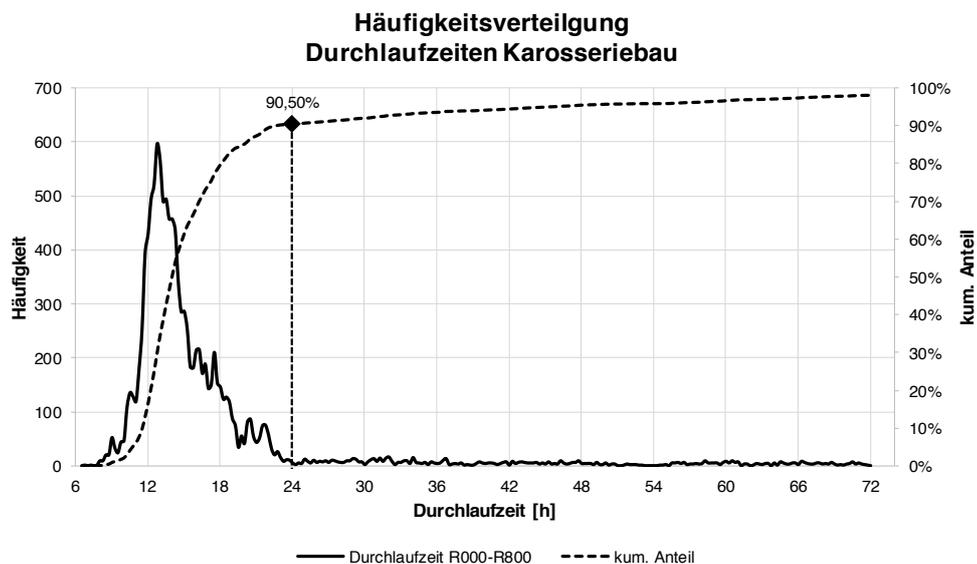


Abbildung 7-2: Häufigkeitsverteilung der Durchlaufzeit von Aufträgen im Karosseriebau

Ein anderes in dem Zusammenhang häufig auftauchendes Problem ist die **ungleichmäßige Verteilung von Produktvarianten** (vgl. Abbildung 7-3), die durch Verwirbelungen in der Produktionsreihenfolge im Karosseriebau entsteht. Dies beeinträchtigt jedoch nicht nur den Karosseriebau selbst, sondern auch die nachfolgenden Produktionsschritte. Die Endmontage unterliegt z.B. in der Regel zahlreichen Restriktionen hinsichtlich der Bearbeitungsreihenfolge unterschiedlicher Varianten (z.B. aufgrund unterschiedlicher Bearbeitungszeiten). Unerwartete Ereignisse und Störungen im Karosseriebau lösen daher häufig Produktionsunregelmäßigkeiten aus, die später zu

weiteren Problemen in der Montage führen können. Dazu zählt unter anderem auch, dass die Versorgung der Montagelinien mit Bauteilen durch die im nächsten Kapitel betrachteten Routenzugsysteme nicht gewährleistet werden kann, wenn der Produktmix zu stark vom Ziel abweicht.

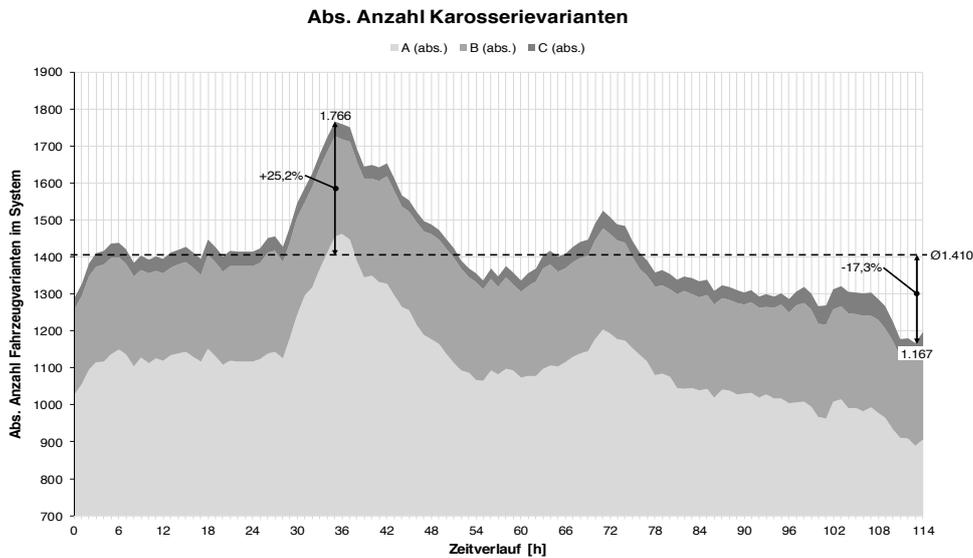


Abbildung 7-3: Umlaufbestand der drei Produktvarianten A, B und C

Im Rahmen der Produktionssteuerung werden solche Abweichungen häufig durch logistische Störungen wie z.B. verspätetes Material verursacht (vgl. Abschnitt 7.1.2). Die Disponenten wirken dem entgegen, indem sie z.B. andere Aufträge freigeben, um die Produktion am Laufen zu halten. Diese Entscheidungen sind jedoch in der Regel nicht vollständig datengetrieben und müssen schnell getroffen werden, was zu den dargestellten Ungleichverteilungen führt. Auch diese Beobachtungen entsprechen den Erfahrungen des Autors aus verschiedenen Fabriken von unterschiedlichen Herstellern und scheinen repräsentativ für den Automobilsektor zu sein.

7.1.2 Produktionssteuerung nach dem Perlenkettenprinzip

Untersuchungen zur Steuerung der Auftragsreihenfolge in der Automobilproduktion wie in [Mey04] und [GMG07] zeigen, dass die Automobilhersteller im Detail zwar unterschiedliche Konzepte verwenden, jedoch alle eine gemeinsame Zielgröße verfolgen: die stabile **Einhaltung der Einplanungsreihenfolge für die Montage** [Mei09]. Um dies zu erreichen wird mehrere Tage vor dem physischen Produktionsbeginn eine für die Fahrzeugendmontage optimierte Auftragsreihenfolge – die sogenannte Perlenkette – festgelegt [Wey02]. Das Ziel des hierauf basierenden Perlenkettenprinzips ist es, diese geplante Auftragsreihenfolge während des gesamten Fertigungsprozesses beizubehalten oder bis spätestens zum Montageeinlauf wiederherzustellen [Gün17]. Die dafür benötigten Materialien werden vorab kommissioniert und in der vorgegebenen Reihenfolge an den Fertigungslinien angestellt. Dieses Konzept wird in einer ähnlichen Weise u.a. bei Ford, Mercedes, BMW, Porsche, Audi und anderen Automobilherstellern verwendet [Mei09].

Durch Verwirbelungen kommt es zu Reihenfolgeabweichungen und logistischen Ineffizienzen [Mei09]. Dazu gehören hohe Auftragsdurchlaufzeiten und Durchlaufzeitschwankungen sowie

hohe Bestände an unfertigen Erzeugnissen, die einen hohen Steuerungsaufwand verursachen. Darüber hinaus entsteht die Notwendigkeit investitionsintensive Sequenzrichter in Form von Karosserienspeichern zu installieren und die in späteren Prozessschritten folgenden Bauteile entsprechend der veränderten Auftragsreihenfolge neu zu ordnen bzw. zu kommissionieren [Gri12]. Die wesentlichsten Ursachen für eine instabile Auftragsfolge sind in Abbildung 7-4 zusammengefasst.

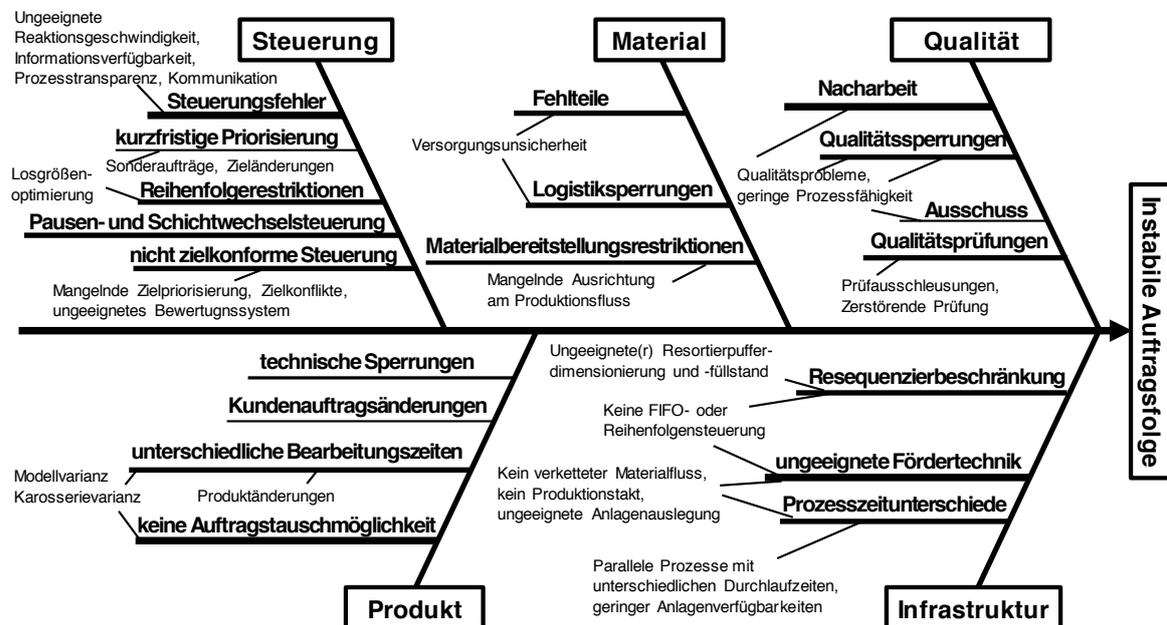


Abbildung 7-4: Ursachen einer instabilen Auftragsfolge i.A.a. [GMG07]

Am häufigsten sind die Ursachen für Verwirbelungen in der Prozesssteuerung zu finden. Zur Steuerung einer komplexen Automobilproduktion muss eine Vielzahl von Informationen wie Pufferfüllstände, Reihenfolgeabweichungen und Störungen permanent verfügbar sein. Aufgrund unzureichender Informationen und mangelhafter Kommunikation kommt es jedoch häufig zu Steuerungsfehlern, die zu einer instabilen Reihenfolge führen. Eine nicht zielkonforme Steuerung kann ebenso zu Reihenfolgeverwirbelungen führen, wenn z.B. die Ziele einzelner Gewerke kollidieren oder nicht untereinander abgestimmt werden. Weitere Ursachen für eine instabile Auftragsfolge sind Engpässe in der Materialversorgung, z.B. aufgrund von Fehlteilen oder Logistiksperrungen. Qualitätsseitige Ursachen sind z.B. Verzögerungen durch Nacharbeit oder Qualitätsprüfungen. Produktseitig sind z.B. hohe Modellvarianz oder technische Sperrungen oft ein Grund für eine schwankende Reihenfolge. Und schließlich kann auch aufgrund der Infrastruktur eine Sequenzverletzung stattfinden, z.B. weil physische Beschränkungen eine Neusortierung verhindern [Mei09].

Ziel des zu entwickelnden Agentensystems muss es folglich sein, die Potenziale einer verbesserten Steuerung zu nutzen und damit die Ursachen für eine instabile Auftragsfolge zu reduzieren. Dafür kommt einerseits die Erstellung der Ausgangsreihenfolge, d.h. der Perlenkette, und andererseits die Auftragsfreigabe und Ressourcenzuweisung in Betracht. Für ersteres werden in modernen Fabriken die folgenden Scheduling-Verfahren angewendet.

7.1.3 Scheduling-Verfahren zur Reihenfolgebildung im Fahrzeugbau

Damit eine effiziente und effektive Massenproduktion hochvariabler Produkte in Mischfahrweise realisiert werden kann, bedarf es einer sorgfältig kalkulierten Planreihenfolge als Grundlage. Die zugrundeliegenden Optimierungsprobleme werden im Bereich des Operations Research als **Flow-Shop-Probleme** behandelt und fallen in die Kategorie der diskreten und kombinatorischen Optimierung [WKS06]. Forschung und Industrie haben eine breite Palette von Ansätzen erarbeitet, um gute Lösungen in einer angemessenen Zeit zu erreichen. Dazu zählen z.B. Simulated Annealing, Tabu-Search und Genetische Algorithmen [JRC09]. Weitere Ansätze sind z.B. Rea-loptionen aus dem Finanzbereich, Fuzzy-Goal-Programmierung sowie Partikelschwarmoptimierung [RRT07, RRJ07, MRT07]. Die verwendeten Ansätze berücksichtigen die Restriktionen des zugrundeliegenden Produktionssystems und ermöglichen die Ermittlung nahezu optimaler Lösungen. Durch kontinuierliche Weiterentwicklung haben sie in den letzten Jahrzehnten erhebliche Verbesserungen erfahren [SCN08]. Für die Planung der Auftragsreihenfolge auf gemischten Montagelinien haben sich drei Hauptansätze herausgebildet: Mixed-Model-Sequencing, Car-Sequencing und Level-Scheduling [BFS09].

Mixed-Model-Sequencing

Als belastungsorientierter Ansatz konzentriert sich das Mixed-Model-Sequencing auf die Vermeidung oder Minimierung von sequenzbedingter Überlastung an einzelnen Arbeitsplätzen. Dieser Ansatz integriert explizit Betriebsmerkmale wie Taktzeiten, Personalrestriktionen, Stationsgrenzen und ähnliche Faktoren [SaP01]. Um teure Bandstillstände zu vermeiden, wird versucht eine Sequenz zu bilden, in der sich Aufträge mit hohem und geringem Arbeitsaufwand abwechseln [BFS09]. Das Mixed-Model-Sequencing ermöglicht daher einerseits eine hohe Genauigkeit, erfordert andererseits aber einen erheblichen Aufwand bei der Datenerfassung [SaP01].

Car-Sequencing

Im Gegensatz dazu erfordert der Ansatz des Car-Sequencing deutlich weniger Aufwand, da er die oben genannten Betriebsmerkmale implizit und nicht explizit berücksichtigt. Dies erfolgt mithilfe von Sequenzierungsregeln vom Typ $H_o:N_o$, d.h. in einer Sequenz von N_o Aufträgen dürfen sich maximal H_o Vorkommnisse des betrachteten Typs befinden [SaP01]. Am Beispiel von Schiebedächern bedeutet ein Verhältnis von 1:3, dass auf eine mit einem Schiebedach ausgestattete Karosserie mindestens drei Aufträge folgen müssen, die kein Schiebedach benötigen, z.B. aufgrund einer beschränkten Anzahl an Manipulatoren für den Einbau. Die erfolgreichsten Implementierungen dieses Ansatzes verwenden Heuristiken wie z.B. Greedy-Algorithmen, Local-Search-Methoden, genetische Algorithmen und Ameisenkolonie-Optimierungsmethoden [SCN08]. Insgesamt ist das Car-Sequencing damit ein nützlicher Ansatz, der in der Praxis häufig verwendet wird, obwohl er mit dem Kompromiss einer geringeren Genauigkeit einhergeht.

Level-Scheduling

Der letzte der drei Hauptansätze strebt nicht die Optimierung der Arbeitsbelastung an, sondern nimmt Just in Time-Ziele in den Fokus. Ziel ist es, den stark vom Produktionsablauf abhängigen Materialbedarf möglichst gleichmäßig über den Planungshorizont zu verteilen. Dafür werden Soll-Produktionsraten für alle Materialien definiert und Produktvarianten so sequenziert, dass Abweichungen zum Zielverhältnis im betrachteten Zeitraum minimiert werden. Dieser Ansatz entstammt dem Toyota Produktionssystem und ist ebenfalls in der Praxis weit verbreitet [BFS07b].

Zusammenfassend lässt sich sagen, dass eine Vielzahl von Ansätzen für die Sequenzierung in der Variantenfließfertigung existiert und dass sowohl im akademischen Bereich als auch in der Industrie große Anstrengungen unternommen werden, um die gegebenen Werkzeuge weiter zu verbessern. In der Praxis ist zudem zu beobachten, dass sich Lösungen wie das Car-Sequencing bei vielen Automobilherstellern durchgesetzt haben, da sie einen geringeren Rechenaufwand verursachen und für den Menschen besser interpretierbare Ergebnisse liefern.

Herausforderungen in der Reihenfolgebildung

Die oben genannten Methoden können in angemessener Zeit nahezu optimale Lösungen für die Auftragsreihenfolge ermitteln. Eine Herausforderung ist es jedoch diese geplante Auftragsreihenfolge im realen Produktionsumfeld über einen längeren Zeitraum einzuhalten. Das liegt daran, dass Störungen wie z.B. Eilaufträge, Nacharbeit, Fehlteile und Maschinenausfälle den Ablauf beeinflussen und damit auch die Fertigungsreihenfolge verändern. Ein Mittel um auf die veränderten Bedingungen zu reagieren ist die Resequenzierung. Dadurch wird die Auftragsreihenfolge nachträglich angepasst, um z.B. die Überlastung an einzelnen Stationen zu reduzieren und Leertakte in der Produktion zu vermeiden. Der **Resequenzierung** wurde in der Vergangenheit jedoch vergleichsweise wenig Beachtung geschenkt, sodass hier Raum für weitere Forschung existiert [BFS07a].

Sotskov und Werner fassen in [WeS14] aktuelle Herausforderungen bei der Produktionsplanung und Sequenzierung zusammen. Sie konzentrieren sich dabei insbesondere auf den Umgang mit **ungenauen Daten** – ein Zustand, der in der Automobilproduktion aufgrund der Größe und Komplexität unvermeidbar ist. Dabei halten die Autoren unter anderem fest, dass sich die Mehrzahl der Studien z.B. zum Line Balancing, auf die Optimierung eines einzelnen Leistungsmaßes, wie z.B. der Zykluszeit, konzentrieren. Maschinenausfälle oder Fehlzeiten erfahrener Mitarbeiter können die Leistung von Anlagen jedoch erheblich beeinträchtigen, sodass es schnell zu Leistungseinbußen kommen kann. Ein vielversprechendes Forschungsgebiet stellen folglich Modelle dar, die z.B. kapazitäts- oder kostenbasierte Ziele mit einem **Robustheitskriterium** verbinden, da sie die Anforderungen der Industrie besser abbilden [HaD14]. Durch einen verbesserten Umgang mit den Auswirkungen von Störungen in der Produktion und einer höheren Robustheit von Produktionsprozessen können demnach noch erhebliche Potenziale gehoben werden, was auch im re-

alen Fabrikbetrieb deutlich wird. In der Praxis ist zu beobachten, dass viele Hersteller Schwierigkeiten haben die durch das Perlenkettenkonzept vorgegebene Reihenfolge einzuhalten, da Störungen und komplexe, parallele Produktionslinien zu Abweichungen von der Planreihenfolge führen [BoZ13]. Dies hat oft weitere Verwerfungen zur Folge, denn eine neue Sequenz kann unter Echtzeitbedingungen nicht in ausreichender Güte berechnet werden [BrK12]. Da das flexible Flow-Shop-Problem zur Komplexitätsklasse **NP-schwer** gehört [TDT14], wird dafür deutlich mehr Rechenzeit benötigt als Reaktionszeit verfügbar ist. Um eine neue Sequenz zu ermitteln brauchen heutige Algorithmen Laufzeiten von etwa 30 Minuten [DFS12]. Eine Begrenzung der Laufzeit wie z.B. bei der ROADEF'05 Challenge auf 10 Minuten [SCN08] ermöglicht zwar eine bessere Reaktionsfähigkeit, führt aber in der Regel zu einer geringeren Lösungsqualität. In dem Beitrag [JRC09] werden solche Probleme allgemein als **flexibles Flow-Shop-Problem mit unabhängigen Parallelmaschinen** betrachtet, d.h. es gibt in jeder Phase unterschiedliche Parallelmaschinen und die Geschwindigkeiten der Maschinen sind von den Aufträgen abhängig. Dies trifft auf den hier betrachteten Karosseriebau, aber auch auf die Montage zu. Fortschrittliche Lösungsansätze wie z.B. der Insertion-Sort-Algorithmus NEH erzielen in gewissen Szenarien sehr gute Ergebnisse. Trotz der Schwierigkeit des zugrundeliegenden Optimierungsproblems kann eine nahezu optimale Lösung für Probleme mit einer Größe von bis zu 50 Aufträgen in unter 10 Sekunden ermittelt werden. Auch andere Heuristiken wie Simulated Annealing, Tabu-Search und Genetische Algorithmen ermöglichen gute Lösungsansätze [JRC09]. Ein aktueller Survey von Werner et al. zeigt eine Reihe von Arbeiten auf, die die Rechenzeit mithilfe neuer Ansätze weiter zu reduzieren versuchen [WBS18]. Ein besonders erwähnenswerter Ansatz wird von Fuchigami et al. präsentiert. Die Autoren stellen mehrere Heuristiken zur Maximierung der Ausbringung für Just-in-Time-Probleme in Flow Shops vor und erreichen damit sogar noch bessere Ergebnisse als mit einem NEH-Algorithmus. Im konkreten Fall werden für Probleme mit 100 Aufträgen und 20 Maschinen mit einer mathematischen Methode Berechnungszeiten von 3,3 Minuten erreicht, während eine Enumeration-Method eine Rechenzeit von 1,5 Sekunden aufweist und Heuristiken sogar noch schneller eine Lösung finden [FSR18]. Derzeit sind die benannten Lösungen jedoch nicht auf die Größe eines modernen Karosseriebaus mit etwa 2.000 Aufträgen und 200 unterschiedlichen Produktionsschritten ausgelegt, sodass die Ermittlung einer geeigneten Lösung in Echtzeit nicht gewährleistet werden kann. Die Entwicklungen in diesem Bereich sollten jedoch weiter genau beobachtet werden.

Als Folge der Dynamik komplexer Produktionssysteme und der derzeit unzureichenden Echtzeitfähigkeiten hat sich der Zustand eines Produktionssystems häufig bereits verändert bevor eine Lösung berechnet wurde. Daher ist es für ein robustes Produktionssystem unumgänglich, sich selbstständig an Störungen anzupassen zu können, sobald sie auftreten. Herkömmliche **monolithische Unternehmenssoftware** ist jedoch nicht für die Bewältigung unerwarteter Ereignisse ausgelegt, da sie auf einem sehr detaillierten Regelwerk für sehr spezifische Situationen basiert.

Dies ermöglicht zwar hervorragende Ergebnisse unter vordefinierten Bedingungen. Bei Ereignissen, die nicht spezifisch im Voraus definiert wurden, führt es jedoch zu einer schlechten Performance, da monolithische Systeme nicht adäquat und rechtzeitig auf solche Situationen reagieren können. Obwohl die meisten Automobilfabriken die Möglichkeit haben, Produktionsaufträge zu einem gegebenen Zeitpunkt virtuell oder physisch neu zu sequenzieren, können sie nur geplante, vordefinierte Operationen durchführen wie z.B. die Anpassung der Farbbausteine vor dem Eintritt in die Lackiererei [BSW12]. Ungeplante Resequenzierungen führen in der Regel zu Abweichungen von der ursprünglich geplanten Reihenfolge und vom geforderten Produktmix, was Probleme im späteren Produktionsverlauf verursacht. Inkonsistente Datentypen können solche Komplikationen noch verstärken [LSH17b]. Die Komplexität des Gesamtsystems macht es zudem **menschlichen Operatoren** sehr schwer die benötigten Informationen in Echtzeit zu verarbeiten. Da die Entscheidungsträger in solchen Situationen aber oft Menschen sind, werden häufig nur suboptimale Ergebnisse erreicht. Agentenbasierte Ansätze könnten hier durch schnelle, dezentrale Entscheidungen für Abhilfe sorgen.

Zusammenfassend ist festzuhalten, dass aufgrund der für die Ermittlung einer geeigneten Lösung erforderlichen Rechenzeit eine ausreichend schnelle Reaktion auf unvorhergesehene Störungen derzeit nicht gegeben ist. Die Ergänzung der fortschrittlichen Lösungsansätze aus dem Operations Research mit der Agilität von agentenbasierten Systemen, die schnell und autonom auf Störungen reagieren, könnte hier Abhilfe schaffen und eine Brücke zwischen einer optimalen Lösung und einer schnellen, agilen Reaktion auf Veränderungen darstellen. Die hierfür auszuführenden Steuerungsaufgaben gehören im vorliegenden Anwendungsfall der Karosseriefolgesteuerung zum Fertigungssteuerungsbereich der PPS und genauer zur Reihenfolgesteuerung. Beispiele für Agentensysteme aus diesem Bereich werden im Folgenden vorgestellt.

7.1.4 Agentensysteme für die Reihenfolgesteuerung

Die folgenden drei Beispiele stellen eine Auswahl von Agentensystemen in dem für den Anwendungsfall relevanten Bereich der Reihenfolgesteuerung dar und werden herangezogen, um eine prinzipielle Validierung des Agentenansatzes für die Karosseriefolgesteuerung durchzuführen.

Gepäckfördersystem für Flughäfen von Siemens

Das Routing von Gepäckstücken an großen Flughäfen ist eine komplexe Herausforderung. Es existieren oft mehrere potenzielle Wege von der Gepäckabgabe bis zum vorgesehenen Flugsteig und die Auslastung der Strecken variiert. Die Siemens AG hat ein agentenbasiertes System zur Steuerung solcher Gepäckförderanlagen realisiert, um die Potenziale eines dezentralen Ansatzes zu erschließen. In dem System repräsentieren Transportgut-Agenten die Gepäckstücke. Sie erhalten über eine Schnittstelle zum Flughafen-Managementsystem den Zielflugsteig und ermitteln an jeder Verzweigung den aktuell schnellsten Weg. Dafür kommunizieren sie mit Fördertechnik-Agenten und beziehen so die momentane Auslastung der Strecken mit ein. Weitere Dienste

wie z.B. Check-In, Gepäckprüfung und Umleitung für Anschlussflüge werden entweder in Auktionen zwischen dem Transportgut-Agenten und den Dienstleistern oder in Verhandlungen zwischen den Dienstleistern selbst bestimmt [EHH10].

Elektrohängebahnsystem der Technischen Universität München

Im Rahmen des BMBF-Projektes „Internet der Dinge“ wurde eine Elektrohängebahn mit der Zielsetzung realisiert, Transporte ohne einen zentralen Materialflussrechner umzusetzen. Dazu wurden in einem Demonstrator zwei Laufkatzen und ein Transportsystem aus Rollenförderern und Weichen mittels Agenten abgebildet. In dem System werden die Transportaufträge dynamisch zwischen den Entitäten verhandelt und Daten wie z.B. der Aufenthaltsort anderer Fahrzeuge, Wegreservierungen oder manuell eingetragene Streckensperrungen mittels Blackboard-Ansatz allen Teilnehmern verfügbar gemacht. Der schnellste Weg zum Ziel wird durch die Agenten der Transportkatzen selbst bestimmt. Sie übernehmen alle Aufgaben zur Durchführung der Wegplanung, des Transports und des Lastwechsels. Mit dem System konnte die Layoutflexibilität des Systems sowie die Erreichbarkeit von Lager- und Produktionseinrichtungen verbessert werden [CKG10].

PABADIS'PROMISE-basiertes Steuerungssystem für die Fahrzeugfertigung von FIAT

Ein im Bereich der Fahrzeugfertigung entstandenes Agentensystem ist das im Rahmen des PABADIS'PROMISE-Projektes entwickelte System beim Automobilhersteller FIAT. Ähnlich wie der hier betrachtete Fall der Karosseriefolgesteuerung setzt sich der Anwendungsfall von FIAT mit der gewerkeübergreifenden Fertigung von Automobilen auseinander. Die Projekte unterscheiden sich jedoch hinsichtlich der Zielsetzung. Bei FIAT sollten vor allem zwei Anforderungen erfüllt werden. Erstens sollten Kunden den Produktionsfortschritt ihres Fahrzeugs mithilfe von RFID jederzeit verfolgen und die Bestellung während der Produktion ändern können. Daher besteht mithilfe des Agentensystems die Möglichkeit die Fahrzeugkonfiguration auch nach Beginn der Produktion noch anzupassen, um z.B. ein anderes Navigationsgerät zu bestellen. Daraufhin übernimmt das Agentensystem dynamisch die Umverteilung der Kundenaufträge und berücksichtigt alle Auswirkungen auf die Lieferkette und/oder die internen Bestände. Zweitens wurde mit dem Ansatz das Ziel verfolgt das Produktionssystem möglichst effizient auf die Fertigung neuer Fahrzeugmodelle umzustellen. Die Umstellung von einem alten Modell auf ein neues bedarf im Normalfall einer Neukonfiguration der Linien, die mit einem hohen Entwicklungsaufwand verbunden ist. Um eine schnelle Anpassung zu ermöglichen wurde das Agentensystem so konzipiert, dass es selbstständig die Rekonfiguration jedes Arbeitsplatzes entlang der Montagelinie vornehmen und die Anlagen mit neuen Parametern einstellen kann. Für jedes Gewerk besteht eine eigene Agentenstruktur, mit der alle Aufträge innerhalb des Gewerks verwaltet werden. Die Steuerung der Lager wird durch Ressourcenagenten übernommen. Das resultierende Produktionssteuerungssystem verfügt über eine deutlich höhere Flexibilität und Rekonfigurierbarkeit als bisherige konventionelle Systeme [PAB08].

7.1.5 Zielsetzung

Die oben aufgeführten Beispiele von Agentensystemen in ähnlichen Anwendungsgebieten veranschaulichen das grundsätzliche Potenzial des Agentenansatzes für den betrachteten Fall der Karosseriefolgesteuerung. Die Eigenschaften der Agenten ermöglichen es hochflexible, adaptive, rekonfigurierbare und damit auch robuste Produktionssysteme zu realisieren [Roi10, Uni15a]. Durch Agenten alleine werden jedoch keine qualitativ höherwertigen Lösungen für Optimierungsprobleme erreicht als z.B. mit den in Abschnitt 7.1.3 vorgestellten Ansätzen zu Flow-Shop-Problemen [Mön06].

Darüber hinaus existieren zwei weitere Implikationen, die in dem zu erarbeitenden Agentensystem Beachtung finden müssen. Erstens werden bei Störungen Steuerungsentscheidungen auch in modernen Produktionssystemen trotz ihrer Komplexität und Echtzeitanforderungen zum Teil von menschlichen Entscheidungsträgern getätigt. Und zweitens ist die Rechenzeit ein entscheidender Faktor für den Erfolg eines Produktionssteuerungssystems im Fertigungsbereich. In der Erfahrung des Autors variiert die Häufigkeit der Sequenzbildung in Abhängigkeit von Hersteller und Werk zwischen einmal pro Tag bis zweimal pro Stunde. Angesichts der Tatsache, dass hochkomplexe Produktionssysteme mit mehreren tausend Lieferanten und hunderten von internen Produktionsressourcen einer erheblichen Anzahl von unerwarteten Störungen ausgesetzt sind, erscheint es sinnvoll, die Autonomie in diesem Bereich zu erhöhen. Da die vorgestellten Sequenzierungsstrategien jedoch bereits nahezu optimale Lösungen bieten und keine signifikanten Laufzeitverbesserungen zu erwarten sind, ist es schwierig, in diesem Bereich große Produktivitätssprünge zu erzielen. Es ist jedoch davon auszugehen, dass durch Erhöhung der Flexibilität und Autonomie von Produktionssystemen wesentliche Verbesserungen realisiert werden können. Daher soll im Weiteren ein Produktionssteuerungssystem umgesetzt werden, das die Stärken beider Ansätze nutzt, indem es bestehende Planungsalgorithmen und Agentensysteme kombiniert. So kann das System einerseits in regelmäßigen Abständen neue, nahezu optimale Produktionsreihenfolgen für den nächsten Planungszeitraum berechnen, die der aktuellen Situation am besten entsprechen und sich damit iterativ an Veränderungen anpassen. Andererseits können unerwartete Störungen, die eine Reaktion in Echtzeit erfordern, von dem Agentensystem gehandhabt werden, was zu einem hochgradig optimierten und adaptiven System führt. Die Entwicklung dieses Systems ist Inhalt des nächsten Abschnitts.

7.2 Entwicklung des Agentensystems

Das Konzept des zu erstellenden Agentensystems wird in den folgenden drei Abschnitten mithilfe der DACS-Methode erarbeitet. In Abschnitt 7.2.1 werden dafür zunächst die Steuerungsentscheidungen und die zwischen ihnen bestehenden Abhängigkeiten betrachtet. Das daraus abgeleitete Entscheidungsmodell ist die Grundlage für die Definition von Agenten und das Skizzieren eines Agentenmodells im nachfolgenden Abschnitt 7.2.2. In Abschnitt 7.2.3 werden schließlich die Interaktionen zwischen den Agenten untersucht und geeignete Protokolle für sie ausgewählt.

7.2.1 Entscheidungsmodell

Analyse der Steuerungsentscheidungen

Zur Untersuchung der Steuerungsentscheidungen wurde auf der GFSS und der Zusammenstellung der Steuerungsaufgaben und -informationen aus Kapitel 5 aufgesetzt. Dazu wurden zunächst die wesentlichen Entscheidungen identifiziert und anschließend auf Basis der studentischen Arbeit [Bec16] die Prozesse im Status quo aufgenommen und ausgewertet. Zu den untersuchten Aufgaben gehört die Erstellung der Karosseriebausequenz, die Auftragsfreigabe sowie die physische und virtuelle Resequenzierung von Aufträgen. Da die Erstellung der Karosseriebausequenz im vorliegenden Anwendungsfall als gegeben angenommen wird, ist die erste zu treffende Entscheidung im vorliegenden Karosseriebau die Freigabe des nächsten Fertigungsauftrags im System. Die parallelen Fertigungslinien führen im Weiteren dazu, dass an den Ein- und Ausgängen der Fertigungsabschnitte entschieden werden muss, welcher Auftrag auf welcher Ressource gefertigt werden soll (vgl. Abbildung 7-5).

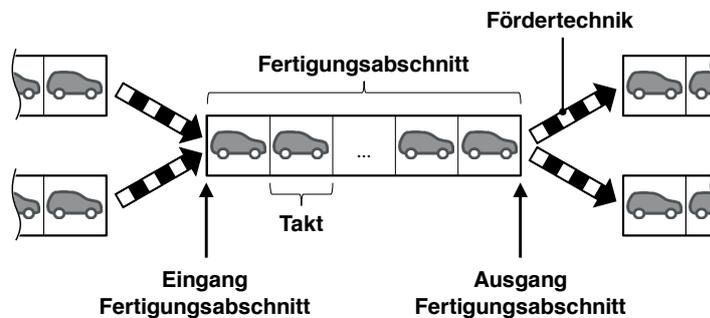


Abbildung 7-5: Materialfluss im Karosseriebau

Die Steuerungsentscheidungen, die in dem System notwendig sind um einen optimalen Karosserienfluss zu gewährleisten, können daher auf die folgenden vier grundlegenden lokalen Entscheidungen (E) heruntergebrochen werden (vgl. Abbildung 7-6).

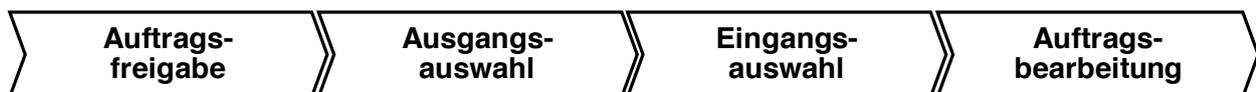


Abbildung 7-6: Prozessschritte zur Karosseriefolgesteuerung

Für alle Steuerungsentscheidungen wurden detaillierte Analysen mithilfe der GFSS erarbeitet. Als Beispiel ist hier der in Abbildung 7-7 dargestellte Entscheidungsprozess zur Auftragsfreigabe aufgeführt. Die verbleibenden Entscheidungsprozesse sind in Anhang 18-20 zu finden.

Für die **Auftragsfreigabe (E1)** muss zunächst ein Auftrag aus dem existierenden Produktionsprogramm ausgewählt werden, der in der aktuellen Situation gefertigt werden kann. Bevor die finale Freigabe für einen Auftrag gegeben werden kann, wird geprüft ob der Auftrag aktuellen Sperrungen unterliegt. Ist das nicht der Fall, wird der Auftrag freigegeben und es erfolgt eine Kennnummernvergabe (z.B. auf dem Querträger der Karosserie) und falls notwendig werden Kommissionieraufträge verteilt. Nach der Auftragsfreigabe muss entschieden werden auf welcher Ressource der Auftrag gefertigt werden soll, damit er auf die richtige Förderstrecke gelangt. Diese

Entscheidung entspricht der **Ausgangsauswahl (E2)** in Abbildung 7-6. Dafür wird die Art des nächsten Produktionsschritts und die Verfügbarkeit passender Ressourcen, die diesen Schritt ausführen können geprüft. Auf Basis dieser Daten wird verglichen welche Ressource für welchen Auftrag unter Berücksichtigung aller verfügbaren Informationen die geeignetste ist. Die dritte Entscheidung bezieht sich auf die **Eingangsauswahl (E3)**, um zur nächsten Förderstrecke zu gelangen, die den Auftrag zur vorgesehenen Ressource transportiert. Hierfür müssen Produktionsanfragen gesammelt und bewertet werden. Anschließend erfolgt die Auswahl des am besten geeigneten Auftrags. Im vierten Schritt erfolgt im Rahmen der **Auftragsbearbeitung (E4)** die Ausführung des eigentlichen Fertigungsschritts. Ferner werden unter Umständen weitere Informationen wie z.B. Statusmeldungen an relevante Instanzen versendet. Sobald der Bearbeitungsschritt abgeschlossen ist, muss der Auftrag erneut einem Ausgang zugewiesen werden (E2). Sind alle Fertigungsschritte ausgeführt, geht der Auftrag in die Lackiererei über.

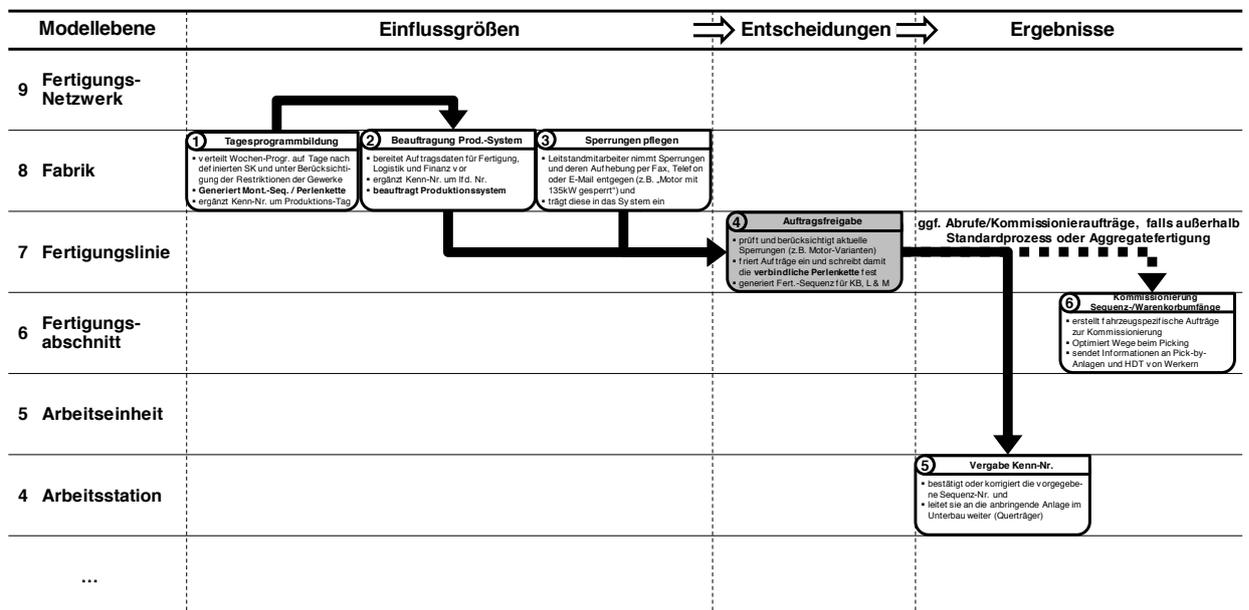


Abbildung 7-7: Auftragsfreigabe Karosseriebau im Status quo i.A.a. [Bec16]

Um die Steuerungslogik in dem System aufzuwerten, bietet die DACS-Methode über lokale Steuerungsentscheidungen hinaus auch die Möglichkeit verteilte, abstrakte Entscheidungen zu entwerfen. Für die Auftragsauswahl und Ressourcenzuweisung sollten idealer Weise alle in Frage kommenden Aufträge und Ressourcen zum Zeitpunkt der Entscheidungsfindung berücksichtigt werden. Daher wird als zusätzliche fünfte Entscheidung die **Zuordnung (E5)** von Aufträgen zu Ressourcen als verteilte Steuerungsentscheidung definiert. Darin werden die gesammelten Aufträge mit den zur Verfügung stehenden Ressourcen abgeglichen und eine unter den zugrundeliegenden Optimierungskriterien bestmögliche Zuordnung getroffen. Die Steuerungsentscheidungen sind in Tabelle 7-2 zusammengefasst und ihre Beziehung zueinander sind in Abbildung 7-8 dargestellt.

ID	E1	E2	E3	E4	E5
Titel	Auftragsfreigabe (Bandauflage)	Ausgangsauswahl (Ressourcen- auswahl)	Eingangsauswahl (Auftragsauswahl)	Auftragsbearbeitung	Zuordnung
Para- meter	Ressource R_i	Ressourcen $R_i - R_j$	Aufträge $A_i - A_j$	Auftrag A_i	Ressourcen $R_i - R_j$, Aufträge $A_i - A_j$
Steuerungs- interface	Steuerungsinterface der Ressource	Steuerungsinterface der Fördertechnik	Steuerungsinterface der Fördertechnik	Steuerungsinterface der Ressource	virtuell, da verteilte Entscheidung
Auslöser	Ressource ist verfügbar geworden	Auftrag A_i erreicht (gleich) Weiche der Fördertechnik	Auftrag A_i am Eingang 1 und A_j am Eingang 2 erreichen Ressource (fast) zeitgleich	Auftrag A_i erreicht Ressource	Auftrag A_i bereit für nächste Operation
Entschei- dungsraum	Menge verfügbarer Aufträge/Karosserien in Produktionsprogramm	Menge verfügbarer Ausgänge {Ausgang 1, Ausgang 2}; ({Ressource 1, Ressource 2})	Menge verfügbarer Eingänge {Eingang 1, Eingang 2}; ({Auftrag 1; Auftrag 2})	Menge verfügbarer Operationen M_i an der Ressource	Menge verfügbarer Ein- und Ausgänge
Lokale Ent- sch.-Regel	Den nächsten verfügbaren Auftrag A_i aus Produktionsprogramm wählen	Den ersten freien Ausgang (bzw. die erste Ressource) wählen	Den Auftrag mit der geringsten Zeit bis zum Liefertermin wählen	Die maximale Menge verfügbarer Operationen ausführen	-

Tabelle 7-2: Steuerungsentscheidungen Karosseriefolgesteuerung

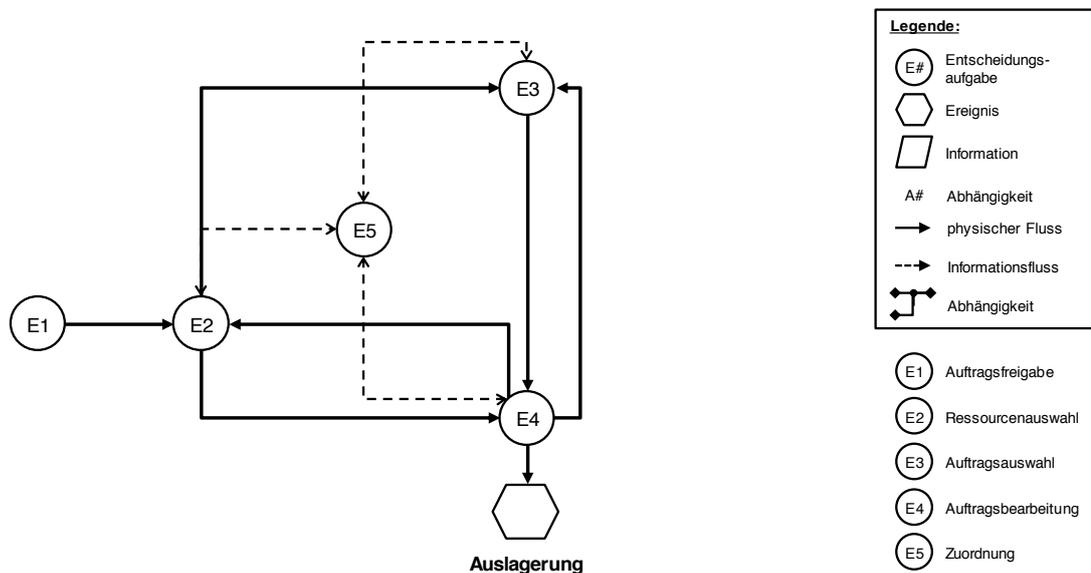


Abbildung 7-8: Steuerungsentscheidungen Karosseriefolgesteuerung

Identifikation von Abhängigkeiten

Die identifizierten Steuerungsentscheidungen müssen im nächsten Schritt um die zwischen ihnen bestehenden Abhängigkeiten ergänzt werden, damit ein aussagekräftiges Entscheidungsmodell entstehen kann. Die Abhängigkeiten (A) in dem System können zu den vier in Tabelle 7-3 dargestellten übergreifenden Punkten verdichtet werden.

ID	A1	A2	A3	A4
Titel	Liefertreue	Auslastung	Engpassvermeidung	Zuordnung
Entscheidungen	E1, E2, E3	E2, E3, E4	E2, E3, E4	E2, E3, E5
Restriktionen	Verspätungen und Schleppe vermeiden; Abweichungen vom Liefertermin minimieren	Leerlauf vermeiden; ungleichmäßige Belastung von Ressourcen vermeiden	Blockaden/Staus durch ungeeignete Varianten vermeiden	Nur Ressourcen wählen, die die Variante bearbeiten können; Aufträge nach Priorität bearbeiten
Präferenzen	Zuweisung der Aufträge mit dem frühesten Liefertermin zu freien Ressourcen	Freie Ressourcen immer mit Aufträgen versehen; Ressourcenauslastung maximieren	Reihenfolge optimieren, um Staus zu vermeiden; Status nachfolgender Ressourcen einbeziehen	Ressourcen bestmöglich ausnutzen; Aufträge so früh wie möglich bearbeiten

Tabelle 7-3: Abhängigkeiten von Entscheidungen der Karosseriefolgesteuerung

Die grundsätzlichen Ziele der Produktion und Logistik lassen sich im Kern reduzieren auf die Erreichung von Wirtschaftlichkeit durch die Maximierung der logistischen Leistung und die Minimierung des Faktoreinsatzes [Mei09]. In Bezug auf den vorliegenden Anwendungsfall entspricht dies einerseits der Einhaltung der **Liefertreue (A1)** für die betrachteten Aufträge und andererseits der Maximierung der **Auslastung (A2)** der verwendeten Ressourcen. Zur Einhaltung der Liefertreue muss sichergestellt werden, dass Aufträge mit einer kurzen Zeit bis zum Liefertermin bzw. einer hohen Priorität bevorzugt behandelt werden (E1). Verspätungen und ungleichmäßige Durchlaufzeiten bzw. die Entstehung einer Schleppe sind zu vermeiden (E2, E3). Zur maximalen Ressourcenauslastung und zur Vermeidung von Leerlauf sollten freie Ressourcen jederzeit mit neuen Aufträgen versorgt werden können. Um dies zu erreichen müssen sowohl die Eingänge (E3) als auch die Ausgänge (E2) der Ressourcen optimal angesteuert werden und auch die Auftragsfreigabe (E1) und -bearbeitung (E4) muss abgestimmt sein. Eine aus den beiden Abhängigkeiten A1 und A2 entspringende Anforderung ist die **Vermeidung von Engpässen (A3)**. Dies ist insbesondere zu beachten, wenn zwischen mehreren Aufträgen und/oder Ressourcen gewählt werden kann, z.B. an Kreuzungspunkten wie dem Karosserieframer S40. Um Staus und Blockaden zu vermeiden muss die Ansteuerung von Ein- und Ausgängen (E2, E3) sowie die Auftragsbearbeitung (E4) aufeinander abgestimmt sein. Dies führt zur vierten Abhängigkeit, denn um die zuvor genannten Ziele zu erreichen wird ein zuverlässiger Mechanismus für die **Zuordnung (A4)** von Aufträgen zu Ressourcen benötigt. Dieser sollte die zuvor genannten Abhängigkeiten berücksichtigen, um zwischen konkurrierenden Aufträgen und/oder Ressourcen wählen zu können. Für die Definition eines solchen Mechanismus, werden im nächsten Abschnitt die für die Entscheidungen notwendigen Informationen untersucht.

Für das Treffen von optimalen Entscheidungen werden an relevanten Stellen Informationen benötigt. Um definieren zu können welche Informationen einem Agenten zur Ausführung seiner Aufgaben zur Verfügung stehen müssen, ist daher eine Informationsanalyse notwendig.

Zur Einhaltung der Liefertreue (A1) werden Informationen zum Liefertermin des Auftrags benötigt. Kommt es ferner dazu, dass ein oder mehrere Aufträge um mehrere Ressourcen werben, muss auch berücksichtigt werden, wann die entsprechenden Ressourcen für die Bearbeitung verfügbar werden. Zur Maximierung der Ressourcenauslastung (A2) müssen mehrere Ressourcen aufeinander abgestimmt werden. Dazu werden Informationen benötigt, die Aussagen darüber erlauben, wann die Ressourcen frei werden, welche Produktvarianten sie bearbeiten können und welche Aufträge auf ihnen bereits bearbeitet werden. Um Blockaden und Staus zu vermeiden (A3) müssen die Verfügbarkeit und die Fähigkeiten der nachfolgenden Ressourcen sowie die Eigenschaften der vorgelagerten Aufträge berücksichtigt werden. Wenn z.B. eine Störung vorliegt, die in nachfolgenden Schritten die Bearbeitung einer gewissen Variante verhindert, sollte diese Variante auf vorgelagerten Ressourcen nicht angenommen werden. Für die Zuordnung von Aufträgen zu Ressourcen (A4) werden schließlich Informationen zu den Fähigkeiten der Ressourcen sowie zu den Produktvarianten der Aufträge benötigt. Die identifizierten Entscheidungen können im letzten Schritt der Analyse nun um die Abhängigkeiten und notwendigen Informationen ergänzt werden, was zu dem in Abbildung 7-9 dargestellten vollständigen Entscheidungsmodell führt.

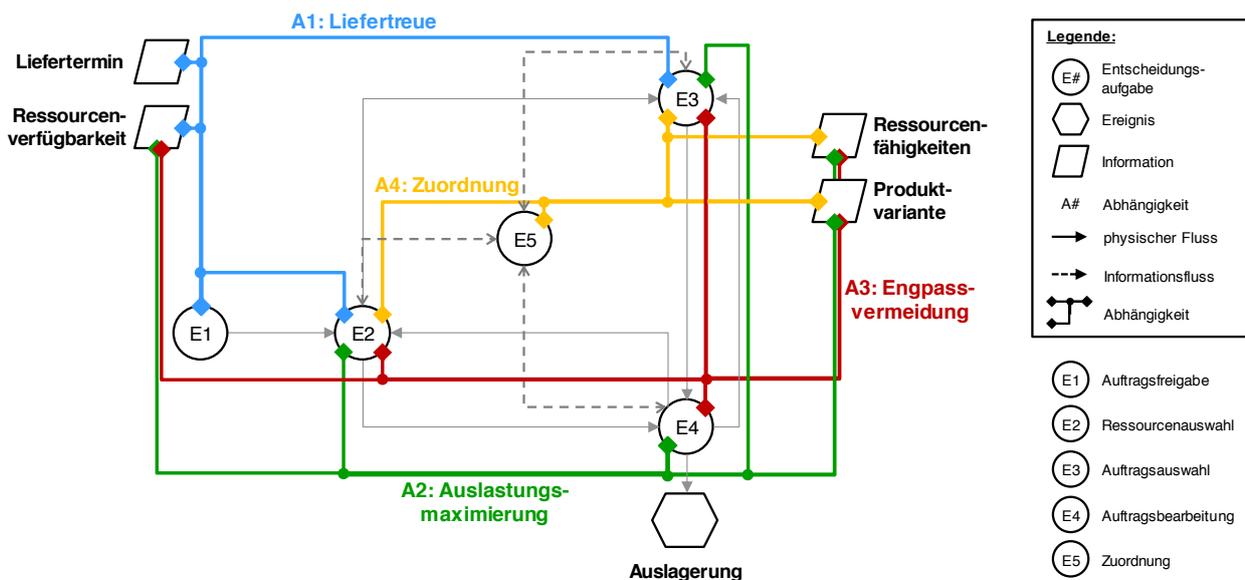


Abbildung 7-9: Entscheidungsmodell Karosseriefolge

7.2.2 Agentenmodell

Für den zweiten Schritt der DACS-Methode – die Entwicklung eines Agentenmodells – erfolgt zunächst eine Clusterung der Entscheidungsaufgaben und im Anschluss, falls notwendig, eine Anpassung des Entscheidungsmodells mit den neu gewonnenen Erkenntnissen. Da die nachträgliche Integration der verteilten Steuerungsentscheidung E5 bereits im vorangehenden Abschnitt erläutert wurde, liegt der Fokus dieses Abschnitts auf dem Bilden der Cluster sowie der Ableitung und Definition darauf basierender Agenten. Ziel ist es dabei die Steuerungsentscheidungen so stark wie möglich voneinander zu trennen, um die Komplexität der einzelnen Agenten

zu minimieren und das Optimierungsproblem bestmöglich zu verteilen. Nach [BJW04] sollten Entscheidungsaufgaben jedoch immer demselben Agenten zugewiesen werden, wenn sie eine oder mehrere der folgenden drei Bedingungen erfüllen: Die Entscheidungsaufgaben

- (a) haben dasselbe Steuerungsinterface,
- (b) verändern oder beeinflussen dieselben Systemkomponenten oder
- (c) sind immer stark voneinander abhängig.

Auf Grundlage des Entscheidungsmodells wird im Weiteren daher ein Agentenmodell erarbeitet, das die Entscheidungsaufgaben so gegliedert, dass sie möglichst sinnvoll in einem Agenten konsolidiert werden können.

Bei der Betrachtung des Entscheidungsmodells fällt auf, dass alle Entscheidungsaufgaben entweder Bezug zum Fertigungsauftrag, zur Ressource oder zu beiden haben. Um die Liefertreue zu optimieren, sollten Agenten existieren, die individuelle Aufträge und ihre jeweiligen Interessen repräsentieren. Ein solcher **Auftragsagent (AA)** sollte Einfluss auf die Auswahl geeigneter Ressourcen (E2) haben, damit er sein Interesse an einer zügigen Bearbeitung durchsetzen kann. Damit es bei der Auftragsfreigabe nicht zu einer Pattsituation zwischen konkurrierenden Auftragsagenten kommt, sollte ein unabhängiger Agent für die Freigabe von Aufträgen (E1) verantwortlich sein. Diese Aufgabe übernimmt der **Scheduling-Agent (SA)**, der als Einziger Zugriff auf die Produktionssequenz hat und die dynamischen Umgebungseigenschaften bei der Freigabe berücksichtigt. Als Gegenstück zum Auftragsagenten sollten Agenten existieren, die das Interesse der Ressourcen an der Auslastungsmaximierung vertreten. Dafür sollten spezielle **Ressourcenagenten (RA)** die Auswahl von Aufträgen (E3) sowie deren Bearbeitung (E4) beeinflussen können. Da die Arbeitseinheiten im vorliegenden Fall durch einen verketteten Materialfluss zu Fertigungsabschnitten zusammengefasst werden, repräsentiert ein RA jeweils einen gesamten Fertigungsabschnitt. Ein letztes Entscheidungscluster bildet die Entscheidungsaufgabe E5. Damit Konflikte zwischen den Interessen von Aufträgen und Ressourcen möglichst optimal gelöst werden können, ist es sinnvoll einen unabhängigen Schiedsrichter zu implementieren, der auf Basis von objektiven Kriterien die bestmögliche Zuordnung vornimmt. Da dieser Agent zwischen Auftrags- und Ressourcenagenten vermitteln muss, wird als letztes der **Mediatoragent (MA)** definiert. Die Auftrags- und Ressourcenagenten führen vordefinierte Aktionen in einem begrenzten Optionsraum aus und sind daher als Reflexagenten zu klassifizieren. Im Scheduling- und Mediatoragenten hingegen ist eine Entscheidungslogik mit einer Nutzenfunktion hinterlegt, weshalb diese zur Klasse der nutzenbasierten Agenten gehören. Die entstehenden vier Haupt-Agenten-Cluster sind in Abbildung 7-10 zusammengefasst.

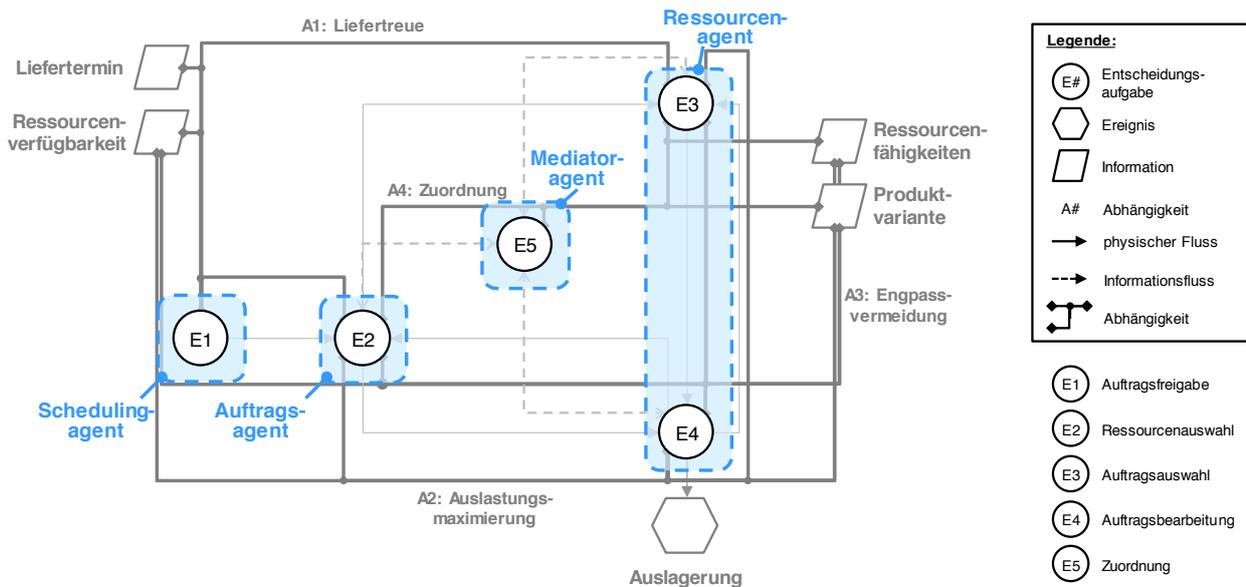


Abbildung 7-10: Entscheidungscluster und Agenten Karosseriefolgesteuerung

Für die auf diesen Clustern aufbauende Entwicklung eines Agentenmodells wird auf die Vorarbeiten aus Kapitel 4.1.6 zurückgegriffen. Die darin beschriebenen allgemeingültigen Entwurfsmuster eignen sich auch in diesem Fall für die Entwicklung eines agentenbasierten Ressourcenallokationsprozesses. In Analogie zu dem Entwurfsmuster in Abbildung 4-1 wird daher ein produkttypbezogener Agent (Produktagent) und ein systemstatus- bzw. ressourcenüberwachender Agent (Shop-Management-Agent) ergänzt.

Die **Produktagenten (PA)** repräsentieren Produktvarianten und verfügen über alle notwendigen Informationen, die das Produkt betreffen. Dazu gehören sowohl Informationen über deren Zusammensetzung in Form einer aktuellen Stückliste als auch Informationen über die Prozessfolge für die Varianten in den jeweiligen Gewerken. Der PA stellt diese Information anderen Agenten auf Anfrage zur Verfügung.

Der **Shop-Management-Agent (SMA)** ist für die Kontrolle und Überwachung des System- bzw. Ressourcenstatus verantwortlich. Er registriert Ressourcen sobald sie online bzw. offline gehen, z.B. aufgrund einer Störung oder Rekonfiguration der Fertigungslinie. Zudem verfügt er über das Layout und die Verkettung von Anlagen in Form einer Materialflussmatrix und informiert Agenten wie sie von einem Produktionsschritt zum nächsten gelangen können.

Die letzte Ergänzung zu den vier Haupt-Clustern ist ein üblicher **Directory Facilitator (DF)**. Er dient der Implementierung eines Gelbe-Seiten-Services und komplettiert das Agentenmodell. Der DF erfasst und sammelt alle im System verfügbaren Dienstleistungen und vermittelt anfragende Agenten an die Agenten, die die gesuchten Dienstleistungen oder Fähigkeiten zur Verfügung stellen.

Das finale Agentenmodell enthält sieben Agententypen und ist in Tabelle 7-4 zusammengefasst.

Agententyp	Steuerungsaufgaben	Benötigte Steuerungsinformationen
Scheduling-Agent (SA)	<ul style="list-style-type: none"> Produktionssequenz verwalten OAs auf Anforderung von RA oder SMA instanziiieren 	<ul style="list-style-type: none"> Gültiger Produktionsplan (Auftrags-ID, Variante, Reihenfolge, Liefertermin) Ausbringung und unfertige Erzeugnisse Fähigkeiten der anfordernden Ressource Verfügbarkeit von Ressourcen und andere Einschränkungen (z.B. logistische oder technische Restriktionen)
Auftrags-agent (AA)	<ul style="list-style-type: none"> Initialisierung und Überwachung aller erforderlichen Produktionsschritte 	<ul style="list-style-type: none"> Auftragsinformationen (z.B. Auftrags-ID, Variante, Liefertermin, Priorität)
Ressourcen-agent (RA)	<ul style="list-style-type: none"> Vorgesehene Aufträge bearbeiten Verfolgen der Reservierungsliste Auftragsstatus dokumentieren Neue Aufträge bei Nichtbelegung anfordern SMA über Status informieren (verfügbar/gestört) 	<ul style="list-style-type: none"> Ressourcenfähigkeiten (Zykluszeit, Anzahl der Arbeitseinheiten, Varianten etc.) Ressourcenstatus (Zeit bis zum Ausfall, Zeit bis zur Reparatur) Reservierungsliste Bestellstatus
Mediator-Agent (MA)	<ul style="list-style-type: none"> Vermittlungsanfragen von OAs sammeln und Produktionsvorschlägen von RAs während des Reservierungszeitraums einholen Aufträge den Ressourcen zuordnen 	<ul style="list-style-type: none"> Bedarf an Produktionsschritten (Anzahl und Zeitpunkt der Anfragen von OAs) Versorgung der Produktionsschritte (Anzahl und ESOP der RAs) Regeln für die Priorisierung von Aufträgen (z.B. Lieferzeit, Reservierungszeit, Variante, Priorität)
Produktagent (PA)	<ul style="list-style-type: none"> Fertigungsinformationen für jede Produktvariante verwalten Informationen für AAs bereitstellen 	<ul style="list-style-type: none"> Produktvariante Fertigungsschritte (z.B. Schweißpunktfolge im Karosseriebau, Montagereihenfolge in der Montage usw.) Technische/vertriebsseitige Restriktionen
Shop-Management-Agent (SMA)	<ul style="list-style-type: none"> RAs Instanziiieren Verfolgen des Ressourcenstatus (verfügbar/gestört) Bereitstellung von Routing-Informationen für den Materialfluss zu OAs 	<ul style="list-style-type: none"> Hallenlayout und Materialflussgraf Ressourcenstatus (verfügbar/angezeigt) über Nachricht von RA
Directory Facilitator (DF)	<ul style="list-style-type: none"> Aller verfügbaren Dienste im System registrieren und abmelden Informationen für alle Agenten bereitstellen 	<ul style="list-style-type: none"> Registrierte Dienstleistungen und zugehörige Informationen

Tabelle 7-4: Agentenmodell mit zugehörigen Steuerungsaufgaben und -informationen

Mit dem Agentenmodell sind die notwendigen Entitäten des Systems definiert. Damit das Agentensystem seine Funktion erfüllen kann, muss im dritten und letzten Schritt der DACS-Methode die Art der Zusammenarbeit und Kommunikation zwischen den Agenten definiert werden. Dies ist Inhalt des nächsten Abschnitts.

7.2.3 Interaktionsmodell

Im letzten Schritt der DACS-Methode wird das Interaktionsmodell für das Agentensystem erarbeitet. Dafür werden zunächst die notwendigen Interaktionen zwischen den Agenten analysiert, indem die bestehenden Abhängigkeiten im Agentenmodell klassifiziert werden. Im Anschluss findet ein Abgleich mit verfügbaren Interaktionsprotokollen statt und es werden geeignete Protokolle ausgesucht und gegebenenfalls spezifisch angepasst [BJW04].

Klassifizierung der Abhängigkeiten

Einige der identifizierten Steuerungsentscheidungen können nicht isoliert ausgeführt werden, weil sie von anderen Entscheidungen abhängig sind. Wenn Abhängigkeiten Entscheidungsaufgaben enthalten, die verschiedenen Agenten zugeordnet sind, müssen die entsprechenden Agenten interagieren, um Entscheidungsalternativen zu finden, die nicht nur den lokalen, sondern auch den nicht-lokalen Einschränkungen und Präferenzen genügen. Um für die jeweiligen Interaktion die am besten geeigneten Protokolle zu identifizieren, schlägt Bussmann eine Klassifizierungssystematik anhand von neun Kriterien vor [BJW04]. Eine Übersicht der Kriterien und der möglichen Ausprägungsmerkmale findet sich in Anhang 21. Im Folgenden werden zunächst die Interaktionseigenschaften der vier Abhängigkeiten beschreiben und anschließend klassifiziert.

A1. Zur Einhaltung der Liefertreue muss einerseits die Freigabe von Aufträgen optimal gesteuert und andererseits eine Priorisierung von Aufträgen vorgenommen werden, falls mehrere Aufträge um knappe Ressourcen konkurrieren. Da letzteres in der Abhängigkeit A4 behandelt wird, verbleiben für A1 zwei Fälle, in denen Kommunikation notwendig ist. Diese Kommunikationsfälle treten ein, wenn:

- a) eine Ressource aufgrund einer Störung ausfällt oder nach Behebung einer Störung wieder verfügbar wird oder
- b) eine Bandaufgabe-Ressource – also eine Ressource mit der Fähigkeit einen neuen Auftrag physisch zu initiieren – frei und damit eine Auftragsfreigabe notwendig wird.

In beiden Fällen findet zunächst eine bilaterale Kommunikation zwischen einem RA und dem SA statt, um entweder über den Ressourcenstatus zu informieren oder einen neuen Auftrag anzufragen. Im Fall der Auftragsfreigabe erfolgt nach Prüfung der aktuellen Situation in der Fertigung die Auswahl eines Auftrags und die Instanziierung eines AA durch den SA.

A2. Zur Maximierung der Ressourcenauslastung müssen Ressourcen jederzeit mit Aufträgen versorgt werden. Für Bandaufgabe-Ressourcen wird dies über die Abhängigkeit A1 gewährleistet. Für alle anderen Ressourcen informieren die RAs den DF in einer bilateralen Kommunikation darüber, wann sie frei werden. Als Folge werden sie, ebenfalls über die Abhängigkeit A4, durch den MA in den nächsten Zuordnungsprozess einbezogen, für den sie sich qualifizieren. Die Kommunikation erfolgt anschließend bilateral zwischen einem AA und einem RA.

A3. Zur Reduzierung von Engpässen müssen Reihenfolgekombination vermieden werden, die aufgrund der aktuellen Situation in der Produktion Wartezeiten hervorrufen. Die häufigste Art von Engpässen entsteht durch falsche Steuerungsentscheidungen an Ein- und Ausgängen von Ressourcen. Zur Vermeidung von Engpässen bedarf es daher vor allem einer Abstimmung zwischen AAs und RAs, damit die aktuelle Situation in die Entscheidungsfindung einfließt. Dabei müssen sowohl Störungen in nachgelagerten Prozessen als auch die Reihenfolge der Auftragsvarianten in vorgelagerten Prozessen berücksichtigt werden (ein dafür entwickelter Algorithmus wird in 7.3.2 vorgestellt). Da dies eine optimale Zuordnung von Aufträgen zu Ressourcen erfordert, wird

dieser Teil der Abhängigkeit in A4 behandelt. Im Extremfall kann aber auch die Auftragsfreigabe zu Engpässen führen. Nämlich dann, wenn Aufträge aufgelegt werden, die im späteren Verlauf der Produktion nicht gefertigt werden können, weil die benötigten Ressourcen nicht zur Verfügung stehen. Dieses Szenario wird über die Abhängigkeit A1 bei der Auftragsfreigabe berücksichtigt.

A4. In der Zuordnung von Aufträgen zu Ressourcen besteht die komplexeste Aufgabe des Agentensystems. Einerseits, weil die Zielgrößen der involvierten Agenten konfliktär sind: Alle Aufträge möchten so schnell wie möglich bearbeitet werden und alle Ressourcen benötigen schnellstmöglich neue Aufträge. Andererseits, weil die Anzahl der beteiligten Agenten zur Designphase unbekannt ist und sich während der Interaktion verändern kann. Im einfachsten Fall kommuniziert der MA mit einem AA und einem RA. Aufgrund der parallelen Fertigungsabschnitte ist es aber auch möglich, dass zwei Aufträge um eine Ressource werben und umgekehrt. Im komplexesten Fall konkurrieren im betrachteten Anwendungsfall zwei Aufträge um zwei Ressourcen.³³ Um all diese Fälle abzudecken und jederzeit die beste Lösung zu finden, ist eine rein bilaterale Kommunikation nicht ausreichend. Für die systematische Auswahl geeigneter Protokolle erfolgt daher im Folgenden die Klassifizierung der Abhängigkeiten nach den neun Kriterien von Bussmann.

Da die Abhängigkeiten A1, A2 und A3 gleichartige Anforderungen haben, werden sie gemeinsam betrachtet. Die Abhängigkeit A4 unterscheidet sich hiervon erheblich und wird separat untersucht.

Klassifizierung von A1, A2 & A3. In den drei betrachteten Fällen kommunizieren immer jeweils nur zwei Agenten bilateral miteinander. Entweder ein RA mit einem SA (A1) oder ein RA mit einem AA (A2, A3). Die Anzahl *involvierter Agenten* ist daher stets gleich *zwei*. Die Beschränkungen und Präferenzen sind *kompatibel*, da die Anfragen anderenfalls nicht berücksichtigt würden. Es werden ferner nur die individuellen Beschränkungen und Präferenzen berücksichtigt, d.h. sie sind *nicht-lokal*. Demzufolge wird bei den hier betrachteten Interaktionen genau *eine gemeinsame Verpflichtung* eingegangen. Das ist einerseits die Verpflichtung einen durch den SA freigegebenen Auftrag von einem RA zu bearbeiten (A1) und andererseits die Verpflichtung zwischen einem AA und einem RA, dass ein bestimmter Auftrag zu einer bestimmten Zeit durch eine bestimmte Ressource gefertigt werden kann (A2, A3). Spätere Wechsel von Aufträgen oder Ressourcen sind nicht zulässig. Dadurch dass zwei Agenten an der Interaktion beteiligt sind, hat die gemeinsame Verpflichtung jeweils die *Größe zwei*. Da jeder Agent nur eine Verpflichtung eingeht, existiert zwischen den Verpflichtungen *keine Beziehung*. Ferner sind in der Interaktion alle *Rollen fest definiert*, d.h. es existieren im Umkehrschluss keine variablen Rollen. Der Agent, der die Interaktion initiiert erbittet nur eine einzige Handlung von einem anderen Agenten, sodass nur auf Ebene von *Alternativen* interagiert wird. Da die initiierten Agenten alle gültigen Antworten akzeptieren ohne eine Möglichkeit zur Ablehnung zu haben, liegt schließlich eine *partielle Delegation* vor.

³³ Bei Anwendung des Prinzips der Matrix-Produktion ist theoretisch eine größere Anzahl konkurrierender Aufträge und Ressourcen möglich. Dieser Fall wird in der vorliegenden Arbeit aufgrund der Gegebenheiten der Produktionslinien nicht betrachtet. Durch eine Anpassung des Algorithmus wäre es aber auch möglich diesen Fall abzudecken.

Klassifizierung von A4. In der Abhängigkeit A4 ist die Anzahl der involvierten Agenten *variabel*. Im einfachsten Fall ist ein AA, ein RA und der MA beteiligt. Es kann während der Interaktion jedoch auch ein zusätzlicher AA hinzukommen, der auf einer anderen Ressource gefertigt wird und auch zusätzliche Ressourcen könnten verfügbar werden. Falls mehrere Agenten des gleichen Typs an der Interaktion beteiligt sind, kommt es zu *gegenseitlichen Präferenzen*. Zwei AAs möchten jeweils so schnell wie möglich bearbeitet werden, während zwei RAs jeweils so schnell wie möglich mit neuen Aufträgen versorgt werden möchten. Ferner werden nur die individuellen Beschränkungen und Präferenzen berücksichtigt, d.h. sie sind *nicht-lokal*. Die Anzahl der *gemeinsamen Verpflichtungen* ist *variabel*, aber beschränkt. Es werden mindestens eine und maximal zwei gemeinsame Verpflichtung eingegangen, da mindestens ein AA und ein RA und maximal zwei AAs und zwei RAs sich für die Ausführung gemeinsamer Aufgaben verpflichten. Dadurch, dass bei den gemeinsamen Verpflichtungen immer zwei Agenten involviert sind, ist die *Größe* der gemeinsamen Verpflichtungen auf *zwei* festgelegt. Da jeder Agent an maximal einer Verpflichtung beteiligt ist, sind die Beziehungen zwischen den Verpflichtungen *nicht überschneidend*. Zudem kann es sowohl dazu kommen, dass zwei AAs um eine Ressource als auch zwei RAs um einen Auftrag konkurrieren. Daher ist die Beziehung zwischen den Verpflichtungen zudem *unvollständig gedeckt*. Die *Rollen* sind in dieser Interaktion wie auch bei den anderen Abhängigkeiten *fest definiert* und es existieren keine variablen Rollen. Jeder AA und RA teilt dem MA jeweils nur den frühesten Startzeitpunkt für die Bearbeitung und die Variante bzw. Fähigkeiten mit, sodass der MA *nur Alternativen* erhält. Da sowohl die AAs als auch die RAs gültige Lösungen akzeptieren ohne eine Möglichkeit zur Ablehnung zu haben, liegt zudem *partielle Delegation* vor. Die Ausprägungen der beschriebenen Abhängigkeiten sind in Tabelle 7-5 zusammengefasst und stellen die Grundlage für die Auswahl geeigneter Interaktionsprotokolle im nächsten Schritt dar.

Aspekt	Klassifikationskriterium	Ausprägungen A1, A2, A3	Ausprägungen A4
Startsituation	1. Anzahl involvierter Agenten	n=2	variabel
	2. Kompatibilität der Beschränkungen & Präferenzen	kompatible Präferenzen	gegenseitliche Präferenzen
	3. Globale Beschränkungen & Präferenzen	nicht-lokal	nicht-lokal
Gemeinsame Verpflichtungen	4. Anzahl gemeinsamer Verpflichtungen	n=1	variabel
	5. Größe gemeinsamer Verpflichtungen	n=2	n=2
	6. Beziehung zwischen gemeinsamen Verpflichtungen	-	nicht überschneidend, unvollständig gedeckt
Agentenrollen	7. Rollenzuweisung (Anzahl variabler Rollen)	fest (keine)	fest (keine)
Prozessanforderungen	8. Informationsverfügbarkeit	nur Alternativen	nur Alternativen
	9. Delegation von Entscheidungskompetenz	partielle Delegation	partielle Delegation

Tabelle 7-5: Klassifikation der Abhängigkeiten in der Karosseriefolgesteuerung

Auswahl und Anpassung von Interaktionsprotokollen

Zur Auswahl geeigneter Protokolle für die benötigten Interaktionen werden die klassifizierten Abhängigkeiten den Eigenschaften unterschiedlicher Protokolle gegenübergestellt. Dafür wurde zunächst auf Basis von [BJW04] eine Menge von 14 unterschiedlichen Protokollen ausgewählt und entsprechend des oben aufgeführten Klassifikationsschemas bewertet (vgl. Anhang 22 und 23). Anschließend wurden die Ausprägungen der Abhängigkeiten mit denen der Protokolle verglichen und bewertet. Die Gegenüberstellungen sind in Anhang 24 und 25 zu finden.

Interaktionsprotokoll für A1, A2 & A3. Bei der Suche nach passenden Protokollen fällt auf, dass sich mehrere Ansätze für die Abhängigkeiten A1 bis A3 eignen. Dazu gehört z.B. das Plurality Voting Protocol, die Service-oriented Negotiation und das Contract Net Protocol. Dies wird jedoch nur aufgrund der Abwärtskompatibilität erreicht, was bedeutet, dass die Protokolle eine deutlich höhere als die benötigte Komplexität besitzen. Abwärtskompatibilität bedeutet in diesem Fall, dass ein Protokoll grundsätzlich für komplexere Aufgaben ausgelegt ist und daher auch einfachere Aufgaben in einem bestimmten Spektrum erfüllen kann. Wenn ein Protokoll beispielsweise gegensätzliche Präferenzen berücksichtigen kann, kann es auch kompatible Präferenzen oder sogar nur reine Beschränkungen berücksichtigen. Die Anzahl involvierter Agenten ist ebenfalls abwärtskompatibel: Wenn ein Protokoll eine feste, aber nicht zur Designphase bekannte Anzahl von Agenten berücksichtigen kann, kann es auch eine feste, bekannte Anzahl von Agenten berücksichtigen. Selbiges gilt für die Anzahl und Größe gemeinsamer Verpflichtungen. Um zu vermeiden, dass das Agentensystem durch überdimensionierte Protokolle eine unnötig hohe Kommunikationslast bewältigen muss, gilt es daher ein möglichst einfaches Protokoll zu wählen, das die Anforderungen dennoch erfüllt. Aus diesem Grund fällt die Entscheidung auf das einfache aber effektive Requesting Action Protocol (RAP). Das RAP wird üblicherweise verwendet, wenn ein Agent zur Ausführung seiner Aufgabe Unterstützung durch einen anderen Agenten benötigt. Dabei wird angenommen, dass einerseits für den anfragenden Agenten offensichtlich ist, welcher Agent ihn unterstützen sollte und andererseits, dass der unterstützende Agent in den meisten Fällen der Aufforderung nachkommen wird [BJW04]. Der erste Agent versendet daher eine Anfrage an den zweiten mit einer Beschreibung der geforderten Aktion. Der zweite Agent antwortet entweder mit einer Bestätigung und führt die angeforderte Aktion aus oder sendet eine Ablehnung. Für den vorgesehenen Zweck ist das RAP ein passendes Interaktionsprotokoll.

Interaktionsprotokoll für A4. Für die schwierigere Aufgabe der Zuordnung von mehreren Aufträgen zu mehreren Ressourcen wird ein komplexeres Interaktionsprotokoll benötigt. Auch hier kommen mehrere Protokolle in Frage. Ein üblicherweise für die Allokation von Aufgaben zu Knoten in einem Netzwerk eingesetztes Protokoll ist das Contract Net Protocol (CNP, dt. *Kontraktnetzprotokoll*). Für den Fall das mehrere Aufträge um eine Ressource oder umgekehrt konkurrieren, wäre das CNP eine gute Wahl. Da bei der hier betrachteten Interaktion aber auch der Fall eintreten kann, dass mehrere Aufträge auf mehrere Ressourcen verteilt werden müssen,

wären aufwändige Anpassungen des CNP erforderlich. Ein Protokoll, das sich hierfür besser eignet ist die kontinuierliche Doppelauktion (engl. *Continuous Double Auction*, kurz CDA). Die CDA ist ein Auktionsverfahren, bei dem sowohl Käufer als auch Verkäufer Angebote abgeben. Dabei können diese Angebote von sämtlichen Marktteilnehmern eingesehen werden, aber ihre individuellen Präferenzen bleiben stets unbekannt. Durch Hinzufügen eines Mediatoragenten als Mittelsmann für die Sammlung und den Abgleich der jeweiligen Gebote, kann das CDA genutzt werden, um die gewünschten Interaktionen zu erreichen.

Übersicht der Kommunikationsflüsse

Die wesentlichen Kommunikationsprozesse bei der Ressourcenallokation zwischen den Agenten beinhalten neun Schritte und sind in Abbildung 7-11 erfasst. Zur Veranschaulichung der grundsätzlichen Kommunikationsabläufe wird hier – aufgrund seiner allgemein gebräuchlichen Notationsweise – das aus der UML bekannte Kommunikationsdiagramm verwendet.

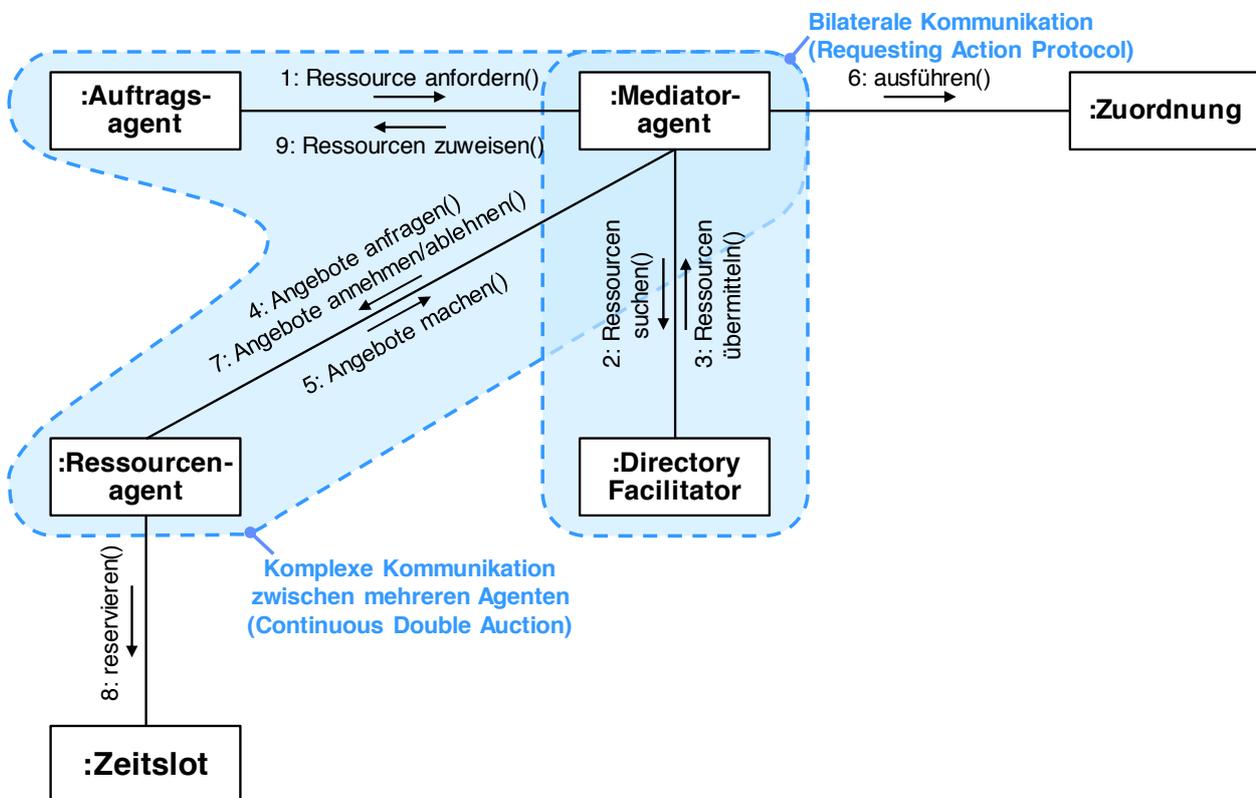


Abbildung 7-11: Kommunikationsdiagramm für die Ressourcenallokation

Die Interaktion beginnt mit einer Ressourcenanforderung durch einen AA an den MA, wodurch das CDA-Protokoll initiiert wird. Der MA benötigt für die Anforderung von Angeboten eine Liste verfügbarer RAs für die Aufgabe. Diese bekommt er vom DF, der eine Datenbank der Fähigkeiten und Verfügbarkeiten der RAs enthält. Da die Interaktion zwischen MA und DF eine einfache bilaterale Kommunikation darstellt, wird hierfür das RAP verwendet. Nachdem der MA die Liste geeigneter RAs erhalten hat, fordert er bei ihnen Angebote an. In der Zwischenzeit kann ein weiterer AA der Auktion beitreten. Wenn alle Angebote eingetroffen sind, nimmt der MA die nach den zugrunde gelegten Optimierungskriterien bestmögliche Zuordnung vor und antwortet mit Ab- oder

Zusagen an die betroffenen RAs. Die Gewinner der Auktion reservieren einen Zeitslot für die Produktion des zugewiesenen Auftrags. Im letzten Schritt der Auktion werden die AAs über den Erfolg oder Misserfolg informiert, sodass sie entweder eine neue Ressource beim MA anfordern oder die Produktion auf dem zugewiesenen RA anstoßen können.

Mit der Erarbeitung des Interaktionsmodells ist die Konzeptionierung des Agentenmodells nach der DACS-Methode abgeschlossen. Im nächsten Schritt wird das entwickelte Modell als Computerprogramm unter Verwendung einer Agentenplattform programmiert und die Validität des Modells überprüft. Im Folgenden erfolgt die Beschreibung der dabei eingesetzten Konzepte und Algorithmen.

7.3 Programmierung des Agentensystems

Als letzter Schritt vor der Simulation des betrachteten Anwendungsfalls muss das im vorherigen Abschnitt konzipierte Agentensystem programmiert werden. In Abschnitt 6.3.1 wurde dafür das JADE-Framework als Agentenplattform festgelegt. Im Folgenden werden die Umsetzungsdetails und die dabei verwendeten Algorithmen beschrieben. Dabei wird auf die drei Bereiche mit dem größten Einfluss auf das Agentensystem eingegangen. Im Abschnitt 7.3.1 werden zunächst die Algorithmen für die Auftragsfreigabe diskutiert. Daran anknüpfend setzt sich Abschnitt 7.3.2 mit den detaillierten Allokations- und Reservierungsmechanismen von Ressourcen auseinander. In Abschnitt 7.3.3 wird schließlich die Initialisierung und Überwachung der Produktionsaufträge behandelt.

7.3.1 Auftragsfreigabe

Im betrachteten Agentensystem obliegt die Auftragsfreigabe dem Scheduling-Agenten (SA). Damit der SA diese Aufgabe bestmöglich erfüllen kann, wurden zwei Komponenten implementiert. Einerseits ein Mechanismus zum Initiieren des Auftragsfreigabeprozesses und andererseits ein Verfahren zur Anpassung der Produktionssequenz bei unvorhergesehenen Störungen. Die Funktionsweise der beiden Komponenten wird im Folgenden diskutiert.

Initiierung des Auftragsfreigabeprozesses

Der Scheduling-Agent setzt auf der Produktionssequenz auf, die in einem vorangehenden Schritt von einem Sequenzierungsalgorithmus ermittelt wird. In den meisten Fällen folgen die Algorithmen dabei den in Abschnitt 7.1.3 beschriebenen Scheduling-Verfahren. Die resultierende Produktionssequenz beinhaltet Informationen zum Auftrag (z.B. Auftrags-ID, Produktvariante, Liefertermin) und wird nur geändert, falls Störungen auftreten. Zur Vermeidung von Verschwendung durch ungleichmäßige und nicht nach dem Kundentakt ausgerichtete Produktion wurde – in Analogie an die Prinzipien des Lean Managements – ein Mechanismus implementiert, der den Auftragsfreigabeprozess nach dem Pull-Prinzip steuert. D.h. die Auftragsfreigabe wird nur angestoßen, wenn eine Bandaufgabe-Ressource einen neuen Auftrag vom SA anfordert. Dies erfolgt immer unmittelbar nach Beendigung eines vorausgehenden Auftrags durch eine Ressource, sodass

die als nächstes freiwerdende Anlage sofort mit einem Auftrag versorgt wird. Dieses Vorgehen hat gegenüber einem Push-Verfahren den Vorteil, dass die zu dem tatsächlichen Zeitpunkt der physischen Bandaufgabe bestehenden Umweltbedingungen in die Auftragsauswahl einbezogen werden. Nach der erfolgten Auswahl eines passenden Auftrags instanziiert der SA einen entsprechend parametrisierten Auftragsagenten. Das dahinterliegende Prinzip ist in Abbildung 7-12 skizziert.

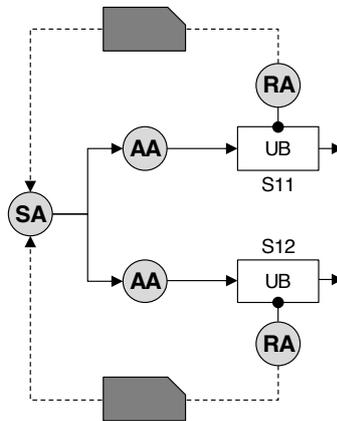


Abbildung 7-12: Pull-Prinzip zur Initialisierung des Auftragsfreigabeprozesses

Da der vorliegende Anwendungsfall über zwei Fertigungslinien verfügt, existieren folglich zwei Ressourcen, über die eine Bandaufgabe erfolgen kann. Durch die Anfrage eines RA wird faktisch ein digitaler Kanban ausgelöst, der einen dezentralen Regelkreis zwischen RA und SA konstituiert. Der Mechanismus lässt auch Spielraum für eine zukünftige Rekonfiguration der Produktion, z.B. wenn zusätzliche Fertigungslinien hinzugefügt werden sollen. Eine neue Linie könnte ohne weitere Anpassungen des Freigabeprozesses in das Produktionssystem integriert werden.

Erfolgt eine derartige Anfrage eines RA muss der SA im Normalfall lediglich auf die vorhandene Produktionssequenz zurückgreifen und Aufträge der Reihenfolge nach freigeben. Häufig kommt es im Produktionsablauf jedoch zu nicht vorhergesehenen Störungen, z.B. durch Anlagenausfälle, Fehlteile, Qualitätsprobleme, Personalmangel etc. In solchen Störungsfällen werden Anpassungen der Produktionssequenz notwendig, um Liefertreue und Auslastungsmaximierung zu gewährleisten. In herkömmlichen Produktionssystemen erfolgt eine solche Anpassung oft durch Mitarbeiter des Leitstandes auf Basis ihrer Erfahrungen. Aufgrund der zunehmenden Komplexität der Fertigung ist es allerdings sinnvoll solche Entscheidungen datenbasiert durch das Steuerungssystem treffen zu lassen (vgl. Abschnitt 7.1). Daher wurde im SA ein Mechanismus implementiert, mit dem sich die Auftragsfreigabe an die aktuelle Situation in der Produktion anpasst.

Adaptive Auftragsfreigabe

Aufgrund der fertigungstechnischen Unterschiede der zahlreichen Produktvarianten, die in heutigen Fabriken gefertigt werden, unterliegen Produktionssysteme zahlreichen Restriktionen hinsichtlich des Produktmix, die für einen wirtschaftlichen Fabrikbetrieb stets eingehalten werden

müssen. Störungen führen häufig jedoch zu schwer abschätzbaren Verwerfungen in nachfolgenden Produktionsschritten wie der Lackiererei und vor allem der Montage. Ziel ist es daher die Auswirkungen von Störungen auf den Produktmix zu kompensieren, damit das entstehende Ungleichgewicht weniger Probleme verursacht und leichter abgebaut werden kann. Um die Auftragsfreigabe zu nivellieren muss der SA auf den vorhandenen Produktionsplan zurückgreifen, Echtzeitinformationen anderer Agenten mit einbeziehen und Aufträge entsprechend ihrer Variante priorisieren. Die entwickelte adaptive Auftragsfreigabe besteht daher einerseits aus einem Adaptionsalgorithmus, der auf Basis der aktuellen Situation den besten Auftrag auswählt und andererseits aus einem Mechanismus, der den SA über die Art und Dauer von Störungen informiert.

Adaptionsalgorithmus. Der SA überprüft bei jeder Anfrage, ob Störungen vorliegen, die bei der Auftragsfreigabe berücksichtigt werden müssen. Ist dies der Fall, schaltet er in den Störungsmodus und wählt den als nächstes freizugebenden Auftrag aus der Produktionssequenz auf Basis der verfügbaren Informationen aus. Im vorliegenden Anwendungsfall sind die beiden Fertigungslinien durch Restriktionen hinsichtlich des maximal produzierbaren Verhältnisses von Produktvarianten eingeschränkt (vgl. Tabelle 7-6). Es gilt daher stets einen Produktmix der Varianten A, B und C im Verhältnis von 6:3:1 zu erreichen.

	Fertigungslinie 1			Fertigungslinie 2		
Variante	A	B	C	A	B	C
Verhältnis	60%	40%	-	80%	-	20%
Anzahl/Tag	600 Stk.	400 Stk.	-	800 Stk.	-	200 Stk.
	↓			↓		
	Fertigungslinie 1 & 2					
Variante	A		B		C	
Verhältnis	60%		30%		10%	
Anzahl/Tag	1.400 Stk.		400 Stk.		200 Stk.	

Tabelle 7-6: Eigenschaften der Fertigungslinien 1 und 2

Wenn der Ist-Produktmix aus dem Gleichgewicht gerät, muss in Abhängigkeit der aktuellen Situation in der Produktion entschieden werden welche Variante als nächstes aufgelegt werden sollte. Da die Fertigungslinien unterschiedliche Fähigkeiten haben, kann nicht jede Variante freigegeben werden. Daher werden vor der Freigabe die Auswirkungen der jeweiligen verfügbaren Varianten auf den Produktmix bewertet. Das Beispiel in Abbildung 7-13 zeigt eine Situation, in der es durch eine Störung in der Ressource R1 zu einer Abweichung gekommen ist.

Fzg.-Nr.	Ressource	Variante	Freigegebene Aufträge			nächste freizugebende Variante									
			Soll (rel.)	Ist		Variante A			Variante B			Variante C			ind. Delta (rel.)
				lst (abs.)	lst (rel.)	lst (abs.)	lst (rel.)	Delta (rel.)	lst (abs.)	lst (rel.)	Delta (rel.)	lst (abs.)	lst (rel.)	Delta (rel.)	
16	R1	A	60,00%	15	75,00%	16	76,19%	16,19%	15	71,43%	11,43%	X			
		B	30,00%	3	15,00%	3	14,29%	-15,71%	4	19,05%	-10,95%				
		C	10,00%	2	10,00%	2	9,52%	-0,48%	2	9,52%	-0,48%				
		Summe	100,00%	20	100,00%	21	100,00%	32,38%	21	100,00%	22,86%				
17	R2	A	60,00%	15	71,43%	16	72,73%	12,73%	X			15	68,18%	8,18%	17,50%
		B	30,00%	4	19,05%	4	18,18%	-11,82%				4	18,18%	-11,82%	26,67%
		C	10,00%	2	9,52%	2	9,09%	-0,91%				3	13,64%	3,64%	
		Summe	100,00%	21	100,00%	22	100,00%	25,45%				22	100,00%	23,64%	
18	R1	A	60,00%	15	68,18%	16	69,57%	9,57%	15	65,22%	5,22%	X			
		B	30,00%	4	18,18%	4	17,39%	-12,61%	5	21,74%	-8,26%				
		C	10,00%	3	13,64%	3	13,04%	3,04%	3	13,04%	3,04%				
		Summe	100,00%	22	100,00%	23	100,00%	25,22%	23	100,00%	16,52%				
19	R2	A	60,00%	15	65,22%	16	66,67%	6,67%	X			15	62,50%	2,50%	10,00%
		B	30,00%	5	21,74%	5	20,83%	-9,17%				5	20,83%	-9,17%	40,00%
		C	10,00%	3	13,04%	3	12,50%	2,50%				4	16,67%	6,67%	
		Summe	100,00%	23	100,00%	24	100,00%	18,33%				24	100,00%	18,33%	
20	R1	A	60,00%	16	66,67%	17	68,00%	8,00%	16	64,00%	4,00%	X			
		B	30,00%	5	20,83%	5	20,00%	-10,00%	6	24,00%	-6,00%				
		C	10,00%	3	12,50%	3	12,00%	2,00%	3	12,00%	2,00%				
		Summe	100,00%	24	100,00%	25	100,00%	20,00%	25	100,00%	12,00%				
21	R2	A	60,00%	16	64,00%	17	65,38%	5,38%	X			16	61,54%	1,54%	8,24%
		B	30,00%	6	24,00%	6	23,08%	-6,92%				6	23,08%	-6,92%	35,00%
		C	10,00%	3	12,00%	3	11,54%	1,54%				4	15,38%	5,38%	
		Summe	100,00%	25	100,00%	26	100,00%	13,85%				26	100,00%	13,85%	

Abbildung 7-13: Beispielhafte Anwendung des Adaptionalgorithmus

Der freigegebene Produktmix hat das Verhältnis 15:3:2 und die Ressource R1, die die Varianten A und B fertigen kann, fordert einen neuen Auftrag an. Der Algorithmus vergleicht welche der beiden Varianten die Abweichung zum Soll-Produktmix minimiert, indem er die theoretischen ex post Szenarios berechnet. Dafür ermittelt er die Abweichungen für jede Variante und summiert ihre Beträge auf. Da die Variante B zu einer geringeren Gesamtabweichung führt (22,86% ggü. 32,38%) wird sie durch den SA bevorzugt und daher freigegeben. Falls es zu einer Pattsituation wie bei Fahrzeug Nr. 19 kommt, entscheidet die individuelle Abweichung über die aufzulegende Variante (10% ggü. 40%). Falls auch dies keinen eindeutigen Sieger hervorbringt, entscheidet schließlich der Liefertermin des Auftrags. Der Algorithmus wirkt sich sowohl auf die Maximierung der Ressourcenauslastung als auch auf die Optimierung der Liefertreue vorteilhaft aus, da keine Aufträge freigegeben werden, die nicht produziert werden können oder zu Verzögerungen in der Fertigung führen würden. Der programmierte Adaptionalgorithmus ist im Detail in Anhang 26 abgebildet.

Wenn es zu Störungen kommt, reicht die theoretische Betrachtung des Produktmix alleine nicht mehr aus. Wenn z.B. die Ressource S22 für längere Zeit ausfällt, kann die Variante C nicht mehr gebaut werden. Es würde an der Stelle zu einem Materialstau kommen, der die gesamte Fertigungslinie blockiert. Damit der Adaptionalgorithmus seine Funktion bestmöglich erfüllen kann, muss bei Störungen daher entschieden werden, ob alle Varianten bei der Auftragsfreigabe angeboten werden sollten. Dafür müssen die folgenden Störungsfälle berücksichtigt werden.

Sperrung von Varianten im Störungsfall. Es können grundsätzlich drei Störungsfälle unterschieden werden, die für den SA relevant sind (vgl. Abbildung 7-14).

Fall	Beispiel	Strategie
<p>1. Fall: Interne Störung mit</p> <p>DLZ bis Störung > progn. Störungsdauer</p>	<p>Anlagenstörung mit</p> <ul style="list-style-type: none"> - progn. Dauer = 1 Std. - DLZ bis Störung < 1 Std. <p>DLZ > 1h</p> <p>Bandauflage Störung</p>	<p>► Keine Anpassung der Bandauflage</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> $t_i^{Sperrung} = 0$ </div>
<p>2. Fall: Interne Störung mit</p> <p>DLZ bis Störung ≤ progn. Störungsdauer</p>	<p>Anlagenstörung mit</p> <ul style="list-style-type: none"> - progn. Dauer = 1 Std. - DLZ bis Störung < 1 Std. <p>DLZ < 1h</p> <p>Bandauflage Störung</p>	<p>► Anpassung der Bandauflage für $t_i^{Sperrung}$ mit:</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> $t_i^{Sperrung} = t_i^{Störung} - t_i^{BA,Störung}$ <p style="text-align: center;">bzw.</p> $t_i^{Sperrung} = (1 + s) \cdot (t_i^{Störung} - t_i^{BA,Störung})$ </div>
<p>3. Fall: Externe Störung</p>	<p>Fehlmaterial aufgrund eines verspäteten LKW mit</p> <ul style="list-style-type: none"> - progn. Dauer = 1 Std. - DLZ bis Störung = 0 Std. <p>Bandauflage Störung</p>	<p>► Anpassung der Bandauflage für $t_i^{Sperrung}$ mit:</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> $t_i^{Sperrung} = t_i^{Störung}$ </div>

Abbildung 7-14: Störungsfälle und Umgangsstrategien

Im **ersten Störungsfall** ist die Durchlaufzeit eines neu aufgelegten Auftrags bis zum Störungsort größer als die prognostizierte Dauer der Störung. In diesem Fall greift der SA nicht ein, da die Störung keine Auswirkungen auf den Produktmix der Karossen im System, sondern lediglich auf den Output-Mix hat. Grund ist die zeitliche Trennung von Bandaufgabe und Störung, denn ein zum Zeitpunkt $t = 0$ aufgelegter Auftrag kommt erst an der Ressource an, wenn die Störung bereits wieder behoben ist. Die negative Auswirkung der Störung auf den vorübergehenden Output-Mix des Karosseriebaus kann steuerungstechnisch nicht beeinflusst werden und muss im nachgelagerten Sortierpuffer ausgeglichen werden.

Im **zweiten Störungsfall** dauert die Störung länger als die Durchlaufzeit von Bandaufgabe bis zum Störungsort. Wenn durch die Störung z.B. die Variante B an der gestörten Ressource nicht mehr bearbeitet werden kann, ist ein Eingreifen des SA erforderlich. Als Strategie wird die Variante B in diesem Beispiel so lange gesperrt bis sichergestellt werden kann, dass ein neu aufgelegter Auftrag die Anlage erst erreicht, wenn diese bereits wieder funktionstüchtig ist. Die Sperrzeit der Variante ergibt sich aus der Differenz der Störungsdauer und der Entfernung bis zur gestörten Ressource. Da die Prognose der Störungsdauer häufig von Mitarbeitern der Instandhaltung auf Basis individueller Erfahrungen erstellt wird, kann es hier zu unerwünschten Abweichungen kommen. Um Ungenauigkeiten in der Störungsprognose zu kompensieren, kann die Sperrzeit um einen Sicherheitsfaktor s ergänzt werden.

Im **dritten Störungsfall** wird die Störung im Gegensatz zu den anderen Fällen nicht von innen, sondern von außen induziert. Ein häufiges Beispiel für eine solche Störung sind temporäre Lieferengpässe von kritischen Komponenten wie Motoren und Getrieben. Da eine solche Störung

unmittelbare Auswirkungen auf die Auftragsfreigabe hat, wird eine daraus resultierende Sperrung sofort wirksam und ihre Dauer entspricht der Störungsdauer.

Um die beschriebenen drei Störungsfälle in geeigneter Weise zu berücksichtigen, wird der SA von den RAs mit Echtzeitinformationen zu ihrem Status versorgt. Fällt eine Ressource aus oder wird sie wieder verfügbar, erhält der SA eine Nachricht und entscheidet, ob und wie die Information für ihn relevant ist. Ein Zustandsdiagramm des SA ist in Anhang 27 zu finden. Die dem SA zugrundeliegende Heuristik ist bewusst einfach gehalten und birgt daher Potenzial für Verbesserungen. Insbesondere durch die Anwendung von Machine Learning-Techniken wie Reinforcement Learning könnten die in der Fertigung anfallenden Daten nutzbar gemacht werden, um die Entscheidungsqualität zu erhöhen. Daher wurde der SA unabhängig vom verwendeten Algorithmus programmiert, sodass der Algorithmus jederzeit durch einen anderen Ansatz ersetzt werden kann.

7.3.2 Allokation und Reservierung von Ressourcen

Für die Implementierung der erforderlichen Ressourcenzuweisung wurde das in Abschnitt 7.2.3 ausgewählte Continuous Double Auction-Protokoll den Anforderungen des Anwendungsfalls angepasst und um einen Reservierungsmechanismus ergänzt. Dadurch werden die erforderlichen Verhandlungen zwischen den AAs und RAs ausgeführt ohne einen Leerlauf in der Produktion zu verursachen. Wie das Sequenzdiagramm in Abbildung 7-15 zeigt, spielt der MA eine zentrale Rolle bei der Abwicklung der Verhandlung. Er koordiniert den Verhandlungsprozess als Broker und wird durch eine Mediationsanfrage eines Auftragsagenten ausgelöst.

Nach Erhalt der Anfrage sucht der MA im Gelbe-Seiten-Verzeichnis des DF nach Ressourcen, die die Anforderungen des Auftrags erfüllen. Nach dem Empfang dieser Informationen startet der MA eine Ausschreibung, indem er von den verfügbaren Ressourcen Angebote für die Produktion des Auftrags anfordert (engl. *Call for Proposition*, kurz CFP). Die RAs antworten mit einem Angebot, das den geschätzten Zeitpunkt des Produktionsstarts (engl. *Estimated Start Of Production*, kurz ESOP) enthält und warten auf eine Rückmeldung des MA. Während dieser Zeit kann ein weiterer AA der Verhandlung beitreten, sodass im vorliegenden Anwendungsfall – aufgrund der parallelen Fertigungslinien – maximal zwei Aufträge um zwei Ressourcen konkurrieren können. Um den Verlust von produktiver Zeit während der Verhandlung zu vermeiden, wurde ein Reservierungsmechanismus implementiert, der bereits vor Abschluss des aktuellen Fertigungsschritts beginnt.

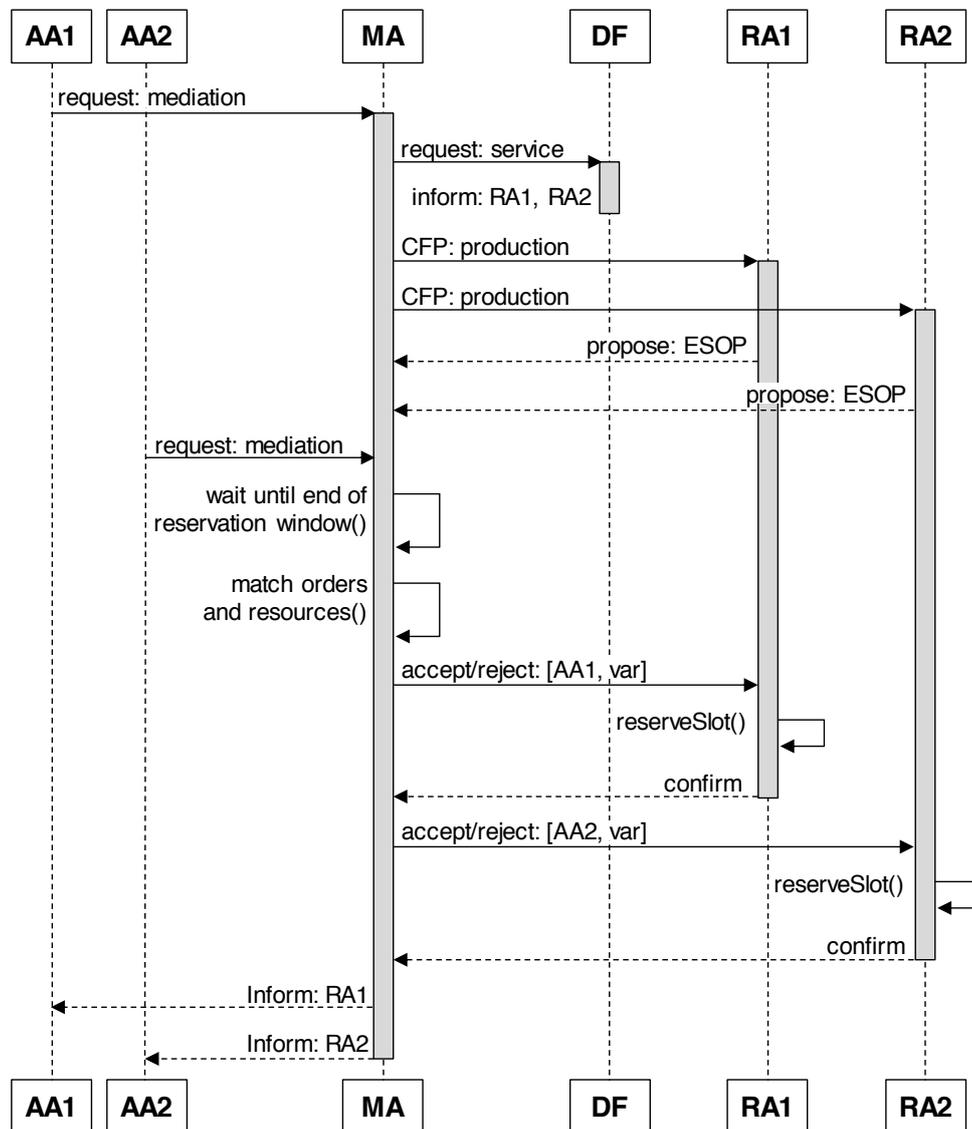


Abbildung 7-15: Agentenkommunikation zur Ressourcenallokation in der KFS

In dem in Abbildung 7-16 dargestellten Beispiel stößt der Auftrag A1 eine Verhandlung für die nachfolgenden Ressourcen R3 und R4 bereits an, während er noch auf der Ressource R1 bearbeitet wird. Der Reservierungsprozess wird ausgelöst, wenn die Zeit bis zur Fertigstellung des Auftrags auf der aktuellen Ressource einen definierten Schwellwert unterschreitet. In Anlehnung an die Arbeiten [HWG12] und [AdM12] wurde in den Prozess ein Reservierungszeitfenster implementiert. Dieses startet mit dem Zeitpunkt, zu dem der AA eine Anfrage an den MA sendet und hat eine feste Dauer. Durch das Platzieren des Reservierungsprozesses an das Ende eines Bearbeitungsschrittes wird zudem die Wahrscheinlichkeit, dass Störungen während des laufenden Produktionsprozesses auftreten gering gehalten.

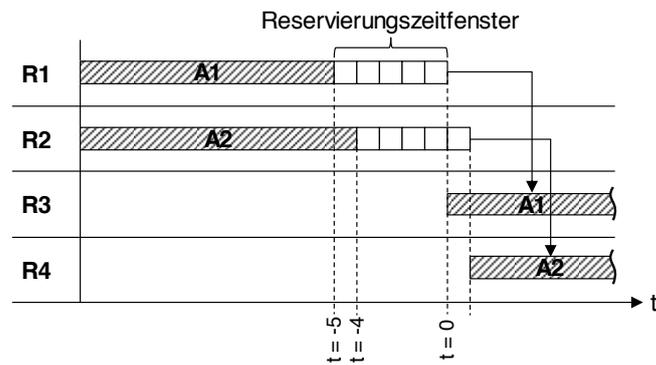


Abbildung 7-16: Reservierungsmechanismus

Nach Ablauf des Reservierungszeitfensters können keine weiteren Aufträge und Ressourcen berücksichtigt werden, sodass der MA die Zuordnung der Aufträge zu Ressourcen auf Basis eines Algorithmus ermittelt. Bei der Zuordnung wird das Ziel verfolgt die Liefertreue der Aufträge und die Auslastung der Ressourcen zu maximieren sowie das Entstehen von Engpässen zu vermeiden. Dabei können im vorliegenden Anwendungsfall drei Szenarien unterschieden werden:

1. Wenn zwei ansonsten gleich geeignete Aufträge um eine Ressource oder die frühestmögliche Bearbeitung auf mehreren Ressourcen konkurrieren, erhält derjenige Auftrag den frühesten Zuschlag, der am wenigsten Zeit bis zum vorgesehenen Liefertermin hat.
2. Konkurrieren umgekehrt mehrere Ressourcen um einen oder mehrere Aufträge, für die sie gleich geeignet sind, dann erhält diejenige Ressource den ersten Zuschlag, die am frühesten bereitsteht. Auf diese Weise soll der Leerlauf von Ressourcen minimiert werden.
3. Wenn durch die ersten beiden Regeln ein Engpass verursacht werden sollte, wird die Zuordnung so angepasst, dass kein Engpass entsteht oder die Auswirkungen von Engpässen minimiert werden.

Der aus Engpässen entstehende Schaden kann den Nutzen der kurzfristigen Optimierung der Liefertreue oder Ressourcenauslastung um ein Vielfaches übersteigen. Daher werden die Zielgrößen Liefertreue und Auslastung hinter die Anforderungen der Engpassvermeidung zurückgestellt. Dabei können die zwei in Abbildung 7-17 dargestellten Fälle unterschieden werden.

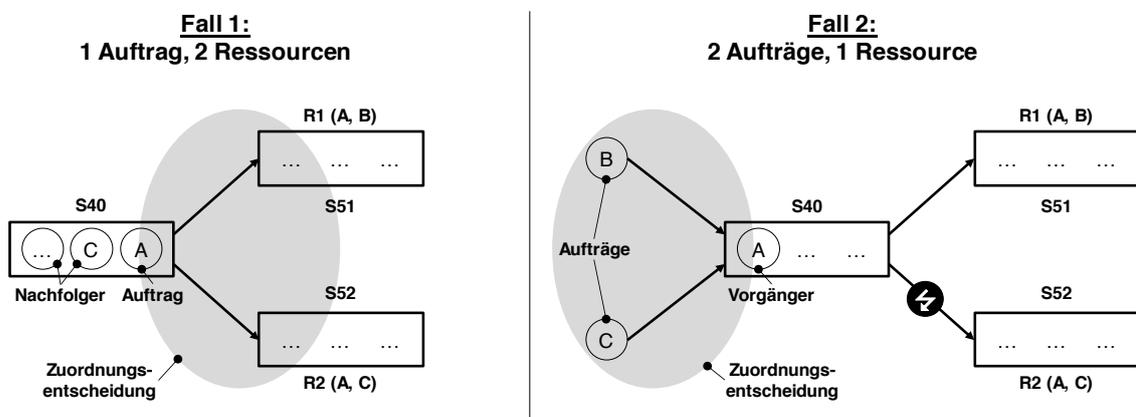


Abbildung 7-17: Engpassvermeidung durch Berücksichtigung der Auftragsreihenfolge

Im ersten Fall konkurrieren zwei Ressourcen um einen Auftrag. Dabei kann es aufgrund der unterschiedlichen Bearbeitungsgeschwindigkeiten von Ressourcen in Fabriken mit parallelen Fertigungslinien zu Engpässen kommen. Damit die Ressource S40 die nachfolgenden Ressourcen S51 und S52 kontinuierlich mit Aufträgen versorgen kann, muss sie doppelt so schnell wie sie arbeiten. Da die Ressourcen S51 und S52 beide die Variante A fertigen können, könnte es dazu kommen, dass z.B. S52 einen Auftrag der Variante A zugewiesen bekommt. Falls der darauffolgende Auftrag jedoch von der Variante C ist, führt diese Zuordnung zu einem Engpass, denn S51 kann die Variante C nicht bearbeiten. Damit vermieden wird, dass es zu unnötigen Staus kommt, müssen solche Konstellationen vermieden bzw. bei der Zuordnung beachtet werden.

Auch im zweiten Fall ist die Auftragsreihenfolge ein wesentlicher Stellhebel zur Engpassvermeidung. Wenn wie im obigen Beispiel illustriert die Ressource S52 ausfällt, kann in diesem Produktionsschritt die Variante C von keiner anderen Ressource bearbeitet werden. Wenn S40 einen Auftrag dieser Variante zugeordnet bekommen würde, käme es zu einer unnötigen Blockade und S40 könnte so lange keine weiteren Aufträge annehmen bis die Ressource S52 wieder verfügbar ist. Folglich wurde der Zuordnungsalgorithmus so programmiert, dass S40 in einem solchen Fall nur Aufträge der Varianten A und B annimmt. Damit wird die Entstehung von Engpässen vermieden, was sich wiederum positiv auf Liefertreue und Ressourcenauslastung auswirkt.

Die beiden Fälle wurden als einfache Heuristik in den Zuordnungsalgorithmus implementiert. Die Heuristik berücksichtigt in Abhängigkeit des zutreffenden Falls die Vorgänger und Nachfolger der Aufträge derart, dass Engpässe vermieden werden (vgl. Anhang 28). Durch Anpassung des Zuordnungsalgorithmus könnten auch komplexere Anwendungsfälle mit dem programmierten Agentensystem behandelt werden. Im Extremfall könnte auch eine Matrix-Produktion realisiert werden, bei der die starre Verkettung von Produktionsanlagen aufgebrochen wird [Gre16].

In diesem Zusammenhang existieren auch andere vielversprechende Ansätze, die für einen solches Szenario ebenfalls potenzielle Lösungen bereitstellen. In einer hervorzuhebenden Forschungsarbeit von Ivanov et al. wurde ein dynamisches Modell und ein Algorithmus für das kurzfristige Supply Chain Scheduling in der Smart Factory Industrie 4.0 entwickelt [IDS16]. Darin wird eine Vernetzung zwischen dem Scheduling und der Fertigungssteuerung angestrebt und eine Dekomposition des Optimierungsproblems derart vorgenommen, dass viele einfacher zu lösende lokale Probleme entstehen. Die in diesem Kapitel beschriebene Agententechnologie würde sich potenziell auch für die Implementierung dieses Ansatzes in eine reale Fabrikanwendung eignen.

Im Anschluss an die nach den obigen Schritten erfolgte Zuordnung von Aufträgen zu Ressourcen werden die Teilnehmer der Verhandlung über das Ergebnis der Mediation informiert. Falls ein Auftrag innerhalb des Reservierungszeitfensters nicht zugewiesen werden kann, wird er in der nächsten Verhandlung bevorzugt behandelt. Der dafür entwickelte Algorithmus ist in Anhang 29 abgebildet. Da nun alle Agenten über die notwendigen Informationen verfügen, kann im nächsten Schritt die Bearbeitung der Aufträge angestoßen werden.

7.3.3 Initiierung und Überwachung von Produktionsaufträgen

Alle AAs erhalten entweder bei der Auftragsfreigabe durch den SA oder bei der Ressourcenallokation durch den MA eine Ressource zugewiesen. Die Aufgabe der AAs ist es nun den tatsächlichen Produktionsschritt anzustoßen und die Ausführung zu überwachen. Dies geschieht nach dem in Abbildung 7-18 dargestellten Verfahren auf Basis des RAP.

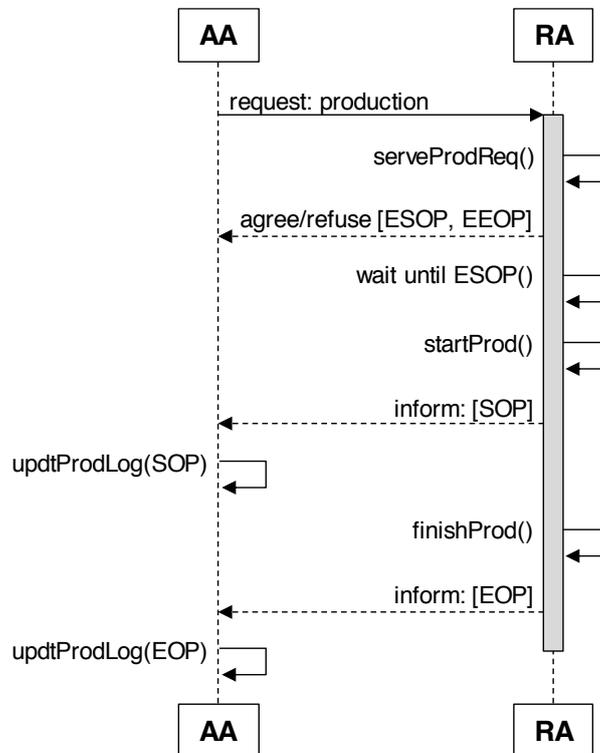


Abbildung 7-18: Sequenzdiagramm Initiierung und Überwachung der Produktion

Nachdem der AA die Informationen zur designierten Ressource erhalten hat, stellt er eine Produktionsanfrage an besagten RA. Dieser wurde vorab bereits vom MA über den Auftrag informiert und hat dafür einen Produktionsplatz in seiner intern geführten Reservierungsliste reserviert. Jeder Auftrag durchläuft in Analogie zu [HWG12] vier für die Reservierung notwendige Phasen. Die korrespondierenden Status der Reservierungsphasen lauten:

0. in Anfrage (engl. *call for proposition*)
1. reserviert (engl. *reserved*)
2. in Bearbeitung (engl. *processing*)
3. fertiggestellt (engl. *terminated*).

Falls es in der Zwischenzeit keine Störungen gibt, nimmt der RA die Anfrage an und sendet dem AA ggf. eine Aktualisierung des voraussichtlichen Start- (ESOP) und Endzeitpunktes der Produktion (engl. *Estimated End of Production*, kurz EEOP). Sobald der Produktionsschritt startet informiert der RA den AA über den tatsächlichen Produktionsstart (SOP). Dieser dokumentiert den Zeitpunkt in seiner ebenfalls intern geführten Datenbank. Kurz vor Erreichen des EEOP überprüft der AA, ob dies der letzte notwendige Fertigungsschritt ist. Ist das nicht der Fall, stellt er entsprechend dem im vorangehenden Abschnitt beschriebenen Prozess eine Mediationsanfrage für den

gewünschten nächsten Produktionsschritt. Nach Fertigstellung des Schrittes informiert der RA den AA über den tatsächlichen Zeitpunkt, zu dem die Bearbeitung abgeschlossen wurde (EOP). Auch diese Information wird vom AA in seiner Datenbank dokumentiert. Wenn weitere Produktionsschritte notwendig sind, sollte dem AA zu diesem Zeitpunkt bereits eine neue Ressource zugewiesen worden sein, sodass sich das hier beschriebene Verfahren wiederholt. Die entsprechenden Ablaufdiagramme des AA und des RA sind in Anhang 30 und 31 angehängt.

Die in diesem Abschnitt vorgestellten Konzepte und Algorithmen wurden zur Realisierung des Agentensystems programmiert und mithilfe einfacher Anwendungsfälle validiert. Auf dieser Basis wurde im Anschluss eine Reihe von Simulationen durchgeführt, um die Leistungsfähigkeit des Agentensystems zu untersuchen. Die Simulationsexperimente und die aus ihnen resultierenden Ergebnisse werden im folgenden Abschnitt diskutiert.

7.4 Simulationsexperimente

Vor der Durchführung der Simulationsexperimente wurde analog des Vorgehens nach Law zunächst das Design der Simulationsexperimente festgelegt. In Abschnitt 7.4.1 erfolgt die Vorstellung dieses Designs, das die zugrunde gelegten Annahmen, die zur Leistungsmessung angesetzten Kennzahlen sowie die verwendeten Parameter beinhaltet. Im nachfolgenden Abschnitt 7.4.2 findet schließlich die Analyse und Diskussion der Simulationsergebnisse statt.

7.4.1 Design der Simulationsexperimente

Zur Beantwortung der dritten Forschungsfrage gilt es zu klären wie agentenbasierte Steuerungssysteme die Leistungsfähigkeit in komplexen Produktionssystemen wie dem hier betrachteten Anwendungsfall verbessern können. Dies soll über einen Vergleich der Leistung von konventionellen und agentenbasierten Systemen geschehen. Das Design der Simulationsexperimente sieht daher Simulationsläufe eines konventionellen und des in dieser Arbeit entwickelten agentenbasierten Steuerungssystems vor. Das konventionelle Steuerungssystem unterscheidet sich vom agentenbasierten dadurch, dass es weder die Funktionalitäten der in Abschnitt 7.3.1 beschriebenen adaptiven Auftragsfreigabe, noch die der in Abschnitt 7.3.2 vorgestellten situationsabhängigen Ressourcenallokation besitzt. Das bedeutet konkret, dass die Bandaufgabe – unabhängig von eintretenden Störungen – nach der vorab ermittelten Auftragssequenz erfolgt und dass Aufträge entweder der Fertigungslinie 1 oder 2 fest zugewiesen werden. Ein Wechsel zwischen den Fertigungslinien ist nicht möglich. Mit Ausnahme dieser Unterschiede unterliegen beide Modelle denselben im Folgenden beschriebenen Annahmen und Parametern.

Vereinfachende Annahmen für die Modellierung

Das entwickelte Agentensystem ist ein Modell, das ein typisches Szenario einer realen Automobilfertigung repräsentiert. Wie jedes Modell stellt es eine Vereinfachung der Realität dar. Um die Komplexität des Modells dem Rahmen dieser Arbeit anzupassen, wurden für die Simulation vereinfachende Annahmen getroffen. Dank seiner modularen Bauweise kann das Modell bei Bedarf

einfach erweitert werden, indem man die Softwaremodule und/oder Algorithmen anpasst bzw. austauscht. Es wurden folgende Annahmen getroffen:

Zykluszeiten. In der Automobilproduktion können die Zykluszeiten von Bearbeitungsschritten aufgrund unterschiedlicher Produktkonfigurationen abweichen. Da dies zu einer Über- oder Unterlastung von Fertigungseinrichtungen führen kann, stellen variable Zykluszeiten eine Herausforderung für die Produktionsablaufplanung dar [SKP11]. Aufgrund des begrenzten Umfangs dieser Arbeit werden im betrachteten Anwendungsfall Zykluszeiten als konstant angenommen. D.h. die Bearbeitungsdauer für Produkte aller Varianten ist immer identisch. Falls erwünscht, können variable Zykluszeiten nachträglich implementiert werden. Bestehende Sequenzierungsalgorithmen, die dies unterstützen sind in [SaP01] und [SCN08] beschrieben.

Anzahl Varianten. In modernen Fabriken werden häufig dutzende Fahrzeugvarianten parallel gefertigt. Die Varianten unterscheiden sich teilweise durch gravierende strukturelle Eigenschaften (z.B. verschiedene Karosserieformen), teilweise durch länderspezifische Merkmale (z.B. Crashverstärkungen) und teilweise durch kundenindividuelle Ausstattungswünsche (z.B. Schiebedach). Im vorliegenden Fall konnten die Varianten mithilfe eines Variantenbaums auf drei grundlegende Modelle reduziert werden, die sich durch die Karosserieform unterscheiden. Für den Betrachtungsumfang dieser Arbeit wird dies als hinlänglich angesehen.

Varianteentstehungspunkte. Die Konkretisierung von Varianten in realen Automobilwerken erfolgt häufig stufenweise in Abhängigkeit der Varianteentstehungspunkte. Diese Punkte können je nach Variante unterschiedlich verortet sein, was die Vergleichbarkeit erschwert. Daher wurde im vorliegenden Fall auf die Betrachtung von Varianteentstehungspunkten verzichtet. Stattdessen werden Varianten ab dem Moment der Auftragsfreigabe als fixe, nachträglich nicht veränderbare Eigenschaften von physischen Aufträgen behandelt.

Anbindung an Maschinensteuerung. Die Beeinflussung der anvisierten Leistungskennzahlen des Produktionssystems erfolgt im Wesentlichen auf MES-Ebene, sodass eine Anbindung des Agentensystems an die Maschinensteuerung bzw. SPS-Ebene in diesem Fall nicht notwendig ist. Wie in Abschnitt 3.3.1 bereits beschrieben könnte eine solche Einbindung z.B. über IEC 61131 bzw. IEC 61499 realisiert werden. Ihre Umsetzung ist jedoch nicht Inhalt dieser Arbeit.

Kennzahlen zur Leistungsmessung

Um die Leistung des Systems nach Industriestandards zu messen, wurden die folgenden sieben Kategorien ausgewählt: Ausbringung, Ressourcen, Liefertermin, Durchlaufzeit, Produktionsprogramm, Produktionsreihenfolge und unfertige Erzeugnisse. Für jeden dieser sieben Bereiche wurden spezifische Kennzahlen ausgewählt, die die Leistung des Systems in geeigneter Weise messen. Die Liste enthält sowohl typische in der Automobilindustrie verwendete Kennzahlen als auch deskriptive statistische Standardmethoden wie Mittelwert und Standardabweichung und ist in Tabelle 7-7 zusammengefasst.

Kategorie	Kennzahl	Beschreibung	Einheit
1. Ausbringung	Systemausbringung	Anzahl gefertigter, fehlerfreier Karossen pro Zeiteinheit.	[jobs per hour (jph)]
2. Ressourcen	Ressourcen- ausnutzung	Auslastung der Ressourcen während der Betriebszeit. Entspricht OEE, da keine Ausschussteile/Aufträge betrachtet werden.	[%]
3. Liefertermin	Liefertreue	Prozentsatz der Aufträge, die innerhalb eines Toleranzfensters von +/-2,5% des ursprünglichen Lieferdatums ausgeführt werden.	[%]
	Mittlere Liefertermin- abweichung	Mittelwert der Lieferterminabweichungen aller Aufträge.	[Tage]
	σ Liefertermin	Standardabweichung des Liefertermins.	[Tage]
4. Durchlaufzeit	Mittlere DLZ	Mittelwert der DLZ aller Aufträge.	[Stunden]
	σ DLZ	Standardabweichung der DLZ.	[Stunden]
5. Produktions- programm	Programm- einhaltung	Grad, zu dem das ursprüngliche Produktionsprogramm mit einem Toleranzfenster von Null erfüllt wird.	[%]
	Mittlere Sequenz- abweichung R000	Mittelwert der Sequenzabweichungen bei Auftragsfreigabe (R000).	[Anzahl Karossen]
	σ Programm	Standardabweichung von Ist- zu Soll-Reihenfolge bei Auftragsfreigabe.	[Anzahl Karossen]
6. Produktions- reihenfolge	Perlenkettengüte	Grad, zu dem die freigegebene Auftragssequenz das geplante Produktionsprogramm mit einem Toleranzfenster von Null abbildet.	[%]
	Mittlere Sequenz- abweichung R800	Mittelwert der Sequenzabweichungen am Ende der Produktion (R800).	[Anzahl Karossen]
	σ Sequenz	Standardabweichung von Ist- zu Soll-Produktionsablauf am Ende der Produktion (R800).	[Anzahl Karossen]
7. Bestand unfertiger Karossen	Systemfüllungsgrad	Grad, zu dem die technische Kapazität des Produktionssystems durch physische Aufträge genutzt wird.	[%]

Tabelle 7-7: Kennzahlen für die Evaluation der Leistung des Agentensystems

Da die Simulation zufällig ausgewählte Ereignisse wie MTBF und MTTR enthält, ist ein Simulationslauf nicht replizierbar. Um Ausreißer zu kompensieren, wurde die Simulation für jede Konfiguration mehrfach wiederholt und die Ergebnisse für die Kennzahlen gemittelt. Ausreißer können jedoch wertvolle Informationen über die Qualität der Ergebnisse liefern und werden daher in Abschnitt 7.4.2 diskutiert. Das programmierte Agentensystem wurde mit den folgenden Eingangsdaten und Prozessparametern aus einer echten Automobilproduktion gespeist.

Programm- und Prozessparameter

Auftragsreihenfolge. Ausgangspunkt der Simulationen ist ein echtes Produktionsprogramm für 24 Stunden, das bereits durch einen Sequenzierungsalgorithmus unter Berücksichtigung der technischen Restriktionen der Fabrik in die optimale Reihenfolge gebracht wurde. Die resultierende Produktionssequenz umfasst eine typische Auftragsmenge von ca. 2.000 Aufträgen.

Produktmix. Die beiden Fertigungslinien können, wie bereits in Tabelle 7-6 dargestellt, die drei Karosserietypen A, B und C in ihren jeweiligen Verhältnissen fertigen. Insgesamt soll ein Produktmix im Verhältnis 6:3:1 erreicht werden.

Liefertermine. Die internen Liefertermine, d.h. das Datum, an dem Karosserien an die Lackiererei als internen Kunden geliefert werden müssen, liegen bei gewöhnlichen Aufträgen zwischen 12 und 24 Stunden ab dem geplanten Produktionsstart. Etwa 1% der Aufträge sind Eilaufträge, die eine kurze Lieferzeit von 10 bis 12 Stunden haben und schneller gefertigt werden müssen.

Ressourcenparameter. Die Bearbeitungszeiten sind, mit Ausnahme der Engpassressourcen S40, S60 und S80, auf 60 Sekunden pro Takt festgelegt. Die Engpassressourcen sind doppelt so schnell, da sie den Input von zwei Ressourcen aufnehmen müssen. Ihre Bearbeitungszeit beträgt daher 30 Sekunden pro Takt. Ein Fertigungsabschnitt setzt sich aus jeweils zehn Arbeitseinheiten zusammen und fasst daher bis zu zehn Karosserien. Der Reservierungsprozess startet eine Sekunde vor Fertigstellung eines Produktionsschrittes und das Reservierungszeitfenster ist ebenfalls auf eine Sekunde festgelegt.

Störungen. Um die tatsächliche Situation in der Fertigung möglichst genau abzubilden, wurden realistische Störungen in das Modell integriert. Dafür wurden die zwei Instandhaltungsparameter mittlere Ausfallzeit (engl. *Mean Time Between Failures*, kurz MTBF) und mittlere Reparaturzeit (engl. *Mean Time To Repair*, kurz MTTR) simuliert. Jeder RA erhält diese Daten beim Start über eine JSON-Datei und verwaltet seinen Zustand anschließend selbst. Um Ressourcenausfälle zu simulieren, steht den RAs eine Funktion zur Verfügung, mit der sie eine zufällige Zeit bis zum nächsten Ausfall (engl. *Time Before Failure*, kurz TBF) und eine zufällige Zeit bis zur Reparatur (engl. *Time To Repair*, kurz TTR) berechnen können (vgl. Abbildung 7-19 und 7-20). Die TBF wird über den MTBF-Wert abgeleitet. Elektronische Bauteile haben für gewöhnlich einen exponentiell verteilten Störungsverlauf, der bei Vernachlässigung von Verschleiß- und Alterungerscheinungen auch für mechanische Bauteile gilt. Die TBF kann daher über eine Exponentialverteilung gut approximiert werden, weshalb sie typischerweise für Lebensdauerverteilungen verwendet wird [KüS14]. Die TTR basiert auf dem MTTR-Wert der Ressource und kann durch eine Erlang-2-Verteilung gut simuliert werden, da sie von Menschen durchgeführte Reparaturvorgänge besser abbildet als eine Exponentialverteilung [Bir14]. Da eine computergenerierte gleichverteilte Zufallszahl nicht direkt verwendet werden kann, um zufällige TBF und TTR zu ermitteln, muss der RA zur Berechnung der beiden Zeiten auf die Inversionsmethode als Simulationslemma zurückgreifen.

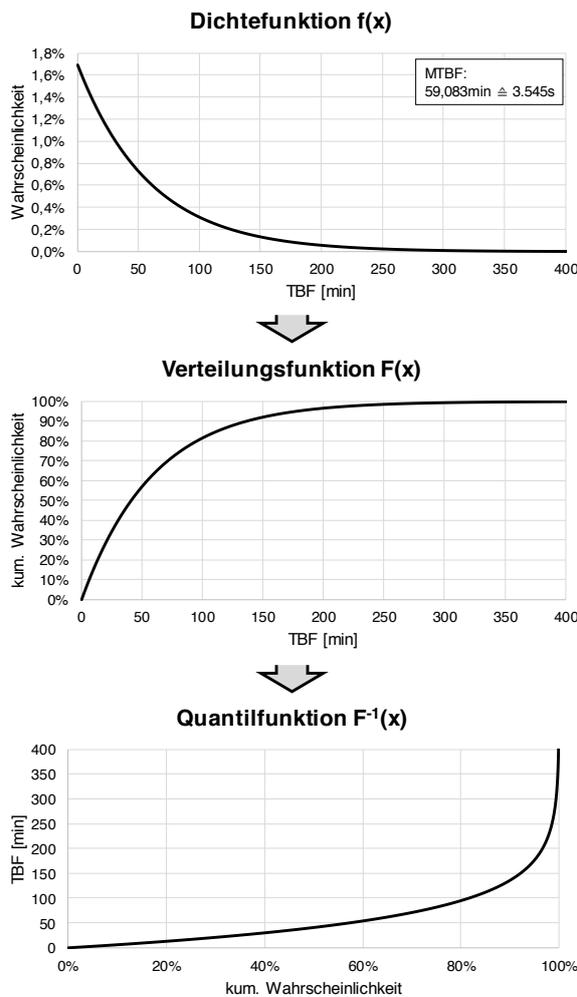


Abbildung 7-19: Zufallsfunktion für TBF als Exponentialverteilung

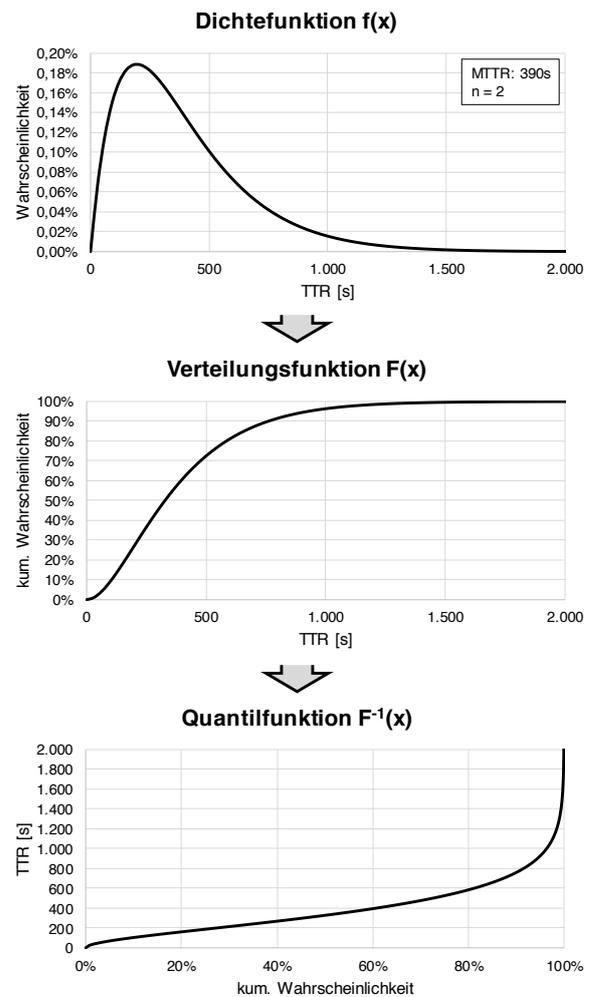


Abbildung 7-20: Zufallsfunktion für TTR als Erlang-Verteilung mit $n = 2$

Durch Anwendung der Inversionsmethode auf eine Verteilungsfunktion $F(x)$ erhält man die Quantilfunktion $F^{-1}(x)$. Diese hat auf der Abszisse die kumulierte Wahrscheinlichkeit und auf der Ordinate die entsprechende TBF oder TTR aufgetragen. Setzt man für x nun eine computergenerierte gleichverteilte Zufallszahl ein, erhält man eine der zugrundeliegenden Verteilung entsprechende Zeit für TBF und TTR. Nachdem der RA diese Werte berechnet hat, weiß er, wann eine Störung simuliert werden muss und schaltet die Ressource zum berechneten Zeitpunkt ab. Dies erreicht er, indem er dem DF den Status „gestört“ mitteilt. Im Anschluss an die Reparaturzeit setzt der Agent den Status der Ressource beim DF wieder auf „verfügbar“, sodass die Ressource weiterarbeitet.

Auf Basis des beschriebenen Designs wurden Simulationsexperimente durchgeführt, deren Ergebnisse im Folgenden diskutiert werden.

7.4.2 Simulationsergebnisse und kritische Beurteilung

Für den Leistungsvergleich zwischen den Simulationsläufen des konventionellen und des agentenbasierten Produktionssteuerungssystems wurde eine direkte Gegenüberstellung gewählt. Zur Veranschaulichung werden die Ergebnisse der Simulationsläufe des konventionellen Systems als

Index herangezogen. Dies ermöglicht es Leistungsunterschiede zwischen den Systemen als Prozentsatz darzustellen und Verbesserungen sowie Verschlechterung mit dem Index zu vergleichen. Die Ergebnisse der Gegenüberstellung sind in Tabelle 7-8 zusammengefasst.

Kennzahl	Verschlechterung	Verbesserung
Systemausbringung [jph]		+8%
Ressourcenausnutzung [%]		+8%
Mittlere Lieferterminabweichung [d]		+1%
σ Liefertermin [d]		+8%
Mittlere DLZ [h]		+9%
σ DLZ [h]		+21%
Programmeinhaltung [%]	-27%	
Mittlere Seq.-Abw. R000 [Karossen]	-36%	
σ Programm [Karossen]	-32%	
Perlenkettengüte [%]	-22%	
Mittlere Seq.-Abw. R800 [Karossen]	-9%	
σ Sequenz [Karossen]	-29%	
Systemfüllungsgrad [%]		+8%

Tabelle 7-8: Leistungsvergleich beider Produktionssteuerungssysteme

Obwohl die Leistung des Agentensystems bei den meisten Kennzahlen besser als beim konventionellen System ist, wird bei der eingehenden Analyse deutlich, dass die Ergebnisse stark von der Kategorie der betrachteten Kennzahlen abhängen. Während das agentenbasierte System bei einigen Kategorien bessere Ergebnisse erreicht, ist das konventionelle System bei anderen Kategorien stärker. Beachtliche Verbesserungen erzielt das Agentensystem vor allem bei den Kennzahlen zur Ausbringung und Ressourcenauslastung, die in direkter Korrelation stehen. Die höhere Ressourcennutzung wird durch die Anpassungsfähigkeit des Systems erreicht, die es ermöglicht, dynamisch auf unvorhergesehene Ereignisse zu reagieren und somit die Gesamtleistung des Produktionssystems zu verbessern. Dies gilt auch für Kennzahlen im Bereich der Liefertreue. Da das Agentensystem die Freigabe von Aufträgen ablehnt, die aktuellen technischen oder logistischen Störungen unterliegen, ist die Liefertreue besser als im konventionellen System. Es ist andererseits jedoch auch zu beobachten, dass in den Bereichen Produktionsprogramm und Produktionsreihenfolge die Stabilität der agentenbasierten Lösung deutlich hinter der des konventionellen Ansatzes zurückliegt. Dies liegt daran, dass sich das Agentensystem kontinuierlich an die aktuelle Situation auf dem Shopfloor anpasst, was den Nachteil hat, dass es zu Abweichungen vom geplanten Produktionsprogramm und zu einer Durchmischung der Produktionsreihenfolge kommen kann. Dies ist eine logische Konsequenz der Adaptivität und muss keinen wirtschaftlichen Nachteil der agentenbasierten Lösung darstellen, auch wenn sich die Anpassungsfähigkeit des Systems auf die genannten Kennzahlen negativ auswirkt. Änderungen der Produktionsreihenfolge können für Hersteller, die das Perlenkettenprinzip anwenden, zwar eine

Herausforderung darstellen. Heutige Produktionssysteme arbeiten jedoch häufig mit Vorlaufzeiten, die groß genug sind um eine Anpassung an diese Veränderungen zu ermöglichen ohne die Effizienz des Systems zu beeinträchtigen. Eine Verschlechterung der Programmeinhaltung und Reihenfolge kann daher bewusst in Kauf genommen werden, um einen ausgewogeneren Produktmix, bessere Liefertreue und höhere Ressourcenauslastung zu erzielen.

Das vorgeschlagene agentenbasierte Steuerungssystem hat dennoch Limitationen. Eine Herausforderung ist die Genauigkeit der vorhergesagten Störungsdauer, von der die Systemleistung wesentlich abhängt. Viele der heutigen Produktionssysteme können solche Prognosen nicht selbstständig in entsprechender Güte bereitstellen. Stattdessen sind sie darauf angewiesen, dass das Wartungspersonal oder Disponenten die Daten manuell in das System eingeben, was in der Regel mit Verzögerungen verbunden ist. Es ist jedoch zu erwarten, dass auch in diesem Szenario der Einsatz des Agentensystems sinnvoller ist als wie heute üblich lediglich auf Grundlage menschlicher Entscheidungen zu handeln. Dabei kann insbesondere ausgenutzt werden, dass das Agentensystem sofort auf neue Informationen reagieren und datenbasiert Entscheidungen in Echtzeit treffen kann, um z.B. neue Aufträge freizugeben oder zu sperren. Durch den Einsatz von Methoden der Künstlichen Intelligenz kann die Entscheidungsfindung potenziell noch weiter verbessert werden. Weitere Details zu dem Anwendungsfall können in [Bec16], [LZB17] und [ZaL17] nachgelesen werden.

Zusammenfassend stellt das in diesem Kapitel vorgeschlagene agentenbasierte Produktionssteuerungssystem einen für die Fertigungsindustrie praktikablen Ansatz dar, um der ständig steigenden Komplexität in der Produktionsplanung und -steuerung sowie den zunehmend turbulenten Umwelteinflüssen zu begegnen. Das entwickelte System verbindet die Vorteile moderner Sequenzierungsalgorithmen und agentenbasierter Produktionssteuerung. Es ermöglicht daher gleichzeitig das Erreichen qualitativ hochwertiger Lösungen für Sequenzierungsprobleme und die Fähigkeit sich selbstständig an unerwartete Veränderungen anzupassen. Die entwickelte Systemarchitektur basiert auf gängigen Entwurfsmustern agentenbasierter Produktionssteuerungssysteme und ermöglicht eine optimale Ressourcenzuweisung zur Laufzeit des Systems. Die in den Simulationen des betrachteten Anwendungsfalls erreichten signifikanten Verbesserungen bei wichtigen Leistungsindikatoren sind voraussichtlich auch auf andere Produktionssysteme übertragbar. Es kann daher davon ausgegangen werden, dass der Einsatz agentenbasierter Steuerungsansätze in anderen Bereichen der Fertigungsindustrie zu signifikanten Produktivitätsverbesserungen führen kann.

Trotz der erreichten Leistungssteigerung existieren zahlreiche Möglichkeiten zur weiteren Verbesserung des Agentensystems. Es könnten systemspezifische Verbesserungen eingeführt werden, um z.B. die Effizienz des Nachrichtenaustausches, die Schwere des Programmcodes und die zeitliche Koordination der Agenten zu optimieren. Solche Ansätze führen jedoch lediglich zu inkrementellen Verbesserungen. Mehr Potenzial liegt daher vermutlich in der Implementierung

von künstlicher Intelligenz in das Agentensystem. Durch Ansätze wie Reinforcement Learning oder Deep Learning könnten z.B. die Entscheidungsprozesse bei der Ressourcenzuweisung und Auftragsfreigabe deutlich verbessert werden. Als konsequenter nächster Schritt sollte schließlich die schrittweise Implementierung des agentenbasierten Produktionssteuerungssystems in eine moderne Fabrik angestrebt werden, um die beschriebenen Potenziale auszuschöpfen und die Anwendbarkeit in der Praxis zu demonstrieren.

Die Vielseitigkeit des agentenbasierten Ansatzes wird im nächsten Kapitel anhand eines weiteren Beispiels demonstriert, das sich von der Karosseriefolgesteuerung deutlich unterscheidet.

8 Anwendungsfall 2: Routenzugsteuerung

Die Eignung von Agentensystemen für die Automobilproduktion wurde im vorangehenden Kapitel bereits am spezifischen Beispiel der Karosseriefolgesteuerung aufgezeigt. Darauf aufbauend soll zur abschließenden Beantwortung der dritten Forschungsfrage im Folgenden die Übertragbarkeit der eingesetzten Methoden und Entwurfsmuster demonstriert sowie das Spektrum der Anwendbarkeit von Agentensystemen aufgezeigt werden. Zudem wird ein zweites Agentensystem benötigt um eine Integration der interdependenten Systeme vornehmen zu können, die Kern der vierten Forschungsfrage ist. Folglich findet auch in diesem Kapitel die Konzeption, Programmierung, Simulation und Bewertung eines Agentensystems am Beispiel der Steuerung von Routenzügen zur werksinternen Materialbelieferung statt. Diese unterscheidet sich in ihren Ausprägungen und Anforderungen deutlich von der Karosseriefolgesteuerung. Der Aufbau dieses Kapitels folgt dennoch der gleichen Gliederungslogik wie Kapitel 7, da sie universell anwendbar ist. In Abschnitt 8.1 erfolgen die ersten drei Schritte des Ablaufs von Simulationsstudien nach Law, d.h. es wird die Problemstellung aufgezeigt, Daten werden aufgenommen und Annahmen formuliert sowie validiert. Im daran anknüpfenden Abschnitt 8.2 erfolgt die Konzeptionierung des Agentensystems nach der DACS-Methode. Im Anschluss daran wird in Abschnitt 8.3 gemäß dem vierten und fünften Schritt nach Law auf die Konzepte und Algorithmen eingegangen, die für die Programmierung und Validierung des Systems relevant sind. In Abschnitt 8.4 werden schließlich die letzten beiden Schritte nach Law durchgeführt, indem zunächst das Design der Simulationsexperimente sowie anschließend die Analyse und Bewertung der Simulationsergebnisse stattfindet.

8.1 Ausgangssituation und Zielsetzung

Zum Einstieg in den Anwendungsfall erfolgt in Abschnitt 8.1.1 zunächst die Vorstellung der grundsätzlichen Problemstellung sowie der wesentlichen Rahmendaten der vorliegenden Materialbelieferung mit Routenzügen. Im darauffolgenden Abschnitt 8.1.2 wird das zugrundeliegende Schedulingproblem bei der Steuerung von Routenzügen untersucht und Annahmen formuliert, wie dieses mit einem Agentensystem gelöst werden könnte. Zur Validierung des agentenbasierten Lösungsansatzes wird in Abschnitt 8.1.3 analysiert, in welchen angrenzenden Bereichen Agentensysteme bereits erfolgreich eingesetzt werden. Schließlich wird in Abschnitt 8.1.4 die mit dem Anwendungsfall zu erreichende Zielsetzung festgelegt.

8.1.1 Problemstellung

Die werksinterne Materiallogistik ist durch eine Vielzahl von parallelen, miteinander vernetzten Prozessen charakterisiert und stellt daher eine steuerungstechnische Herausforderung dar [Mei09]. Die Routenzuglogistik ist hierfür ein repräsentatives Anwendungsszenario in der Automobilindustrie, da sie bei praktisch allen großen Herstellern Anwendung findet. Aufgrund der hohen Anzahl unabhängiger Entitäten ist sie ferner ein potenzielles Anwendungsszenario für Industrie 4.0.

Routenzüge gehören, wie auch Gabelstapler und Fahrerlose Transport-Fahrzeuge (FTF), zur Kategorie der flurgebundenen Fördersysteme und sind kritische Elemente des innerbetrieblichen Werksverkehrs. In der Automobilindustrie werden Routenzüge benutzt, um standardisierte Behälter mit Material zum Ort der Wertschöpfung zu transportieren. Sie werden eingesetzt, um eine schnelle und bedarfsgerechte Anlieferung zu realisieren und das Vorhalten von Material an den Fertigungslinien zu vermeiden. Daher sind sie ein zentraler Bestandteil zur Umsetzung von Lean Production Prinzipien. An eine Zugmaschine werden meist mehrere Anhänger geschlossen, die mindestens einen Ladungsträger fassen können. Man unterscheidet dabei üblicherweise zwischen Klein-, Groß- und Spezial-Ladungsträgern. Die ersten beiden haben eine vorgegebene Größe und können für unterschiedliche Teile verwendet werden, auch wenn Behälter immer nur eine Sorte Teile enthalten. Spezialbehälter werden vor allem für große Teile verwendet, die sich aufgrund ihrer Form nicht für den Transport mit Standardbehältern eignen [GDK13].

Als konkreter Anwendungsfall wurde das in Abbildung 8-1 dargestellte Routenzugsystem ausgewählt. Es dient der Belieferung von zwei separaten Endmontagelinien in einem großen Automobilwerk und hat Zugriff auf ein Materiallager und einen Routenzugbahnhof. Das Materiallager wird von mehreren externen Lieferanten mit Bauteilen beliefert und versorgt die Routenzüge mit dem notwendigen Material. Entsteht ein entsprechender Bedarf an einem der 30 Bedarfssorte an den Montagelinien, werden die benötigten Behälter ausgelagert und im Routenzugbahnhof auf die Routenzüge verladen. Um eine pünktliche Lieferung an die Bedarfssorte zu gewährleisten, werden bedarfsgesteuert Touren gebildet und ausgeführt. Dafür enthält das System 15 Routenzüge, die mit jeweils vier Anhängern ausgestattet sind. Die Anhänger fassen jeweils einen Groß- oder Spezialladungsträger und bewegen sich auf vordefinierten, statischen Routen – ähnlich Buslinien im öffentlichen Personennahverkehr. Zusammen führen die Routenzüge täglich etwa 1.500 Belieferungen aus. Die Montagelinien selbst bestehen aus mehreren Fertigungsabschnitten zwischen denen Fahrwege liegen. Die Fahrwege können teilweise in beiden Richtungen, teilweise nur in einer Richtung befahren werden. An den Bedarfssorten angekommen wird das Material schließlich in die Karosserien verbaut.

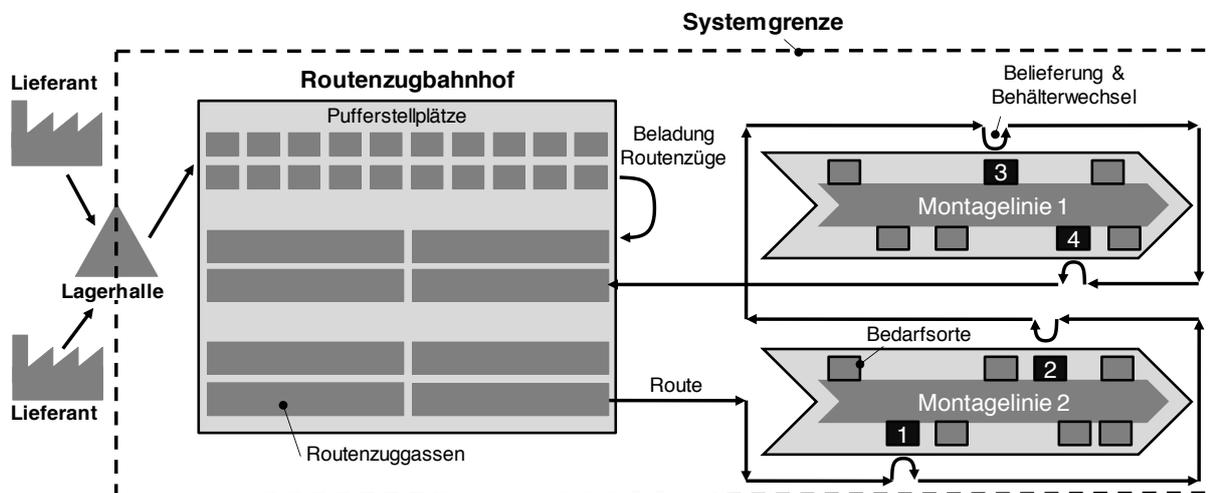


Abbildung 8-1: Anwendungsfall 2 – Layout Routenzugsystem

In der Wertschöpfungskette ist das Routenzugsystem horizontal nach dem ersten Anwendungsfall der Karosseriefolgesteuerung verortet und bezieht sich auf einen physisch kleineren Bereich (vgl. Abbildung 6-2). Vertikal ist es daher unter dem ersten Anwendungsfall angesiedelt und Entscheidungen finden hier vor allem auf den Ebenen 6 und 7 statt. Trotzdem weist auch die Routenzugsteuerung eine große Komplexität auf, da der Detaillierungsgrad hier größer ist – statt ganze Karossen, werden einzelne Bauteile betrachtet.

Die ohnehin komplexe Steuerung der Routenzüge wird dadurch erschwert, dass in der Tourenplanung ausschließlich statische, vordefinierte Routen verwendet werden. Jeder Bedarfsort ist dabei genau einer Route zugeordnet, sodass nur Aufträge des gleichen Fahrkreises miteinander kombinierbar sind. Das reduziert die Kombinationsmöglichkeiten erheblich und schränkt die Agilität des Systems deutlich ein. Dies ist ein typisches Szenario im Standardfall des statischen Routenverkehrs [DGA14]. Von diesem Standardfall unterscheidet sich das vorliegende Beispiel nur dadurch, dass die Zeitabstände zwischen den Touren nicht konstant sind, sondern nach dem aufkommenden Bedarf gesteuert werden.

Die Komplexität des Steuerungsproblems, kombiniert mit den beschränkten Freiheitsgraden der statischen Routenplanung resultiert im Status quo in zahlreichen Ineffizienzen. Ein Materialabriss an der Montagelinie verursacht in der Regel einen sofortigen Stillstand der Produktion und kann je nach Fabrik oft zu einem Verlust zwischen 10.000 und 30.000 Euro pro Minute führen. Diese Kosten können nur durch technische Entkoppelung der Bandabschnitte und mithilfe von Pufferstrecken begrenzt werden [Her12]. Es ist daher verständlich, dass die Vermeidung von Verspätungen bei der Materialanlieferung häufig als oberste Maxime für die Auslegung derartiger Steuerungssysteme angesetzt wird. Die entsprechend hohen hinterlegten Sicherheitszeiten führen jedoch dazu, dass Routenzüge häufig Touren starten müssen, ohne dass alle Anhänger voll beladen sind. Dadurch sinkt die Auslastung der vorhandenen Routenzugkapazität und die Anzahl der erforderlichen Touren steigt. Dies zeigt auch die Erhebung der Fahrzeiten je Auftrag über einen Zeitraum von fünf Tagen in Tabelle 8-1, die im Rahmen der studentischen Arbeit [Hab16] erstellt wurde. Darin schwankt die Fahrzeit je Auftrag im betrachteten System aufgrund der teilweise niedrigen Auslastung zwischen 6,8 und 7,8 Minuten. Durchschnittlich weichen die Anlieferzeiten zwischen 14,6 und 27,3 Minuten von der Wunschzeit (WZ) ab.

Tag	Fahrzeit/Auftrag	Aufträge/Tour	Terminabweichung
1	6,808	3,31	25,817
2	7,442	3,16	25,883
3	7,619	3,15	27,317
4	7,334	3,07	14,583
5	7,798	2,96	14,783

Tabelle 8-1: Anwendungsfall 2 – Kennzahlen im Status quo i.A.a. [Hab16]

Die Steuerungsarchitektur des Routenzugsystems ist zentral aufgebaut und besteht aus drei separaten IT-Systemen, die für die Bedarfsverwaltung, die Lagerverwaltung und die Routenzugsteuerung verantwortlich sind. Ähnlich wie die Karosseriefolgesteuerung, ist auch die Steuerung der Routenzuglogistik nicht auf die zunehmende Dynamik und Komplexität ausgelegt, mit der sich die Automobilindustrie konfrontiert sieht. Durch die steigende Anzahl kundenindividueller Fahrzeugausstattungen und technischer Innovationen ist der an den Montagelinien anfallende Teilebedarf jedoch zu einem großen Teil stochastisch definiert. Um diesen Bedarf optimal abzudecken, bedarf es einer hohen Flexibilität und Robustheit der Produktion. Ein Trend, der sich in Verbindung mit dem steigenden Wettbewerb auf volatilen Märkten noch verstärken wird [KrS14]. Um diesem Trend mithilfe eines Agentensystems angemessen begegnen zu können, wird im Folgenden das hinter der Routenzugsteuerung stehende Schedulingproblem erläutert.

8.1.2 Tourenplanung von Routenzügen

Die Steuerungsentscheidungen des betrachteten Routenzugsystems können auf ein bekanntes Standardproblem des Operations Research zurückgeführt werden. Dabei handelt es sich um das sogenannte Vehicle Routing Problem (VRP) – im Deutschen auch als Tourenplanungsproblem bezeichnet – bei dem allgemein die Zuweisung von Fahrzeugen zu Transportaufträgen zwischen Knoten eines Netzwerkes behandelt wird [DGA14]. Das VRP fällt in den Bereich der ganzzahligen linearen Optimierung bzw. der kombinatorischen Optimierung [FLP04].

Dewitz und Ramser [DaR59] setzten sich bereits 1959 mit diesem Zuordnungsproblem auseinander. In der als „Truck Dispatching Problem“ bezeichneten Arbeit ermittelten sie den Tourenplan für die Benzinbelieferung zwischen einem Depot und mehreren Knoten unter Berücksichtigung der Kapazitätsbeschränkungen der Transportfahrzeuge. Im Standardfall entsprechen alle Start- und Zielpunkte dem Depot und es existiert eine begrenzte oder unbegrenzte Anzahl Fahrzeuge mit beschränkter Kapazität. Übertragen auf den hier betrachteten Anwendungsfall der Routenzugsteuerung entspricht das Depot dem Routenzugbahnhof und die Knoten den Bedarfsorten an den Montagelinien. Die Lösung des VRP besteht aus zwei Teilen: der Zusammenfassung von Aufträgen zu einer Tour (auch Clustering genannt) und dem Routing, in dem die Reihenfolge der Knoten definiert wird, die innerhalb einer Tour angefahren werden. Letzteres Teilproblem entspricht dem klassischen Problem des Handlungsreisenden, bei dem ein Fahrzeug mit unbegrenzter Kapazität mehrere Ziele in beliebiger Reihenfolge anfahren soll. Im Standardfall ist das Ziel die Minimierung der Fahrwege bzw. der Fahrtkosten, es existiert aber eine Vielzahl von Erweiterungen dieses Standardfalls, die teilweise andere Zielsetzungen verfolgen [DGA14].

In einigen bekannten Erweiterungen enthält das VPR zusätzliche Bedingungen oder Merkmale wie z.B. mehrere Depots (Multiple Depot VRP, kurz MDVRP), eine festgelegte Anzahl an Belieferungsfahrzeugen (Capacitated VRP, kurz CVRP), definierte Zeitfenster für die Belieferung (VRP with Time Windows, kurz VRPTW) oder beliebige Start- und Zielpunkte (VPR with Pickup- and Delivery, kurz VRPPD) [DGA14, VCG14]. Im Falle der dynamischen Tourenplanung kann sich

darüber hinaus die Auftragslage während der Planung jederzeit verändern, z.B. indem neue Aufträge hinzukommen oder andere storniert werden. Der vorliegende Anwendungsfall der Routenzugsteuerung stellt demzufolge eine Form eines dynamischen VRPTW dar.

Je nach Ausprägung werden in den Erweiterungen des VRP unterschiedliche Zielsetzungen verfolgt. Neben der oben genannten Minimierung der Fahrzeit bzw. Fahrtkosten wird z.B. auch die Minimierung der eingesetzten Fahrzeuge, der Tourenanzahl, der Einsatzzeit (bestehend aus Fahrzeit und Wartezeit), des CO₂-Ausstoßes oder der Strafkosten für schlechten Service (z.B. Verspätungen) angestrebt. Es können aber auch mehrere Faktoren in komplexeren Zielfunktionen zusammengefasst werden [DGA14]. Dabei können zwischen den Faktoren Zielkonflikte bestehen, da z.B. die geringste Gesamtentfernung nicht mit der geringsten Tourenanzahl erreicht wird [Boo07].

Obwohl das Tourenplanungsproblem dem Job Shop Scheduling Problem ähnlich ist, werden für die beiden Optimierungsprobleme in der Praxis meist unterschiedliche Ansätze verwendet [BPS03]. Die Lösungsverfahren können in die drei Kategorien exakte Verfahren, problemspezifische Heuristiken und Meta-Heuristiken eingeteilt werden [DoS10].

Exakte Lösungsverfahren

Zur Modellierung des VRP existieren in der Literatur drei grundsätzliche Ansätze für exakte Lösungsverfahren: Vehicle flow formulations, Commodity flow formulations und das Set partitioning problem [ToV02]. Aufgrund der intensiven Erfordernisse an Rechenkapazität bzw. -zeit können mathematisch exakte Methoden nur für verhältnismäßig kleine Probleme angewendet werden. Für die Nutzung in der Praxis sind sie daher oft ungeeignet, weshalb an dieser Stelle darauf verzichtet wird tiefer auf sie einzugehen. Die Einschränkungen resultieren daraus, dass das Grundproblem der Tourenplanung – ähnlich wie im Beispiel der Karosseriefolgesteuerung in Anwendungsfall 1 – zur Klasse der NP-schweren Optimierungsprobleme gehört [ToV02]. Vor diesem Hintergrund wurden in Wissenschaft und Industrie zahlreiche Heuristiken und Meta-Heuristiken entwickelt, die das zugrundeliegende Optimierungsproblem in einer angemessenen Zeit näherungsweise lösen können.

Heuristiken

Clarke and Wright haben 1964 eine Heuristik für das Grundproblem der VRP vorgestellt. Dafür haben sie den ursprünglichen Lösungsalgorithmus von Dantzig weiterentwickelt, indem sie zur Lösung des Problems einen Greedy-Ansatz angewendet haben. Das daraus resultierende Verfahren wird Savings-Heuristik genannt [CIW64]. Eine weitere Heuristik ist das Sweep-Verfahren, das zu den vielseitigsten Techniken der algorithmischen Geometrie gehört. Hierbei handelt es sich um ein Paradigma, bei dem eine räumliche Dimension in eine zeitliche verwandelt wird, indem aus einem statischen d-dimensionalen Problem ein dynamisches (d-1)-dimensionales Problem formuliert wird. Eine detaillierte Erläuterung des Verfahrens und typischer Beispiele ist in [Kle98] zu finden.

In vielen Fällen haben sich jedoch Metaheuristiken gegenüber den vorgestellten, einfachen Heuristiken als überlegen herausgestellt. Im Gegensatz zu den problemspezifischen Heuristiken, definieren Metaheuristiken abstrakte Folgen von Schritten für die Lösung von Optimierungsproblemen, die auf viele Problemstellungen übertragen werden können.

Metaheuristiken und neue Ansätze

Die im Zusammenhang mit dem VRP verwendeten Metaheuristiken gehen meist auf lokale Suchstrategien (engl. *local search*) zurück. Das dahinterliegende Prinzip besteht aus drei Schritten: Erstens wird eine Startlösung bestimmt. Zweitens wird eine Nachbarschaft von ähnlichen Lösungen definiert. Drittens wird diese Nachbarschaft ganz oder teilweise nach der besten Lösung durchsucht [BHW96, BHW97]. Auf das Problem der Routenzüge übertragen bedeutet dies, dass die Reihenfolge und die Zuordnung von Aufträgen zu Routenzügen zwischen unterschiedlichen Lösungen getauscht werden bis eine ausreichend gute Lösung erreicht ist. Beispiele für solche Metaheuristiken sind im Zuge der Reihenfolgeplanung (vgl. Abschnitt 7.1.3) bereits vorgestellt worden. Dazu zählen u.a. evolutionäre bzw. genetische Algorithmen, Simulated Annealing und Tabu-Search, aber auch unterschiedliche Varianten von Ameisenkoloniealgorithmen. Ein ameisenkoloniebasierter Ansatz für dynamisch-deterministische Tourenplanungsprobleme ist z.B. der dynACO-Algorithmus von Boomgarden. Er unterscheidet sich von anderen Ameisenalgorithmen u.a. dadurch, dass das Pheromonupdate multiplikativ anstatt additiv vorgenommen wird [Boo07].

Zu den neueren Ansätzen in dem Bereich gehört die Unified Hybrid Genetic Search (UHGS) genannte Metaheuristik von Vidal et al. [VCG14]. Der UHGS-Ansatz kann auf eine Vielzahl von unterschiedlichen Tourenplanungsproblemen angewendet werden und eignet sich auch zur Lösung großer Optimierungsprobleme mit einer Vielzahl von Restriktionen. In einem Benchmark, in dem die Leistung von UHGS mit 1.099 anderen Methoden verglichen wurde, erzielte der Ansatz in 1.045 Fällen eine bessere oder gleich gute Lösung. Der Abstand zur besten bekannten Lösung lag dabei fast immer unter einem Prozent [VCG14].

Der Survey von Werner et al. [WBS18] zeigt eine Reihe neuer Ansätze auf, die auch für die Tourenplanung und -anpassung von Relevanz sind. In der Arbeit von Gholami und Törnquist-Krasmann [GhT18] wird eine neue Heuristik zur Lösung des Re-Scheduling-Problems von Fernverkehrszügen in Echtzeit beschrieben. Die Autoren interpretieren das Problem als blocking Job-Shop Problem und nutzen einen hybriden Ansatz aus gemischten und alternativen Graphen für die Modellierung der Infrastruktur. Zur Lösung von Konflikten wurde eine Heuristik entwickelt, die für eine Modellgröße von 80 Zügen eine nahezu optimale Lösung in etwa 10 Sekunden ermitteln kann. Um diese kurzen Berechnungszeiten zu erreichen wird eine Liste potenzieller Tauschkandidaten für die Konfliktlösung erstellt und das Problem auf lokaler Ebene betrachtet. Dafür benötigt das Modell eine dynamische Aktualisierung der Daten des Netzwerks. Agentensysteme könnten hier helfen den aktuellen Zustand und mögliche Rekonfigurationen in Echtzeit abzubilden

sowie – in Kombination mit dem gezeigten Ansatz – Anweisungen an die jeweiligen Züge weiterzuleiten und umzusetzen. Ähnlich wie im Falle der Karosseriefolgesteuerung gilt es also auch in diesem Bereich die Weiterentwicklungen der Algorithmen, (Meta-)Heuristiken und Rechenkapazitäten zu beobachten. Darüber hinaus könnten Potenziale durch Kombination neuer, dynamischer (Meta-)Heuristiken und agentenbasierter Steuerungssysteme entstehen.

Herausforderungen in der Routenzugsteuerung

Routenzugsysteme, wie das in diesem Anwendungsfall beschriebene, unterliegen im Realbetrieb einer Vielzahl interner und externer Störungen wie z.B. fehlendes Material, Anlagen-/Routenzugstörungen, Staus in Fahrgassen, Fehllieferungen etc. Solche Störungen treten in der Regel ohne lange Vorwarnzeiten auf und bedürfen daher einer schnellen Reaktion. Für die Ermittlung von Lösungen mit gängigen Verfahren werden im Durchschnitt jedoch Berechnungszeiten von etwa 10 Minuten benötigt. In Extremfällen erfolgt ein Timeout nach 30 Minuten [VCG14]. Damit eignen sich diese Ansätze sehr gut für wöchentliche oder tägliche Planungsvorgänge. Treten jedoch spontan Störungen auf, werden die ermittelten Lösungen in vielen Fällen instabil und es bedarf einer ad-hoc Reaktion. Die Dynamisierung der Steuerung rückt folglich in den Vordergrund [KIG13]. Da die Berechnungszeiten der beschriebenen Verfahren eine agile Reaktion erschweren, bedarf es für die dynamische Echtzeit-Tourenplanung der Routenzuglogistik Mechanismen, die eine schnellere Adaption ermöglichen.

Agentensysteme verfügen über die notwendigen Voraussetzungen, um sich unter den in dem Anwendungsfall der Routenzuglogistik vorherrschenden Bedingungen innerhalb kürzester Zeit auf eine neue Situation einzustellen. Um die Realisierbarkeit und den Nutzen von Agentensystemen für den vorliegenden Anwendungsfall zu validieren, werden im folgenden Abschnitt einige erfolgreich umgesetzte Agentensysteme aus angrenzenden Bereichen vorgestellt.

8.1.3 Agentensysteme in der Intralogistik

Die folgenden drei Beispiele stellen eine Auswahl von agenten- bzw. CPS-basierten Steuerungssystemen aus dem für den Anwendungsfall der Routenzuglogistik relevanten Bereich dar. Weitere Systeme sind z.B. das Depalettiesystem der TU München [ScC10], das multimodale Transportsteuerungssystem von Jung [Jun15] sowie das Projekt Railcab der Universität Paderborn [DKR08]. Sie können in der entsprechenden Literatur nachgeschlagen werden.

Kommissionierlagersystem der viastore systems GmbH

Die viastore systems GmbH hat ein dezentral gesteuertes Kommissionierlagersystem entwickelt, das als Puffer zwischen Produktion und Versand fungiert [KuG13]. Versandaufträge verfügen über einen RFID-Tag und können mit einer Transporteinheit kommunizieren, um die Kommissionierung zu steuern. Die Transporteinheit ist an drei verschiedene Lagerbereiche angeschlossen, deren Bahnhöfe aufgrund der verschiedenen Nachfragehäufigkeit über unterschiedliche Versorgungsprinzipien bestückt werden. Die Aufträge und Transporteinheiten werden durch Agenten

repräsentiert und ihre Services werden in einem Gelbe-Seiten-Verzeichnis registriert. In Abhängigkeit der Streckenauslastung stimmen sich mehrere Transporteinheitagenten ab und passen an Fördertechnikkreuzungen ihre weitere Route zum Ziel an. Die Routeninformationen werden auf den RFID-Tags gespeichert. Aufträge, die aus mehreren Paketen bestehen, werden vor dem Warenausgang mithilfe von zwei Regalbediengeräten zusammengeführt [GLL10, KuG13].

In dem Projekt konnte die operative Leistung des Systems durch die Echtzeit-Routenführung mit schnellen, dezentralen Entscheidungen und Stauberücksichtigung verbessert werden. Gleichzeitig wurde auch die Komplexität des zugrundeliegenden Steuerungsproblems durch Zerlegung in einzelne Teilprobleme deutlich reduziert. Durch die Kapselung der Steuerungslogik in Agenten wurde zudem die Wiederverwendbarkeit von Softwareblöcken und die Austauschbarkeit von mechatronischen Modulen erhöht.

Staplerleitsystem der PSI Logistics GmbH

Im Rahmen eines Forschungsprojektes der PSI Logistics GmbH wurde ein Agentensystem für die Steuerung eines innerbetrieblichen Staplerleitsystems entwickelt [GöL10]. Damit die Übergabe von Transporteinheiten gewährleistet werden kann, ist das Agentensystem über einen Schnittstellenagenten mit anderen Transportsystemen verbunden. Intern verfügt das System über einen zentralen Auftragsmanageragenten, der sowohl für die Verwaltung der Auftragslisten und Stellplätzen als auch für die Vergabe von Transportzielen verantwortlich ist. Ein zentraler Ressourcenmanageragent legt die Transportprioritäten der Aufträge fest und weist sie unter Berücksichtigung der Stellplatzkapazität anschließend den Staplern zu. Transporteinheitagenten und Stapleragenten kommunizieren miteinander, um eine gemeinsame Auftragsausführung festzulegen. Positionsänderungen werden über RFID und Eingaben am mobilen Datenterminal erfasst.

Das agentenbasierte Staplerleitsystem zeichnet sich durch einen hohen Grad an Skalierbarkeit, kurze Hochlaufzeiten sowie eine schnelle Rekonfigurierbarkeit im Falle von Veränderungen aus. Die selbstständig ausgeübten Ausgleichsprozesse bei Sollabweichungen machen das System zudem robust gegenüber Störungen im Prozessablauf [GöL10].

Bedarfsorientiertes Milkrunsystem Wittenstein bastian GmbH

In einem Verbundforschungsprojekt unter Beteiligung der Wittenstein bastian GmbH wurde eine Schaufensterfabrik mit einem bedarfsorientierten Milkrunsystem realisiert [SSL14], das Ähnlichkeiten zum hier betrachteten Anwendungsfall der Routenzuglogistik hat. Ein Milkrunsystem ist ein manuell betriebenes, zyklisches Transportsystem mit vordefinierten Routen und Fahrplänen [GPS13]. In dem vorliegenden Beispiel verfügt das Milkrunsystem über einen Elektrozug, der einmal pro Stunde bedarfsorientiert Transportaufträge entlang statischer Routen ausführt. Damit hat das System zwar einen deutlich geringeren Umfang als der in diesem Kapitel betrachtete Anwendungsfall der Routenzuglogistik, es wird aber dennoch der Gedanke der Dezentralisierung

verfolgt. Dabei wird der CPS-Gedanke umgesetzt, indem Ressourcenverbräuche dezentral erfasst, bewertet und zur Optimierung der Produktionsprozesse genutzt werden. Wenn die Maschinen in der Schaufensterfabrik stochastisch verteilte Abhol- und Anlieferbedarfe auslösen, werden die Mitarbeiter mittels Tablet-PCs darüber informiert. Die Identifikation von Transportaufträgen erfolgt über Quick-Response-Codes (kurz QR-Codes). Anschließend wird der nächste Abfahrtszeitpunkt mithilfe einer zentralen Tourenplanung ermittelt und später den Mitarbeitern mitgeteilt. Eine systeminterne Kommunikation zwischen Teilnehmern ist zwar möglich, das System an sich ist jedoch geschlossen, sodass z.B. kein Lager direkt angebunden ist. Mit dem System konnte eine deutliche Reduzierung der Anzahl notwendiger Fahrzyklen und gefahrener Teilrunden erreicht werden, sodass mit den verfügbaren Ressourcen mehr Transportaufträge durchgeführt werden können. Die manuelle Datenaufnahme soll in der nächsten Umsetzungsstufe durch den Einsatz intelligenter Werkstückträger ersetzt werden [SSL14].

Die beschriebenen Beispiele zeigen bereits hohes Potenzial auf, denn die Systeme erhöhen die Flexibilität und Leistungsfähigkeit in den für sie relevanten Bereichen. Sie gehen jedoch nach Auffassung des Autors an einigen Stellen nicht weit genug. So sind in obigem Milkrunsystem nur statische Routen abgebildet, obwohl mit der Implementierung von dynamischen Routen weitere Potenziale wie z.B. die Erhöhung der Ressourcenauslastung und die Verkürzung von Fahrzeiten realisiert werden könnte. Ferner hat das angebundene Staplerleitsystem keine Funktion zur Bestimmung des kürzesten Weges und der Fokus ist teilweise nur auf einzelne Funktionen wie z.B. das Depalettieren beschränkt. Es liegt also ein ähnliches Bild wie im Falle der fixen Förderbänder im Beispiel des Elektrohängebahnsystems und des Gepäck-Transport-Systems in Kapitel 7.1.4 vor. Schließlich beziehen sich Systeme wie das Kommissionerlagersystem oder das Milkrunsystem auf verhältnismäßig überschaubare Bereiche. Es ist jedoch offen wie frei bewegliche Transportfahrzeuge auf sehr weitläufigen Flächen mit hunderten Prozessschritten und tausenden Einzelteilen gesteuert werden können. Dies ist jedoch im Automobilbau gängig. Im verbleibenden Teil dieses Kapitels wird daher versucht diese Lücken zu schließen, indem ein Agentensystem zur Steuerung der komplexen Routenzuglogistik für den vorliegenden Anwendungsfall der Automobilindustrie erarbeitet wird. Die dabei zu erreichenden Ziele sind wie folgt definiert.

8.1.4 Zielsetzung

Aufgrund der Vielzahl an stochastisch anfallenden und über eine große Fläche verteilten Materialbedarfe ist eine hohe Flexibilität und Robustheit des Steuerungssystems erforderlich, was mit den im Status quo vorliegenden statischen Routen nur schwer zu erreichen ist. Insbesondere im Falle von Störungen oder spontanen Bedarfsänderungen ist der Optionsraum im Status quo deutlich eingeschränkt. Folglich müssen hohe Sicherheitszeiten eingeplant werden und das Material wird oftmals viel früher angeliefert als es tatsächlich benötigt wird. Dies verursacht unnötigen Materialbestand und nimmt zudem viel Platz ein. Daher sollen die statischen Routen mithilfe ei-

nes Agentensystems aufgebrochen werden, um die Flexibilität, Robustheit und Effizienz des Gesamtsystems zu erhöhen. Ziel ist es dabei einerseits eine zuverlässige Belieferung der Bedarfsorte sicherzustellen und verspätete Materialanlieferungen unbedingt zu vermeiden. Andererseits soll aber auch die Auslastung der Routenzüge gesteigert werden, indem Wartezeiten reduziert und die Anzahl Aufträge pro Tour erhöht werden. Damit soll schließlich auch die Menge des am Fließband vorgehaltenen Materials und der damit zusammenhängenden Kapitalbindungskosten gesenkt werden. In einer weiteren Entwicklungsstufe könnte unter Umständen die Umstellung von einem Zwei-Behälter- auf ein Ein-Behälter-Prinzip für die Vorhaltung von Teilen verfolgt werden. Dies gehört im ersten Schritt jedoch ausdrücklich nicht zur Zielstellung.

Zusammenfassend sollen folgende konzeptionelle Eigenschaften zur dynamischen Steuerung des Routenzugsystems in dem zu entwickelnden Agentensystem realisiert werden:

- 1) Die zentralistische und statische **Tourenbildung** des Ausgangssystems soll durch eine dezentral und dynamisch in Echtzeit stattfindende Tourenplanung ersetzt werden, mit der die Wunschzeit und die Fahrzeit berücksichtigt wird.
- 2) Der **Tourenbildungszeitraum**, der im Status quo mit der Bedarfsermittlung beginnt und mit dem Tourenstart endet, soll im Agentensystem – sofern genügend Aufträge vorliegen – erst bei Ankunft des Routenzuges am Bahnhof stattfinden.
- 3) Der **Tourenstart** ist derzeit unabhängig von der verbleibenden Zeit zum Wunschtermin und wird ausgelöst, wenn der Routenzug voll beladen ist. Als Schwellwert für den spätesten Starttermin ist ein Abstand von 25 Minuten zum Wunschtermin definiert. Im Agentensystem soll der Tourenstart zu einer festgelegten Zeit vor der mittleren Wunschzeit der Aufträge stattfinden, um deutlich verfrühte Anlieferungen zu vermeiden.
- 4) Während im Status Quo keine Regeln zum **Umgang mit fehlendem Material** bestehen, soll das Agentensystem eine automatische Nachbestellung bei externen Dienstleistern auslösen können.

Zur Realisierung der beschriebenen Eigenschaften wird im nächsten Schritt ein Agentensystem zur Steuerung des beschriebenen Routenzugsystems entwickelt. Dafür wird – wie im vorangehenden Kapitel auch – auf die DACS-Methode zurückgegriffen. Die dafür benötigten Modelle werden im nächsten Abschnitt erarbeitet.

8.2 Entwicklung des Agentensystems

Für die Konzeptionierung des Agentensystems zur Routenzugsteuerung wird das gleiche Vorgehen gewählt, das bereits im Anwendungsfall der Karosseriefolgesteuerung verwendet wurde. In Abschnitt 8.2.1 werden daher zunächst die Steuerungsentscheidungen und die zwischen ihnen bestehenden Abhängigkeiten betrachtet. Das daraus abzuleitende Entscheidungsmodell ist die

Grundlage für die Definition von Agenten und das Skizzieren eines Agentenmodells im nachfolgenden Abschnitt 8.2.2. In Abschnitt 8.2.3 werden schließlich die Interaktionen zwischen den Agenten untersucht und geeignete Protokolle für sie ausgewählt, was im Interaktionsmodell resultiert.

8.2.1 Entscheidungsmodell

Analyse der Steuerungsentscheidungen

Basis für die Untersuchung der Steuerungsentscheidungen ist eine Analyse der Prozesse im Status quo, die durch die studentische Arbeit [Hab16] begleitet wurde. Die Übersicht der Steuerungsaufgaben und -informationen aus Kapitel 5.2 wurde dabei genutzt, um die unterschiedlichen Kategorien von Steuerungsentscheidungen zu identifizieren und zu klassifizieren. Da in diesem Anwendungsfall kein klar bestimmter physischer Materialkreislauf existiert, sondern die Routenzüge sich auf beliebigen Strecken bewegen können, wurde der Entscheidungsbegriff für diesen Anwendungsfall etwas weiter ausgelegt und die Steuerungsentscheidungen entlang ihrer chronologischen Abfolge betrachtet. Die wesentlichen Prozessschritte zur Sicherstellung der Materiallogistik gliedern sich in die folgenden vier Bereiche (vgl. Abbildung 8-2).

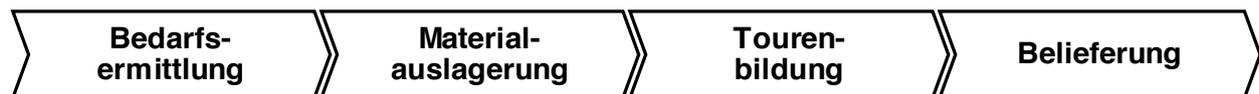


Abbildung 8-2: Prozessschritte Materialbelieferung mit Routenzügen

Der Auslöser für den Prozess ist die Überquerung des Montageeinlaufs durch eine Karosse in der durch die Montagesequenz vorgegebenen Karosseriefolge. Die Eingangsgrößen für die Routenzugsteuerung hängen also von den Ausgangsgrößen der Karosseriefolgesteuerung ab. An einem Erfassungspunkt wird das Fahrzeug identifiziert und anschließend eine **Bedarfsermittlung** für den konkreten Auftrag sowie ein Abgleich mit dem Materialbestand ausgeführt. Falls notwendig erfolgen auch Lagerabrufe, damit das notwendige Material rechtzeitig an den vorgesehenen Stellen vorliegt. Die Belieferung der Lagerplätze erfolgt im Voraus durch Lieferanten und basiert auf Prognosen und wöchentlichen Bestellungen. Für die im nächsten Schritt notwendige **Materialauslagerung** existieren im vorliegenden Fall sowohl konventionelle, durch Facharbeiter bediente Lager, als auch moderne, automatisierte Lager, die keinen menschlichen Eingriff erfordern. In konventionellen Lagern erfolgt die Ein- und Auslagerung von Material mithilfe von Gabelstaplern. In den automatischen Lagern werden Behälter selbstständig über eine Fördertechnikbindung ausgelagert und zum Routenzugbahnhof befördert. Am Zielort werden die Behälter mit Gabelstaplern aufgenommen und auf freien Stellplätzen zwischengelagert. Im nächsten Schritt erfolgt die **Tourenbildung**, die im Status quo abgeschlossen wird, wenn entweder vier Behälter eingeplant sind (maximale Kapazität erreicht) oder die Zeit zum Wunschzeitpunkt eines Auftrags 25 Minuten unterschreitet. Touren werden nach dem First-Come-First-Serve-Prinzip dem nächsten Routenzug zugewiesen, der am Bahnhof eintrifft. Die anschließende Beladung der Routenzüge erfolgt – wie die Materialauslagerung auch – mittels Gabelstapler. Für die Ausführung der

Belieferung werden zwei Typen von Routenzügen verwendet: sogenannte Linksentlader können ihre Behälter nur zur linken Seite ausladen, Rechtsentlader nur zur rechten. Die Zuordnung des Materials zum jeweiligen Routenzugtyp erfolgt automatisiert. Sobald die Beladung abgeschlossen ist wird die Tour gestartet. Ein Fahrer bewegt den Routenzug dabei entlang der vorgegebenen statischen Route zu den jeweiligen Bedarfsorten, entlädt die Ware an den vorgesehenen Stellen und nimmt leere Behälter wieder mit. Nachdem das Material abgeladen wurde, werden die leeren Behälter zum Leergutbahnhof gebracht und der Routenzug kehrt zum Bahnhof zurück. Dort erhält er eine neue Tour, die entlang jeder der möglichen statischen Routen führen kann.

Als Basis für das zu konzipierende Agentensystem ergibt sich der folgende in Abbildung 8-3 dargestellte Ablauf für die Tourenplanung und Transportsteuerung mit den damit zusammenhängenden Entscheidungen.

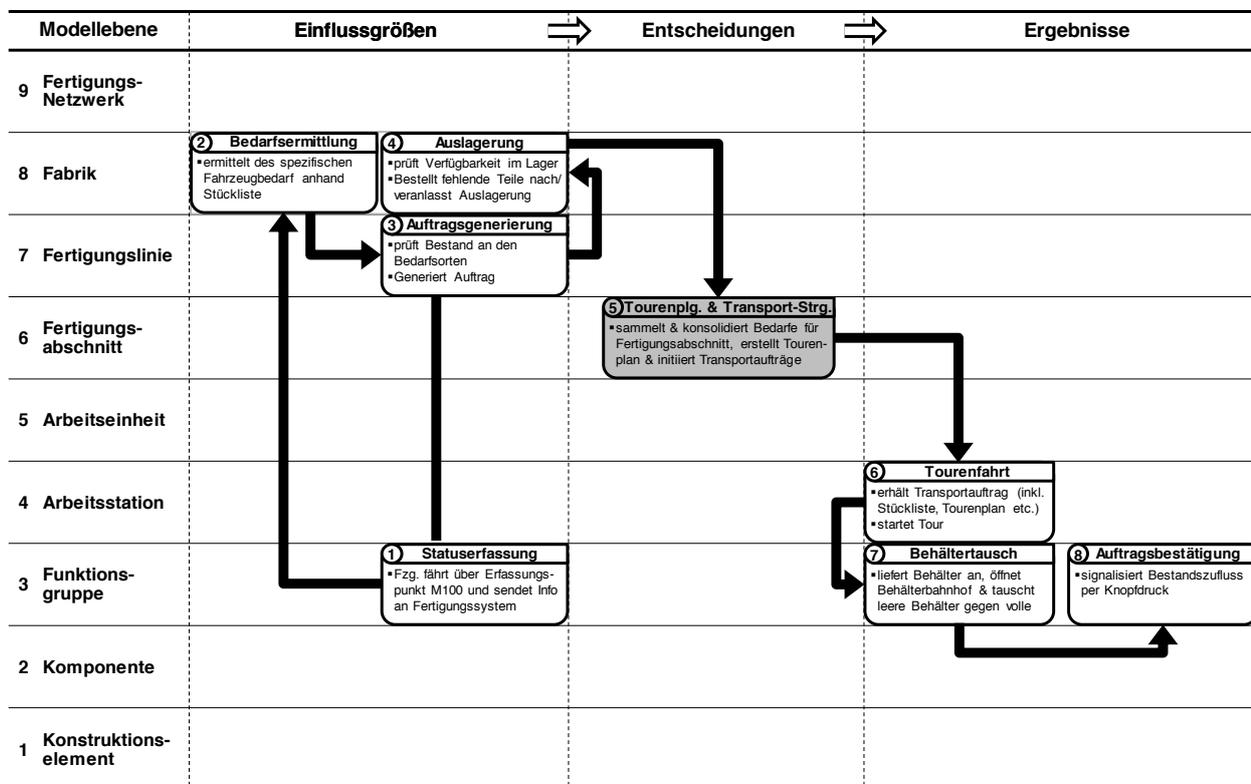


Abbildung 8-3: Tourenplanung und Transportsteuerung i.A.a [Hab16]

Während die Informationen sowohl von niedrigen (z.B. Fahrzeu erfassung) als auch hohen Ebenen (z.B. Bedarfsmeldung über mehrere Montagelinien hinweg) erfolgen, findet die Entscheidung zur Bestimmung einer Tour auf einer mittleren Ebene statt. Die Signalisierung des Tourenstarts und die Quittierung der Aufträge erfolgt über Handdatenterminals (HDT) der Routenzugfahrer.

Aus den obigen Betrachtungen ergeben sich für den vorliegenden Anwendungsfall neun Steuerungsentscheidungen, die samt ihrer jeweiligen detaillierten Eigenschaften in Tabelle 8-2 zusammengetragen sind.

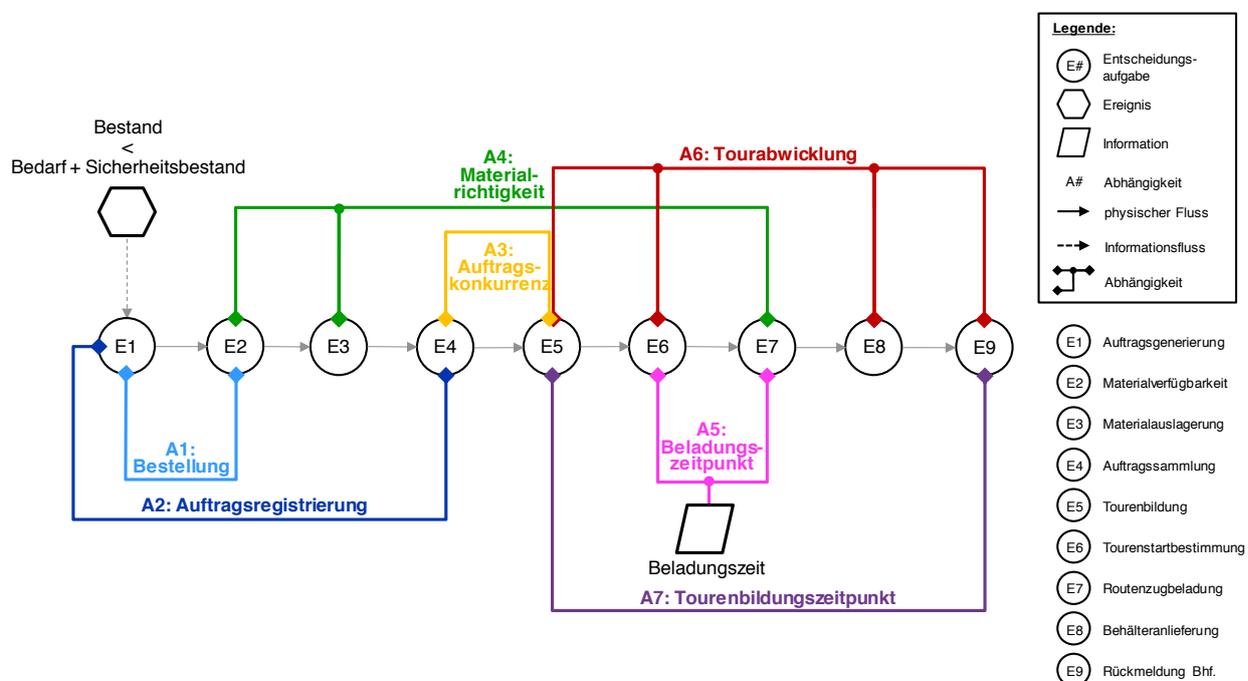
ID	E1	E2	E3	E4	E5	E6	E7	E8	E9
Titel	Auftragsgenerierung	Überprüfung der Materialverfügbarkeit und ggf. Nachbestellung	Auslagerung des Materials	Auftragssammlung	Tourenbildung aus den Aufträgen	Bestimmung des Tourenstarts	Routenzugbelastung	Anlieferung der Behälter auf der Tourenfahrt	Rückmeldung am Bahnhof
Parameter	Bedarfsort, Bedarfsorttyp, Wunschzeit	Material, FIFO-Datum, Lagerplatz	Material, Lagerplatz	Bedarfsort, Bedarfsorttyp, Material, Wunschzeit	Tournummer für den Routenzug, Aufträge (enthalten Parameter aus E1), Distanzen zwischen Bedarfsorten und dem Bahnhof, Routenzug-ID	Aufträge einer Tour (enthalten Parameter aus E1)	Tourenstartzeit, Beladungszeit, Routenzug-ID, Behälter	Tourenstartzeit, Aufträge auf der Tour, Fahrzeit der Tour	Ankunftszeit, Routenzug-ID
Steuerungs-Interface	Leitsystem	Leitsystem	HDT der Staplerfahrer	Leitsystem	Leitsystem	System	HDT des Staplerfahrers	HDT des Routenzugfahrers	HDT des Routenzugfahrers
Auslöser	Bestand \leq Bedarf + Sicherheitsbestand	Bedarfsmeldung im Lager	Materialverfügbarkeit	Materialverfügbarkeit und 40 Minuten vor WZ	Rückkehr eines Routenzuges, sofern genügend kompatible Aufträge bekannt sind	abgeschlossene Tourenbildung	Zeit \geq Startzeit - Beladungszeit	Ankunft am Bedarfsort	Ankunft des Routenzuges am Bahnhof
Entscheidungsraum	Alle Materialien, die an diesem Bedarfsort zu einer WZ innerhalb der nächsten drei Stunden benötigt werden	Verfügbare Behälter mit dem richtigen Material im Lager und jedes beliebige Material im Falle der Nachbestellung	Alle Lagerplätze mit dem benötigten Material	Jeglicher Zeitpunkt nach Feststellung der Materialverfügbarkeit	Alle bereits bekannten und noch nicht eingeplanten Aufträge des passenden Routenzugtyps	jegliche Zeitpunkte ab der jetzigen Zeit	jegliche Zeitpunkte nach Tourenplanung	jegliche Zeitpunkte nach dem Tourenstart	jegliche Zeitpunkte nach der letzten Anlieferung
Entscheidungsregel	-	Wähle für die Reservierung den Behälter mit der richtigen Sachnummer aus, sodass FIFO eingehalten wird. Ist das Material nicht vorhanden, bestelle sofort nach.	Wähle den für diesen Auftrag reservierten Behälter aus.	Nimm den Auftrag im Auftragspool auf, sobald der Auslöser erfolgt.	Bevorzuge Aufträge mit einem geringen Wert aus gewichteter Zeit bis zur WZ und Fahrzeit. Plane so viele Aufträge wie möglich ein. Berücksichtige die Beschränkung aufgrund der Typen Links- und Rechtsentlader.	Wähle den Startzeitpunkt als eine definierte Zeit vor der mittleren WZ der Aufträge.	Wähle Beladungszeit möglichst spät, aber so, dass Routenzug starten eingehalten wird.	Bestätige die Auftragsausführung, sobald der Behälter gewechselt wurde.	Bestätige die Rückkehr am Bahnhof, sobald der Routenzug am Bahnhof eingetroffen ist.
Lokale Entscheidungsregel	-	-	-	-	-	-	-	-	-

Tabelle 8-2: Steuerungsentscheidungen Routenzugsteuerung i.A.a. [Hab16]

Interdependenzen auch zwischen den Aufträgen selbst (**A3 – Auftragskonkurrenz**). Bei der Auslagerung von Material ist zu berücksichtigen, dass diese mit dem Ergebnis der Verfügbarkeitsprüfung zusammenhängt, da nur Lagerplätze in Betracht kommen auf denen sich Bauteile der bestellten Sachnummern befinden (**A4 – Materialrichtigkeit**). Um Routenzüge rechtzeitig zu beladen ist die Ermittlung eines konkreten Tourenstarts erforderlich (**A5 – Beladungszeitpunkt**), denn sie erfolgt als Rückwärtsrechnung vom geplanten Starttermin. Für die Konfiguration der Tour selbst sind mehrere Faktoren zu berücksichtigen (**A6 – Tourabwicklung**). Einerseits muss für die Ermittlung des Anlieferungszeitpunkts die Auftragsreihenfolge und Tourenstart einbezogen werden. Andererseits ist der Tourenstart selbst durch die jeweiligen Wunschzeiten der ausgewählten Aufträge determiniert. Das Tourenende ist schließlich erst erreicht, wenn alle Aufträge ausgeführt sind und der Routenzug wieder im Bahnhof eingetroffen ist. Daraus ergibt sich die letzte Abhängigkeit, denn neue Touren können erst geplant werden, wenn mindestens ein Routenzug am Bahnhof verfügbar ist (**A7 –Tourenbildungszeitpunkt**).

Weitere Details zu den Abhängigkeiten der Entscheidungen in dem betrachteten Routenzugsystem sind in Tabelle 8-3 dargestellt.

Die Analyse der Steuerungsentscheidungen und ihrer Abhängigkeiten stellt die Basis für die Entwicklung eines Agentensystems dar, dass die Anforderungen des Anwendungsfalls angemessen umsetzen kann. Die Zusammenhänge der Entscheidungen und Abhängigkeiten für den Anwendungsfall sind in Abbildung 8-5 zusammengetragen und werden im nächsten Schritt für die Identifikation von Agenten in dem Steuerungssystem verwendet.



ID	A1	A2	A3	A4	A5	A6	A7
Titel	Bestellung	Auftragsregistrierung	Auftragskonkurrenz	Materialrichtigkeit	Beladungszeitpunkt	Tourenfahrt	Zeitpunkt der Tourenbildung
Entscheidungen	E1, E2	E1, E4	E4, E5	E2, E3, E7	E6, E7	E5, E6, E8, E9	E5, E9
Restriktionen	Berücksichtige keine Behälter bzw. Lagerplätze mit der falschen Sachnummer.	Registriere keine Aufträge, deren Material nicht im Lager verfügbar ist.	Plane nur einen Auftrag bzw. Behälter je Trailer eines Routenzuges ein.	Lagere keine Behälter mit nicht bestellten Sachnummern aus.	Halte den Tourenstart möglichst ein.	Schließe keine Aufträge ab, die nicht eingeplant und die nicht tatsächlich ausgeliefert wurden. Kehre erst zum Bahnhof zurück, wenn alle Aufträge ausgeführt wurden.	Plane erste Aufträge in einer Tour ein, wenn sich ein verfügbarer Routenzug am Bahnhof eingefunden hat.
Präferenzen	-	-	Wähle den Auftrag, der den geringsten Wert an gewichteter Fahrzeit und Dauer bis zur WZ bietet.	Berücksichtige bei der Minimierung der Lagerdauer nur Behälter mit der passenden Sachnummer.	Veranlasse die Routenzugbeladung rechtzeitig vorm Tourenstart.	Schließe den Auftrag per Knopfdruck am HDT ab, sobald der richtige Behälter auf dem Routenzug mit dem leeren Behälter am Bedarfsort ausgetauscht wurde.	-

Tabelle 8-3: Abhängigkeiten von Entscheidungen der Routenzugsteuerung i.A.a. [Hab16]

8.2.2 Agentenmodell

Bei der Entwicklung des Agentenmodells wird wie schon im vorangehenden Anwendungsfall das Ziel verfolgt, Steuerungsentscheidungen so stark wie möglich voneinander zu trennen, um die Komplexität der einzelnen Agenten zu minimieren und das Optimierungsproblem bestmöglich zu verteilen. Dabei werden auch hier Entscheidungsaufgaben zusammengefasst, die dasselbe Steuerungsinterface haben, dieselben Systemkomponenten beeinflussen oder immer stark voneinander abhängig sind. Auf Basis dieser Kriterien und des Entscheidungsmodells aus Abbildung 8-5

wird im Weiteren daher ein Agentenmodell erarbeitet, das die Entscheidungsaufgaben so gegliedert, dass sie möglichst sinnvoll in einem Agenten konsolidiert werden können. Dabei ergeben sich die vier in Abbildung 8-6 skizzierten Agentencluster.

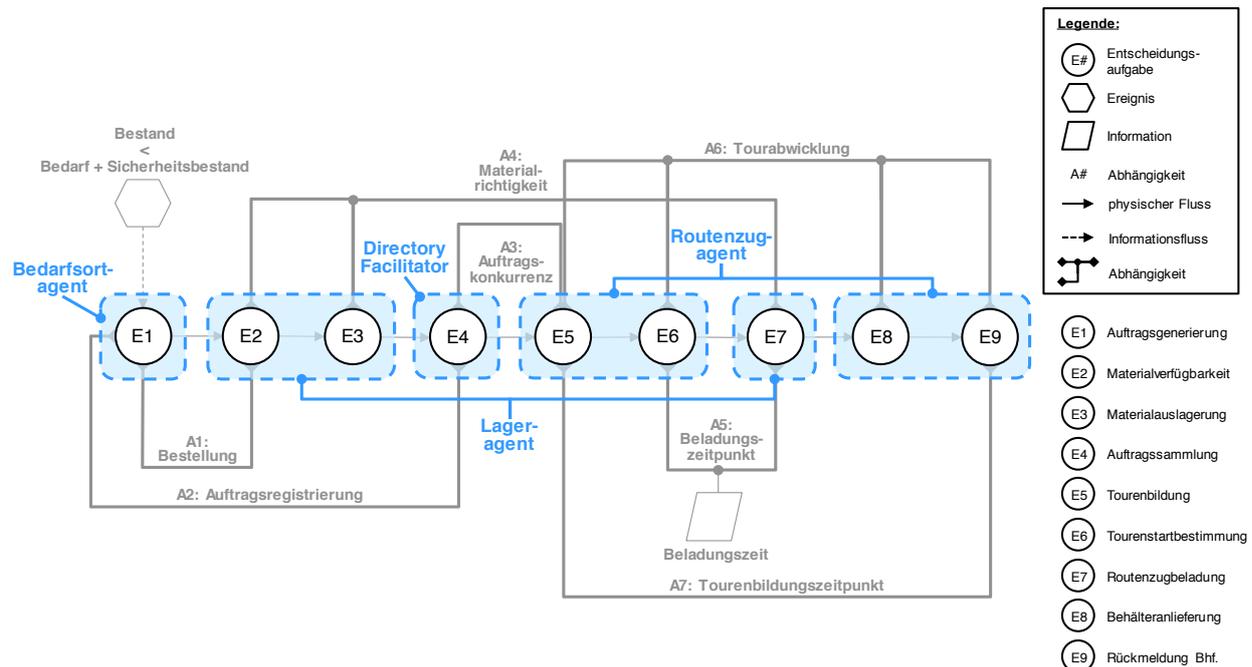


Abbildung 8-6: Entscheidungscluster und Agenten Routenzugsteuerung

Das erste Cluster besteht aus den **Bedarfsortagenten (BOA)**, die repräsentativ für die Orte in der Montagehalle stehen, an denen der Materialbedarf entsteht. Zu den Aufgaben der BOAs gehört die Auftragsgenerierung und -verfolgung. Dies beinhaltet auch die Verwaltung und Weitergabe aller auftragsbezogenen Daten an andere Agenten, z.B. bei der Auftragsregistrierung, der Verfügbarkeitsprüfung und der Tourenbildung. Die BOAs werden aktiv sobald der Materialbestand vor Ort unter den definierten Sicherheitsbestand fällt.

Das zweite Agentencluster wird durch die **Lageragenten (LA)** konstituiert, die drei konkrete Aufgaben haben: Erstens fällt ihnen die Überprüfung der Materialverfügbarkeit zu, nachdem ein Bedarf durch einen BOA ausgelöst wurde. Zweitens obliegt ihnen die Steuerung der Auslagerung bzw. das Nachbestellen von Material. Und drittens sind sie für die Steuerung der Beladung von Routenzügen mit dem bestellten Material verantwortlich. Als Steuerungsinterface für den LA kann das HDT des Staplerfahrers benutzt werden, der die Ein- und Auslagerung vornimmt.

In das dritte Cluster fallen die **Routenzugagenten (RZA)**, die Verantwortung für die meisten Aufgaben im System tragen. Einerseits sind sie für den planerischen Teil der Tourenbildung verantwortlich, d.h. sie müssen die eintreffenden Bedarfsmeldungen priorisieren, daraus Touren bilden, sie Routenzügen zuweisen und schließlich den Tourenstart bestimmen. Andererseits müssen sie die Ausführung der Touren steuern, die Anlieferung der Behälter überwachen und die Rückmeldung der Routenzüge am Bahnhof erfassen. Als Steuerungsinterface der RZA kann das HDT des Routenzugfahrers genutzt werden, der den Routenzug lenkt und die Behälter tauscht.

Um all diese Aufgaben anforderungsgerecht ausführen zu können, benötigen die oben genannten Agentencluster einen **Directory Facilitator (DF)**. Der DF ist ihre zentrale Anlaufstelle, um Services zu finden und Informationen zu erhalten. Er fungiert dabei als Blackboard, auf dem alle anderen Agenten wichtige, für die Agentenpopulation notwendige Informationen ablegen können. Darüber hinaus gehört zu den Aufgaben des DF das Sammeln der von Bedarfsortagenten generierten Aufträge und das Weitergeben dieser Informationen für die Tourenplanung. Der DF stellt aber auch Entfernungsinformationen in Form einer Distanzmatrix zur Verfügung. Auf Basis dieser Wegetopologie werden die Fahrzeiten zwischen zwei Punkten in der Halle geschätzt.

Schließlich wird als Spezifikum der Implementierung mit AnyLogic in der Laufzeitumgebung ein **Main-Agent (MnA)** benötigt. Der MnA stellt die Umwelt für die Agenten im System zur Verfügung. Dies schließt die Simulationszeit, den Raum, das Layout, das Netzwerk und die Kommunikation ein, die die Agenten benötigen, um miteinander zu interagieren und das Verhalten des Systems zu simulieren. Über den MnA lassen sich diverse Einstellungen für die Simulationsumgebung definieren [Gri15].

Die Steuerungsaufgaben und -informationen der sich aus den obigen Betrachtungen ergebenden Agententypen sind in Tabelle 8-4 zusammengefasst.

Agententyp	Steuerungsaufgaben	Benötigte Steuerungsinformationen
Bedarfsort-agent (BOA)	<ul style="list-style-type: none"> • Auftragsgenerierung (E1) 	<ul style="list-style-type: none"> • Sachnummer • Bestand & Bedarf am Bedarfsort • Sicherheitsbestand am Bedarfsort • Wunschtermin für Lieferung
Lageragent (LA)	<ul style="list-style-type: none"> • Überprüfung Materialverfügbarkeit & ggf. Nachbestellung (E2) • Auslagerung Material (E3) • Routenzugbeladung (E7) 	<ul style="list-style-type: none"> • Sachnummer • Lagerbestand • Auslagerzeit • Nachbestellzeit
Routenzug-agent (RA)	<ul style="list-style-type: none"> • Tourenbildung aus den Aufträgen (E5) • Bestimmung des Tourenstarts (E6) • Auslieferung der Behälter auf der Tourenfahrt (E8) • Rückmeldung am Bahnhof (E9) 	<ul style="list-style-type: none"> • Beladungszeit der Routenzüge • Fahrzeit (aus Entfernungsmatrix von DF)
Directory Facilitator (DF)	<ul style="list-style-type: none"> • Auftragsammlung (E4) 	<ul style="list-style-type: none"> • Standorte • Entfernungsmatrix (Bedarfsorte, Lager, Routenzüge etc.) • Bedarfsmeldungen
Main-Agent (MnA)	<ul style="list-style-type: none"> • Umwelt für die Agenten im System bereitstellen 	<ul style="list-style-type: none"> • Simulationszeit, Raum, Layout, Netzwerk, Kommunikationsflüsse

Tabelle 8-4: Agentenmodell mit zugehörigen Steuerungsaufgaben und -informationen

Auf der Grundlage des vorgestellten Agentenmodells können im dritten und letzten Schritt des Entwurfsprozesses die Interaktionsprotokolle für die Kommunikation der Agenten untereinander entworfen werden. Dies ist Inhalt des nächsten Abschnitts.

8.2.3 Interaktionsmodell

Im letzten Schritt der DACS-Methode wird das Interaktionsmodell für das Agentensystem erarbeitet. Dafür werden zunächst die notwendigen Interaktionen zwischen den Agenten bzw. die zwischen ihnen bestehenden Abhängigkeiten im Agentenmodell klassifiziert. Im Anschluss findet ein Abgleich mit verfügbaren Interaktionsprotokollen statt und es werden geeignete Protokolle ausgesucht und gegebenenfalls spezifisch angepasst.

Klassifizierung der Abhängigkeiten

Im Folgenden werden zunächst die Interaktionseigenschaften der identifizierten Abhängigkeiten beschreiben und anschließend klassifiziert. Bei der Betrachtung des Agentenmodells in Abbildung 8-6 wird deutlich, dass die Abhängigkeiten A4, A6 und A7 jeweils einem einzigen Agenten zugeordnet sind. Sie lösen folglich keine Interaktion mit anderen Agenten aus und entziehen sich daher der Interaktionsbetrachtung. Die interaktionsrelevanten Ausprägungen der verbleibenden vier Abhängigkeiten ergeben sich wie folgt:

A1. Im Rahmen der Abhängigkeit A1 verfolgt der LA gemeinsam mit einem BOA das Ziel das passende Material für einen konkreten Auftrag zu identifizieren. Dafür müssen lediglich die beiden beteiligten Agenten miteinander kommunizieren und Daten wie z.B. die Sachnummer und die Wunschzeit untereinander austauschen.

A2. Im Fall der Abhängigkeit A2 kommunizieren ebenfalls nur zwei Agenten bilateral miteinander. Die Beteiligten sind jeweils der DF und ein BOA. Ziel ist es dafür zu sorgen, dass ein konkreter Auftrag in einer Tourenplanung berücksichtigt wird. Eine im Vorfeld zu erfüllende Voraussetzung ist, dass das Material auch tatsächlich bereitsteht.

A3. Die Abhängigkeit A3 ist durch eine höhere Komplexität als die obigen Abhängigkeiten charakterisiert. Hier konkurrieren mehrere Aufträge in Form von BOAs um die vier Transportplätze auf den Anhängern eines Routenzugs, der durch einen RZA repräsentiert wird. Auch die Zielsetzung hat mehr Dimensionen als bei den anderen Abhängigkeiten, denn es soll einerseits sichergestellt werden, dass Aufträge weder zu früh noch zu spät am Bedarfsort ankommen und andererseits die Auslastung der Routenzüge maximiert wird. Diese multilaterale Kommunikation erfordert Interaktionsprotokolle einer höheren Komplexität.

A5. Bei der Abhängigkeit A5 liegt erneut ein vergleichsweise einfacher Kommunikationsfall vor. Hier sind ein RZA und ein LA bestrebt eine rechtzeitige Beladung der Routenzüge sicherzustellen. Auch in diesem Fall ist dafür eine simple, bilaterale Kommunikation ausreichend.

Damit ergibt sich für die Interaktionen im Routenzugsystem ein ähnliches Bild wie bereits in Anwendungsfall 1: Es existieren einerseits einfache Kommunikationsflüsse, wie z.B. das Versenden einer Materialanforderung, die mit einfachen Interaktionsprotokollen erfolgen kann. Andererseits bedarf es für die komplexen Kommunikationsflüsse wie sie bei der Verhandlung von Touren vorliegen Protokolle, die der Komplexität der benötigten Kommunikation Rechnung tragen. Um für

diese Anforderungen angemessene Kommunikationsprotokolle zu identifizieren, werden die Abhängigkeiten als nächstes anhand der bereits in Kapitel 7.2.3 angewendeten Klassifikationskriterien von Busmann (vgl. Anhang 21) klassifiziert.

Klassifizierung von A1, A2 & A5. Die Abhängigkeiten A1, A2 und A5 sind durch einfache, bilaterale Kommunikationswege charakterisiert, bei denen jeweils immer *zwei Agenten involviert* sind. So muss z.B. der BOA mit dem LA kommunizieren, um eine Bestellung abzusetzen und der RZA muss mit dem LA interagieren, um den Beladungszeitpunkt zu bestimmen. Da es dabei immer eine eindeutig beste Alternative gibt, die von beiden Agenten bevorzugt wird, sind die verfolgten *Präferenzen kompatibel*. Es liegen keine globalen Beschränkungen vor, sodass die Präferenzen zudem als *nicht-lokal* einzuordnen sind. In jeder Abhängigkeit wird des Weiteren je *eine einzige Verpflichtung* eingegangen und – dadurch, dass zwei Agenten an der Interaktion beteiligt sind – hat die gemeinsame Verpflichtung die *Größe zwei*. Da jeder Agent nur eine Verpflichtung eingeht, ist das Kriterium der Beziehung zwischen den Verpflichtungen *irrelevant*. Die Rollenzuweisung steht bereits zur Designphase *fest* und erfordert keine weitere Rollenzuweisung während der Interaktion. Der Agent, der die Interaktion initiiert erbittet ferner nur eine einzige Handlung von einem anderen Agenten, sodass nur auf Ebene von *Alternativen* interagiert wird. Schließlich akzeptieren die initiiierenden Agenten alle gültigen Antworten ohne eine Möglichkeit zur Ablehnung zu haben, sodass eine *partielle Delegation* vorliegt.

Klassifizierung von A3. Die Abhängigkeit A3 ist durch eine komplexe Kommunikation zwischen mehreren Agenten charakterisiert. Die Anzahl dabei involvierter Agenten ist *fest*, aber zur Designphase unbekannt. Sie ist von den vorliegenden Aufträgen abhängig und entspricht der Anzahl der am DF registrierten und noch nicht eingeplanten BOAs. Darüber hinaus ist an der Tourenbildung stets auch ein RZA beteiligt. Da der RZA die Transportkosten minimieren will, erstellen die BOAs möglichst niedrige Angebote um eingeplant zu werden. Die Präferenzen sind daher auch in diesem Fall *kompatibel*. Wie im vorangehenden Fall sind sie zudem *nicht-lokal* und es wird stets nur *eine einzige gemeinsame Verpflichtung* eingegangen. Ein Unterschied zu den bilateralen Abhängigkeiten besteht darin, dass die Größe der gemeinsamen Verpflichtungen bereits zu Beginn der Interaktion *festgelegt* ist, denn die Anzahl der zu verpflichtenden BOAs ergibt sich aus der Anzahl kompatibler Aufträge, die der Routenzug transportieren kann. Die maximale Anzahl an zu verpflichtenden BOAs ist aufgrund der verfügbaren Anzahl an Anhängern auf vier begrenzt, der RZA hat jedoch die Möglichkeit Aufträge abzulehnen. Die abgelehnten BOAs gehen keine Verpflichtung ein, sodass nicht jeder Agent zwangsweise einer gemeinsamen Verpflichtung unterworfen sein muss. Da die sich ergebende Tour als Bündnis betrachtet wird besteht *keine Beziehung* zwischen mehreren Verpflichtungen. Die Rollenzuweisung ist in der Abhängigkeit A3 *variabel*, da sie nur teilweise während der Designphase vorhergesehen werden kann. Während der am Bahnhof eintreffende RZA und der DF eine von Beginn an definierte Rolle haben, wird

erst während der Interaktion festgelegt, welche BOAs für die Tourenbildung berücksichtigt werden. Die BOAs teilen dem RZA genau einen Wert mit, um auf einer Tour eingeplant zu werden, sodass für die Entscheidung *nur Alternativen* vorliegen, aus denen der RZA auswählt. Die Entscheidungskompetenz liegt daher stets beim RZA und wird *nicht delegiert*.

Alle Eigenschaften der betrachteten Abhängigkeiten sind in Tabelle 8-5 als Übersicht dargestellt.

Aspekt	Klassifikationskriterium	Ausprägungen A1, A2 & A5	Ausprägungen A3
Startsituation	1. Anzahl involvierter Agenten	n=2	fest
	2. Kompatibilität der Beschränkungen & Präferenzen	kompatible Präferenzen	kompatible Präferenzen
	3. Globale Beschränkungen & Präferenzen	nicht-lokal	nicht-lokal
Gemeinsame Verpflichtungen	4. Anzahl gemeinsamer Verpflichtungen	n=1	n=1
	5. Größe gemeinsamer Verpflichtungen	n=2	fest
	6. Beziehung zwischen gemeinsamen Verpflichtungen	-	-
Agentenrollen	7. Rollenzuweisung (Anzahl variable Rollen)	fest (keine)	variabel (Teilmenge)
Prozessanforderungen	8. Informationsverfügbarkeit	nur Alternativen	nur Alternativen
	9. Delegation von Entscheidungskompetenz	partielle Delegation	keine Delegation

Tabelle 8-5: Klassifikation der Abhängigkeiten in der Routenzugsteuerung

Auf Basis der Klassifikation können im nächsten Schritt jeder Interaktion passende Interaktionsprotokolle zugewiesen werden.

Auswahl und Anpassung von Interaktionsprotokollen

Zur Auswahl geeigneter Protokolle für die benötigten Interaktionen wurde auch für diesen Anwendungsfall ein Abgleich mit der in Anhang 22 und 23 beigefügten Protokollbibliothek aus [BJW04] durchgeführt. Der detaillierte Abgleich der Anforderungen für die vorliegenden Anwendungsfälle kann in Anhang 32 und 33 nachgeschlagen werden.

Interaktionsprotokoll für A1, A2 und A5. Für die einfache, bilaterale Kommunikationsform der Abhängigkeiten A1, A2 und A5 kommen aufgrund der Abwärtskompatibilität – wie schon im vorherigen Anwendungsfall – mehrere Protokolle in Frage. Dazu gehört das Plurality Voting Protocol, die Service-oriented Negotiation und das Requesting Action Protocol. Das einfachste für die Zwecke einsetzbare Protokoll ist jedoch für diesen Fall erneut das Requesting Action Protocol (RAP). Mit Ausnahme der Delegation der Entscheidungskompetenz erfüllt es alle Auswahlkriterien und durch Weglassen einer finalen Zustimmung kann das RAP so angepasst werden, dass es den Anforderungen der Interaktion vollumfänglich gerecht wird. Da zudem der Kommunikationsaufwand reduziert wird, ist das RAP die am besten geeignete Technik für den vorgesehenen Verwendungszweck und wird daher als Interaktionsprotokoll ausgewählt.

Interaktionsprotokoll für A3. Auch für die komplexere Kommunikation in der Abhängigkeit A3 kommen mehrere Interaktionsprotokolle in Frage. Diese sind u.a. das Plurality Voting Protocol, das DECIDE Conflict Resolution Protocol sowie das Contract Net Protocol. In diesem Fall ist die durch das Protokoll zu lösende Aufgabe nicht mit einer einfachen, bilateralen Kommunikation möglich, sodass eines der aufwändigeren Protokolle gewählt werden muss. Aufgrund seiner Vielseitigkeit und Flexibilität sowie der zahlreichen damit in der Praxis realisierten Anwendungsfälle ist das Contract Net Protocol (CNP) zur Bewältigung dieser Aufgabe besonders geeignet. Das CNP ist ein einfaches, aber effizientes Protokoll zur Zuweisung von Aufgaben an einzelne Knoten eines Netzwerks. Dabei wird davon ausgegangen, dass ein Knoten die Lösung einer Aufgabe benötigt, die von einem anderen Knoten ausgeführt werden muss und dass potentiell mehrere Knoten existieren, die diese Aufgabe ausführen können. Der erste Knoten schreibt die Aufgabe aus und erhält Angebote von anderen Knoten. Der Knoten, der das beste Angebot abgibt, erhält eine Zuschlagsmeldung, sodass ein Vertrag über die Ausführung der Aufgabe geschlossen wird [BJW04]. Über das CNP können mehrere BOAs um einen Platz auf einem Routenzug werben. Für die bis zu vier Anhänger wird jeweils eine Vereinbarung zwischen dem BOA mit dem besten Angebot und dem RZA geschlossen. Die Herausforderung ist im vorliegenden Anwendungsfall die Größe der Verpflichtung, denn es wird das Ziel verfolgt bis zu vier BOA und einen RZA für eine Tour zu verpflichten. Da jeweils nur ein BOA einem spezifischen Anhänger auf dem Routenzug zugeteilt wird, bedarf es für die vollständige Beladung des Routenzugs eines Zwischenschritts. Daher erfolgt eine Anpassung des CNP in Form einer Ergänzung des Protokolls um ein Blackboard. Das Blackboard wird verwendet, um die Zwischenergebnisse der Verhandlungen festzuhalten und die volle Beladung der Routenzüge sicherzustellen.

Übersicht der Kommunikationsflüsse

Mit den obigen Ausführungen sind allen Interaktionen, die im vorliegenden Anwendungsfall erforderlich sind, passende Protokolle zugewiesen. Als Ergebnis des Entwurfsprozesses ist das aus zwölf Schritten bestehende Interaktionsmodell des Agentensystems in Form eines Kommunikationsdiagramms in Abbildung 8-7 zusammengefasst.

Im ersten Schritt wird durch einen BOA Material bei einem LA angefordert. Für diese einfache bilaterale Kommunikation wird das RAP verwendet. Der LA initiiert die Auslagerung des Materials und bestätigt die Verfügbarkeit dem BOA. Anschließend registriert dieser den Bedarf am DF, der wiederum alle Aufträge sammelt und für die nächsten Touren bereithält. Wenn ein Routenzug am Bahnhof eintrifft, bietet er seine Transportkapazitäten an und der Tourenbildungsprozess startet. Für die komplexere Kommunikation zwischen mehreren BOAs und dem RZA wird das CNP verwendet. Der RZA fordert bei den noch nicht zugewiesenen BOAs ein Angebot für die Materialbelieferung an und erhält von ihnen entsprechende Rückmeldungen. Die Aufträge mit den besten Zielfunktionswerten erhalten den Zuschlag und es wird die Beladung der Routenzüge beim LA angestoßen. Nachdem dies abgeschlossen ist, erhält der RZA eine Rückmeldung vom LA und

beginnt mit der Ausführung des Materialtransports. Nach der erfolgreichen Rückkehr des Routenzugs am Bahnhof wird der Prozess wiederholt.

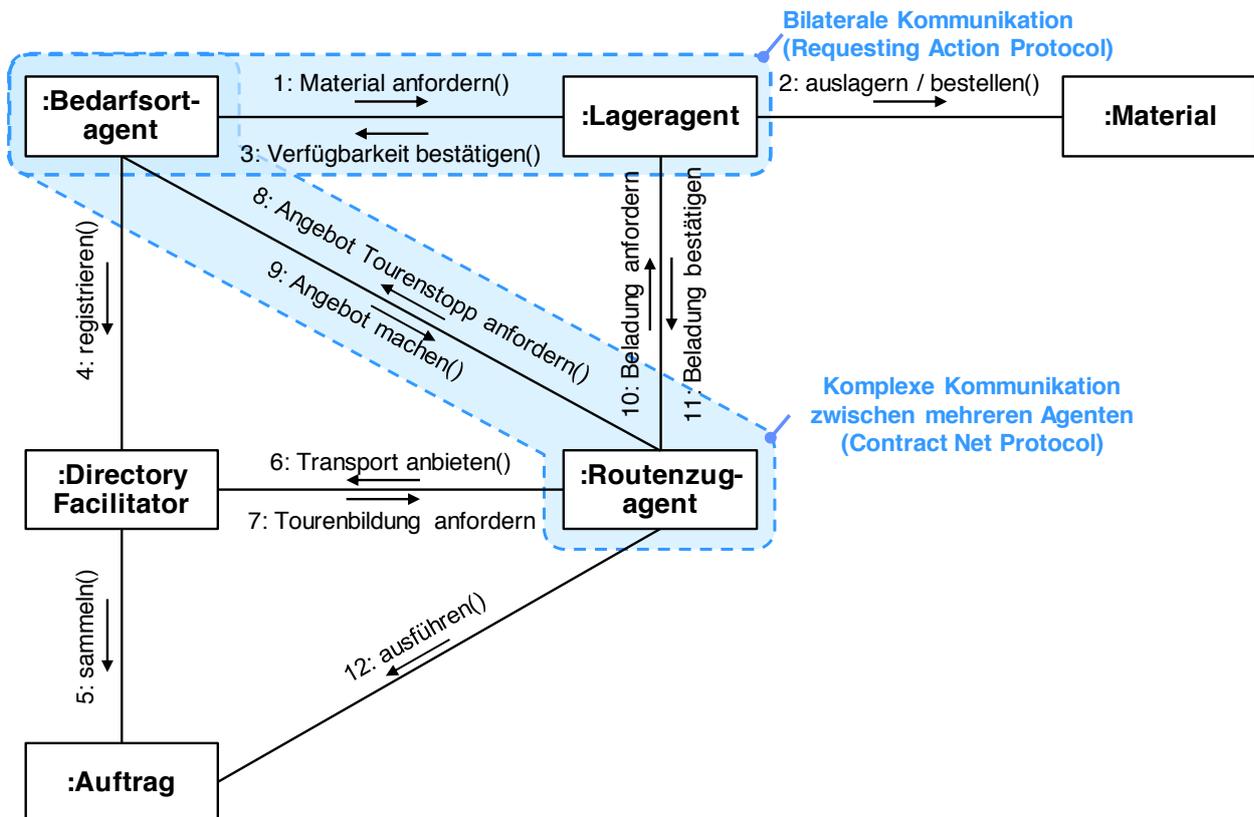


Abbildung 8-7: Kommunikationsdiagramm der Routenzugsteuerung

Die programmiertechnische Umsetzung des CNP und weitere Umsetzungsdetails werden im nächsten Abschnitt vorgestellt.

8.3 Programmierung des Agentensystems

Als letzter Schritt vor der Simulation des betrachteten Anwendungsfalls muss das im vorherigen Abschnitt konzipierte Agentensystem programmiert werden. In Abschnitt 6.3.1 wurde dafür die kommerzielle Simulationssoftware AnyLogic als Agentenplattform ausgewählt. Im Folgenden werden die Umsetzungsdetails der definierten Agenten und ihrer Interaktionen beschrieben. Die vergleichsweise einfachen Lageragenten und der Directory Facilitator wird in Abschnitt 8.3.1 vorgestellt. In den darauffolgenden Abschnitten 8.3.2 und 8.3.3 werden die komplexeren Routenzug- und Bedarfsortagenten samt ihrer Zustandsdiagramme diskutiert. Im Abschnitt 8.3.4 werden die in die Systemarchitektur eingebetteten Ein- und Ausgabedateien sowie die konkreten Interaktionen in Form eines umfassenden Sequenzdiagramms beschrieben.

8.3.1 Umsetzung von Lageragent und Directory Facilitator

Die zwei Agententypen LA und DF unterscheiden sich von den übrigen Agenten dadurch, dass sie während der gesamten Simulation bestehen und nur einfache Funktionen ausführen. Daher benötigen sie weder eine Parametrisierung noch klassische Zustandsdiagramme. Im Programm

sind sie daher als einzelne Agenten im Main-Agenten implementiert, der in AnyLogic für die Steuerung übergreifender Prozesse verantwortlich ist.

Lageragent

Der Lageragent erfüllt drei Funktionen: den Bestand nach bestelltem Material durchsuchen, fehlendes Material nachbestellen und die Auslagerung von Behältern durch Staplerfahrer beauftragen (vgl. Abschnitt 8.2.2). Damit er diese Funktionen ordnungsgemäß ausführen kann, steht ihm eine Bestandsliste in Form einer einfachen CSV-Datei zur Verfügung. Sie reflektiert den Zustand des Lagers zu jedem Zeitpunkt der Simulation und ergibt sich aus den freien und belegten Lagerplätzen. Fehlendes Material wird nachbestellt und beim Eintreffen in die Liste eingetragen. Für die Auslagerung des Materials ist ein FIFO-Prinzip implementiert, um eine möglichst kurze Lagerdauer zu erzielen. Im Anschluss an die Auslagerung wird die Bestandsliste ebenfalls aktualisiert. Der LA ist ein nutzenbasierter Agent.

Directory Facilitator

Der DF ist als zentrale Anlaufstelle für die übrigen Agenten in der Simulation für mehrere Funktionen verantwortlich: Er registriert eingehende Aufträge von BOAs und übermittelt sie an RZAs sobald eine ausreichende Anzahl vorhanden sind. Zur Maximierung der Routenzugauslastung wurde dieser Mechanismus so implementiert, dass die Tourenbildung erst erfolgt, sobald eine Mindestanzahl an Aufträgen vorliegt. Der DF muss aber auch Informationen zu den Fahrzeiten zwischen Bedarfsorten und dem Routenzugbahnhof zur Verfügung stellen, weshalb er über eine Distanzmatrix verfügt. Diese wurde auf Basis der durchschnittlichen Fahrzeiten zwischen den Bedarfsorten ermittelt und dem DF als einfaches Tabellenkalkulationsblatt zur Verfügung gestellt. Da der DF lediglich einfache Anweisungen befolgt, die durch externe Impulse ausgelöst werden ist er als einfacher Reflexagent zu klassifizieren.

8.3.2 Umsetzung von Routenzugagenten

Für jeden Routenzug im System wird ein eigener RZA instanziiert, der über die gesamte Simulationsdauer besteht und sowohl über dezidierte Parameter als auch über ein spezifisches Zustandsdiagramm verfügt. Die Parameter setzen sich aus den Eigenschaften „Kapazität“, „Zugtyp“ (Links-/Rechtsentlader), „ID“, „Beladungszeit am Routenzugbahnhof“ und „Behälterwechselzeit“ zusammen. Das Zustandsdiagramm eines RZA stellt einen geschlossenen Kreislauf aus den Zuständen „am Bahnhof“, „Tour zugewiesen“, „wird beladen“, „auf Fahrt“ und „nicht benötigt“ dar (vgl. Abbildung 8-8, links). Als Auslöser werden dabei in Abhängigkeit der Zustände entweder Zeittrigger oder Nachrichten verwendet. Am Bahnhof angekommen fragt der RZA neue Aufträge vom DF ab und bildet auf Basis des zuvor gewählten CNP eine Tour mit Aufträgen für jeden Anhänger. Dabei verfolgt er das Ziel, die Bedarfsorte möglichst kostengünstig zu beliefern und wählt dafür das beste Angebot der BOAs aus, weswegen es sich um einen nutzenbasierten Agenten handelt. Sollte der DF nicht über ausreichend Aufträge für den RZA verfügen, tritt der Zustand

„nicht benötigt“ ein. Ist dies der Fall, erfolgt die nächste Anfrage durch den RZA nach Ablauf einer Wartezeit von einer Minute.

8.3.3 Umsetzung von Bedarfsortagenten

Die BOAs unterscheiden sich von den übrigen Agententypen dadurch, dass sie nur temporär für die Dauer des bestehenden Materialbedarfs existieren. Sie werden instanziiert, wenn ein Bedarf auf Basis einer ebenfalls CSV-basierten Bedarfsliste entsteht, die dem Main-Agenten vorliegt. Nachdem die Belieferung an den Bedarfsort ausgeführt wurde, wird der BOA terminiert. Dafür ist eine Agentenpopulation mit vererbaren Eigenschaften implementiert. Jeder BOA der Agentenpopulation enthält die Parameter „Kante“ und „Ortstyp“, mit denen die Lage und die Belieferungsseite des Bedarfsorts bestimmt wird, eine „Sachnummer“ und einen „Lagerort“ für das benötigte Material sowie eine „Wunschzeit“ und eine „Auftragsnummer“ für die Belieferung. Das Zustandsdiagramm stellt hier keinen geschlossenen Kreislauf, sondern einen sequenziellen Prozess mit definiertem Anfang und Ende dar (vgl. Abbildung 8-8, rechts). Bei Instanziierung haben BOAs den Ausgangszustand „Bedarf offen“ und fordern den LA zur Prüfung der Materialverfügbarkeit auf. Falls das Material nicht verfügbar ist, wird der BOA bis zum Eintreffen des Materials im Lager auf den Zustand „angehalten“ gestellt. Anschließend wird die Auslagerung beauftragt, die bis zu 60 Minuten dauern kann. Währenddessen bleibt der BOA im Zustand „wartend“. Ist das Material am Bahnhof eingetroffen wechselt er in den Zustand „am Bahnhof“ und geht frühestens 40 Minuten vor der definierten Wunschzeit in den „in Planung“-Zustand über. Dann wird der Auftrag am DF registriert und eine Verhandlung mit einem RZA beginnt. Dieser Zustand wird als „in Verhandlung“ bezeichnet. Als Angebot geben die BOAs einen gewichteten Wert unter Berücksichtigung der Lage des Bedarfsortes und der Wunschzeit ab. Falls der gewünschte Lieferzeitpunkt bereits überschritten ist, wird der Gewichtungsfaktor für die Termintreue um den Faktor 1.000 verstärkt. Aufgrund der negativen Differenz aus Wunschzeit und aktueller Modellzeit führt dies zu einem besonders niedrigen Angebot, das von RZAs präferiert wird. Wenn ein BOA keinen Zuschlag vom RZA erhält, geht er in den Zustand „in Planung“ zurück, anderenfalls geht er in den Zustand „ausgeplant“ über. Im nächsten Schritt gehen alle durch den RZA ausgewählten BOAs in den Zustand „auf Beladung wartend“ über und werden anschließend ausgeliefert. Nachdem die Aufträge abgeschlossen sind, gehen die BOAs in den Zustand „ausgeführt“ über und werden terminiert. Bei BOAs handelt es sich um zielbasierte Agenten, da sie ihren inneren Zustand selbst verwalten, über ihre Historie verfügen und das Ziel verfolgen schnellstmöglich auf einer Tour eingeplant zu werden.

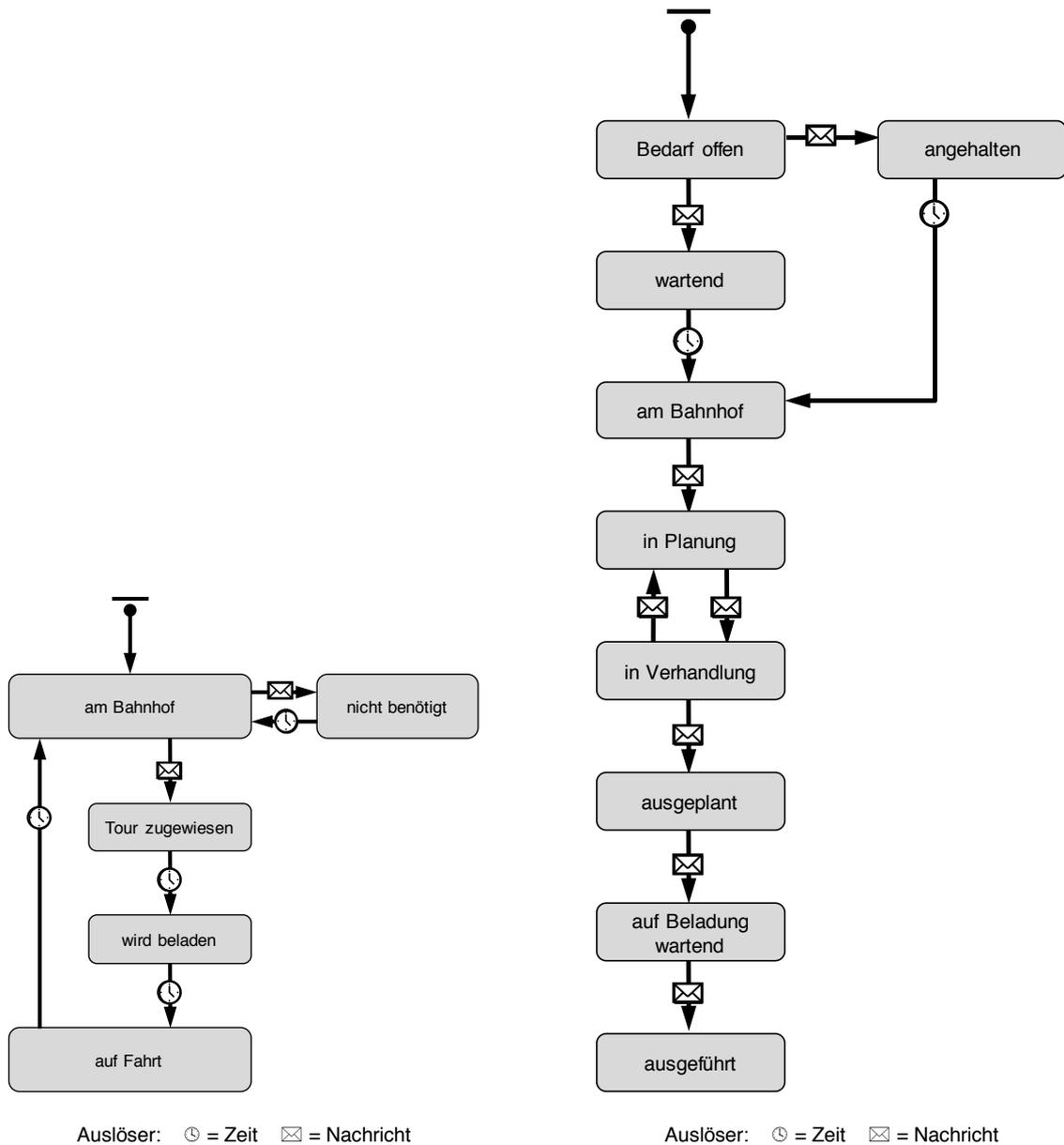


Abbildung 8-8: Zustandsdiagramm RZA (links) und BOA (rechts) i.A.a [Hab16]

8.3.4 Systemarchitektur und detaillierte Kommunikationsprozesse

Um die in den obigen Abschnitten beschriebenen Eigenschaften der Agenten umzusetzen wurde die in Abbildung 8-9 dargestellte Systemarchitektur und der in Abbildung 8-10 veranschaulichte Kommunikationsprozess implementiert. Die Systemarchitektur beinhaltet alle Ein- und Ausgabe-Größen, die für die im nächsten Schritt beabsichtigte Simulation notwendig sind.

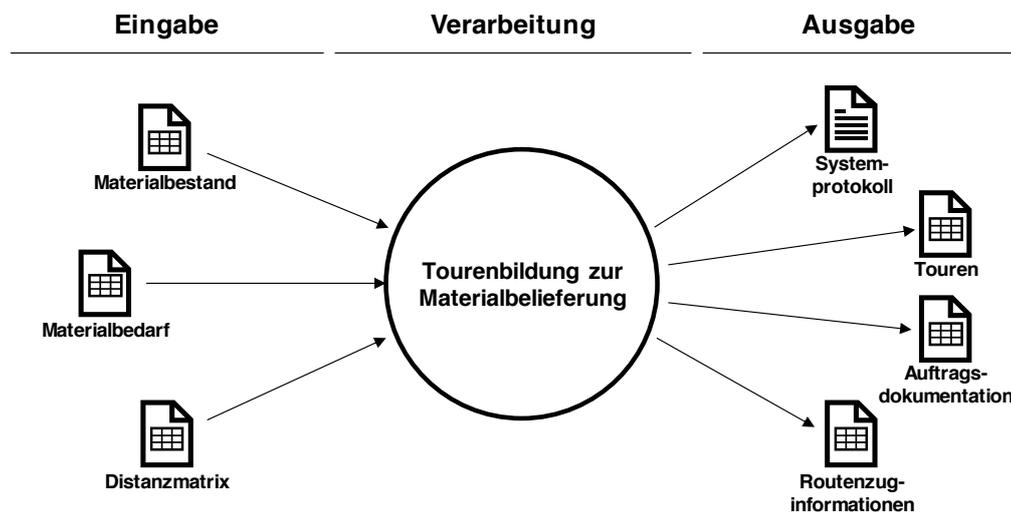


Abbildung 8-9: Ein- und Ausgangsdaten im Simulationsmodell

Die Agenten in dem System erhalten die wesentlichen Informationen aus drei Quellen, die zum Systemstart bereits verfügbar sind. Der Ausgangszustand der Lager wird über eine Materialbestandsliste repräsentiert, in der alle Sachnummern, Bestandsmengen und Lagerplätze erfasst sind. Durch Anpassung dieser Liste aktualisiert der LA den Status des Lagers in Abhängigkeit der ausgeführten Auslagerungs- und Nachbestellvorgänge. Damit wird der innere Zustand des Lagers dokumentiert. Die Abrufe hängen von dem entstehenden Materialbedarf ab, der über die Auftragsliste erfasst und gesteuert wird. Dafür wurden reale Abrufe aus dem Bedarfsmanagementsystem im Status quo exportiert und als Quelle für die BOAs gewählt. Erreicht die Simulationszeit den Zeitpunkt eines Abrufs aus der Liste, wird ein entsprechender BOA instanziiert. Gesteuert wird dies Verhalten über den in AnyLogic vorgegebenen Main-Agenten, der auch die Parametrisierung voreinstellbarer Größen ermöglicht. Als dritte Datenquelle wird eine Distanzmatrix für die Ermittlung der Fahrzeiten zwischen Bahnhof und den Bedarfsorten verwendet. Damit kann geplant werden, welcher Routenzug wann welche Materialien an die Bedarfsorte liefert.

Um die im Modell verarbeitenden Daten im Anschluss an die Simulation auswerten zu können, sind als Ausgabegrößen des Modells vier Datensenzen implementiert. Die erste Datei ist eine Protokolldatei in der alle Prozessschritte dokumentiert werden, um den ordnungsmäßigen Ablauf des Systems zu validieren. In der zweiten Datei werden die mit den ausgeführten Touren zusammenhängenden Informationen gespeichert. Dazu gehört die Fahrzeit, die Anzahl der Aufträge pro Routenzug, die Tourenplanungszeit sowie die Routenzug-ID. Auf dieser Grundlage lassen sich zahlreiche Kennzahlen bestimmen, die zur Leistungsmessung der Routenzüge verwendet werden können (vgl. Abschnitt 8.4.1). Zur Nachverfolgung der mit der Auftragsausführung zusammenhängenden Kennzahlen ist als drittes eine Datei zur Auftragsdokumentation implementiert. Darin enthalten sind alle Daten zur Ermittlung von Kennwerten bezüglich der Termintreue. Diese beinhalten den Bedarfsort, die Sachnummer des Materials, die Wunschzeit, den Zeitpunkt der Tourenplanung, des Tourenstarts und der Anlieferung sowie die Abweichung zum Wunschtermin.

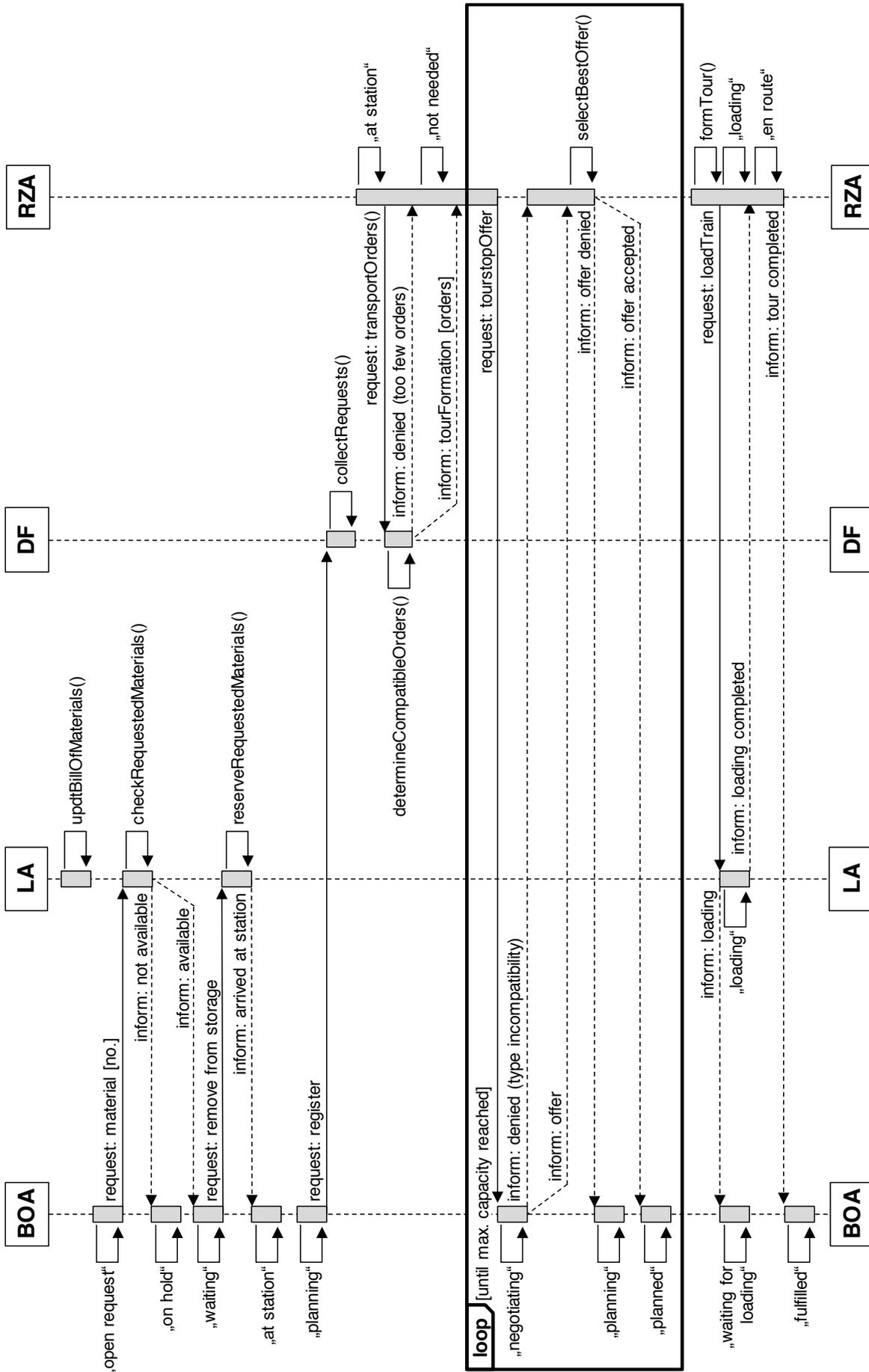


Abbildung 8-10: Agentenkommunikation zur Materialbelieferung in der RZS i.A.a. [Hab16]

Da systemtechnisch keine Bestätigung des tatsächlichen Anlieferungszeitpunkts erfolgt, wird dieser mithilfe der geschätzten Fahrzeiten ermittelt. In einer letzten Ausgabedatei werden Informationen zu den Routenzügen erfasst. Diese beinhalten den Zeitpunkt der Tourenplanung, den Beginn der Beladung sowie den Endzeitpunkt einer Tour für jeden Routenzug. Die Auswertung dieser Daten ermöglicht die Bestimmung der Zeiträume, in denen Routenzüge ungenutzt am Bahnhof stehen und gibt damit Rückschlüsse auf deren Auslastung.

Mithilfe der Informationen aus den Ein- und Ausgabedateien sind die Agenten im System handlungsfähig und eine Auswertungsmöglichkeit der Simulationsergebnisse ist gegeben. Vor diesem Hintergrund wurde das im vorangehenden Abschnitt konzipierte Kommunikationsmodell mithilfe der begleitenden studentischen Arbeit [Hab16] in Programmcode umgesetzt, sodass sich im realisierten Agentensystem der in Abbildung 8-10 dargestellte Kommunikationsfluss ergibt.

Das Programm wurde mithilfe einfacher Anwendungsfälle validiert und dient als Basis für die Durchführung der für die Überprüfung der Leistungsfähigkeit des Agentensystems erforderlichen Simulationsexperimente. Diese Experimente und die aus ihnen resultierenden Ergebnisse werden im folgenden Abschnitt diskutiert.

8.4 Simulationsexperimente

Nach der softwaretechnischen Umsetzung des Agentensystems wird analog des Vorgehens nach Law zunächst das Design der Simulationsexperimente festgelegt. In Abschnitt 8.4.1 erfolgt die Vorstellung dieses Designs, das die zugrundegelegten Annahmen, die zur Leistungsmessung angelegten Kennzahlen sowie die verwendeten Parameter beinhaltet. Im nachfolgenden Abschnitt 8.4.2 findet schließlich die Analyse und Diskussion der Simulationsergebnisse statt.

8.4.1 Design, Ausführung und Analyse der Simulationsexperimente

Mit dem vorliegenden Anwendungsfall sollen die Beobachtungen aus Kapitel 7 unterstützt werden, um damit die dritte Forschungsfrage abschließend zu beantworten. Daher gilt es auch im vorliegenden Anwendungsfall zu klären wie agentenbasierte Steuerungssysteme die Leistungsfähigkeit in komplexen Produktionssystemen verbessern können. Auch hier erfolgt daher ein Leistungsvergleich zwischen dem konventionellen System im Status quo und dem entwickelten agentenbasierten System. Während für das System im Status quo die notwendigen Daten aus den betrieblichen Systemen gewonnen werden konnten, müssen für die Leistungsmessung des Agentensystems Simulationen durchgeführt werden, für die einige Annahmen getroffen wurden.

Vereinfachende Annahmen für die Modellierung

Das entwickelte Agentensystem zur Routenzugsteuerung ist ein typisches Szenario einer realen Automobilfertigung und stellt – wie jedes Modell – eine Vereinfachung der Realität dar. Um die Komplexität des Modells dem Rahmen dieser Arbeit anzupassen wurden für die Simulation folgende vereinfachende Annahmen getroffen:

Materialauslagerung. Die Zeit, die für die Auslagerung von Behältern aus dem Lager benötigt wird, kann im Realfall variieren und folgt in Näherung einer Erlang-Verteilung. Da diese Zeiten nicht dokumentiert sind, wird für die Simulation auf Basis von Beobachtungen und Befragungen die Annahme getroffen, dass eine Auslagerung von vorhandenem Material aus dem Lager innerhalb einer Vorlaufzeit von 60 Minuten durch Stapler bzw. Palettenwagen garantiert werden kann. Dies entspricht den in Prozessaufnahmen vorgefundenen Werten.

Routenzugbeladung. Wie bereits bei der Materialauslagerung kann es auch bei der Beladung der Routenzüge mit den benötigten Behältern zu zeitlichen Fluktuationen kommen. Da auch für diesen Fall keine Realdaten vorliegen und im Normalfall eine ausreichende Anzahl an Personal und Geräten zur Verfügung steht, wird davon ausgegangen, dass eine Beladung jederzeit stattfinden kann, sobald das Material und der Routenzug im Bahnhof eingetroffen sind. Bei der Beladung wird ferner festgelegt, dass in den Angeboten, die von BOAs abgegeben werden sowohl die gewichtete Fahrzeit als auch die gewichtete Zeit bis zum Wunschtermin berücksichtigt wird. Der Auftrag mit dem niedrigsten Wert wird dem hintersten freien Anhänger zugewiesen, da die Routenzüge von hinten nach vorne entladen werden. Auf der Grundlage durchgeführter Messungen wird für die Beladung eine einheitliche Dauer von zwei Minuten pro Behälter festgelegt.

Tourenausführung. Zur Ermittlung des Tourenstarts wird die mittlere Wunschzeit der Aufträge eines Routenzugs herangezogen und mit der geschätzten Fahrzeit ergänzt. Zudem fließt die Beladungszeit in die Startzeit mit ein. Nach dem Tourenstart steuert ein Fahrer den Routenzug zu den designierten Bedarfsorten und tauscht die leeren Behälter am Bedarfsort gegen volle aus. Die Dauer dieses Tauschprozesses wird ebenfalls auf Basis von Messungen auf eine Minute festgesetzt.

Staubildung auf Fahrwegen. Die teilweise engen und nur in eine Richtung befahrbaren Fahrwegen in den Montagehallen bergen grundsätzlich das Risiko, dass sich mehrere Routenzüge stauen bzw. sich nicht gegenseitig überholen können. Auf Basis der Beobachtungen des Autors und angefertigter studentischer Arbeiten wird davon ausgegangen, dass dies ein Szenario ist, das aufgrund seiner geringen Wahrscheinlichkeit in dem betrachteten Anwendungsfall vernachlässigt werden kann.

Kennzahlen zur Leistungsmessung

Der langfristige Erfolg eines Produktionssystems wird an seiner Wirtschaftlichkeit gemessen, die sich vor allem aus Entwicklungs- und Betriebskosten zusammensetzt. Da die Entwicklungskosten langfristig aufgrund der einfachen Rekonfigurierbarkeit von Agentensystemen und der vergleichsweise aufwandsarmen Entwicklung als gegeben angenommen wird, erfolgt über die Simulation ausschließlich die Bewertung der Betriebskosten. Dabei sind die wesentlichen Kostenarten wie Personalkosten für direkte und indirekte Arbeiter, Sachgemeinkosten, Beschaffungskosten und Investitionen zu berücksichtigen. Als größter Kostenfaktor wurden anhand von Befragungen mit

Produktionsplanern die Personalkosten der Logistik- und Produktionsmitarbeiter identifiziert. Insbesondere in Ländern mit hohen Lohnkosten wie in Westeuropa machen diese direkten Personalkosten einen wesentlichen Teil der Bilanz aus. Mangelhaft koordinierte Abläufe beeinflussen diese Kostenart negativ. Verfrühte Materialanlieferungen erhöhen den Arbeitsaufwand in der Logistik, da die Routenzugfahrer bei Platzmangel die Behälterinhalte teilweise umfüllen oder auf die weitere Leerung des auszutauschenden Behälters warten müssen. Verspätet angeliefertes Material erhöht die Kosten der Produktion, da es zu einem Bandstopp und/oder zu einer Erhöhung der Nacharbeitszeiten kommen kann. Ein weiterer großer Kostenfaktor für die Wirtschaftlichkeit des Systems ist die Auslastung der Routenzüge. Wenn Routenzüge mehr Aufträge pro Tour ausliefern können bzw. weniger Fahrzeit pro Auftrag benötigen, führt dies zu einer Verbesserung der Effizienz und damit zu einer Kostensenkung. Folglich werden als Kennzahlen zur Leistungsmessung des betrachteten Routenzugsystems die in Tabelle 8-6 zusammengefassten drei Kennzahlen herangezogen.

Kennzahl	Beschreibung	Einheit
1. Fahrzeit/Auftrag	Mittlere Fahrzeit der Routenzüge pro erfolgreich ausgeliefertem Auftrag	[Min.]
2. Terminabweichung	Differenz aus Wunschzeit und tatsächlicher Anlieferungszeit des Materials am Bedarfsort	[Min.]
3. Wartezeit Routenzüge	Mittlere Zeitspanne zwischen Ankunft von Routenzügen am Bahnhof und Beginn ihrer Beladung	[%]

Tabelle 8-6: Kennzahlen für die Evaluation der Leistung des Agentensystems

Damit im Agentensystem eine gleichzeitige Optimierung der drei Zielgrößen stattfinden kann, werden sie in einer Zielfunktion zusammengefasst. Diese Zielfunktion repräsentiert den Stellenwert einer von einem BOA aufgegebenen Bedarfsmeldung mit einem Scoring-Wert und kann durch das Agentensystem daher einfach als Entscheidungsgrundlage für die Zuweisung zu Routenzuganhängern herangezogen werden. Die Zielkonflikte zwischen den Faktoren werden mithilfe von drei Gewichtungsfaktoren a , b und c berücksichtigt. Damit entsteht folgende Zielfunktion:

$$\text{Minimiere } a \cdot \frac{\text{Fahrzeit}}{\text{Anzahl Aufträge}} + b \cdot \text{Terminabweichung} + c \cdot \text{Wartezeit}$$

$$\text{mit } \{a, b, c \in \mathbb{R} \mid 0 \leq a, b, c \leq 1\} \quad (1)$$

Die Wartezeit der Routenzüge am Bahnhof wird im untersuchten Anwendungsfall im Status quo nicht im internen Transportleitsystem erfasst und ihre Bestimmung wäre nur unter erheblichem Aufwand möglich. Die Wartezeit wird daher nicht für einen Vergleich zwischen Status quo und Konzept herangezogen. Da sie jedoch für die Interpretation der Systemleistung nützlich ist wird sie im konzipierten Agentensystem gemessen.

Das Vorgehen zur Ermittlung konkreter Zahlenwerte für die Gewichtungsfaktoren sowie die Vorstellung weiterer Parameter erfolgt im nächsten Absatz.

Programm- und Prozessparameter

Die Basisdaten für das Agentensystem stammen direkt aus den Steuerungssystemen des untersuchten Routenzugsystems. Die Materialbestandsliste des LA speist sich aus einem realen Lagerverwaltungssystem, die Materialbestellungen entspringen dem Bedarfsmanagementsystem und die Routenzüge sind analog des Status quo des Anwendungsfalls parametrisiert, d.h. von den 15 vorhandenen Routenzügen sind neun vom Typ Linksentlader und sechs vom Typ Rechtsentlader. Die zur Verfügung stehenden Daten werden über die in Abschnitt 8.3.4 vorgestellten Datenquellen eingelesen und stehen dem System zur Laufzeit zur Verfügung.

Während die Parametrisierung der obigen Elemente vergleichsweise einfach ist, bedarf es für die Parametrisierung der Zielfunktion des Agentensystems mehrerer Schritte, da zunächst die optimalen Werte für die Gewichtungsfaktoren a , b und c ermittelt werden müssen. Dafür wird auf die Methode für das Design und die Analyse von Simulationsstudien nach Lorscheid et al. zurückgegriffen (vgl. Abschnitt 6.3.1). Als Resultat wurden in Summe drei Versuchsreihen entworfen, um das Modellverhalten zu untersuchen: erstens eine Effektanalyse zur Untersuchung der Einflüsse der Faktoren auf das System, zweitens die Ermittlung einer optimalen Faktorstufenkombination für die Zielfunktion und drittens die Durchführung der eigentlichen Leistungsmessung des Agentensystems.

In der ersten Versuchsreihe wurde eine **Effektanalyse** ausgewählter Einflussparameter durchgeführt, um Effekte und Wechselwirkungen kritischer Faktoren zu identifizieren. Dafür wurden die Modellvariablen zunächst in die drei in Tabelle 8-7 dargestellten Kategorien eingeteilt.

unabhängige Variablen	steuernde Variablen	abhängige Variablen
1. Gewichtungsfaktor Fahrzeit	1. Bedarfsregistrierungszeitpunkt am DF	1. Fahrzeit je Tour
2. Gewichtungsfaktor Termintreue	2. Anzahl Routenzüge	2. Anzahl Aufträge je Tour
3. Tourenstartzeit (Zeit bis zur mittleren Wunschzeit der Anlieferung)		3. Terminabweichung bei Anlieferung
4. Mindestanzahl Aufträge pro Tour		4. Wartezeit der Routenzüge

Tabelle 8-7: Klassifikation der Modellvariablen

Abhängige Variablen stellen die Ergebnisse der Simulation dar, also die mittlere Fahrzeit pro Tour, die Anzahl der transportierten Aufträge, die Terminabweichung der Anlieferung sowie die Zeit, die Routenzüge mit Warten verbracht haben. Steuernde Variablen haben einen direkten Einfluss auf die Ergebnisse der Simulation, sind jedoch für den Entwickler nicht beeinflussbar. Zu diesen Größen zählt der Zeitpunkt der Entstehung von Materialbedarfen sowie die im System vorhandene Anzahl an Routenzügen, die im vorliegenden Fall fest ist. Unabhängige Variablen hingegen haben Einfluss auf die Simulationsergebnisse und sind gleichzeitig durch den Entwickler beeinflussbar. Folglich sind für eine Effektanalyse von Bedeutung und wurden daher herangezogen, um Simulationsexperimente durchzuführen. Die dabei untersuchten Kennzahlen sind

die vier Größen Fahrzeit, Termintreue, Tourenstartzeit (als Zeit vor der mittleren Wunschzeit) und Auftragsanzahl pro Routenzug. Die Ergebnisse der Simulationsexperimente sind in Abbildung 8-11 zusammengefasst. Die Legendeneinträge der Abbildung enthalten die jeweiligen Gewichtungsfaktoren für die Größen Fahrzeit, Termintreue und den Tourenstart.

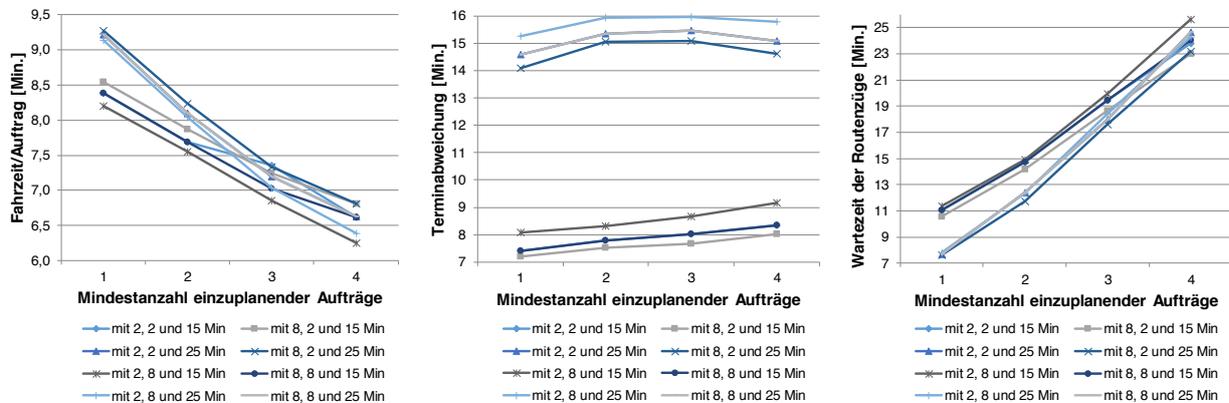


Abbildung 8-11: Effektanalyse der Zielfunktionsfaktoren i.A.a. [Hab16]

Aus der Effektanalyse wird deutlich, dass bei der Angebotserstellung der BOA die Faktoren „Tourenstartzeit“ und „Auftragsanzahl pro Routenzug“ einen stärkeren Einfluss auf die Systemleistung haben als die Gewichtungsfaktoren „Termintreue“ und „Fahrzeit“. Folglich spielen Startzeit und die Auftragsanzahl eine besondere Rolle bei der Auswahl einer geeigneten Faktorstufenkombination. Die Ergebnisse der ersten Versuchsreihe sind in Anhang 34 beigefügt.

Um die **Faktorstufenkombination** für die Gewichtungsfaktoren der Zielfunktion genau zu bestimmen wurde eine zweite Versuchsreihe durchgeführt. Ziel war es dabei die Gewichtungsfaktoren so zu wählen, das für ein beliebiges Produktionsprogramm bestmögliche Zielwerte erreicht werden. Mit den vier oben genannten unabhängigen Variablen wurden auf zwei bzw. vier Stufen insgesamt 32 Simulationsdurchläufe durchgeführt, um die Zielgrößenwerte für alle Faktorstufenkombinationen zu ermitteln. Folglich handelt es sich um einen vollfaktoriellen Versuchsplan. Zur Optimierung der Auslastung wird die Fahrzeit im Modell höher priorisiert als die Wartezeit, da letztere keinen Auftragsbezug hat. Noch höher ist die Priorität des Faktors Terminabweichung zu bewerten, da von ihr nicht nur die Produktivität des Routenzugsystems selbst, sondern auch die der Produktionslinien abhängig ist. Auf Basis der dargelegten Analysen wurden daher für die drei Gewichtungsfaktoren folgende Werte festgelegt:

- Gewichtungsfaktor für Fahrzeit: $a = 0,3$
- Gewichtungsfaktor für Terminabweichung: $b = 0,6$
- Gewichtungsfaktor für Wartezeit: $c = 0,1$

Um die Fahrzeit pro Auftrag zu reduzieren und die Auslastung der Routenzüge weiter zu erhöhen wird die Mindestanzahl der Aufträge pro Tour auf vier gesetzt. Ferner werden die Touren zur Steigerung der Termintreue 10 Minuten vor der mittleren Wunschzeit gestartet, da mit diesem

Wert die besten Ergebnisse erzielt werden (vgl. Anhang 35). Damit sind alle Parameter des Modells festgelegt, sodass zur dritten und letzten Versuchsreihe übergegangen werden kann.

In der dritten Versuchsreihe findet die tatsächliche **Leistungsmessung** des Agentensystems statt. Für die Durchführung von Simulationsläufen mit dem Agentensystem wurden reale Produktionsdaten einer repräsentativen Arbeitswoche mit fünf Arbeitstagen verwendet. D.h. jeder im Modell simulierte Tag besitzt ein eigenes Fahrzeugproduktionsprogramm mit unterschiedlichen Materialbedarfen und Lagerbeständen. Die Ergebnisse der Simulationsläufe sind Gegenstand des nächsten Abschnitts.

8.4.2 Simulationsergebnisse und kritische Beurteilung

Das programmierte und parametrisierte Simulationsmodell wurde mit den vorhandenen realen Produktionsdaten für die fünf Arbeitstage gespeist und es wurden mit jedem Datensatz separate Simulationsläufe durchgeführt. Die Ergebnisse dieser Experimente sind in Anhang 36 dokumentiert. Zur weiteren Auswertung wurden die Leistungsdaten des Agentensystems mit denen des Status quo verglichen. Die Gegenüberstellung des agentenbasierten (MAS) und des konventionellen Ansatzes (Status quo) zeigt, dass das Agentensystem an allen fünf untersuchten Arbeitstagen für alle Kennzahlen bessere Ergebnisse erzielt. Die Unterschiede zwischen den Ergebnissen der zwei Systeme sind in Abbildung 8-12 und 8-13 visualisiert.

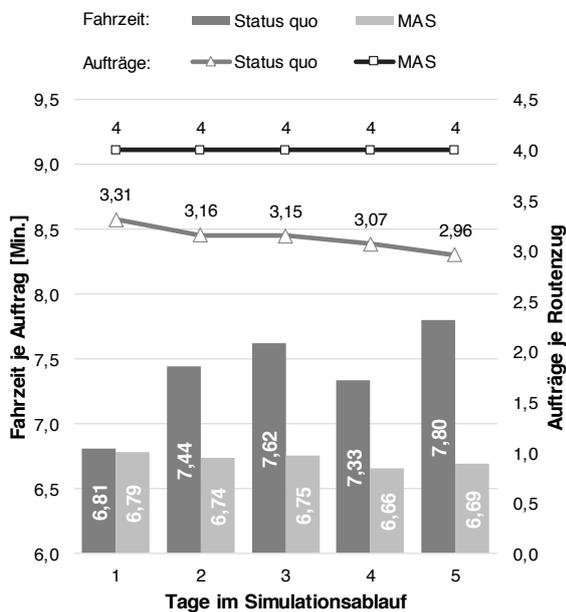


Abbildung 8-12: Gegenüberstellung Fahrzeit/Auftrag und Aufträge/Tour

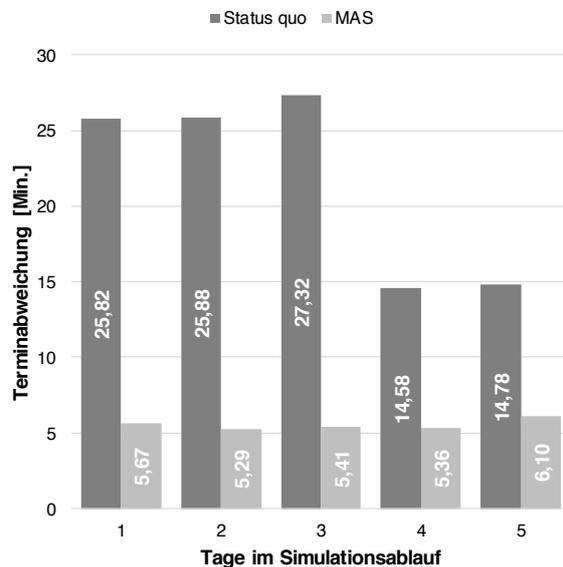


Abbildung 8-13: Gegenüberstellung Terminabweichung

In der linken Abbildung sind die beiden voneinander abhängigen Kennzahlen Fahrzeit pro Auftrag und Auftragsanzahl pro Routenzug dargestellt. Das Agentensystem erreicht an allen untersuchten Tagen kürzere Fahrzeiten und eine höhere Anzahl Aufträge pro Routenzug als das konventionelle System. Dies liegt unter anderem daran, dass im Agentensystem eine Tour erst startet, wenn ein Routenzug über vier Aufträge verfügt und damit voll beladen ist. Da der Tourenstart im

konventionellen System hingegen mit weniger Aufträgen starten kann, ist die Routenzugauslastung im Status quo geringer. Pro Routenzug wird im Mittel etwa ein Auftrag weniger transportiert als mit dem Agentensystem, was in einem Leistungsdefizit von knapp 25% resultiert. Auch bei der Terminabweichung erzielt das Agentensystem deutlich bessere Ergebnisse als das konventionelle System: Die in der rechten Darstellung zusammengefassten Simulationsergebnisse zeigen, dass die mittlere Terminabweichung mit dem agentenbasierten System deutlich geringer ist. Ein direkter Vergleich der Mittelwerte ist in Tabelle 8-8 abgebildet.

	Fahrzeit/Auftrag			Aufträge/Tour			Terminabweichung		
	Status quo	MAS	Δ	Status quo	MAS	Δ	Status quo	MAS	Δ
\bar{x}	7,40 Min.	6,73 Min.	-0,67 Min. (x 1,1)	3,13 Stk.	4,00 Stk.	+0,87 Stk. (x 1,3)	21,68 Min.	5,57 Min.	-16,11 Min. (x 3,9)
σ (abs.)	0,34 Min.	0,04 Min.	-0,29 Min. (x 7,5)	0,12 Stk.	0,00 Stk.	-0,12 Stk. (x ∞)	5,74 Min.	0,30 Min.	-5,44 Min. (x 19,2)
σ (rel.)	4,54%	0,66%	-3,87% (x 6,8)	3,68%	0,00%	-3,68% (x ∞)	26,46%	5,35%	-21,11% (x 4,9)

Tabelle 8-8: Gegenüberstellung Agentensystem und Status quo

Aus der Tabelle wird ersichtlich, dass die Fahrzeit je Auftrag und die Auftragsanzahl je Routenzug mit dem Agentensystem um den Faktor 1,1 respektive 1,3 verbessert werden konnte. Die Terminabweichung konnte sogar um den Faktor 3,9 reduziert werden. Dies konnte vor allem aufgrund der Verwendung dynamischer Routen erreicht werden, da sie im Gegensatz zu den statischen Fahrkreisen im ursprünglichen System ermöglichen die Bedarfsorte deutlich flexibler zu beliefern. Darüber hinaus ist anhand der Kennzahlen aus der Tabelle auch eine Bewertung der Robustheit der beiden Systeme möglich, indem die Standardabweichung als Bewertungsmaß hinzugezogen wird. Auch hier wirken sich die Eigenschaften des Agentensystems positiv auf die Leistungskennzahlen aus: Die feste Auftragsanzahl pro Routenzug führt in Verbindung mit den dynamischen Routen zu konstanteren Fahrzeiten und weniger Ausschlägen bei der Terminabweichung (Faktor 6,8 bzw. Faktor 4,9).

In Summe werden mit dem Agentensystem also einerseits die Leistungskennzahlen des Routenzugsystems verbessert und andererseits die Unregelmäßigkeiten geglättet. Ein über diese Produktivitätsverbesserung des Systems im Betrieb hinausgehender Vorteil des Agentensystems ist, dass ein großer Teil des Entwicklungs- und Planungsaufwands reduziert wird. Im Gegensatz zu einem herkömmlichen Steuerungssystem, passt sich das Agentensystem flexibel der Situation in der Produktion an und bedarf z.B. bei Veränderungen der Entfernungen von Bedarfsorten, der Anzahl verfügbarer Routenzüge, der Vergrößerung oder Verkleinerung des Hallenlayouts etc. keiner erneuten Ausplanung durch Spezialisten. Stattdessen adaptiert sich das Agentensystem größtenteils selbstständig. Ein weiterer Vorteil der eingesetzten Agententechnologie ist, dass für die Umsetzung in der Praxis keine hohen Anschaffungskosten notwendig sind, da es zur Sicherstellung der Funktionsfähigkeit keiner weiteren Technologien wie z.B. RFID bedarf.

Bei allen Stärken des Systems existieren auch Schwachpunkte. So wurden in den Simulationen trotz der im Mittel deutlich kleineren Terminabweichung vereinzelt Behälter erst nach dem Wunschtermin angeliefert. Da die Fahrzeugendmontage nach dem Zwei-Behälter-Prinzip organisiert ist, führten die Verspätungen zwar in keinem Fall zu einem Produktionsstillstand. Dennoch sollten verspätete Lieferungen grundsätzlich vermieden werden und bedürfen daher vorbeugender Maßnahmen. So könnte z.B. ein durch die erhöhte Auslastung des Systems freigewordener Routenzug verwendet werden, um zeitkritische Aufträge gesondert zu transportieren oder es könnte die Bedingung implementiert werden, dass ein Routenzug losfahren darf bevor er voll beladen ist. Dies wäre ein akzeptabler Trade-off, der eine niedrigere Auslastung zugunsten einer verbesserten Termintreue erreicht.

Darüber hinaus existieren zahlreiche Anknüpfungspunkte zur Verbesserung und Weiterentwicklung des Agentensystems, die in weiteren Forschungsprojekten realisiert werden können. Durch die Anbindung des Staplerleitsystems und die Berücksichtigung von Staus könnte die Leistung des Agentensystems weiter gesteigert werden. Die bisher starre Zuordnung von Material zu spezifischen Lagerplätzen könnte in einem nächsten Entwicklungsschritt ebenfalls aufgehoben und durch eine dynamische bzw. chaotische, durch Agenten gesteuerte Lagerplatzbelegung ersetzt werden. Um das System zudem den realen Bedingungen weiter anzupassen, könnten z.B. Strafzeiten/-kosten für stark befahrene Strecken eingeführt und Echtzeit-Positionsdaten einbezogen werden. Damit könnte die Genauigkeit der Tourenplanung erhöht sowie eine Verbesserung der geschätzten Ankunftszeit von Material an Bedarfsorten und Routenzügen am Bahnhof erreicht werden. Möglich ist dies z.B. über Warehouse-GPS-Systeme. Um Hardware-Investitionen niedrig zu halten bieten sich aber auch Indoor-Localization-Technologien an, die auf Basis des in dieser Arbeit verwendeten JADE-Frameworks funktionieren und keine dezidierte Infrastruktur benötigen [MoB17]. Die reale Umsetzung des Systems sollte zunächst im kleinen Rahmen geschehen und eine Notfallstrategie beinhalten, falls unvorhergesehenes Verhalten auftritt. Dafür kommen z.B. kleine, unabhängig belieferte Bandabschnitte bzw. Kleinserien-Montagen in Frage. Weitere Informationen zu dem Anwendungsfall können in [Hab16], [ZHL17] und [ZaL17] nachgelesen werden.

Zusammenfassend wurde mit den beiden Anwendungsfällen aus Kapitel 7 und 8 gezeigt, dass Agentensysteme sich für die Anwendung in der industriellen Fertigung eignen und gerade in störungsanfälligen Systemen Flexibilität und Robustheit bieten können. Darüber hinaus wurde aufgezeigt welche architektonischen Merkmale die Agentensysteme vorweisen müssen und wie diese konkret umgesetzt werden können. Damit ist die Forschungsfrage drei im Sinne der Anforderungen dieser Arbeit beantwortet. Im nächsten Kapitel wird daher der Schwerpunkt auf die Beantwortung der finalen Forschungsfrage gelegt, die sich auf die Erarbeitung einer Methode zur Integration von interdependenten Agentensystemen bezieht.

9 AOSE-Methode für interdependente Agentensysteme

Komplexe Automatisierungssysteme wie sie heute in großen Industrieanwendung vorliegen, können nicht in einem Schritt entwickelt werden. Nach der erfolgreichen Umsetzung der individuellen Agentensysteme in Kapitel 7 und 8 verbleibt zur Beantwortung der letzten Forschungsfrage dieser Arbeit daher zu klären, auf welche Weise solche Systeme, die Teil eines größeren Metasystems wie einer Fabrik oder eines Produktionsnetzwerks sind, integriert werden können. Ziel dieses Kapitels ist es daher eine agentenbasierte Softwareentwicklungsmethode zu definieren, mit der zwei oder mehr interagierende Agentensysteme über ein schrittweises Vorgehen miteinander vernetzt werden können. In Abschnitt 9.1 wird zunächst auf die Anforderungen und die Zielsetzung der Integration eingegangen. Um sicherzustellen, dass interagierende Agentensysteme ihre Funktion jederzeit erfüllen und sowohl ihre individuellen als auch gemeinsamen Ziele bestmöglich erreichen, werden im daran anknüpfenden Abschnitt 9.2 Herausforderungen diskutiert, die bei der Integration zu berücksichtigen sind. Mit diesen notwendigen Vorarbeiten als Basis erfolgt in Abschnitt 9.3 schließlich die Entwicklung der geforderten Entwicklungsmethode. Als Grundlage wird die in Abschnitt 4.2.5 beschriebene und in zahlreichen Projekten erfolgreich eingesetzte DACS-Methode verwendet. Die Überprüfung und Bewertung der entwickelten Methode findet anschließend in Kapitel 10 anhand der Anwendungsfälle aus den vorangehenden Kapiteln statt.

9.1 Anforderungen und Zielsetzung der Integration

Die Größe und Komplexität heutiger Produktionssysteme macht es erforderlich bei der Entwicklung der Steuerungsarchitektur modular vorzugehen. Dadurch wird einerseits die Komplexität der Entwicklungsarbeit reduziert und andererseits die Parallelisierung von Entwicklungsprozessen ermöglicht. Dies führt schließlich zu einer Beschleunigung der Entwicklung und Umsetzung der Produktionssysteme, die für die Wirtschaftlichkeit von Unternehmen überlebenswichtig ist [LoC07]. Für die Entwicklung und Implementierung von Agentensystemen für abgeschlossene Problembereiche von Produktionssystemen existieren zahlreiche Methoden und Architekturen, die in Kapitel 4 bereits ausführlich diskutiert wurden. Insbesondere die in dieser Arbeit verwendete DACS-Methode eignet sich für den Automatisierungsbereich sehr gut und bietet ein universell auf Steuerungsprobleme übertragbares Vorgehen [ZaL].

Dabei ist jedoch zu beachten, dass sich fast alle beschriebenen Entwicklungsmethoden und Steuerungsarchitekturen einerseits nur auf eine begrenzte Anzahl von Steuerungsentscheidungen innerhalb eines gesamten Produktionssystems konzentrieren und andererseits auf ähnlichen Strukturen beruhen. In realen Anwendungsfällen sind Steuerungssysteme jedoch häufig lediglich Subsysteme eines größeren Produktionssystems, die zusammen an der Herstellung eines einzigen, gemeinsamen Produktes, wie z.B. einer komplexen Anlage, eines Automobils oder eines Smartphones, arbeiten. Folglich sind die Steuerungsaufgaben in den Subsystemen nicht unabhängig

voneinander, sondern bedingen sich gegenseitig. Die Subsysteme können dabei unterschiedliche, teilweise sogar konfliktäre Steuerungs- oder Optimierungsziele verfolgen, sodass – aus ganzheitlicher Sicht fertiger Unternehmen – eine Integration der Subsysteme ein erstrebenswertes Ziel ist, für das die derzeitigen Entwicklungsmethoden keine adäquate Lösung bieten.

Am Beispiel der unterschiedlichen Gewerke der Automobilproduktion kann die Notwendigkeit der Integration von Subsystemen anschaulich gemacht werden: Die in Kapitel 6 vorgestellte Abbildung 6-1 zeigt auf einer hohen Aggregationsebene einige der miteinander in Beziehung stehenden Steuerungsaufgaben. Sie reichen von der Planung von Losgrößen für Bauteile im Presswerk, über die Karosseriefolgesteuerung für den Rohbau und die Lackiererei, bis hin zur Betriebsmittelsteuerung in der Montage. Dabei ist offensichtlich, dass die Steuerungsentscheidungen nicht unabhängig voneinander sind. Der einmal freigegebene Kundenauftrag im Karosseriebau durchläuft schrittweise alle Stationen der Fertigung und hat damit Auswirkungen auf alle Ebenen der nachfolgenden Gewerke. Das beinhaltet die Losgrößen für die Lackierung ebenso wie JIT-Abrufe von internen und externen Materialien bis hin zur Ausführung der Transportlogistik. Umgekehrt haben die Systeme der späteren Wertschöpfungsschritte auch Einfluss auf die frühen Phasen: Störungen in Lackier- oder Montageeinrichtungen, fehlendes Material und Lieferprobleme von Zulieferern beeinträchtigen die im Karosseriebau freizugebenden Aufträge und können zu Reihenfolgeänderungen führen.

In den Kapiteln 7 und 8 wurde bereits aufgezeigt, dass die individuellen Steuerungsaufgaben der Subsysteme gut mit Agentensystemen implementiert werden können. Klassischer Weise werden interdependente Steuerungssysteme über Schnittstellen miteinander verbunden. Dabei werden in den meisten Fällen die Ausgangsdaten eines Systems als Eingangsdaten für das in der Wertschöpfung folgende System übergeben, wie z.B. die Karosseriefolge beim Einlauf in die Montagehalle. Die Abhängigkeiten zwischen den Subsystemen werden dabei jedoch zu einem Großteil vernachlässigt, sodass die Einzelsysteme lediglich ihre internen Ziele verfolgen. Dies geht vor dem Hintergrund der zunehmenden Komplexität und Notwendigkeit der Integration nicht weit genug. Es besteht folglich ein zunehmender Bedarf, bestehende Steuerungsarchitekturen zu verbinden.

Im Bereich von Agentensystemen existieren grundsätzlich zwei Wege für den Informationsaustausch zwischen Steuerungssystemen: Erstens, über Kommunikationsschnittstellen mit denen Telegramme zwischen zwei Systemen ausgetauscht werden können (z.B. über Sockets mit einem TCP/IP-Protokoll). Zweitens, über gemeinsame Datenbanken, auf die alle voneinander abhängige Systeme zugreifen können (z.B. über eine standardisierte SQL-Schnittstelle wie ODBC) [DaC10]. Eine Alternative zur Lösung des Schnittstellenproblems kann im Einsatz dezidierter Schnittstellenagenten bestehen, die für den Austausch und die Verfügbarmachung kritischer Informationen verwendet werden. Sie können als Kombination der nachrichten- und datenbankba-

sierten Schnittstellen aufgefasst werden und entsprechen Adaptern, die externe Nachrichten verarbeiten und den Datenaustausch mit benachbarten Systemen ermöglichen [LCL10]. Es bleibt jedoch unklar wie die Abhängigkeiten zwischen den Steuerungsentscheidungen der Einzelsysteme berücksichtigt werden können, damit die Subsysteme miteinander kooperieren und ein gemeinsames Optimum verfolgen, anstatt wie bisher individuelle Ziele. Da die bekannten Entwicklungsmethoden diese Frage nicht in ausreichender Tiefe beantworten, ist es das Ziel nachfolgend eine im Produktionsumfeld anwendbare AOSE-Methode zu entwickeln, mit der diese Lücke geschlossen werden kann. Dabei müssen mehrere Anforderungen erfüllt werden.

Grundsätzliche Voraussetzung für die zu entwickelnde AOSE-Methode ist, dass sie in einfacher Weise von Automatisierungstechnikern angewendet werden kann und eine schrittweise Migration bestehender Steuerungssysteme hin zu Verbänden interagierender Agentensysteme, z.B. analog des Vorgehens in [CLC17], ermöglichen. Um zudem die Komplexität und den Aufwand bei der Entwicklung und Umsetzung der Systeme zu minimieren, sollte die Methode modular aufgebaut sein und ein iteratives Vorgehen ermöglichen. Folglich sollte es möglich sein die verschiedenen zu lösenden Steuerungsprobleme separat zu identifizieren und individuelle Agentensysteme für sie zu entwickeln. Um die Komplexität des zu lösenden Gesamtproblems bei der Integration handhabbar zu machen, sollte auch für das Zusammenführen der resultierenden Einzelsysteme ein schrittweises Vorgehen möglich sein. Auf diese Weise kommen bei jeder Iteration neue Informationen hinzu, was die Transparenz und das Verständnis zum Verhalten des Gesamtsystems erhöht und die Bewältigung der Integrationsaufgabe erleichtert. Zur Sicherstellung des ordnungsgemäßen Verhaltens des resultierenden Gesamtsystems sollten in jeder Iteration sowohl die individuellen Zielsetzungen der Subsysteme als auch deren gemeinsame, globale Ziele berücksichtigt werden. Folglich ist es notwendig, dass die individuellen Stellhebel der Subsysteme zur Erreichung dieser Ziele und die Abhängigkeiten zwischen den Subsystemen ebenfalls identifiziert werden. Auf diese Weise soll sichergestellt werden, dass die Integration der unterschiedlichen Zielsetzungen der Subsysteme zur Erreichung eines globalen Optimums des gesamten Produktionssystems beiträgt. Ferner gilt es die folgenden Herausforderungen bei der Entwicklung zusammenwirkender Agentensysteme zu berücksichtigen.

9.2 Herausforderungen beim Design interdependenter Systeme

Das Zusammenspiel interdependenter Steuerungssysteme ist mit Risiken verbunden, die es zu reduzieren gilt. Zu den entscheidenden Risiken gehört dabei die Möglichkeit von Schneeballeffekten sowie die mangelnde Berücksichtigung unterschiedlicher Zielstrukturen, die nachfolgend umrissen werden.

9.2.1 Schneeballeffekte in Agentensystemen

Der potenzielle Nutzen zusammenwirkender Agentensysteme wird bereits lange diskutiert [Jam97], den damit einhergehenden Risiken wird jedoch erst in jüngster Vergangenheit zunehmend Bedeutung beigemessen (vgl. z.B. [MCN17]). Ein wesentliches Risiko ist dabei, dass sich die Entscheidungen von Agenten unterschiedlicher Systeme derart aufeinander auswirken, dass es zu einer Kettenreaktion oder einem gegenseitigen Aufschaukeln kommt [LWF13, WLF14]. Ein solcher Effekt – auch Schneeballeffekt genannt – könnte die Leistung des Systems negativ beeinflussen oder es sogar ganz zum Stillstand bringen.

Als Beispiel für einen solchen Effekt sei ein Ereignis aus dem Finanzbereich herangezogen – der Flash Crash 2010. Die Krise wurde durch den Verkauf einer großen Menge von Futures-Kontrakten durch einen algorithmisch gesteuerten Hochfrequenzhändler ausgelöst. Da es zeitweise an Käufern mangelte, wurden die Kontrakte an andere algorithmische Händler weiterverkauft, die wiederum an weitere algorithmische Händler verkauften usw. Dieser Effekt führte zu einem stark zunehmenden Handelsvolumen, das wiederum den Verkaufsalgorithmus veranlasste die Verkaufsrate zu erhöhen und die Abwärtsspirale weiter voranzutreiben. Diese Spirale wurde erst durch einen automatischen Abschaltmechanismus gestoppt. Da sich die Preise im Anschluss an diesen nur fünf Sekunden andauernden Vorgang wieder stabilisierten, konnte ein großer Teil der Verluste wieder gut gemacht werden. Zeitweise führte diese Kettenreaktion jedoch zu einer Wertvernichtung in Höhe von einer Billion Dollar. Anhand dieses Beispiels wird deutlich, dass für sich genommen einfache individuelle Komponenten zu komplizierten und unerwarteten Effekten führen können. Zudem ist festzuhalten, dass die Automatisierung zwar zu dem Vorfall beigetragen hat, aber auch zu seiner Lösung. Folglich wird in solchen Systemen eine vorinstallierte und automatisch ablaufende Sicherheitsfunktionalität benötigt, da die menschliche Reaktionsfähigkeit nicht ausreicht, um die schnell ablaufenden Prozesse in solchen Systemen zu überwachen [Bos14]. Ein bekanntes Beispiel für einen ähnlichen Effekt im Bereich der Logistik ist der sogenannte Bullwhip-Effekt, bei dem die Bestellungen entlang der Lieferkette zu größeren Schwankungen neigen als Bestellungen von Endkunden [LPW97]. Die Möglichkeit solcher Ereignisse ist auch bei interagierenden Agentensystemen zu berücksichtigen.

Vor diesem Hintergrund wurden von Wassermann und Fay folgende vier Regeln für das Zusammenwirken von Agentensystemen definiert. Erstens müssen die Aussagen von Agenten wahrhaftig sein, damit sich andere Agenten darauf verlassen können, dass zugesagte Aktionen auch ausgeführt werden. Zweitens müssen die von Agenten eingegangenen Verpflichtungen eingehalten werden und ihre Zusage muss für einen bestimmten Zeitraum Geltung haben, damit andere Agenten ihre Entscheidungen auf dieser Grundlage treffen können. Eine Neubewertung der Entscheidungen muss also für einen gewissen Zeitraum verboten sein. Drittens muss die Aktivität von Agenten beschränkt werden, z.B. indem sie eine Obergrenze erhält. Dadurch soll vermieden werden, dass die Anfragen sehr eifriger Agenten nicht zu einer Überlastung des Systems führen.

Schließlich muss als viertes auch der umgekehrte Fall vermieden werden, dass Agenten zu langsam oder überhaupt nicht antworten, da dies zu einem Stillstand des Systems führen könnte. Diese Regeln sollten nicht nur in die Designphase einfließen, sondern auch mithilfe von Unit Tests und Laufzeitverifizierungen in der Umsetzungsphase überprüft werden [WaF17].

Um die Gefahr von Schneeballeffekten von vornherein auszuschließen bzw. zu reduzieren, bedarf es daher zwangsweise einer Analyse der Steuerungsentscheidungen und ihrer Interdependenzen. Dabei sollten iterativ die Auswirkungen von Entscheidungen der involvierten Agenten auf die Entscheidungen der übrigen Beteiligten analysiert und bewertet werden, um Aufschaukelungseffekte zu vermeiden.

9.2.2 Integration von Zielstrukturen

Bei dem Entwurf interagierender Agentensysteme sind des Weiteren auch die Zielstrukturen und die damit zusammenhängende Systemarchitektur zu berücksichtigen. Dabei können grundsätzlich die zwei in Abbildung 9-1 dargestellten Formen von Zielstrukturen unterschieden werden: (a) hierarchische und (b) heterarchische Zielstrukturen.

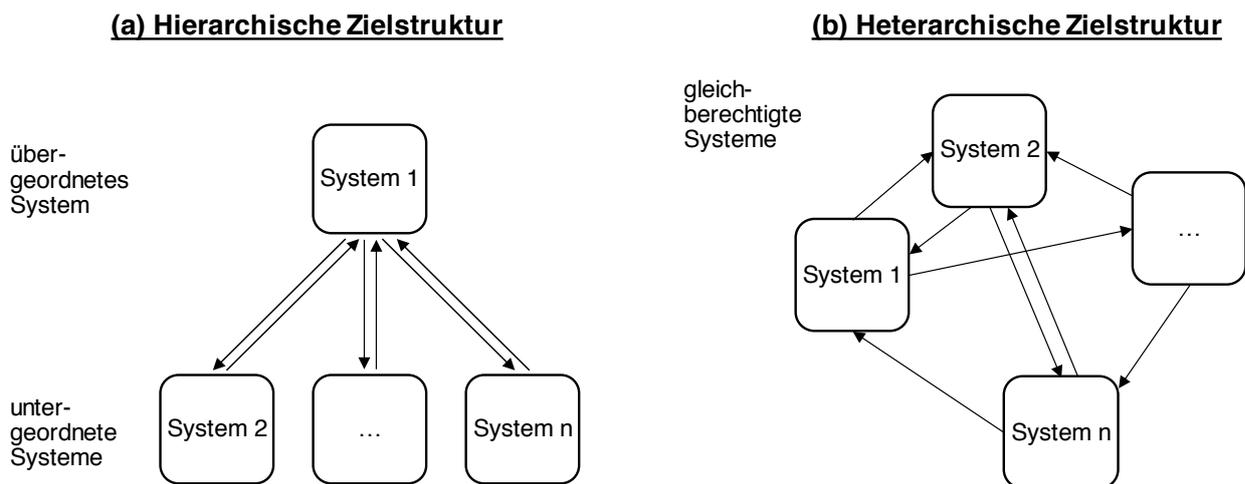


Abbildung 9-1: Grundsätzliche Arten von Zielstrukturen

Hierarchische Zielstrukturen zeichnen sich dadurch aus, dass sie aus über- und untergeordneten Systemen bestehen. Die übergeordneten Systeme führen die untergeordneten Systeme z.B. anhand von Zielvorgaben. D.h. im Umkehrschluss folgen die untergeordneten Systeme den Anweisungen der übergeordneten Systeme. Der Informationsfluss, z.B. zum aktuellen Zustand und der Zielerreichung der untergeordneten Systeme, findet also bottom-up statt, während der Befehlsfluss vom übergeordneten zu den untergeordneten Systemen top-down stattfindet. Damit eine solche Zielstruktur funktioniert, muss die hierarchische Architektur der Systeme bekannt und die Weisungsbefugnisse der Teilnehmer müssen geklärt sein. Dies ist der Fall, wenn es sich um Systeme innerhalb eines horizontalen Wertschöpfungsschritts handelt und die Ziele der untergeordneten Systeme Teilziele der übergeordneten Systems wie z.B. eines Karosseriebaus sind. Auf

den Anwendungsfall der Karosseriefolgesteuerung bezogen könnten Leitziele z.B. die Anzahl und Qualität der gefertigten Karossen pro Zeiteinheit sein.

Im Gegensatz dazu bestehen **heterarchische Zielstrukturen** aus gleichberechtigten Systemen, die der Erfüllung eines gemeinsamen Zwecks dienen und horizontal unterschiedliche Schritte in der Wertschöpfungskette darstellen. Diese Systeme beeinflussen sich gegenseitig, sodass eine Abstimmung ihrer Ziele untereinander notwendig ist. Dies kann z.B. über eine gemeinsame Zielfunktion oder eine zusätzliche Instanz zur gemeinsamen Lösungsfindung geschehen. Die Schwierigkeit ist hierbei, dass potenziell eine große Menge an Faktoren berücksichtigt werden muss, die sich gegenseitig beeinflussen und im Zielkonflikt stehen können. Mit einer zunehmenden Anzahl an Faktoren steigt die Komplexität der Integration überproportional, sodass das Programmieren mithilfe von einfachen Regeln oder Algorithmen bei der Realisierung schnell an praktische Grenzen stößt. Gerade in solchen Fällen hat das Ausbalancieren der jeweiligen Variablen in den Zielfunktionen aber eine hohe Bedeutung für die Erreichung des globalen Optimums des Gesamtsystems. Einen möglichen Ansatz, um in einem solchen Fall mit wenig Entwicklungsaufwand eine praktikable Lösung herbeizuführen, könnten neuronale Netzwerke darstellen. Sie verfügen über die Fähigkeit komplexe multikriterielle Funktionen über ein Netzwerk aus Neuronen mit mehreren Ebenen zu bewerten und in einem einfachen Wert zusammenzufassen bzw. zu klassifizieren. Als Ergebnis könnte so ein Leitwert ermittelt werden – unter Umständen als Matrix mit Steuerungswerten für die einzelnen Systeme – und an die individuellen Systeme zurückgegeben werden.

Bei der Bewertung der Abhängigkeiten zwischen interagierenden Zielsystemen sind daher sowohl die Art der Zielstruktur (hierarchisch/heterarchisch) als auch die horizontale und vertikale Stellung der Systeme in der Wertschöpfungskette zu berücksichtigen. Um diese Anforderungen zu erfüllen wird im nächsten Abschnitt auf Basis von DACS eine Entwurfsmethodik erarbeitet, die eine schrittweise Entwicklung und Integration von agentenbasierten Steuerungssystemen ermöglicht und damit den Weg für umfassendere, vertikal und horizontal integrierte Steuerungsarchitekturen bereitet.

9.3 Erweiterung der DACS-Methode

Die DACS-Methode stellt ein universelles Vorgehen zur Entwicklung von Agentensystemen mit einer geschlossenen Menge von Steuerungsentscheidungen zur Verfügung. Die Gesamtheit der Steuerungsentscheidungen in komplexen Produktionssystemen ist jedoch in der Regel zu groß, um in einem Schritt erarbeitet und von einem einzigen Agentensystem gesteuert zu werden. Daher ist es erforderlich, separate Agentensysteme für verschiedene Arten von Steuerungsentscheidungen zu entwickeln und diese anschließend zu integrieren. Vor diesem Hintergrund werden in Abschnitt 9.3.1 zunächst die Resultate der drei Schritte der DACS-Methode anhand der Anwendungsfälle aus dieser Arbeit untersucht. Da sie die Ausgangssituation für die Integration darstellen, werden sie als Phase 1 bezeichnet. Auf ihrer Grundlage werden im darauffolgenden Abschnitt

9.3.2 drei ergänzende Schritte als Erweiterung der DACS-Methode für die Integration voneinander abhängiger Agentensysteme definiert. Diese konstituieren die Phase 2 der erweiterten DACS-Methode. In Abschnitt 9.3.3 werden die beiden Phasen schließlich zu einem ganzheitlichen, iterativen Vorgehen zusammengeführt.

9.3.1 Phase 1: Konzeption individueller Agentensysteme

Die in der Vergangenheit betriebenen Modularisierungsanstrengungen und die aus ihnen resultierende Entwicklung modularer Steuerungsarchitekturen [VFL14] führt dazu, dass individuelle Agentensysteme für geschlossene Steuerungsprobleme entwickelt werden können.

Bezogen auf die DACS-Methode wird in jedem Schritt des in Abschnitt 4.2.5 beschriebenen Vorgehens ein auf einen konkreten Anwendungsfall ausgerichtetes Modell erarbeitet. Im ersten Schritt entsteht so ein Entscheidungsmodell, im zweiten ein Agentenmodell und im dritten schließlich ein Interaktionsmodell. Durch ein iteratives Vorgehen können mit der DACS-Methode beliebig viele Bestandteile eines Produktionssystems in ein agentenbasiertes Steuerungssystem umgewandelt werden. In Abbildung 9-2 sind die auf diese Weise erarbeiteten spezifischen Modelle für die konkreten Anwendungsfälle der Karosseriefolgesteuerung und der Routenzuglogistik aus Kapitel 7 und 8 dargestellt. Es können aber auch alle anderen Steuerungssysteme eines Produktionssystems bei denen eine agentenbasierte Steuerung sinnvoll ist auf diese Art entworfen werden (z.B. die Wochenprogrammbildung, JIT-Abrufe bei externen Lieferanten oder die Cockpitvormontage).

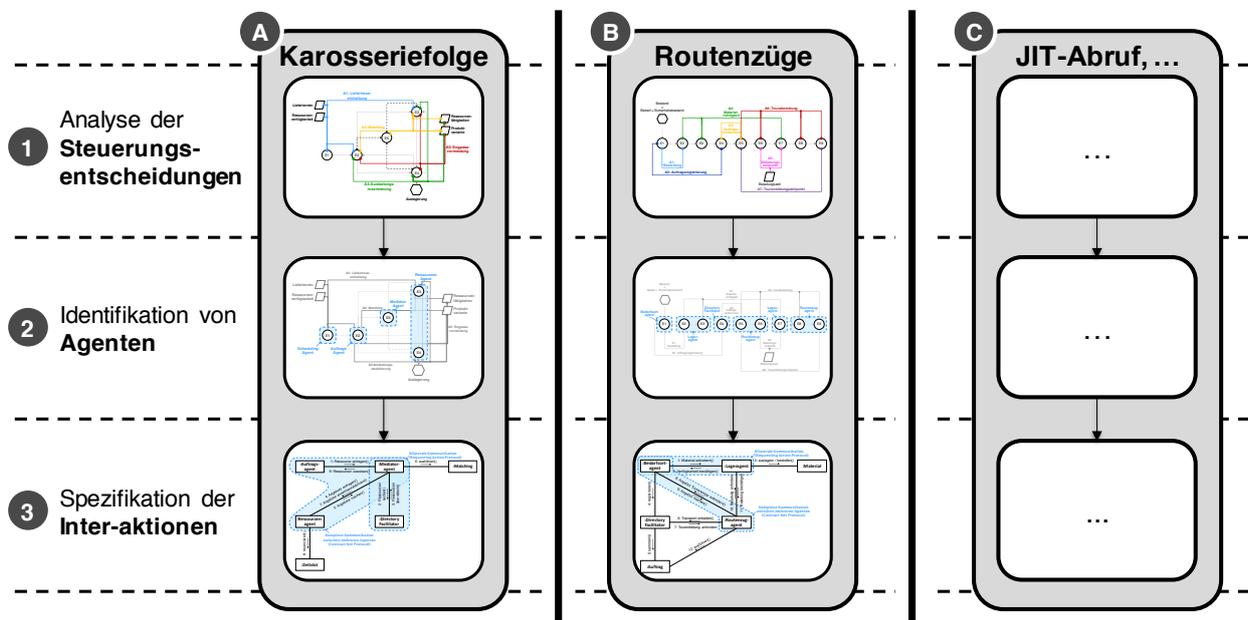


Abbildung 9-2: Individuelle Agentensysteme als Ergebnisse der Phase 1

Die resultierenden Agentensysteme sind für sich gesehen funktionsfähig und können nebeneinander existieren, ohne miteinander in Interaktion zu treten. In vielen Fällen bestehen zwischen

den Systemen jedoch Abhängigkeiten, die nicht im Rahmen der Entwicklung mit der DACS-Methode berücksichtigt wurden, weil sich das Vorgehen explizit auf geschlossene Systeme bezieht. Da die Systeme nicht miteinander kommunizieren, treffen sie ihre Entscheidungen größtenteils unabhängig von den Erfordernissen der anderen Systeme. Die von den Systemen zu bewältigenden Steuerungsaufgaben sind jedoch nicht unabhängig voneinander, sodass es oft zweckmäßig ist die Grenzen zwischen diesen Insellösungen aufzubrechen und die Systeme ineinander zu integrieren. Ein für diesen Zweck mögliches Entwicklungsvorgehen wird im Folgenden vorgestellt.

9.3.2 Phase 2: Integration von Agentensystemen

Die Erweiterung der DACS-Methode setzt auf den Ergebnissen der ersten Phase auf, d.h. es wird davon ausgegangen, dass bereits konzipierte Agentensysteme vorhanden sind, die nun integriert werden sollen. Bevor eine Integration der Systeme ausgeführt wird, sollte zunächst geprüft werden, ob dies tatsächlich zielführend ist. Um dies sicherzustellen sollten folgende Grundsatzfragen geklärt werden: 1) Zahlen die zu integrierenden Systeme auf gemeinsame Ziele ein und existieren Abhängigkeitsbedingungen zwischen den Systemen? 2) Werden Entscheidungen in den Systemen unabhängig getroffen und können sie sich positiv oder negativ beeinflussen? 3) Können die für das Ausbalancieren der gemeinsamen Stellgrößen benötigten Informationen beschafft werden?

Im Falle einer positiven Beantwortung dieser Fragen, sind als nächstes nacheinander die folgenden drei Schritte auszuführen.

Schritt 1: Analyse der Koordinationsentscheidungen

Der Integrationsprozess beginnt mit der Identifizierung der Entscheidungen, die für die fachgerechte Zusammenführung der betrachteten Agentensysteme und der in ihnen getroffenen Steuerungsentscheidungen notwendig sind. Als Grundlage hierfür werden die Entscheidungsmodelle der Agentensysteme aus Phase 1 herangezogen und geprüft, ob bereits alle notwendigen Entscheidungen erkannt wurden. Bei der Prüfung kann es unter Umständen dazu kommen, dass neue Entscheidungen identifiziert und die bestehenden Entscheidungsmodelle ergänzt werden müssen. Da diese Entscheidungen für gewöhnlich auf einer höheren Hierarchieebene der GFSS stattfinden und für die Koordination der Handlungen der Agentensysteme untereinander verantwortlich sind, werden sie nachfolgend als „Koordinationsentscheidungen“ bezeichnet. Wenn keine weiteren Koordinationsentscheidungen mehr identifiziert werden können, gilt es als nächstes die Abhängigkeiten zwischen den neuen integrationsbezogenen Koordinationsentscheidungen und den bestehenden Steuerungsentscheidungen zu identifizieren. Dabei können in Abhängigkeit davon wie verflochten die Agentensysteme miteinander sind eine oder mehrere neue Abhängigkeiten identifiziert werden. Resultat ist ein neues Entscheidungsmodell, das sowohl die Entscheidungen und Abhängigkeiten der bestehenden Agentensysteme als auch die neu hinzugekommenen Erweiterungen enthält. Für die Analyse der neuen Koordinationsentscheidungen und -abhängigkeiten bedarf es darüber hinaus der Erfassung von Eigenschaften, die über die

Kriterien der einfachen Steuerungsentscheidungen und Abhängigkeiten der individuellen Agentensysteme hinausgehen. Zu berücksichtigen sind dabei die in Abschnitt 9.2 diskutierten Eigenschaften hinsichtlich der relativen Verortung der Entscheidungen in der Wertschöpfungskette sowie die Zielstruktur, um potenzielle Schneeballeffekte und Zielkonflikte zu erkennen. Diese Kriterien und Ausprägungsmöglichkeiten definieren sich wie folgt:

1. Relative Verortung der Entscheidungen in der Wertschöpfungskette:

- Horizontal: vorgelagert, nachgelagert, identisch
- Vertikal: übergeordnet, untergeordnet, identisch

2. Zielstruktur von Entscheidungen:

- Hierarchisch: führend, folgend
- Heterarchisch

Das resultierende übergreifende Entscheidungsmodell ist Grundlage für den nächsten Schritt.

Schritt 2: Analyse von Koordinationsagenten

Aus dem übergreifenden Entscheidungsmodell sollten Koordinationsentscheidungen und -abhängigkeiten hervorgehen, die für eine ganzheitliche Steuerung der zu integrierenden Agentensysteme notwendig sind. Ist dies nicht der Fall, bestehen entweder keine Abhängigkeiten zwischen den Systemen oder sie wurden nicht korrekt identifiziert. Im letzten Fall sollte der erste Schritt wiederholt werden, andernfalls besteht kein Bedarf die Systeme zu integrieren. Wurden Koordinationsentscheidungen und -abhängigkeiten identifiziert, müssen im nächsten Schritt die Agentenmodelle der zu integrierenden Systeme erweitert werden. Dazu werden die zusätzlichen Koordinationsentscheidungen zunächst geclustert und die beste und effizienteste Struktur für die Lösung des gemeinsamen Optimierungsproblems, das in der Erreichung eines Gesamtsystemoptimums besteht, gesucht. Als Ergebnis werden die Entscheidungen entweder bestehenden Agenten zugewiesen oder es werden neue Agenten entworfen und ihnen die Verantwortung für die systemübergreifende Koordination übertragen. Diese Agenten werden im Folgenden als „Koordinationsagenten“ bezeichnet. Für die richtige Zuordnung von Entscheidungen zu Agenten hat die Identifikation und Analyse der für diese Entscheidungen notwendigen Informationen einen hohen Stellenwert. Bevor eine qualifizierte Aussage getroffen werden kann, sollten daher alle Informationen identifiziert werden, die zwischen Agenten ausgetauscht werden müssen, um übergreifende Koordinationsentscheidungen treffen zu können. Als Ergebnis entsteht ein übergreifendes Agentenmodell für die zu integrierenden Agentensysteme.

Schritt 3: Spezifikation der Koordinationsinteraktionen

Im letzten Schritt der erweiterten AOSE-Methode werden schließlich die notwendigen Wechselwirkungen zwischen identifizierten Koordinationsentscheidungen auf Interaktionsprotokolle abgebildet und damit die notwendigen Interaktionen zwischen den Agentensystemen festgelegt. Diese Protokolle werden an die speziellen Bedürfnisse der zu treffenden Koordinationsentscheidungen

angepasst und untereinander harmonisiert, sodass als Ergebnis des dritten Schritts ein übergreifendes Interaktionsmodell entsteht.

Dies kann bedeuten, dass die zuvor für die Interaktionen innerhalb eines bestehenden Agentensystems ausgewählten Protokolle nicht mehr ausreichen, um die zusätzlichen Anforderungen der neuen Koordinationsentscheidungen abzubilden. Hierbei kommen prinzipiell zwei Ansätze in Frage: Einerseits kann das in dem bestehenden Agentensystem verwendete Interaktionsprotokoll durch ein komplexeres, für den Anwendungsfall geeignetes Protokoll ersetzt werden. Dies bringt den Nachteil mit sich, dass das bereits bestehende und sich möglicherweise bereits im Einsatz befindliche Agentensystem erneuert werden muss, was mit zusätzlichem Aufwand für Entwicklung, Erprobung und Inbetriebnahme verbunden sein kann. Andererseits besteht die Möglichkeit die betroffenen Agenten mit zusätzlichen Interaktionsprotokollen auszustatten, die ausschließlich für die neu hinzugekommenen Koordinationsaufgaben verantwortlich sind, und beide Protokolle parallel anzuwenden. Der offensichtliche Vorteil des letzteren Vorgehens besteht in den deutlich geringeren Kosten für die Entwicklung und Bereitstellung des finalen Systems. Befinden sich die individuellen Systeme jedoch z.B. noch in der Entwicklungsphase, können die Kostenvorteile u.U. schnell durch im Betrieb entstehende Vorteile aufgewogen werden. Die Entscheidung für eines der Vorgehen hängt von zahlreichen weiteren Faktoren wie z.B. der Kommunikationslast, der Rechenleistung der zur Verfügung stehenden Systeme, den Performanceanforderungen etc. ab und muss daher im Einzelfall bewertet werden.

Zusammengefasst beinhaltet die beschriebene Erweiterung der DACS-Methode die folgenden drei übergeordneten Schritte, die sich auf die Koordination zwischen den Systemen fokussieren:

1. Analyse der Koordinationsentscheidungen und Abhängigkeiten
2. Identifikation der für die Koordination erforderlichen Agenten und Informationen
3. Spezifikation der Interaktionen und Protokolle, die für die Koordination erforderlich sind.

Durch Anwendung dieses Vorgehens kann die Integration bestehender Agentensysteme realisiert werden. Damit das hier entwickelte Vorgehen auf eine beliebige Anzahl zu entwickelnder und zu integrierender Subsysteme eines übergreifenden Produktionssystems angewendet werden kann, werden die Schritte der beiden Phasen im nächsten Abschnitt zu einer ganzheitlichen Methode zusammengefasst.

9.3.3 Zusammenführung der Phasen zu einer erweiterten AOSE-Methode

Komplexe Produktionssysteme mit einer Vielzahl von Steuerungsproblemen benötigen ein systematisches Vorgehen, um die Transition von konventionellen zu ganzheitlich integrierten agentenbasierten Steuerungssystemen zu bewältigen. Die in Abschnitt 9.3.1 und 9.3.2 vorgestellten Schritte für die iterative Entwicklung und Integration von Agentensystemen stellen zwei Phasen

da, die schrittweise durchlaufen werden können, um diese Transition für Unternehmen handhabbar zu machen. In Summe entsteht dadurch eine erweiterte AOSE-Methode mit insgesamt sechs Schritten – drei für die Entwicklung individueller Agentensysteme und drei für ihre Integration – die in Abbildung 9-3 dargestellt ist.

In der Phase 1 der Methode wird ein zu agentifizierendes Steuerungsproblem ausgewählt und ein entsprechendes Agentensystem auf Basis der drei Schritte der DACS-Methode konzipiert. Als Ergebnis entsteht ein Entscheidungs-, ein Agenten- und ein Interaktionsmodell für das ausgewählte Problem. Anschließend wird überprüft, ob weitere relevante Agentensysteme bestehen, die es zu integrieren gilt. Ist dies nicht der Fall, wird ein neues zu agentifizierendes Steuerungsproblem ausgewählt und es werden die ersten drei Schritte für dieses Problem wiederholt. Kann die Frage nach relevanten, zu integrierenden Agentensystemen anschließend positiv beantwortet werden, wird zur Phase 2 – der Integration der Systeme – übergegangen.

In dieser Phase werden die ausgewählten Agentensysteme anhand des in Abschnitt 9.3.2 definierten Vorgehens integriert. Dabei werden sequentiell die drei Schritte Analyse der Koordinationsentscheidungen, Identifikation von Koordinationsagenten und die Spezifikation der Koordinationsinteraktionen durchgeführt. Als Ergebnis der Integrationsphase entsteht das aus den übergreifenden Entscheidungs-, Agenten- und Interaktionsmodellen bestehende Konzept des integrierten Agentensystems. Die Phase 2 schließt mit der Frage nach weiteren zu integrierenden Steuerungsproblemen. In Abhängigkeit der Verfügbarkeit zu integrierender Agentensysteme wird die Methode mit Schritt eins bzw. vier fortgesetzt. Anderenfalls wurde das Ziel der AOSE-Methode erreicht und es kann zur Implementierung des nun integrierten Systemkonzepts übergegangen werden.

Zusammenfassend ist mit der vorgestellten erweiterten AOSE-Methode eine schrittweise Entwicklung von zunächst selbstständigen Agentensystemen möglich, die anschließend integriert werden können. Dies ermöglicht im Gegensatz zum Status quo ein iteratives Vorgehen bei der Migration von zentralen zu dezentralen Steuerungssystemen und macht den Einzug der Agententechnologie in Industrieanwendungen praktikabel. Da die entwickelte AOSE-Methode lediglich die Frage nach dem „wie“ jedoch nicht die Frage nach dem „was“ klärt, kann sie mit systematischen Migrationsmethoden kombiniert werden. Ein dafür in Frage kommender Ansatz ist das Migrationsvorgehen von [CLC17], das aus den fünf Schritten Vorbereitung, Optionsprüfung, Design, Implementierung und Deployment besteht.

Um die Anwendbarkeit der entwickelten AOSE-Methode anhand von konkreten Beispielen zu verifizieren wird das Vorgehen im letzten Schritt dieser Arbeit anhand der in Kapitel 7 und 8 vorgestellten Anwendungsfälle überprüft.

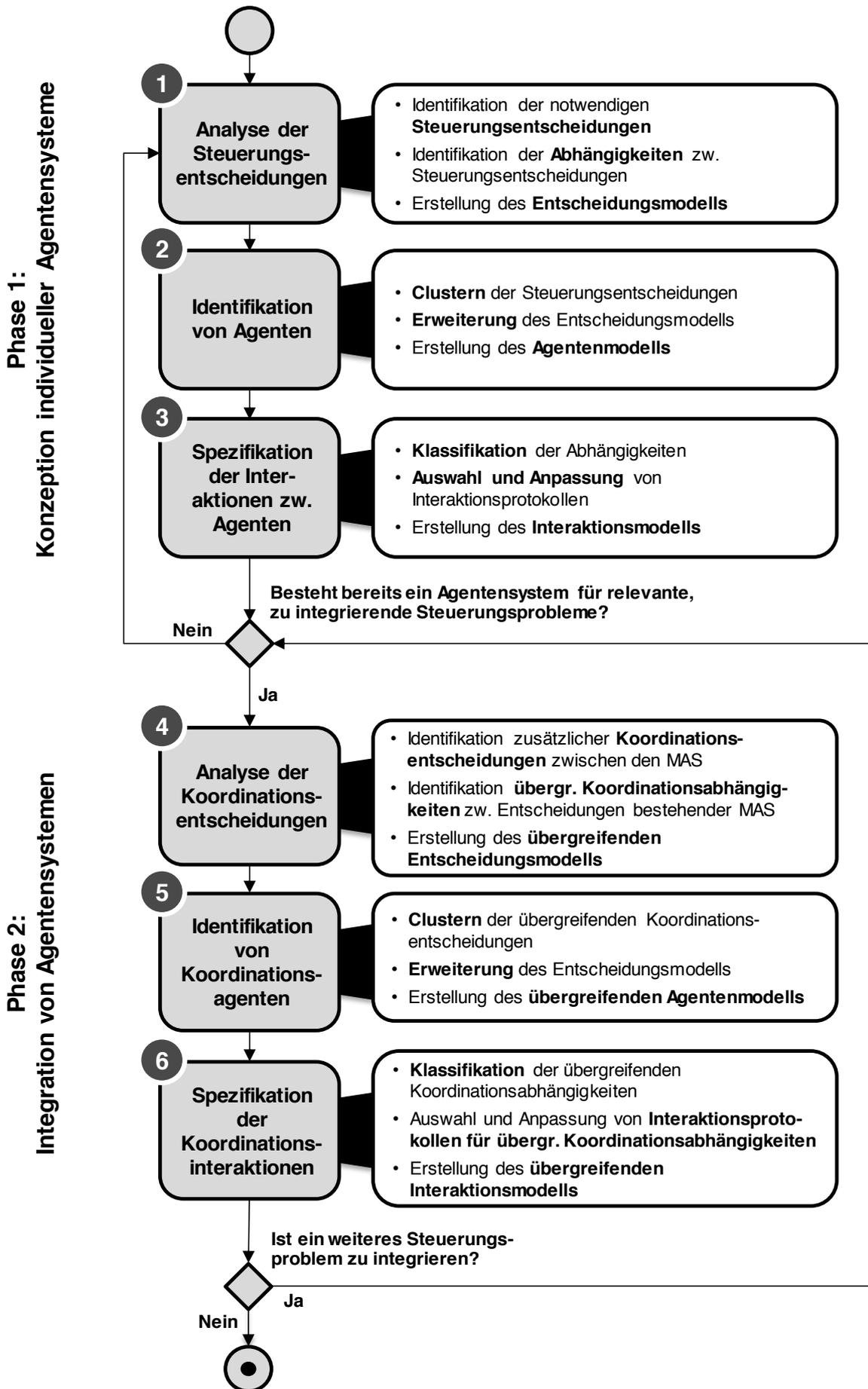


Abbildung 9-3: Erweiterte AOSE-Methode für interdependente Agentensysteme

10 Integration der Anwendungsfälle

Zur Veranschaulichung der entwickelten AOSE-Methode erfolgt abschließend die Integration der in dieser Arbeit betrachteten Anwendungsfälle mit dem formulierten Vorgehen. Dafür werden zunächst in Abschnitt 10.1 Vorbetrachtungen zu den Systemen, ihren Zielsetzungen und den möglichen Korrelationen vorgenommen. Im Anschluss daran erfolgt in Abschnitt 10.2 die systematische Integration der beiden Anwendungsfälle. Da die ersten drei Schritte der Methode bereits in den Kapiteln 7 und 8 durchgeführt wurden, setzt dieses Kapitel am vierten Schritt an. Folglich werden auf dieser Grundlage die drei übergreifenden Modelle – Koordinationsentscheidungs-, Agenten- und Interaktionsmodell – entworfen. Als letzter Schritt dieser Arbeit erfolgt in Abschnitt 10.3 eine Betrachtung und kritische Beurteilung der Ergebnisse sowie eine Diskussion der weiteren Stoßrichtungen für anknüpfende Forschungsarbeiten.

10.1 Vorbetrachtungen

Die beiden zu implementierenden Agentensysteme sind an unterschiedlichen Stellen der GFSS verortet. Horizontal ist der Schritt der Karosseriefolgesteuerung dem Schritt der Routenzugsteuerung in der Montage vorgelagert, sodass sich die Karosseriefolge direkt auf die Materiallogistik auswirkt. Vertikal finden die Entscheidungen des ersten Systems auf den Ebenen 7 und 8 statt, während die Entscheidungen des zweiten Systems darunter liegen und auf den Ebenen 6 und 7 getroffen werden.

Die individuellen Zielsetzungen der Systeme weichen voneinander ab und können daher die gemeinsame Zielsetzung sowohl positiv als auch negativ beeinflussen. Die Karosseriefolgesteuerung verfolgt das Ziel die eigene Liefertreue und die Auslastung der eigenen Ressourcen zu optimieren. Dafür muss das System Engpässe vermeiden und benötigt die bestmögliche Zuordnung von Aufträgen zu Ressourcen (vgl. Kapitel 7). Die Routenzugsteuerung hat hingegen das Ziel eine zuverlässige Belieferung der Bedarfsorte sicherzustellen und verspätete Materialanlieferungen zu vermeiden. Damit einher geht auch die Maximierung der Routenzugauslastung, die Reduzierung von Wartezeiten und die Erhöhung der Auftragsanzahl pro Tour. Sekundär soll dadurch auch die Menge des am Fließband vorgehaltenem Materials und der damit zusammenhängenden Kapitalbindungskosten reduziert werden (vgl. Kapitel 8). Während die Verfolgung dieser Ziele aus der Perspektive der jeweiligen Systeme sinnvoll ist, kann die Bewertung der daraus entspringenden Handlungen aus Sicht des integrierten Gesamtsystems widersprüchlich ausfallen.

Allgemein verfolgen Unternehmen in der Produktion das Ziel eine möglichst hohe Ausbringung an Produkten der angestrebten Qualität mit einer hohen Termineinhaltung und bei geringen Kosten zu erreichen. Als Unterziele ergeben sich Faktoren wie eine hohe Auslastung der Fertigungsbereiche und Maschinen sowie eine geringe Durchlaufzeit [BBB14]. Die obigen Systeme zahlen auf die Erreichung dieser Ziele ein, können aber bei der Verfolgung ihrer eigenen Ziele das End-

ergebnis für das Gesamtsystem negativ beeinflussen, z.B. wenn der Karosseriebau zur Maximierung seiner eigenen Ausbringung mehr Modelle eines spezifischen Fahrzeugtyps fertigt, die in der Montage allerdings nicht in ausreichender Geschwindigkeit bearbeitet oder mit dem richtigen Material versorgt werden können.

Diese Korrelationen müssen berücksichtigt werden, um die Subsysteme auf die Erreichung der gemeinsamen Ziele des Gesamtsystems auszurichten. Dafür sind im vorliegenden Fall grundsätzlich zwei Arten des Informationsaustausches zwischen den Systemen denkbar. Einerseits kann die Karosseriefolgesteuerung dem Routenzugsystem Informationen über den geplanten bzw. tatsächlichen Produktmix (mit allen relevanten Informationen wie z.B. Liefertermine, Produktvarianten, DLZ etc.) mitteilen. Andererseits kann die Routenzugsteuerung die Karosseriefolgesteuerung informieren, sobald Ereignisse auftreten, die für die Reihenfolge relevant sind (z.B. eine Überschreitung der temporären Liefer- und Bereitstellungskapazitäten von Lagern und Routenzügen, Anlagenstörungen und Sperrungen).

Im ersten Fall könnte die Karosseriefolgesteuerung z.B. in regelmäßigen Zeitabständen Informationen hinsichtlich des voraussichtlichen Füllstandes der verschiedenen Gewerke mit den unterschiedlichen Varianten des Produktmix prognostizieren und an das Routenzugsystem weiterleiten. Darin könnten Daten wie z.B. die aktuelle Reihenfolgeplanung und die Anzahl der jeweiligen Produktvarianten an unterschiedlichen Statuspunkten der Gewerke enthalten sein. Mit diesen Informationen ausgestattet könnte die Routenzugsteuerung eine Aussage darüber treffen wie gut die geplante Karosseriefolge, d.h. die Reihenfolge der Ausbringung der jeweiligen Gewerke, durch das Routenzugsystem bewältigt werden kann und einen entsprechenden Fitnesswert an die Karosseriefolgesteuerung zurückgeben. Auf dieser Grundlage könnten notwendige Anpassungen in der Karosseriefolgesteuerung berücksichtigt und auf die Bedürfnisse der nachfolgenden Systeme wie der Routenzugsteuerung angepasst werden. Eine schematische Darstellung dieses Ablaufs ist in Abbildung 10-1 dargestellt.

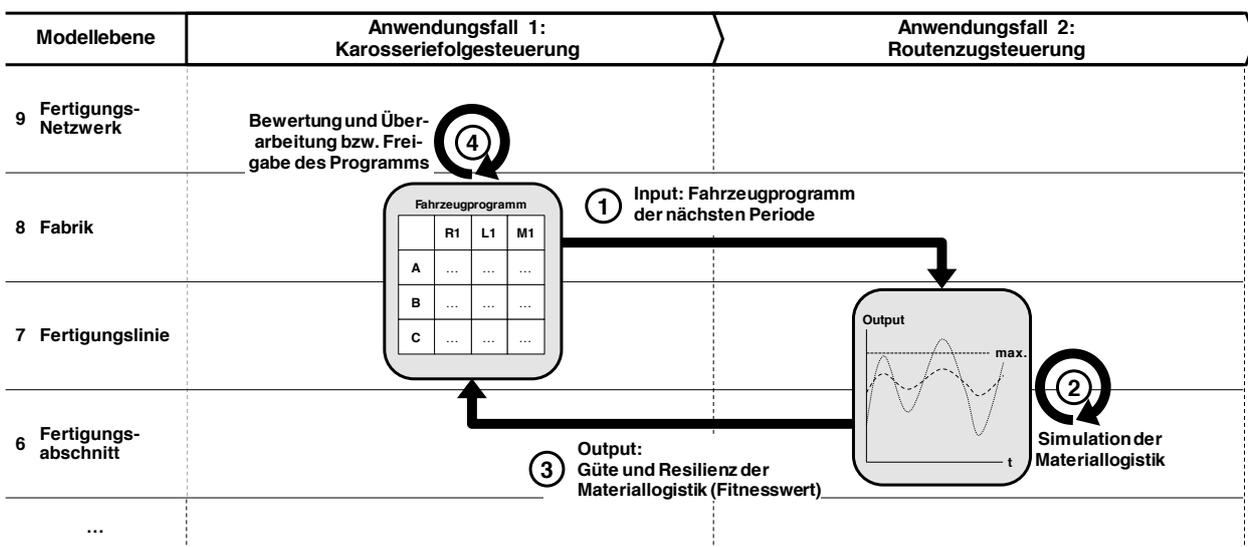


Abbildung 10-1: Möglicher Informationsaustausch zwischen den Agentensystemen

Damit ist teilweise auch die zweite Art des Informationsaustausches abgedeckt: wenn das Routenzugsystem selbst von Störungen betroffen ist, die die Bearbeitung der aufgelegten Auftragsfolge erschwert, könnte es auch proaktiv einen entsprechenden Fitnesswert an die Karosseriefolgesteuerung mitteilen, um die Auswirkungen der Störungen für das Gesamtsystem abzuschwächen. Dies kann sowohl bei vorhersehbaren Ereignissen wie Instandhaltungsmaßnahmen als auch bei unvorhersehbaren Ereignissen wie Anlagenstörungen oder Materialengpässen in den Lagern notwendig sein, falls dadurch die Materiallogistik nicht in vollem Umfang ausgeführt werden kann.

Die unterschiedlichen Zielsetzungen und Korrelationen zwischen den Systemen erhöhen die Komplexität bei der Integration der beiden Anwendungsfälle. Zudem ist bei der Integration zu berücksichtigen, dass die Anwendungsfälle nicht nur horizontal, sondern auch vertikal unterschiedlich verortet sind. Vor dem Hintergrund der in Abschnitt 9.2 aufgeführten Herausforderungen besteht ferner ein gewisses Risiko von ungewolltem Verhalten im Sinne von Schwingungen im integrierten System. Um dies zu vermeiden, bedarf es einer systematischen Analyse der Ziel-funktionen und des Systemverhaltens des integrierten Steuerungssystems. Dafür werden Methoden benötigt, die derzeit nicht zur Verfügung stehen und weiterer Forschungsanstrengungen bedürfen, die im Rahmen dieser Arbeit nicht vollumfänglich geleistet werden können. Zur Lösung der damit zusammenhängenden Fragen bedarf es zunächst jedoch der Integration der Systeme. Daher sollen mit der Anwendung der erarbeiteten AOSE-Methode auf den nächsten Seiten die konzeptionellen Fragen bei der Integration der beiden Anwendungsfälle geklärt und der Weg zur Entwicklung weiterer Methoden in anknüpfenden Forschungsprojekten bereitet werden. Die dafür notwendigen Schritte werden im nächsten Abschnitt beschrieben.

10.2 Entwicklung des integrierten Agentensystems

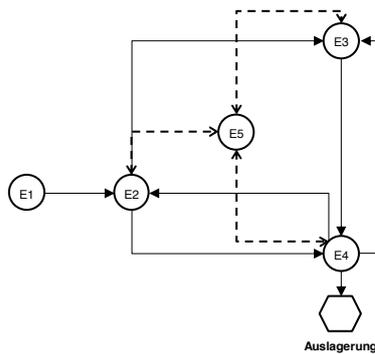
Die Konzeptionierung des integrierten Agentensystems aus Karosseriefolge- und Routenzugsteuerung folgt den Schritten 4 bis 6 der erweiterten AOSE-Methode aus Abbildung 9-3. In Abschnitt 10.2.1 wird zunächst ein übergreifendes Entscheidungsmodell auf Basis der individuellen Agentensysteme entwickelt. Dabei werden die Koordinationsentscheidungen und die zwischen ihnen bestehenden Abhängigkeiten identifiziert. Im nächsten Schritt erfolgt in Abschnitt 10.2.2 die Entwicklung eines übergreifenden Agentenmodells. Es basiert auf der Analyse und Definition von Koordinationsagenten, die übergreifende Koordinationsentscheidungen treffen und Informationen zwischen den Systemen austauschen. In Abschnitt 10.2.3 werden schließlich die Interaktionen zwischen den Agenten untersucht und geeignete Protokolle für sie ausgewählt, was in einem übergreifenden Interaktionsmodell resultiert.

10.2.1 Übergreifendes Entscheidungsmodell

Analyse der Koordinationsentscheidungen

Im ersten Schritt bei der Integration von Systemen mit der erweiterten DACS-Methode gilt es die Menge der notwendigen Koordinationsentscheidungen und ihrer Abhängigkeiten zu identifizieren. Die Grundlage für die Analyse der Koordinationsentscheidungen stellen die beiden in Kapitel 7 und 8 erarbeiteten und in Abbildung 10-2 zusammengetragenen individuellen Entscheidungsmodelle der Agentensysteme dar.

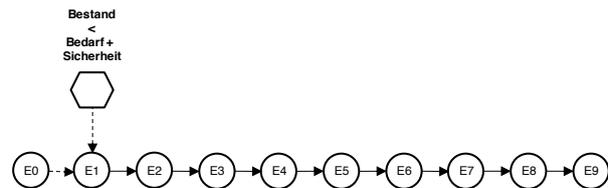
Anwendungsfall 1: Karosseriebaufolge



E1: Auftragsfreigabe
E2: Ressourcenauswahl
E3: Auftragsauswahl

E4: Auftragsbearbeitung
E5: Zuordnung

Anwendungsfall 2: Routenzuglogistik



E1: Auftragsgenerierung
E2: Materialverfügbarkeit
E3: Materialauslagerung
E4: Auftragssammlung
E5: Tourenbildung

E6: Tourenstartbestimmung
E7: Routenzugbeladung
E8: Behälteranlieferung
E9: Rückmeldung Bahnhof

Abbildung 10-2: Entscheidungsmodelle der individuellen Agentensysteme

Aus den Vorbetrachtungen im vorangehenden Abschnitt wird deutlich, dass die Kompatibilität der Planungen und aktuellen Zustände der beiden Systeme sichergestellt werden muss. Da eine solche Prüfung bisher weder in der Karosseriefolge- noch in der Routenzugsteuerung berücksichtigt wird, bedarf es der Ergänzung der Systeme um eine virtuelle Koordinationsentscheidung E0. Ziel ist es hierbei die aktuellen Planungsstände auszutauschen, um bei Bedarf Anpassungen vornehmen zu können. Die Karosseriefolgesteuerung kann auf Basis eines Fitnesswertes der Routenzugsteuerung über Freigabezeitpunkte, aufzulegende Produktvarianten sowie die daraus resultierende Fertigungssequenz entscheiden. Umgekehrt kann das Routenzugsystem die Bedarfsmeldungen unterschiedlich priorisieren sowie die Auslagerung und Beladung von Material auf Routenzüge beeinflussen, um sich den Anforderungen der Reihenfolgesteuerung anzupassen. Die Details der zusätzlichen Koordinationsentscheidung E0 sind in Tabelle 10-1 zusammengefasst.

ID	E0
Titel	Kompatibilitätsprüfung der Planungen
Parameter	Fitnesswerte der Agentensysteme
Steuerungsinterface	virtuelle Entscheidung
Auslöser	Zyklischer Informationsaustausch bzw. auftretende Störfälle in einem der Systeme
Entscheidungsraum	KFS: Freigabezeitpunkte, freizugebende Produktvarianten und Fertigungssequenz RZS: Priorisierung von Bedarfsmeldungen, Auslagerung und Beladung von Material
Lokale Entscheidungsregel	Fitnesswerte optimieren

Tabelle 10-1: Zusätzliche Koordinationsentscheidung im integrierten Agentensystem

Identifikation von Koordinationsabhängigkeiten

Aus der Gegenüberstellung der Steuerungsentscheidungen der beiden Agentensysteme ergeben sich aufgrund der bestehenden Abhängigkeiten vier zusätzliche Koordinationsbedarfe. Sie können zu den vier in Tabelle 10-2 dargestellten Punkten zusammengefasst werden.

ID	A1	A2	A3	A4
Titel	Materialbelieferungs-kapazitäten	Lagerkapazitäten	Routenzugkapazitäten	Montageeinlauf
Entscheidungen	KFS: E1, E5 RZS: E2	KFS: E1, E5 RZS: E3	KFS: E1, E5 RZS: E8	KFS: E1, E5 RZS: E0
Restriktionen	Materialabrufe vermeiden, die zu einer Über- oder Unterauslastung interner und externer Lieferanten führen würde	Lagerplätze und Auslagerungskapazitäten für Materialsorten einhalten	Hohe Konzentration von Abrufen in örtlich begrenztem Raum vermeiden	Keine Karosseriefolge zulassen, die zu einer Über- oder Unterauslastung des Routenzugsystems führen würde
Präferenzen	Gleichmäßige Abrufe von Bauteilen für alle Produktvarianten	Zeitliche und örtliche Verteilung der Ein- und Auslageraufträge	Zeitliche und örtliche Verteilung der Materialabrufe an Bedarfsorten	Sequenzierungsregeln der Montage einhalten und aufwändige Produktvarianten gleichmäßig verteilen

Tabelle 10-2: Abhängigkeiten von Entscheidungen zwischen den Systemen

Als erstes kann eine Koordinationsabhängigkeit bezüglich der **Materialbelieferungskapazitäten (A1)** identifiziert werden, denn um einen definierten Umfang an Produktvarianten herzustellen, müssen alle Materialien in ausreichender Menge vorhanden sein. Falls interne oder externe Lieferanten die Teilezufuhr nicht schnell genug sicherstellen können, ist die designierte Karosseriefolge nicht ohne Störungen abbildbar. Ein Abgleich der notwendigen und verfügbaren Materialbelieferungskapazitäten ist daher stets notwendig. Ein ähnliches Bild ergibt sich bei den **Lagerkapazitäten (A2)**. Hier sind die Grenzen der verfügbaren Lagerplätze sowie die Geschwindigkeit der Materialauslagerung als Restriktionen zu beachten. Falls der Bedarf an spezifischen Bauteilen zu hoch wird, kann es zu Engpässen und Wartezeiten kommen, die zu vermeiden sind. Ebenso sind die **Routenzugkapazitäten (A3)** zu beachten, da eine zu hohe Konzentration an Materialanforderungen an wenigen Bedarfsorten zur Bildung von Staus in den Fahrgassen führen kann, die ebenfalls mit unnötigen Wartezeiten verbunden sein können. Schließlich stellt auch der **Montageeinlauf (A4)** eine Abhängigkeit zwischen den Entscheidungen der beiden Agentensysteme dar, die im Rahmen der Integration Beachtung finden sollte. Die im Karosseriebau definierte

und bis zur Lackiererei anpassbare Auftragsreihenfolge hat einen starken Einfluss auf die Prozesse in der Montage. Um die notwendige Reihenfolge beim Montageeinlauf trotz Verwirbelungen zu realisieren verwenden Automobilhersteller heute große und teure KarosSENSpeicher als Zwischenlager. Diese verfügen über frei wählbare Speicherplätze für teilweise bis zu 1.000 Fahrzeug. Sie sind mit hohem Platz- und Investitionsaufwand verbunden, reichen aber oft dennoch nicht aus um die geforderten Sequenzabstände zwischen komplexen Fahrzeugkonfigurationen sicherzustellen. Diese KarosSENSpeicher werden von separaten Systemen gesteuert und können im vorliegenden Fall weder durch die KarosSERIEfolge- noch durch die Routenzugsteuerung direkt beeinflusst werden. Stattdessen können die Systeme durch ihre Steuerungsentscheidungen indirekt den Füllstand des KarosSENSpeichers steuern. Dies stellt eine Herausforderung bei der Integration dar, die ohne eine sorgfältige Analyse der Koordinationsentscheidungen und -abhängigkeiten unter Umständen nicht ersichtlich wäre. Als Lösungsansatz für diese Problematik kann eine indirekte Beeinflussung des Montageeinlaufs erfolgen, indem sich die beiden Systeme mittels Fitnesswert zur aktuellen Last im jeweiligen System austauschen und daraufhin Anpassungen ihrer internen Entscheidungen vornehmen. Auf diese Weise kann der Produktmix im KarosSENSpeicher und damit indirekt der Montageeinlauf beeinflusst werden.

Als Resultat der Analyse entsteht das in Abbildung 10-3 zusammengefasste übergreifende Entscheidungsmodell des integrierten Agentensystems.

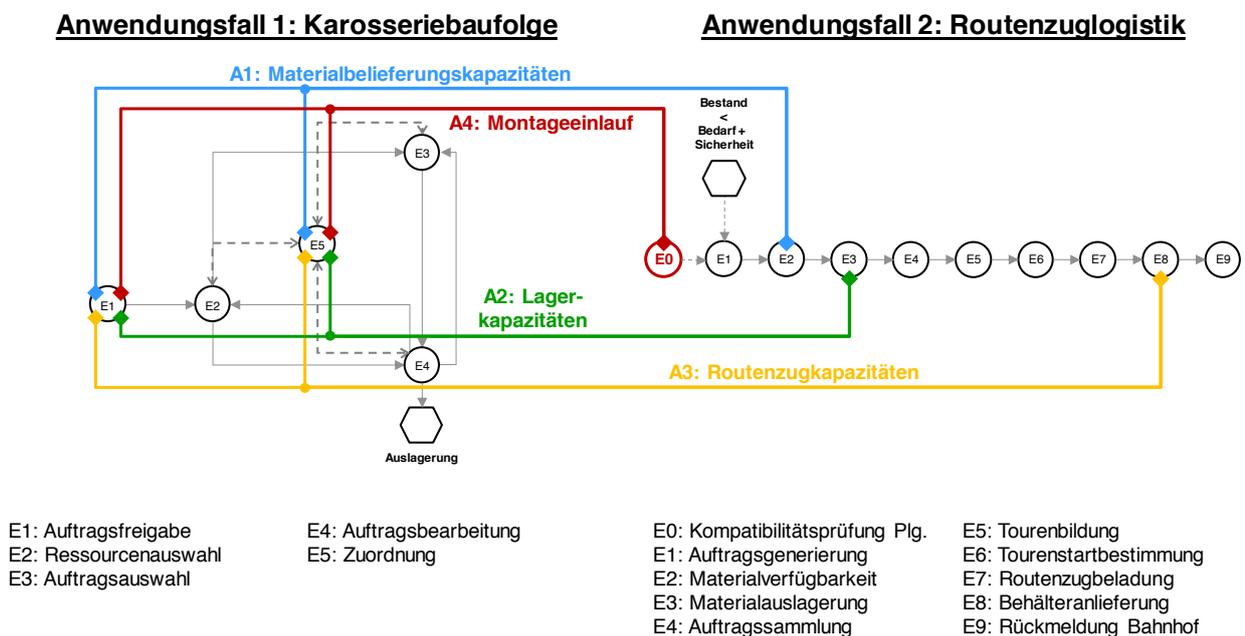


Abbildung 10-3: Übergreifendes Entscheidungsmodell des integrierten Agentensystems

Aus der obigen Darstellung wird ersichtlich, dass starke Abhängigkeiten zwischen den Systemen bestehen. Insbesondere die Auftragsfreigabe und die Auftrags-Ressourcen-Zuordnung der KarosSERIEfolgesteuerung hat Auswirkungen auf Entscheidungen der Routenzugsteuerung wie z.B. die Auftragsgenerierung, die Materialauslagerung oder die Behälteranlieferung. Diese Abhängigkeiten müssen bei der Integration der Systeme Berücksichtigung finden.

10.2.2 Übergreifendes Agentenmodell

Auf Basis der übergreifenden Steuerungsentscheidungen und Abhängigkeiten kann im nächsten Schritt das übergreifende Agentenmodell zur Integration der beiden Anwendungsfälle erstellt werden. Ziel ist es dabei Agenten zu identifizieren, die für die Ausführung der benötigten Koordinationsaufgaben in Frage kommen. Um die Koordination zwischen den Systemen zu ermöglichen müssen Koordinationsagenten ausgewählt werden, die die relevanten Informationen zur rechten Zeit untereinander austauschen und entsprechende Steuerungsentscheidungen treffen können.

Die Verteilung der Verantwortung für die einzelnen Steuerungsentscheidungen kann prinzipiell auf zwei Wegen erfolgen:

Einerseits können die betroffenen Agenten auf Ausführungsebene Informationen direkt untereinander austauschen. Als Beispiel könnte der für die Behälteranlieferung verantwortliche Routenzugagent des Routenzugsystems die Auftrags- und Scheduling-Agenten informieren, wenn zu viele Anfragen für bestimmte Bauteile eingehen und eine gleichmäßigere Verteilung der Anfragen notwendig ist. Dies hat den Vorteil kurzer Kommunikationswege, erhöht jedoch in Summe die Kommunikationslast im System und könnte, aufgrund der Unvorhersehbarkeit vieler parallel ablaufender Kommunikationsprozesse, potenziell zu ungewünschten Effekten wie z.B. den in Abschnitt 9.2 diskutierten Schneeballeffekten führen.

Andererseits gibt es die Möglichkeit die Bedürfnisse bzw. Anforderungen der Systeme konsolidiert über Vertreter der jeweiligen Systeme auszutauschen. Diese Vertreteragenten können die Informationen gegeneinander abwägen und die erforderlichen Schlussfolgerungen ziehen. Dies hat den Vorteil, dass die Kommunikationslast reduziert wird und Entscheidungen zunächst innerhalb der einzelnen Systeme bewertet werden können, bevor sie an das andere System weitergegeben werden. Der dabei entstehende Fitnesswert, der aus einem einzelnen Wert, einem Tupel oder einer Matrix mit mehreren Werten bestehen kann, dient dann zur Justierung des anderen Systems entsprechend der Bedarfe. Nachteil dieser Methode ist, dass diese Konsolidierung Zeit beansprucht und daher Entscheidungen verzögern kann. Da es sich in den hier betrachteten Fällen jedoch nicht um echtzeitkritische Anwendungen im Millisekundenbereich handelt, ist diese Verzögerung vernachlässigbar, sodass eine konsolidierte Übergabe von Informationen bevorzugt wird. Sinnvoll ist diese Methode zudem, da sich mehrere zeitgleich auftretende Anfragen gegenseitig aufheben oder potenzieren können. Diese Effekte können in einer konsolidierten Informationsübertragung grundsätzlich besser gehandhabt werden. Umgekehrt können die Informationen in dem empfangenden System auf einer übergeordneten Ebene heruntergebrochen und anschließend an die unteren Ebenen verteilt werden.

Zur Erfüllung der Aufgaben aus dem übergreifenden Entscheidungsmodell reichen die Informationen untergeordneter Agenten wie z.B. von Produkt- oder Auftragsagenten nicht aus. Stattdessen werden Agenten benötigt, die über ein gutes Gesamtbild der jeweiligen Systeme verfügen. Im Anwendungsfall der Karosseriefolgesteuerung besitzt der Shop-Management-Agent (SMA)

die notwendigen Informationen und Fähigkeiten, um diese Rolle auszufüllen. Er trägt das notwendige Wissen zum Status der Ressourcen und kann auf den Scheduling- und Mediatoragenten einwirken. Damit verfügt er über relevante Stellhebel zur Beeinflussung der Abhängigkeiten zwischen den beiden Agentensystemen. Auf Seite der Routenzugsteuerung existiert ebenfalls ein Agent, der Information zum Zustand des Systems hat und auf Agenten im System einwirken kann: der Main-Agent (MnA). Er repräsentiert im Routenzugsystem die Umwelt für die übrigen Agenten samt aller Umgebungsvariablen und hat damit Einfluss auf wesentliche Stellhebel im System. Die beiden Agenten eignen sich folglich gut für die Übernahme der Koordinationsaufgaben zwischen den zwei Agentensystemen. Das übergreifende Agentenmodell für die zu integrierenden Systeme ergibt sich wie in Abbildung 10-4 dargestellt.

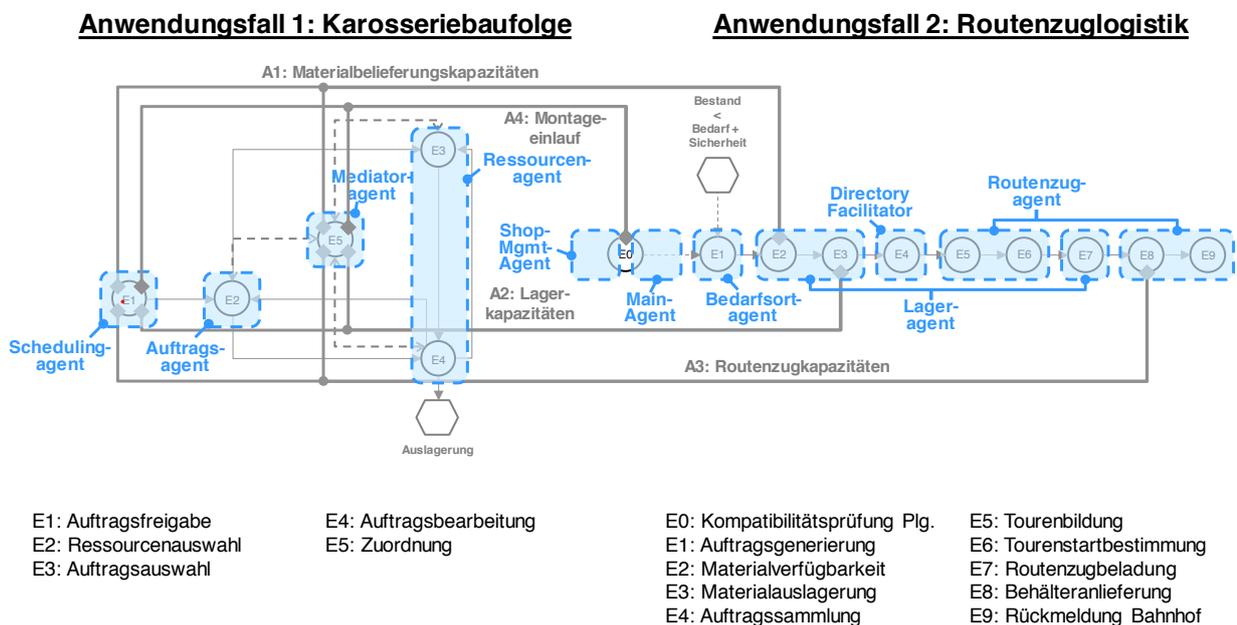


Abbildung 10-4: Übergreifendes Agentenmodell des integrierten Agentensystems

Damit der SMA und der MnA die Koordinationsentscheidungen ordnungsgemäß ausführen können, müssen sie ihre internen Interaktionen kapseln und Planungsstände untereinander austauschen. Daher definieren sich die Aufgaben der Agenten wie folgt:

Der **Shop-Management-Agent** ist dafür verantwortlich alle relevanten Steuerungsinformationen der Karosseriebaufolge zu konsolidieren. Darunter fällt einerseits die Erfassung der geplanten Bandaufgabe inkl. der wesentlichen Informationen wie geplante Start- und Endzeit, Produktvariante und Priorität der Aufträge. Andererseits muss er auch die tatsächliche Produktionsreihenfolge des Karosseriebaus erfassen und die prognostizierte Ausbringung in der richtigen Reihenfolge ermitteln. All dies sind Informationen, die für die Routenzugsteuerung relevant sind. Um sie der Routenzugsteuerung zur Verfügung zu stellen, muss der SMA diese Daten bewerten und in einer konsolidierten Form an den MnA übermitteln. Darüber hinaus muss der SMA auch

die Rückmeldungen des MnA verarbeiten und damit seine geplante Auftragsfreigabe und die Zuweisung von Aufträgen zu Ressourcen ggf. an die Anforderungen der Routenzugsteuerung anpassen.

Der **Main-Agent** auf der anderen Seite muss alle relevanten Steuerungsinformationen der Routenzugsteuerung konsolidieren, um sie der Karosseriefolgesteuerung zu übergeben. Dafür benötigt der MnA Informationen zu derzeitigen und zukünftigen Zuständen im Routenzugsystem wie den Materialbelieferungskapazitäten der internen und externen Lieferanten, den Lagerbeständen in den angebundenen Lagern, der Auslastung der Routenzüge sowie der Lage der Bedarfsorte in der Montagehalle. Mit diesen Informationen ausgestattet kann der MnA eine Bewertung der Kennwerte vornehmen und sie in konsolidierter Form als Fitnesswert an den SMA übergeben. Dieser zieht aus den Daten die notwendigen Schlüsse und passt die eigene Planung entsprechend an. Umgekehrt fließen auch die Meldungen des SMA in die Planung der Routenzugsteuerung ein.

In Tabelle 10-3 sind die Aufgaben der beiden Koordinationsagenten und die zu ihrer Ausführung benötigten Informationen zusammengetragen. Um sie erfolgreich umzusetzen muss darauf aufbauend die Kommunikation zwischen den Koordinationsagenten konzipiert werden.

Agententyp	Steuerungsaufgaben	Benötigte Steuerungsinformationen
Shop-Management-Agent (SMA)	<ul style="list-style-type: none"> • Relevante Steuerungsinformationen der Karosseriefolgesteuerung konsolidieren, bewerten und an die Routenzugsteuerung kommunizieren • Rückmeldungen der Routenzugsteuerung interpretieren, bewerten und in die Auftragsfreigabe und Ressourcenzuweisung einfließen lassen 	<ul style="list-style-type: none"> • Fitnesswert der Routenzugsteuerung • Geplante Bandauflage des Karosseriebaus • Auslastung der Ressourcen im Karosseriebau • Aufträge im Durchlauf des Karosseriebaus • Störungen im Karosseriebau • Sperrungen für den Karosseriebau
Main-Agent (MnA)	<ul style="list-style-type: none"> • Relevante Steuerungsinformationen der Routenzugsteuerung konsolidieren, bewerten und an die Karosseriefolgesteuerung kommunizieren • Meldungen der Karosseriefolgesteuerung aufnehmen, Simulationen durchführen, Fitnesswert ermitteln und zurückmelden 	<ul style="list-style-type: none"> • Geplante Bandauflage des Karosseriebaus • Tatsächliche Produktionsreihenfolge des Karosseriebaus • Materialbelieferungskapazitäten der Montage • Lagerbestände der Montage • Auslastung der Routenzüge • Lage der Bedarfsorte in der Montagehalle

Tabelle 10-3: Steuerungsaufgaben und -informationen im übergreifenden Agentenmodell

10.2.3 Übergreifendes Interaktionsmodell

Als letzter Schritt der erweiterten AOSE-Methode wird ein übergreifendes Interaktionsmodell für die zu integrierenden Agentensysteme erstellt. Um ein für den beabsichtigten Zweck passendes Interaktionsprotokoll bestimmen zu können, bedarf es zunächst der Klassifizierung der Koordinationsabhängigkeiten. Im Gegensatz zu der internen Kommunikation innerhalb der individuellen Agentensysteme erfüllt die separate Betrachtung der Koordinationsabhängigkeiten nicht den gewünschten Zweck, da eine getrennte Übermittlung der Informationen zu den einzelnen Abhängigkeiten zu unvorhersehbaren Effekten führen könnte (vgl. Abschnitt 10.2.2). Daher werden die

Koordinationsabhängigkeiten A1 bis A4 im Folgenden mithilfe eines Fitnesswertes zu einer einzigen Systemabhängigkeit zusammengefasst, die konsolidiert zwischen den Systemen ausgetauscht werden kann. Das dafür notwendige Interaktionsprotokoll wird nach den definierten Regeln der bekannten DACS-Methode klassifiziert.

Klassifizierung der Koordinationsabhängigkeiten

Der SMA und der MnA verfügen beide über direkten Zugang zu den für die Integration der beiden Agentensysteme notwendigen Informationen. Das verbleibende zu lösende Interaktionsproblem ist daher auf den Austausch von Nachrichten zwischen diesen beiden Agenten beschränkt. Wie in solchen Fällen üblich sind die Koordinationsabhängigkeiten in diesem Fall durch vergleichsweise einfache, bilaterale Kommunikationswege charakterisiert, bei denen jeweils immer *zwei Agenten involviert* sind. Entweder der SMA informiert den MnA über die geplante und tatsächliche Bandaufgabe und Produktionsreihenfolge oder der MnA informiert den SMA über den Fitnesswert, den er auf Basis dieser und weiterer interner Informationen ermittelt. Da die beiden Agentensysteme auf die übergeordnete Zielerreichung des Gesamtsystems ausgerichtet sind, existiert immer eine eindeutig beste Alternative, die von beiden Agenten bevorzugt wird. Die Präferenzen sind daher *kompatibel*. Da lediglich zwei Agenten in die Kommunikation eingebunden sind, sind die Präferenzen zudem als *nicht-lokal* zu klassifizieren. In jeder Abhängigkeit wird des Weiteren je *eine einzige Verpflichtung* eingegangen und dadurch, dass zwei Agenten an der Interaktion beteiligt sind hat die gemeinsame Verpflichtung die *Größe zwei*. Da jeder Agent nur eine Verpflichtung eingeht, ist das Kriterium der Beziehung zwischen den Verpflichtungen *irrelevant*. Die Rollenzuweisung steht bereits zur Designphase *fest* und erfordert keine weitere Rollenzuweisung während der Interaktion. Der Agent, der die Interaktion initiiert erbittet ferner nur eine einzige Handlung von dem anderen Agenten, sodass nur auf Ebene von *Alternativen* interagiert wird. Schließlich akzeptiert der initiiierende Agent alle gültigen Antworten ohne eine Möglichkeit zur Ablehnung zu haben, sodass eine *partielle Delegation* vorliegt. Die Merkmale der Koordinationsabhängigkeiten im integrierten Agentensystem sind in Tabelle 10-4 zusammengefasst.

Aspekt	Klassifikationskriterium	Ausprägungen
Startsituation	1. Anzahl involvierter Agenten	n=2
	2. Kompatibilität der Beschränkungen & Präferenzen	kompatible Präferenzen
	3. Globale Beschränkungen & Präferenzen	nicht-lokal
Gemeinsame Verpflichtungen	4. Anzahl gemeinsamer Verpflichtungen	n=1
	5. Größe gemeinsamer Verpflichtungen	n=2
	6. Beziehung zwischen gemeinsamen Verpflichtungen	-
Agentenrollen	7. Rollenzuweisung (Anzahl variable Rollen)	fest (keine)
Prozessanforderungen	8. Informationsverfügbarkeit	nur Alternativen
	9. Delegation von Entscheidungskompetenz	partielle Delegation

Tabelle 10-4: Klassifikation der Koordinationsabhängigkeiten im integrierten System

Auswahl und Anpassung eines Interaktionsprotokolls

Zur Auswahl eines geeigneten Protokolls für die notwendigen Interaktionen wurde in Analogie an das Vorgehen der anderen beiden Anwendungsfälle ein Abgleich mit der in Anhang 22 und 23 beigefügten Protokollbibliothek aus [BJW04] durchgeführt. Der detaillierte Abgleich der Anforderungen für den vorliegenden Anwendungsfall ist in Anhang 37 beigefügt.

Für die einfache, bilaterale Kommunikation von Informationen zwischen den Agentensystemen eignen sich erneut prinzipiell mehrere Protokolle. Aufgrund seiner Einfachheit und der hohen Übereinstimmung mit den geforderten Eigenschaften ist das Requesting Action Protokoll (RAP) jedoch anderen Vorgehen vorzuziehen. Mit Ausnahme der Delegation der Entscheidungskompetenz erfüllt es alle Auswahlkriterien und durch Weglassen einer finalen Zustimmung kann das RAP so angepasst werden, dass es den Anforderungen der Interaktion vollumfänglich gerecht wird. Da durch die Nutzung des RAP zudem der Kommunikationsaufwand gering gehalten wird, ist es das am besten geeignete Protokoll für den vorgesehenen Verwendungszweck.

Trotz seiner Einfachheit sind auch mit dem RAP grundsätzlich Schwingungen, gegenseitiges Aufschaukeln von Kennwerten durch unpassende Steuerungsentscheidungen und Schneeballeffekte möglich. Um dieses Risiko zu begrenzen, kann die Frequenz des Datenaustausches mithilfe von Regeln begrenzt werden. Im vorliegenden Fall sollte eine Frequenz von 5 Minuten zwischen zwei Datenübermittlungen genügen, um die oben genannten Effekte einzuschränken und gleichzeitig die Agilität nicht zu stark zu behindern. Da in der Automobilproduktion üblicherweise ein Produktionstakt von etwa einer Minute zugrunde liegt, werden in diesem Zeitraum maximal fünf Fahrzeuge produziert. Eine Reaktion auf eine Störung könnte also maximal diese Anzahl Fahrzeuge betreffen. Gleichzeitig lässt der Zeitpuffer den Systemen die Möglichkeit Simulationen und Bewertungen durchzuführen und vermeidet eine unkontrollierte Zunahme des Nachrichtenvolumens sowie der damit einhergehenden negativen Effekte. Die finale Größe dieses Zeitpuffes sollte in Simulationen getestet und entsprechend angepasst werden, damit sich das optimale Zeitintervall für den Austausch ergibt.

Übersicht der Kommunikationsflüsse

Die Konsolidierung der Informationen in den Einzelsystemen und der Informationsaustausch über das RAP ermöglichen bei Bedarf Anpassungen der berechneten Steuerungsentscheidungen im jeweils empfangenden Agentensystem vorzunehmen. In Abbildung 10-5 ist ein möglicher Kommunikationsfluss zwischen den beiden Koordinationsagenten sowie den ihnen unterstellten internen Agenten der jeweiligen Agentensysteme abgebildet.

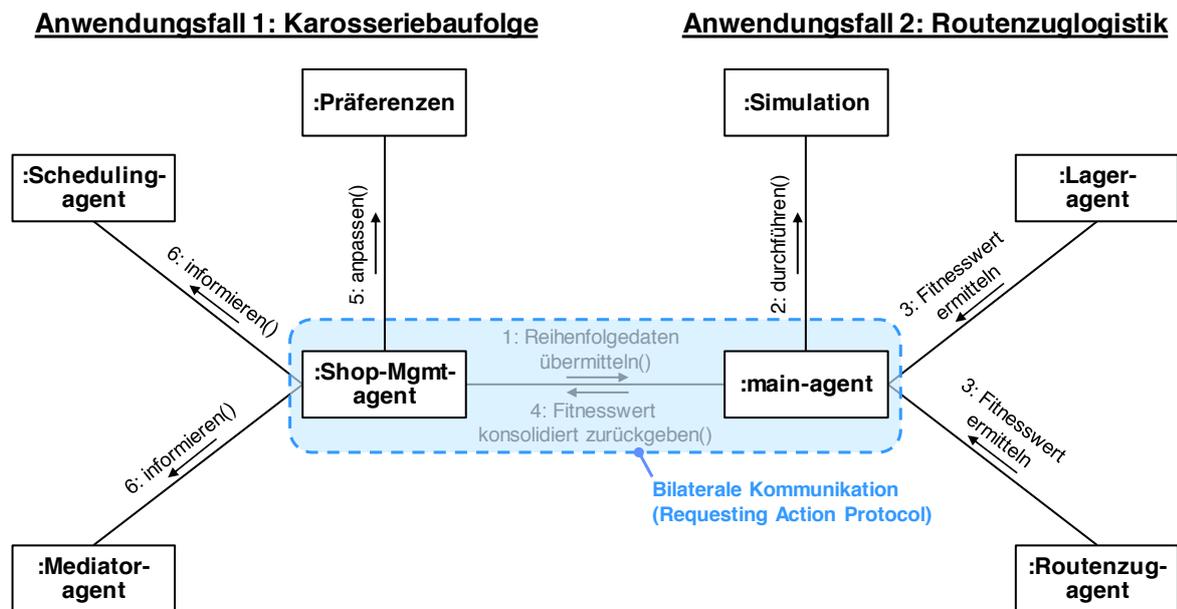


Abbildung 10-5: Übergreifendes Interaktionsmodell des integrierten Agentensystems

Der SMA und der MnA sind für das Management ihrer jeweiligen Agentensysteme verantwortlich und haben daher direkten Zugriff auf systeminterne Informationen. Der SMA in der Karosseriebaufolgesteuerung ist mit dem Scheduling- und dem Mediatoragenten verbunden, was ihn in die Position versetzt Informationen abzufragen und Anweisungen zu geben. Auf Seite der Routenzugsteuerung ist der MnA mit dem Lager- und Routenzugagenten verbunden, von denen er ebenfalls notwendige Informationen erhalten kann, die die Grundlage für interne Simulationen darstellen. Eine einzelne Informationsschleife zum Austausch von Informationen zwischen den Systemen besteht aus folgenden Schritten: Zunächst übermittelt der SMA die aktuellen Reihenfolgedaten aus dem Karosseriebau an den MnA. Dieser startet eine Simulation der internen Prozesse unter Zuhilfenahme der Daten von den Lager- und Routenzugagenten. Mit Abschluss der Simulation ist ein Fitnesswert ermittelt, der durch den MnA an den SMA übersendet wird. Der SMA passt auf Basis des Ergebnisses aus der Routenzugsteuerung seine internen Präferenzen an (z.B. Priorisierung von Produktvarianten, anzustrebender Produktmix für die nächste Zeiteinheit, Überholvorgänge zur Anpassung der Outputreihenfolge etc.) und übermittelt sie an den Scheduling- und Mediatoragenten. Mit diesen Informationen ausgestattet können diese Agenten wiederum alle notwendigen Steuerungsentscheidungen für den Zeitraum bis zum nächsten Informationsaustausch mit dem MnA treffen.

Um das Risiko von Schneeballeffekten zu reduzieren, sollten vor der Inbetriebnahme des konzipierten übergreifenden Agentensystems einerseits Unit Tests und andererseits eine Verifizierung der Funktionalität in einer Laufzeitsimulation durchgeführt werden. Diese umfangreiche und forschungstechnisch noch nicht ausreichend geklärte Frage bedarf weiterer Forschungsanstrengungen, die auf den hier erarbeiteten Ergebnissen aufsetzen können.

10.3 Ergebnisse und kritische Beurteilung

10.3.1 Ergebnisse

Die DACS-Methodik stellt ein Vorgehen für die Entwicklung von Agentensystemen zur Verfügung in denen eine definierte Anzahl an Steuerungsentscheidungen getroffen werden muss. Da die Gesamtheit der Steuerungsentscheidungen innerhalb eines komplexen Produktionssystems in der Regel zu groß ist, um in einem Schritt erfasst und durch ein einziges Agentensystem gesteuert zu werden, ist es erforderlich, separate Agentensysteme für verschiedene Steuerungsentscheidungen zu entwickeln und zu integrieren. Dafür wurde im Rahmen dieser Arbeit als Ergänzung zu DACS eine erweiterte AOSE-Methode entwickelt, die das ursprüngliche Vorgehen um drei weitere Schritte erweitert, die wie folgt definiert sind:

Im ersten Schritt werden die zusätzlich notwendigen kooperationsbezogenen Steuerungsentscheidungen und deren Abhängigkeiten identifiziert und modelliert. Darauf aufbauend findet die Erweiterung der ursprünglichen Entscheidungsmodelle statt. Dabei werden sowohl die Abhängigkeiten untereinander als auch zu den Steuerungsentscheidungen innerhalb der einzelnen Entscheidungsmodelle der zu integrierenden Agentensysteme berücksichtigt. In einem zweiten Schritt werden die Agentenmodelle erweitert, indem die zusätzlichen Steuerungsentscheidungen gebündelt und entweder auf bestehende Agenten innerhalb der zu integrierenden Agentensysteme oder auf zusätzliche Agenten abgebildet werden. Schließlich werden in einem dritten Schritt den notwendigen Wechselwirkungen zwischen den Steuerungsentscheidungen geeignete Interaktionsprotokolle zugeordnet. Diese Protokolle werden an die speziellen Bedürfnisse der zu treffenden Koordinationsentscheidungen angepasst und bei Bedarf erweitert. Damit ergibt sich eine erweiterte AOSE-Methode, mit der die Integration von Agentensystemen auf Basis von bereits bestehenden oder noch zu entwerfenden Agentensystemen realisiert werden kann.

Das erarbeitete Vorgehen wurde anhand von zwei Anwendungsfällen erprobt. Dafür wurden in den Kapiteln 7 und 8 zunächst zwei separate Agentensysteme für die Karosseriebausteuerung und die Routenzugsteuerung konzipiert, programmiert und getestet. Beide Systeme verantworten jeweils große Steuerungsaufgaben, stellen jedoch gleichzeitig nur einen Teil eines komplexeren Produktionssystems dar. Nach der Entwicklung und Erprobung der Agentensysteme mithilfe der drei Schritte von DACS wurden die ergänzenden drei Schritte der erweiterten AOSE-Methode angewendet um die Anwendungsfälle zu integrieren. Zunächst wurden dafür zusätzliche Koordinationsentscheidungen und deren Abhängigkeiten identifiziert. Im Beispielfall ist hierfür eine zusätzliche Steuerungsentscheidung zur Kombination der internen Planungen der beiden Systeme notwendig, sodass diese untereinander ausgetauscht und angepasst werden müssen. Die Karosseriefolgsteuerung übergibt die Freigabe- und Reihenfolgeinformationen, die für die Endmontage relevant sind, und die Routenzugsteuerung ermittelt auf dieser Basis einen Fitnesswert, der sich auf die Fähigkeit bezieht, alle Anforderungen der kommenden Karosserien zu erfüllen. Für

die benötigte Informationsvorverarbeitung wurde in jedem der Einzelsysteme ein Agent identifiziert, der die notwendigen Informationen konsolidiert, verarbeitet und an betroffene Agenten weiterleitet. Als letzter Schritt wurde für die Kommunikation zwischen den beiden Repräsentanten der jeweiligen Agentensysteme ein Interaktionsprotokoll definiert, mit dem der erforderliche Informationsaustausch stattfinden kann.

Das Ergebnis der Integration der beiden Anwendungsfälle ist in Abbildung 10-6 zusammengefasst. Weitere Details zu den Ergebnissen können in [ZaL17], [ZLC18] und [ZLR18] nachgelesen werden.

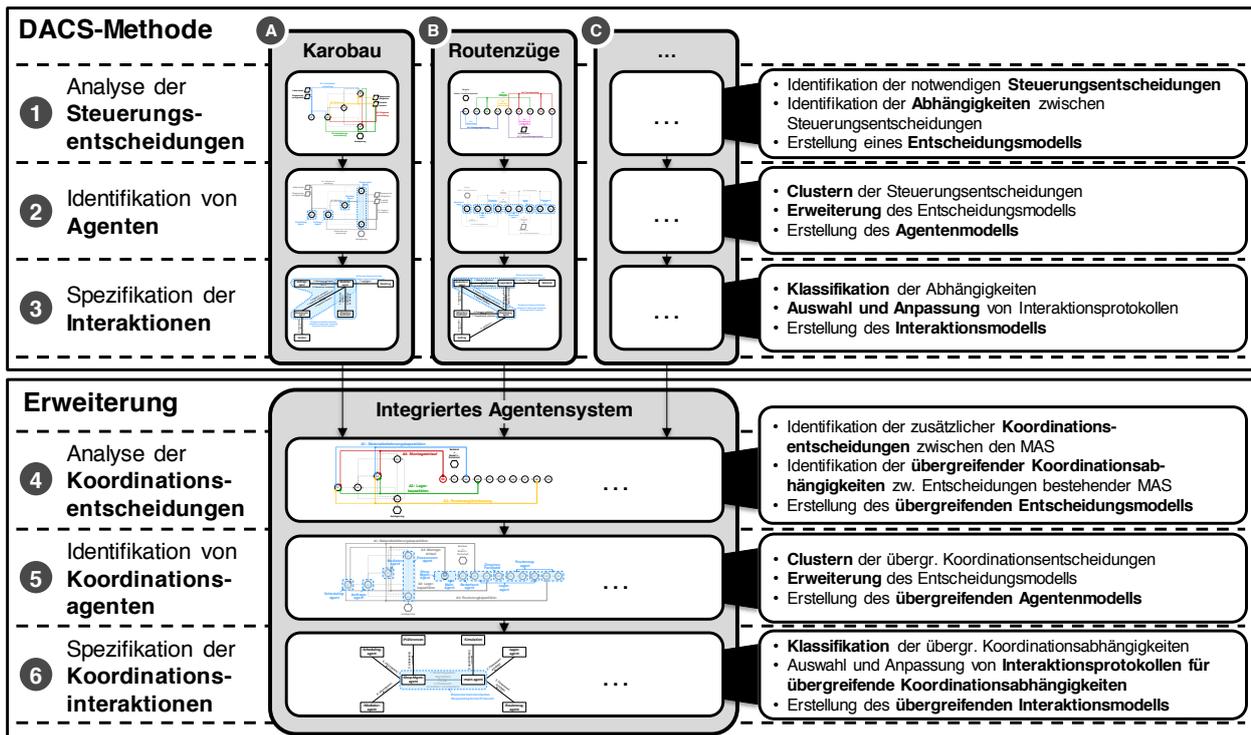


Abbildung 10-6: Ergebnisse der Integration der Anwendungsfälle

10.3.2 Kritische Beurteilung

Die in dieser Arbeit entwickelte erweiterte AOSE-Methode ermöglicht ein inkrementelles Vorgehen für die Realisierung und Integration umfassender agentenbasierter Steuerungssysteme innerhalb von komplexen Produktionsumgebungen der Fertigungsindustrie. Vor dem Hintergrund beschränkter Entwicklungskapazitäten und -budgets in Unternehmen werden dadurch die Hürden für eine Implementierung von Agentensystemen gesenkt und der Einzug von Agentensystemen in die Industrie erleichtert. Das Vorgehen basiert auf der verbreiteten und in zahlreichen Industrie- und Forschungsprojekten bewährten DACS-Methode, die eine strukturierte Erschließung von Steuerungsproblemen und die Entwicklung geeigneter Agentenarchitekturen ermöglicht. Die Erweiterung der DACS-Methode sieht eine schrittweise Realisierung individueller Agentensysteme und deren anschließende Integration vor. Dafür erfolgt zunächst die Analyse von Koordinationsentscheidungen und -abhängigkeiten, dann die Zuordnung dieser Entscheidungen zu dezidierten

Agenten und schließlich die Auswahl von Interaktionsprotokollen für die übergreifende Kommunikation der Agentensysteme. Falls während der Integration neue Informationen hinzukommen, die für das Zusammenspiel der Systeme relevant sind, können die individuellen und übergreifenden Modelle mithilfe der Methode mit wenig Aufwand angepasst werden. Dies birgt zwei Vorteile: Einerseits wird die Reife der Agentensysteme auf diese Weise nach und nach erhöht, andererseits wird das Verständnis des Systemverhaltens kontinuierlich verbessert. Da die Methode zudem auf bestehende Bibliotheken von Interaktionsprotokollen zurückgreift, ermöglicht sie ferner ein analytisches, quantifizierbares Vorgehen bei der Bewertung und Auswahl passender Kommunikationsprotokolle. Dies vereinfacht die vertikale und horizontale Integration von interagierenden Agentensystemen und die Abstimmung ihrer Zielstrukturen. Insbesondere können individuelle Ziele durch den Austausch von Informationen zwischen den Systemen auf die übergeordneten Ziele des Gesamtsystems ausgerichtet werden.

Zur Veranschaulichung der erweiterten ASOE-Methode wurden zwei Anwendungsfälle aus der Automobilfertigung ausgewählt. Dabei zeigte sich, dass ein entscheidender Aspekt für ihre erfolgreiche Anwendung ein detailliertes Verständnis der zu treffenden Steuerungsentscheidungen und der Abhängigkeiten zwischen ihnen ist. Ferner wurde deutlich, dass die Integration von sowohl vertikal als auch horizontal getrennten Wertschöpfungsschritten mit Herausforderungen verbunden ist. Im konkret vorliegenden Fall wurde erkannt, dass die Steuerung des Montageeinlaufs durch ein drittes System außerhalb des Betrachtungsumfangs erfolgt und somit nur indirekt durch die beiden betrachteten Systeme beeinflusst werden kann. Zudem kann die Integration gleichzeitig horizontal und vertikal getrennter Systeme dazu führen, dass bei der Abstimmung der Zielstrukturen den jeweiligen Systemen eine über- oder unterproportional große Bedeutung beigemessen wird. Grundsätzlich sollten daher zunächst Agentensysteme integriert werden, die direkten Einfluss aufeinander haben und möglichst viele der zu berücksichtigenden Abhängigkeiten abdecken. D.h. es sollte vermieden werden vertikale Ebenen oder horizontale Wertschöpfungsschritte bei der Integration zu überspringen.

Bei der Auswahl von zu integrierenden Steuerungssystemen und der Bewertung ihrer Integrierbarkeit sollte auch die Abfolge und die Richtung der Integration festgelegt werden. Horizontal ist grundsätzlich ein Vorgehen mit oder entgegen dem Wertschöpfungsfluss möglich. Vertikal kann die Integration top-down oder bottom-up erfolgen. Allgemein kann empfohlen werden den Integrationsprozess vom Endergebnis her zu initiieren, da die relevanten Kenngrößen wie Ausbringung, Qualität und Liefertreue der gesamten Wertschöpfungskette hier gemessen werden. Im vorliegenden Fall der Automobilproduktion entspricht dieser Punkt der Fertigstellung eines Fahrzeugs in der Endmontage. In Abhängigkeit der Komplexität der Umsetzbarkeit sowie der erwarteten Potenziale kann von hier aus beginnend entweder vertikal oder horizontal integriert werden. Die horizontale Integration entgegen dem Wertschöpfungsfluss ermöglicht es unter Umständen höhere Potenziale zu realisieren, kann jedoch auch mit einem hohen Aufwand verbunden sein.

Dennoch sollte es erstrebenswert sein zunächst mit der horizontalen Integration entgegen der Wertschöpfungsrichtung zu beginnen, da dies eine Abstimmung der Fertigungslinien untereinander forciert, wodurch bereichsübergreifende Hemmnisse abgebaut werden. Zudem wird durch dieses Vorgehen das Risiko gesenkt, dass wesentliche Abhängigkeiten übersehen werden, die hohe Nacharbeitskosten verursachen können. Erst anschließend sollte eine top-down-Integration der untergeordneten Systeme stattfinden. In Unternehmen, in denen dies vom Management nicht unterstützt wird oder technisch nicht möglich ist, kann eine Integration innerhalb von Silos – also einzelnen Wertschöpfungsschritten – der einfachere Weg sein. Auf diese Weise können bottom-up – auch ohne Grundsatzentscheidungen des Top-Managements – Leuchtturmprojekte realisiert werden, mit denen die Akzeptanz im Unternehmen erhöht werden kann.

Als noch zu lösende Herausforderung bei der Integration der beschriebenen Agentensysteme verbleibt die Abstimmung der Zielfunktionen, insbesondere bei konfliktären Zielgrößen. Dabei kann eine detaillierte Analyse der Steuerungs- bzw. Koordinationsentscheidungen sowie deren Priorisierung helfen, dieses Problem zu lösen. Ferner kann es hilfreich sein zwischen hierarchischen und heterarchischen Zielfunktionen zu unterscheiden. Zur Vermeidung von Schneeballeffekten und Kettenreaktionen müssen entsprechende Mechanismen implementiert werden, die bisher nicht ausreichend erforscht sind. In diesem Zusammenhang fehlen derzeit auch Methoden zur Ermittlung und dynamischen Anpassung der Gewichtungsfaktoren der einzelnen Zielfunktionswerte, um die Subsysteme stets auf die aktuell vorliegende Situation im Gesamtsystem auszurichten. Auch hierfür kommen selbstlernende Systeme basierend auf neuronalen Netzwerken in Frage, auf die im Ausblick noch eingegangen wird.

Die erarbeitete Entwicklungsmethode ermöglicht die schrittweise Entwicklung und Integration voneinander abhängiger Agentensysteme und stellt damit notwendige Werkzeuge für umfassende, vertikal und horizontal integrierte Steuerungsarchitekturen zur Verfügung. Ihre praktische Anwendbarkeit wurde anhand von zwei separat entwickelten und im Anschluss integrierten Anwendungsfällen demonstriert. Damit stellt die Methode eine Antwort auf die vierte und letzte Forschungsfrage dieser Arbeit dar und erfüllt die Anforderungen an die gestellte Zielsetzung. Abschließend erfolgt im letzten Kapitel dieser Arbeit eine Zusammenfassung und ein Ausblick auf offene Forschungsfragen und mögliche weiterführende Forschungsthemen.

11 Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurde das Ziel verfolgt eine AOSE-Methode zu erarbeiten, die eine schrittweise Entwicklung und Integration agentenbasierter Steuerungssysteme ermöglicht und damit den Weg für umfassendere, vertikal und horizontal integrierte Steuerungsarchitekturen im Fertigungsumfeld bereitet. Um dieses Ziel zu erreichen wurden vier Forschungsfragen aufgestellt und bearbeitet. Im folgenden Abschnitt 11.1 werden diese Forschungsfragen aufbauend auf den Erkenntnissen der vorangehenden Kapitel zusammenfassend beantwortet und die Lösungen im Hinblick auf bestehende Ansätze diskutiert. Die Arbeit schließt in Abschnitt 11.2 mit einem Ausblick auf verbleibende Forschungsfragen und Stoßrichtung für weitere Forschungsarbeiten.

11.1 Beantwortung der Forschungsfragen

In der Einleitung zu dieser Arbeit wurden in Kapitel 1 vier Forschungsfragen definiert, die ein schrittweises Vorgehen zur systematischen Erarbeitung der geforderten AOSE-Methode für die Integration von Agentensystemen darstellen. Im Folgenden werden diese Fragen zusammenfassend beantwortet und die erarbeiteten Lösungsansätze diskutiert.

Forschungsfrage 1:

Was sind relevante Ebenen für Industrie 4.0-Komponenten von Fertigungssystemen und wie definieren sie sich?

Für den Bereich der Fertigungsindustrie wurden neun Ebenen identifiziert, die eine Industrie 4.0-Komponente ausmachen. Die übergeordneten Ebenen aggregieren die darunterliegenden Ebenen und definieren sich, von unten nach oben, wie folgt: Konstruktionselemente, Komponenten, Funktionsgruppen, Arbeitsstationen, Arbeitseinheiten, Fertigungsabschnitte, Fertigungslinien, Fabriken und Fertigungsnetzwerke. Die aus diesen neuen Ebenen bestehende generische Fertigungssystemstruktur wurde anhand von mehreren Fertigungssystemen aus unterschiedlichen Branchen modelliert und validiert. Sie verfügt im Vergleich zu anderen Ansätzen wie ISA 95 und RAMI 4.0 über eine höhere Detaillierungstiefe, die sich bei der Entwicklung von Agentensystemen in den Kapiteln 7 und 8 als vorteilhaft erwiesen hat. Die Ebenenstruktur ermöglicht es verschiedene Subsysteme mit derselben Methode abzubilden und sie damit vergleichbar zu machen. Dies ist insbesondere für die Identifizierung von Steuerungsentscheidungen und -abhängigkeiten entscheidend. Basierend auf der Validierung und den Erkenntnissen dieser Arbeit ist eine breite Anwendbarkeit der generischen Fertigungssystemstruktur gegeben.

Forschungsfrage 2:

Welche Informationen müssen auf den verschiedenen Ebenen der virtuellen Repräsentation während der Inbetriebnahme- und Nutzungsphase zur bestmöglichen Ausführung komplexer, interdependenter Fertigungsprozesse verfügbar sein?

Jede Ebene der generischen Fertigungssystemstruktur verfügt über spezifische Steuerungsaufgaben und -informationen, die zur ordnungsgemäßen Ausführung ihrer Funktionen benötigt werden. Während Konstruktionselemente lediglich definierte Zustände annehmen und ihren Aktivitätsstatus kommunizieren können, sind Komponenten bereits in der Lage Ein- und Ausgangsinformationen zu verarbeiten und können daher auch parametrisiert werden. Funktionsgruppen erbringen feste, parametrierbare Prozesse sowie Überwachungs- und Diagnosefunktionen, die durch eine Mensch-Maschine-Schnittstelle angepasst werden können. Sie verfügen über Prozessparameter sowie Produktinformationen. Arbeitsstationen führen steuerungstechnisch festgelegte, aber parametrierbare und produktspezifische Prozesse zur Umsetzung einer definierten Funktionsmenge aus, die einer simultanen Zustandserfassung und Ansteuerung mehrerer Funktionsgruppen bedarf. Arbeitseinheiten enthalten alle notwendigen Informationen zur Realisierung einer definierten Prozessabfolge sowie zur Anlagenführung und -bedienung. Auf Ebene der Fertigungsabschnitte werden Informationen zur operativen Betriebsmittelsteuerung und -überwachung, Materialbestandsführung und Materialflusssteuerung sowie für Lagerabrufe benötigt, um übergeordnete Steuerungsentscheidungen zu treffen. Fertigungslinien verfügen über Steuerungsinformationen für operative Planungsfunktionen wie die Auslastungsvorschau und Personalplanung sowie externe Materialabrufe und das Monitoring untergeordneter Ressourcen und Kennzahlen. Die Fabrikebene verarbeitet taktisch-planerische Aufgaben zur Sicherstellung der Teileverfügbarkeit, vorbereitende bzw. kurzfristige Produktionsplanungsaufgaben zur Realisierung der Produktionsaufträge sowie operativ-überwachende Aufgaben zur Sicherstellung der Produktionseffektivität und -effizienz. Schließlich werden auf Netzwerkebene Informationen für strategische und taktische Fertigungsaufgaben wie z.B. die Festlegung der Absatz-/Produktionsprogramme und der Werkbelegung, die Kapazitätsplanung und Materialbedarfsermittlung, das Produktdaten- und Auftragsmanagement sowie die Baubarkeitsprüfung konsolidiert. Die Kenntnis der unterschiedlichen Arten von Steuerungsaufgaben und -informationen stellt eine Erleichterung bei der Agentifizierung von Steuerungssystemen dar und ist ein kritisches Element bei der Identifizierung aller Abhängigkeiten zwischen zu integrierenden Agentensystemen.

Forschungsfrage 3:

Wie können agentenbasierte Steuerungsarchitekturen die Leistungsfähigkeit komplexer Produktionssysteme verbessern, um den gegenwärtigen Herausforderungen hinsichtlich Flexibilität, Agilität und Autonomie gerecht zu werden?

Agentenbasierte Steuerungsarchitekturen können die Leistungsfähigkeit komplexer Produktionssysteme verbessern, indem sie durch die Dekomposition komplexer Optimierungsprobleme die notwendigen zu treffenden Steuerungsentscheidungen auf eine Vielzahl kleiner, aber agiler Agenten verteilen. Diese Agenten verfügen über einfache Regeln und Algorithmen, können aber durch die Reduzierung der zugrundeliegenden Problemkomplexität auf einen lokal abgrenzbaren

Bereich schneller proaktive und reaktive Handlungen ausführen als dies umfangreiche, zentralistische Ansätze können. Dennoch bedeutet dies nicht, dass die Lösungen von Agentensystemen grundsätzlich zu bevorzugen sind, denn bestehende Algorithmen des Operations Research können teilweise Lösungen generieren, die sehr nahe am Optimum liegen. Agenten benötigen für ihre Entscheidungen jedoch generell weniger Zeit, was vor allem in turbulenten Produktionssystemen mit vielen Einflussfaktoren hilfreich ist, bei denen sich Parameter kontinuierlich ändern können. Da in solchen Systemen die schnelle Reaktionsfähigkeit und Agilität bei aufkommenden Problemen höher zu gewichten ist als die Ermittlung einer vermeintlich optimalen Lösung, können Agentensysteme in einem solchen Umfeld einen großen Beitrag zur Glättung und Nivellierung des Produktionsflusses sowie zur Steigerung der Effizienz beitragen.

Um diese positiven Eigenschaften von Agentensystemen quantitativ nachzuweisen wurden zwei unterschiedliche Steuerungsaufgaben aus einem komplexen Produktionssystem der Automobilfertigung ausgewählt und als Agentensysteme modelliert. Der erste Anwendungsfall bezieht sich auf die Steuerung der Auftragsreihenfolge im Karosseriebau und der zweite auf die Materialbelieferung der Endmontagelinien mithilfe eines Routenzugfuhrparks. In beiden Anwendungsfällen wurde durch das Verfolgen des dezentralen Steuerungsansatzes die Flexibilität, Agilität und Autonomie der Produktionssysteme positiv beeinflusst. Auf Grundlage dieser Ergebnisse kann eine Empfehlung für die Umsetzung von Agentensystemen im Kontext der Fertigungsindustrie ausgesprochen werden. Um das zusätzliche Effizienzpotenzial zu heben, das in der Integration voneinander abhängiger Agentensysteme besteht, bedarf es des Weiteren einer schrittweisen Methode zur Verknüpfung der Einzelsysteme, die die folgende Forschungsfrage beantwortet.

Forschungsfrage 4:

Wie können voneinander abhängige agentenbasierte Steuerungssysteme miteinander verbunden werden, um umfassendere, vertikal und horizontal integrierte Steuerungsarchitekturen und Datendurchgängigkeit zu realisieren?

Für die Integration voneinander abhängiger Agentensysteme wurde auf Basis des bewährten Vorgehens zur Agentifizierung eigenständiger Systeme nach der DACS-Methode eine Erweiterung erarbeitet, die eine schrittweise Entwicklung von Agentensystemen für spezifische Steuerungsprobleme und ihre anschließende Zusammenführung ermöglicht. Die Erweiterung der Methode zur Integration voneinander abhängiger Agentensysteme besteht aus drei Schritten:

1. Analyse der Koordinationsentscheidungen:

Auf Basis der Entscheidungsmodelle vorhandener Agentensysteme werden alle im integrierten System zu treffenden Koordinationsentscheidungen identifiziert und gegebenenfalls ergänzt. Darauf aufbauend werden die zwischen den Koordinationsentscheidungen bestehenden Abhängigkeiten zusammengetragen und analysiert. Für diese Analyse müssen zusätzliche Kriterien herangezogen werden. Diese beinhalten einerseits die relative

Verortung der Entscheidungen in der Wertschöpfungskette und andererseits die Zielstruktur von Entscheidungen. Für deren Bewertung wurde ein Klassifikationsschema entworfen. Als Resultat des ersten Schrittes entsteht ein übergreifendes Entscheidungsmodell des integrierten Systems.

2. Identifikation von Koordinationsagenten:

Im zweiten Schritt werden die Koordinationsentscheidungen auf Basis des übergreifenden Entscheidungsmodells geclustert und es wird eine geeignete Agentenstruktur für die Lösung des Optimierungsproblems gesucht. Die Koordinationsentscheidungen können dabei entweder bestehenden Agenten zugewiesen werden oder es können für sie dezidierte Agenten entwickelt werden. Damit diese Koordinationsagenten ihre Rollen angemessen ausfüllen können, müssen darüber hinaus alle für die zu treffenden Entscheidungen benötigten Informationen identifiziert und analysiert werden. Als Ergebnis des zweiten Schrittes entsteht ein übergreifendes Agentenmodell des integrierten Steuerungssystems.

3. Spezifikation der Koordinationsinteraktionen:

Als dritter und letzter Schritt der AOSE-Methode findet die Konzeption der für die Koordinationsentscheidungen benötigten Interaktionen statt. Zunächst erfolgt dafür die Klassifikation der übergreifenden Koordinationsabhängigkeiten und die Definition der sich aus ihr ergebenden Interaktionsbedarfe. Einerseits können diese Bedarfe gedeckt werden indem die bestehenden Protokolle durch komplexere ersetzt werden, die den zusätzlichen Kommunikationsbedarf abbilden können. Andererseits können sie auch einfach durch zusätzliche Protokolle ergänzt werden, die sich ausschließlich um die Abwicklung der neuen Koordinationsbedarfe kümmern. Nachdem die Entscheidung für eines der Vorgehen getroffen ist, erfolgt schließlich die Auswahl und Anpassung von geeigneten Interaktionsprotokollen für die übergreifenden Koordinationsabhängigkeiten. Als Resultat des letzten Schrittes entsteht ein übergreifendes Interaktionsmodell.

Die erweiterte AOSE-Methode ermöglicht die schrittweise Realisierung und Zusammenführung individueller Agentensysteme zu einem integrierten Gesamtsystem, in dem die Abhängigkeiten und Zielfunktionen auf die Erreichung eines globalen Optimums abgestimmt werden. Als besondere Herausforderung hat sich dabei herausgestellt Systeme zu integrieren, die sowohl vertikal als auch horizontal voneinander getrennt sind. Daher wird bei der Integration ein Vorgehen empfohlen, das am Ende der Wertschöpfungskette ansetzt und zunächst eine horizontale Integration der obersten Ebenen verfolgt. Erst anschließend erscheint in einer Vielzahl komplexer Produktionssteuerungssysteme die vertikale Ausweitung der Integration sinnvoll.

Zusammenfassend ist mit der erweiterten AOSE-Methode ein systematisches Vorgehen für die schrittweise Migration hin zu integrierten cyber-physischen Produktionssystemen gegeben. Der beschriebene Ansatz wurde erfolgreich auf die beiden Anwendungsfälle der Karosseriefolge- und Routenzugsteuerung aus der Automobilindustrie angewendet, womit die Praxistauglichkeit der

Methode demonstriert ist. Dennoch zeigte sich, dass ein entscheidendes Kriterium für die erfolgreiche Anwendung der Methode ein detailliertes Verständnis der zu treffenden Steuerungsscheidungen und deren Abhängigkeiten ist. Sie müssen daher im Entscheidungsmodell vollständig erfasst und dokumentiert werden.

Abschließend wird im folgenden letzten Abschnitt dieser Arbeit ein Ausblick auf noch offene, mit dem Themengebiet zusammenhängende Fragen und mögliche Forschungsrichtungen gegeben.

11.2 Ausblick

Ziel der in dieser Arbeit entwickelten erweiterten AOSE-Methode ist es, die schrittweise Integration von agentenbasierten Steuerungssystemen in die Steuerung von Produktionssystemen der Fertigungsindustrie zu ermöglichen. Die hierbei betrachteten Anwendungsfälle entspringen der Automobilindustrie und wurden anhand von Beispielen aus anderen Branchen validiert. Als nächster Schritt könnten weitere Systeme aus der GFSS wie z.B. die Wochenprogrammbildung, JIT-Abrufe bei externen Lieferanten, die Cockpitvormontage oder weitere Materialflusssysteme wie die Routenzuglogistik anderer Bereiche oder Staplerleit- und Lagersysteme integriert werden.

Dem Autor sind keine wesentlichen Unterschiede bekannt, die eine Anwendung der vorgestellten Methoden in anderen Bereichen der Fertigungsindustrie behindern könnten. Die Erprobung und gegebenenfalls notwendige Adjustierung der Methode ist jedoch eine potenzielle Forschungslücke, die von anderen Forschungsarbeiten geschlossen werden sollte.

Um eine reibungslose Transition von zentralen zu dezentralen, agentenbasierten Steuerungssystemen in der Wirtschaft zu realisieren sowie organisatorische Hemmnisse bei der Implementierung aus dem Weg zu räumen ist es ferner zielführend das erarbeitete Vorgehen in einen geeigneten Migrationsansatz wie z.B. den von [CLC17] zu integrieren. Zudem sollte beachtet werden, dass der Einsatz von SysML-Modellen innerhalb der Optionsprüfungsphase, wie in [CLC17] vorgeschlagen, auch für den Fall einer schrittweisen Implementierung von Agentensystemen sinnvoll ist. Dadurch wird die klare und einfache Identifikation und Darstellung des Entscheidungsmodells ermöglicht. In diesem Kontext bedarf auch das explizite Vorgehen bei der Priorisierung des vertikalen und horizontalen Migrationspfades weiterer Untersuchungen. Da Migrationspfade stark von den spezifischen Steuerungsproblemen und der Unternehmenskultur abhängen, ist zudem eine weitere Erforschung der unternehmensorganisatorischen Rahmenbedingungen zielführend. Damit die Integrationsbemühungen sanfter und schneller ablaufen sind schließlich weitere Analysen sowie der Austausch von Praxiserfahrungen und Best Practices zwischen Unternehmen und Branchen sinnvoll.

Eine wesentliche Herausforderung für zukünftige Forschungsarbeiten ist des Weiteren die Sicherstellung des Gleichgewichts zwischen konfliktären Zielstrukturen interagierender Systemteile. Während hierarchische Zielstrukturen den Vorteil haben verhältnismäßig überschaubar zu sein und top-down kaskadiert werden zu können, sind heterarchische Zielstrukturen schwieriger

zu handhaben. Mit der in dieser Arbeit dargelegten AOSE-Methode ist ein wesentlicher Betrag zur Identifizierung der Koordinationsentscheidungen und -abhängigkeiten erfolgt. Weitere Forschungsanstrengungen könnten die noch bestehenden Lücken schließen, indem die vorgestellte Methode z.B. durch weitere Methoden und Werkzeuge für die frühzeitige Erkennung und Behebung von Stabilitätsproblemen erweitert oder ergänzt wird. Ziel sollte es dabei sein die Abhängigkeiten und Zielkonflikte zwischen den Systemen bei der Integration realitätsnah und in Echtzeit zu berücksichtigen sowie die Zielfunktionen im Gleichgewicht zu halten. Besonderes Augenmerk sollte zudem auf das Erkennen und Verhindern von Schwingungen und einem gegenseitigen Aufschaukeln von Koordinationsentscheidungen gelegt werden, um Schneeballeffekte analog des Bullwhip-Effektes aus der Supply-Chain zu vermeiden.

Ein kurzfristiger Lösungsansatz könnte in der Implementierung von automatischen Abschaltmechanismen bestehen, wie sie bereits im High-Frequency-Trading aus der Finanzwirtschaft bekannt sind. Mittelfristig könnte ferner die Entwicklung von KI-basierten Kontrollsystemen einen erfolgsversprechenden Lösungsansatz darstellen. Sie könnten zwischen den Bedarfen und Prioritäten der jeweiligen Agentensysteme abwägen und situationsbedingt Gewichtungsfaktoren und sonstige Stellgrößen anpassen, um unwirtschaftliche Entscheidungen abzufangen und die Systeme noch stärker auf die gemeinsame Zielerreichung des Gesamtsystems ausrichten. Um die komplexen Funktionen und Abhängigkeiten zwischen den Systemen nachzubilden erscheinen Strategien des maschinellen Lernens wie das auf neuronalen Netzwerken basierende Deep Learning oder das Reinforcement Learning als besonders aussichtsreiche Ansätze. Dabei können mathematische Werkzeuge wie die Softmax-Funktion oder der Stochastic Gradient Descent, die aus der KI-Forschung und -Praxis bekannt sind, eine wesentliche Rolle spielen. Dieselben Technologien könnten ferner eingesetzt werden, um relevante Steuerungsparameter besser einzuschätzen bzw. zu prognostizieren. Als Beispiel dafür ist die Dauer von Anlagenstörungen, Lieferengpässen und sonstigen unvorhergesehenen Ereignissen ein großes Optimierungsfeld, da diese Daten den Steuerungssystemen derzeit oft nicht zur Verfügung stehen oder zu ungenau sind. Eine Weiterentwicklung dieses Ansatzes könnte in der Vorhersage von aufkommenden Störungen in Analogie an die vorausschauende Instandhaltung sein. Wenn eine solche vorausschauende Produktionssteuerung die betroffenen Agenten präzise Störungsinformationen zur Verfügung stellen könnte, würde dies die Steuerung der relevanten Systemparameter erheblich erleichtern und bisher verborgene Effizienzpotenziale in der gesamten Wertschöpfungskette offenlegen.

Unabhängig vom Lösungsansatz bleibt festzuhalten, dass weitere Untersuchungen hinsichtlich der Stabilität interagierender Systeme notwendig sind. Dies ist ein weites Forschungsfeld, das im Rahmen weiterer Forschungsarbeiten behandelt werden sollte. Die Ergebnisse der vorliegenden Arbeit bilden die Grundlage für das weitere Vorgehen in diese Richtung.

Literaturverzeichnis

- [Ada88] Adam, D.: Grundlagen der Produktionsplanung und -steuerung, 1988
- [AdM12] Adhau, S.; Mittal, M. L.: A Multiagent Based System for Resource Allocation and Scheduling of Distributed Projects. In International Journal of Modeling and Optimization, 2012; S. 524–528
- [AIF18] AIF: Industrie du Futur. <http://www.industrie-dufutur.org/>, 21.04.2018
- [AKS07] Adams, M. et al.: DIN EN 62264. Die neue Norm zur Interoperabilität von Produktion und Unternehmensführung – Teil 1. In atp, 2007, 49; S. 52–57
- [AMS08] Asnar, Y. et al.: Risk as Dependability Metrics for the Evaluation of Business Solutions. A Model-driven Approach. In (Jakoubi, S. Hrsg.): Proceedings of the Third International Conference on Availability, Security, and Reliability. March 4 - 7, 2008, Barcelona, Spain ; [including workshops papers]. IEEE Computer Society, Los Alamitos, Calif., 2008; S. 1240–1247
- [AnB11] Antonova, I.; Batchkov, I.: Development of Multi-Agent Control Systems using UML/SysML. In (Alkhateeb, F. Hrsg.): Multi-Agent Systems - Modeling, Control, Programming, Simulations and Applications. InTech, 2011
- [Ash56] Ashby, W. R.: An Introduction to Cybernetics. Chapman & Hall, London, 1956 (1999)
- [Aud18] Audi AG: Audi hautnah erleben. Erlebnisführungen im Werk Ingolstadt. http://www.audi.de/etc/medialib/ngw/brand/audi_forum_ingolstadt/pdf.Par.0002.File.pdf/afi-erlebnisuehrungen.pdf, 07.03.2018
- [Ay06] Ay, N.: Prinzipien der Robustheit, Leipzig, 2006
- [Bäu09] Bäumann, L.: Leitlinie Perlenkette 2.0, Wolfsburg, 2009
- [Bau12] Bauernhansl, T.: Industrie 4.0: Das Ende der Kontrolle
- [Bau13] Bauernhansl, T.: Industrie 4.0 - Herausforderungen und Chancen für die Automobilindustrie. K-P Managementkonferenz 2013, 2013
- [Bau14] Bauernhansl, T.: Die Vierte Industrielle Revolution – Der Weg in ein wertschaffendes Produktionsparadigma. In (Bauernhansl, T.; Hompel, M. t.; Vogel-Heuser, B. Hrsg.): Industrie 4.0 in Produktion, Automatisierung und Logistik. Springer Fachmedien Wiesbaden, Wiesbaden, 2014; S. 5–35
- [BBB14] Bloech, J. et al.: Einführung in die Produktion. Springer Gabler, Berlin, 2014
- [BCG08] Bellifemine, F.; Caire, G.; Greenwood, D. P. A.: Developing multi-agent systems with JADE. Wiley, Chichester, 2008

- [BCG10] Buzacott, J. A. et al.: Produktionsplanung und -steuerung. Grundlagen, Konzepte und integrative Entwicklungen. Oldenbourg Verlag, München, 2010
- [BDR09] Bangemann, T. et al.: Integration of Automation Devices in Web Service supporting Systems. In Preprints of the 30th IFAC Workshop on Real-Time Programming and 4th International Workshop on Real-Time Software, S. 161-166, 2009
- [Bec16] Becker, A.: Konzeption eines Agentensystems für die Fertigungssteuerung im Automobilbau am Beispiel des Werkes Wolfsburg, 2016
- [Bee01] Beedle, M. e. a.: Manifesto for Agile Software Development. <http://agilemanifesto.org/>, 10.03.2018
- [Bes14] Beste, D.: Esperanto für die Fabrik 4.0. Produktion + Logistik. <http://www.springer-professional.de/esperanto-fuer-die-fabrik-40/4896908.html>, 23.09.2014
- [BFS07a] Boysen, N.; Fliedner, M.; Scholl, A.: Produktionsplanung bei Variantenfließfertigung. Planungshierarchie und Elemente einer Hierarchischen Planung. In Zeitschrift für Betriebswirtschaft, 2007, 77; S. 759–793
- [BFS07b] Boysen, N.; Fliedner, M.; Scholl, A.: Level-Scheduling bei Variantenfließfertigung. Klassifikation, Literaturüberblick und Modellkritik. In Journal für Betriebswirtschaft, 2007, 57; S. 37–66
- [BFS09] Boysen, N.; Fliedner, M.; Scholl, A.: Sequencing mixed-model assembly lines. Survey, classification and model critique. In European Journal of Operational Research, 2009, 192; S. 349–373
- [BGG02] Bresciani, P. et al.: Troops: An agent-oriented software development methodology. Technical Report DIT-02-0015, 2002
- [BHS13] Birkert, A. R.; Haage, S.; Straub, M.: Umformtechnische Herstellung komplexer Karosserieteile. Auslegung von Ziehanlagen. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013
- [BHV14] Bauernhansl, T.; Hompel, M. t.; Vogel-Heuser, B.: Industrie 4.0 in Produktion, Automatisierung und Logistik. Springer Fachmedien Wiesbaden, Wiesbaden, 2014
- [BHW96] Brucker, P.; Hurink, J.; Werner, F.: Improving local search heuristics for some scheduling problems - Part I. In Discrete Applied Mathematics, 1996, 65; S. 97–122
- [BHW97] Brucker, P.; Hurink, J.; Werner, F.: Improving local search heuristics for some scheduling problems - Part II. In Discrete Applied Mathematics, 1997, 72; S. 47–69
- [Bir14] Birolini, A.: Reliability Engineering. Theory and Practice. Springer Berlin Heidelberg, Berlin, Heidelberg, s.l., 2014

- [BIT14] BITKOM e.V.: Studie Industrie 4.0. Volkswirtschaftliches Potenzial für Deutschland, 2014
- [BJW01] Bussmann, S.; Jennings, N. R.; Wooldridge, M.: On the Identification of Agents in the Design of Production Control Systems. In (Goos, G. et al. Hrsg.): Agent-Oriented Software Engineering. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001; S. 141–162
- [BJW04] Bussmann, S.; Jennings, N. R.; Wooldridge, M.: Multiagent Systems for Manufacturing Control. A Design Methodology. Springer, Berlin, Heidelberg, 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, 2010; S. 113–138
- [BMB16] BMBF: Industrie 4.0. <https://www.bmbf.de/de/zukunftsprojekt-industrie-4-0-848.html>, 21.04.2018
- [BMG08] Badr, I.; Mubarak, H.; Göhner, P.: Extending the MaSE Methodology for the Development of Embedded Real-Time Systems. In (Carbonell, J. G. et al. Hrsg.): Languages, Methodologies and Development Tools for Multi-Agent Systems. First International Workshop, LADS 2007, Durham, UK, September 4-6, 2007. Revised Selected Papers. Springer, Berlin, Heidelberg, 2008; S. 106–122
- [BMK16] Block, C.; Morlock, F.; Kühlenkötter, B.: Ganzheitliche flexible Vernetzung durch Erweiterung bestehender IT-Strukturen zu Serviceorientierten Architekturen mithilfe von Agentensystemen zur humanzentrierten Entscheidungsunterstützung. Ein Konzept zur RAMI Umsetzung. In (Schlick, C. Hrsg.): Megatrend Digitalisierung - Potenziale der Arbeits- und Betriebsorganisation. GITO, Berlin, 2016
- [BoG08] Botti, V.; Giret, A.: ANEMONA. A Multi-agent Methodology for Holonic Manufacturing Systems. Springer Verlag London Limited, s.l., 2008
- [Bog13] Bogon, T.: Agentenbasierte Schwarmintelligenz. Zugl.: Trier, Univ., Diss., 2012. Springer Vieweg, Wiesbaden, 2013
- [Boh98] Bohn, M.: Toleranzmanagement im Entwicklungsprozeß. Reduzierung der Auswirkungen von Toleranzen auf Zusammenbauten der Automobil-Karosserien, 1998
- [Boo07] Boomgaarden, C.: Dynamische Tourenplanung und -steuerung. Dissertation, Pusan, 2007
- [Bor13] Borshchev, A.: The Big Book of Simulation Modeling. Multimethod Modeling with AnyLogic 6. AnyLogic, North America, 2013

- [Bos14] Bostrom, N.: Superintelligence. Paths, dangers, strategies. Oxford Univ. Press, Oxford u.a., 2014
- [Boy05] Boysen, N.: Variantenfließfertigung. Zugl.: Hamburg, Univ., FB Wirtschaftswiss., Diss., 2005. Dt. Univ.-Verl., Wiesbaden, 2005
- [BoZ13] Boysen, N.; Zenker, M.: A decomposition approach for the car resequencing problem with selectivity banks. In Computers & Operations Research, 2013, 40; S. 98–108
- [BPS03] Beck, J. C.; Prosser, P.; Selensky, E.: Vehicle routing and job shop scheduling: what's the difference?: Proceedings of the 13th International Conference on Artificial Intelligence Planning and Scheduling, Trento, Italy, 2003; S. 267–276
- [Bra14] Brandt, M.: Dramatischer Preisverfall bei Festplattenspeichern. <https://de.statista.com/infografik/2544/entwicklung-preis-pro-gigabyte-festplattenspeicher/>, 15.08.2018
- [BrK12] Brucker, P.; Knust, S.: Complex scheduling. Springer-Verlag Berlin Heidelberg, Heidelberg, New York, 2012
- [Bru07] Brucker, P.: Scheduling algorithms. With 32 tables. Springer, Berlin, 2007
- [BSW12] Boysen, N.; Scholl, A.; Wopperer, N.: Resequencing of mixed-model assembly lines. Survey and research agenda. In European journal of operational research: EJOR, 2012, 216; S. 594–604
- [Bus07] Bush, J. W.: Ward's world motor vehicle data 2007. Ward's Automotive group, Southfield, 2007
- [Bus12] Bussmann, S.: Software Agents - Production 2000+. <http://www.stefan-bussmann.de/en/agents/p2plessons.html>, 25.10.2015
- [BVZ15] BITKOM e.V.; VDMA e.V.; ZVEI e.V.: Umsetzungsstrategie Industrie 4.0. Ergebnisbericht der Plattform Industrie 4.0, 2015
- [Cas17] Castelvechi, D.: IBM's quantum cloud computer goes commercial. In Nature, 2017, 543; S. 159
- [CCF11] Castellini, P. et al.: Towards the integration of process and quality control using multi-agent technology: Proceedings IECON 2011. 37th Annual Conference of the IEEE Industrial Electronics Society, Crown Conference Centre, Melbourne, Australia, 07-10 November, 2011. IEEE, Piscataway, NJ, 2011; S. 421–426

- [CeZ09] Cernuzzi, L.; Zambonelli, F.: Gaia4E: A Tool Supporting the Design of MAS using Gaia. In (ICEIS Hrsg.): Proceedings of the 11th International Conference on Enterprise Information Systems, Volume SAIC, Milan, Italy, May 6-10, 2009, 2009; S. 82–88
- [CeZ11] Cernuzzi, L.; Zambonelli, F.: Improving comparative analysis for the evaluation of AOSE methodologies. In International Journal of Agent-Oriented Software Engineering, 2011, 4; S. 331
- [CJS04] Cernuzzi, L. et al.: The Gaia Methodology. Basic Concepts and Extensions. In (Bergenti, F.; Zambonelli, F.; Gleizes, M.-P. Hrsg.): Methodologies and software engineering for agent systems. The agent-oriented software engineering handbook. Kluwer Academic, Boston Mass., London, 2004; S. 69–88
- [CKG10] Chisu, R.; Kuzmany, F.; Günthner, W. A.: Realisierung einer agentenbasierten Steuerung für Elektrohängebahnsysteme. In (Günthner, W.; Hompel, M. t. Hrsg.): Internet der Dinge in der Intralogistik. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2010; S. 263–274
- [CKM15] Colombo, A. W. et al.: Industrial Agents in the Era of Service-Oriented Architectures and Cloud-Based Industrial Infrastructures. In (Leitão, P.; Karnouskos, S. Hrsg.): Industrial Agents: Emerging Applications of Software Agents in Industry. Elsevier, Amsterdam, Oxford, Waltham, 2015; S. 67–87
- [Cla05] Clarke, C.: Automotive Production Systems and Standardisation. From Ford to the Case of Mercedes-Benz. Physica-Verlag Heidelberg, Heidelberg, 2005
- [CLC17] Cala, A. et al.: Migration from traditional towards cyber-physical production systems: 2017 IEEE 15th International Conference on Industrial Informatics (INDIN). University of Applied Science Emden/Leer, Emden, Germany, 24-26 July 2017 proceedings. IEEE, Piscataway, NJ, 2017; S. 1147–1152
- [CIW64] Clarke, G.; Wright, J. W.: Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. In Operations Research, 1964, 12; S. 568–581
- [CMO11] Cernuzzi, L. et al.: Adaptable Multi-Agent Systems. The Case of the Gaia Methodology. In International Journal of Software Engineering and Knowledge Engineering, 2011, 21; S. 491–521
- [Cri11] Crimson Consulting Group: Oracle SOA vs. IBM SOA White Paper. <http://www.oracle.com/us/technologies/soa/oracle-soa-vs-ibm-soa-345791.pdf>, 07.05.2018
- [CTM07] Cavalieri, S.; Terzi, S.; Macchi, M.: A Benchmarking Service for the evaluation and comparison of scheduling techniques. In Computers in Industry, 2007, 58; S. 656–666

- [DaC10] Daniluk, D.; Chisu, R.: Simulation und Emulation im Internet der Dinge. In (Günthner, W.; Hompel, M. t. Hrsg.): Internet der Dinge in der Intralogistik. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2010; S. 149–170
- [DaR59] Dantzig, G. B.; Ramser, J. H.: The Truck Dispatching Problem. In Management Science, 1959, 6; S. 80–91
- [DaW04] Dam, K. H.; Winikoff, M.: Comparing Agent-Oriented Methodologies. In (Kanade, T. et al. Hrsg.): Agent-Oriented Information Systems. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004; S. 78–93
- [DBH06] Dam, K. H. van et al.: Modelling an electricity infrastructure as a multi-agent system — Lessons learnt from manufacturing control. In (Marquardt, W.; Pantelides, C. Hrsg.): 16th European Symposium on Computer-Aided Process Engineering and 9th International Symposium on Process System Engineering. Elsevier, Amsterdam, 2006; S. 1741–1746
- [DeO10] DeLoach, S. A.; Ojeda, J. C. G.: O-MaSE. A customisable approach to designing and building complex, adaptive multi-agent systems. In International Journal of Agent-Oriented Software Engineering, 2010, 4; S. 244
- [DFS12] Downing, N.; Feydy, T.; Stuckey, P. J.: Explaining Flow-Based Propagation. In (Beldiceanu, N.; Jussien, N.; Pinson, É. Hrsg.): Integration of AI and OR techniques in constraint programming for combinatorial optimization problems. 9th international conference, CPAIOR 2012, Nantes, France, May 28 - June 1, 2012 ; proceedings. Springer, Berlin, 2012; S. 146–162
- [DGA14] Dewitz, M.; Günthner, W. A.; Arlt, T.: Fahrplanoptimierung für innerbetriebliche Routenverkehre. Wissenschaftliche Gesellschaft für Technische Logistik
- [DHL08] Dam, K. H. van et al.: Agent-based control of distributed electricity generation with micro combined heat and power—Cross-sectoral learning for process and infrastructure engineers. In Computers & Chemical Engineering, 2008, 32; S. 205–217
- [DIN00] DIN e.V. Bratke: DIN EN 61512-1: Chargenorientierte Fahrweise - Teil 1: Modelle und Terminologie. Beuth Verlag GmbH, Berlin, 2000
- [DIN03] DIN e.V.: DIN 8580: Fertigungsverfahren – Begriffe, Einteilung. Beuth Verlag GmbH, Berlin, 2003
- [DIN04] DIN e.V.: DIN EN 61131-1: Speicherprogrammierbare Steuerungen - Teil 1: Allgemeine Informationen. Beuth Verlag GmbH, Berlin, 2004
- [DIN08] DIN e.V.: DIN EN 61131-2: Speicherprogrammierbare Steuerungen - Teil 2: Betriebsmittelanforderungen und Prüfungen. Beuth Verlag GmbH, 2008

- [DIN14a] DIN e.V.: DIN EN 62264-1: Integration von Unternehmensführungs- und Leitsystemen – Teil 1: Modelle und Terminologie. Beuth Verlag GmbH, Berlin, 2014
- [DIN14b] DIN e.V.: DIN EN 61499-1: Funktionsbausteine für industrielle Leitsysteme – Teil 1: Architektur, 2014
- [DIN14c] DIN e.V.: DIN EN 61131-3: Speicherprogrammierbare Steuerungen – Teil 3: Programmiersprachen. Beuth Verlag GmbH, Berlin, 2014
- [DIN16] DIN e.V. DIN e.V.: DIN SPEC 91345: Referenzarchitekturmodell Industrie 4.0 (RAMI4.0). Beuth Verlag GmbH, Berlin, 2016
- [DIN18] DIN e.V.: Deutsche Normungsroadmap Industrie 4.0 – Version 3.
<https://www.dke.de/resource/blob/778174/7b1a6b3764d64b8669d97cba2852bc73/deutsche-normungsroadmap-industrie-4-0-version-3-0-data.pdf>, 15.08.2018
- [DKR08] Dangelmaier, W. et al.: Aspects of Agent Based Planning in the Demand Driven Railcab Scenario. In (Haasis, H.-D.; Kreowski, H.-J.; Scholz-Reiter, B. Hrsg.): Dynamics in Logistics. First International Conference, LDIC 2007. Springer, Berlin, 2008; S. 171–178
- [DLH11] Diedrich, C.; Lüder, A.; Hundt, L.: Bedeutung der Interoperabilität bei Entwurf und Nutzung von automatisierten Produktionssystemen. In at – Automatisierungstechnik, 2011, 59; S. 426–438
- [Dör13] Dörmer, J.: Produktionsprogrammplanung bei variantenreicher Fließproduktion. Untersucht am Beispiel der Automobilendmontage. Springer Gabler, Wiesbaden, 2013
- [DoS10] Domschke, W.; Scholl, A.: Logistik. Rundreisen und Touren. Oldenbourg, München, 2010
- [DOS13] Dieter Spath (Hrsg.) et al.: Produktionsarbeit der Zukunft - Industrie 4.0. [Studie]. Fraunhofer-Verl., Stuttgart, 2013
- [DPD15] Demin, E. et al.: IEC 61499 Distributed Control Enhanced with Cloud-based Web-Services: 2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA). 15 - 17 June 2015, Auckland, New Zealand. IEEE, Piscataway, NJ, 2015; S. 972–977
- [Dru66] Drucker, P. F.: Das Grossunternehmen. Sinn, Arbeitsweise u. Zielsetzung in unserer Zeit. Econ-Verl., Düsseldorf, 1966
- [DSV93] Domschke, W.; Scholl, A.; Voß, H.: Produktionsplanung. Ablauforganisatorische Aspekte. Springer-Verlag, Berlin, Heidelberg, 1993

- [Duh16] Duhigg, C.: Smarter faster better. The secrets of being productive in life and business, 2016
- [DWS01] DeLoach, S. A.; Wood, M. F.; SPARKMAN, C. H.: Multiagent Systems Engineering. In International Journal of Software Engineering and Knowledge Engineering, 2001, 11; S. 231–258
- [EHH10] Elger, J. et al.: Chancen und Herausforderungen von dezentral gesteuerten Flughafen-Gepäckförderanlagen. In (Günthner, W.; Hompel, M. t. Hrsg.): Internet der Dinge in der Intralogistik. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2010; S. 275–294
- [Elb87] Elbracht, D. Hrsg.: Flexibilität und Wirtschaftlichkeit neuer Technologien in der Produktion. Tagungsbericht [zum Symposium "Strategische Konzepte für die Flexible und Wirtschaftliche Fabrik der Zukunft"], München, 1987
- [ElH10] Elger, J.; Haußner, C.: Entwicklungen in der Automatisierungstechnik. In (Günthner, W.; Hompel, M. t. Hrsg.): Internet der Dinge in der Intralogistik. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2010; S. 23–27
- [EIM05] ElMaraghy, H. A.: Flexible and reconfigurable manufacturing systems paradigms. In International Journal of Flexible Manufacturing Systems, 2005, 17; S. 261–276
- [EIM09] ElMaraghy, H.A. Hrsg.: Changeable and Reconfigurable Manufacturing Systems. Springer London, Guildford, Surrey, 2009
- [Erk11] ErKayhan, S.: Ein Vorgehensmodell zur automatischen Kopplung von Services am Beispiel der Integration von Standardsoftwaresystemen. Zugl.: Karlsruhe, KIT, Diss., 2011. Technische Informationsbibliothek u. Universitätsbibliothek; KIT Scientific Publishing, Hannover, Karlsruhe, 2011
- [ESE13] ElMaraghy, H. et al.: Product variety management. In CIRP Annals - Manufacturing Technology, 2013, 62; S. 629–652
- [Eva16] Evans, J.: Mastering Chaos. A Netflix Guide to Microservices. <https://www.infoq.com/presentations/netflix-chaos-microservices>, 06.05.2018
- [Eve89] Eversheim, W.: Organisation in der Produktionstechnik Band 4. Fertigung und Montage. Springer Berlin Heidelberg, Berlin, Heidelberg, 1989
- [Eve96] Eversheim, W.: Produktionstechnik und -verfahren. [Verlag nicht ermittelbar], [Erscheinungsort nicht ermittelbar], 1996
- [FaM07] Farid, A.; McFarlane, D.: Design Structure Matrix Based Method for Reconfigurability Measurement of Distributed Manufacturing Systems. In International Journal of Intelligent Control and Systems, 2007; S. 1–12

- [FaR15] Farid, A. M.; Ribeiro, L.: An Axiomatic Design of a Multiagent Reconfigurable Mechatronic System Architecture. In IEEE Transactions on Industrial Informatics, 2015, 11; S. 1142–1155
- [FeL10] Feldhorst, S.; Libert, S.: Software-Methoden für die Automatisierung. In (Günthner, W.; Hompel, M. t. Hrsg.): Internet der Dinge in der Intralogistik. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2010; S. 29–37
- [FLP04] Fukasawa, R. et al.: Robust Branch-and-Cut-and-Price for the Capacitated Vehicle Routing Problem. In (Bienstock, D.; Nemhauser, G. Hrsg.): Integer Programming and Combinatorial Optimization. 10th International IPCO Conference, New York, NY, USA, June 7-11, 2004. Proceedings. Springer, Berlin, Heidelberg, 2004; S. 1–15
- [FML03] Fletcher, M. et al.: The Cambridge Packing Cell — A Holonic Enterprise Demonstrator. In (Mařík, V.; Pěchouček, M.; Müller, J. Hrsg.): Multi-Agent Systems and Applications III. 3rd International Central and Eastern European Conference on Multi-Agent Systems, CEEMAS 2003 Prague, Czech Republic, June 16-18, 2003 Proceedings. Springer, Berlin, Heidelberg, 2003; S. 533–543
- [FMV18] Fischer, J.; Marcos, M.; Vogel-Heuser, B.: Model-based development of a multi-agent system for controlling material flow systems. In at - Automatisierungstechnik, 2018, 66; S. 438–448
- [For15] Fordiac: IEC 61499. https://en.wikipedia.org/wiki/IEC_61499, 05.05.2018
- [Fow14] Fowler, M.: Microservices. a definition of this new architectural term. <https://martinfowler.com/articles/microservices.html>, 06.05.2018
- [Fow15] Fowler, M.: Microservices Guide. <https://martinfowler.com/microservices/>, 06.05.2018
- [Fri09] Friedrich, A. D.: Anwendbarkeit von Methoden und Werkzeugen des konventionellen Softwareengineering zur Modellierung und Programmierung von Steuerungssystemen. Zugl.: Kassel, Univ., Diss., 2009. Kassel Univ. Press, Kassel, 2009
- [FSR18] Fuchigami, H.; Sarker, R.; Rangel, S.: Near-Optimal Heuristics for Just-In-Time Jobs Maximization in Flow Shop Scheduling. In Algorithms, 2018, 11; S. 43
- [GaJ79] Garey, M. R.; Johnson, D. S.: Computers and intractability. A guide to the theory of NP-completeness. Freeman, New York, NY, 1979

- [GAP06] García-Ojeda, J. C.; Arenas, A. E.; Pérez-Alcázar, J. d. J.: Paving the Way for Implementing Multiagent Systems: Refining Gaia with AUML. In (Müller, J. P.; Zambonelli, F. Hrsg.): Agent-oriented software engineering VI. 6th international workshop, AOSE 2005, Utrecht, The Netherlands, July 25, 2005 ; revised and invited papers. Springer, Berlin, 2006; S. 179–189
- [Gar17] Gartner Inc.: Top Trends in the Gartner Hype Cycle for Emerging Technologies, 2017. <https://www.gartner.com/smarterwithgartner/top-trends-in-the-gartner-hype-cycle-for-emerging-technologies-2017/>, 22.04.2018
- [GBS17] Göpfert, I.; Braun, D.; Schulz, M. Hrsg.: Automobillogistik. Stand und Zukunftstrends. Springer Gabler, Wiesbaden, 2017
- [GCK10] Günthner, W. A.; Chisu, R.; Kuzmany, F.: Die Vision vom Internet der Dinge. In (Günthner, W.; Hompel, M. t. Hrsg.): Internet der Dinge in der Intralogistik. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2010; S. 43–46
- [GDK13] Günthner, W. A. et al.: Schlanke Logistikprozesse. Handbuch für den Planer. Springer, Dordrecht, 2013
- [GhT18] Gholami, O.; Törnquist Krasemann, J.: A Heuristic Approach to Solving the Train Traffic Re-Scheduling Problem in Real Time. In Algorithms, 2018, 11; S. 55
- [GiH07] Gillert, F.; Hansen, W.-R.: IT-Architekturen und Services. In (Gillert, F.; Hansen, W.-R. Hrsg.): RFID für die Optimierung von Geschäftsprozessen. Prozess-Strukturen, IT-Architekturen, RFID-Infrastruktur. Hanser, München, 2007
- [GiT11] Gilbert, G. N.; Troitzsch, K. G.: Simulation for the social scientist. Open Univ. Press, Maidenhead, 2011
- [GiT15] Giret, A.; Trentesaux, D.: Software Engineering Methods for Intelligent Manufacturing Systems. A Comparative Survey. In (Mařík, V. et al. Hrsg.): Industrial applications of holonic and multi-agent systems. 7th international conference, HoloMAS 2015, Valencia, Spain, September 2-3, 2015 ; proceedings. Springer, Cham, 2015; S. 11–21
- [GLL10] Gehlich, D.; Luft, A.; Lorenz, S.: Ein dezentral gesteuertes Kommissionierlager. In (Günthner, W.; Hompel, M. t. Hrsg.): Internet der Dinge in der Intralogistik. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2010; S. 295–312
- [GMG07] Günthner, W. A.; Meissner, S.; Grinninger, J.: Komplexitätsbeherrschung in der Automobilfertigung durch Analyse und Optimierung der Auftragssteuerungsprozesse. In (Bruns, R. Hrsg.): 3. Fachkolloquium der WGTL, Hamburg, 2007; S. 79–90

- [GMN03] Giorgini, P. et al.: Reasoning with Goal Models. In (Spaccapietra, S.; March, S. T.; Kambayashi, Y. Hrsg.): Conceptual Modeling - ER 2002. 21st International Conference on Conceptual Modeling Tampere, Finland, October 7-11, 2002 Proceedings. Springer, Berlin, Heidelberg, 2003; S. 167–181
- [Göh13] Göhner, P. Hrsg.: Agentensysteme in der Automatisierungstechnik. Springer, Berlin, 2013
- [GöL10] Göhring, S.; Lorenz, T.: Agentenbasierte Staplerleitsysteme. In (Günthner, W.; Hompel, M. t. Hrsg.): Internet der Dinge in der Intralogistik. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2010; S. 313–328
- [GPS13] Gyulai, D. et al.: Milkrun Vehicle Routing Approach for Shop-floor Logistics. In Procedia CIRP, 2013, 7; S. 127–132
- [Gre16] Greschke, P.: Matrix-Produktion als Konzept einer taktunabhängigen Fließfertigung. Dissertation, Essen, 2016
- [Gri01] Gribble, S. D.: Robustness in complex systems: Eighth Workshop on Hot Topics in Operating Systems. Proceedings [HotOS-VIII] 20-22 May 2001, Elmau, Germany. IEEE Computer Society Press, Los Alamitos, Calif, 2001; S. 21–26
- [Gri12] Grinninger, J.: Schlanke Produktionssteuerung zur Stabilisierung von Auftragsfolgen in der Automobilproduktion. Dissertation, München, 2012
- [Gri15] Grigoryev, I.: AnyLogic 7 in three days. A quick course in simulation modeling. Selbstverlag, Middletown, 2015
- [GrK94] Groth, U.; Kammel, A.: Lean-Management. Konzept, kritische Analyse, praktische Lösungsansätze. Gabler, Wiesbaden, 1994
- [GSW17] Göpfert, I.; Schulz, M.; Wellbrock, W.: Trends in der Automobillogistik. In (Göpfert, I.; Braun, D.; Schulz, M. Hrsg.): Automobillogistik. Stand und Zukunftstrends. Springer Gabler, Wiesbaden, 2017; S. 1–26
- [GüH10] Günthner, W.; Hompel, M.t. Hrsg.: Internet der Dinge in der Intralogistik. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2010
- [Gun01] Gunasekaran, A.: Agile manufacturing. The 21st century competitive strategy, 2001
- [Gün10] Günthner, W. A.: Fazit. In (Günthner, W.; Hompel, M. t. Hrsg.): Internet der Dinge in der Intralogistik. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2010; S. 349–351
- [Gün17] Günther, M. T.: Produktionssteuerung nach dem Perlenketten-Prinzip am Beispiel der Automobilfertigung. Dissertation, Jena, 2017

- [Gun99] Gunasekaran, A.: Agile manufacturing. A framework for research and development. In International Journal of Production Economics, 1999, 62; S. 87–105
- [Gut02] Gutenschwager, K.: Online-Dispositionsprobleme in der Lagerlogistik. Modellierung - Lösungsansätze - praktische Umsetzung ; mit 16 Tabellen. Zugl.: Braunschweig, Techn. Univ., Diss., 2001. Physica-Verl., Heidelberg, 2002
- [GüT12] Günther, H.-O.; Tempelmeier, H.: Produktion und Logistik. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012
- [Hab16] Habermann, R.: Konzeption eines Softwareagentensystems zur Steuerung werkslogistischer Prozesse. Masterarbeit, 2016
- [HaD14] Hazir, Ö.; Dolugi, A.: Robust Assembly Line Balancing: State of the Arte and New Research Perspectives. In (Werner, F.; Sotskov, I. N. Hrsg.): Sequencing and scheduling with inaccurate data. Nova Science Publishers Inc, Hauppauge, New York, 2014; S. 211–223
- [Hal99] Haller, M.: Bewertung der Flexibilität automatisierter Materialflußsysteme der variantenreichen Großserienproduktion. Zugl.: München, Techn. Univ., Diss, 1999. Utz Wiss, München, 1999
- [Han01] Handrich, W.: Flexible, flurfreie Materialflusstechnik für dynamische Produktionsstrukturen. Dissertation, München, 2001
- [Han06] Hansmann, K.-W.: Industrielles Management. Oldenbourg, München, 2006
- [Har73] Harrington, J.: Computer integrated Manufacturing. Industrial Pr, New York, N.Y., 1973
- [Hei15] Heibold, T.: Einführung in die Automatisierungstechnik. Automatisierungssysteme, Komponenten, Projektierung und Planung ; mit 43 Tabellen. Fachbuchverl. Leipzig im Carl-Hanser-Verl., München, 2015
- [Her03] Hernández Morales, R.: Systematik der Wandlungsfähigkeit in der Fabrikplanung. Zugl.: Hannover, Univ., Diss., 2002. VDI-Verl., Düsseldorf, 2003
- [Her05] Herold, L.: Kundenorientierte Prozesssteuerung in der Automobilindustrie. Die Rolle von Logistik und Logistikcontrolling im Prozess „vom Kunden bis zum Kunden“. Dt. Univ.-Verl., Wiesbaden, 2005
- [Her12] Herlyn, W. J.: PPS im Automobilbau. Produktionsprogrammplanung und -steuerung von Fahrzeugen und Aggregaten. Hanser, München, 2012
- [HHL16] Hell, K. et al.: Demands on virtual representation of physical Industrie 4.0 components. In CEUR workshop proceedings, 2016, 1728 (2016); S. 65–71

- [HLG15] Heinrich, B.; Linke, P.; Glöckler, M.: Grundlagen Automatisierung. Springer Fachmedien Wiesbaden, Wiesbaden, 2015
- [Höc98] Höck, M.: Produktionsplanung und -steuerung einer flexiblen Fertigung. Ein prozeßorientierter Ansatz. Gabler Verlag, Wiesbaden, s.l., 1998
- [Hom10] Hompel, M. t.: Individualisierung als logistisch-technisches Prinzip. In (Günthner, W.; Hompel, M. t. Hrsg.): Internet der Dinge in der Intralogistik. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2010; S. 3–7
- [HWG12] Horenburg, T.; Wimmer, J.; Günthner, W. A.: Resource Allocation in Construction Scheduling based on Multi-Agent Negotiation. In (Telichenko, V. Hrsg.): Abstracts of the 14th International Conference on Computing in Civil and Building Engineering. Moscow, June 27-29, 2012. ASV, Moscow, Moscow, 2012
- [IBF09] Ivanova, D.; Batchkova, I.; Frey, G.: Modeling and verification approach based on IEC61499 function blocks. In (Mladenov, V. Hrsg.): Advanced aspects of theoretical electrical Engineering. Part II: Regular Papers, ISSN: 1313-9487, Sozopol, 2009; S. 43–49
- [IDS16] Ivanov, D. et al.: A dynamic model and an algorithm for short-term supply chain scheduling in the smart factory industry 4.0. In International Journal of Production Research, 2016, 54; S. 386–402
- [Ihm06] Ihme, J.: Logistik im Automobilbau. Logistikkomponenten und Logistiksysteme im Fahrzeugbau ; Tabellen. Hanser, München, Wien, 2006
- [IPT16] IPT Fraunhofer-Institut für Produktionstechnologie: Industrie 4.0 - Vernetzte, adaptive Produktion. <https://www.ipt.fraunhofer.de/content/dam/ipt/de/documents/Broschueren/Industrie%2040-Vernetzte%20adaptive%20Produktion.pdf>
- [ISA10] ISA: ANSI/ISA-88.00.01-2010 Batch Control Part 1: Models and Terminology, 2010
- [Ise16] Isenberg, M.-A.: Produktionsplanung und -steuerung in mehrstufigen Batchproduktionen. Dissertation, Bremen, 2016
- [Jak17] Jakobi, M.: Development of model-based control applications compliant with IEC 61499 for building energy systems with a focus on photovoltaics, 2017
- [Jam97] Jamison, W. C.: Approaching interoperability for heterogenous multiagent systems using high order agencies. In (Kandzia, P.; Klusch, M. Hrsg.): Cooperative Information Agents. First International Workshop, CIA'97 Kiel, Germany, February 26-28, 1997 Proceedings. Springer, Berlin, Heidelberg, 1997; S. 222–234
- [Jan12] Janda, G.: Variantenmanagement im Karosseriebau. Variantenreduzierung vs. Gewichtsoptimierung. Bachelorarbeit, 2012

- [JPS02] Juan, T.; Pearce, A.; Sterling, L.: ROADMAP. Extending the Gaia Methodology for Complex Open Systems. In (Gini, M. Hrsg.): Proceedings of the first international joint conference on Autonomous agents and multiagent systems part 2. ACM, New York, NY, 2002; S. 3
- [JRC09] Jungwattanakit, J. et al.: A comparison of scheduling algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. In Computers & Operations Research, 2009, 36; S. 358–378
- [Jun15] Jung, H.: Konzept einer agentenbasierten Transportsteuerung für komplexe, dynamische und multimodale Logistiknetzwerke. Karlsruhe, 2015
- [Kad15] Kadera, P.: Methods for Developing industrial MAS, 2015
- [KAG16] Kagermann, H. et al.: Industrie 4.0 in a Global Context. Strategies for Cooperating with International Partners (acatech STUDY). Herbert Utz Verlag GmbH, München, 2016
- [KiD14] Kis, T.; Drótos, M.: Planning and scheduling in the digital factory: KoMSO Challenge Workshop "Math for the Digital Factory", 2014
- [Kie07a] Kiefer, J.: Mechatronikorientierte Planung automatisierter Fertigungszellen im Bereich Karosserierohbau. Universität des Saarlandes, 2007
- [Kie07b] Kiel, E.: Antriebslösungen. Mechatronik für Produktion und Logistik ; mit 51 Tabellen, 2007
- [KLB18] Kreinovich, V.; Longpré, L.; Beltran, A.: QFT + NP = P Quantum Field Theory (QFT): A Possible Way of Solving NP-Complete Problems in Polynomial Time. In (Hassanien, A. E.; Elhoseny, M.; Kacprzyk, J. Hrsg.): Quantum Computing: An Environment for Intelligent Large Scale Real Application. Springer International Publishing, Cham, 2018; S. 229–249
- [Kle06] Kletti, J.: MES - Manufacturing Execution System. Moderne informationstechnologie zur prozessfähigkeit der wertschöpfung. Springer, Berlin, 2006
- [Kle12] Klein, N.: The impact of decentral dispatching strategies on the performance of intralogistics transport systems. Dissertation, Dresden, 2012
- [Kle16] Kleppmann, W.: Versuchsplanung: Produkte und Prozesse optimieren. Carl Hanser Verlag GmbH & Company KG, 2016

- [Kle98] Klein, R.: Das Sweep-Verfahren. In (Ottmann, T. Hrsg.): Prinzipien des Algorithmenentwurfs. [die Realisierung dieses Produkts wurde im Rahmen des MEDOC-Projekts unter dem Titel "Multimediale Aufbereitung des Gebietes Algorithmen und Datenstrukturen" (Förderkennzeichen 08C5825 0) vom BMBF gefördert]. Spektrum Akad. Verl., Heidelberg, Berlin, 1998
- [KIG13] Klenk, E.; Galka, S.: Das Hauptziel. Dynamische und flexible Systeme. In Hebezeuge Fördermittel, 2013, 53; S. 7–8
- [KIS14] Kletti, J.; Schumacher, J.: Die perfekte Produktion. Manufacturing Excellence durch Short Interval Technology (SIT). Springer Vieweg, Berlin, 2014
- [Klü14] Klügl, F.: Multiagentensysteme. In (Görz, G.; Schneeberger, J.; Schmid, U. Hrsg.): Handbuch der künstlichen Intelligenz. Oldenbourg, München, 2014; S. 527–556
- [Klu18] Klug, F.: Logistikmanagement in der Automobilindustrie. Grundlagen der Logistik im Automobilbau. Springer Vieweg, Berlin, 2018
- [KrB15] Kravari, K.; Bassiliades, N.: A Survey of Agent Platforms. In Journal of Artificial Societies and Social Simulation, 2015, 18
- [KRK02] Krog, E. H. et al.: Kooperatives Bedarfs- und Kapazitätsmanagement der Automobilhersteller und Systemlieferanten. In Logistik-Management internationale Konzepte, Instrumente, Anwendungen Zeitschrift der Kommission Logistik im Verband der Hochschullehrer für Betriebswirtschaft e.V, 2002
- [Kro09] Kropik, M.: Produktionsleitsysteme in der Automobilfertigung. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009
- [KrS14] Krause, A.; Strauche, J.: Routenzüge in der variantenreichen Großserienfertigung. Wandlungsfähigkeit als Schlüssel zur Effizienz. In (Müller, E. Hrsg.): Produktion und Arbeitswelt 4.0 – Aktuelle Konzepte für die Praxis? 15. Tage des Betriebs- und Systemingenieurs – TBI2014, Wissenschaftliche Schriftenreihe des IBF, Sonderheft 20, TU Chemnitz, 2014; S. 157–166
- [KrW13] Kremer, H.; Westerkamp, C.: Einsatz von Agentensystemen zur Optimierung der Logistik in Produktions- und Agrarprozessen. In (Göhner, P. Hrsg.): Agentensysteme in der Automatisierungstechnik. Springer, Berlin, 2013; S. 187–205
- [KuG13] Kugler, W.; Gehlich, D.: Einsatz von Agentensystemen in der Intralogistik. In (Göhner, P. Hrsg.): Agentensysteme in der Automatisierungstechnik. Springer, Berlin, 2013; S. 113–128
- [Küh06] Kühn, W.: Digitale Fabrik. Fabriksimulation für Produktionsplaner. Carl Hanser Fachbuchverlag, s.l., 2006

- [Kul87] Kulla, B.: Ergebnisse oder Erkenntnisse — liefern makroanalytische Simulationsmodelle etwas Brauchbares? In (Biethahn, J.; Schmidt, B. Hrsg.): Simulation als betriebliche Entscheidungshilfe. Methoden, Werkzeuge, Anwendungen. Springer, Berlin, Heidelberg, 1987; S. 3–25
- [KüS14] Kütting, H.; Sauer, M. J.: Elementare Stochastik. Mathematische Grundlagen und didaktische Konzepte. Springer Spektrum, Berlin, 2014
- [KuT08] Kuchiki, A.; Tsuji, M.: The Flowchart Approach to Industrial Cluster Policy. Palgrave Macmillan, New York, 2008
- [KWH12] Kagermann, H.; Wahlster, W.; Helbig, J.: Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0. Abschlussbericht des Arbeitskreises Industrie 4.0, 2012
- [KWH13] Kagermann, H.; Wahlster, W.; Helbig, J.: Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0. Abschlussbericht des Arbeitskreises Industrie 4.0. Deutschlands Zukunft als Produktionsstandort sichern, 2013
- [LaS18] Langmann, R.; Stiller, M.: Cloud-Based Industrial Control Services. The Next Generation PLC. In (Auer, M. E.; Zutin, D. G. Hrsg.): Online engineering & Internet of Things. Proceedings of the 14th International Conference on Remote Engineering and Virtual Instrumentation REV 2017, held 15-17 March 2017, Columbia University, New York, USA. Springer International Publishing, Cham, 2018; S. 3–18
- [LaT10] Lanfer, A.; Trautmann, A.: Der Lebenszyklus eines Internet der Dinge Materialflusssysteme. Umbau und Modernisierung. In (Günthner, W.; Hompel, M. t. Hrsg.): Internet der Dinge in der Intralogistik. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2010; S. 237–247
- [Law08] Law, A. M.: How to build valid and credible Simulation Models. In (Mason, S. J. et al. Hrsg.): Proceedings of the 40th Conference on Winter Simulation. Winter Simulation Conference, s.l., 2008; S. 39–47
- [Law15] Law, A. M.: Simulation modeling and analysis. McGraw-Hill Education, New York, NY, 2015
- [LCL10] Libert, S.; Chisu, R.; Luft, A.: Softwarearchitektur für eine agentenbasierte Materialflussteuerung. In (Günthner, W.; Hompel, M. t. Hrsg.): Internet der Dinge in der Intralogistik. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2010; S. 95–106
- [LCR05] Leitao, P.; Colombo, A. W.; Restivo, F. J.: ADACOR. A Collaborative Production Automation and Control Architecture. In IEEE Intelligent Systems, 2005, 20; S. 58–66
- [LCZ17] Lüder, A. et al.: Design pattern for agent-based production system control — A survey: 13th IEEE International Conference on Automation Science and Engineering (CASE). IEEE, Piscataway, NJ, 2017; S. 717–722

- [LeK15] Leitão, P.; Karnouskos, S. Hrsg.: *Industrial Agents: Emerging Applications of Software Agents in Industry*. Elsevier, Amsterdam, Oxford, Waltham, 2015
- [LeR02] Leitao, P.; Restivo, F.: *Holonic adaptive production control systems: IECON-2002. Proceedings of the 2002 28th Annual Conference of the IEEE Industrial Electronics Society* Sevilla, Spain, November 5-8, 2002. IEEE, Piscataway, NJ, 2002; S. 2968–2973
- [LeR03] Leitão, P.; Restivo, F.: *Identification of ADACOR Holons for Manufacturing Control*. In *IFAC Proceedings Volumes*, 2003, 36; S. 109–114
- [LeR08] Leitão, P.; Restivo, F. J.: *Implementation of a Holonic Control System in a Flexible Manufacturing System*. In *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 2008, 38; S. 699–709
- [LES98] Luczak, H.; Eversheim, W.; Schotten, M. Hrsg.: *Produktionsplanung und -steuerung. Grundlagen, Gestaltung und Konzepte*. Springer, Berlin, 1998
- [Lew13] Lewis, T. G.: *Cognitive stigmergy. A study of emergence in small-group social networks*. In *Cognitive Systems Research*, 2013, 21; S. 7–21
- [LHM12] Lorscheid, I.; Heine, B.-O.; Meyer, M.: *Opening the ‘black box’ of simulations. Increased transparency and effective communication through the systematic design of experiments*. In *Computational and mathematical organization theory*, 2012, 18; S. 22–62
- [Lik04] Liker, J. K.: *The Toyota way. 14 management principles from the world's greatest manufacturer*. McGraw-Hill, New York, NY, 2004
- [Lil12] Lillis, D. J.: *Internalising interaction protocols as first-class programming elements in multi agent systems*. University College Dublin, Dublin, 2012
- [Lin16] Lindemann, U. Hrsg.: *Handbuch Produktentwicklung*. Hanser, München, 2016
- [Lit13] Litz, L.: *Grundlagen der Automatisierungstechnik. Regelungssysteme - Steuerungssysteme - Hybride Systeme*. Oldenbourg Wissenschaftsverlag, s.l., 2013
- [LJM11] Lang, W. et al.: *The “Intelligent Container”—A Cognitive Sensor Network for Transport Management*. In *IEEE Sensors Journal*, 2011, 11; S. 688–698
- [LMB09] Lindemann, U.; Maurer, M.; Braun, T.: *Structural complexity management. An approach for the field of product design*. Springer, Berlin, Heidelberg, 2009
- [LoC07] Loureiro, G.; Curran, R. Hrsg.: *Complex Systems Concurrent Engineering. Collaboration, Technology Innovation and Sustainability*. Springer-Verlag London Limited, London, 2007

- [Lö08] Lödding, H.: Verfahren der Fertigungssteuerung. Grundlagen, Beschreibung, Konfiguration. Springer, Berlin, Heidelberg, 2008
- [LPW97] Lee, H. L.; Padmanabhan, V.; Whang, S.: The Bullwhip Effect in Supply Chains. In Sloan Management Review 38(3), 1997; S. 93–102
- [LRT15] 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: Emerging Applications of Software Agents in Industry. Elsevier, Amsterdam, Oxford, Waltham, 2015; S. 287–300
- [LSH17a] Lüder, A. et al.: Identification of artifacts in life cycle phases of CPPS. In (Biffel, S.; Lüder, A.; Gerhard, D. Hrsg.): Multi-disciplinary engineering for cyber-physical production systems: data models and software solutions for handling complex engineering projects. Springer-Verlag, Cham, 2017; S. 139–167
- [LSH17b] Lüder, A. et al.: Description means for information artifacts throughout the life cycle of CPPS. In (Biffel, S.; Lüder, A.; Gerhard, D. Hrsg.): Multi-disciplinary engineering for cyber-physical production systems: data models and software solutions for handling complex engineering projects. Springer-Verlag, Cham, 2017; S. 169–183
- [Lüd06] Lüder, A.: Strukturen zur verteilten Steuerung von Produktionssystemen, 2006
- [LüF13] Lüder, A.; Foehr, M.: Identifikation und Umsetzung von Agenten zur Fabrikautomation unter Nutzung von mechatronischen Strukturierungskonzepten. In (Göhner, P. Hrsg.): Agentensysteme in der Automatisierungstechnik. Springer, Berlin, 2013; S. 45–61
- [LüF15] 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, Oxford, Waltham, 2015; S. 171–190
- [Lun08] Lunze, J.: Automatisierungstechnik. Methoden für die Überwachung und Steuerung kontinuierlicher und ereignisdiskreter Systeme. Oldenbourg, München, 2008
- [LVG14] Lüder, A.; Vogel-Heuser, B.; Göhner, P.: Elektronische Datenverarbeitung - Agentenbasiertes Steuern: Dubbel Taschenbuch für den Maschinenbau. Springer Vieweg, Berlin [u.a.], 2014; S. 1924–1928
- [LVM09] Lepuschitz, W. et al.: Integration of a Heterogeneous Low Level Control in a Multi-Agent System for the Manufacturing Domain. In IEEE Conference on Emerging Technologies & Factory Automation, 2009

- [LWF13] Linnenberg, T.; Wior, I.; Fay, A.: Analysis of potential instabilities in agent-based smart grid control systems: IECON 2013. 39th annual conference of the IEEE Industrial Electronics Society ; 10 - 13 Nov. 2013, Vienna, Austria ; proceedings. IEEE, Piscataway, NJ, 2013; S. 7420–7425
- [LZB17] Lüder, A.; Zawisza, J.; Becker, A.: Advancing the performance of complex manufacturing systems through agent-based production control. In (Berndt, J. O.; Petta, P.; Unland, R. Hrsg.): Multiagent System Technologies 15th German Conference, MATES 2017 Leipzig, Germany, August 23-26, 2017 proceedings. Springer, Cham, 2017; S. 103–120
- [Mab08] Mabrouk, M. I.: SOA fundamentals in a nutshell. Prepare to become an IBM Certified SOA Associate. <https://www.ibm.com/developerworks/webservices/tutorials/ws-soa-ibmcertified/ws-soa-ibmcertified.html>, 07.05.2018
- [Mai04] Maier, F.: Komplexität und Dynamik als Herausforderung für das Management. Festschrift zum 60. Geburtstag von Peter Milling. Deutscher Universitätsverlag, Wiesbaden, 2004
- [MaO08] Marsh, L.; Onof, C.: Stigmergic epistemology, stigmergic cognition. In Cognitive Systems Research, 2008, 9; S. 136–149
- [MaS09] Maeder, J.; Schmidt, R.: Perlenkette versus späte Taufe. Symbiose für die Produktionssteuerung. <http://www.it-production.com/druck.php?id=47731>, 06.01.2015
- [MaW04] Maxton, G. P.; Wormald, J.: Time for a model change. Re-engineering the global automotive industry. Cambridge University Press, Cambridge, 2004
- [McB03] McFarlane, D.; Bussmann, S.: Holonic Manufacturing Control: Rationales, Developments and Open Issues. In (Deen, S. M. Hrsg.): Agent-based manufacturing. Advances in the holonic approach ; with 17 tables. Springer, Berlin, 2003; S. 303–326
- [McK15] McKinsey: Industry 4.0. How to navigate digitization of the manufacturing sector. https://www.mckinsey.de/files/mck_industry_40_report.pdf, 08.02.2018
- [McL16] McLarty, M.: Learn from SOA: 5 lessons for the microservices era. <https://www.in-foworld.com/article/3080611/application-development/learning-from-soa-5-lessons-for-the-microservices-era.html>, 06.05.2018
- [MCN17] Mitton, N. et al. Hrsg.: Interoperability, safety and security in IoT. Second International Conference, InterIoT 2016 and Third International Conference, SaseIoT 2016, Paris, France, October 26-27, 2016 revised selected papers. Springer, Cham, 2017
- [Mei09] Meissner, S.: Logistische Stabilität in der automobilen Variantenfließfertigung. Techn. Univ., Diss.--München, 2009. Technische Universität München, München, 2009

- [Mel13] Meling, F.: Methodik für die Rekombination von Anlagentechnik. Zugl.: München, Techn. Univ., Diss., 2012, München, 2013
- [MES08] MESA: SOA in Manufacturing Guidebook. ftp://public.dhe.ibm.com/software/plm/pdif/MESA_SOAINManufacturingGuidebook.pdf, 07.05.2018
- [Mey04] Meyr, H.: Supply chain planning in the German automotive industry. In OR Spectrum, 2004, 26
- [Mil10] Miller, P.: Smart swarm. Collins, London, 2010
- [Mil88] Minsky, M.; Lee, J.: Society of Mind // The society of mind. Simon & Schuster, New York, NY, 1988
- [MKW17] Marseau, E.; Kolberg, D.; Weyer, S.: Exemplarische Übertragung der RAMI 4.0-Verwaltungsschale auf die SmartFactoryKL. Systemarchitektur für Industrie 4.0-Produktionsanlagen. http://smartfactory.de/wp-content/uploads/2017/08/SF_WhitePaper_2-1_DE.pdf, 17.08.2018
- [MMV03] Mařík, V.; McFarlane, D.; Valckenaers, P. Hrsg.: Hologic and multi-agent systems for manufacturing. First International Conference on Industrial Applications of Hologic and Multi-Agent Systems, HoloMAS 2003, Prague, Czech Republic, September 1 - 3, 2003 ; proceedings. Springer, Berlin, Heidelberg, 2003
- [MoB17] Monica, S.; Bergenti, F.: Indoor Localization of JADE Agents Without a Dedicated Infrastructure. In (Berndt, J. O.; Petta, P.; Unland, R. Hrsg.): Multiagent System Technologies 15th German Conference, MATES 2017 Leipzig, Germany, August 23-26, 2017 proceedings. Springer, Cham, 2017; S. 256–271
- [Mön06] Mönch, L.: Agentenbasierte Produktionssteuerung komplexer Produktionssysteme. Deutscher Universitäts-Verlag GWV Fachverlage GmbH, Wiesbaden, Wiesbaden, 2006
- [Moo13] Moonen, H.: Multi-Agent-Systems for Transportation Planning and Coordination, 2013
- [MoS06] Moraitis, P.; Spanoudakis, N.: The Gaia2JADE Process for Multi-Agent Systems Development. In Applied Artificial Intelligence, 2006, 20; S. 251–273
- [MoV15] Mousavi, A.; Vyatkin, V.: Energy Efficient Agent Function Block. A semantic agent approach to IEC 61499 function blocks in energy efficient building automation systems. In Automation in Construction, 2015, 54; S. 127–142

- [MRT07] Mirghorbani, S. M. et al.: A Multi-Objective Particle Swarm for a Mixed-Model Assembly Line Sequencing. In (Waldmann, K.-H.; Stocker, U. M. Hrsg.): Operations Research Proceedings 2006. Selected Papers of the Annual International Conference of the German Operations Research Society (GOR), Jointly Organized with the Austrian Society of Operations Research (ÖGOR) and the Swiss Society of Operations Research (SOSOR). Springer-Verlag, s.l., 2007; S. 181–186
- [MVK06] Monostori, L.; Váncza, J.; Kumara, S.R.T.: Agent-Based Systems for Manufacturing. In CIRP Annals - Manufacturing Technology, 2006, 55; S. 697–720
- [MZK12] Merdan, M. et al.: Adaptive Produktionssysteme durch den Einsatz von autonomen Softwareagenten. In e & i Elektrotechnik und Informationstechnik, 2012, 129; S. 53–58
- [NeK10] Nettsträter, A.; Kuzmany, F.: Rechenplattformen und RFID für das Internet der Dinge. In (Günthner, W.; Hompel, M. t. Hrsg.): Internet der Dinge in der Intralogistik. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2010; S. 107–118
- [New15] Newman, S.: Building microservices. Designing fine-grained systems. O'Reilly Media, Sebastopol, CA, 2015
- [Nop10] Nopper, J. R.: Der Lebenszyklus eines Internet der Dinge Materialflusssystemes. Betrieb. In (Günthner, W.; Hompel, M. t. Hrsg.): Internet der Dinge in der Intralogistik. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2010; S. 229–235
- [Nyh08] Nyhuis, P. Hrsg.: Wandlungsfähige Produktionssysteme. Heute die Industrie von morgen gestalten. PZH Produktionstechnisches Zentrum; Technische Informationsbibliothek u. Universitätsbibliothek, Garbsen, Hannover, 2008
- [NYS00] Narain, R. et al.: The strategic implications of flexibility in manufacturing systems. In International Journal of Agile Management Systems, 2000, 2; S. 202–213
- [Ohn93] Ohno, T.: Das Toyota-Produktionssystem. Campus-Verl., Frankfurt, 1993
- [OIC18] OICA: Production Statistics. <http://www.oica.net/production-statistics/>, 07.03.2018
- [PAB08] PABADIS'PROMISE consortium: Industrial Application of the PABADIS'PROMISE System. White Paper, 2008
- [Pac12] Pachow-Frauenhofer, J.: Planung veränderungsfähiger Montagesysteme. Univ., Diss.--Hannover, 2012. PZH Produktionstechn. Zentrum, Garbsen, 2012
- [PaG08] Parry, G.; Graves, A.: Build to order. The road to the 5-day car, 2008

- [PaW02] Padgham, L.; Winikoff, M.: Prometheus. A Methodology for Developing Intelligent Agents. In (Gini, M. et al. Hrsg.): Proceedings of the first international joint conference on Autonomous agents and multiagent systems part 1 - AAMAS '02. ACM Press, New York, New York, USA, 2002; S. 37
- [Pes14] Peschl, M.: An Architecture for flexible Manufacturing Systems based on task-driven Agents, Oulu, 2014
- [Pic16] Pichler, A.: Forschungsergebnisse im Bereich Verteilter Automation aus Österreich finden weltweit Einzug in Labors und Produkte. https://open4innovation.at/resources/pdf/Project_microns_v2.pdf, 29.04.2018
- [PiH04] Pil, F. K.; Holweg, M.: Linking Product Variety to Order-Fulfillment Strategies. In Interfaces, 2004, 34; S. 394–403
- [Pla16a] Plattform Industrie 4.0: Struktur der Verwaltungsschale. Fortentwicklung des Referenzmodells für die Industrie 4.0-Komponente, 2016
- [Pla16b] Plattform Industrie 4.0: Gemeinsamer Aktionsplan Plattform Industrie 4.0 & Industrie du Futur, April 2016. http://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publikation/aktionsplan-plattform-i40-und-industrie-du-futur.pdf?__blob=publication-File&v=6
- [PMD14] Pantförder, D. et al.: Agentenbasierte dynamische Rekonfiguration von vernetzten intelligenten Produktionsanlagen – Evolution statt Revolution. In (Bauernhansl, T.; Hompel, M. t.; Vogel-Heuser, B. Hrsg.): Industrie 4.0 in Produktion, Automatisierung und Logistik. Springer Fachmedien Wiesbaden, Wiesbaden, 2014; S. 145–158
- [PMD17] Pantförder, D. et al.: Agentenbasierte dynamische Rekonfiguration von vernetzten intelligenten Produktionsanlagen. Evolution statt Revolution. In (Vogel-Heuser, B.; Bauernhansl, T.; Hompel, M. t. Hrsg.): Handbuch Industrie 4.0. Bd. 2 Automatisierung. Springer Vieweg, Berlin, 2017; S. 31–44
- [PML18] Pfitzer, F. et al.: Event-driven production rescheduling in job shop environments - Application to sheet metal production processes. eingereicht zur CASE 2018 "14th IEEE International Conference on Automation Science and Engineering", München, 2018
- [PoB18] Pokahr, A.; Braubach, L.: Jadex BDI Agent System. <https://sourceforge.net/projects/jadex/>, 03.06.2018
- [PSA14] Pfrommer, J. et al.: Modelling and orchestration of service-based manufacturing systems via skills: IEEE [International Conference on] Emerging Technologies and Factory Automation (ETFA), 2014. 16 - 19 Sept. 2014, Barcelona, Spain. IEEE, Piscataway, NJ, 2014; S. 1–4

- [RBE02] Reinhart, G. et al.: Wandlungsfähige Fabrikgestaltung. In ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb, 2002, 97; S. 18–23
- [REF90] Planung und Gestaltung komplexer Produktionssysteme. Hanser, München, 1990
- [Rib15] Ribeiro, L.: The Design, Deployment, and Assessment of Industrial Agent Systems. In (Leitão, P.; Karnouskos, S. Hrsg.): Industrial Agents: Emerging Applications of Software Agents in Industry. Elsevier, Amsterdam, Oxford, Waltham, 2015; S. 45–63
- [Rie11] Ries, E.: The lean startup. How constant innovation creates radically successful businesses. Portfolio Penguin, London, 2011
- [Roi10] Roidl, M.: Kooperation und Autonomie in selbststeuernden Systemen. In (Günthner, W.; Hompel, M. t. Hrsg.): Internet der Dinge in der Intralogistik. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2010; S. 65–78
- [Rot16] Roth, A. Hrsg.: Einführung und Umsetzung von Industrie 4.0. Grundlagen, Vorgehensmodell und Use Cases aus der Praxis. Springer Gabler, Berlin, Heidelberg, 2016
- [RRJ07] Rabbani, M. et al.: A New Approach for Mixed-Model Assembly Line Sequencing. In (Waldmann, K.-H.; Stocker, U. M. Hrsg.): Operations Research Proceedings 2006. Selected Papers of the Annual International Conference of the German Operations Research Society (GOR), Jointly Organized with the Austrian Society of Operations Research (ÖGOR) and the Swiss Society of Operatio. Springer-Verlag, s.l., 2007; S. 169–174
- [RRT07] Rahimi-Vahed, A. et al.: Mixed-Model Assembly Line Sequencing Using Real Options. In (Waldmann, K.-H.; Stocker, U. M. Hrsg.): Operations Research Proceedings 2006. Selected Papers of the Annual International Conference of the German Operations Research Society (GOR), Jointly Organized with the Austrian Society of Operations Research (ÖGOR) and the Swiss Society of Operatio. Springer-Verlag, s.l., 2007; S. 161–167
- [RuN10] Russell, S. J.; Norvig, P.: Artificial intelligence. A modern approach. Prentice-Hall, Upper Saddle River, NJ, 2010
- [Rya16] Ryashentseva, D.: Agents and SCT based self* control architecture for production systems, Magdeburg, 2016
- [SaK99] Saygin, C.; Kilic, S. E.: Integrating Flexible Process Plans with Scheduling in Flexible Manufacturing Systems. In The International Journal of Advanced Manufacturing Technology, 1999, 15; S. 268–280

- [SaP01] Sarker, B. R.; Pan, H.: Designing a mixed-model, open-station assembly line using mixed-integer programming. In *Journal of the Operational Research Society*, 2001, 52; S. 545–558
- [SaP11] Samad, T.; Parisini, T.: *Systems of Systems*. In (Samad, T.; Annaswamy, A. Hrsg.): *The Impact of Control Technology*. IEEE Control Systems Society, 2011
- [SBH10] Siebertz, K.; Bebbler, D.; Hochkirchen, T.: *Statistische Versuchsplanung. Design of Experiments (DoE)*. Springer, Berlin, 2010
- [ScC10] Schaaf, W.; Chisu, R.: Hochflexible, RFID-gesteuerte Handhabung von Stückgut. In (Günthner, W.; Hompel, M. t. Hrsg.): *Internet der Dinge in der Intralogistik*. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2010; S. 329–343
- [Sch07] Scholten, B.: *The road to integration. A guide to applying the ISA-95 standard in manufacturing*. ISA, Research Triangle Park NC, 2007
- [Sch13] Schreiber, S.: *Entwicklung einer Vergleichs- und Bewertungsmöglichkeit von dezentralen Steuerungsarchitekturen für Produktionssysteme*. Dissertation, Hamburg, 2013
- [ScK15] Schwede, C.; Kuhn, A.: *Integration von Auftragsreihenfolge- und Distributionstransportplanung in der Automobilindustrie*. Zugl.: Dortmund, Techn. Univ., Diss., 2014, Dortmund, 2015
- [SCN08] Solnon, C. et al.: The car sequencing problem. Overview of state-of-the-art methods and industrial case-study of the ROADEF'2005 challenge problem. In *European Journal of Operational Research*, 2008, 191; S. 912–927
- [ScS12] Schuh, G.; Stich, V.: *Produktionsplanung und -steuerung 1. Grundlagen der PPS*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012
- [ScT16] Scharte, B.; Thoma, K.: *Resilienz – Ingenieurwissenschaftliche Perspektive*. In (Wink, R. Hrsg.): *Multidisziplinäre Perspektiven der Resilienzforschung*. Springer, Wiesbaden, 2016; S. 123–150
- [Ses03] Sesterhenn, M.: *Bewertungssystematik zur Gestaltung struktur- und betriebsvariabler Produktionssysteme*. Techn. Hochsch., Diss.--Aachen, 2002. Shaker, Aachen, 2003
- [SGS16] SGS: *Asset Management*. <http://www.sgs.com/en/industrial-manufacturing/services-related-to-production-and-products/in-plant-and-operational-support/advanced-systems/asset-management-ahead>, 16.02.2016
- [Sho90] Shoham, Y.: *Agent-oriented programming*. Department of Computer Science Stanford University, Stanford, Calif., 1990

- [Sie16] Siepman, D.: Industrie 4.0 – Technologische Komponenten. In (Roth, A. Hrsg.): Einführung und Umsetzung von Industrie 4.0. Grundlagen, Vorgehensmodell und Use Cases aus der Praxis. Springer Gabler, Berlin, Heidelberg, 2016; S. 47–72
- [SiW11] Sitte, J.; Winzer, P.: Demand-Compliant Design. In IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 2011, 41; S. 434–448
- [SJF11] Schreiber, S.; Jerenz, S.; Fay, A.: Anforderungen an Steuerungskonzepte für moderne Fertigungsanlagen. Herausforderungen für dezentrale Ansätze in der Automatisierungstechnik. In (Adolphs, P. Hrsg.): Automation 2011. Der 12. Branchentreff der Mess- und Automatisierungstechnik ; Kongress Baden-Baden, 28. und 29. Juni 2011. VDI-Verl., Düsseldorf, 2011; S. 7–11
- [SLH16] Schmidt, N. et al.: A generic model for the end-of-life phase of production systems: IECON 2016 42th annual conference of the IEEE Industrial Electronics Society ; Florence, Italy, October 24 - 27, 2016. IEEE, Piscataway, NJ, 2016; S. 5693–5698
- [SMS07] Strasser, T. et al.: An Advanced Engineering Environment for Distributed & Reconfigurable Industrial Automation & Control Systems based on IEC 61499. In (Pham, D. T.; Eldukhri, E. E.; Soroka, A. J. Hrsg.): Intelligent Production Machines and Systems. 2nd I*PROMS Virtual International Conference 3-14 July 2006. Elsevier professional, s.l., 2007; S. 493–498
- [Sod17] Soder, J.: Use Case Production. Von CIM über Lean Production zu Industrie 4.0. In (Vogel-Heuser, B.; Bauernhansl, T.; Hompel, M. t. Hrsg.): Handbuch Industrie 4.0. Bd. 1: Produktion. Springer Vieweg, Berlin, 2017; S. 3–25
- [SpM09] Spanoudakis, N.; Moraitis, P.: Gaia Agents Implementation through Models Transformation. In (Yang, J.-J. et al. Hrsg.): Principles of practice in multi-agent systems. 12th international conference, PRIMA 2009, Nagoya, Japan, December 14 - 16, 2009 ; proceedings. Springer, Berlin, 2009; S. 127–142
- [SPS14] SPS-Magazin: Automatisieren mit der IEC 61499. http://www.sps-magazin.de/?inc=artikel/article_show&nr=84717, 29.04.2018
- [SSL14] Schlick, J. et al.: Industrie 4.0 in der praktischen Anwendung. In (Bauernhansl, T.; Hompel, M. t.; Vogel-Heuser, B. Hrsg.): Industrie 4.0 in Produktion, Automatisierung und Logistik. Springer Fachmedien Wiesbaden, Wiesbaden, 2014; S. 57–84
- [StS14] Sturm, A.; Shehory, O.: The Landscape of Agent-Oriented Methodologies. In (Shehory, O.; Sturm, A. Hrsg.): Agent-Oriented Software Engineering. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014; S. 137–154
- [StT17] Steen, M. van; Tanenbaum, A. S.: Distributed systems, 2017

- [SWH06] Shen, W.; Wang, L.; Hao, Q.: Agent-based distributed manufacturing process planning and scheduling. A state-of-the-art survey. In IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2006, 36; S. 563–577
- [Sys06] Syska, A.: Produktionsmanagement. Das A - Z wichtiger Methoden und Konzepte für die Produktion von heute. Gabler, Wiesbaden, 2006
- [TaM06] Takeda, H.; Meynert, A.: Das synchrone Produktionssystem. Just-in-time für das ganze Unternehmen. mi-Fachverl., Landsberg am Lech, 2006
- [TDT14] Tahriri, F.; Dawal, S. Z. M.; Taha, Z.: Fuzzy mixed assembly line sequencing and scheduling optimization model using multiobjective dynamic fuzzy GA. In TheScientificWorldJournal, 2014, 2014; S. 505207
- [TeK93] Tempelmeier, H.; Kuhn, H.: Flexible Fertigungssysteme. Entscheidungsunterstützung für Konfiguration und Betrieb mit 98 Tabellen. Springer, Berlin, 1993
- [Ter15] Termer, F.: Determinanten der IT-Agilität. Dissertation, 2015
- [TeT09] Terkaj, W.; Tolio, T.: Focused Flexibility in Production Systems. In (ElMaraghy, H. A. Hrsg.): Changeable and Reconfigurable Manufacturing Systems. Springer London, Guildford, Surrey, 2009; S. 47–70
- [The17] The State Council - The Peoples Republic of China: China to invest big in 'Made in China 2025' strategy. http://english.gov.cn/state_council/ministries/2017/10/12/content_281475904600274.htm, 21.04.2018
- [ToT98] Toni, A. de; Tonchia, S.: Manufacturing flexibility. A literature review. In International Journal of Production Research, 1998, 36; S. 1587–1617
- [ToV02] Toth, P.; Vigo, D.: The Vehicle Routing Problem. Society for Industrial and Applied Mathematics, 2002
- [Toy18] Toyota Motor Corporation: How many parts is each car made of? <http://www.toyota.co.jp/en/kids/faq/d/01/04/>, 06.03.2018
- [Uli05] Ulich, E.: Arbeitspsychologie. vdf Hochschulverl. an der ETH [u.a.], Zürich, 2005
- [Ulr06] Ulrich, T.: Änderungsmanagement in hoch integrierten Produktentstehungs- und Fertigungsprozessen. In ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb, 2006, 101; S. 653–657
- [Unl15a] Unland, R.: Software Agent Systems. In (Leitão, P.; Karnouskos, S. Hrsg.): Industrial Agents: Emerging Applications of Software Agents in Industry. Elsevier, Amsterdam, Oxford, Waltham, 2015; S. 3–22

- [UnI15b] Unland, R.: Industrial Agents. In (Leitão, P.; Karnouskos, S. Hrsg.): Industrial Agents: Emerging Applications of Software Agents in Industry. Elsevier, Amsterdam, Oxford, Waltham, 2015; S. 23–44
- [UsE15] Usländer, T.; Epple, U.: Reference model of Industrie 4.0 service architectures. In at - Automatisierungstechnik, 2015, 63
- [VaB15] Valckenaers, P.; Brussel, H. van: Design for the unexpected. From holonic manufacturing systems towards a humane mechatronics society. Elsevier, Waltham MA, 2015
- [VBH17a] Vogel-Heuser, B.; Bauernhansl, T.; Hompel, M.t. Hrsg.: Handbuch Industrie 4.0. Bd. 1: Produktion. Springer Vieweg, Berlin, 2017
- [VBH17b] Vogel-Heuser, B.; Bauernhansl, T.; Hompel, M.t. Hrsg.: Handbuch Industrie 4.0. Bd. 2 Automatisierung. Springer Vieweg, Berlin, 2017
- [VBW98] Valckenaers, P. et al.: Designing Holonic manufacturing systems. In Robotics and Computer-Integrated Manufacturing, 1998, 14; S. 455–464
- [VCG14] Vidal, T. et al.: A unified solution framework for multi-attribute vehicle routing problems. In European Journal of Operational Research, 2014, 234; S. 658–673
- [VDA07] Qualitätsmanagement-Center VDA: Das gemeinsame Qualitätsmanagement in der Lieferkette. VDAQMC-Projektdokumentation. VDA Qualitäts-Management-Center (QMC), Oberursel
- [VDA14] VDA: Unsere Werke. Nachhaltige Automobilproduktion in Deutschland. <https://www.vda.de/dam/vda/publications/2014/unsere-werke.pdf>, 07.03.2018
- [VDB13] Vogel-Heuser, B.; Diedrich, C.; Broy, M.: Anforderungen an CPS aus Sicht der Automatisierungstechnik. In at – Automatisierungstechnik, 2013, 61
- [VDE13] VDE: Die deutsche Normungsroadmap Industrie 4.0. DKE Normungs-Roadmap
- [VDI10] VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik, Düsseldorf, VDI e.V. VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik, Düsseldorf, VDI e.V.: Agentensysteme in der Automatisierungstechnik - Grundlagen, 2010
- [VDI12] VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik, Düsseldorf, VDI e.V. Ziesche, S.: Agentensysteme in der Automatisierungstechnik - Entwicklung, 2012
- [VDI17] VDI: VDI 5201 Blatt 1:2017-01 Wandlungsfähigkeit - Beschreibung und Messung der Wandlungsfähigkeit produzierender Unternehmen. Beuth, Berlin, 2017
- [VeV06] Verstraete, P.; Valckenaers, P.: Towards Cooperating Planning And Manufacturing Execution Systems, 2006

- [VFL14] Vogel-Heuser, B.; Folmer, J.; Legat, C.: Anforderungen an die Softwareevolution in der Automatisierung des Maschinen- und Anlagenbaus. In at – Automatisierungstechnik, 2014, 62
- [VGH06] Verstraete, P. et al.: On Applying the PROSA Reference Architecture in Multiagent Manufacturing Control Applications, 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: Emerging Applications of Software Agents in Industry. Elsevier, Amsterdam, Oxford, Waltham, 2015; S. 153–170
- [VLL15] Vogel-Heuser, B.; Lee, J.; Leitão, P.: Agents enabling cyber-physical production systems. In at - Automatisierungstechnik, 2015, 63
- [WaF17] 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). University of Applied Science Emden/Leer, Emden, Germany, 24-26 July 2017 proceedings. IEEE, Piscataway, NJ, 2017; S. 89–95
- [Wag18] Wagner, R.M. Hrsg.: Industrie 4.0 für die Praxis. Mit realen Fallbeispielen aus mittelständischen Unternehmen und vielen umsetzbaren Tipps. Springer Fachmedien Wiesbaden, Wiesbaden, 2018
- [Wan10] Wannagat, A. A.: Entwicklung und Evaluation agentenorientierter Automatisierungssysteme zur Erhöhung der Flexibilität und Zuverlässigkeit von Produktionsanlagen. Zugl.: München, Univ. Diss., 2010. Sierke, Göttingen, 2010
- [Wan17] Wang, J.: Real-time embedded systems. Wiley, Hoboken, NJ, USA, 2017
- [War92] Warnecke, H.-J.: Die Fraktale Fabrik. Revolution der Unternehmenskultur. Springer Berlin Heidelberg, Berlin, Heidelberg, s.l., 1992
- [WaW14] Waltl, H.; Wildemann, H.: Modularisierung der Produktion in der Automobilindustrie. TCW Transfer-Zentrum, München, 2014
- [WBS18] Werner, F.; Burtseva, L.; Sotskov, Y.: Special Issue on Algorithms for Scheduling Problems. In Algorithms, 2018, 11; S. 87
- [WeB06] Weck, M.; Brecher, C.: Werkzeugmaschinen 4. Automatisierung von Maschinen und Anlagen. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2006
- [Wec01] Weck, M.: Werkzeugmaschinen Fertigungssysteme. Automatisierung von Maschinen und Anlagen. Springer, Berlin, Heidelberg, 2001
- [Weg04] Wegehaupt, P.: Führung von Produktionsnetzwerken. Dissertation, Aachen, 2004

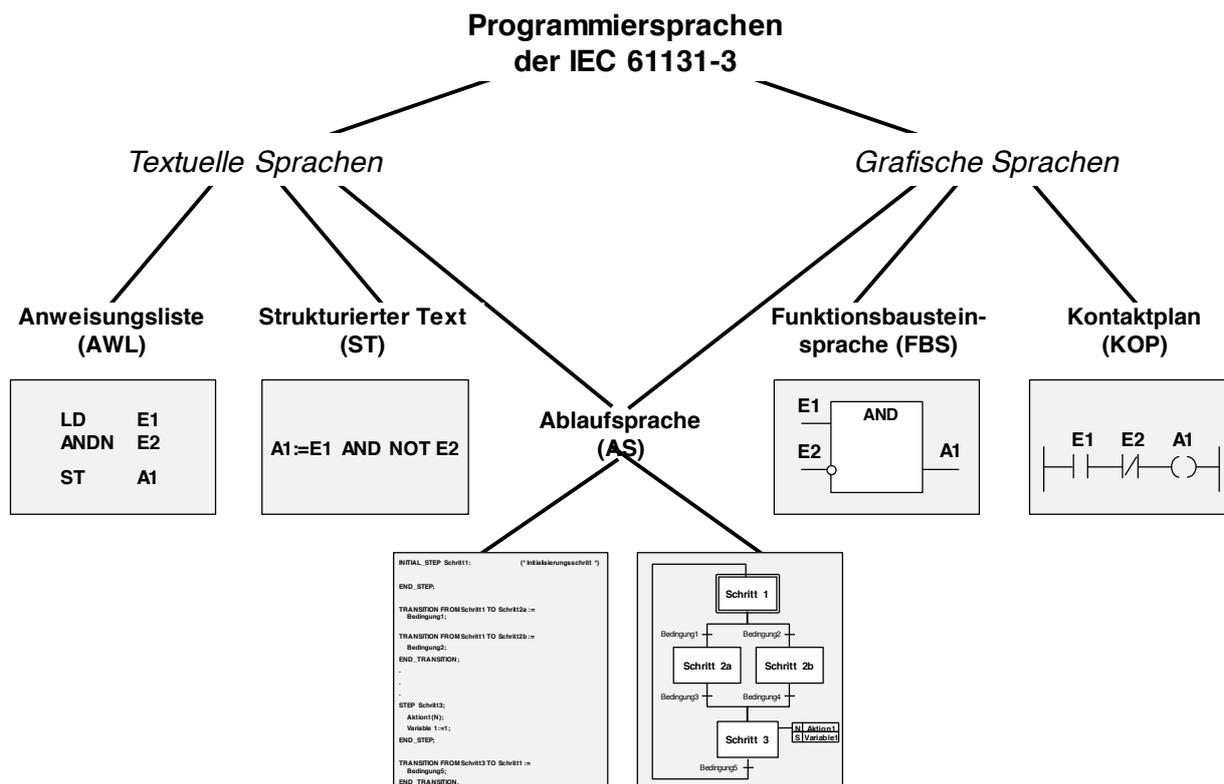
- [WEN07] Wiendahl, H.-P. et al.: Changeable Manufacturing - Classification, Design and Operation. In CIRP Annals - Manufacturing Technology, 2007, 56; S. 783–809
- [WeS09] Weyer, M.; Spath, D.: Das Produktionssteuerungskonzept "Perlenkette". In Zeitschrift für wirtschaftlichen Fabrikbetrieb, 2009, 2009; S. 1126–1130
- [WeS14] Werner, F.; Sotskov, I.N. Hrsg.: Sequencing and scheduling with inaccurate data. Nova Science Publishers Inc, Hauppauge, New York, 2014
- [Wey02] Weyer, M.: Das Produktionssteuerungskonzept Perlenkette und dessen Kennzahlensystem. Helmesverl., Karlsruhe, 2002
- [Wey10] Weyns, D.: Architecture-Based Design of Multi-Agent Systems. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2010
- [WeZ15] Wellenreuther, G.; Zastrow, D.: Automatisieren mit SPS - Theorie und Praxis. Programmieren mit STEP 7 und CoDeSys, Entwurfsverfahren, Bausteinbibliotheken ; Beispiele für Steuerungen, Regelungen, Antriebe und Sicherheit ; Kommunikation über AS-i-Bus, PROFIBUS, PROFINET, Ethernet-TCP/IP, OPC WLAN ; mit 108 Steuerungsbeispielen und 8 Projektierungen. Springer Vieweg, Wiesbaden, 2015
- [WGP08] Wagner, G. et al.: Modellierung und Simulation von Multiagenten-Systemen. In Forum der Forschung, 2008, 21; S. 47–52
- [WHW00] Wu, J.-F.; Hamada, M.; Wu, C. F. J.: Experiments. Planning, analysis, and parameter design optimization. Wiley, New York NY u.a., 2000
- [Wie02] Wiendahl, H.-P.: Wandlungsfähigkeit. Schlüsselbegriff der zukunftsfähigen Fabrik. In wt Werkstattstechnik, 2002, 92 (2002); S. 122–127
- [Wie14] Wiendahl, H.-P.: Betriebsorganisation für Ingenieure. Mit 3 Tabellen. Hanser, München, 2014
- [Wik17] Wikipedia: Comparison of agent-based modeling software - Wikipedia. <https://en.wikipedia.org/w/index.php?oldid=760609941>, 29.01.2017
- [Wit11] Wittmann, S.: Verfahren zur Simulation und Analyse der Auswirkungen toleranzbedingter Bauteilabweichungen. Univ.-Bibliothek, Erlangen, 2011
- [WJK00] Wooldridge, M.; Jennings, N. R.; Kinny, D.: The Gaia Methodology for Agent-Oriented Analysis and Design. In Autonomous Agents and Multi-Agent Systems, 2000, 3; S. 285–312
- [WJK99] Wooldridge, M.; Jennings, N. R.; Kinny, D.: A methodology for agent-oriented analysis and design: Proceedings of the Third International Conference on Autonomous Agents (Agents 99), Seattle, WA, 1999; S. 69–76

- [WJR07] Womack, J. P.; Jones, D. T.; Roos, D.: The machine that changed the world. [the story of lean production ; Toyota's secret weapon in the global car wars that is revolutionizing world industry]. Free Press, New York, NY, 2007
- [WJR92] Womack, J. P.; Jones, D. T.; Roos, D.: Die zweite Revolution in der Autoindustrie. Konsequenzen aus der weltweiten Studie aus dem Massachusetts Institute of Technology. Campus-Verl., Frankfurt/Main, 1992
- [WKS06] Waldmann et al. Hrsg.: Operations Research Proceedings. Selected Papers of the Annual International Conference of the German Operations Research Society (GOR), Jointly Organized with the Austrian Society of Operations Research (ÖGOR) and the Swiss Society of Operations Research (SVOR). Springer, Berlin, Heidelberg, 2006
- [WLF14] Wior, I.; Linnenberg, T.; Fay, A.: Stability Analysis of an Agent-Based Smart Grid Control Marketplace. In IFAC Proceedings Volumes, 2014, 47; S. 1452–1458
- [Woo02] Wooldridge, M. J.: An introduction to multiagent systems. Wiley, Chichester, 2002
- [Woo09] Wooldridge, M. J.: An introduction to multiagent systems. Wiley, Chichester, 2009
- [WSV13] Wannagat, A.; Schütz, D.; Vogel-Heuser, B.: Einsatz von Softwareagenten am Beispiel einer kontinuierlichen, hydraulischen Heizpresse. In (Göhner, P. Hrsg.): Agentensysteme in der Automatisierungstechnik. Springer, Berlin, 2013; S. 169–185
- [WSW11] Weyrich, M. et al.: Identification of mechatronic units based on an example of a flexible customized multi lathe machine tool. In (Mammeri, Z. Hrsg.): IEEE 16th Conference on Emerging Technologies & Factory Automation (ETFA), 2011. Toulouse, France, 5 - 9 Sept. 2011. IEEE, Piscataway, NJ, 2011; S. 1–4
- [WZB00] Westkämper, E. et al.: Ansätze zur Wandlungsfähigkeit von Produktionsunternehmen. Ein Bezugsrahmen für die Unternehmensentwicklung im turbulenten Umfeld. In Werkstattstechnik, 2000; S. 22–26
- [YVP14] Yang, C.-H.; Vyatkin, V.; Pang, C.: Model-Driven Development of Control Software for Distributed Automation. A Survey and an Approach. In IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2014, 44; S. 292–305
- [ZaL] Zawisza, J.; Lüder, A.: Integration kooperierender Multiagentensysteme: ifak (Hg.) 2018 – 15. Fachtagung EKA, 2. und 3. Mai 2018
- [ZaL17] Zawisza, J.; Lüder, A.: Integration von Steuerungsaufgaben entlang der Wertschöpfungskette durch kooperierende Multi-Agenten-Systeme. Expertenforum Agentensysteme: Industrielle Agenten - die flexiblen Allrounder für die Optimierung der Produktion, Anlagen und das Energiemanagement. Von Self-X über Deep Learning., Garching bei München, 2017

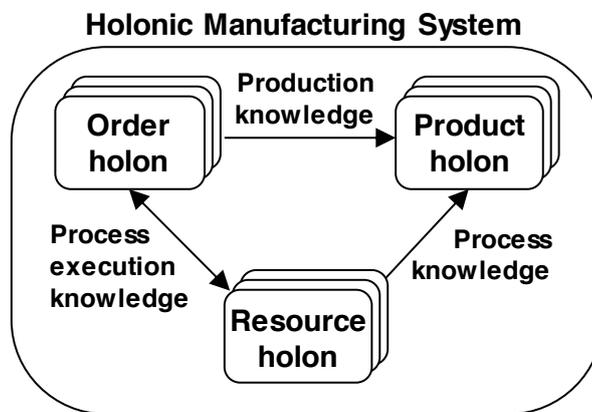
- [ZHL17] Zawisza, J.; Habermann, R.; Lüder, A.: Agentenbasierte Steuerung eines Routenzugsystems für die werksinterne Materiallogistik einer Automobilendmontage: 13. Magdeburger Maschinenbau-Tage 2017 autonom - vernetzt - nachhaltig, 27. und 28. September 2017 Tagungsband. Universitätsbibliothek, Magdeburg, 2017; S. 64–73
- [ZHR16] Zawisza, J. et al.: Generische Strukturierung von Produktionssystemen der Fertigungsindustrie: Automation 2016: 17. Branchentreff der Mess- und Automatisierungstechnik. Secure & reliable in the digital world: Kongresshaus Baden-Baden, 07. und 08. Juni 2016. VDI Verlag GmbH, Düsseldorf, 2016
- [Zhu05] Zhu, H.: Challenges to reusable services: IEEE International Conference on Services Computing, 2005. 11 - 15 July 2005, [Orlando, Florida]. IEEE Computer Society, Los Alamitos, Calif., 2005; 243-244 vol.2
- [ZLC18] Zawisza, J.; Lüder, A.; Gala, A.: Designing cooperating multi-agent systems — An extended design methodology: Proceedings 2018 IEEE Industrial Cyber-Physical Systems (ICPS). ITMO University, Saint Petersburg, Saint Petersburg, Russia, 15-18 May 2018. IEEE, Piscataway, NJ, 2018; S. 252–257
- [ZLR18] Zawisza, J.; Lüder, A.; Rosendahl, R.: Application of design pattern within the engineering of agent-based control systems: 14th IEEE International Conference on Automation Science and Engineering (CASE 2018). Knowledge-based automation, 2018
- [ZMV05] Zäh, M. F.; Möller, N.; Vogl, W.: Symbiosis of Changeable and Virtual Production. The Emperor's New Clothes or Key Factor for Future Success? In (Zäh, M.; Reinhart, G. Hrsg.): 1st International Conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV 2005). Utz, München, 2005
- [Zob05] Zobel, A.: Agilität im dynamischen Wettbewerb. Deutscher Universitätsverlag, Wiesbaden, 2005
- [Zoi09] Zoitl, A.: Real-time execution for IEC 61499. ISA; oooneida, Research Triangle Park, NC, Research Triangle Park, NC, 2009
- [ZoL14] Zoitl, A.; Lewis, R.: Modelling control systems using IEC 61499. The Institution of Engineering and Technology, London, 2014
- [ZoP13] Zoitl, A.; Prahofer, H.: Guidelines and Patterns for Building Hierarchical Automation Solutions in the IEC 61499 Modeling Language. In IEEE Transactions on Industrial Informatics, 2013, 9; S. 2387–2396

- [ZoS15] Zoitl, A.; Strasser, T.: Distributed control applications. Guidelines, design patterns, and application examples with the IEC 61499. CRC Press, Boca Raton, London, New York, 2015
- [ZRW03] Zambonelli, F.; R. Jennings, N.; Wooldridge, M.: Developing Multiagent Systems. The Gaia Methodology, 2003
- [ZVE10] ZVEI - Fachverband Automation Hrsg.: Manufacturing execution systems (MES). Branchenspezifische Anforderungen und herstellerneutrale Beschreibung von Lösungen. ZVEI, Frankfurt, 2010
- [ZVE15] ZVEI e.V.: Das Referenzarchitekturmodell Industrie 4.0 (RAMI 4.0). https://www.vdi.de/fileadmin/user_upload/VDI-GMA_Statusreport_Referenzarchitekturmodell-Industrie40.pdf, Oktober 2016

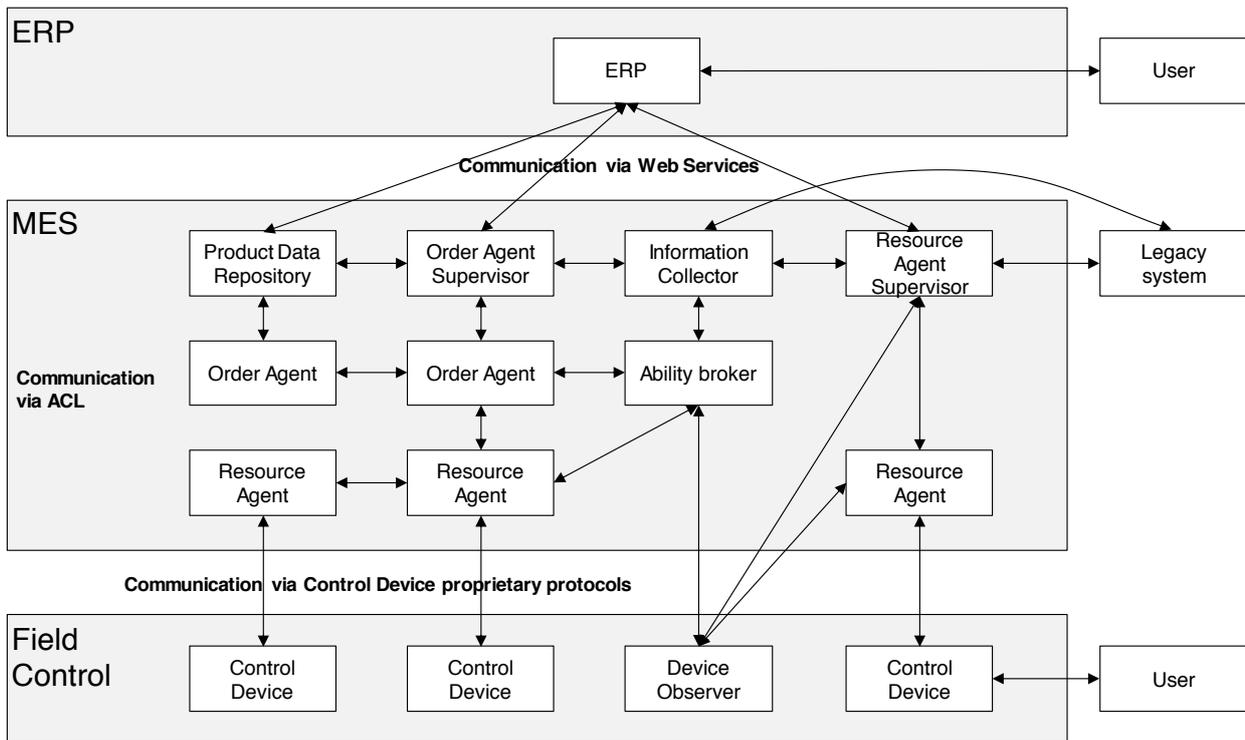
Anhang



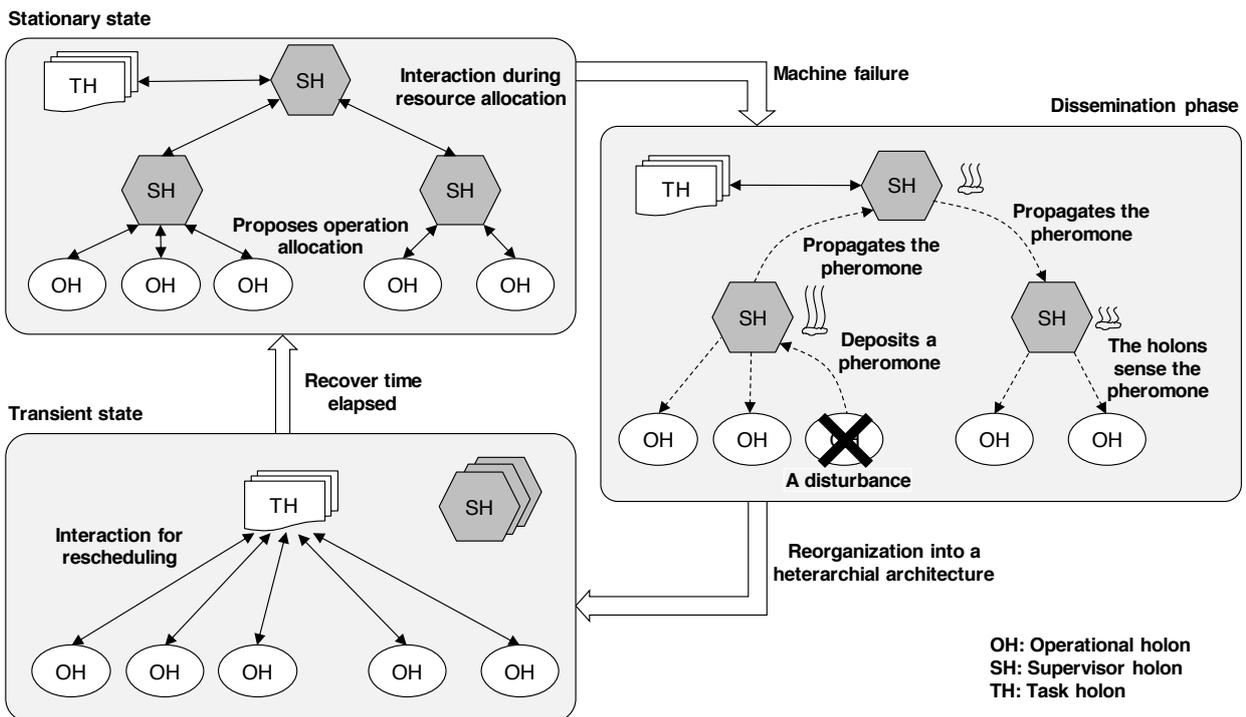
Anhang 1: Überblick der Programmiersprachen nach IEC 61131-3 i.A.a. [Fri09]



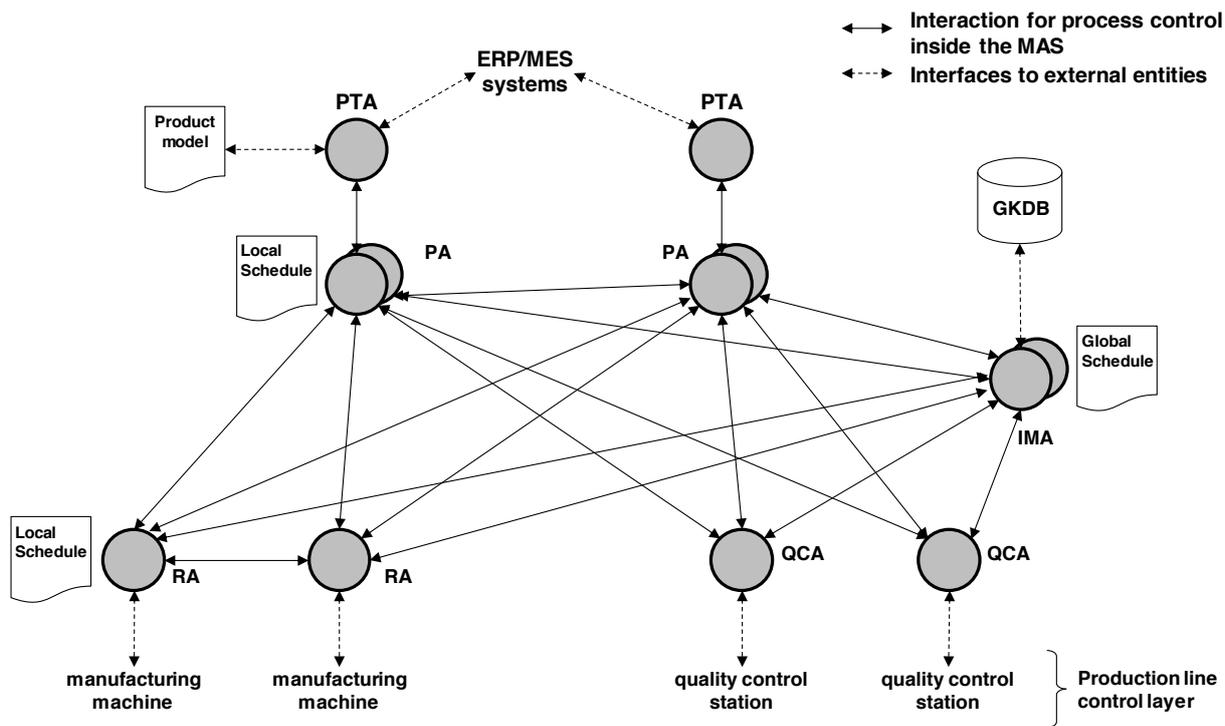
Anhang 2: Bausteine der PROSA-Architektur und ihre Beziehungen i.A.a. [VBW98]



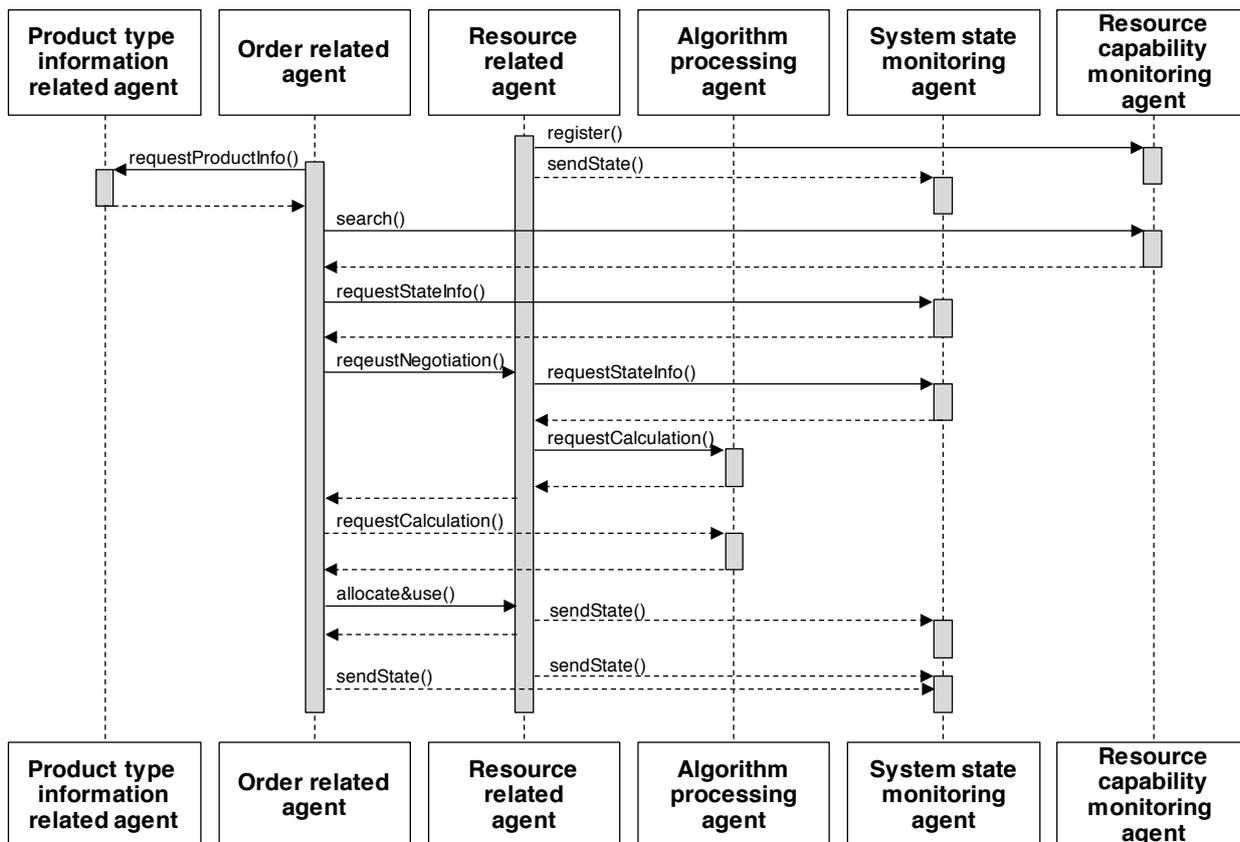
Anhang 3: Übersicht der PABADIS/PROMISE-Architektur i.A.a. [PAB08]



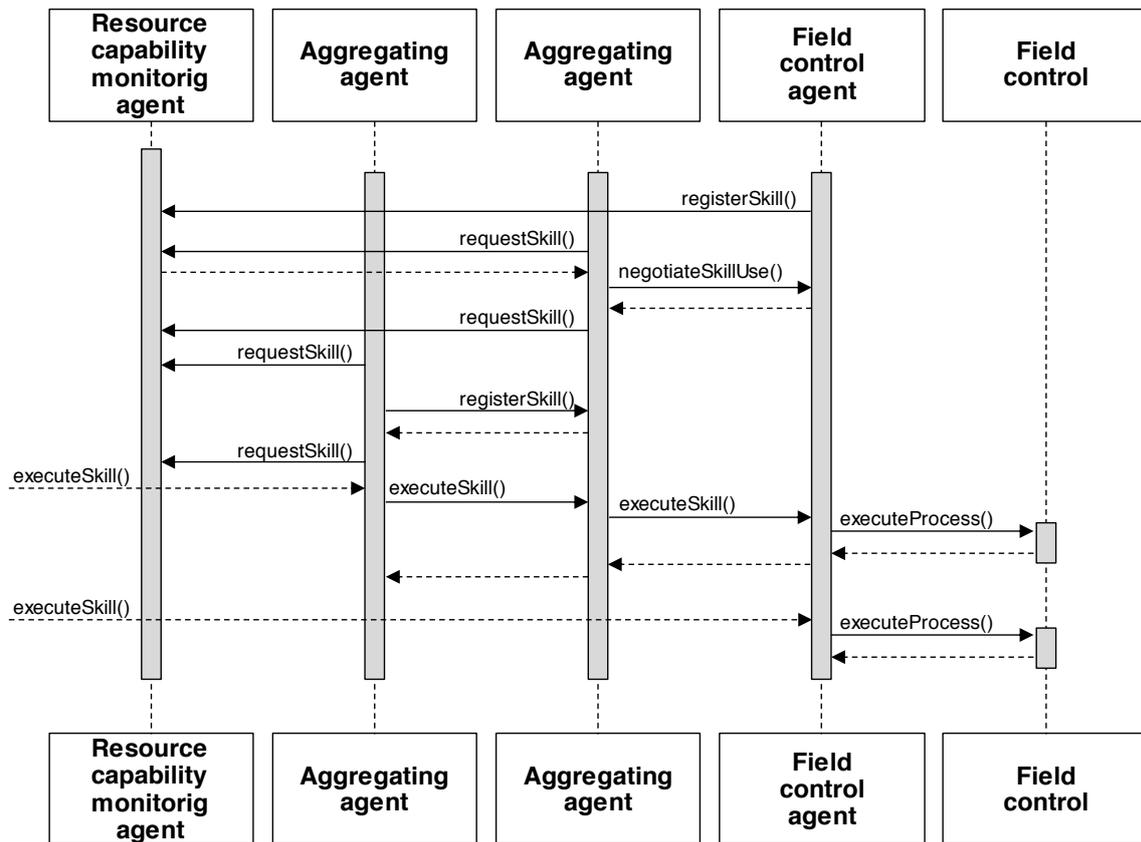
Anhang 4: Übersicht der ADACOR-Architektur i.A.a. [LCR05]



Anhang 5: Übersicht der GRACE-Architektur i.A.a. [CCF11]



Anhang 6: Entwurfsmuster für Ressourcenallokation (Implementierungsbsp.) i.A.a. [LCZ17]



Anhang 7: Entwurfsmuster für Ressourcenzugriff (Implementierungsbsp.) i.A.a. [LCZ17]

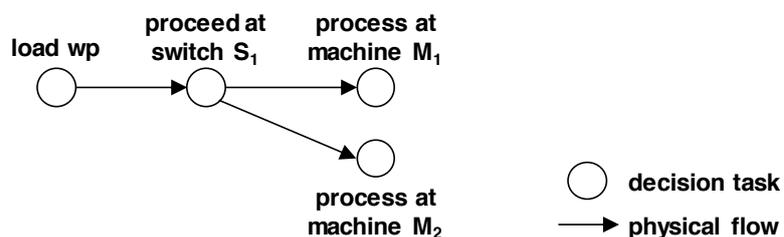
<i>Dam and Winikoff, 2003</i>	<i>Sturm et al., 2004</i>
<i>Concepts and properties</i>	
Autonomy	Autonomy
Reactive	Reactiveness
Proactive	Proactiveness
Mental attitudes	Mental notions (*)
Teamwork	Organisation (*)
Protocols	Protocol
<i>Modelling and notations</i>	
Consistency check	Analysability
Refinement	Complexity management
Language adequate and expressive	Expressiveness
Easy to use + easy to learn + clear notation	Accessibility
<i>Process</i>	
Lifecycle coverage (*)	Lifecycle coverage
<i>Pragmatics</i>	
Domain specific	Domain applicability
Scalable	Scalability
Maturity (quality) (*)	Resources

Anhang 8: Attribute für die Bewertung von AOSE-Methoden i.A.a. [CeZ11]

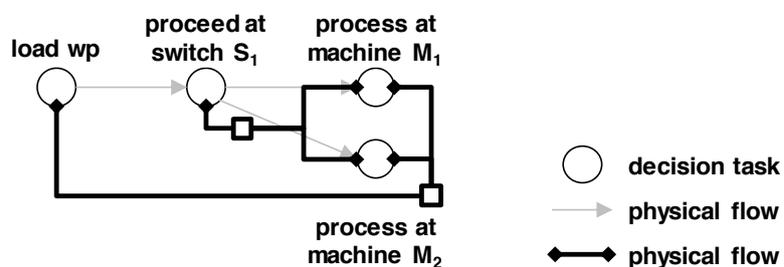
Slot	Description
id	unique identifier
params	subject and object of decision
trigger	situation that triggers decision
decision space	set of possible choices

Anhang 9: Schema zur Beschreibung von effektorischen Entscheidungen i.A.a. [BJW01]

Slot	Description
id	#2
params	switch S_1 , work piece
trigger	work piece at entry
decision space	{left, right}

Anhang 10: Beispiel einer effektorischen Entscheidung bei Switch S_1 i.A.a. [BJW01]

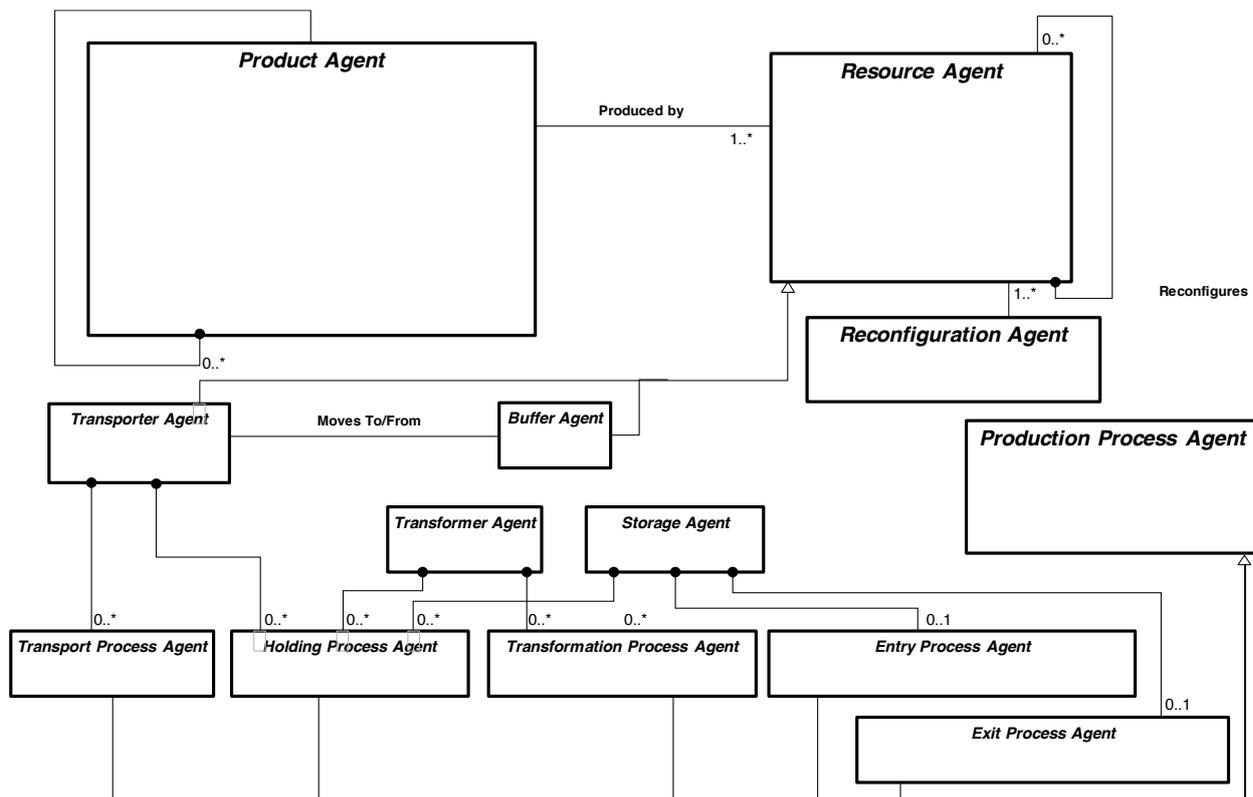
Anhang 11: Trigger Diagramm i.A.a. [BJW01]



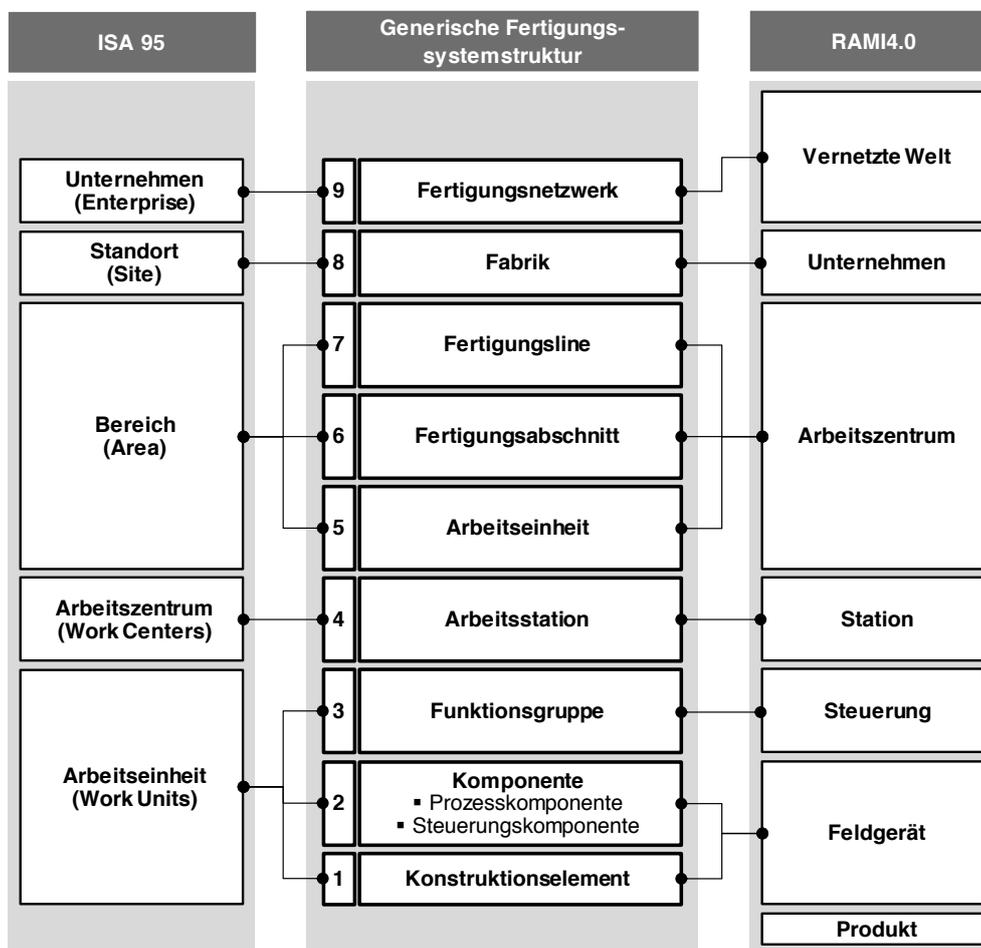
Anhang 12: Abhängigkeitsdiagramm i.A.a. [BJW01]

Prinzip	Beschreibung
Prinzip 1	Anwendung von Unabhängigkeitsaxioms: Beschreiben Sie explizit die Agentenarchitektur im Hinblick auf die Produktionsfreiheitsgrade des Produktionssystems.
Prinzip 2	Existenz physischer Agenten: Als Entscheidungs- und Kontrollsystem muss das Agentensystem eine 1:1-Beziehung zu den in der Fertigung vorhandenen physischen Kapazitäten haben.
Prinzip 3	Heterogenität: Die Produktionsfreiheitsgrade innerhalb der Agentenarchitektur müssen die Heterogenität der in der Produktion vorhandenen Fähigkeiten respektieren, seien es verschiedene Arten von Verarbeitungs-, Transport- oder Lagerungsprozessen.
Prinzip 4	Physikalische Aggregation: Die Agentenarchitektur muss die physische Aggregation der von ihnen repräsentierten Objekte widerspiegeln.
Prinzip 5	Verfügbarkeit: Die Agentenarchitektur muss explizit das Potenzial für sequenzunabhängige Einschränkungen modellieren, die die Verfügbarkeit eines bestimmten Produktionsfreiheitsgrades behindern.
Prinzip 6	Interaktion: Die Agentenarchitektur muss Agenteninteraktionen eine minimale Menge an physischen sequenzabhängigen Einschränkungen enthalten.
Prinzip 7	Maximales Rekonfigurationspotenzial: Abgesehen von der minimalen Menge an physische sequenzabhängigen Einschränkungen sollte die Agentenarchitektur keine weiteren Agenteninteraktionen einführen (die weitere Einschränkungen mit sich bringen können).
Prinzip 8	Umfang der physischen Agenten: Richten Sie den Umfang und die Grenzen der Agenten an den entsprechenden physischen Ressourcen und den damit verbundenen Produktionsfreiheitsgraden aus.
Prinzip 9	Verkapselung: Die Informationen des Produktivsystems sollten in dem Agenten, der der physischen Entität entspricht, die er beschreibt, eingelagert werden.
Prinzip 10	Rekonfigurationsmethode: Ein und derselbe Rekonfigurationsprozess kann je nach der für die Durchführung der Rekonfiguration verwendeten Methode (und nicht nur der rekonfigurierten Ressourcen) erheblich unterschiedlichen Aufwand (gemessen in Zeit, Kosten oder Energie) erfordern.

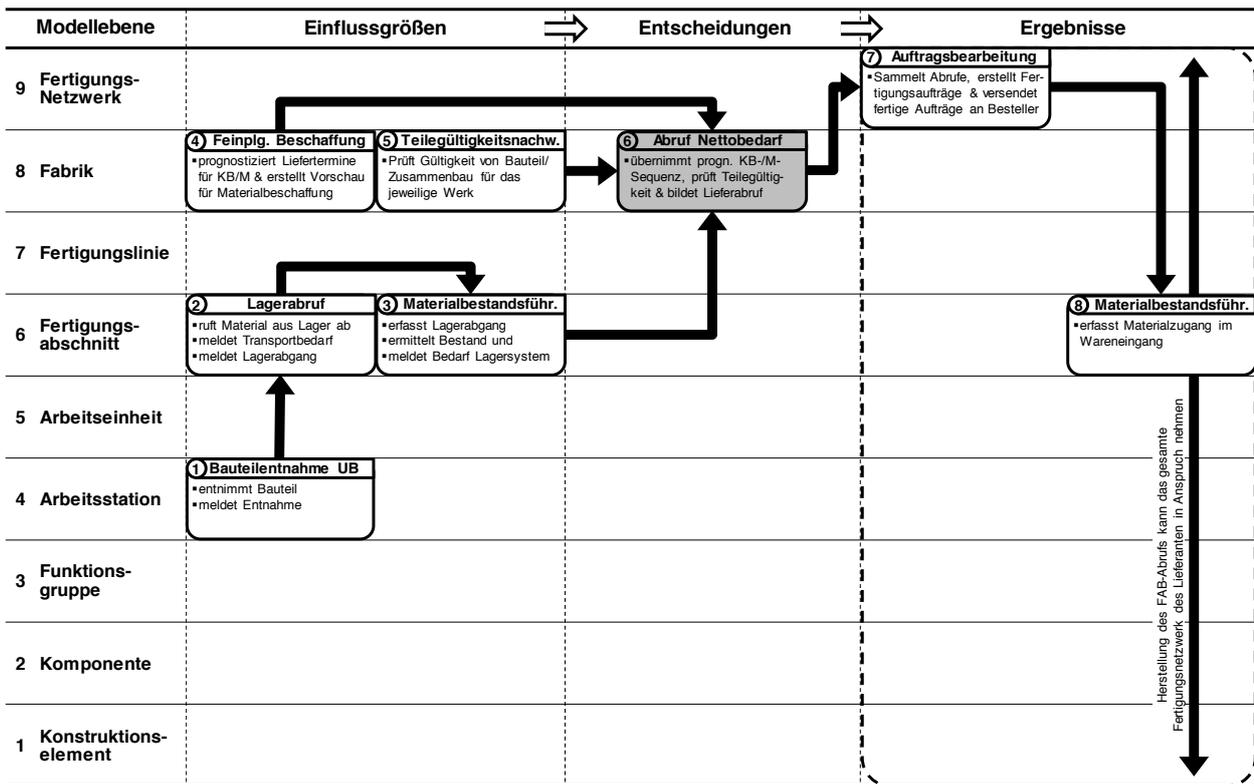
Anhang 13: Zehn Designprinzipien der ADMARMS-Methode i.A.a. [FaR15]



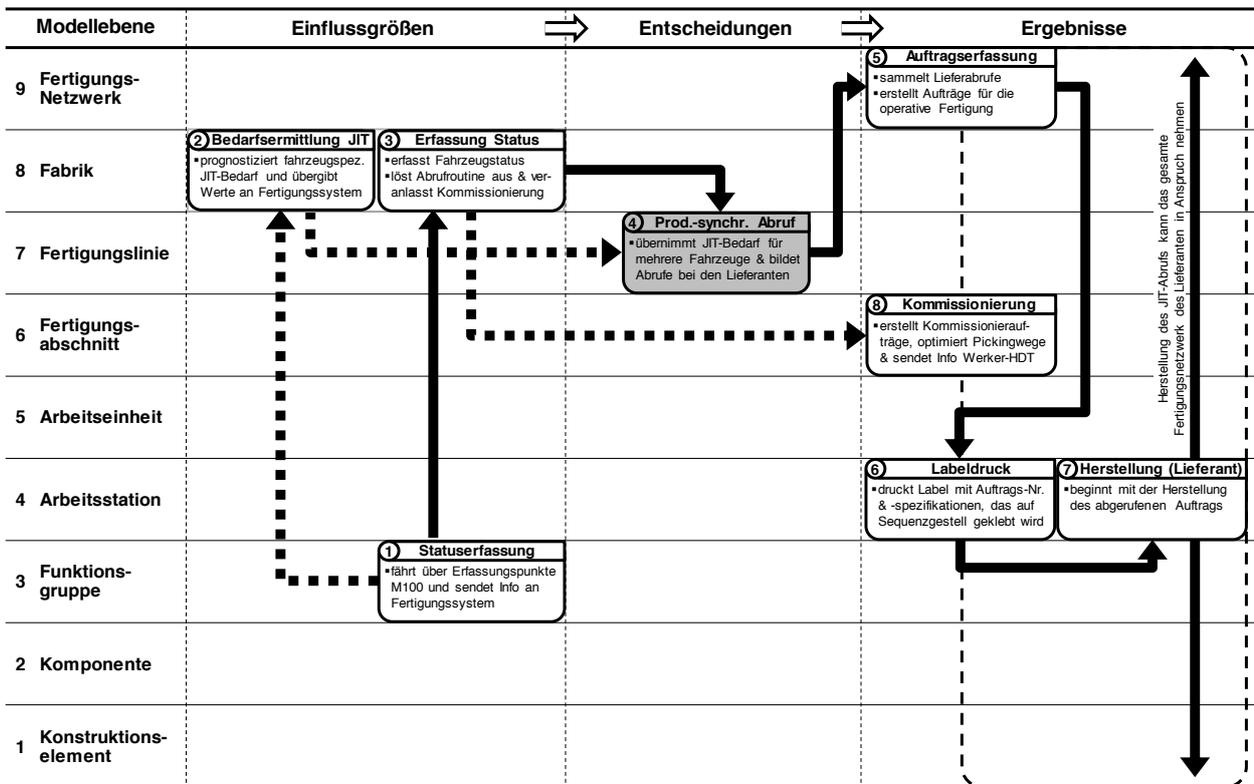
Anhang 14: ADMARMS-Architektur und Datenmodell i.A.a. [FaR15]



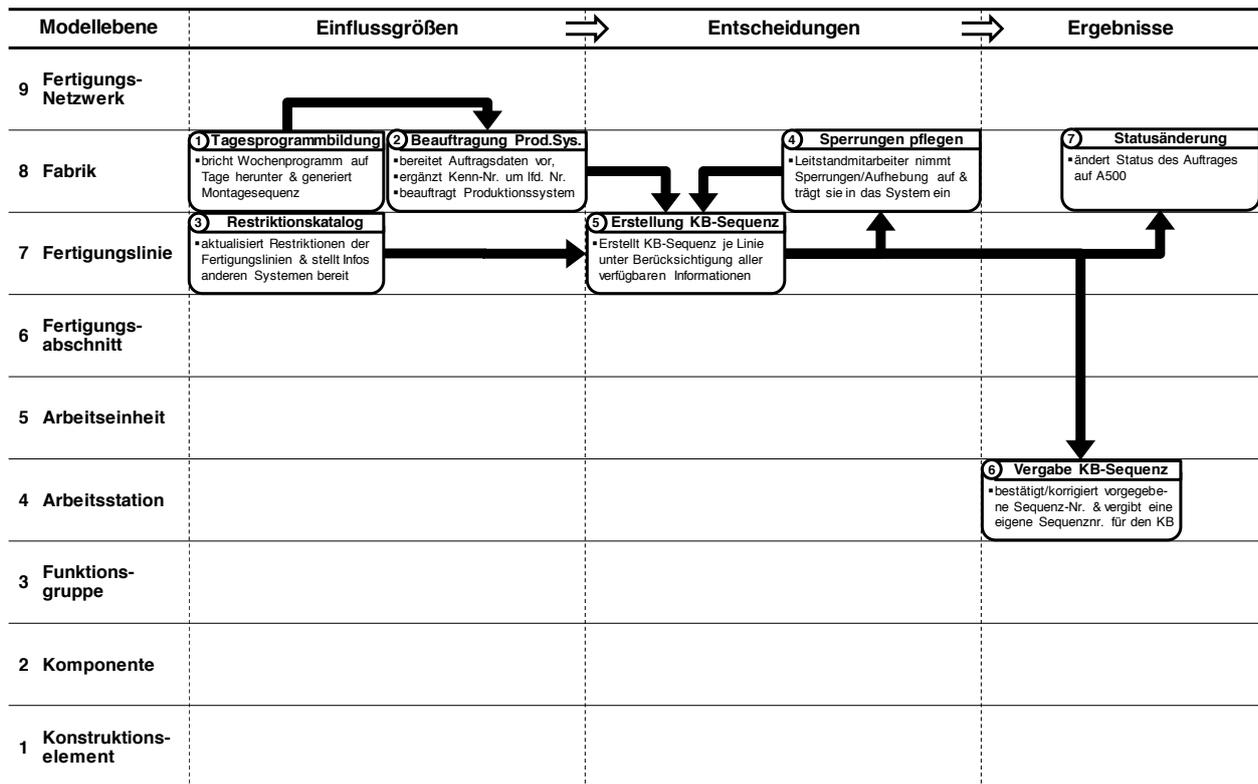
Anhang 15: Ebenenzuordnung von ISA 95 und RAMI4.0 zur GFSS



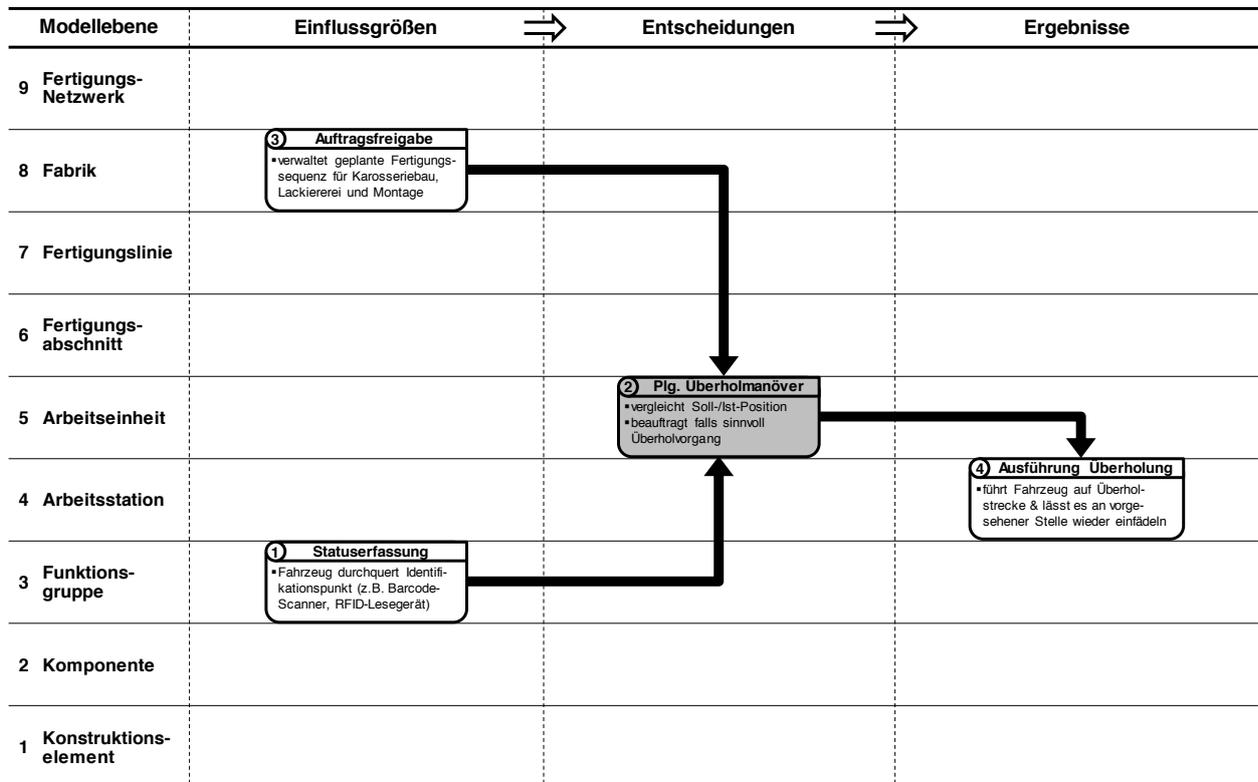
Anhang 16: Feinabruf des lieferantenspezifischen Nettobedarfs



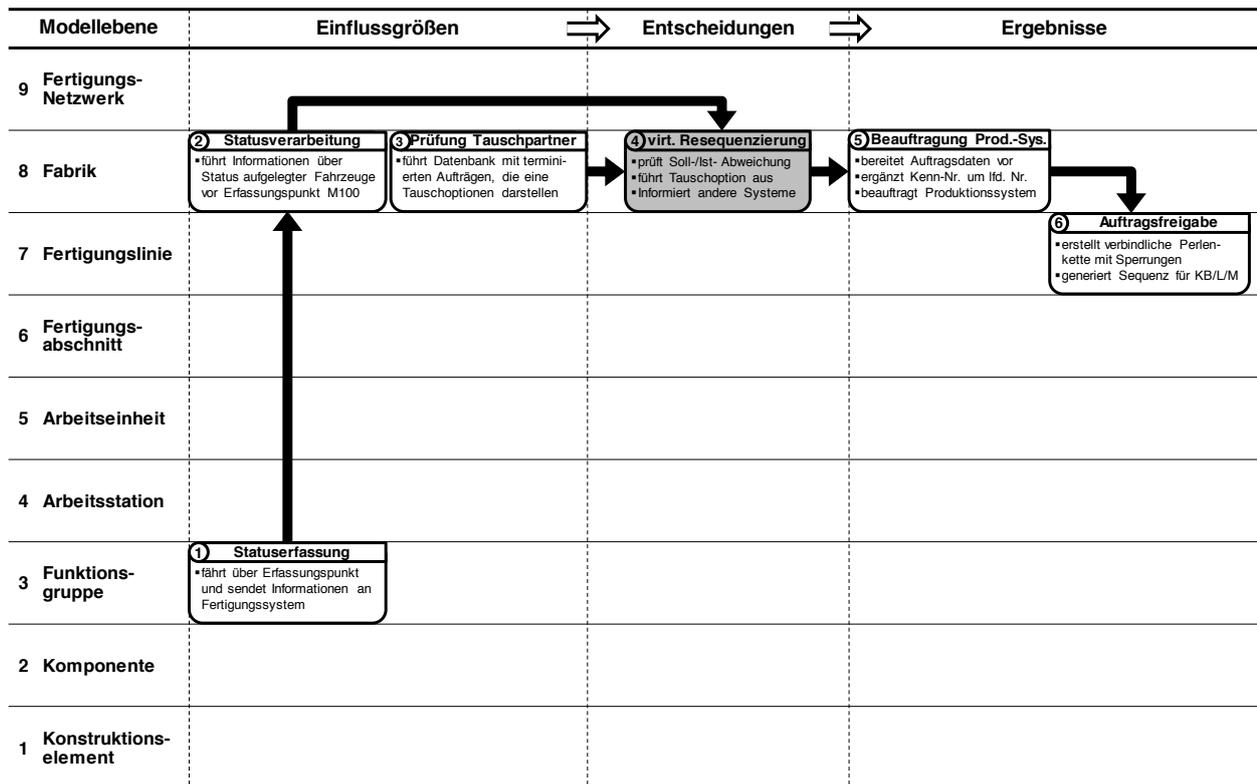
Anhang 17: Produktionssynchroner Abruf bei Montageeinlauf



Anhang 18: Erstellung Karosseriebausequenz



Anhang 19: Physische Resequenzierung mit Überholstrecke im Karosseriebau



Anhang 20: Virtuelle Resequenzierung im Karosseriebau

Aspekt	Klassifikationskriterium	Mögliche Ausprägungen
Startsituation	1. Anzahl involvierter Agenten	<ul style="list-style-type: none"> Anzahl n feste Anzahl variable Anzahl
	2. Kompatibilität der Beschränkungen & Präferenzen	<ul style="list-style-type: none"> nur Beschränkungen kompatible Präferenzen gegensätzliche Präferenzen
	3. Globale Beschränkungen & Präferenzen	<ul style="list-style-type: none"> global nicht-lokal
Gemeinsame Verpflichtungen	4. Anzahl gemeinsamer Verpflichtungen	<ul style="list-style-type: none"> Anzahl n feste Anzahl variable Anzahl
	5. Größe gemeinsamer Verpflichtungen	<ul style="list-style-type: none"> Anzahl n feste Anzahl variable Anzahl
	6. Beziehung zwischen gemeinsamen Verpflichtungen	<ul style="list-style-type: none"> überschneidend/nicht überschneidend vollständige/unvollständige Deckung
Agentenrollen	7. Rollenzuweisung (Anzahl variabler Rollen)	<ul style="list-style-type: none"> fest (keine) variabel (Teilmenge) variabel (Anzahl n) variabel (alle)
Prozessanforderungen	8. Informationsverfügbarkeit	<ul style="list-style-type: none"> nur Alternativen Präferenzen
	9. Delegation von Entscheidungskompetenz	<ul style="list-style-type: none"> keine Delegation partielle Delegation

Anhang 21: Klassifikationskriterien von Abhängigkeiten i.A.a. [BJW04]

Kriterium	Abstimmung		Verhandlung			Auktion		
	Plurality Voting Protocol	Clarke Tax Protocol	Service-oriented Negotiation	Monotonic Concession Protocol	DECIDE Conflict Resolution Protocol	English Auction	Contract Net Protocol	Continuous Double Auction
1	fest	fest	2	2	fest	fest	fest	variabel
2	gegensätzlich	gegensätzlich	gegensätzlich	gegensätzlich	Nur Beschränkungen	gegensätzlich	kompatibel	gegensätzlich
3	nicht-lokal	nicht-lokal	nicht-lokal	nicht-lokal	nicht-lokal	nicht-lokal	nicht-lokal	nicht-lokal
4	1	1	1	1	1	1	1	variabel
5	fest	fest	2	2	fest	2	2	2
6	-	-	-	-	-	-	-	nicht überschneidend, unvollständig
7	variabel (alle)	variabel (alle)	fest (keine)	fest (keine)	fest (keine)	variabel (1)	variabel (1)	variabel (alle)
8	nur Alternativen	Präferenzen	nur Alternativen	Präferenzen	nur Alternativen	nur Alternativen	nur Alternativen	nur Alternativen
9	partiell	partiell	keine	keine	keine	partiell	partiell	keine

Anhang 22: Charakterisierung ausgewählter Interaktionsprotokolle i.A.a. [BJW04] – Teil 1

Kriterium	Verteilte Beschränkungs-Optimierung	Koalition	Koordination von Multi-Agenten-Plänen		Anwendungsspezifische Protokolle	Einfache Protokolle
	Asynchronous Backtracking & weak-commitment search	Coalition Formation Algorithm	Partial Global Planning	Consensus-Based Distributed Planning	Kowest Work-in-process Control Protocol	Requesting Action Protocol
1	fest	fest	variabel	fest	variabel	2
2	nur Beschränkungen	kompatibel	kompatibel	gegensätzlich	kompatibel	kompatibel
3	global	nicht-lokal	global	nicht-lokal	global	nicht-lokal
4	variabel	variabel	variabel	1	variabel	1
5	variabel	variabel	variabel	fest	2	2
6	beliebig	überschneidend, unvollständig	beliebig	-	überschneidend	-
7	variabel (alle)	variabel (alle)	fest (keine)	variabel (alle)	variabel (Teilmenge)	fest (keine)
8	nur Alternativen	nur Alternativen	Präferenzen	Präferenzen	nur Alternativen	nur Alternativen
9	partiell	partiell	partiell	partiell	partiell	keine

Anhang 23: Charakterisierung ausgewählter Interaktionsprotokolle i.A.a. [BJW04] – Teil 2

Kriterien	Plurality Voting Protocol	Clarke Tax Protocol	Service-oriented Negotiation	Monotonic Concession Protocol	DECIDE Conflict Resolution Protocol	English Auction	Contract Net Protocol	Continuous Double Auction	Asynchronous Backtracking and weak-commitment search	Coalition Formation Algorithm	Partial Global Planning	Consensus-Based Distributed Planning	Kowest Work-in-process Control Protocol	Requesting Action Protocol
1	(✓)	(✓)	✓	✓	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)	✓
2	(✓)	(✓)	(✓)	(✓)	-	(✓)	✓	(✓)	-	✓	✓	(✓)	✓	✓
3	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	-	✓	-	✓
4	✓	✓	✓	✓	✓	✓	✓	(✓)	(✓)	(✓)	(✓)	✓	(✓)	✓
5	(✓)	(✓)	✓	✓	(✓)	✓	✓	✓	(✓)	(✓)	(✓)	(✓)	✓	✓
6	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	✓	-	✓
7	-	-	✓	✓	✓	-	-	-	-	-	✓	-	-	✓
8	✓	-	✓	-	✓	✓	✓	✓	✓	✓	-	-	✓	✓
9	✓	✓	-	-	-	✓	✓	-	✓	✓	✓	✓	✓	-
Σ	8 (5)	7 (4)	8 (7)	7 (6)	7 (5)	8 (6)	8 (7)	6 (3)	5 (2)	7 (4)	6 (3)	7 (4)	6 (4)	8

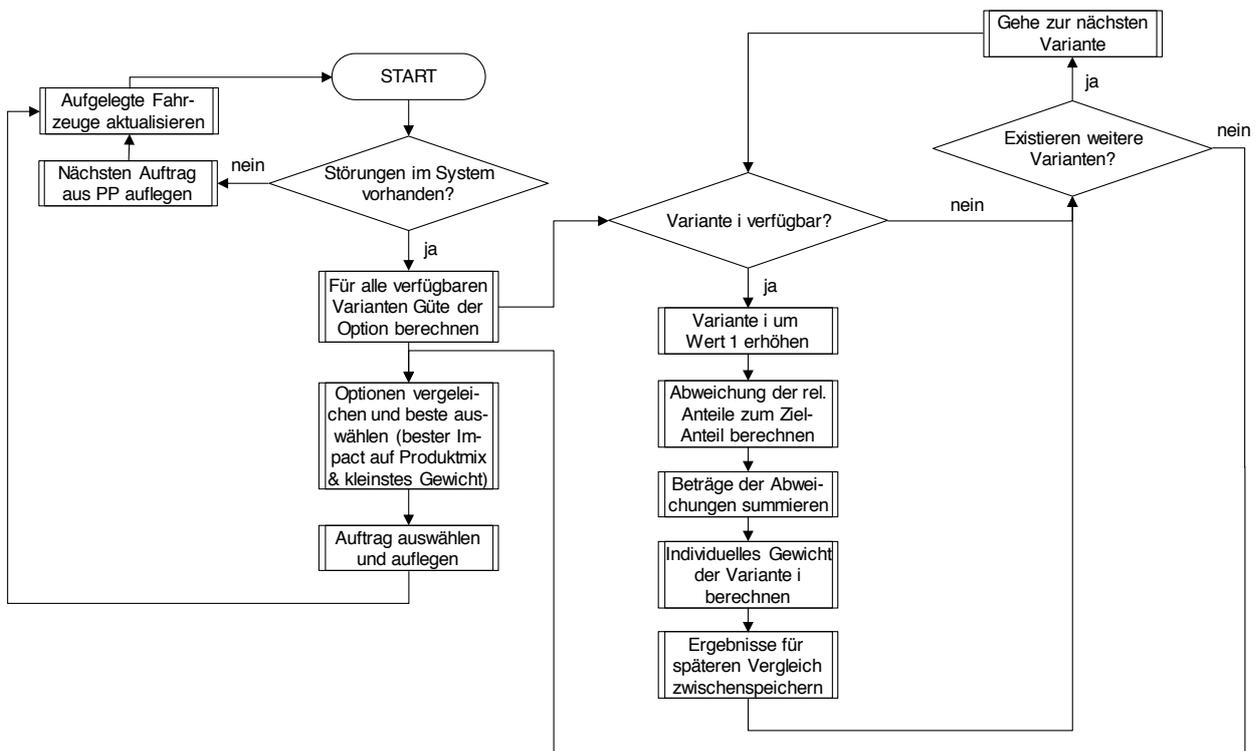
✓ = erfüllt (✓) = abwärtskompatibel - = nicht erfüllt

Anhang 24: KFS – Abgleich der Anforderungen von A1/A2/A3 mit Interaktionsprotokollen

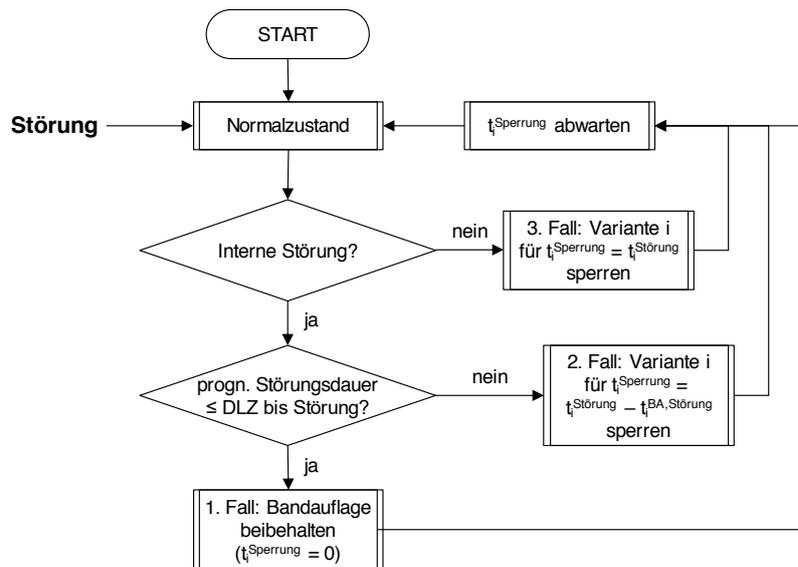
Kriterien	Plurality Voting Protocol	Clarke Tax Protocol	Service-oriented Negotiation	Monotonic Concession Protocol	DECIDE Conflict Resolution Protocol	English Auction	Contract Net Protocol	Continuous Double Auction	Asynchronous Backtracking and weak-commitment search	Coalition Formation Algorithm	Partial Global Planning	Consensus-Based Distributed Planning	Kowest Work-in-process Control Protocol	Requesting Action Protocol
1	-	-	-	-	-	-	-	✓	-	-	✓	-	✓	-
2	✓	✓	✓	✓	-	✓	-	✓	-	-	-	✓	-	-
3	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	-	✓	-	✓
4	-	-	-	-	-	-	-	✓	✓	✓	✓	-	✓	-
5	(✓)	(✓)	✓	✓	(✓)	✓	✓	✓	(✓)	(✓)	(✓)	(✓)	✓	✓
6	-	-	✓	✓	✓	✓	✓	✓	(✓)	✓	(✓)	✓	(✓)	✓
7	-	-	✓	✓	✓	-	-	-	-	-	✓	-	-	✓
8	✓	-	✓	-	✓	✓	✓	✓	✓	✓	-	-	✓	✓
9	✓	✓	-	-	-	✓	✓	-	✓	✓	✓	✓	✓	-
Σ	5 (4)	4 (3)	6	5	5 (4)	6	5	7	5 (3)	6 (5)	6 (4)	5 (4)	6 (5)	5

✓ = erfüllt (✓) = abwärtskompatibel - = nicht erfüllt

Anhang 25: KFS – Abgleich der Anforderungen von A4 mit Interaktionsprotokollen



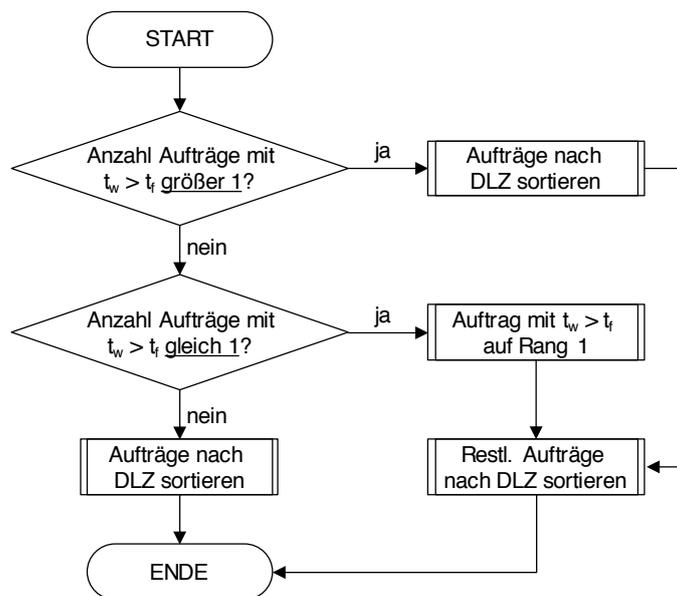
Anhang 26: KFS – Adaptionalgorithmus des Scheduling-Agenten



Anhang 27: KFS – Zustandsdiagramm des Scheduling-Agenten

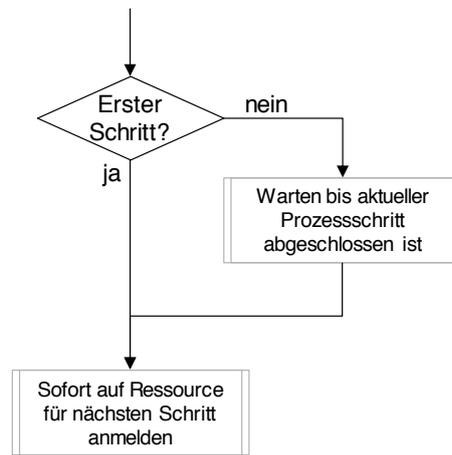
Fall 1: 1 Auftrag, 2 Ressourcen				Fall 2: 2 Aufträge, 1 Ressource			
Nachfolger 2	Nachfolger 1	Auftrag	Gewinner	Auftrag 2	Auftrag 1	Vorgänger	Gewinner
A	A	A	indifferent	A	A	A	indifferent
B	A	A	R1	B	A	A	B
C	A	A	R2	C	A	A	C
A	B	A	R2	A	B	A	B
B	B	A	R2	B	B	A	indifferent
C	B	A	R2	C	B	A	indifferent
A	C	A	R1	A	C	A	C
B	C	A	R1	B	C	A	indifferent
C	C	A	R1	C	C	A	indifferent
A	A	B	R1	A	A	B	indifferent
B	A	B	R1	B	A	B	A
C	A	B	R1	C	A	B	C
A	B	B	R1	A	B	B	A
B	B	B	R1	B	B	B	indifferent
C	B	B	R1	C	B	B	C
A	C	B	R1	A	C	B	C
B	C	B	R1	B	C	B	C
C	C	B	R1	C	C	B	indifferent
A	A	C	R2	C	A	A	indifferent
B	A	C	R2	B	A	C	B
C	A	C	R2	C	A	C	A
A	B	C	R2	A	B	C	B
B	B	C	R2	B	B	C	indifferent
C	B	C	R2	C	B	C	B
A	C	C	R2	A	C	C	A
B	C	C	R2	B	C	C	B
C	C	C	R2	C	C	C	indifferent

Anhang 28: KFS – Heuristik zur Vermeidung von Engpässen

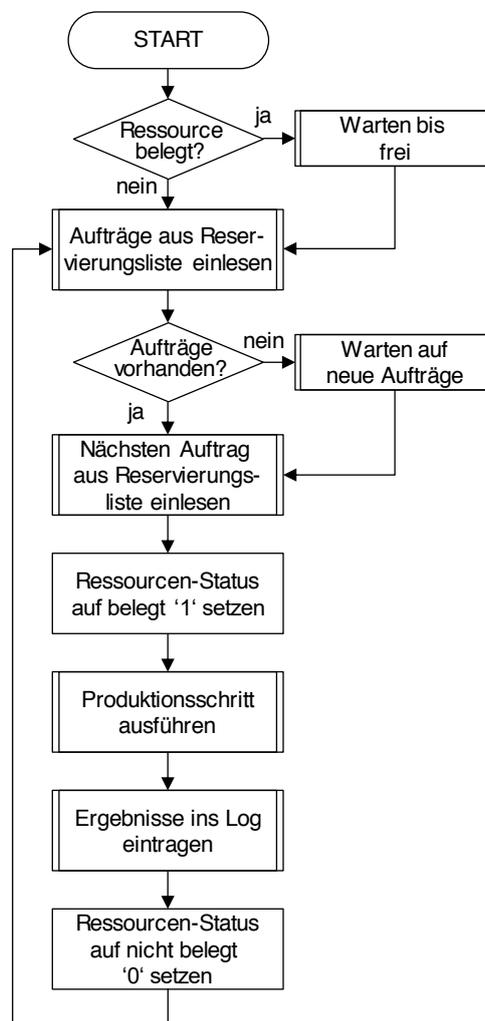


t_w = Wartezeit des Auftrags seit Produktionsanfrage t_r = Dauer des Reservierungszeitfensters

Anhang 29: KFS – Algorithmus zur Priorisierung von Aufträgen



Anhang 30: KFS – Ablaufdiagramm Auftragsagent



Anhang 31: KFS – Ablaufdiagramm Ressourcenagent

Kriterien	Plurality Voting Protocol	Clarke Tax Protocol	Service-oriented Negotiation	Monotonic Concession Protocol	DECIDE Conflict Resolution Protocol	English Auction	Contract Net Protocol	Continuous Double Auction	Asynchronous Backtracking and weak-commitment search	Coalition Formation Algorithm	Partial Global Planning	Consensus-Based Distributed Planning	Kowest Work-in-process Control Protocol	Requesting Action Protocol
1	(✓)	(✓)	✓	✓	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)	✓
2	(✓)	(✓)	(✓)	(✓)	-	(✓)	✓	(✓)	-	✓	✓	(✓)	✓	✓
3	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	-	✓	-	✓
4	✓	✓	✓	✓	✓	✓	✓	(✓)	(✓)	(✓)	(✓)	✓	(✓)	✓
5	(✓)	(✓)	✓	✓	(✓)	✓	✓	✓	(✓)	(✓)	(✓)	(✓)	✓	✓
6	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	✓	-	✓
7	-	-	✓	✓	✓	-	-	-	-	-	✓	-	-	✓
8	✓	-	✓	-	✓	✓	✓	✓	✓	✓	-	-	✓	✓
9	✓	✓	-	-	-	✓	✓	-	✓	✓	✓	✓	✓	-
Σ	8 (5)	7 (4)	8 (7)	7 (6)	7 (5)	8 (6)	8 (7)	6 (3)	5 (2)	7 (4)	6 (3)	7 (4)	6 (4)	8

✓ = erfüllt (✓) = abwärtskompatibel - = nicht erfüllt

Anhang 32: RZS – Abgleich der Anforderungen von A1/A2/A5 mit Interaktionsprotokollen

Kriterien	Plurality Voting Protocol	Clarke Tax Protocol	Service-oriented Negotiation	Monotonic Concession Protocol	DECIDE Conflict Resolution Protocol	English Auction	Contract Net Protocol	Continuous Double Auction	Asynchronous Backtracking and weak-commitment search	Coalition Formation Algorithm	Partial Global Planning	Consensus-Based Distributed Planning	Kowest Work-in-process Control Protocol	Requesting Action Protocol
1	✓	✓	-	-	✓	✓	✓	(✓)	✓	✓	(✓)	✓	(✓)	-
2	(✓)	(✓)	(✓)	(✓)	-	(✓)	✓	(✓)	-	✓	✓	(✓)	✓	✓
3	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	-	✓	-	✓
4	✓	✓	✓	✓	✓	✓	✓	(✓)	(✓)	(✓)	(✓)	✓	(✓)	✓
5	✓	✓	-	-	✓	-	-	-	(✓)	(✓)	(✓)	✓	-	-
6	✓	✓	✓	✓	✓	✓	✓	-	(✓)	-	(✓)	✓	-	✓
7	-	-	-	-	-	-	-	-	-	-	-	-	✓	-
8	✓	-	✓	-	✓	✓	✓	✓	✓	✓	-	-	✓	✓
9	-	-	-	-	✓	-	-	✓	-	-	-	-	-	✓
Σ	7 (6)	6 (5)	5 (4)	4 (3)	7	6 (5)	6	6 (3)	5 (2)	6 (4)	5 (1)	6 (5)	5 (3)	6

✓ = erfüllt (✓) = abwärtskompatibel - = nicht erfüllt

Anhang 33: RZS – Abgleich der Anforderungen von A3 mit Interaktionsprotokollen

Experiment	Gewichtungsfaktor der Termintreue (- = 2; + = 8)	Gewichtungsfaktor der Fahrzeit (- = 2; + = 8)	Startzeit vor mittlerer Wunschzeit (- = 15; + 25 Min.)	Mindestanzahl an Aufträgen (1; 2; 3; 4)	Fahrzeit je Auftrag in Minuten	Terminabweichung in Minuten	Wartezeit der Routenzüge in Minuten	Zielfunktionswert in Minuten
1	-	-	-	1	8,386	7,389	11,084	8,057
2	-	-	-	2	7,689	7,770	14,733	8,442
3	-	-	-	3	7,357	8,026	19,460	8,969
4	-	-	-	4	6,623	8,331	23,806	9,366
5	-	-	+	1	9,213	14,600	7,685	12,293
6	-	-	+	2	8,096	15,358	12,389	12,882
7	-	-	+	3	7,201	15,464	17,991	13,238
8	-	-	+	4	6,625	15,075	24,647	13,497
9	-	+	-	1	8,205	8,080	11,368	8,446
10	-	+	-	2	7,550	8,320	14,944	8,751
11	-	+	-	3	6,853	8,651	19,950	9,242
12	-	+	-	4	6,254	9,170	25,616	9,940
13	-	+	+	1	9,136	15,250	7,824	12,673
14	-	+	+	2	8,042	15,927	12,366	13,205
15	-	+	+	3	7,038	15,962	18,393	13,528
16	-	+	+	4	6,387	15,778	24,319	13,815
17	+	-	-	1	8,539	7,210	10,602	7,948
18	+	-	-	2	7,870	7,529	14,188	8,297
19	+	-	-	3	7,243	7,662	18,646	8,635
20	+	-	-	4	6,811	8,015	22,998	9,152
21	+	-	+	1	9,274	14,099	7,667	12,008
22	+	-	+	2	8,232	15,063	11,766	12,684
23	+	-	+	3	7,338	15,093	17,611	13,018
24	+	-	+	4	6,807	14,620	23,204	13,135
25	+	+	-	1	8,386	7,389	11,084	8,057
26	+	+	-	2	7,689	7,770	14,733	8,442
27	+	+	-	3	7,024	8,026	19,460	8,869
28	+	+	-	4	6,623	8,331	24,039	9,389
29	+	+	+	1	9,213	14,600	7,685	12,293
30	+	+	+	2	8,096	15,358	12,389	12,882
31	+	+	+	3	7,201	15,464	17,991	13,238
32	+	+	+	4	6,625	15,075	24,647	13,497

Anhang 34: RZS -Ergebnisse der ersten Versuchsreihe i.A.a. [Hab16]

Experiment	Gewichtungsfaktor der Termintreue	Gewichtungsfaktor der Fahrzeit	Startzeit vor der mittleren WZ in Minuten	Mindestanzahl an Aufträgen	Fahrzeit je Auftrag in Minuten	Terminabweichung in Minuten	Wartezeit der Routenzüge in Minuten	Zielfunktionswert in Minuten
1	8	2	15	4	6,811	8,015	22,998	9,152
2	8	2	10	4	6,795	6,977	23,295	8,554
3	8	2	5	4	6,760	9,396	23,112	9,977
4	7	2	10	4	6,785	6,966	23,261	8,542
5	6	2	10	4	6,774	6,953	23,410	8,545

Anhang 35: RZA – Ergebnisse der zweiten Versuchsreihe i.A.a. [Hab16]

Tag	Gewichtungsfaktor der Termintreue	Gewichtungsfaktor der Fahrzeit	Startzeit vor der mittleren WZ	Mindestanzahl an Aufträgen	Fahrzeit je Auftrag in Minuten	Terminabweichung in Minuten	Wartezeit der Routenzüge in Minuten	Zielfunktionswert in Minuten
1	7	2	10	4	6,785	6,966	23,261	8,542
2	7	2	10	4	6,737	6,584	22,426	8,214
3	7	2	10	4	6,754	6,701	27,444	8,791
4	7	2	10	4	6,660	6,651	22,004	8,189
5	7	2	10	4	6,693	7,396	17,971	8,243

Anhang 36: RZS – Ergebnisse der dritten Versuchsreihe i.A.a. [Hab16]

Kriterien	Plurality Voting Protocol	Clarke Tax Protocol	Service-oriented Negotiation	Monotonic Concession Protocol	DECIDE Conflict Resolution Protocol	English Auction	Contract Net Protocol	Continuous Double Auction	Asynchronous Backtracking and weak-commitment search	Coalition Formation Algorithm	Partial Global Planning	Consensus-Based Distributed Planning	Kowest Work-in-process Control Protocol	Requesting Action Protocol
1	(\surd)	(\surd)	\surd	\surd	(\surd)	(\surd)	(\surd)	(\surd)	(\surd)	(\surd)	(\surd)	(\surd)	(\surd)	\surd
2	(\surd)	(\surd)	(\surd)	(\surd)	-	(\surd)	\surd	(\surd)	-	\surd	\surd	(\surd)	\surd	\surd
3	\surd	\surd	\surd	\surd	\surd	\surd	\surd	\surd	-	\surd	-	\surd	-	\surd
4	\surd	\surd	\surd	\surd	\surd	\surd	\surd	(\surd)	(\surd)	(\surd)	(\surd)	\surd	(\surd)	\surd
5	(\surd)	(\surd)	\surd	\surd	(\surd)	\surd	\surd	\surd	(\surd)	(\surd)	(\surd)	(\surd)	\surd	\surd
6	\surd	\surd	\surd	\surd	\surd	\surd	\surd	-	(\surd)	-	(\surd)	\surd	-	\surd
7	-	-	\surd	\surd	\surd	-	-	-	-	-	\surd	-	-	\surd
8	\surd	-	\surd	-	\surd	\surd	\surd	\surd	\surd	\surd	-	-	\surd	\surd
9	\surd	\surd	-	-	-	\surd	\surd	-	\surd	\surd	\surd	\surd	\surd	-
Σ	8(5)	7(4)	8(7)	7(6)	7(5)	8(6)	8(7)	6(3)	6(2)	7(4)	7(3)	7(4)	6(4)	8

\surd = erfüllt (\surd) = abwärtskompatibel - = nicht erfüllt

Anhang 37: Integration – Abgleich der Anforderungen mit Interaktionsprotokollen