

Automatische Planung der Ablaufreihenfolge von Automatisierungsfunktionen

Christoph Matthias Johannes Legat

Vollständiger Abdruck der von der Fakultät für Maschinenwesen der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:

Prof. Dr. phil. Klaus Bengler

Prüfende der Dissertation:

1. Prof. Dr.-Ing. Birgit Vogel-Heuser
2. apl. Prof. Dr.-Ing. habil. Arndt Lüder, Otto-von-Guericke-Universität Magdeburg

Die Dissertation wurde am 04.05.2018 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 29.08.2018 angenommen.

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <<http://dnb.ddb.de>> abrufbar.

Automatische Planung der Ablaufreihenfolge von Automatisierungsfunktionen

Autor:
Christoph Legat

ISBN 13: 978-3-96548-016-2
1. Auflage 2019

Copyright
© sierke VERLAG
Friedländer Weg 54
37085 Göttingen
Tel.: +49 (0)551 5036647

Coverdesign: sierke MEDIA



Alle Rechte vorbehalten. Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlags unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Anwendungsbeispiele	3
1.1.1. Assistenzsystem für die Konfiguration von Steuerungssoftware	4
1.1.1.1. User Story 1: Anlagenkonfigurationsassistentz	4
1.1.1.2. Konzeptionelle Betrachtung	5
1.1.2. Dynamische Rekonfiguration der Anlage im Falle technischer Störungen	5
1.1.2.1. User Story 2: Assistenz zur dynamischen Störungskompensation durch Rekonfiguration des Steuerungsverhaltens	6
1.1.2.2. Technologische Betrachtung	7
1.2. Zielsetzung der Dissertation	9
1.3. Aufbau der Dissertation	9
2. Eingrenzung des Betrachtungsbereichs und Anforderungsermittlung	13
2.1. Eingrenzung des Betrachtungsbereichs	13
2.1.1. Automatische Planungsverfahren	13
2.1.2. Automatisierung industrieller Fertigungssysteme	16
2.1.3. Steuerungsstruktur industrieller Produktionssysteme	17
2.1.4. Automatisierungstechnik auf der Steuerungsebene	20
2.2. Anforderungen an den Lösungsansatz	24
2.2.1. Anforderung A1: Berücksichtigung unterschiedlicher, funktionaler Aspekte innerhalb eines Handlungsplans	25
2.2.2. Anforderung A2: Selektion eines einzelnen Ablaufplanes unter Berücksichtigung eines Gütekriteriums	29
2.2.3. Anforderung A3: Zeit- und Skalierbarkeitsverhalten des Planungsalgorithmus	31
2.2.4. Anforderung A4: Eignung des automatischen Planungsverfahrens für IEC 61131-3-basierte Steuerungssoftware	32
2.2.5. Anforderung A5: Möglichkeit der Integration mit modellbasierten Entwicklungsansätzen	33
3. Stand der Wissenschaft und Technik	35
3.1. Ansätze zur Formulierung und Lösung von Planungsproblemen	35
3.1.1. Logik-basierte Planungsverfahren	36
3.1.2. Constraintprogrammierung	39
3.1.2.1. Constraint-Satisfaction-/Constraint-Optimization-Problem	39
3.1.2.2. Lineare Programmierung	40
3.1.3. Stellen-/Transitionsnetze	44
3.1.3.1. Sequentielle Schaltsemantik und Erreichbarkeitsgraphen	45
3.1.3.2. Nebenläufige Schaltsemantik und Kausalnetze	46
3.2. Anwendung automatischer Planungsverfahren	48
3.2.1. Anwendung von Planungsverfahren in unterschiedlichen Software Architekturen	49
3.2.1.1. Dienstorientierte Architektur	50
3.2.1.2. Multi-Agenten Systeme und (Selbst-)Adaptive Systeme	53
3.2.1.3. Kombination von Dienstorientierter Archektur und Multi-Agenten Systemen	57
3.2.2. Anwendung von Planungsverfahren in der Anlagenautomatisierung	58
3.2.2.1. Handlungsplanung für Werkzeugmaschinen und robotergestützte Montage	58
3.2.2.2. Ansätze zur Bestimmung der Ausführungsreihenfolge von Softwarefunktionen SPS-basierter Steuerungssoftware	61

3.3. Zusammenfassung	63
4. Bestimmung der Ablaufreihenfolge von Automatisierungsfunktionen	67
4.1. Das HiTraP-AT Planungsverfahren (Anforderungen A1 bis A3)	67
4.1.1. Das interne HiTraP-AT Planungsmodell	68
4.1.1.1. Zusammenhängende Handlungsräume	69
4.1.1.2. Hierarchisierung des Zustandsraumes	71
4.1.1.3. Hierarchische, zusammenhängende Zustandsräume	76
4.1.2. Das HiTraP-AT Planungsproblem	80
4.1.3. Formulierung und Lösung des HiTraP-AT Planungsproblems mittels Linearer Programmierung	85
4.1.3.1. Bestimmung des Optimalitätskriterium	86
4.1.3.2. Formulierung des HiTraP-AT Planungsverfahrens als Lineares Programm	87
4.1.3.3. Bestimmung des Handlungsplans durch Lösung des Linearen Programms	97
4.2. Integration von HiTraP-AT in einen modellgetriebenen Entwicklungsansatz (Anforderung A5)	99
4.2.1. Die UML-OP	100
4.2.1.1. Das Komponentenmodell der UML-OP	100
4.2.1.2. Das Systemmodell der UML-OP	103
4.2.1.3. Das Produktmodell der UML-OP	104
4.2.2. Gegenüberstellung der UML-OP und des HiTraP-AT Planungsproblems	105
4.3. Anwendung von HiTraP-AT für IEC 61131-3 basierte Steuerungssoftware (Anforderung A4)	109
4.3.1. Realisierung von Steuerungscode mit statischen Handlungsplänen	110
4.3.2. Realisierung von Steuerungscode für konfigurierbare Handlungssequenzen	111
5. Untersuchung der Anwendbarkeit und Evaluation von HiTraP-AT	117
5.1. Untersuchung der Integration in einen modellgetriebenen Entwicklungsansatz (Anforderung A5)	117
5.2. Untersuchung von Zeit- und Skalierbarkeitsverhalten des HiTraP-AT Planungsalgorithmus (Anforderung A3)	119
5.2.1. Das HiTraP-AT Planungswerkzeug	121
5.2.2. Untersuchung flacher, zusammenhängender Handlungsräume	121
5.2.3. Untersuchung und Bewertung der Skalierbarkeitseigenschaft des HiTraP-AT Ansatzes	123
5.2.4. Bewertung des Skalierungsverhaltens des HiTraP-AT Planungsverfahrens	124
5.2.5. Bewertung des Zeitverhaltens des HiTraP-AT Planungsverfahrens	126
5.2.6. Zusammenfassung	127
5.3. Untersuchung der Eignung des HiTraP-AT Planungsverfahrens für IEC 61131-3 basierte Steuerungen (Anforderung A4)	128
5.4. Evaluation der Anforderungen A1 und A2	130
5.5. Zusammenfassung der Untersuchungsergebnisse	132
5.6. Grenzen und Erweiterungspotentiale des Ansatzes	134
5.7. Bewertung des Ansatzes im Kontext der Anwendungsbeispiele	137
5.7.1. Anwendungsbeispiel 1: Konfigurationsassistenz	137
5.7.2. Anwendungsbeispiel 2: Dynamische Rekonfiguration zur Störkompensation	139
5.7.3. Zusammenfassung	143
6. Zusammenfassung und Ausblick	145
Abbildungsverzeichnis	149
Tabellenverzeichnis	151

Literaturverzeichnis	153
A. Anhang – Beschreibung verwendeter Systemfallstudien	189
A.1. Die Stempelanlage (PPU)	189
A.2. Das Hybride Prozessmodell	192

1. Einleitung

Deutschland wird als eine der führenden Industrienationen bezeichnet. Dies ist nicht zuletzt darauf zurückzuführen, dass 25% des Bruttoinlandsproduktes in Deutschland durch das prozierende Gewerbe erwirtschaftet werden [Sta16b]. Ferner sind im Jahr 2016 mehr als 5,4 Millionen Menschen in diesem Sektor beschäftigt [Sta16c] – dies entspricht einem Anteil der Erwerbstätigen in Deutschland von knapp 12,5% [Sta16a]. Das produzierende Gewerbe stellt demzufolge das Rückgrat der deutschen Wirtschaft dar. Durch die stetig wachsende Dynamik globaler Märkte geraten Unternehmen jedoch zunehmend unter Druck, den Produktionsstandort Deutschland erhalten zu können. Um diesem Trend entgegenzuwirken, eine Abwanderung von Unternehmen in Niedriglohnländer entgegenzuwirken und Deutschland als Leitmarkt und Leitanbieter von intelligenten technischen Systemen zu etablieren, wurde durch die deutsche Bundesregierung die Initiative „Industrie 4.0“ als Teil der Hightech-Strategie der Bundesregierung initiiert [Bun14]. Der Begriff „Industrie 4.0“ bezieht sich dabei auf eine potentielle, vierte industrielle Revolution, welche den technologischen Wandel heutiger Produktionstechnik hin zu cyber-physischen Produktionssystemen [MKB⁺16, LBK15, And15] beschreibt. Cyber-physische Systeme stellen in diesem Kontext die konzeptuelle Grundlage für die Realisierung intelligenter, technischer Systeme dar: vernetzte eingebettete Systeme, welche physische (technische) Prozesse beobachten oder steuern und dabei auf lokale sowie global verfügbare Datenquellen – d.h. andere über das Internet erreichbare cyber-physische Systeme – zurückgreifen [Lee08, RLSS10]. Eine ähnliche Idee und Zielsetzung wie Industrie 4.0 – ebenfalls basierend auf dem Konzept cyber-physischer Systeme – wurde in Nordamerika unter dem Namen „Industrial Internet“ initial durch die Firma General Electric [EA12] gestartet und wird seither ebenfalls aktiv durch Industrie und Forschung dort vorangetrieben [OK15]. Während Industrie 4.0 – entsprechend der Namensgebung – die industrielle Produktion fokussiert, sind die Anwendungsbereiche bei „Industrial Internet“ weiter gefasst und umfassen beispielsweise auch intelligente Stromnetze oder Verkehrsinfrastruktur.

Es existiert eine Vielzahl unterschiedlicher Definitionen, welche die Vision von Industrie 4.0 in Form von Gestaltungsleitsätzen konkretisieren, wie beispielsweise [BTW⁺15, BVZ15, HPO16, HVHB⁺16]. Dabei vereinen die meisten dieser Definition fünf Gestaltungsleitsätze bzw. Funktionsbereiche: (i) Vernetzung und Integration, (ii) Informationstransparenz, Datenerfassung und –verarbeitung, (iii) Assistenzsysteme, (iv) Serviceorientierung und Dezentralisierung sowie (v) Selbstorganisation und Autonomie. Diese werden im Folgenden kurz erläutert.

Vernetzung und Integration: Die (globale) Vernetzung von Maschinen, Geräten, Sensoren und Menschen spielen in der Vision von Industrie 4.0 eine zentrale Rolle. Die vertikale und horizontale Integration wird als ein wesentlicher Hebel für zukünftige (digitale) Wertschöpfungsketten betrachtet [SRHD15, BFKR14]. Aus technischer Sicht spiegelt dies die Kerneigenschaft cyber-physischer Systeme – die globale Vernetzung – wieder; beinhaltet jedoch auch darüber hinausgehende Fragen, wie z.B. der Arbeitsorganisation [Bun15] und Geschäftsmodelle [EDB⁺15].

Informationstransparenz, Datenerfassung und –verarbeitung: Durch die zunehmende Digitalisierung im Kontext von Industrie 4.0 ist eine deutlich steigende Anzahl miteinander verbundener cyber-physischer Systeme zu erwarten [BBH13, OK15]. Hierdurch können Informationen jederzeit (soweit Geräte zu diesem Zeitpunkt mit einander vernetzt sind) ausgetauscht werden; die globale Vernetzung ermöglicht die Verfügbarkeit von Informationen weltweit. Dies ermöglicht eine zunehmende Verschmelzung von realer und virtueller Welt hin zu einer cyber-

1. Einleitung

physischen Welt mit einer neuen Form von Informationstransparenz [Kag15]. So wird durch die Verbindung von (einer großen Menge an) Sensordaten und deren Verknüpfung mit virtuellen (Anlagen-)Modellen die Erstellung präziser, virtueller Abbildungen der realen Welt möglich. Dies ermöglicht wiederum verbesserte oder neue Anwendungen, wie beispielsweise eine verbesserte Assistenz des Menschen, wie im nachfolgenden Abschnitt detaillierter beschrieben.

Assistenzsysteme: Ziel von Assistenzsystemen ist die einfache, schnelle Bereitstellung von Informationen für Beschäftigte, um die Ausführung ihrer Tätigkeit zu unterstützen, zu vereinfachen und sich dabei auf Kernaufgaben zu konzentrieren. Zielstellung von Assistenzsystemen – gemeint ist hier technische Assistenz in der Produktion – ist dabei letztendlich die Rolle des Menschen in der Produktion von der Maschinenbedienung zu einem strategischen Entscheider und flexiblen Problemlöser weiterzuentwickeln [HPO16]. Die Verbesserung der Produktivität, eine Erhöhung der Prozess- und Produktqualität sowie Fehlerreduktion bzw. Fehlervermeidung stellen nur einige der erwarteten Potentiale durch die Anwendung solcher Assistenzsystemen bzw. Assistenzfunktionen dar [GSLZ14, PRK⁺15].

Serviceorientierung und Dezentralisierung: Die Vision und Zielsetzung von Industrie 4.0 sieht einen Wandel von der Produktorientierung hin zu einer Kunden- bzw. Serviceorientierung vor, welcher das gesamte produzierende Unternehmen umfasst: Organisationseinheiten werden wiederum aus Leistungseinheiten aufgebaut, welche ihre Leistungen in Form von Diensten unternehmensintern sowie an Partner innerhalb eines Netzwerkes anbieten [BTW⁺15]. Beispiele für diese Entwicklung sind sogenannte Produkt-Service-Systeme [BLE⁺07], datengetriebene Geschäfts- und Servicemodelle [BCM11] oder aber Produktionsnetzwerke [SF99, JLRM09]. Dieser Ansatz findet sich auch in der Produktion bzw. Produktionstechnik selbst in Form cyberphysischer Systeme wieder: Eine Abkehr von monolithischen, strikt hierarchischen Steuerungsstrukturen hin zu modulareren Systemen, welche analog zu (technischen) Leistungseinheiten – den cyber-physischen Systemen – ihre Dienste/Dienstleistungen anbieten können.

Selbstorganisation und Autonomie: Um die politische Zielsetzung von Industrie 4.0 – die Stärkung des deutschen Standortes für produzierendes Gewerbes – erreichen zu können, wird die Sicherstellung der Anpassbarkeit von Produktionssystemen in der Industrie als ein entscheidender Wettbewerbsfaktor betrachtet [BVZ15, HPE⁺10]. Die Anpassbarkeit umfasst dabei sowohl Wandlungsfähigkeit als auch Rekonfigurierbarkeit. Unter Wandlungsfähigkeit versteht man antizipierte Änderungen innerhalb vorgegebener Grenzen [NHR⁺08]. Hierdurch können kurzfristige Anpassungen an veränderte Bedingungen durchgeführt werden, beschränkt sich jedoch auf einen „vorgedachten Umfang von Merkmalen sowie deren Ausprägung“ [WZT00]. Rekonfiguration ist in diesem Zusammenhang die Eigenschaft eines Produktionssystems, durch Austausch von Funktionseinheiten Änderungen der Rahmenbedingungen zu berücksichtigen [WEN⁺07, MUK00]. Um Anpassungen wirtschaftlich und wettbewerbsfähig durchführen zu können, wird ein möglichst hoher Automatisierungsgrad der Anpassungen angestrebt, um letztendlich auch Anpassungen im laufenden Betrieb – beispielsweise ohne kostspielige Produktionsausfälle durch lange Rüstzeiten – durchführen zu können [Rze03, LKP⁺05, VHDF⁺14]. Einhergehend mit dieser Zielsetzung ist ein Einsatz von Technologien und Prozessen, die eine automatische Datenauswertung ermöglichen und auf deren Ergebnissen die (cyber-physischen) Systeme anschließend selbstständig (autonom) agieren sowie sich unter einander geeignet organisieren können. Die Forderung nach autonomer Entscheidungsfindung technischer Systeme bedingt insbesondere den Einsatz moderner IT Technologien, wie beispielsweise aus dem Bereich der Künstlichen Intelligenz [BBW07].

Die im Zuge von Industrie 4.0 angestrebte Informationstransparenz und präzise virtuelle Abbildungen der Realität ermöglicht in diesem Zusammenhang eine bessere Entscheidungsfindung und letztendlich steigende Gesamtproduktivität [Mal99].

Jedes dieser einzelnen Themenfelder ist bereits seit Jahrzehnten Gegenstand intensiver Forschung. Dementsprechend existieren für eine Vielzahl der darin enthaltenen Aspekte und Technologien bereits vielversprechende Ergebnisse. Der technologische Reifegrad ist dabei sehr unterschiedlich [BTW⁺15]; insbesondere werden teils die spezifischen Anforderungen der Maschinen- und Anlagenautomatisierung, wie z.B. Zuverlässigkeit, Echtzeit, funktionale Sicherheit, etc. noch nicht geeignet adressiert oder zumindest noch nicht ausreichend untersucht [VHDB13, VHH16, LCK16]. Auch setzt die Zielsetzung von Industrie 4.0 eine synergetische Kombination einiger (oder gar aller) dieser Aspekte voraus, was im Kontext von Industrie 4.0 eine zusätzliche Herausforderung darstellt [VHH16, HVHB⁺16, LKR⁺16]. In Zusammenarbeit von Industrie und Forschung wurde daher eine Forschungsroadmap [BVZ15, DDD⁺15, BTW⁺15] erarbeitet, um die Zielsetzung „Industrie 4.0 by Design“ bis 2035 erreichen zu können. Hierbei wurde identifiziert, dass insbesondere auch im Themenbereich „Intelligenz, Flexibilität und Wandelbarkeit“ langfristiger Entwicklungs- und Forschungsbedarf existiert. Hierbei wird erwartet, dass durch die zunehmend stärkere Rolle der Informationstechnologie insbesondere Technologien aus dem Umfeld der Künstlichen Intelligenz zunehmend Bedeutung in der Maschinen- und Anlagenautomatisierung gewinnen. Bischoff et al. [BTW⁺15] sehen hierbei insbesondere im Einsatzbereich der Selbstorganisation und Autonomie von Maschine und Anlagen noch deutliches Forschungspotential. Automatische Planung (engl. *automated planning*) stellt ein solches Forschungsfeld aus dem Bereich der Künstlichen Intelligenz dar und beschäftigt sich im Wesentlichen mit der Frage, wie geeignete Strategien oder Aktionssequenzen automatisch auf Basis eines gegebenen Modells abgeleitet werden können [RN95, GNT04]. Dabei wird insbesondere der problemlösende Charakter, d.h. welche Aktionen (und potentiell in welcher Reihenfolge) genutzt werden sollen, um ein gegebenes Ziel zu erreichen, als grundlegend betrachtet [DK04, BEP⁺13]. Dem entsprechend spielen automatische Planungsverfahren eine bedeutende Rolle, um ein gewisses Maß an Autonomie durch Problemlösungsfähigkeiten zu realisieren und wurden unlängst als unumgängliche Technologie für zukünftige technische Systeme identifiziert [BBB⁺11, BBW07].

In [ABD⁺16] wurden Anwendungsszenarien in Form generischer allgemeiner Beschreibungen von Herausforderungen eines Anwenders beschrieben, wo bei insbesondere der geschäftlichen Rahmen in Form des Wertschöpfungsnetzes ohne Vorgriff auf eine konkrete Lösung beschrieben und auf die Aspekte der Industrie 4.0 Forschungsroadmap [BVZ15] abgebildet. Um die Bedeutung automatischer Planungsverfahren und deren Rolle in unterschiedlichen Anwendungsfeldern zu konkretisieren, welche im Zuge dieser Arbeit näher betrachtet werden sollen, werden im Folgenden zwei exemplarisch Anwendungsbeispiele beschrieben, welche als mögliche konkrete Ausprägungen der Anwendungsszenarien verstanden werden können.

1.1. Anwendungsbeispiele

Zur Beschreibung von Anwendungsfällen (engl. *Use Case*¹) wird in der Software- und Systementwicklung klassisch eine Anwendungsfallbeschreibung eingesetzt [JCJ92, AZ02]. Diese werden

¹Während der Begriff Anwendungsfall häufig in klassischen Vorgehensmodellen genutzt wird, wird in agilen Vorgehensmodellen für grobgranulare Beschreibungen auch der englische Begriff „Epics“ verwendet [Coh04].

1. Einleitung

typischer Weise sehr detailliert und facettenreich beschrieben, um ihrer Rolle als Anforderungsspezifikation bzw. Dokumentation der geforderten Funktionalität gerecht werden zu können. Im Gegensatz dazu werden Anwendungserzählungen (engl. User Story²) eingesetzt, um einzelne Funktionalitäten eines Systems eingängig zu beschreiben und so als Planungsinstrument für die Umsetzung zu nutzen. Anwendungsfallbeschreibungen werden häufig in agilen Vorgehensmodellen eingesetzt [Coh04]. Da an dieser Stelle eine Konkretisierung der Zielsetzung dieser Arbeit erfolgen soll, wird im weiteren Verlauf dieses Abschnitts auf das Mittel der Anwendungserzählung zurückgegriffen und dabei jeweils eine Anwendung aus dem Bereich der Entwicklung sowie dem Betrieb automatisierter Maschinen und Anlagen angeführt wird. Grund hierfür ist die Tatsache, dass die zunehmende Digitalisierung und Vernetzung im Maschinen und Anlagenbau im Zuge der Bestrebungen im Kontext von Industrie 4.0 in ein für den Menschen stetig komplexeres Arbeitsumfeld, sowohl in der Entwicklung von Maschinen und Anlagen, wie auch während deren Betrieb, resultiert. Die Anwendungsbeispiele werden in Form einer Anwendungserzählung eingangs plakativ eingeführt und im Anschluss inhaltlich unter konzeptionellen Aspekten hinsichtlich möglicher Lösungen betrachtet.

1.1.1. Assistenzsystem für die Konfiguration von Steuerungssoftware

Die Komplexität von Maschinen und Anlagen wird durch die zunehmende Digitalisierung und Vernetzung im Zuge der Bestrebungen von Industrie 4.0 nochmals deutlich gesteigert. Nichtsdestotrotz wird auch im Umfeld der vierten industriellen Revolution der Mensch als "Dirigent der Wertschöpfung" [BVZ15] eine wesentliche Rolle spielen. Um das Potential menschlicher Arbeit optimal einsetzen zu können und dabei der steigenden Komplexität geeignet zu entgegnen, werden computergestützte Assistenzsysteme als ein wesentlicher, wirksamer Hebel betrachtet, um Unterstützung in der Entscheidungsfindung zu geben – sowohl im unternehmerischen Bereich als auch in der Entwicklung von Maschinen und Anlagen.

Dementsprechend adressiert der hier betrachtete Anwendungsfall ein Assistenzsystem während der Entwicklung kundenspezifischer Steuerungssoftware. Hierbei wird ferner die Herausforderung zu Grunde gelegt, dass Fabriken auf Grund zunehmender Marktdynamik zusehends schneller an verändernde Begebenheiten und Bedürfnisse angepasst werden müssen.

1.1.1.1. User Story 1: Anlagenkonfigurationsassistenz

Manuel arbeitet als Entwickler bei einem Hersteller für flexible Fertigungsmodule und zeichnet sich für die Bearbeitung von durch Kunden gewünschten Änderungen der Steuerungssoftware verantwortlich. In Abhängigkeit der bei den jeweiligen Kunden im Einsatz befindlichen Fertigungsmodulen des Herstellers und deren physisches (d.h. mechanisches) Layout innerhalb der Fertigungsanlage muss die Steuerungssoftware einzelner oder mehrerer dieser Module von Manuel stets derart angepasst werden, um ein gewünschtes Produkt fertigen zu können.

Hierzu öffnet Manuel das Anlagenkonfigurationsprogramm in dem die kundenspezifischen Informationen hinsichtlich der Anlage verwaltet werden: Die bei den jeweiligen Kunden verfügbaren Module sind dort ebenso graphisch dargestellt wie deren aktuell geplantes Layout. Durch Eingabe der Charakteristika des zu fertigenden Produktes erhält Manuel augenblicklich einen durch den

²Um die Brücke zwischen klassischen Anwendungsfällen und Anwendungserzählungen herzustellen, wird beispielsweise von Jacobsen et al. der Begriff "Use Cases Slices" verwendet [JSB11].

Anlagenkonfigurator automatisch bestimmten Vorschlag des zu realisierenden Steuerungsablaufes. Dieser wird nun durch Manuel manuell geprüft. Erweist sich der vorgeschlagene Steuerungsablauf als geeignet, kann Manuel per Knopfdruck automatisiert die notwendige, kundenindividuelle Steuerungssoftware generieren.

1.1.1.2. Konzeptionelle Betrachtung

Die Grundfunktionalität des in zuvor angeführter Anwendungserzählung beschriebenen Anlagenkonfigurators stellt die automatische Generierung der Steuerungssoftware dar. Dieser Forschungsbereich ist bereits intensiver Gegenstand der Forschung. Gerade im Bereich der sogenannten modellgetriebenen Software- und Systementwicklung spielt die Generierung von Quellcode für (gegebenenfalls unterschiedliche Zielplattformen) auf Basis gegebener Modelle eine wesentliche Rolle [SSV08, VHDF⁺14]. Aber auch in kommerziell bereits erhältlichen Werkzeugen für die Maschinen- und Anlagenautomatisierung ist die Funktionalität der Generierung von (Steuerungs-)Software auf Basis von Modellen bereits enthalten. Betrachtet man die Softwarekomponente zur Generierung von Software als geschlossenes Softwaresystem unter Vernachlässigung des inneren Aufbaus, unterscheiden sich die verschiedenen Ansätze insbesondere hinsichtlich der Art – d.h. welche Aspekte werden durch die Modelle abgedeckt, wie z.B. Komponenten, Abläufe, Abhängigkeiten, etc. – und Detailtiefe – d.h. wie detailliert werden unterschiedliche Aspekte berücksichtigt – der Eingabemodelle sowie der entsprechend dem Informationsgehalt der Modelle generierten Software. So existieren beispielsweise Ansätze zur Generierung von Steuerungssoftware auf Basis von UML Zustandsdiagrammen [Wit12] und der automatischen Generierung von Reglern auf Basis von Matlab/Simulink Modellen [BMUVH12] bis hin zur Generierung ganzer Automatisierungsprojekte samt Konfiguration der Hardware [EMO07]. Auf Basis der Anwendungserzählung ist diesbezüglich eine Konkretisierung möglich: Manuel konfiguriert kundenspezifische Lösungen auf Basis des mechanischen Layouts abgeschlossener Module. Das Verhalten der Module soll ferner entsprechend den gewünschten Charakteristika des zu fertigenden Produktes konfiguriert werden. Dementsprechend stellen Module einen fixen Konfigurationsraum bereit, wobei die Auswahl der geeigneten Konfiguration in Abhängigkeit des Layouts und Produktcharakteristika erfolgt. Neben der Generierung der Software stellt der Mechanismus zur Bestimmung von Vorschlägen für notwendige Steuerungsabläufe eine wesentliche Funktionalität des in dieser Anwendungserzählung beschriebenen Assistenzsystems zur kundenindividuellen Konfiguration der Steuerungssoftware modularer Anlagen dar. Abbildung 1.1 stellt hierzu ein mögliches Lösungskonzept für den Anlagenkonfigurator schematisch dar. Dieses enthält im Kern eine Komponente, welche automatisch mögliche Steuerungsabläufe identifiziert und den vermeintlich besten Ablauf selektiert um diesen als Vorschlag dem Entwickler grafisch darzustellen. Hierzu greift diese Kernkomponente auf Entwicklungsdaten zurück, welche steuerungstechnische Möglichkeiten und Einschränkungen einzelner Module ebenso beinhalten wie das mechanische Anlagenlayout und die Produktanforderungen. Ein durch den Entwickler ausgewählter Ablauf wird dann an den Code Generator übergeben.

1.1.2. Dynamische Rekonfiguration der Anlage im Falle technischer Störungen

Alarm- und Störungsmanagement stellt eine wesentliche Funktionalität im Maschinen- und Anlagenbau dar, da Störungen innerhalb einer Fertigungsanlage häufig auftreten und bei ungeeig-

1. Einleitung

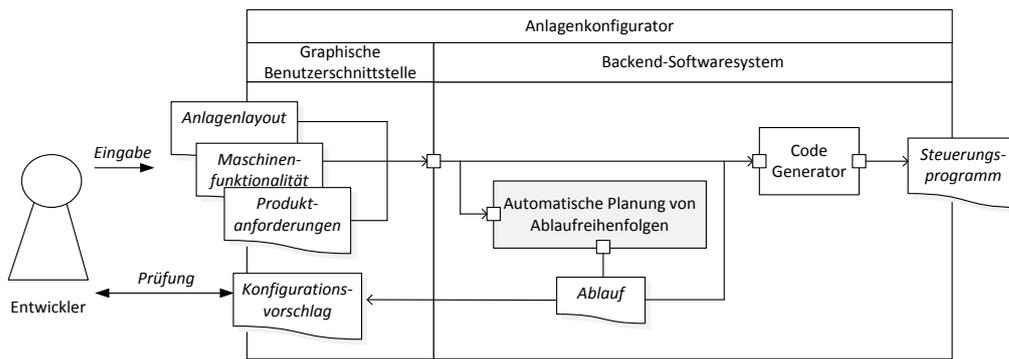


Abbildung 1.1.: Konzeptionelle Übersicht eines Anlagenkonfigurators.

meter Behandlung – z.B. der Eingriff durch den Bediener erfolgt nicht schnell genug oder die durchgeführten Maßnahmen sind nicht zielführend – zu hohen Kosten durch Materialausschuss oder gar Stillstand der Anlage führen. Wie aktuelle Forschungsergebnisse zeigen, stellt bereits die Identifikation notwendiger Eingriffe sowie die Identifikation der Fehlerursache verantwortliches Bedienpersonal auf Grund der großen Menge an Daten vor große Herausforderungen [VHSF15]. Entsprechend stellt die Identifikation geeigneter Maßnahmen eine mindestens ebenso komplexe Herausforderung dar. Ein Teilbereich möglicher Maßnahmen befasst sich mit der Anpassung des Steuerungsverhaltens, um einer auftretenden Störung entgegenzuwirken. Dies wird in der hier beschriebenen, exemplarischen Anwendung betrachtet.

1.1.2.1. User Story 2: Assistenz zur dynamischen Störungskompensation durch Rekonfiguration des Steuerungsverhaltens

Willi Werker zeichnet sich für den reibungslosen Betrieb einer Fertigungsanlage verantwortlich. Plötzlich erscheint auf Willis Display eine rote Warnung: "Technischer Fehler erkannt und System stabilisiert. Manuelle Prüfung notwendig.". Dank der Stabilisation des Systems kann Willi die aktuelle Situation geeignet prüfen, denn die Anlage wartet aktuell auf seine Rückmeldung. Willi analysiert nun innerhalb des Betriebs- und Störungssystems (BSS) den aktuellen Zustand der Anlage sowie den identifizierten Fehler: Eine Weiche des Transportsystems lässt sich offenbar nicht mehr in die notwendige Position bewegen. Dieser mechanische Fehler lässt sich meist nicht schnell beheben. Daher fragt Willi das BSS nach einer möglichen Rekonfiguration, um den Betrieb trotz des aktuellen Fehlers aufrecht erhalten zu können. Das BSS visualisiert Willi nun einen alternativen Ablauf, um das aktuell gewünschte Produkt zu fertigen, welcher ohne die verklemmte Weiche durchgeführt werden kann. Obwohl, wie angezeigt, der Durchsatz dieses Ablaufs deutlich geringer ist, entscheidet Willi sich für eine dynamische Rekonfiguration des Ablaufs. Schließlich ist bereits in ein paar Stunden eine geplante Wartungspause. Die Weiche wäre bis zu diesem Zeitpunkt wahrscheinlich ohnehin nicht repariert. Willi entnimmt die aktuell in Produktion befindlichen Werkstücke und teilt dem BSS seine Entscheidung für die störungsbedingte Rekonfiguration mit. Sofort beginnt die Fertigungsanlage mit dem Wiederanlauf und der Fertigung entsprechend dem durch das System zuvor bestimmten und vorgeschlagenen Ablauf.

1.1.2.2. Technologische Betrachtung

Eine, wie in zuvor beschriebenem Anwendungsbeispiel 1 notwendige Anpassung des Fertigungssystems im Fehlerfall kann auf unterschiedliche Arten realisiert werden. So ist beispielsweise eine interdisziplinäre Lösung denkbar: ein fehlerhaftes Fertigungsmodul bestehend aus Mechanik, Elektrik/Elektronik und ggf. Steuerungssoftware wird mittels einer automatischen, mechanischen Lösung durch ein identisches, redundantes Modul ersetzt. In der vorangegangenen Anwendungserzählung wird ausschließlich die Anpassung des Steuerungsverhaltens, d.h. der Software, betrachtet. Auch hier lassen sich unterschiedliche, mögliche Ausprägungen der Anpassung der Steuerungssoftware unterscheiden. Die Bandbreite reicht hierbei von Lösungen, in denen vollständig neuer Steuerungscode generiert und auf den Steuerungen aufgespielt wird, bis hin zu einer reinen Anpassung von Parametern des Steuerungsprogramms. Eine Änderung des Steuerungsprogramms selbst erfordert hohen konzeptionellen Aufwand und ist nicht ohne weiteres möglich. Gründe hierfür sind vielfältig. Beispielsweise unterstützen Steuerungen für die Maschinen- und Anlagenautomatisierung häufig keine dynamische Speicherverwaltung, um Determinismus hinsichtlich der Ausführung eines Steuerungsprogramms bereits a priori sicherstellen zu können. Ferner muss im Falle einer Änderung des Programms sichergestellt werden, dass kein Schaden an Maschine, Mensch oder Umwelt verursacht wird. Auch nationale Regularien und notwendige Zertifizierungen, z.B. hinsichtlich Betriebs- und Ausführungssicherheit, sind für Mechanismen mit beliebiger Veränderung des Steuerungsprogramms sehr komplex oder teils nicht möglich. Nichtsdestotrotz existieren bereits einige Forschungsansätze, welche diese Herausforderungen (zumindest teilweise) adressieren. Eine Änderung der Steuerungssoftware ist insbesondere dann notwendig, wenn sich durch Änderung physikalischer Komponenten (d.h. mechanische oder elektrische/elektronische Komponenten, Automatisierungsgeräte, etc.) aus Steuerungssicht eine Veränderung der (möglichen) Funktionalität ergibt. So kann beispielsweise durch das Hinzufügen von Komponenten eine Erweiterung der Funktionalität erfolgen. Aber auch eine Verbesserung oder Fehlerbehebung der Steuerungssoftware wäre ein denkbarer Grund für eine Aktualisierung der Steuerungssoftware³. Im Rahmen der zuvor beschriebenen Anwendungserzählung erfolgt jedoch weder eine Erweiterung der Funktionalität noch wird der Fall einer Softwareaktualisierung betrachtet. Eine etwaige Störung verursacht dabei eher eine funktionale Einschränkung der Maschine bzw. Anlage. Dementsprechend ist für den hier vorliegenden Fall eine Anpassung des Steuerungsverhaltens durch Parameterierung ohne Veränderung des Programmcodes ausreichend.

Im Kontext der Bestrebungen von Industrie 4.0 hin zu durchgängig digitalisierten Produktionssystemen werden selbstorganisierende, autonome Anlagen als eine der Kernideen genannt. Analog zu Diskussionen im Bereich autonomer Fahrzeuge, bei denen ein menschlicher Fahrer aktuell nach wie vor anwesend sein muss, um im Bedarfsfall während einer autonomen Fahrt eingreifen zu können, ist eine automatische Rekonfiguration ohne Prüfung durch den Mensch für den Betrieb von Maschinen und Anlagen aktuell noch nicht sinnvoll. Die Gründe hierfür sind vielseitig. So stehen erfahrene Anlagenbetreiber einer vollständig automatisierten Rekonfiguration teils aus emotionalen Gründen noch kritisch gegenüber. Auch stellen die Überwachung und der Betrieb sich stetig selbst anpassender Maschinen- und Anlagen eine große Herausforderung

³In Abhängigkeit des technischen Systems sowie des realisierten technischen Prozesses kann es auch notwendig sein, Aktualisierung der Steuerungssoftware zur Verbesserung oder Fehlerbehebung im laufenden Betrieb vorzunehmen.

1. Einleitung

dar, für die noch kaum Erfahrungswerte vorhanden sind. Aber auch aus rechtlicher Sicht existieren aktuell (noch) Hürden für eine vollautomatische Rekonfiguration. So sind beispielsweise Haftungsfragen für Schäden durch fehlerhaftes (automatisches) Verhalten noch nicht gänzlich geklärt. In diesem Zusammenhang steht auch die rechtliche Frage offen, was in diesem Kontext fehlerhaftes Verhalten charakterisiert, für welches ggf. gehaftet werden müsste. Ferner existiert noch Forschungsbedarf, um automatisch rekonfigurierbare Systeme in der industriellen Praxis flächendeckend einsetzen zu können [BTW⁺15].

In der hier betrachteten Anwendungserzählung soll jedoch – im Gegensatz zu der Anwendungserzählung aus Abschnitt 1.1.1 – nicht die Assistenzfunktion im Vordergrund stehen, sondern vielmehr die Anpassung des Steuerungsverhaltens zur Störkompensation selbst. Nichtsdestotrotz muss für den Benutzer – hier am Beispiel von Willi Werker – geeignet die geplante Rekonfiguration (graphisch) dargestellt werden, um über die Korrektheit der Anpassung zu entscheiden. Abbildung 1.2 stellt einen möglichen, konzeptionellen Lösungsansatz für diese Anwendungsfallbeschreibung und die Realisierung des BSS dar. Das BSS besteht dabei im Wesentlichen aus drei Softwaresubsystemen: dem Softwaresystem zur dynamischen Rekonfiguration (auch als Backend-System bezeichnet), einem Alarmmanagement- und Diagnosesystem sowie Software für das Rekonfigurationsmanagement auf der Anlagensteuerung. Informationen über Störungen im Betrieb werden durch Alarmmanagement und Diagnosesysteme in Form einer Fehlerinformation an das Backend-System übergeben. Es existieren bereits eine Vielzahl von Ansätzen für Alarmmanagement und Diagnose. Dieses Subsystem wird im Folgenden daher nicht näher betrachtet. Kernkomponente stellt das Backend-System dar, welches mit Hilfe einer automatischen Planung von Ablaufreihenfolgen einen neuen Ablauf der Steuerungsfunktionen identifiziert. Auf Basis der Fehlerinformationen wird das Planungsmodell geeignet angepasst. Somit weisen die Planungskomponenten in beiden Anwendungsbeispielen eine identische Schnittstellen auf. Der identifizierte Ablauf wird einerseits zur manuellen Verifikation an die graphische Benutzerschnittstelle weitergegeben und, im Falle einer Freigabe durch den Benutzer, für die Paramtrierung der Steuerungssoftware verwendet. Diese den neuen Ablauf beschreibenden Parameter werden durch einen Teil der Steuerungssoftware, dem Rekonfigurationsmanagement geeignet übernommen und ausgeführt. Damit realisiert diese Komponente den Wiederanlauf im Fehlerfall auf Basis des durch die Planungskomponente identifizierten, alternativen Ablaufs.

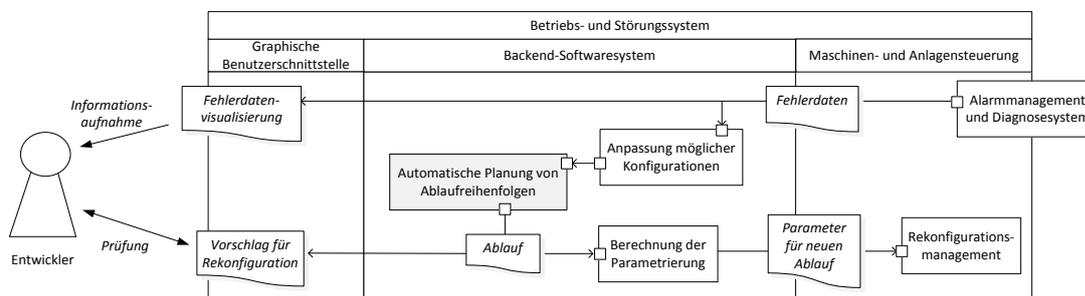


Abbildung 1.2.: Konzeptionelle Übersicht eines Assistenzsystems für die Rekonfiguration des Steuerungsverhaltens im Fehlerfall.

1.2. Zielsetzung der Dissertation

Die beiden in den vorangegangenen Abschnitten beschriebenen Anwendungsbeispiele konkretisieren exemplarisch die im Zuge der Bestrebungen von Industrie 4.0 formulierten Anwendungsszenarien und spiegeln dementsprechend beispielhaft die Zielsetzung von Industrie 4.0 wider. Dabei beziehen sich die Anwendungsbeispiele im Wesentlichen auf eine Kombination von eingangs in dieser Arbeit motivierten Themenfelder „Assistenzsysteme“ sowie „Selbstkonfiguration und Autonomie“ wodurch letztendlich die für Industrie 4.0 typische Herausforderung einer Kombination von Themenfeldern reflektiert wird. Im Zuge der technologischen Betrachtung der Anwendungsbeispiele wurde aufgezeigt, dass eine automatische Planung von Ablaufreihenfolgen einen wesentlichen Baustein für die Realisierung beider Anwendungsbeispiele darstellt.

Existierende Planungsverfahren adressieren aktuell nicht ausreichend die Anforderungen der Steuerungsebene für die Maschinen- und Anlagenautomatisierung an ein automatisches Planungsverfahren. Smith et al. [SFJ00] führen dies unter anderem darauf zurück, dass praktische Probleme technischer Systeme meist nicht vollständig den Annahmen unterliegen, die klassische Planungsverfahren zu Grunde legen und schlussfolgert, dass für spezifische Systeme und Zielsetzungen spezifisch Planungskonzepte notwendig sind. Dieser Argumentation und Herausforderung folgend soll im Rahmen dieser Dissertation die Anwendbarkeit automatischer Planungsverfahren für die Steuerungsebene des Maschinen- und Anlagenbaus untersucht und ein geeignetes Planungsverfahren zur automatischen Bestimmung von Ausführungsreihenfolgen von Softwarefunktionen der Steuerungsebene für die Maschinen- und Anlagenautomatisierung (im Folgenden auch als Automatisierungsfunktionen bezeichnet) entwickelt werden.

1.3. Aufbau der Dissertation

Entsprechend der Zielsetzung dieser Dissertation ist diese Arbeit in insgesamt sechs Kapitel gegliedert. Eine grafische Darstellung der Kapitelstruktur ist in Abbildung 1.3 gegeben.

Anschließend an dieses Kapitel werden in Abschnitt 2.1 die wesentlichen Merkmale und Charakteristika automatischer Planungsverfahren sowie der Steuerung industrieller Produktionssysteme im Allgemeinen sowie des Maschinen- und Anlagenbaus im Speziellen beschrieben. Hierdurch wird die Domäne der Automatisierung im Maschinen- und Anlagenbau und damit der in dieser Arbeit adressierte Betrachtungsbereich konkretisiert und charakterisiert. Basierend hierauf werden in Abschnitt 2.2 Anforderungen an einen geeigneten Verfahren zur Bestimmung der Ablaufreihenfolgen von Automatisierungsfunktionen detailliert abgeleitet und beschrieben. Dabei werden einerseits Anforderungen identifiziert, die sich spezifisch auf die Funktionsweise des in dieser Arbeit vorgeschlagenen Planungsverfahrens beziehen (Anforderungen A1 bis A3) sowie andererseits Anforderungen hinsichtlich der Integration des Planungsverfahrens in die Domäne des Maschinen- und Anlagenbaus, d.h. in die Entwicklung von Maschinen- und Anlagen wie dies beispielsweise in Anwendungsbeispiel 1 der Fall ist (Anforderung A5) sowie im Betrieb von Anlagen (Anforderung A4) wie in Anwendungsbeispiel 2 beschrieben.

Im weiteren Verlauf der Arbeit werden in Kapitel 3 eingangs unterschiedliche Formalismen zur Beschreibung von Planungsproblemen sowie darauf aufbauende, theoretische Lösungsansätze beschrieben. Daran anschließend werden verwandte Ansätze und Anwendungen automatischer Planungsverfahren untersucht und den zuvor gestellten Anforderungen gegenübergestellt.

1. Einleitung

Kapitel 4 umfasst die Beschreibung aller konzeptuellen Bausteine, um die gestellten Anforderungen an ein geeignetes Planungsverfahren für die Bestimmung von Ablaufreihenfolgen von Automatisierungsfunktionen zu adressieren. Hierbei wird in Abschnitt 4.1 der in dieser Dissertation vorgeschlagene neue Ansatz zur automatischen Handlungsplanung für die Steuerungsebene des Maschinen- und Anlagenbaus beschrieben: das Planungsverfahren HiTraP-AT. Dieser auf einem verknüpften, hierarchischen Transitionssystem basierende Ansatz ermöglicht die Ableitung partiell geordneter Ablaufreihenfolgen von Automatisierungsfunktionen. Entsprechend der zuvor bereits beschriebenen Anforderungsgruppierung aus Abschnitt 2.2 umfasst der HiTraP-AT Ansatz aufbauend auf dem zuvor bereits eingeführten Planungsverfahren darüber hinaus ein Konzept für die Integration des Planungsverfahrens in existierende Entwicklungsprozesse (Anforderung A5) und ein Konzept für die Kopplung und Anwendung des Verfahrens für die Software der Steuerungsebene des Maschinen- und Anlagenbaus (Anforderung A4). Diese Aspekte werden in den Abschnitten 4.2 und 4.3 betrachtet.

Kapitel 5 umfasst die Untersuchung der Anwendbarkeit und Evaluation des in dieser Arbeit vorgeschlagenen HiTraP-AT Ansatzes. Dabei orientiert sich die Struktur dieses Kapitels stark an den gestellten Anforderungen. Durchgeführte Untersuchungen, deren Ergebnisse sowie ihre Bewertung hinsichtlich der Erfüllung gestellter Anforderungen ist jeweils in den entsprechenden Abschnitten 5.1 bis 5.4 beschrieben. Aufbauend darauf werden die Untersuchungsergebnisse und identifizierte Grenzen des Ansatzes im Hinblick auf die gestellten Anforderungen in Abschnitt 5.5 zusammengefasst sowie darüber hinausgehende, im Zuge der Evaluation identifizierte Grenzen des Ansatzes in Abschnitt 5.6 beschrieben. In Abschnitt 5.7 erfolgt eine Bewertung des Ansatzes im Hinblick auf die in diesem Kapitel eingangs beschriebenen Anwendungsbeispiele.

Den Abschluss dieser Dissertation bildet eine Zusammenfassung in Kapitel 6, welche die zentralen Aspekte des im Rahmen dieser Dissertation entwickelten Planungsverfahrens beschreibt. Ferner wird ein Ausblick auf weiterführende Forschungsthemen und Anwendungsfelder des in dieser Arbeit beschriebenen Konzeptes für die Bestimmung der Ablaufreihenfolgen von Automatisierungsfunktionen gegeben.

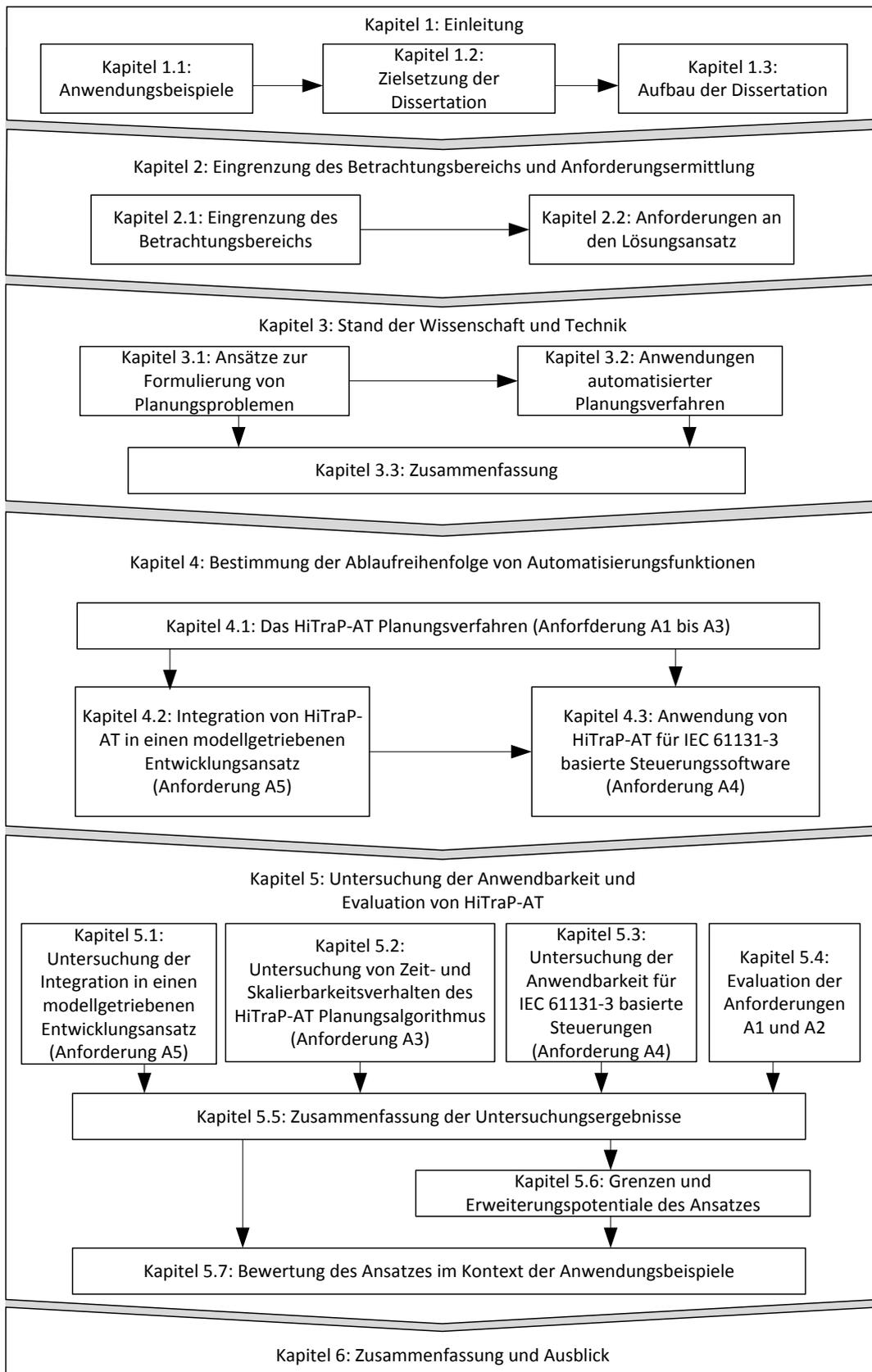


Abbildung 1.3.: Grafischer Überblick über den Aufbau dieser Dissertation.

2. Eingrenzung des Betrachtungsbereichs und Anforderungsermittlung

Im Rahmen dieses Kapitels werden notwendige Begriffsdefinitionen im Kontext der automatischen Planung und Automatisierung auf der Steuerungsebene des Maschinen- und Anlagenbaus angegeben. Ferner werden auf Basis der gegebenen terminologischen Einführung der Forschungsgegenstand dieser Arbeit – ein neues Verfahren zur automatischen Handlungsplanung – und dessen Eigenschaften präzise charakterisiert, um letztendlich im weiteren Verlauf dieses Abschnitts konkrete Anforderungen an den Lösungsansatz formulieren zu können.

2.1. Eingrenzung des Betrachtungsbereichs

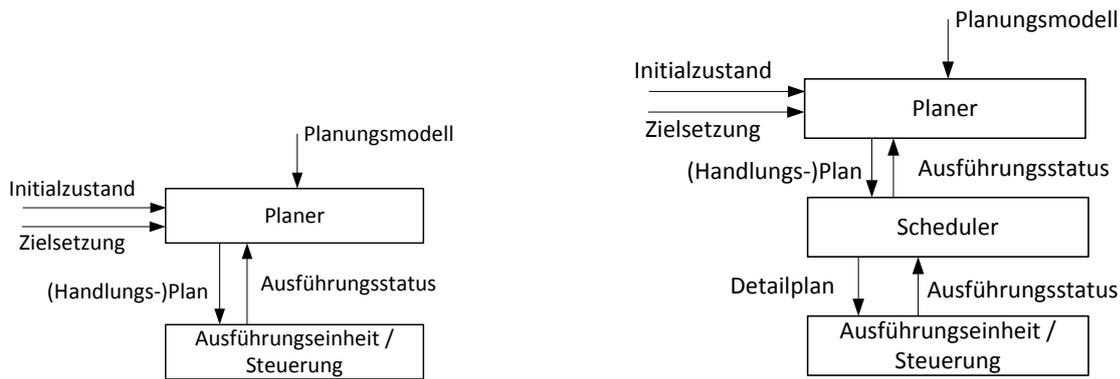
2.1.1. Automatische Planungsverfahren

Automatische Planungsverfahren stellen eines der grundlegenden Fundamente der Forschungen im Bereich der Künstlichen Intelligenz dar und beschäftigen sich, ganz allgemein beschrieben, mit Algorithmen zur Bestimmung einer Strategie, um ein spezifisches Ziel in einem betrachteten System zu erreichen. Unter einer Strategie (oder auch Handlungsplan) ist in diesem Kontext eine durch einen Algorithmus bestimmte Reihenfolge von durch das System ausführbaren Aktionen zu verstehen [GNT04]. Automatische Planungsverfahren stellen damit die algorithmische Grundlage für die Entscheidungsfindung autonomer, intelligenter System [RN95] dar. Damit bilden automatische Planungsverfahren, allgemein betrachtet, ebenfalls die Grundlage intelligenter, technischer System [BBW07, BAB⁺03].

Abbildung 2.1(a) stellt das konzeptuelle Modell dar, welches der Theorie automatischer Planungsverfahren zu Grunde liegt. Das die Planung ausführende Softwaresystem – die Implementierung des Planungsalgorithmus sowie entsprechender Schnittstellen – wird als *Planer* bezeichnet. Ein *Planungsmodell* beschreibt dabei das betrachtete System (in der dieser Arbeit zu Grunde gelegten klassischen Planung) als diskretes Zustandsübergangssystem $\Sigma = \langle S, A, \gamma \rangle$ mit der Menge der Zustände S , der Menge der Zustandsübergänge A – auch als Aktionen bezeichnet – sowie einer Zustandsübergangsfunktion $\gamma : S \times A \rightarrow S$. Der *Initialzustand* reflektiert die diskrete, zustandsorientierte Betrachtungsweise der klassischen Planung und stellt den Ausgangspunkt hinsichtlich des System dar. Die *Zielsetzung* stellt das zu erfüllende Ziel dar, welches am Ende einer Ausführung eines gegebenen Handlungsplans erreicht werden soll. Die Zielsetzung nimmt dabei ebenfalls Bezug auf das Planungsmodell. Planungsmodell, Initialzustand und Zielsetzung gemeinsam werden auch als *Planungsproblem* bezeichnet. Ziel eines Planers ist auf Basis eines gegebenen Planungsmodells einen (Handlungs-)Plan⁴ zu generieren, um durch ein im Initialzustand befindliches System mit Hilfe der im Planungsmodell beschriebenen Möglichkeiten die Zielsetzung zu erreichen. Ein Handlungsraum im Kontext automatischer Planungsverfahren stellt dabei die Menge aller durch das Planungsmodell definierter Handlungspläne dar. Ein Handlungsraum kann dabei, je nach Ansatz, auch Handlungspläne enthalten, welche nicht das Planungsproblem erfüllen. In diesem Zusammenhang wird auch Teils von Zustandsräumen oder Zustandsgraphen

⁴Um den durch einen Planer erstellten Plan von einem durch einen Scheduler genierten Plan abzugrenzen, wird im Kontext automatischer Planungsverfahren auch von einem Handlungsplan gesprochen im Gegensatz zu einem Detailplan, der durch einen Scheduler bestimmt wird.

2. Eingrenzung des Betrachtungsbereichs und Anforderungsermittlung



(a) Konzeptuelles Modell automatischer Planungsverfahren.

(b) Integration des Schedulers in das konzeptuelle Modell automatischer Planungsverfahren.

Abbildung 2.1.: Konzeptuelles Modell der automatischen Planung (in Anlehnung an [GNT04]).

gesprochen, da ein Handlungsraum die Menge erreichbarer Zustände und deren Zustandsübergänge (Aktionen) enthält. Im Verlaufe dieser Arbeit werden diese Begriffe synonym verwendet.

Auf Grund der zu Grunde gelegten Formulierung des Planungsproblems auf Basis von Zielzuständen wird in diesem Zusammenhang auch oft von zielbasierten (Planungs-)Verfahren (engl. goal based planning) gesprochen. Der Handlungsplan wird im Anschluss an eine durchgeführte Planung an eine *Ausführungseinheit* – auch als Steuerung bezeichnet – zur Ausführung übergeben, um letztendlich die Zielsetzung durch das System tatsächlich zu erreichen. In einigen Planungskonzepten, insbesondere solchen, die eine erneute Planung oder Umplanung berücksichtigen, um beispielsweise unerwartete oder veränderte Situationen während der Ausführung geeignet handhaben zu können, wird zusätzlich eine informationelle Rückkopplung von der Ausführungseinheit zum Planer auf Basis des aktuellen *Ausführungsstatus* integriert. Die Formulierung des Planungsproblems solcher Neuplanungssysteme (engl. replanning) kann sich jedoch deutlich von der Formulierung des Planungsproblems unterscheiden, insbesondere wenn zur Verbesserung des Verfahrens beispielsweise auch Informationen aus einer vorangegangenen Planung Berücksichtigung finden. Auch werden in diesem Fall Konzepte zur Beurteilung von Plänen – z.B. Durchführbarkeit des aktuellen Plans - benötigt. Diese Aspekte sind nicht Betrachtungsgegenstand dieser Arbeit.

Im Gegensatz zu automatischen Planungsverfahren adressieren Schedulingverfahren die Frage, wann und wie gegebene Aktionen ausgeführt werden und berücksichtigen dabei insbesondere zeitliche Aspekte und Ressourceneinschränkungen. Daher sind Schedulingverfahren beispielsweise im Bereich von Betriebssysteme allgegenwärtig. Die Fragestellungen aus dem Bereich des Scheduling, d.h. insbesondere quantitative zeitliche Aspekte und etwaige Ressourceneinschränkungen, werden im Rahmen dieser Dissertation nicht betrachtet. Eine tiefgreifende Diskussion und weiterführende Charakterisierungen von Planungs- und Schedulingverfahren können beispielsweise [DK04, SFJ00] entnommen werden.

Abbildung 2.1(b) stellt die Integration eines Schedulers im konzeptuellen Modell der automatisierten Planung dar: Ein *Scheduler* übernimmt die Aufgabe einer detaillierteren Planung unter

Berücksichtigung weiterer Bedingungen auf Basis des durch einen Planer bestimmten Handlungsplans. Der durch ein Scheduling resultierende *Detailplan*, welcher im Gegensatz zu einem Handlungsplan detailliertere Informationen hinsichtlich der Ausführung enthält, wird nun der Ausführungseinheit übergeben. Die Bedeutung des *Ausführungsstatus* ist dabei analog zu den vorherigen Ausführungen. Ergänzend existieren durch die Integration des Schedulers in das konzeptionelle Modell jedoch nicht nur Informationen über den Ausführungsstatus der Ausführungseinheit sondern auch des Schedulers selbst. Eine notwendige Neuplanung kann dabei (ergänzend zu den vorherigen Beschreibung des konzeptionellen Modelles ohne Scheduler) auch durch einen Scheduler bedingt sein, da beispielsweise für einen gegebenen Handlungsplan auf Basis der durch den Scheduler zu berücksichtigenden Bedingungen – insbesondere hinsichtlich Zeit und Ressource – kein valider Schedule gefunden wurde.

Typischer Weise werden Planungsverfahren hinsichtlich ihrer Anwendungsdomäne – auch als *Planungsdomäne* bezeichnet – klassifiziert [GNT04]. Als Domäne wird in diesem Zusammenhang das System und dessen Umfeld betrachtet. *Domänenunabhängige Planer*, die theoretisch in beliebigen Planungsdomänen eingesetzt werden können, führen eine Planung mittels allgemeiner Planungsalgorithmen durch, ohne dabei auf domänenspezifisches Wissens zurückgegriffen und zu nutzen. *Domänenspezifische Planer* hingegen adressieren die Herausforderungen einer spezifischen Anwendungsdomäne oder spezifischer Anwendungsfälle, wie z.B. für den Mars Rover [SMC⁺14, BJMR05] oder Computerspiele [KBK07, SNT98]. In diesem Kontext stellt auch der Maschinen- und Anlagenbau eine Domäne dar, welche im Kontext dieser Arbeit fokussiert wird. Typischer Weise sind für eine gegebene Domäne operierende domänenunabhängige Planer schlechter als spezialisierte, domänenspezifische Planer. Die Entwicklung domänenspezifischer Planer ist jedoch komplex. *Konfigurierbare Planer* adressieren diesen Nachteil: hierbei werden domänenunabhängige Planungskonzepte und Algorithmen angewandt, jedoch durch zusätzliche, domänenspezifische Informationen konfiguriert, um ein domänenspezifisches Planungsproblem geeignet lösen zu können.

Im Rahmen dieser Dissertation werden sowohl Aspekte domänenspezifischer Planung für die Domäne der Steuerungsebene des Maschinen- und Anlagenbaus als auch Aspekte domänenunabhängiger Planung im Hinblick auf die Formulierung und Lösung von Planungsproblemen betrachtet. Daher kann das in dieser Arbeit beschriebene Planungsverfahren unter Bezugnahme auf zuvor eingeführte Klassifikation als konfigurierbares Planungsverfahren betrachtet werden.

Neben der zuvor beschriebenen Charakterisierung automatischer Planungsverfahren auf Basis der Planungsdomäne stellen die Eigenschaften des erzeugten Plans eine weitere Klassifikationsmöglichkeit automatischer Planungsverfahren dar. Handlungspläne sind, wie oben beschrieben, eine Menge von Aktionen, die zur Erreichung einer Zielsetzung führen. Im Gegensatz zum Schedulingverfahren konzentrieren sich automatische Planungsverfahren, wie ebenfalls bereits beschrieben, auf die Auswahl der zielführenden Aktionen sowie deren qualitative zeitliche Ordnung durch Bestimmung notwendiger Reihenfolgen der Aktionen, d.h. die mathematische Ordnungsrelation zwischen Aktionen. Entsprechend stellt die durch den Planer bestimmte Art der Ordnungsrelation ein Klassifikationsmerkmal automatischer Planungsverfahren dar. Hierbei wird insbesondere zwischen einem *total geordneten Handlungsplan* und einem *partiell geordneten Handlungsplan* unterschieden [Wei94]. Ein total geordneter Handlungsplan entspricht dabei einer Sequenz von Aktionen und der traditionellen Zielsetzung klassischer Planungsverfahren. Die kausale Ordnung von Aktionen ist innerhalb eines total geordneten Plans eindeutig bestimmt. Für jedes Paar

2. Eingrenzung des Betrachtungsbereichs und Anforderungsermittlung

von Aktionen eines Handlungsplans kann stets eine Aussage hinsichtlich ihrer Vorgänger- bzw. Nachfolgerbeziehung (d.h. sequentielle Reihenfolge) zueinander getroffen werden. Im Gegensatz dazu existieren innerhalb eines partiell geordneten Handlungsplans (engl. partial order plan) zwar ebenfalls kausale Abhängigkeiten zwischen Aktionen des Plans hinsichtlich ihrer Vorgänger-/Nachfolgerbeziehung, es kann jedoch nicht für jedes Paar von Aktionen eine Aussage getroffen werden. Zwei Aktionen, über die keine Aussage hinsichtlich ihre sequentiellen Reihenfolge getroffen werden kann, können als parallel ausführbare Aktionen betrachtet werden. Ein partiell geordneter Plan beschreibt dabei eine Menge total geordneter Pläne. Daher kann eine Planung mit partieller Ordnung als Verallgemeinerung der klassischen Planung zur Bestimmung eines total geordneten Plans betrachtet werden [NK01, RHN04]. Sind zwei Aktionen, für die keine sequentielle Ordnung existiert, unabhängig voneinander ausführbar, können diese auch parallel ausgeführt werden. In diesem Fall können aus einem partiell geordneten Handlungsplan auch sogenannte parallele Pläne (engl. parallel plans) erzeugt werden [Kno94]. Allgemein spricht man bei Planungsverfahren, welche einen partiell geordneten Handlungsplan erzeugen von Planung mit partieller Ordnung – oder im englischen Sprachraum von partial order planning (POP).

Im Rahmen dieser Dissertation werden initial keine Anforderung hinsichtlich etwaiger Ordnungsrelationen des zu bestimmenden Handlungsplans gestellt. Das in dieser Arbeit vorgeschlagene Planungsverfahren HiTraP-AT kann letztendlich als Planung mit partieller Ordnung und parallelen Plänen klassifiziert werden.

2.1.2. Automatisierung industrieller Fertigungssysteme

Unter Automatisierung versteht man das Äusrüsten einer Einrichtung, so dass sie ganz oder teilweise ohne Mitwirkung des Menschen bestimmungsgemäß arbeitet"[DIN98]. Diese Zielsetzung wird bereits seit der Antike verfolgt, wobei insbesondere durch die Elektrifizierung im 19.Jahrhundert die Automatisierungstechnik als eigenständiger Technologie- und Forschungsbereich an Bedeutung gewinnt. Die *Automatisierungstechnik* stellt dabei ein interdisziplinäres Betrachtungsfeld dar, welches sich mit der Automatisierung und Steuerung technischer Prozesse befasst [Lun03].

Allgemein wird unter einem *Prozess*, die "Gesamtheit von aufeinander einwirkender Vorgänge in einem System, durch die Materie, Energie oder auch Information umgeformt, transportiert oder auch gespeichert wird"[DIN87] verstanden. Man unterscheidet dabei *Rechenprozesse*, welche die "Gesamtheit der Vorgänge in einem Rechnerystem [darstelle], die an der Ausführung eines Programms oder eines sinnvoll abgegrenzten Programmteils beteiligt sind"[DIN98]. Im Gegensatz dazu versteht man unter einem *technischen Prozess* die "Gesamtheit der Vorgänge in einer technischen Anlage zur Bewältigung einer bestimmten technischen Aufgabenstellung"[DIN98]. Ein *technisches System* (auch technische Anlage nach DIN V 19223 [DIN98]) stellt dabei die Gesamtheit aller technischen Einrichtungen und Betriebsmittel dar. Der Zusammenhang zwischen Prozess und System ist graphisch in Abbildung 2.2 dargestellt. Die Automatisierung technischer Prozesse wird dabei auch als Prozessautomatisierung bezeichnet. Ein technisches System mit technischem Prozess, Rechner- und Kommunikationssystem sowie Bedienpersonal wird als (Prozess-)Automatisierungssystem bezeichnet [LG99].

Man unterscheidet hinsichtlich der Anwendungsdomäne ferner zwischen Produkt- und Anlagenautomatisierung. Man spricht dabei von *Produktautomatisierung*, wenn technische Prozesse

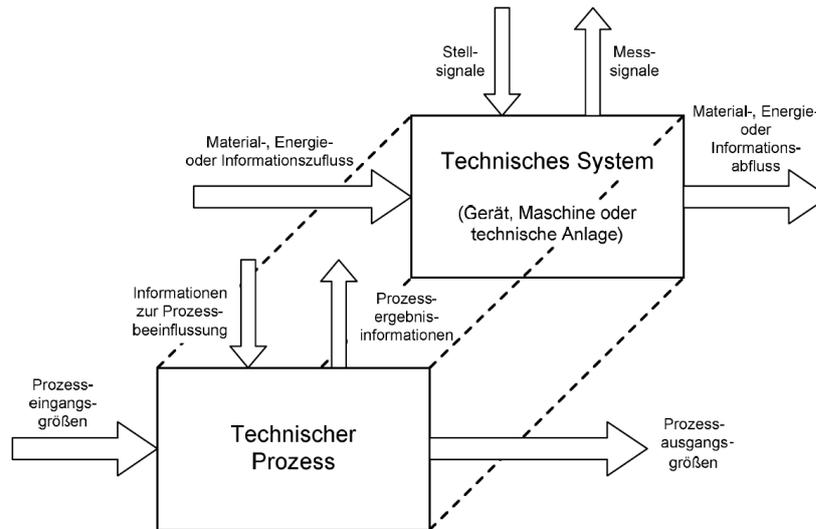


Abbildung 2.2.: Zusammenhang zwischen technischem System und technischem Prozess [LG99].

innerhalb einer einzelnen Komponente ablaufen, wie z.B. Kaffeemaschinen, Aufzüge oder Autos. Im Gegensatz dazu spricht man von *Anlagenautomatisierung*, falls ein technischer Prozess aus einzelnen Teilvorgängen (Teilprozesse) gebildet wird die durch möglicherweise räumlich ausge-dehnte, technische Anlagen ablaufen [LG99]. In Kontext der Anlagenautomatisierung, d.h. der Automatisierung im Maschinen- und Anlagenbau, kann auf Basis der Charakteristika des technischen Prozesses hinsichtlich der stofflichen Wandlung ferner zwischen fertigungs- und verfahrenstechnischen Prozessen unterschieden werden. *Fertigungstechnische Prozesse* (auch vereinfacht als Fertigungsprozesse bezeichnet) stellen technische Prozesse dar, die als eine Abfolge diskreter Schritte zur Herstellung eines Produktes oder Verarbeitung eines Werkstücks verstanden werden können. Im Gegensatz dazu stellen *verfahrenstechnische Prozesse* kontinuierliche, dynamische Vorgänge dar. Auf Basis dieser Klassifikation werden im Rahmen dieser Arbeit Anlagenauto-matisierungssysteme mit fertigungstechnischen Prozessen betrachtet. Wird daher im Folgenden von Automatisierungssystem gesprochen, ist damit ein Prozessautomatisierungssystem gemeint, welches fertigungstechnische, strikt diskrete Prozesse realisiert. Die Anlagenautomatisierung ist dadurch charakterisiert, dass sie im Gegensatz zur Produktautomatisierung einen technischen Prozess realisiert, welcher aus Teilprozessen gebildet wird (vgl. Abschnitt 2.1.2). Diese Teilprozesse – auch als Teilfunktionalitäten bezeichnet – werden durch räumlich getrennte technische (Teil-)Anlagen – im weiteren Verlauf dieser Arbeit auch als Module bezeichnet – realisiert. Die Funktionalität einer Anlage, d.h. des technischen Prozesses, wird durch die geeignete Zusammenstellung der Funktionalität einzelner Maschinen der Anlage gebildet und durch eine Vielzahl unterschiedlicher Automatisierungsfunktionen realisiert, welche durch Software auf Steuerungsebene realisiert werden.

2.1.3. Steuerungsstruktur industrieller Produktionssysteme

Ein häufig angewandter Ansatz zur Strukturierung unterschiedlicher Aufgaben eines Systems zur Produktionsautomatisierung stellt das Ebenenmodell zur Führung technischer Prozesse dar

2. Eingrenzung des Betrachtungsbereichs und Anforderungsermittlung

[LG99]. Die sogenannte Automatisierungspyramide gibt dabei zusätzlich den hierarchischen Charakter von (Steuerungs-)Entscheidungen auf den unterschiedlichen Ebenen wieder (vgl. Abbildung 2.3). Die Einteilung der Ebenen folgt dabei der im IEC 62264-1 [IEC13b] beschriebenen funktionalen Hierarchie. Steuerungsaspekte auf höheren Schichten der Softwarearchitektur werden dabei abstrahierter betrachtet als in darunterliegenden Schichten. Analog verhält sich der Zeithorizont von Steuerungsentscheidungen auf den verschiedenen Architekturschichten: der Zeithorizont sinkt innerhalb der Automatisierungspyramide von oben nach unten.

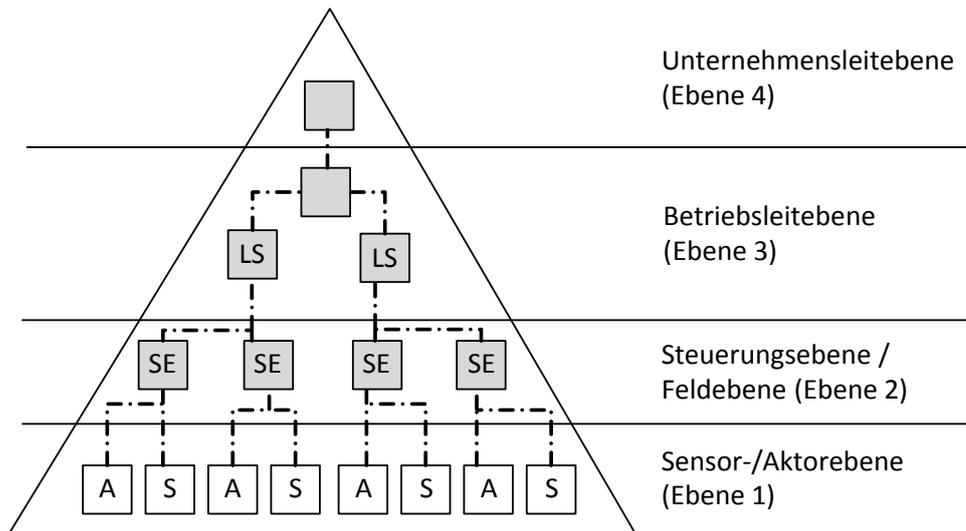


Abbildung 2.3.: Kanonische (hierarchische) Steuerungsstruktur industrieller Produktionssysteme (in Anlehnung an [IEC13b]).

Die abstrakteste Schicht (auch als Ebene 4 der Automatisierungspyramide bezeichnet) stellt die *Unternehmensleitebene* dar, in der Funktionen wie beispielsweise Ressourcenplanung, Grobplanung der Produktion, Planung von Zulieferlogistik und Lagerbeständen mit einem mittleren bis langfristigen Zeithorizont im Bereich von Monaten bis Tagen zugeordnet ist. Das Softwaresystem, welches diese Funktionalitäten realisiert, wird im deutschen als Warenwirtschafts- oder Unternehmensressourcenplanungssystem bezeichnet. Man spricht hierbei auch häufig von ERP, stellvertretend für die englische Bezeichnung “Enterprise Resource Planning“, wodurch die wesentliche Aufgabe des Systems hervorgehoben wird.

Innerhalb der *Betriebsleitebene* (Ebene 3 der Automatisierungspyramide) werden unter anderem die Feinplanung der Produktion unter Berücksichtigung von Rezepten zur Herstellung des Endproduktes realisiert, aber auch die Produktionsdatenerfassung und Qualitätsmanagement gehören zu dessen Aufgaben. Nähere Beschreibung der in Ebene 3 zusammengefassten Funktionalität kann [IEC16] entnommen werden. Softwaresysteme wie Manufacturing Execution Systems (MES) und Process Control Systems (PCS) realisieren in der Fertigungs- bzw. Prozessindustrie beispielsweise die Funktionalitäten der Ebene 3. Die Betriebsleitebene operiert üblicher Weise innerhalb eines zeitlichen Planungs- bzw. Verwaltungshorizonts von Tagen bis Minuten. Die Steuereinheiten, auf denen die Softwaresysteme der Betriebsleitebene betrieben werden, werden als Leitreechner oder Leitstationen (LS) bezeichnet.

Manchmal werden Unternehmens- und Betriebsleitebene auch als Managementebenen bezeichnet, da sie die übergeordnete Planung und Verwaltung produzierender Unternehmen realisieren.

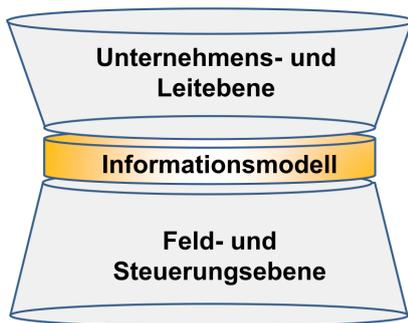
In zahlreichen Schritten werden hierarchisch von oben nach unten innerhalb der Automatisierungspyramide Kundenaufträge zeitlich geordnet, den verschiedenen Betriebsmitteln (d.h. logischen Einheiten, wie Maschinen, Linien, etc.) zugeordnet und jeweils in einen spezifischen Produktionsplan überführt, welcher durch Prozessleit- bzw. Prozesssteuerungssysteme ausgeführt und durch die darunterliegende Steuerungsebene realisiert werden. Die Planungsfunktionalität der Unternehmens- und Betriebsleitebene werden daher auch als Produktions- und Prozessplanung bezeichnet. Während die Produktionsplanung eine effiziente Ordnung von Kundenaufträgen und damit verbundene Produktionsaufträge unter Berücksichtigung von Produktionsressourcen und Logistik adressiert, stehen in der Prozessplanung die für die Fertigung eines Produktes notwendigen Schritte (auch als Rezept bezeichnet) sowie die von Betriebsmitteln auszuführenden Operationen im Vordergrund. Im englischen wird dabei oft auch von process planning und operations planning gesprochen. Im Kontext dieser Planungsaktivitäten wurde bereits Mitte der 1990 Jahre die Analogie zu automatischen (Handlungs-)Planung aus dem Bereich der Künstlichen Intelligent identifiziert [NGR95]. In diesen Planungsbereichen ist eine Planung von zeitlichen Reihenfolgen und komplexer Pläne – dem Scheduling – eng mit der Prozessplanung verbunden. Aus diesem Grund gibt es in diesem Bereich Bestrebungen, die traditionell getrennten Techniken der (Reihenfolgen-)Planung und des zeitlichen Scheduling zu kombinieren [SWH06, TK00].

Ebene 2 der Automatisierungspyramide wird durch die *Steuerungsebene* – auch Feldebene genannt – gebildet, dessen Aufgabe die „Überwachung, Führung und automatisierte Steuerung des Produktionsprozesses“ [IEC13b] darstellt. Hierbei werden harte Echtzeitanforderungen an die Software – auch als Steuerungssoftware bezeichnet – in Zeithorizonten von wenigen Minuten bis hin zu einigen Millisekunden gestellt. Steuerungseinheiten (SE) der Feldebene können Speicherprogrammierbare Steuerungen (SPSen) oder sog. Industrie-PCs (IPC) sein. Speicherprogrammierbare Steuerungen stellen Steuereinheiten dar, welche ein geschlossenes, eingebettetes Computersystem darstellen. Der Zugriff für Anwender ist auf die Implementierung von entsprechender, echtzeitfähiger Steuerungssoftware (auch als Plattform bezeichnet) und ggf. etwaiger Konfiguration der Plattform beschränkt. Im Gegensatz dazu zeichnen sich IPCs durch den Einsatz von in PCs typischer Weise eingesetzten Betriebssystemen wie Windows oder Linux aus, welche für den Anwender meist auch zugänglich sind. Um eine echtzeitfähige Ausführung der Steuerungssoftware zu gewährleisten, werden die angewandten Betriebssysteme geeignet angepasst. Teils kommen virtuelle Plattformen zum Einsatz, welche analog zu SPSen durch den Anwender genutzt werden können.

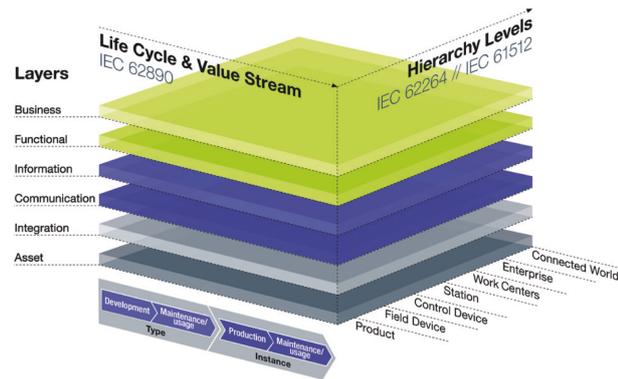
Für die Erfassung und Anpassung des Produktionsprozesses wird innerhalb der Ebene 1 (Sensor-/Aktorebene) entsprechende Hardware eingesetzt. Sensoren (S) liefern durch Messung aktuelle Prozesswerte während Aktoren (A) auf Basis gegebener Stellwerte den technischen Prozess geeignet beeinflussen.

Auf Grund einer Vielzahl von Weiterentwicklungen im Bereich der Steuerung industrieller Produktionsanlagen, insbesondere der zunehmenden Digitalisierung von Automatisierungshardware hin zu intelligenten, cyber-physischen Systemen und eine damit einhergehende Verbreitung zunehmend intelligenterer Feldgeräte mit deutlich gesteigerter Rechenkapazität, wurde durch Vogel-Heuser et al. [VhKBW09] eine an diese neuen Begebenheiten angepasste Weiterentwicklung der traditionellen Automatisierungspyramide vorgeschlagen. Die steigende Anzahl von Feldgeräten und die damit einhergehende, zunehmende Abkehr von einer strikten Hierarchisierung der Steuerungsarchitektur wird durch Vogel-Heuser et al. visuell in die Form eines Diabolo dargestellt (vgl.

2. Eingrenzung des Betrachtungsbereichs und Anforderungsermittlung



(a) Automatisierungsdiabolo nach Vogel-Heuser et al. [VhKBW09].



(b) Referenzarchitekturmodell Industrie 4.0 (RAMI 4.0) nach [BVZ15].

Abbildung 2.4.: Unterschiedliche Weiterentwicklungen der Automatisierungspyramide.

Abbildung 2.4(a): Dabei wird zwischen unterschiedlichen Funktionalität der Unternehmens- sowie der Leitebene und der (mit teilweise harten Echtzeitanforderungen konfrontierten) Feld- und Steuerungsebene unterschieden. Zusätzlich ist ein Informationsmodell enthalten, welches die horizontale und vertikale Abbildung der Kommunikation darstellt. Hierdurch wird insbesondere auch die zunehmende Bedeutung eines (expliziten) Informationsmodells innerhalb des Modells widergespiegelt.

Im Zuge der Bemühungen rund um Industrie 4.0 wurde mit dem Referenzarchitekturmodell Industrie 4.0 (RAMI 4.0) eine weitere Darstellungsform vorgeschlagen. Hierbei handelt es sich um ein Schicht- bzw. Matrixmodell (vgl. Abbildung 2.4(b)): Die Ebenen der Automatisierungspyramide wurden aufgegriffen und den unterschiedlichen funktionalen Schichten gegenübergestellt. Dadurch wird ermöglicht, dass unterschiedliche Funktionen durch Geräte auf verschiedenen Ebenen der ehemaligen Automatisierungspyramide ausgeführt bzw. realisiert werden können. Eine weitere Dimension innerhalb von RAMI 4.0 stellen unterschiedliche Lebenszyklusphasen dar. Dieser Aspekt wird hier im Folgenden nicht näher betrachtet⁵.

Unabhängig von der zu Grunde gelegten Steuerungsarchitektur industrieller Produktionssysteme stellt die Feld- und Steuerungsebene die für die Automatisierung der Anlage essentiellen Bestandteil dar. Dabei bleibt auch dessen traditionelle und essentielle Funktionalität der Realisierung des notwendigen technischen Prozesses zur Herstellung eines gewünschten Produktes in allen Architekturvarianten erhalten. Diese Funktionalität der Feld- und Steuerungsebene wird im weiteren Verlauf dieser Arbeit betrachtet.

2.1.4. Automatisierungstechnik auf der Steuerungsebene

Auf der Feldebene industrieller Produktionssysteme (d.h. Maschinen und Anlagen) werden zumeist Speicherprogrammierbare Steuerungen (SPSen) zur Ausführung notwendiger Rechenprozesse – insbesondere für die Anwendungssoftware, welche sich für die Realisierung des technischen Prozesses verantwortlich zeigt und daher auch im Folgenden als Steuerungssoftware bezeichnet wird – innerhalb eines Automatisierungssystems genutzt.

⁵Der Aspekt des Lebenszyklus wird ebenfalls im Diabolo-Modell von Vogel-Heuser et al. betrachtet und wurde hier lediglich nicht dargestellt, da dieser Aspekt im weiteren Verlauf der Arbeit nicht näher betrachtet wird.

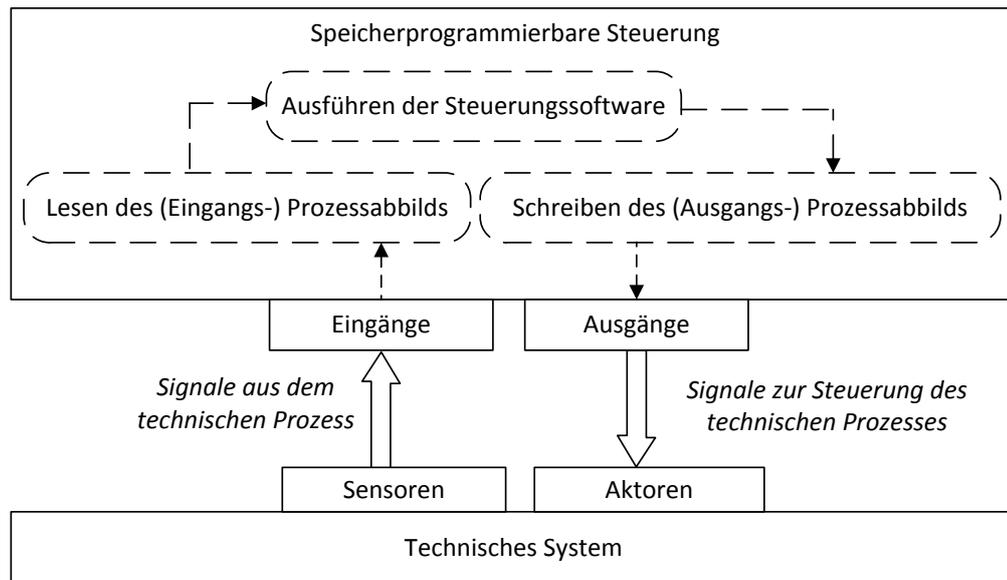


Abbildung 2.5.: Zyklische Abarbeitung eines SPS Programms im Kontext des technischen Systems (Darstellung in Anlehnung an [Grö04, LG99]).

Grundsätzlich lassen sich zwei unterschiedliche Möglichkeiten der Implementierung der Anwendungssoftware anhand ihres Ein-/Ausgabeverhaltens unterscheiden: Ereignis-basierte und zyklisch operierende Systeme. Diese Aufteilung spiegelt sich durch zwei im Maschinen- und Anlagenbau konkurrierende Standards für die Implementierung von Steuerungssoftware wieder. Während der IEC 61499 Standard [IEC12] ereignis-basiertes Steuerungsverhalten zu Grunde legt, spezifiziert der IEC 61131 Standard zyklisch operierende Systeme. In der industriellen Praxis ist dabei der IEC 61131 Standard vorherrschend und am meisten verbreitet. Dabei muss zwischen zwei unterschiedliche Versionen des Teils 3 dieses Standards unterschieden werden: der im Folgenden als klassische Version der IEC 61131-3 bezeichneten Version [IEC09] und der aktualisierten objektorientierten Version [IEC13a] aus dem Jahr 2013. Für die Implementierung der Software auf Steuerungsebene wird die klassische Version des IEC 61131-3 Standards aktuell am meisten eingesetzt [Gro11, TF⁺11]. Der Standard IEC 61499 beschreibt eine in der Forschung verbreitete Sprache; der Durchdringungsgrad von IEC 61499-konformen Steuerungen in der industriellen Praxis ist jedoch verhältnismäßig klein. Je nach Hersteller von SPSen werden weitere Sprachen zur Realisierung von Steuerungssoftware unterstützt. Eine Vielzahl von Herstellern ermöglichen beispielsweise die Implementierung der Steuerungssoftware in der Sprache Signalfussplan⁶, die für die Automatisierung kontinuierlicher und hybrider Prozesse verbreitet ist [FMS14, Bay14]. Auch ermöglichen einzelne Hersteller die Implementierung mittels Hochsprachen wie C oder C++ [VH09]. Auf Grund des industriellen Durchdringungsgrades und der herstellerunabhängigen Standardisierung werden im Folgenden zyklisch operierende SPSen und die Implementierung der Steuerungssoftware nach IEC 61131-3 näher betrachtet.

⁶Häufig findet auch die englische Bezeichnung Continuous Function Chart (CFC) im deutschsprachigen Raum Verwendung.

2. Eingrenzung des Betrachtungsbereichs und Anforderungsermittlung

Abbildung 2.5 stellt zyklisch operierende SPSen im Kontext des technischen Systems dar. Sensoren liefern Signale des technischen Prozesses innerhalb des technischen Systems. Diese werden mittels Kommunikationssystem (nicht in der Abbildung dargestellt) an die Eingänge der SPS übertragen. Zyklisch operierende SPSen lesen zu diskreten Zeitpunkten die Werte der Eingänge und bildet ein informationelles Abbild der Signale des technischen Systems. Dies wird auch als das Lesen des Eingangsprozessabbilds bezeichnet. Nun wird die Steuerungssoftware ausgeführt und auf Basis des Eingangsprozessabbilds Stellwerte der Aktoren berechnet, um einen gewünschten technischen Prozess zu realisieren. Diese Aktorstellwerte werden innerhalb des sogenannten Ausgangsprozessabbilds informationell gespeichert und zu diskreten Zeitpunkten vollständig an die Ausgänge und letztendlich als Signale an die Aktoren übertragen.

Die klassische Version der IEC 61131-3 definiert sowohl die textuellen Sprachen Strukturierter Text (ST) und Anweisungsliste (AWL) als auch die graphische Programmiersprachen Funktionsbausteinsprache (FBS), Kontaktplan (KOP) und Ablaufsprache⁷ (AS). Einen Überblick dieser Programmiersprachen ist in Abbildung 2.6 dargestellt. Eine vollständige Spezifikation eines konsistenten Metamodells für Softwareprojekte entsprechend dem IEC 61131-3 Standard kann [Wit12] entnommen werden.

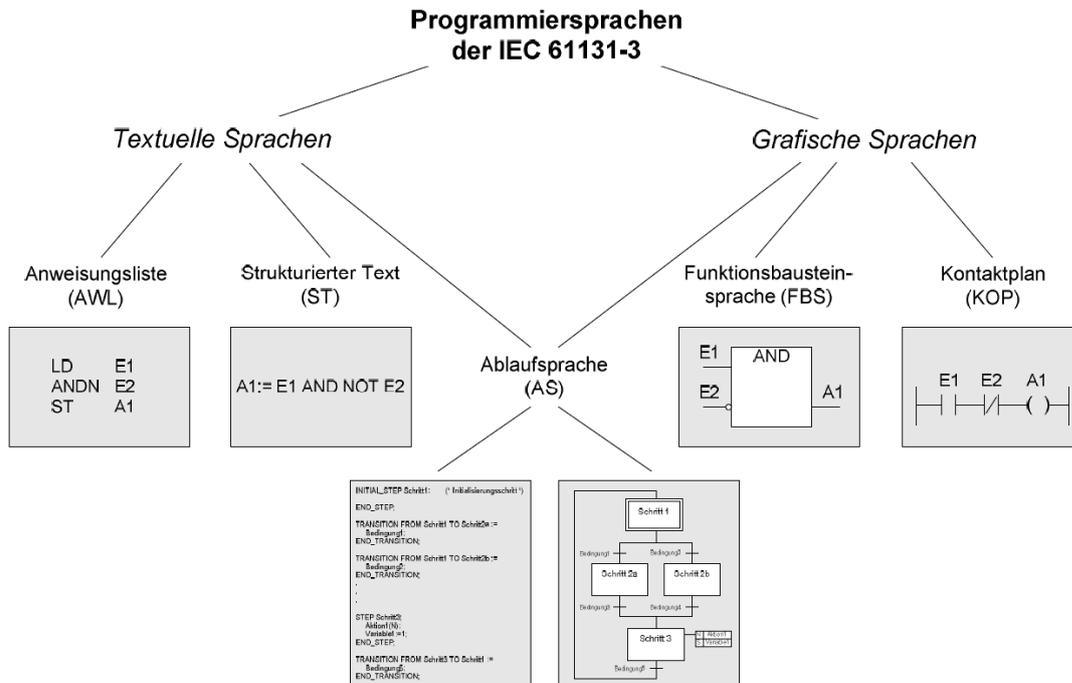


Abbildung 2.6.: Überblick der Programmiersprachen nach IEC 61131-3 [IEC09](visuelle Aufbereitung aus [Fri09] entnommen).

Der Standard definiert Programmorganisationseinheiten (POUs) zur Strukturierung und Wiederverwendung von SPS-Programmcode. Ein Programm stellt dabei die Sammlung aller logischen Elemente dar, die für die Steuerung einer Maschine oder Anlage notwendig ist und bildet den Eintrittspunkt für das Steuerungsprogramm. Innerhalb von Programmen können Funktionsbau-

⁷Für die Programmiersprache AS existiert auch eine textuelle Repräsentation. Meist wird mit AS jedoch graphisch programmiert.

steine (FB) und Funktionen (FC) zur Strukturierung von Programmcode genutzt werden. FBs berechnen Ausgabewerte auf Basis von Parametern und persistenter interner Variablenwerte, während FCs Ausgabewerte ausschließlich auf Basis von Eingabeparametern berechnen können. Programme und FBs können instanziiert werden und halten Variablenwerte zur Laufzeit persistent, während Funktionen als global und zustandslos charakterisiert werden können. POU's können jeweils in einer beliebigen der fünf standardkonformen Sprachen implementiert werden; eine Mischung verschiedener Sprachen innerhalb eines Programms ist üblich; die Ausdrucksmächtigkeit der einzelnen Sprachen unterscheidet sich dabei. Die textuellen Programmiersprachen nach IEC 61131-3 lassen sich wie folgt charakterisieren: Während AWL eine der Programmierung in Assembler ähnliche Sprache ist, bei der einzelne Anweisungen listenartig untereinander – d.h. eine Anweisung je Zeile – angegeben werden, ist ST hingegen ähnlich der Hochsprache Pascal und bietet eine Vielzahl von Operatoren typischer, imperativer Programmiersprachen. Die Programmiersprache AS basiert auf der Modellierung von Abläufen auf Basis der Reihenfolge von Schritten und damit verbundenen Übergangsbedingungen (Transitionen). FBS stellt eine graphische, funktionsbasierte Sprache dar, die es ermöglicht, Netzwerke aus POU's zu definieren, die eine logische Verknüpfung auf Basis der Ein-/Ausgaben der POU's ermöglicht. Im Gegensatz dazu ist die graphische Programmiersprache KOP einem elektrischen Stromlaufplan nachempfunden. Zwischen vertikal verlaufenden Stromschienen werden Bestandteile der (logischen) Schaltung als Netzwerke dargestellt. Weitere Details zu den Programmiersprachen kann dem Standard [IEC09] oder der zahlreich existierenden Fachliteratur entnommen werden.

Aufbauend auf dem zuvor beschriebenen, klassischen IEC 61131-3 wurde im Jahr 2013 eine objektorientierte Erweiterung des IEC 61131-3 Standards (3rd Edition) veröffentlicht [IEC13a]. Dieser definiert keine neuen Sprachen, sondern eine konsequente, objektorientierte Erweiterung des bisherigen Sprachumfangs. An dieser Stelle wird grundsätzliche Kenntnis und Verständnis der objektorientierten Softwareentwicklung vorausgesetzt. Eine detaillierte Einführung der grundlegenden Konzepte objektorientierter Softwareentwicklung kann beispielsweise [RBP⁺91, VHW07] entnommen werden. In der objektorientierten Version der IEC 61131-3 wurden neben der Möglichkeit von Klasse-Instanz-Beziehungen auch Schnittstellen und Vererbung eingeführt [Wer09, WVH09]. Das Konzept der POU's wird dabei dementsprechend erweitert: Funktionsbausteinen werden als Klassen aufgefasst⁸, denen nun zusätzlich Methoden zugeordnet werden können, um Kapselung von Funktionalität zu ermöglichen. Analog zu objektorientierten Hochsprachen besteht eine Methode aus einer Methodensignatur (dem sogenannte Deklarationsteil) und einem Methodenrumpf, welcher die Implementierung der Methode enthält, wobei auf die Variablen des FBs zugegriffen werden kann. Die objektorientierte IEC 61131-3 bietet ferner die Möglichkeit der Definition von Schnittstellen. Analog zu FBs können Schnittstellen Methodendeklarationen enthalten, nicht jedoch einen Methodenrumpf, Instanzen von FBs oder Variablen enthalten. Der objektorientierte Funktionsumfang wird durch die Möglichkeit der einfachen Vererbung⁹ abgerundet und berücksichtigt dabei essentielle Eigenschaften der objektorientierten Modellbildung, wie z.B. Polymorphismus [CW85] und das Liskovsches Substitutionsprinzip [LW94]. Mit diesen Spracherweiterungen ist durch die IEC 61131-3 (3rd Edition) eine objektorientierte Softwareent-

⁸FBs können bereits in der klassischen IEC 61131-3 instanziiert werden und bieten somit eine geeignete Grundlage für die objektorientierte Erweiterung.

⁹Eine Mehrfachvererbung zwischen FBs wird nicht unterstützt, es können jedoch beliebig viele Schnittstellen durch einen FB realisiert werden.

2. Eingrenzung des Betrachtungsbereichs und Anforderungsermittlung

wicklung möglich und bietet die Vorteile der Objektorientierung, wie Kapselung von Funktionalität und Daten, und damit eine Vereinfachung der Wiederverwendbarkeit. Durch die Erweiterung des Sprachkonzeptes ohne Einführung neuer Sprachen ist ferner eine Koexistenz beider Versionen der IEC 61131 möglich, wodurch ein nahtlosen Übergang von der klassischen zur objektorientierten Version ermöglicht wird.

2.2. Anforderungen an den Lösungsansatz

Im Rahmen dieser Dissertation wird die Anwendbarkeit existierender Verfahren zur automatisches Handlungsplanung für die Feldebenesteuerung im Maschinen- und Anlagenbau untersucht und basierend auf den Untersuchungsergebnissen ein geeignetes Planungsverfahren zur automatischen Bestimmung der Ablaufreihenfolgen von Automatisierungsfunktionen entwickelt. Dabei werden an diesen Ansatz einige, für die Software auf der Feldebene der Maschinen- und Anlagenautomatisierung spezifische Anforderungen gestellt, um dessen Anwendbarkeit innerhalb der Domäne im Allgemeinen und für gewünschte Zielsetzungen bzw. Anwendungen im Speziellen sicherzustellen. Letzteres ist insbesondere durch die in Abschnitt 1.1 beschriebenen Anwendungsbeispiele gegeben. Abbildung 2.7 stellt diese Anforderungen den in der konzeptionellen Betrachtung der Anwendungsbeispiele identifizierten, notwendigen Konzeptbausteinen gegenüber.

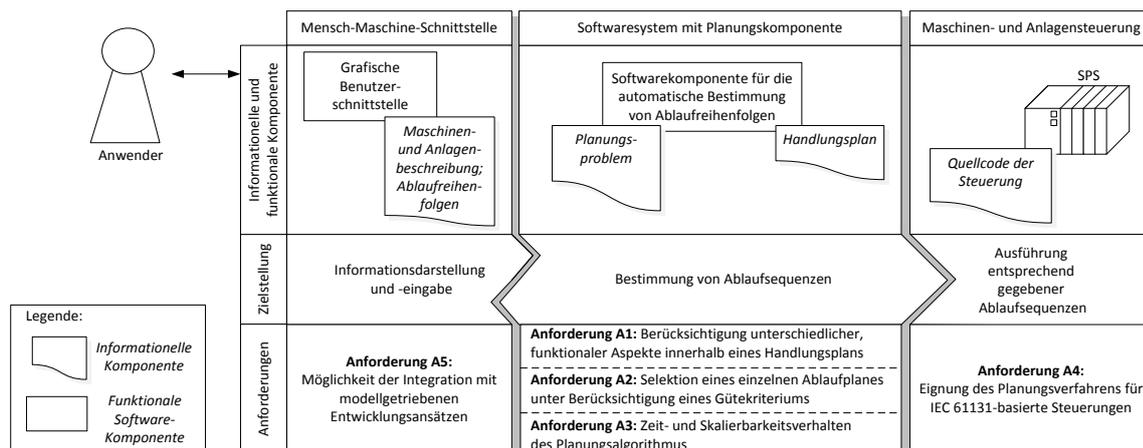


Abbildung 2.7.: Übersicht der Anforderungen im Hinblick auf notwendige, konzeptionelle Komponenten zur Realisierung der Anwendungsfälle.

Wie im Rahmen der konzeptionellen Betrachtung der Anwendungsbeispiele identifiziert, lassen sich sowohl der Anlagenkonfigurator (vgl. User Story 1 in Abschnitt 1.1.1) als auch das Betriebs- und Störungssystem (vgl. User Story 2 in Abschnitt 1.1.2) auf eine Softwarekomponente zur automatischen Handlungsplanung zurückführen. Auf Basis einer detaillierten Betrachtung der Anforderungen an eine solche Planungskomponente werden im Folgenden drei wesentliche funktionale Anforderungen identifiziert. Dabei werden einerseits die zu berücksichtigenden Einschränkungen durch das technische System, dem notwendigen Fertigungsprozess sowie der generellen Struktur von Handlungsplänen für die Maschinen- und Anlagensteuerung im Rahmen der Anforderung A1 hergeleitet und damit die Korrektheit des zu bestimmenden Handlungsplanes für eine spezifische Maschine bzw. Anlage sichergestellt. Für die Maschinen- und Anlagenautomatisierung

gelten ferner weitere Anforderungen hinsichtlich der Selektion eines von möglicherweise mehreren, korrekten Handlungsplänen (Anforderung A2). Ferner werden an ein geeignetes Verfahren für die Bestimmung von Ablaufreihenfolgen spezifische Anforderungen im Hinblick auf die Leistungsfähigkeit – d.h. insbesondere Zeit- und Skalierbarkeitsverhalten des Planungsalgorithmus – gestellt (Anforderung A3). In beiden beschriebenen Anwendungsbeispielen wird eine graphische Benutzerschnittstelle genutzt, um einerseits Informationen für den Anwender geeignet darzustellen als auch die Eingabe notwendiger Informationen für eine automatische Handlungsplanung zu ermöglichen. Diese Anforderung wird mittels Anforderung A5 hinsichtlich einer Integration des Planungsverfahrens in einen modellgetriebenen Entwicklungsansatz repräsentiert. Neben der (graphischen) Schnittstelle zwischen Softwaresystem und Mensch wird in beiden Anwendungsbeispielen eine Anbindung an die Anlagensteuerung benötigt: Während für den Anlagenkonfigurator lediglich die Möglichkeit der Generierung von Programmcode aus durch die Ablaufplanung bestimmten Handlungsplänen benötigt wird, ist für die Realisierung des Betriebs- und Störungssystems eine engere Kopplung zur Anlagensteuerung notwendig, um dessen Verhalten geeignet anzupassen. Dementsprechend stellt die Anbindung des Planungsverfahrens an für die Maschinen- und Anlagenautomatisierung typischer Weise genutzte Steuerungshardware eine Notwendigkeit dar, die durch Anforderung A4 reflektiert wird. Als Beispiel dient in der detaillierten Herleitung und Beschreibung der Anforderungen eine sogenannte Pick-and-Place Einheit, wie exemplarisch in Abbildung 2.8 in Form eines pneumatischen Greifers dargestellt, welche häufig in der Fertigung für die Handhabung von Werkstücken genutzt wird.

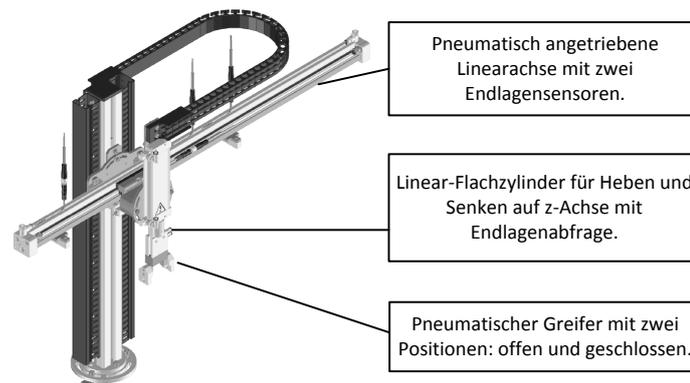


Abbildung 2.8.: Mechanischer Aufbau eines beispielhaften, pneumatischen Greifers [EP10].

2.2.1. Anforderung A1: Berücksichtigung unterschiedlicher, funktionaler Aspekte innerhalb eines Handlungsplans

An die Bestimmung der Ablaufreihenfolge von Automatisierungsfunktionen werden besondere Anforderungen hinsichtlich funktionaler Aspekte des zu realisierenden, technischen Prozesses gestellt. In der klassischen Planung wird primär ein Handlungsplan bestimmt, welcher ein betrachtetes (technisches) System von einem Initial- in einen Zielzustand versetzt ist (vgl. Abschnitt 2.1.1). Im Gegensatz dazu sind valide Handlungspläne für die Steuerungssoftware von Maschinen und Anlagen durch unterschiedliche Teilpläne charakterisiert, welche unterschiedliche Zielsetzungen erfüllen bzw. unterschiedliche funktionale Aspekte realisieren. Die Ordnungsrela-

2. Eingrenzung des Betrachtungsbereichs und Anforderungsermittlung

tion zwischen entsprechenden Teilplänen ist dabei nicht auf eine totale Ordnung beschränkt, d.h. eine sequenzielle Abarbeitung von Teilplänen ist nicht ausreichend. Vielmehr ist eine spezifische teils zyklische Anordnungen der Teilpläne notwendig, um die gewünschte Funktionalität der Steuerungssoftware in Form von Handlungsplänen abbilden zu können.

Aktuell kann ein Trend beobachtet werden, mehr und mehr kundenspezifische und kundenindividuelle Produkte anzubieten. Dies könnte letztendlich zu stetig kleineren Losgrößen in der Produktion bis hin zur Losgröße 1 führen [Pin99, RCD⁺04]. Zahlreiche internationale Forschungsaktivitäten untersuchen bereits diverse Aspekte der kundenindividuelle Massenproduktion, z.B. in der Bekleidungsindustrie [WKC09]. Nichtsdestotrotz ist in einer Vielzahl von Branchen – speziell in der Produktion und Fertigung von Verbrauchsgütern – die Massenproduktion aktuell nach wie vor das vorherrschende Produktionsparadigma und wird dies voraussichtlich auch weiter bleiben. Aus diesem Grund wird im Rahmen dieser Dissertation die automatisierte Massenproduktion fokussiert¹⁰.

Der Eingang eingeführte Sachverhalt der Notwendigkeit verschiedener Teilpläne innerhalb eines Handlungsplanes für die Steuerungsebene des Maschinen- und Anlagenbaus zur Berücksichtigung unterschiedlicher, notwendiger Teilfunktionalität wird im Folgenden im Hinblick auf die Massenfertigung von Gütern näher erläutert und anhand von Beispielen verdeutlicht.

Anforderung A1.1: Realisierung des Fertigungsprozesses

Die automatisierte Ausführung eines technischen (Fertigungs-)Prozesses stellt die wesentliche Zielsetzung der Automatisierung von Maschinen und Anlagen dar [VHDF⁺14, LG99]. Aus diesem Grund soll das im Rahmen dieser Dissertation beschriebene Planungsverfahren eine Identifikation von Ablaufreihenfolgen von Automatisierungsfunktionen unterstützen, welche einen gewünschten technischen Prozess für die Fertigung eines Gutes realisieren. Spezifische Anforderungen, welche an den für die Fertigung notwendigen Fertigungsprozess gestellt werden, müssen demzufolge innerhalb eines geeigneten Handlungsplans berücksichtigt werden, um die Korrektheit eines Handlungsplanes und damit letztendlich des realisierten technischen Prozesses sicherzustellen. Die Korrektheit bzw. Validität eines Handlungsplanes für die Feldebeneensteuerung ist ferner von den Einschränkungen des technischen System abhängig. Diese ergeben sich beispielsweise durch das mechanische System der Maschine bzw. Anlage. Eine Überschreitung dieser dadurch gegebenen Grenzen kann in einer Gefährdung des technischen Systems resultieren und muss demzufolge ausgeschlossen werden. Eine detaillierte Diskussion dieser beiden Klassen von Einschränkungen für die Steuerungsebene der Maschinen- und Anlagenautomatisierung kann u.a. [SWLVH13, Wan10] entnommen werden. Demzufolge muss durch ein geeignetes Planungsverfahren ein (Teil-)Plan bestimmt werden, welcher den technischen Prozess für die Fertigung eines Produktes realisiert und dabei die zuvor beschriebenen Einschränkungen berücksichtigt. Im Folgenden wird davon ausgegangen, dass sich die Maschine bzw. Anlage bereits in einer sog. Grundstellung befindet, welche den direkten Anlauf des notwendigen Fertigungsprozesses ermöglicht.

Beispiel: Für den als durchgängiges Beispiel genutzten, pneumatischen Greifer muss beispielsweise sichergestellt werden, dass eine Bewegung auf der Linearachse über die Endlagensensoren hinaus ausgeschlossen ist, um eine Beschädigung des Aufbaus zu vermeiden. Betrachtet man

¹⁰In Abschnitt 6 wird in Form eines Ausblicks beschrieben, dass sich das in dieser Arbeit erarbeitete Konzept zur Adressierung dieser Anforderung dementsprechend erweitern lässt, um auch eine Planung für eine kundenindividuelle, multivariate Fertigung zu ermöglichen

den Greifer im Kontext einer Anlage mit vor- und nachgelagerter Maschine, wie beispielhaft in Abbildung 2.9 skizziert, ergeben sich weitere, zusätzliche Einschränkungen, die innerhalb eines korrekten Handlungsplanes für die Feldebene Steuerung Berücksichtigung finden müssen.

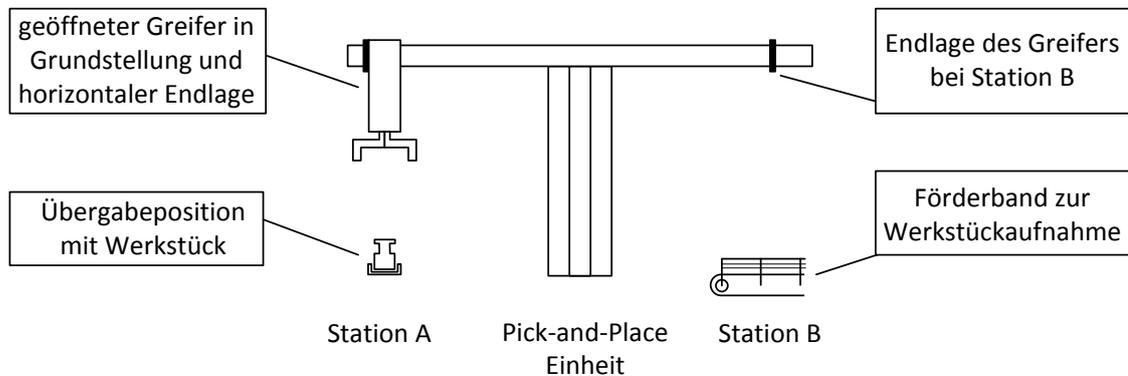


Abbildung 2.9.: Schematische Darstellung des Greifers mit Übergabepositionen der vor- und nachgelagerten Stationen.

Wie in Abbildung 2.9 auf der linken Seite des pneumatischen Greifers dargestellt, wird das Werkstück durch Station A in einer Aufnahmeposition übergeben. Die Fixierung des Werkstücks erfolgt durch eine vertiefte Position des Werkstücks. Auf Grund dieses mechanischen Aufbaus muss das Werkstück durch den Greifer zuerst angehoben werden, bevor horizontale Bewegungen des Greifers ausgeführt werden dürfen. Im Gegensatz zu der rein mechanischen Übergabeposition von Station A ist die Übergabeposition an Station B mittels eines Förderbandes realisiert (wie in Abbildung 2.9 rechts des Greifers schematisch dargestellt). Eine Aufnahme bzw. Ablage des Werkstücks bei Station B ist ausschließlich dann möglich, wenn keine Bewegung durch das Band ausgeführt wird. Andernfalls ist keine gesicherte Werkstückaufnahme durch den Greifer möglich. Die Entscheidung, ob die Werkstückaufnahme bei Station A oder Station B erfolgen muss und dementsprechend, ob ein Transport des Werkstücks von Station A zu Station B oder umgekehrt realisiert werden muss, soll durch das zu realisierende Planungsverfahren unter Berücksichtigung des (hier nicht näher spezifizierten) Anlagenkontextes bestimmt werden.

Anforderung A1.2: Grundstellungsfahrt zur erneuten Ausführung des Fertigungsprozesses

Aus Sicht der Automatisierungstechnik stellt die Massenfertigung von Gütern die Anforderung, große Stückzahlen identischer Güter möglichst effizient fertigen zu können. Im Falle der Fertigung von Stückgütern ergibt sich hieraus eine weitere Herausforderung: Maschinen müssen neben dem eigentlichen, zu realisierenden Fertigungsprozess – d.h. dem technischen Prozess zur Fertigung eines einzelnen Produktes – ggf. weitere Funktionalitäten ausführen, um den (entsprechend Anforderung 1.1) zu realisierenden Fertigungsprozess für ein weiteres Gut erneut durchführen zu können. Dieser Teil des auszuführenden, technischen Prozesses dient im Gegensatz zu dem in Anforderung 1.1 geforderten Teilprozess nicht der Fertigung/Handhabung eines einzelnen Werkstücks, sondern ermöglicht dessen erneute Ausführung durch Manipulation des Zustands der Maschine bzw. Anlage derart, dass diese sich (ausgehend von der Endposition nach dem Ferti-

2. Eingrenzung des Betrachtungsbereichs und Anforderungsermittlung

gungsprozess) wieder in Grundstellung befindet. Während der Ausführung dieses (Teil-)Prozesses müssen die durch den mechanischen Aufbau des technischen Systems gegebenen Grenzen (analog zu Anforderung 1.1) eingehalten werden. Demzufolge muss ein geeignetes Planungsverfahren einen entsprechender Handlungsplan bzw. Teilprozess bestimmen, um eine Maschine bzw. Anlage nach Beendigung des Fertigungsprozesses eines einzelnen Gutes wieder in Grundstellung zu versetzen. Eine spezifische Grundstellung ist damit einerseits von dem eingesetzten, technischen System abhängig, aber andererseits auch von dem zu realisierenden, technischen Fertigungsprozess und damit von einem spezifischen Produkt abhängig.

Beispiel: Die Notwendigkeit einer Grundstellungsfahrt lässt sich erneut anhand des exemplarischen, pneumatischen Greifers veranschaulichen. Nach Ablage von Werkstücken bei Station B muss der pneumatische Greifer erneut zur Grundstellung bewegt werden, um ein weiteres Werkstück bei Station A aufnehmen zu können. Dies beinhaltet unter anderem auch das Anheben des Greifers, da eine direkte horizontale Linearbewegung eine Kollision mit einem in der Aufnahme-position befindlichen Werkstück verursachen würde.

Anforderung A1.3: Berücksichtigung des Initialzustandes von Produktionsanlagen

Nahezu jede Anlage in der Prozessindustrie enthält spezielle Teile des Steuerungsprogramms für den Anlauf, welcher typischer Weise manuell mit wenig Unterstützung erstellt wird [KL93]. Analog gilt dies für Fertigungssysteme, wie eine Vielzahl von Beispielen zeigt: der Druck pneumatischer Bestandteile von Maschinen muss bei Inbetriebnahme erst mittels Kompressor erzeugt werden, bevor entsprechende mechanische Bewegungen erzielt werden können; auch die Ruheposition von Roboterarmen stellt ebenfalls eine Position dar, die während der Produktion selten erreicht wird. Forschungen, welche diese nicht-produktiven Phasen adressieren, um den Anlauf oder Abschalten chemischer Anlagen zu vereinfachen [KB91], den Wiederanlaufs von Montagezellen im Fehlerfall untersuchen [ALFF11] oder neue Applikationen wie beispielsweise ein Art Start-Stop Automatik von Fertigungsanlagen zur energieeffizienten Produktion [MLPM13b] untersuchen, führen zu einer zusehends steigenden Bedeutung des (Wieder-)Anlaufs von Maschinen und Anlagen. Für den Anlauf eines Fertigungssystems sind dabei Informationen über den technischen Prozess von wesentlicher Bedeutung, da nach erfolgreichem Anlauf die (möglichst effiziente) Massenfertigung (vgl. auch Anforderung A1.1) sichergestellt werden muss. Der Initialzustand ist dabei ausschließlich vom technischen System abhängig und kann sich deutlich in Abhängigkeit eingesetzter Komponenten, z.B. Ventile innerhalb des pneumatischen (Teil-)Systems unterscheiden. Aus diesem Grund soll im Rahmen dieser Dissertation die Möglichkeit der Berücksichtigung des Initialzustands einer Maschine oder Anlage — d.h. der Zustand nach Start aller Komponenten ohne Ausführung einer explizit für den Anlauf implementierten Software durch die Steuerung — untersucht und die Planung des Anlaufs ausgehend von diesem Initialzustand durch das entwickelte, automatische Planungsverfahren realisiert werden. Dieser zu bestimmende Handlungsplan realisiert dabei die als Grundstellungsfahrt bezeichnete Funktionalität ausgehend von dem Initialzustand der Maschine bzw. Anlage.

Beispiel: Der zuvor als Beispiel bereits genutzte, pneumatische Greifer befindet sich nach der Inbetriebnahme, d.h. nach Aufbau des Drucks durch einen angeschlossenen Kompressor und

aller Sensoren in Ruhelage¹¹, in einer Position welche durch ein geeignetes Planungsverfahren als valide Grundstellung identifiziert werden würde. Konkret kann dies im hier vorliegenden Fall bedeutet, dass sich der Greifer in der Endlage der Linearachse bei Station A mit eingefahrenem Flachzylinder bzgl. der z-Achse befindet (wie in Abbildung 2.9 dargestellt).

Um die hier beschriebene Anforderung zu veranschaulichen und zu präzisieren, nachfolgend ein weiteres Beispiel: Für die Fertigung unterschiedlicher Varianten eines Produktes werden teils flexible, auf Warenträger basierende Intralogistiksysteme eingesetzt, um Werkstücke zwischen einzelnen Stationen transportieren zu können. Ein solches System wird in der industriellen Praxis beispielsweise von der Siemens AG für die Produktion von Schaltschützen genutzt [KV08]. Abbildung 2.10 zeigt einen Ausschnitt eines solchen, exemplarischen Warenträgersystem zur Verbindung unterschiedlicher Stationen. Die genutzte Weichen befinden sich alle stets in einem identischen Initialzustand durch den alle Warenträger linear geroutet werden. Um die Fertigung einer spezifischen Variante des Produktes zu ermöglichen, muss die Position mancher Weichen derart verändert werden, so dass sich das Transportsystem in der für diese Variante passende Grundstellung befindet und dadurch alle Warenträger samt Werkstücke die für die Fertigung notwendigen Stationen erreichen.

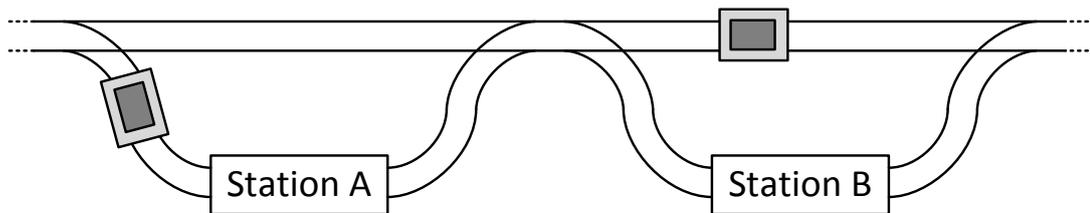


Abbildung 2.10.: Schematische Darstellung eines Intralogistiksystems mittels schienengeführter Warenträger.

2.2.2. Anforderung A2: Selektion eines einzelnen Ablaufplanes unter Berücksichtigung eines Gütekriteriums

Im Hinblick auf den Trend zunehmend flexiblere Maschinen zu entwickeln [MUK00] ist es denkbar und möglich, dass mehrere korrekte und mögliche Abläufe für die Fertigung eines gewünschten Produktes bestimmt werden können. Die eingangs in dieser Arbeit beschriebenen Anwendungsbeispiele erfordern jedoch, dass ein Verfahren zur Bestimmung von Abläufen stets einen spezifischen Ablauf bestimmt, welcher dem Nutzer als Vorschlag im Sinne einer Assistenz unterbreitet wird. Aus diesem Grund muss ein geeignetes Planungsverfahren eine Selektion durchführen können, um einen einzelnen Ablaufplan zu selektieren, falls mehrere korrekte Pläne existieren.

Die Maximierung der Rendite des eingesetzten Kapitals ist für Unternehmen, welche Maschinen und Anlagen zur Fertigung von Gütern betreiben, von wesentlicher Bedeutung. Daher stellt

¹¹In Abhängigkeit der genutzten Ventile für das pneumatische Teilsystem und deren unterschiedliche Ruhepositionen und daraus resultierende Strömungseigenschaften der Druckluft, können sich unterschiedliche Initialzustände des technischen Systems ergeben. Einen Überblick über die Unterschiede einzelner Ventile gibt z.B. [EIPS08].

2. Eingrenzung des Betrachtungsbereichs und Anforderungsermittlung

die Implementierung effizienter Algorithmen bei der Realisierung der Steuerungssoftware auf der Feldebene eine der grundlegenden Zielsetzungen dar. Unter Effizienz wird in diesem Zusammenhang die Sparsamkeit des Algorithmus hinsichtlich der verbrauchten (Produktions-)Ressourcen verstanden. In Abhängigkeit der Zielsetzung können diese Ressourcen beispielsweise Energie oder Materialausschuss sein. Im Zusammenhang mit der Renditemaximierung produzierender Unternehmen stellt jedoch die Maximierung der produzierten Stückzahlen eines Gutes einen der am häufigsten zu Grunde gelegten Leistungsindikatoren dar.

Während auf logisch höheren Schichten der Steuerungsarchitektur¹² im Bereich der Produktionsplanung bzw. Produktionsfeinplanung zumeist Kundenaufträgen, deren Lieferterminen und Produktionsdurchsatz Berücksichtigung mittels abstrahierter (meist relativ kleiner) Menge von Maschinenkonfigurationen erfolgt, ist auf der Feldebene eine wesentlich feingranulare Betrachtung von Funktionen notwendig (vgl. Anforderung A1) und dementsprechend unterscheiden sich die Betrachtungsbereiche des Optimierungskriteriums deutlich.

Folglich ist es unumgänglich, dass ein automatisches Planungsverfahren für die Steuerungsebene des Maschinen- und Anlagenbaus nicht nur einen validen Plan im Hinblick auf zu berücksichtigende Randbedingungen (vgl. Anforderung A1), sondern vielmehr einen guten oder gar optimalen Plan im Hinblick auf ein gegebenes Optimierungskriterium identifizieren kann. Da in dieser Arbeit die Massenproduktion von Gütern zu Grunde gelegt wird (vgl. Anforderung A1), sollte ein geeignetes Planungsverfahren zumindest ermöglichen, einen validen Plan zu identifizieren, welcher eine Maximierung des Durchsatzes einer Maschine bzw. Anlage erlaubt.

Beispiel: Auf Basis des mechanischen Aufbaus des pneumatischen Greifers wäre es grundsätzlich möglich, dass sich dieser theoretisch oft zwischen den Endlagen der Linearachse horizontal bewegt, bevor ein Werkstück aufgenommen bzw. abgelegt wird. In Anbetracht eines, den Durchsatz optimierenden Verhaltens des Greifers wäre eine einmalige, lineare Bewegung (sowohl mit als auch ohne Werkstück) zielführend und sollte dementsprechend durch ein geeignetes Planungsverfahren identifiziert und selektiert werden.

Ein weiteres Beispiel stellen automatisierte Materialflusssysteme für die Intralogistik dar, z.B. bestehend aus zahlreichen mit einander verbundenen Förderbändern. Die Umlenkung von transportierten Werkstücken auf unterschiedliche Routen erfolgt hier häufig durch Weichen (vgl. hierzu auch Abbildung 2.10). In Analogie zum menschlichen Körper beinhalten solche Materialflusssysteme häufig sogenannte Bypässe, welche alternative Routen bilden, falls es auf der Haupttransportroute (der sogenannte Aorta) zu Störungen kommen sollte. Nichtsdestotrotz sollte stets die beste (im durchsatzoptimierenden Fall die schnellste) Route für Werkstücke gewählt werden. Um dies zu erreichen, müssen etwaige installierte Weichen gestellt und die Antriebe der notwendigen Förderbänder geeignet (d.h. im Falle bidirektionaler Antriebe in der korrekten Antriebsrichtung) gestartet werden. Dies sollte durch ein geeignetes Planungsverfahren automatisch und korrekt identifiziert werden.

¹²Auch in nicht hierarchischen Steuerungsstrukturen im Umfeld von Industrie 4.0 wie dem Automatisierungsdiabolo und RAMI 4.0 (vgl. Abschnitt 2.1.3) werden entsprechende Funktionen konzeptionell noch als logisch höher betrachtet, da beispielsweise der Zeithorizont für einen Teil dieser Funktionen nach wie vor größer ist als auf logisch darunter liegenden Schichten; der Übergang zwischen den Schichten ist jedoch fließend.

2.2.3. Anforderung A3: Zeit- und Skalierbarkeitsverhalten des Planungsalgorithmus

Die automatische Bestimmung von Ablaufreihenfolgen stellt eine algorithmisch komplexe und rechenintensive Aufgabenstellung dar [GNT04, RN95]. Das Antwortverhalten von Planungssystemen, d.h. die zeitliche Dauer der Bestimmung eines geeigneten Handlungsplanes, ist im Maschinen- und Anlagenbau nicht unerheblich. Eine konkrete Zeitanforderung hängt dabei jedoch stark von dem jeweiligen Anwendungsfall und Einsatzzweck des Planungsalgorithmus ab (vgl. Abschnitt 2.1.3). Um die Anforderungen an die Leistungsfähigkeit hinsichtlich Zeit- und Skalierbarkeitsverhalten des im Rahmen dieser Arbeit beschriebenen Planungsverfahrens zu konkretisieren, werden im Folgenden die in Abschnitt 1.1 beschriebenen Anwendungsfälle auf die durch sie gestellten zeitlichen Anforderungen hin untersucht.

In Anwendungsbeispiel 1 (vgl. Abschnitt 1.1.1) wird eine Assistenz bei der kundenspezifischen Konfiguration der Steuerung einer Anlage betrachtet. Das zeitliche Verhalten der Generierung von Vorschlägen für Steuerungsabläufe stellen in diesem Anwendungskontext die Wartezeit auf das Ergebnis des Anwenders der Software dar. Ist eine geeignete Integration des Planungswerkzeugs in die betrachtete Konfigurationssoftware gegeben, kann durch den Anwender parallel zur Durchführung der Planung des Software-Werkzeug weiter genutzt werden¹³. Somit ist eine Durchführung der Planung innerhalb einiger Minuten oder weniger als für diesen Anwendungsfall ausreichende, zeitliche Anforderung zu betrachten.

In eigenen Vorarbeiten wurden Einflussfaktoren im Hinblick auf einen industriell vorteilhaften Einsatz einer dynamischen Rekonfiguration (vgl. Anwendungsfall 2 in Abschnitt 1.1.2) der Steuerungssoftware insbesondere auch unter Berücksichtigung des zeitlichen Verhaltens der Berechnung von Handlungsplänen – d.h. der automatischen Planung – und Durchführung einer Rekonfiguration untersucht, diskutiert und mathematisch in Kontext gesetzt [LVH14a]. Dabei wurde angenommen, dass die in Abschnitt 1.1.2.2 als Alarmmanagement und Diagnosesystem bezeichnete Softwarekomponente bereits einen sofortigen Stopp des Fertigungsablaufes vornimmt, so dass durch eine automatische Planung alternativer Abläufe unter Berücksichtigung der Störung eine Zeitkritizität im Hinblick auf den technischen Prozess vermieden wird. Ferner wurde in der Betrachtung, analog zu Anforderung A2, eine durchsatzoptimierte Fertigung zu Grunde gelegt. Im Rahmen dieser Untersuchung konnte identifiziert werden, dass der Zeitbedarf für die Bestimmung eines alternativen Verhaltens der Maschine und Anlage im Fehlerfall lediglich einen marginalen Einflussfaktor – d.h. einen geringen, durch die Durchführung der Planung realisierten Durchsatzverlust – hinsichtlich der vorteilhaften Anwendung einer dynamischen Rekonfiguration darstellt. Im Gegensatz dazu stellt die Güte des im Fehlerfall angepassten Verhaltens, d.h. im Hinblick auf ein durchsatzoptimiertes Verhalten (wie durch Anforderung A2 gefordert) der durch ein alternatives Verhalten realisierbare Durchsatz bzw. die sich dadurch ergebende Durchsatzreduktion im Vergleich zum fehlerfreien Betrieb, einen wesentlichen Einflussfaktor dar.

Abbildung 2.11 stellt diesen Sachverhalt visuell dar: Ein sofortiger Wartungsstopp und damit verbundene Zeitdauer für manuelle Eingriffe für Behebung bzw. Reparatur etwaiger gestörter oder defekter Komponenten (auch als mean time to repair, MTTR bezeichnet) ist insbesondere durch die manuellen Tätigkeiten zumeist wesentlich länger¹⁴ als die Zeitspanne einer

¹³Eine Änderung der Maschinen- und Anlagenmodelle sollte hierbei jedoch vermieden werden, da hierdurch etwaige Planungsergebnisse gegebenenfalls obsolet würden.

¹⁴An dieser Stelle werden Zeiten für den Wiederanlauf nicht näher betrachtet. Häufig verlängert dies jedoch die Zeitdauer des ungeplanten Wartungsstopp nochmals deutlich.

2. Eingrenzung des Betrachtungsbereichs und Anforderungsermittlung

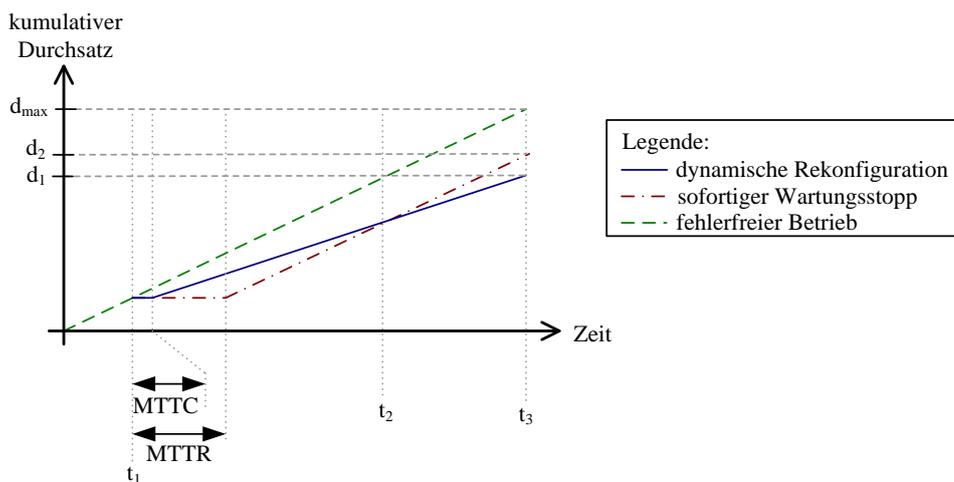


Abbildung 2.11.: Exemplarische Gegenüberstellung des realisierten Durchsatzes im Störfall mit unterschiedlichen Reaktionsstrategien (dynamischen Rekonfiguration, sofortigen Wartungsstopp) in Anlehnung an [LVH14a].

(semi-)automatischen Rekonfiguration durch eine Software (auch als mean time to configure, MTTC bezeichnet). Während jedoch durch eine dynamische Rekonfiguration ein alternatives, möglicher Weise im Hinblick auf den Durchsatz vermindertes Verhalten der Maschine bzw. Anlage realisiert wird, kann nach einem Wartungsstopp der (analog zu einem fehlerfreien Betrieb) optimale Betriebsablauf erneut realisiert werden. Diese Aspekte stellen dabei die wesentlichen Einflussfaktoren auf die Vorteilhaftigkeit und die Gewinnschwelle (Zeitpunkt t_2 in Abbildung 2.11) dar. Die Güte alternativer Verhalten, d.h. die Ergebnisse der Ablaufplanung, ist ausschließlich vom technischen System und dessen implementierten Operationen abhängig und liegt somit außerhalb des Einflussbereiches des Planungsverfahrens. Dementsprechend stellt die MTTC die einzige, durch das Planungsverfahren beeinflussbare Größe dar. Eine Planungszeit im Bereich mehrerer Minuten wäre verglichen mit einem (vergleichsweise langen) manuellen Eingriff für die Instandsetzung fehlerhafter Komponenten bereits vorteilhaft.

Auf Basis dieser Ergebnisse der Voruntersuchungen ergibt sich, dass Anforderung A3 im Hinblick auf das zeitliche Verhalten des Planungsverfahrens moderate Anforderungen stellt und kurze Zeitdauer im Bereich weniger Minuten bereits eine potentiell deutliche Verbesserung und damit Vorteilhaftigkeit für die eingangs in dieser Arbeit angeführten Anwendungsbeispiele darstellt. Dabei sollte jedoch insbesondere auch die Skalierbarkeit des Planungsverfahrens im Hinblick auf die Größe des Planungsgegenstandes, d.h. der gesamten Fertigungsanlage, Berücksichtigung finden.

2.2.4. Anforderung A4: Eignung des automatischen Planungsverfahrens für IEC 61131-3-basierte Steuerungssoftware

Wie bereits zuvor identifiziert, sind beide Anwendungsbeispiele aus Abschnitt 1.1 mit der Anlagensteuerung bzw. der Steuerungssoftware eng verzahnt. Um die Anwendbarkeit des zu ent-

wickelnden Konzepts in der industriellen Praxis zu gewährleisten, ist die Berücksichtigung existierender Standards für die Implementierung von Software zur Steuerung im Maschinen- und Anlagenbau erforderlich.

Wie in Abschnitt 2.1.4 beschrieben, werden für die Maschinen- und Anlagenautomatisierung häufig Speicherprogrammierbare Steuerungen eingesetzt. Obwohl bereits eine überarbeitete Version des IEC 61131 Standards mit Integration objekt-orientierter Programmierparadigmen existiert und bereits von einigen Unternehmen erprobt bzw. genutzt wird, ist diese jedoch noch nicht vollständig in der Industrie etabliert. Aus diesem Grund wird hier auf die aktuell am weitesten verbreitete, vollständig signal-orientierte Variante des IEC 61131-3 [IEC09] Standards Bezug genommen, welche den für die Programmierung von SPSen am weitesten verbreiteten Standard darstellt [Gro11].

Daraus ergibt sich die Notwendigkeit, dass ein geeignetes, automatisches Planungsverfahren für den Maschinen- und Anlagenbau die Konzepte der IEC 61131-3 berücksichtigt und eine Anbindung des Planungssystems in das Umfeld Speicherprogrammierbarer Steuerungen nach IEC 61131-3 erlaubt sowie Ergebnisse der Planung direkt in entsprechende Steuerungssoftware umgesetzt werden kann. Betrachtet man dabei die Charakteristika der beiden, dieser Arbeit zu Grunde gelegten Anwendungsbeispiele im Hinblick auf die Anbindung an die Steuerungssoftware der Feldebene, unterscheiden sich diese: Während es für Anwendungsbeispiel 1 (Assistenz für die Konfiguration von Anlagen) ausreichend ist, Steuerungscode zu generieren, welcher die durch einen Planungsalgorithmus bestimmte Abfolgen von Automatisierungsfunktionen implementiert, bedingt Anwendungsbeispiel 2 (dynamische Rekonfiguration) eine Anpassung des Verhaltens der Steuerungssoftware zu Laufzeit (wie dies bereits in der konzeptionellen Betrachtungen der beiden Anwendungsfälle in Abschnitt 1.1 identifiziert wurde).

Demzufolge ist es notwendig, dass Handlungspläne, welche durch das im Rahmen dieser Arbeit beschriebene Planungsverfahren automatisch bestimmt werden, durch IEC 61131 basierte Steuerungen umgesetzt werden können. Dabei ist es auch notwendig, dass diese Handlungspläne einerseits als statische (unveränderte) Software im Rahmen der in IEC 61131-3 definierten Sprachen umgesetzt werden können, aber auch eine konfigurierbare Implementierung realisiert werden kann, welche die Anpassung des durchzuführenden Handlungsplans durch externe Programme erlaubt.

2.2.5. Anforderung A5: Möglichkeit der Integration mit modellbasierten Entwicklungsansätzen

In beiden, im Rahmen von Abschnitt 1.1 beschriebenen, Anwendungsbeispielen wird eine grafische Benutzerschnittstelle für die Informationsdarstellung, aber auch Eingabe notwendiger Informationen für die Durchführung der gewünschten Funktionen des Anlagenkonfigurators bzw. Betriebs- und Störungssystems beschrieben. Die modellgetriebene Software- und Systementwicklung (engl. model-driven engineering, MDE) gewinnt in diesem Zusammenhang zusehends im Maschinen- und Anlagenbau an Bedeutung [Vya13, VHFST15]. In der Softwareentwicklung stellt die Unified Modeling Language (UML) die am meisten verbreitete Sprache dar [SBF07]. Um jedoch komplexe, technische Systeme unter Berücksichtigung von Software- und Hardwareaspekten zu spezifizieren, wird zusehends die Systems Modeling Language (SysML) eingesetzt und stellt mittlerweile ein etabliertes Mittel in der modellgetriebenen Softwareentwicklung des Maschinen-

2. Eingrenzung des Betrachtungsbereichs und Anforderungsermittlung

und Anlagenbaus dar [Vya13, VHDF⁺14, VHFST15]. Es konnte bereits gezeigt werden, dass modellgetriebene Entwicklungsansätze dabei die kognitive Komplexität reduzieren und folglich die Entwicklung im Maschinen- und Anlagenbau vereinfachen können [VH14].

Um die Akzeptanz gegenüber einem Verfahren zur automatischen Handlungsplanung innerhalb des Maschinen- und Anlagenbaus und dessen vereinfachte Nutzung zu erreichen, ist zu untersuchen, ob und wie eine (möglichst nahtlose) Integration in modellbasierte Ansätze zur Softwareentwicklung des Maschinen- und Anlagenbau möglich ist.

3. Stand der Wissenschaft und Technik

Im Zuge dieses Kapitels werden der aktuelle Stand der Wissenschaft und Technik verwandter Arbeiten im Hinblick auf die Zielsetzung dieser Dissertation untersucht.

Eingangs werden im ersten Teil dieses Kapitels – in Abschnitt 3.1 – eine Auswahl grundlegender Ansätze zur Formulierung von Planungsproblemen und der damit verbundenen Ableitung von Handlungsplänen untersucht. Es wird gezeigt, dass sich ein Planungsmodell – das letztendlich ein Transitionssystem darstellt – mittels unterschiedlicher Modellierungstechniken darstellen lässt. Ferner wird gezeigt, dass sich die Lösung des Planungsproblems letztendlich unabhängig von der Modellierungstechnik algorithmisch stets auf eine Suche in einem Graphen zurückführen lässt. Dieses durch einen Graphen repräsentierte Transitionssystem stellt somit die grundlegende Basis mit niedrigster Abstraktionsstufe für ein Planungsverfahren dar. Ferner wird durch die Diskussion unterschiedlicher Techniken zur Modellbildung und automatischer Planung gezeigt, dass durch Hintergrundwissen die Suche nach einer (optimalen) Lösung effizienter durchgeführt werden kann. Am Beispiel der linearen Programmierung wird ferner gezeigt, dass die Abbildung des Suchproblems auf einen numerischen, kontinuierlichen Suchraum – also eine kontinuierliche Approximation – positive Auswirkungen auf die Leistungsfähigkeit des Planungsalgorithmus hat.

Im zweiten Teil dieses Kapitels – in Abschnitt 3.2 – werden zwei grundlegende, disruptive, architektonische Ansätze zur Entwicklung und Implementierung von Steuerungssoftware diskutiert, die potentiell eine Adaptivitätssteigerung (wie in Abschnitt 1 gefordert) ermöglichen. In diesem Kontext werden insbesondere deren Anwendung in der Produktionstechnik sowie die Anwendung automatischer Planungstechniken adressiert. Ferner werden weitere Ansätze, welche automatische Planungsverfahren im Betrachtungsbereich einsetzen, beschrieben und diskutiert. Ziel dieses Abschnitts ist dabei einerseits, existierende Ansätze im Hinblick auf deren Adressierung bezüglich der im vorangegangenen Kapitel gestellten Anforderungen zu untersuchen und damit verbunden, der Mangel an einem geeigneten Planungsverfahren aufzuzeigen. Des Weiteren wird herausgearbeitet, dass sich architektonische Betrachtungen und die damit verbundene Anwendung automatischer Planungsverfahren letztendlich auf identischen Grundkonzepten basieren, welche sich wiederum auf die im ersten Teil dieses Kapitels diskutierte Planungstechniken zurückführen lassen.

3.1. Ansätze zur Formulierung und Lösung von Planungsproblemen

Es existieren eine Vielzahl unterschiedlicher Formalismen und Formulierungen, um ein Planungsproblem geeignet zu beschreiben. Im Verlaufe dieses Abschnitts werden drei wesentliche Klassen von Formulierungen eingeführt und beschrieben, die sowohl die formale Grundlage eines Großteil der existierenden Planungsverfahren als auch auf konzeptioneller Ebene die Grundlage des in dieser Arbeit beschriebenen Planungsverfahrens darstellen. Im Rahmen dieses Abschnitts wird dabei insbesondere untersucht, welche charakteristischen Merkmale etwaigen Formulierungen inhärent sind, wie sich diese auf die Algorithmik des Planungsverfahrens auswirken und letztendlich welche dieser Formalismen eine geeignete Grundlage für HiTraP-AT bilden können.

Im nachfolgenden Abschnitt 3.1.1 werden Formalismen auf Basis der mathematischen Logik beschrieben, welche historisch die Grundlage des Forschungsbereichs der automatischen Planung bilden. Anschließend werden zwei wesentliche Ansätze aus dem Bereich der sogenannten Cons-

3. Stand der Wissenschaft und Technik

traintprogrammierung in Abschnitt 3.1.2 diskutiert und deren Bezug zu der zuvor beschriebenen, logik-basierten Formulierung hergestellt. Im letzten Unterabschnitt 3.1.3 wird der Formalismus der Stellen-/Transitionsnetze (auch als Petrinetze bekannt) eingeführt und darauf aufbauende, theoretische Ansätze für die automatische Planung beschrieben.

3.1.1. Logik-basierte Planungsverfahren

Um ein (klassisches) Planungsproblem zu formulieren und letztendlich das Planungsproblem zu lösen, werden im Kontext des Forschungsbereichs der Künstlichen Intelligenz bereits seit langen sogenannte Aktions Sprachen (engl. action language) eingesetzt. Sie dienen der Beschreibung von Transitionssystemen auf Basis einer sogenannten Aktionssignatur¹⁵ [GL98], welche typischer Weise auf den Formalismen mathematischer Logik wie z.B. Aussagen- oder Prädikatenlogik basieren. In diesem Zusammenhang sind insbesondere drei bekannte Aktions Sprachen STRIPS, ADL und PDDL zu nennen. Neben ihrer textbasierten Syntax, welche die Beschreibung von Planungsproblemen ermöglicht, stehen diese Aktions Sprachen stellvertretend für unterschiedliche Beschreibungskonzepte – insbesondere im Hinblick auf das Planungsmodell, welches die verfügbaren, planerischen Möglichkeiten eines betrachteten Systems beschreiben. Im Folgenden werden diese drei Aktions Sprachen kurz eingeführt.

Der *Stanford Research Institute Problem Solver* (STRIPS) [FN72] stellt einen domänenunabhängigen Planer dar, welcher als Eingabe die gleichnamige Aktions Sprache verarbeitet. Die Sprache STRIPS basiert dabei vollständig auf einem mathematisch, logischen Formalismus – konkret der Aussagenlogik. Formal betrachtet, kann eine Instanz eines STRIPS Modell – d.h. ein konkretes Planungsmodell für ein betrachtetes System – in Anlehnung an Tom Bylander [Byl94] als 4-Tupel¹⁶ $\langle P, O, I, G \rangle$ aufgefasst werden, wobei P eine endliche Menge logischer Atome darstellt – hier als Bedingungen bezeichnet und die Gesamtmenge der logischen Atome innerhalb dieser Instanz von STRIPS darstellt. O stellt eine Menge von Operationen, $I \subseteq P$ eine Beschreibung des initialen Systemzustandes und G einem Zielzustand dar. STRIPS Modellen liegt dabei die sogenannte Closed-World Assumption¹⁷ zu Grunde. Zustände werden in STRIPS als Menge wahrer, logischer, atomarer Bedingungen aufgefasst. Aus diesem Grund wird der initiale Zustand als Teilmenge $I \in P$ logischer Atome formuliert, die zu Beginn als wahr angenommen werden. Operationen $o \in O$ stellen wiederum ein 4-Tupel $\langle o^+, o^-, o_+, o_- \rangle$ aus Mengen von Bedingungen dar. Eine Operation wird in STRIPS als ausführbar betrachtet, wenn die in der Menge $o^+ \subseteq P$ enthaltenen Bedingungen erfüllt (d.h. wahr bzw. positiv) sowie die in der Menge $o^- \subseteq P$ enthaltenen Bedingungen nicht erfüllt (d.h. unwahr bzw. negativ) sind. Die in den Mengen $o_+ \subseteq P$ und $o_- \subseteq P$ enthalten Bedingungen stellen die logischen Aussagen dar, die nach Ausführung der Operation $o \in O$ wahr bzw. unwahr sind. Jede Operation $o \in O$ stellt in STRIPS eine Zustandsabbildung $\text{PRE}_o \rightarrow \text{POST}_o$ dar. PRE_o ist die logische Konjunktion der

¹⁵Unter einer Signatur wird hier eine Menge nicht-logischer Symbolen verstanden, welche in der betrachteten Sprache zusätzlich zu logischen Symbolen hinzukommen sowie eine entsprechende Abbildungsvorschrift, welche diesen Symbolen eindeutig eine Arität zuordnet.

¹⁶Die Formalisierung wurde im Vergleich zu [Byl94] leicht angepasst, um die Analogie zur allgemeinen Beschreibung der automatischen Planung in Abschnitt 2.1.1 zu entsprechen.

¹⁷Die Closed-World Assumption (CWA) stellt eine Technik dar, die für logisches Schließen (engl. logic inference, logic reasoning) zum Einsatz kommt und ermöglicht, mit negativen Informationen umzugehen. Dabei wird angenommen, dass logische Atome als unwahr angenommen werden, wenn nicht explizit deren positiver Wahrheitsgehalt angegeben wird. Eine detaillierte Betrachtung des aussagenlogischen Schließens mit CWA ist in [CL94] gegeben.

positiven Vorbedingung o^+ und negativen Vorbedingung o^- wohingegen POST_o die Konjunktion der positiven (o_+) und negative (o_-) Nachbedingung einer Operation o darstellt. Häufig wird in diesem Zusammenhang auch von Effekten einer Operation gesprochen. Die Menge aller Operationen O definiert somit auch die in diesem Planungsmodell enthaltene Zustandsübergangsfunktion γ (vgl. Abschnitt 2.1.1). Der zu erreichende Zielzustand $G = \langle G_+, G_- \rangle$ wird in STRIPS als Konjunktion der positiven ($G_+ \subseteq P$) bzw. negativen ($G_- \subseteq P$) Menge an Bedingungen betrachtet. Ein Plan auf Basis von STRIPS stellt eine finite Abfolge (o_1, o_2, \dots, o_n) von Operationen dar, wobei die logische Ausführung des sequentiellen Plans (o_1, o_2, \dots, o_n) – d.h. die Anwendung der Ersetzungsregeln entsprechend der Vor- bzw. Nachbedingungen der Operationen (siehe [Byl94], S. 10) – mit Initialzustand I den Zielzustand G ergibt.

Die *Action Description Language* (ADL) wurde durch Edwin Pednault [Ped86, Ped94] als Gegenentwurf zu STRIPS entwickelt und ist Ausdruckstärker als STRIPS [PW92]. ADL setzt im Gegensatz zu STRIPS auf eine Open World Assumption¹⁸ und bietet zusätzliche Spracherweiterungen (verglichen mit STRIPS) wie die Möglichkeit negativer Literale und Disjunktionen. Die Beschreibung von Operationen bzw. Aktionen erfolgt dabei analog zu STRIPS. Zusätzlich bietet ADL die Möglichkeit sogenannte bedingungsbehaftete Effekte (engl. conditional effects) für Operationen anzugeben – d.h. (ein Teil der) Zusicherungen der Nachbedingung (logische Axiome) ist vom Wahrheitsgehalt einer zusätzlichen Bedingung abhängig. Hierdurch kann u.a. innerhalb der Definition von Effekten einer Operation in Abhängigkeit gegebener Zustände vor Operationsausführung (d.h. Vorbedingung) unterschieden werden. Im Gegensatz zu Nachbedingungen, wie sie beispielsweise bei STRIPS zum Einsatz kommen und lediglich eine spezifische, logische Wertebelegung der Wahrheitsvariablen (und damit eine Teilmenge aller möglichen Zustände) definiert, bekommt dadurch schlussendlich die Definition von Effekten einzelner Operationen eine gewisse Dynamik. Die algorithmische Komplexität der automatische Planung auf Basis von ADL Modellen ist dadurch jedoch komplexer als auf Basis von STRIPS [Ped89].

Im Zuge der Entwicklungen von STRIPS und ADL wurden eine Vielzahl weiterer Aktionsprachen entwickelt [GL98]. Um diesem Trend entgegenzuwirken und eine Vergleichbarkeit unterschiedlicher Planungsverfahren zu ermöglichen, wurde die *Planning Domain Definition Language* (PDDL) im Zuge der Vorbereitung des ersten internationalen Planungswettbewerbs¹⁹ [McD00, LKS⁺00] entwickelt. Seit der Einführung von PDDL im Jahre 1998 wurden eine Vielzahl unterschiedlicher Erweiterungen für unterschiedliche Zielstellungen und Anwendungen vorgeschlagen, um beispielsweise numerische Werte [FL03] (PDDL2.1) oder zusätzliche, weiche Zielsetzungen, die nicht notwendiger Weise erfüllt sein müssen [GL05] (PDDL3), darstellen zu können. Neben diesen deterministischen Versionen der PDDL wurden auch nicht-deterministische Erweiterungen entwickelt, wie z.B. PPDDL [YL04] und RDDDL [San10]. Eine Übersicht der Entwicklung von PDDL kann [CCO⁺12] entnommen werden. Im Folgenden werden nur ausgewählte Aspekte der Basisversion beschrieben, die für das weitere Verständnis dieser Arbeit hilfreich

¹⁸Im Gegensatz zur Closed World Assumption werden bei der Open World Assumption (OWA) negative Wahrheitsaussagen nicht automatisch getroffen, falls ein Sachverhalt unbekannt ist, sondern erfordern die explizite Zusicherung der Unwahrheit eines logischen Axioms.

¹⁹Der internationale Planungswettbewerb IPC wird alle zwei Jahre im Rahmen der Konferenz ICAPS („International Conference on Automated Planning and Scheduling“) ausgetragen. Eine Übersicht der bereits durchgeführten Wettbewerbe kann [CCO⁺12] entnommen werden. Ergebnisse des fünften IPC Wettbewerbs für deterministische Planungssysteme können [GHL⁺09] entnommen werden.

3. Stand der Wissenschaft und Technik

sind. Für eine detaillierte, formale Spezifikation sei an dieser Stelle auf die Dokumentation der entsprechenden PDDL Versionen verwiesen.

Die PDDL ist stark von STRIPS inspiriert, basiert jedoch auf Prädikatenlogik im Gegensatz zu der aussagenlogischen Grundlage von STRIPS. Entsprechend wird ein in PDDL kodiertes Planungsproblem durch eine Menge an Prädikaten formuliert. Die Beschreibung von Initial- und Zielzustand erfolgt analog zu den Konzepten von STRIPS. Wie in der Prädikatenlogik üblich, basiert ein PDDL Modell auf Objekten – d.h. relevanten Dingen innerhalb des betrachteten Systems und dessen Umwelt – über die mittels Prädikaten, Aussagen getroffen werden. Aktionen werden auf Basis von Vorbedingungen und Effekten beschrieben. In Anlehnung an ADL werden durch PDDL auch bedingungsbehaftete Effekte (engl. conditional effects) unterstützt.

Zur Lösung von Planungsproblemen mit einem Planungsmodell auf Basis der mathematischen Logik existieren eine Vielzahl unterschiedlicher Ansätze. Einen Überblick über alle existierenden Verfahren ist im Rahmen dieser Arbeit nicht zielführend. Daher sei an dieser Stelle für Details an entsprechende Literatur [GNT04, RN95, LaV06] verwiesen. Einige Konzepte, die für das Verständnis weiterer Ausführung in dieser Arbeit vorteilhaft sind, werden im Folgenden kurz erläutert. Algorithmisch kann eine Lösung von Planungsproblem allgemein stets als eine Suche innerhalb des durch das Planungsmodell beschriebenen (diskrete) Zustandsraumes verstanden werden. Dabei wird auf Basis des gegebenen (logischen) Planungsmodells der Zustandsraum nach und nach generiert, bis ein gegebener Zielzustand erreicht wird. Die formale Grundlage des Planungsmodells (die mathematische Logik) wird genutzt, um einen Zustandsraum aufzuspannen. Eine alternative Möglichkeit stellt die Betrachtung des Planungsproblems als Erfüllbarkeitsproblem (engl. Satisfiability, kurz SAT) – genauer als boolesches Erfüllbarkeitsproblem – dar [KS92]. Hierbei wird das Planungsproblem auf die Suche nach einer Belegung boolescher Variablen (daher boolesches SAT) zurückgeführt. Alle Bestandteile eines validen Plans werden, wie bereits oben beschrieben, als logische Axiome formuliert. Dies gilt sowohl für die Zustände des betrachteten Systems als auch für die Vor- und Nachbedingungen einzelner Aktionen, welche letztendlich die Dynamik des Systems beschreiben. Um die sich aus der Ausführung von Aktionen ergebende Dynamik in der mathematischen Logik zu formulieren, wird auf entsprechende logische Kalküle, wie Situationskalkül [MH69] (engl. situation calculus) oder Ereigniskalkül [KS89, Sha99] (engl. event calculus) zurückgegriffen. Letztendlich wird für die sich daraus ergebende Menge logischer Axiome eine Belegung der genutzten logischen Variablen gesucht, wodurch wiederum der Handlungsplan bestimmt ist. Voraussetzung für die Anwendung der auf SAT basierenden Verfahren ist zumeist die Angabe der Länge des gesuchten Handlungsplans, da dies für die Generierung des logischen Modells notwendig ist. Gleichzeitig stellt dies eine der deutlichen Nachteile der entsprechenden Verfahren dar. Eine weiterführende, detaillierte Beschreibung der Planung kann [KS92] sowie [Rei01] entnommen werden. Neben der Lösung klassischer Planungsprobleme zur Bestimmung sequentieller Handlungspläne existieren auch Ansätze, um parallele Pläne mittels Erfüllbarkeitsproblem zu bestimmen. Rintanen et al. [RHN06, RHN04] beschreiben beispielsweise eine Kodierung paralleler Pläne auf Basis eines aussagenlogischen Formalismus.

Eine Vielzahl realer Aufgaben sind hierarchisch organisiert. Diesen Sachverhalt versuchen sich Ansätze zur hierarchischen Handlungsplanung (engl. hierarchical task planning, kurz HTN) zu Nutze zu machen und erweitern das Planungsmodell um zusätzliche Informationen hinsichtlich einer Taxonomie von Aktionen. Pläne werden ebenfalls entsprechend hierarchisch organisiert. Hierdurch werden Gewinne hinsichtlich der Verarbeitungseffizienz des Algorithmus erwartet –

insbesondere durch den Rückgriff auf bereits vorab erstellte Pläne. Neben klassischen Planungsansätzen für die Bestimmung sequentieller Pläne mittels HTN, wie z.B. SIPE-2 [Wil99] oder SHOP2 [NAI⁺03], existieren auch Ansätze zur Generierung von Plänen mit partieller Ordnung, wie beispielsweise das Planungssystem HiPop [BBI⁺14].

3.1.2. Constraintprogrammierung

Die Constraintprogrammierung [RVBW06, Apt03] stellt, allgemein betrachtet, ein Paradigma dar, um auf Basis von Beziehungen zwischen Variablen, die in Form von Bedingungen formuliert sind, ein zu lösendes Problem zu beschreiben. Dabei beschreiben diese Bedingungen gewünschte Eigenschaften einer Lösung. Der Terminus Constraintprogrammierung ist dabei auf die Tatsache zurückzuführen, dass es sich dabei um eine Form der deklarativen Programmierung handelt. Die Art der genutzten Bedingungen innerhalb eines Constraintprogramms können unterschiedlich sein: Bedingungen welche üblicher Weise in Constraint-Satisfaction- bzw. Constraint-Optimization-Problemen verwendet werden (siehe hierzu nachfolgenden Unterabschnitt 3.1.2.1) oder Bedingungen in Form eines Linearen Programms (siehe Unterabschnitt 3.1.2.2). Constraintprogrammierung ist dabei eng mit dem zuvor beschriebenen Verfahren der mathematischen Logik verwandt. Einerseits findet der Formalismus der Logik innerhalb der Constraintprogrammierung Anwendung, um Beziehungen zwischen booleschen Variablen herzustellen. Andererseits können zuvor beschriebene Ansätze zur Planung auf Basis boolescher Erfüllbarkeitsprobleme als logische Constraintprogrammierung verstanden werden [JM94].

3.1.2.1. Constraint-Satisfaction-/Constraint-Optimization-Problem

Ein Constraint-Satisfaction-Problem (CSP) besteht aus einer Menge von Variablen mit gegebener Domäne. Die Randbedingungen sind mittels einer Menge von Bedingungen definiert. Gesucht wird nun im Rahmen der Lösung eines CSP eine Wertebelegung der Variablen, so dass dabei alle gegebenen Randbedingungen erfüllt sind. Ein Constraint-Optimization-Problem (COP) kann als Erweiterung eines CSP verstanden werden. Die Lösung eines COP ist die beste Variablenbelegung, welche durch eine Optimierungsfunktion gegeben ist und nicht wie beim CSP eine beliebige Lösung.

Formal kann ein Constraint-Satisfaction-Problem als ein 3-Tupel $\langle X, D, \Phi \rangle$ aufgefasst werden [Apt03, Dec03, RVBW06]. Ein CSP mit $n \in \mathbb{N}$ Variablen besteht aus einer Menge von Variablen $X = \{x_1, \dots, x_n\}$, einer Menge von Domänen $D = \{D_1, \dots, D_n\}$ – die Domänen D_i stellen dabei Mengen von Wertebelegungen der Variable x_i dar sowie einer Menge von $m \in \mathbb{N}$ Bedingungen $\Phi = \{\phi_1, \dots, \phi_m\}$. Als Konfiguration $c \in C = D_1 \times \dots \times D_n$ werden konkrete Wertebelegungen der Variablen $x \in X$ verstanden. Jede Bedingung wird allgemein als eine binäre Funktion aufgefasst, so dass $\phi(c) \in \{0, 1\}$. Eine Bedingung ist erfüllt, wenn $\phi(c) = 1$ gilt. Die Menge der korrekten Konfigurationen ist durch $S = \{c \in C \mid \forall \phi \in \Phi : \phi(c) = 1\} \subseteq C$ gegeben und repräsentiert somit alle Lösungen des CSP.

Ein COP ist ein 4-Tupel $\langle X, D, \Phi, B \rangle$. Es erweitert die Definition eines CSP $\langle X, D, \Phi \rangle$ um eine Bewertungsfunktion $B : S \rightarrow \mathbb{R}$, welche jeder Lösung $s \in S$ ein numerische Bewertung $B(s) \in \mathbb{R}$ zuordnet. Gesucht wird nun im Rahmen eines COP die Konfiguration $s \in S$, welche den – je nach Optimierungsproblem – minimalen (*min*) bzw. maximalen (*max*) Bewertungswert $B(s)$ aufweist. Die Lösung eines COP ist somit definiert als $\min B(s) \in S$ bzw. $\max B(s) \in S$.

3. Stand der Wissenschaft und Technik

Eine Erweiterung des zuvor eingeführten CSP Formalismus stellen sogenannte dynamische CSPs dar [MF90]. Dabei wird eine zusätzliche Menge von Constraints eingeführt (sog. activity constraints), welche es ermöglichen, Bedingungen anzugeben, unter denen bestimmte Variablen bei Bestimmung der Erfüllbarkeit berücksichtigt werden müssen. Diese Erweiterung kann jedoch, wie in [DK01] beschrieben, auf eine traditionelle CSP Formulierung zurückgeführt werden. In [DK01, Kam00] wird die Notation (dynamischer) CSPs für die Lösung von Planungsproblem detailliert beschrieben. Dabei wird gezeigt, dass ein Teil des Graphplan Algorithmus [BF97] als CSP formuliert werden kann. Auch van Beek und Chen [VBC99] nutzen die Formulierung des CSP für die Ableitung von Handlungsplänen ebenso wie Nguyen et al. [NKN02]. In allen Fällen werden die durch CSP gegebenen Möglichkeiten genutzt, kombinatorische Probleme mittels Bedingungen auf die Erfüllbarkeit von Variablenbelegungen in der zuvor eingeführten Form abzubilden. Zustände werden dabei als Variablen modelliert; Bedingungen werden für die Modellierung von Einschränkungen in der Ausführung von Aktionen, deren Effekten sowie weiteren Einschränkungen genutzt (Details siehe hierzu insbesondere [DK01, VBC99]). Gerevini und Serina [GS03] beschreiben ein Verfahren zur Kodierung eines Planungsverfahrens als CSP. Gegenstand der Planung stellen dabei die Handlungsplanung und Koordination verteilter, intelligenter Softwareagenten dar, wofür ebenfalls eine erweiterte STRIPS-Notation – genannt MA-STRIPS – vorgeschlagen wird (näheres hierzu siehe Abschnitt 3.2.1.2).

Letztendlich sind diese Ansätze auf die Bestimmung der Erfüllbarkeit eines Plans zurückzuführen, analog zu dem durch Kautz und Selman [KS92] vorgeschlagenen Ansatz zur Handlungsplanung durch logische Erfüllbarkeitsprüfung (siehe Kapitel 3.1.1). Eine Handlungsplanung auf Basis einer CSP Formulierung kann somit letztendlich ebenfalls auf eine Suche zurückgeführt werden, dessen Suchraum analog zu anderen Planungsverfahren durch die Zustandsvariablen aufgespannt wird. Für die Lösung von Constraint-Satisfaction-/Constraint-Optimization-Problemen existieren eine Vielzahl von Algorithmen. Eine kurze Einführung in entsprechende Algorithmen ist beispielsweise in [SFJ00] gegeben; ausführliche Beschreibungen und Diskussionen können [Apt03, Dec03, RVBW06] entnommen werden. Zusammenfassend lässt sich sagen, dass auf Grund der allgemeinen Formulierung von CSP/COP eine algorithmische Verbesserung des Suchalgorithmus neben Verfahren wie Backtracking²⁰ und Pruning²¹, lediglich durch geeignete Heuristiken erfolgen kann, da kaum Informationen über etwaige Topologien oder Ordnungen des Gesamtsuchraums verfügbar sind.

3.1.2.2. Lineare Programmierung

Die Lineare Programmierung (LP) ist eine mathematische Methode, um auf Basis linearer Abhängigkeiten zwischen numerischen Variablen, gültige Lösungen des entsprechenden, linearen (Un-)Gleichungssystems zu identifizieren bzw. zu berechnen. Die Selektion konkreter (bester) Lösungen im Lösungsraum erfolgt auf Basis einer zu optimierenden, linearen Bewertungsfunktion. Häufig wird in diesem Zusammenhang auch von linearer Optimierung bzw. linearen Optimierungsproblemen gesprochen.

²⁰Backtracking stellt die grundlegende Methode für eine vollständige Suche zur Lösung von Constraint-Satisfaction-Problemen dar [FM06]

²¹Pruning wird ein Mittel zur Verarbeitung von algorithmischer Baumstrukturen genannt, bei dem Äste des Suchbaumes möglichst frühzeitig „abgeschnitten“ werden, wenn ein weiteres, exploratives Aufspannen des Baumes als nicht mehr zielführend erachtet werden kann [BA97]

3.1. Ansätze zur Formulierung und Lösung von Planungsproblemen

Analog zur Definition von CSP/COP (siehe Kapitel 3.1.2.1) wird im Folgenden eine formale Definition eines linearen Optimierungsproblems gegeben. Gesucht wird im Rahmen des linearen Optimierungsproblems eine Wertebelegung der Variablen $x \in X$. Die Anzahl gesuchter Variablen ist dabei durch $|X| \in \mathbb{N}$ gegeben; dabei ist allen Variablen die selbe, numerische Definitionsmenge D zugeordnet. In Abhängigkeit der Art des Linearen Programms ergeben sich die entsprechenden, numerischen Definitionsmengen, z.B. reell ($D = \mathbb{R}$) oder ganzzahlig ($D = \mathbb{N}$). Verschiedene Arten der linearen Optimierung – gegeben durch verschiedene Definitionsmengen der Variablen – werden später in diesem Abschnitt noch näher adressiert. Der Lösungsraum des Linearen Programms ist durch $m \in \mathbb{N}$ lineare (Un-)Gleichungen gegeben. In Matrixnotation können die linearen (Un-)Gleichungen, welche den Lösungsraum bestimmen, in Normalform durch $Ax \leq b$ beschrieben werden. Hierbei stellt $A \in \mathbb{R}^{m,|X|}$ eine gegebene, bekannte Koeffizientenmatrix dar, $b \in \mathbb{R}^m$ einen gegebenen, bekannten Koeffizientenvektor sowie $x = (x_1, \dots, x_n)$ einen Vektor, der durch die Variablen $x \in X$ als Komponenten des Vektors gebildet wird. Hieraus ergibt sich für $n \in \mathbb{N}$ Variablen und $m \in \mathbb{N}$ (Un-)Gleichungen ein (Un-)Gleichungssystem der Form:

$$\begin{array}{rccccl} \phi_1 : & a_{11}x_1 & + \dots & + a_{1n}x_n & \leq b_1 \\ \phi_2 : & a_{21}x_1 & + \dots & + a_{2n}x_n & \leq b_2 \\ \vdots & \vdots & & \vdots & \vdots \\ \phi_m : & a_{m1}x_1 & + \dots & + a_{mn}x_n & \leq b_m \end{array}$$

Jede Gleichung stellt dabei im Sinne der Definition des CSP/COP eine numerische, lineare Konkretisierung der allgemeinen Bedingung $\phi \in \Phi$ dar. Die Menge aller Wertebelegungen kann analog zur Definition des CSP/COP in Kapitel 3.1.2.1 als Konfiguration $c \in C = D^{|X|}$ verstanden werden. Im Rahmen des linearen Optimierungsproblems werden Elemente $s \in S$ der Lösungsmenge gesucht. Die Menge $S \subseteq C$ aller Lösungen wird durch alle, das (Un-)Gleichungssystem erfüllende Variablenbelegung für den Variablenvektor x im Kontext der Matrixnotation definiert, wobei für die Variablen x_i im Gleichungssystem die entsprechende Wertebelegung c_i eingesetzt wird, welche in der Konfiguration $c = (c_1, \dots, c_n) \in C$ gegeben ist. Im Kontext der CSP/COP Definitionen können die Variablen auch als Funktionsdefinition der Bedingung Φ aufgefasst werden, so dass $\phi_i(c) = 1 \Leftrightarrow a_{i1}c_1 + \dots + a_{in}c_n \leq b_i$. Die Lösungsmenge kann dadurch auch konkretisiert definiert werden als $S = \{c \in C \mid \sum_{i=1}^m \phi_i(c) = m\}$. Die Optimierung der Lösung (also die Selektion eines optimalen Elements aus der Lösungsmenge S) erfolgt im Rahmen der Linearen Programmierung durch eine sogenannte Zielfunktion, die im Folgenden analog zur Formulierung des CSP/COP auch Bewertungsfunktion genannt wird. Die Bewertungsfunktion $B : S \rightarrow \mathbb{R}$ stellt dabei eine lineare Funktion dar, die in der Form²² $z^T x = z_1x_1 + \dots + z_nx_n$ gegeben ist. Die Auswertung der Bewertungsfunktion erfolgt durch Belegung der Variablen x_i mit der in der Konfiguration $(s_1, \dots, s_n) \in S$ gegebenen Wertebelegung, so dass $B(s) = z^T s$ die Bewertung einer Konfiguration $s \in S$ darstellt. Gesucht wird nun die Konfiguration $s \in S$ mit dem minimalen oder maximalen Bewertungswert $B(s)$ – analog zur CSP/COP Definition $\min B(s)$ bzw. $\max B(s)$. In der für die Lineare Programmierung übliche Notation in Normalform ist ein Maximierungsproblem bestimmt durch $\max\{z^T s \mid As \leq b \wedge s \geq 0\}$; analog für ein Minimierungsproblem durch $\min\{z^T x \mid Ax \leq b \wedge x \geq 0\}$. Ein Minimierungsproblem lässt sich stets in ein Maximierungsproblem

²²Die in der Literatur übliche Bezeichnung c des Koeffizientenvektors der Zielfunktion wird im Folgenden angepasst, um der eingeführten Konvention der Konfiguration $c \in C$ zu entsprechen.

3. Stand der Wissenschaft und Technik

durch Vorzeichenwechsel des Zielvektors überführen. Im Rahmen der linearen Optimierung wird ferner nach nichtnegativer Wertebelegung der Variablen $x \in X$ gesucht; somit gilt für die Komponenten der Lösungskonfigurationen $s_i \geq 0$; vereinfacht wird im Folgenden $s \geq 0$ geschrieben. Es gilt somit $\max\{z^T s \mid As \leq b \wedge s \geq 0\} = \min\{[(-1)z]^T s \mid As \leq b \wedge s \geq 0\}$. Daher wird im Folgenden ausschließlich von Maximierungsproblemen gesprochen. Analog zur Definition kann ein lineares Optimierungsproblem als spezielle Form eines Constraint-Optimization-Problems aufgefasst werden und somit als 4-Tupel $\langle X, D, \Phi, B \rangle$ unter obigen Konkretisierungen/Einschränkung der Variablenmenge X , der Definitionsmenge D sowie der Bedingungsmenge Φ und der Bewertungsfunktion B beschrieben werden.

Anhand dieser Formalisierung ist erkennbar, dass die Lineare Programmierung eine Spezialisierung des Constraint-Satisfaction bzw. Constraint-Optimization-Problems für numerische Variablen darstellt. Analog zu Bedingungen des CSP/COP müssen in einem Linearen Programm die Menge der (Un-)Gleichungen erfüllt sein. Man unterscheidet unterschiedliche Klassen der Linearen Programmierung in Abhängigkeit der Definitionsmenge der Variablen. Spricht man von Linearer Programmierung im allgemeinen, wird von reellen Zahlen als Variablen ausgegangen, d.h. $D = \mathbb{R}$. Sind alle Variablen ganzzahlig (also $D = \mathbb{N}$) wird von ganzzahliger Linearer Programmierung (engl. integer linear programming, ILP) gesprochen und stellt einen Spezialfall der allgemeinen Linearen Programmierung dar, da $\mathbb{N}_0 \subseteq \mathbb{R}$ gilt. Ein weitere Spezialisierung stellt die binäre, ganzzahlige Lineare Programmierung (engl. binary integer programming, BIP) dar, bei der Variablen ausschließlich die ganzzahligen Werte Null oder Eins annehmen können und folglich $D = \{0, 1\} \subseteq \mathbb{N}_0$ gilt. Ein BIP kann ferner als Generalisierung des logischen Erfüllbarkeitsproblems betrachtet werden [Hoo88]. Wird nur für einen Teil der Variablen eine ganzzahlige Definitionsmenge gefordert, spricht man von gemischt-ganzzahliger Programmierung (engl. mixed integer linear programming, MILP).

Geometrisch betrachtet definieren lineare Ungleichungssysteme einen Polyeder, welcher alle korrekten Lösungen umfasst. Die Dimensionen des Raumes sind durch die Anzahl der Variablen gegeben. Sind Gleichungen im Linearen Programm enthalten, wird die Fläche des Polyeders entsprechend der in der Gleichung enthaltenen Variablen hinsichtlich seiner Dimension geebnet. Für die algorithmische Lösung linearer Optimierungsprobleme existieren einige Algorithmen, wie z.B. der Simplex Algorithmus [Dan90], das innere-Punkt-Verfahren [GT96] oder die Ellipsoidmethode [BGT81, GLS81]. Eine Übersicht der Algorithmen kann beispielsweise [Sie01] entnommen werden. Für unterschiedliche Probleme zeigen verschiedene Algorithmen die beste Leistungsfähigkeit, weshalb eine generelle Aussage über die Leistungsfähigkeit der Algorithmen aus praktischer Sicht nicht eindeutig getroffen werden kann [Bix02]. Um dieser Problematik entgegenzuwirken, wurde das MIPLIP Projekt initialisiert, welches eine Gegenüberstellung unterschiedlicher Implementierungen für spezifische Problemformulierungen in Form einer Art Benchmark ermöglicht [KAA⁺11]. Alle Algorithmen greifen auf die durch die Definitionsmenge der Variablen gegebenen Rauminformationen für eine zielgerichtete Suche nach einer (optimalen) Lösung zurück. Eine effiziente, algorithmische Lösung ist für allgemeine Lineare Programme mit reellen Variablen möglich. Die algorithmische Komplexität der Lösung ganzzahliger Linearer Programme oder gar binärer, ganzzahliger Linearer Programme ist signifikant höher. Dieser Sachverhalt lässt sich auch dadurch verdeutlichen, dass letztendlich binäre, ganzzahlige Lineare Programme auf das logische Erfüllbarkeitsproblem zurückzuführen sind. Können ganzzahlige Lineare Programme auf

ein allgemeines Lineares Programm zurückgeführt werden (sog. Relaxierung), ist eine effiziente Lösung komplexer ganzzahliger Probleme möglich.

Die Lineare Programmierung wird häufig im Kontext des Forschungsgebietes "Operations Research" eingesetzt, findet jedoch auch in anderen Bereichen Anwendung, wie z.B. für Scheduling oder Wegeplanung. Einige graph-basierten Problemformulierungen können als Lineares Programm formuliert und gelöst werden, wie z.B. die Prüfung der Verklemmungsfreiheit auf Basis von S-/T-Netzen [STC98] oder die Verifikation der Lebendigkeitseigenschaften endlicher Automaten [CA95].

Es wurde bereits in frühen Arbeiten gezeigt, dass das logische Erfüllbarkeitsproblem als ganzzahliges Lineares Programm formuliert werden kann [BJL86, Hoo88]. Eine Reformulierung des Planungsproblems als logisches Erfüllbarkeitsproblem wurde ebenfalls, z.B. durch Kautz et al. [KS92, KMS96] oder Blum und Furst [BF97] bereits vorgeschlagen (siehe hierzu auch Abschnitt 3.1.1). Aus diesem Grund liegt auch die Anwendung der Linearen Programmierung für die automatische Planung nahe und wurde bereits in einigen Planungssystemen eingesetzt. Vossen et al. [VBLN99] nutzt eine solche Reformulierung des Planungsproblems als Erfüllbarkeitsproblem und dessen Übersetzung in ein ILP für die Lösung von Planungsproblemen. Ferner zeigen die von ihnen durchgeführten Experimente, dass der Einsatz von LP Techniken für die automatische Planung hinsichtlich der Leistungsfähigkeit vorteilhaft sein können. Ferner identifizieren Ansätze auf Basis der Erfüllbarkeit einen korrekten, jedoch nicht unbedingt optimalen, Plan. Klassische Planungsverfahren adressieren dabei – wie bereits beschrieben – die automatische Bestimmung eines korrekten Handlungsplans, lassen jedoch die Qualität des Planes (z.B. hinsichtlich dessen Länge) zumeist außer Acht. Aus diesem Grund schlagen van den Briel und Kambhampati [BK05] eine Erweiterung und damit verbundene leichte Anpassung des von Vossen et al. vorgeschlagenen ILP Kodierungsansatzes zur Bestimmung optimaler Pläne vor, wobei als Optimalitätskriterium die Planlänge verwendet wird. Die vorgeschlagenen Anpassungen des Linearen Programms zusammen mit der Erweiterung um ein Optimierungsproblem weisen in durchgeführten Versuchen bessere Leistungswerte auf, als der ursprüngliche Kodierungsansatz von Vossen et al. Eine Erweiterung des klassischen Planungsproblems um ein Optimierungsproblem zur automatischen Ableitung des kürzesten, korrekten Plans auf Basis der Linearen Programmierung – genauer ILP – wird auch von Kautz und Walser [KW99] vorgeschlagen. Die Ergebnisse der durchgeführten Experimente zeigen ebenfalls, dass ILP eine effiziente Möglichkeit zur Lösung klassischer Planungsprobleme ist, welche zusätzlich durch die Erweiterung um ein Optimierungskriterium „bessere“ Lösungen als klassische Planungssysteme bestimmen kann. Neben der – wie in der klassischen Planung üblich, domänenunabhängigen – Kodierung des Planungsproblems bietet die Anwendung der Linearen Programmierung ferner die Möglichkeit, zusätzliches domänenspezifisches Wissen deklarativ nahtlos zu integrieren. Bockmayr und Dimopoulos [BD00] verfolgen diesen Ansatz und zeigen, dass hierdurch Leistungsgewinne des Planungsalgorithmus erzielt werden können.

Neben diesen Ansätzen, welche die Planung als Erfüllbarkeitsproblem betrachten, verfolgen Silva et al. [SCK00] einen alternativen Ansatz. Sie nutzen eine Übersetzung des, der Planung zu Grunde gelegten Petrinetzes zur Bestimmung eines optimalen Handlungsplans. Hierbei wird die Schaltsemantik des Petrinetzes mittels Variablen und Gleichungen repräsentiert (nähere Details siehe Abschnitt 3.1.3).

3. Stand der Wissenschaft und Technik

Für eine Vielzahl praktischer Probleme – wie auch die Handlungsplanung für die Steuerungssoftware im Maschinen- und Anlagenbau – stellt sich die klassische Formulierung eines Planungsproblems als nicht ausreichend dar. Eine Erweiterung der klassischen Planung um Optimierungskriterien ist mittels Linearer Programmierung effizient möglich [KW99, VDBBKV07, VDBVK05]. In der Automatisierung von Maschinen und Anlagen findet die Lineare Programmierung beispielsweise in der Entwicklung von Regelungssystemen Anwendung. Auch im Bereich der Produktionsfeinplanung wird die Lineare Programmierung in Form von Auftragsscheduling bzw. Auftragssequenzierung eingesetzt. Existierende Ansätze können dabei insbesondere die Anforderungen Anforderung A1.2 und Anforderung A1.3 nicht gleichzeitig gemeinsam geeignet adressieren, da es sich bei den Verfahren um zielbasierte Planungsverfahren handelt. Geeignete Ansätze zur automatischen Planung für die Ableitung von Handlungssequenzen Speicherprogrammierbarer Steuerungen basierend auf der Linearen Programmierung existieren – soweit bekannt – (noch) nicht.

3.1.3. Stellen-/Transitionsnetze

Den Ursprung hat das Konzept der Petrinetze in der Dissertation von Carl Adam Petri, dessen Ziel es war, "möglichst viele Erscheinungen bei der Informationsübertragung und Informationswandlung in einheitlicher und exakter Weise zu beschreiben" [Pet62]. Seit dieser Arbeit wurde das Konzept der Petrinetze in einer Vielzahl von Arbeiten theoretisch untersucht und weiterentwickelt. Ferner wurden auf Basis des Petrinetzkonzeptes eine Vielzahl unterschiedlicher Anwendungen entwickelt. Der Begriff des Petrinetzes steht heute für unterschiedliche Klassen netzbasierter Modelle (siehe [DJ01]). Im Folgenden werden hauptsächlich sogenannte Stellen-/Transitionsnetze (S/T-Netze) betrachtet. Diese stellen eine fundamentale Klasse der Petrinetze dar, die als natürliche Erweiterung des ursprünglichen Formalismus verstanden werden kann. Übersichten über weitere Klassen von Petrinetzen können beispielsweise [DJ01, PE01] entnommen werden.

Ein *Stellen-/Transitions-Netz* (S/T-Netz) $N = \langle S, T, F, W \rangle$ ist ein Tupel aus einer Menge von Stellen S , einer Menge von Transitionen T mit $T \cap S = \emptyset$ und einer Menge F von gerichteten Kanten (auch Flussrelation genannt) mit $F \subseteq (S \times T) \cup (T \times S)$. Da die Mengen T und S disjunkt sind gilt hierbei auch $F \cap (S \times S) = F \cap (T \times T) = \emptyset$. Das Tripel $\langle S, T, F \rangle$ bildet dabei einen bipartiten, gerichteten Graphen mit den Stellen und Transitionen als Knoten und der Menge der Flussrelationen als gerichtete Kanten. Ferner ist eine Gewichtungsfunktion $W : (F \times T) \cup (T \times F) \rightarrow \mathbb{N}$ gegeben. Die Gewichtungsfunktion ist auf Elemente der Flussrelationsmenge $f \in F$ sowie auf Elemente $(x, y) \in [(S \times T) \cup (T \times F)] \setminus F$ definiert, wobei für diese Elemente stets $W(x, y) = 0$ gilt.

Der Vorbereich $\bullet t$ einer Transition $t \in T$ wird durch alle Stellen gebildet, für die eine Flussrelation $(s, t) \in F$ existiert. Folglich ist der Vorbereich einer Transition durch $\bullet t = \{s \in S \mid \exists t \in T : (s, t) \in F\}$ definiert. Der Nachbereich t^\bullet einer Transition $t \in T$ wird analog durch alle Stellen gebildet, für die eine Flussrelation $(t, s) \in F$ existiert, d.h. $t^\bullet = \{s \in S \mid \exists t \in T : (t, s) \in F\}$. Im Folgenden wird davon ausgegangen, dass jede Transitionen einen nichtleeren, endlichen Vor- und Nachbereich hat ($\bullet t \neq \emptyset$ und $t^\bullet \neq \emptyset$). Entsprechende Netze werden auch als stellenberandet bezeichnet. Eine Markierung eines S/T-Netzes $N = \langle S, T, F, W \rangle$ ist eine Funktion $m : S \rightarrow \mathbb{N}$. Die

Funktion m bildet eine Multimenge²³ \mathbb{N}_0^S über die Menge der Stellen S . Ein markiertes S/T-Netz ist ein Tupel $\langle N, m \rangle$ wobei N ein S/T-Netz ist und m eine Markierung (auch Anfangsmarkierung genannt).

Für S/T-Netze wurden bereits Konzepte der Hierarchisierung wissenschaftlich ausgiebig untersucht. Hierarchisierung stellt ausgehend von einer gegebenen Modellierungsebene entweder eine Abstraktion oder Verfeinerung dar. Für beiden Methoden – Abstraktion und Verfeinerung – wurden S/T-Netze bereits in der Form angewandt, dass wichtige Verhaltenseigenschaften wie Lebendigkeit oder Beschränktheit oder Lebendigkeit bei der Hierarchisierung erhalten bleiben [Val79, SM83]. Die algorithmische Entscheidbarkeit des Nachweises dieser Verhaltenseigenschaften ist für hierarchische S/T-Netze jedoch nur teilweise gegeben [KB14]. Eine stellenbasierte Hierarchisierung von S/T-Netzen folgt dem sog. „Netze in Netze“ Konzept [HKB12], d.h. jede Hierarchieebene stellt ein syntaktisch vollständiges (stellenberandetes) S/T-Netz dar. Neben der stellenbasierten Hierarchisierung kann eine Abstraktion bzw. Verfeinerung von Transitionen durchgeführt werden. Eine transitionsbasierte Hierarchisierung resultiert in transitionsberandeten Netzen, wie z.B. in [PS05] beschrieben. Formale Definitionen hierarchischer S/T-Netze können beispielsweise [Feh93, HKB12] entnommen werden.

Die charakteristische Eigenschaft von S/T-Netzen zur Beschreibung diskreten Verhaltens findet in einer Vielzahl unterschiedlicher Bereiche zur Darstellung aber auch auf Grund der formalen Grundlage zur Verarbeitung – d.h. insbesondere auch für die automatische Handlungsplanung. So schlagen Ziparo et al. [ZI06, ZIL⁺11] beispielsweise die Anwendung von S/T-Netzen für die Verhaltensbeschreibung und Koordination autonomer Roboter vor – d.h. für die Darstellung und Ausführung von Handlungsplänen. Aber auch im Kontext unterschiedlicher, neuer Softwarearchitekturen wie der dienstorientierten Architektur und Multiagentensystemen finden S/T-Netze entsprechend Anwendung (siehe hierzu die entsprechenden Abschnitte 3.2.1.1 bzw. 3.2.1.2). Die automatische Handlungsplanung auf Basis von S/T-Netzen ist dabei von der zu Grunde gelegten Schaltsemantik abhängig. Im Folgenden werden zwei wesentliche Schaltsemantiken von S/T-Netzen eingeführt, wobei teils auf die in [Mau06, PE01] genutzten Formalisierungen zurückgegriffen wird. Auch wird die Anwendung der jeweiligen Schaltsemantik für die automatische Planung in den folgenden Abschnitten beschrieben.

3.1.3.1. Sequentielle Schaltsemantik und Erreichbarkeitsgraphen

Sei $\langle N, m \rangle$ ein markiertes S/T-Netz $N = \langle S, T, F, W \rangle$ und der initialen Markierung m . Eine Transition $t \in T$ des markierten S/T-Netzes gilt als aktiviert in einer Markierung m , falls $\forall p \in \bullet t : m(p) \geq W(p, t)$. Im Folgenden wird die Notation $t[m$ verwendet, um auszudrücken, dass eine Transition t in der Markierung m aktiviert ist. Eine Transition $t \in T$ kann in einer Markierung m schalten, falls diese aktiviert ist (also $t[m$ gilt). Das Schalten einer Transition t in der Markierung m führt zu einer Folgemarkierung m' , wobei gilt: $\forall s \in S : m'(s) = m(s) - W(s, t) + W(t, s)$. Um auszudrücken, dass eine Transition $t \in T$ in der Markierung m aktiviert ist und dessen Schalten zur Folgemarkierung m' führt, wird im Folgenden auch die Notation $m \xrightarrow{t} m'$ ($t \in T$,

²³Eine Multimenge stellt eine Menge dar, in der Elemente mehrfach enthalten sein können. Formal kann eine Multimenge über eine Menge X als Abbildung $\# : X \rightarrow \mathbb{N}_0$ betrachtet werden, wobei jedem Element $x \in X$ eine zugehörige Anzahl $\#(x) \in \mathbb{N}_0$ zugeordnet ist. Ein Element $x \in X$ ist innerhalb einer durch m beschriebenen Multimenge $\#(x)$ -mal enthalten. Die Menge aller Multimengen über X wird durch die Potenzmenge \mathbb{N}_0^X beschrieben.

3. Stand der Wissenschaft und Technik

$m, m' \in \mathbb{N}^S$) verwendet. Eine sequentielle Schaltregel eines S/T-Netzes lässt sich somit auch als Transitionssystem $A = \langle \mathbb{N}^S, T, \Sigma \rangle$ mit $\Sigma \subseteq \mathbb{N}^S \times T \times \mathbb{N}^S$ auffassen. Die Kantenbeschriftungen sind dabei die Transitionen; die resultierenden Folgen von Transitionen $\sigma = t_1 \dots t_n$ mit $n \in \mathbb{N}$, $t_1, \dots, t_n \in T$ und $m \xrightarrow{\sigma} m'$ stellen Schaltfolgen des Transitionssystem A dar. Man spricht dann von in m aktivierten Schaltsequenzen mit der Folgemarkierung m' . Zu beachten ist dabei, dass der Erreichbarkeitsgraph eines markierten S/T-Netzes $\langle N, m \rangle$ abhängig von den Eigenschaften des S/T-Netzes endlich aber auch nicht-endlich sein kann.

Ein Erreichbarkeitsgraph $RG(N)$ eines markierten S/T-Netzes $\langle N, m_0 \rangle$ ist ein Transitionssystem $RG(N) = \langle R(N, m_0), T, B(N) \rangle$, mit der Menge aller erreichbaren Markierungen $R(N, m) = \{m' \mid \exists \sigma : m \xrightarrow{\sigma} m'\} \subseteq \mathbb{N}^S$ von $\langle N, m_0 \rangle$ als Knotenmenge, den Transitionen $t \in T$ als Kantenlabel und $B(N) = \{ \langle m, t, m' \rangle \mid m, m' \in B(N, m_0), t \in T, m \xrightarrow{t} m' \} \subseteq \mathbb{N}^S \times T \times \mathbb{N}^S$ als beschriftete Kanten. Der Erreichbarkeitsgraph $RG(N)$ eines markierten S/T-Netzes $\langle N, m \rangle$ stellt dabei alle Schaltsequenzen bezüglich einer sequentiellen Schaltsemantik des S/T-Netzes – d.h. sequentiellen Schaltsequenzen – dar, und somit auch alle durch sequentielles Schalten von Transitionen erreichbare Zustände des S/T-Netzes, falls die Markenbelegung m als Zustand betrachtet wird. Der Erreichbarkeitsgraph kann daher beispielsweise für die Analyse von Systemverhalten genutzt werden²⁴. Dabei steht die automatische Prüfung von unerwünschtem Systemverhalten (repräsentiert durch das Erreichen eines unerwünschten Systemzustandes, d.h. Markenbelegung) im Vordergrund [BRV04]. Edelkamp et al. [EJ06, EKS11] schlagen eine entgegengesetzte Herangehensweise vor: Die Fragestellungen der automatischen Modellprüfung wird auf die Formulierung des klassischen Planungsproblems zurückgeführt. Das Feuern von Transitionen in S/T-Netze mit sequentieller Schaltsemantik wird dazu als Ausführung einer Aktion entsprechend dem Konzept der klassischen Planung (analog der Problemkodierung in PDDL) interpretiert. Wie durch Edelkamp et al. beschrieben, können dadurch einige Fragestellungen der automatischen Modellprüfung, die sich in Form eines klassischen Planungsproblems (d.h. insbesondere die zielbasierte Planung) darstellen lassen, gelöst werden.

Silva et al. [SCK00, STC98] schlagen mit Petriplan einen auf Basis der Erreichbarkeitsanalyse von S/T-Netzen basierenden Ansatz vor. Dabei wird die klassische Formulierung des Planungsproblems mittels Start- und Zielzuständen auf S/T-Netze übertragen. Ist ein gegebener Zielzustand (d.h. eine Markenbelegung) ausgehend von der initialen Markenbelegung erreichbar, stellt die entsprechende Schaltsequenz zur Erreichung des Zielzustandes den gesuchten Plan dar. Zur Realisierung von Petriplan greifen Silva et al. nicht auf eine explizite Bestimmung des Erreichbarkeitsgraphen zurück, sondern nutzen eine Formulierung des sequentiellen Schaltsemantik als lineares Programm. In Schreiner et al. [SCSK12, SSC⁺13] wird ein hierzu alternativer Ansatz zur Kodierung klassischer Planungsprobleme mittels S/T-Netzen vorgeschlagen. Schreiner et al. schlagen dabei eine Übersetzung der in Graphplan genutzten Konzepte vor und greifen für die Lösung des Planungsproblems auf eine Formulierung als Erfüllbarkeitsproblem zurück.

3.1.3.2. Nebenläufige Schaltsemantik und Kausalnetze

Die Restriktion der sequentielle Schaltsemantik – lediglich eine Transition kann zu einem diskreten Zeitpunkt feuern – wird durch die sogenannte nebenläufige Schaltsemantik von S/T-Netzen

²⁴Es sei an dieser Stelle angemerkt, dass der Erreichbarkeitsgraph auf Grund seiner (ggf. unendlichen) Größe häufig nicht vollständig, sondern gezielt (partiell) generiert wird.

aufgehoben. Die Gewichtungsfunktion wird im Folgenden auf Paare von Elementen $(a, b) \in [(S \times T) \cup (T \times S)] \setminus F$ durch $W(a, b) = 0$ erweitert. Im Zuge einer nebenläufigen Schaltregel ist ein (nebenläufiger) Schritt eine Multimenge von Transitionen $s \in \mathbb{N}_0^T$. Ein nebenläufiger Schritt $s \in \mathbb{N}_0^T$ gilt in einer Markierung $m \in \mathbb{N}_0^S$ als aktiviert, falls $\forall p \in S : m(p) \geq \sum_{t \in T} W(p, t) \cdot s(t)$. Das Schalten eines nebenläufigen Schritts $s \in \mathbb{N}_0^T$ in einer Markierung m führt zu einer Folgemarkierung m' für die gilt: $\forall p \in S : m'(p) = m(p) + \sum_{t \in T} (W(t, p) - W(p, t)) \cdot s(t)$. Analog zur sequentiellen Schaltregel wird durch $m \xrightarrow{s} m'$ ausgedrückt, dass ein Schalten des nebenläufigen Schritts $s \in \mathbb{N}_0^T$ in der Markierung $m \in \mathbb{N}_0^S$ die Markierung $m' \in \mathbb{N}_0^S$ zur Folge hat. Dementsprechend kann ein markiertes S/T-Netz $\langle N, m \rangle$ mit nebenläufiger Schaltregel durch ein Transitionssystem $A = \langle \mathbb{N}_0^S, \mathbb{N}_0^T, \Sigma \rangle$ mit $\Sigma = \mathbb{N}_0^S \times \mathbb{N}_0^T \times \mathbb{N}_0^S$ dargestellt werden. Die Kantenmenge des Transitionssystems stellen die Markierungen (als Multimenge über die Stellen des S/T-Netzes) dar; die Kantenlabel sind im Gegensatz zum Transitionssystem der sequentiellen Schaltsemantik eine Multimenge über die Transitionen des S/T-Netzes (d.h. ein nebenläufiger Schritt). Die Transitionen des Transitionssystems A stellen Tupel $(m, s, m') \in \Sigma$ (oder äquivalent auch $m \xrightarrow{s} m'$) dar. Anhand dieser Definition lässt sich erkennen, dass die nebenläufige Schaltsemantik einer Verallgemeinerung der zuvor eingeführten sequentiellen Schaltsemantik entspricht. Das (bekannteste) Konzept dieser nicht-sequentiellen Schrittsemantiken von S/T-Netzen stellen Prozesse dar [GR83a, GR83b].

Ein Kausalnetz ist ein gerichteter, bipartiter Graph $O = \langle B, E, G \rangle$ mit den Stellen B (hier auch Bedingungen genannt), der zur Menge der Bedingungen disjunkten Menge von Ereignissen E als Transitionen (d.h. $B \cap E = \emptyset$) und einer Menge gerichteter Kanten $G \subseteq (B \times E) \cup (E \times B)$. Dabei sind Bedingungen innerhalb eines Kausalnetzes unverzweigt, weshalb $O: \forall b \in B : |\bullet b|, |b \bullet| \leq 1$ gilt. Ferner besitzen Kausalnetze keine Zyklen. Ein Kausalnetz stellt alle möglichen Abläufe eines markierten S/T-Netzes $\langle N, m \rangle$ mit paralleler Schaltsemantik dar. Die Generierung eines Kausalnetzes wird als Entfaltung bezeichnet und kann als ein mögliches Mittel betrachtet werden, um das Problem der Zustandsexplosion bei der Verifikation nebenläufiger Systeme zu adressieren [McM93]. Für die Entfaltung existieren bereits geeignete, algorithmische Lösungen [ERV02, MP95]. Mittels Entfaltung des S/T-Netzes können u.a. Erreichbarkeitsprobleme adressiert werden [ES01] und somit beispielsweise eine automatische Modellprüfung auf Basis von S/T-Netzen durch Entfaltung realisiert werden [Esp94, Kho03]. Im Kontext der automatischen Planung bilden die Bedingungen die Prämissen von durch Events repräsentierten Aktionen eines durch Entfaltung eines S/T-Netzes erzeugten Kausalnetzes. Hickmott et al. [HRTW07] fassen die Unterschiede zwischen dem durch Entfaltung entstandenen Kausalnetz und einem Planungsgraphen (wie er beispielsweise in GraphPlan verwendet wird) wie folgt zusammen: (a) während durch Entfaltung eine vollständige Erreichbarkeitsanalyse durchgeführt wird, wird in Planungsgraphen diese nur approximiert. (b) Die Entfaltung bzw. das resultierende Kausalnetz garantiert durch Replikation von Knoten, dass jeder Bedingung ein eindeutiges Ereignis vorausgeht (die sog. post-uniqueness²⁵). Der durch Hickmott et al. [HRTW07, HS09] beschriebene Ansatz für die automatische Planung durch Entfaltung bestimmt dabei partial geordnete Pläne – d.h. POP – unter einem frei wählbaren, numerischen Optimalitätskriterium.

²⁵ Aus algorithmischer Sicht im Kontext automatischer Planungsverfahren werden durch die Eigenschaft der post-uniqueness rückwertige Konflikte (engl. backward conflicts) vermieden [BN95], was letztendlich dazu führt, dass die Menge von Aktionen, die zu einer spezifischen Wertebelugung von Zustandsvariablen führen, eindeutig bestimmt ist.

3.2. Anwendung automatischer Planungsverfahren

Der grundlegende technologische Hintergrund von Industrie 4.0 stellt die Einführung und Anwendung moderner Informationstechnologie im Zusammenspiel mit einigen technischen Visionen dar [DH14]. Vernetzung, globaler Datenaustausch und autonome, selbstorganisierte Entscheidungsfindung stellen dabei geforderte Charakteristiken automatisierter Produktionssysteme dar. Diese Technologien werden aus konzeptioneller Sicht in cyber-physische Systeme zusammengefasst, welche die Grundlage für die Realisierung der in Industrie 4.0 proklamierten, intelligenten, technischen Systeme darstellen [MKB⁺16, LBK15, And15]. Die Anwendung des allgemeinen Konzeptes cyber-physischer Systeme (siehe hierzu [Lee08, RLSS10]) in der Maschinen- und Anlagenautomatisierung – häufig auch als cyber-physische Produktionssysteme bezeichnet – ist dabei nicht ohne weiteres möglich, was nicht zuletzt auf die Menge und spezifische Kombination von Anforderungen wie beispielsweise harte Echtzeitanforderungen, etablierte Standards, Zuverlässigkeit und funktionale Sicherheit zurückzuführen ist [VHDB13]. Daher existieren noch zahlreiche Herausforderungen in der Anwendung des Konzeptes cyber-physischer Systeme in der Produktion. Im Kontext dieser Arbeit wird entsprechend der Zielstellung insbesondere die Herausforderung der (autonomen) Entscheidungsfindung betrachtet. Automatische Planungsverfahren stellen in diesem Zusammenhang eine essentielle Technologie und algorithmische Grundlage dar [BBB⁺11, NHS⁺15]. Eine weiterführende, ausführliche Beschreibung und Übersicht weiterer Herausforderungen für die Anwendung cyber-physischer Systeme in der Produktion kann beispielsweise [LCK16, Mon14, MKB⁺16] entnommen werden.

Das Konzept cyber-physischer Systeme ist jedoch nicht von Grund auf völlig neu, sondern vielmehr eine evolutorische Weiterentwicklung von in den vergangenen Jahrzehnten entwickelten Konzepten in unterschiedlichen (Forschungs-)Bereichen. Diese weisen teils eine gemeinsame Basis, jedoch ebenfalls Unterschiedlichkeiten auf. Eine Übersicht und detaillierte Charakterisierung bieten [LCK16, MKMO11]. Das in den 1990er Jahren von Mark Weiser als Zukunftsvision für das 21. Jahrhundert formulierte *Ubiquitous Computings* [Wei91] stellt für viele dieser Forschungsrichtungen Inspiration und konzeptionelle Basis dar: Eine Welt, in der IT im Alltag allgegenwärtig ist und von Anwendern die Anwesenheit dieser ubiquitären Systeme nicht mehr explizit wahrgenommen wird, sich jedoch gleichzeitig technisches Equipment automatisiert an die Bedürfnisse und Wünsche anpasst. Diese Vision deckt sich im Kern mit der Zielsetzung von Industrie 4.0 (vgl. hierzu auch die Diskussionen hinsichtlich zukünftiger Arbeitswelten [Bun15] und die Integration des Menschen in zukünftigen Produktionsumgebungen [PRK⁺15, GSLZ14]), weshalb eine zunehmende Annäherung und Weiterentwicklung an diese Vision beobachtet werden kann [Abo16, Zue10]. Automatische Planungsverfahren wurden bereits als eine der wesentlichen Technologien ubiquitärer Systeme identifiziert, um die gewünschte Umgebungsintelligenz und Adaptivität zu erreichen [RAS08, Aug07].

Das sogenannte Internet der Dinge (im Englischen als "internet of things" bezeichnet) stellt in diesem Zusammenhang ein weiteres (im Vergleich zu anderen Strömungen eher objektzentriertes) Forschungsfeld dar. Eine allgemeingültige Definition ist hierfür kaum anzugeben, da eine Vielzahl unterschiedlicher Interpretationen der Vision existieren [AIM10]. Allgemein weist das Konzept jedoch stets drei wesentliche Säulen auf: (a) eindeutige Identifizierbarkeit physischer Objekte mit (b) Kommunikationsfähigkeit und der (c) Fähigkeit untereinander zu interagieren [MSDPC12, LDXX15]. Die Anwendungsfelder eines Internets der Dinge reichen dabei von Strom-

netzen und Hausautomation über Verkehrsinfrastrukturen für Automobil oder Zug bis hin zu Produktion [Bor14, WADX15, DXHL14]. Ausgehend von der zentralen Idee eines Internets der Dinge existieren wiederum spezifische Strömungen bzw. Forschungsrichtungen innerhalb der Maschinen- und Anlagenautomatisierung, welche hinsichtlich der drei zuvor genannten Eigenschaften eine mehr oder weniger starke Ausprägung aufweisen. Machine-to-machine (M2M) Kommunikation stellt beispielsweise eine Forschungsrichtung dar, welche (wie der Name bereits suggeriert) primär die Kommunikationsfähigkeit zwischen Dingen (hier speziell Maschinen und/oder deren Bauelementen) fokussiert [Pro12]. Daher wird M2M häufig auch als Wegbereiter und ersten Schritt hin zur Anwendung cyber-physischer Systeme in der Produktion und Industrie 4.0 betrachtet [HTM⁺14, WYL⁺13]. Eine weitere Strömung im Kontext cyber-physischer Systeme stellen Forschungen im Hinblick auf intelligente bzw. smarte Produkte dar [MGWH13, MFH09], was letztendlich als eine spezifische Interpretation der Vision eines Internet der Dinge mit dem Fokus der auf die Interaktionsmechanismen zwischen Entitäten und deren potentiell inhärente Intelligenz betrachtet werden kann. Die Ausprägung der Intelligenz solcher smarten Produkte reicht dabei von einfacher Datenhaltung²⁶ – im Hinblick auf ihr Kommunikationsverhalten häufig auch als passive Produkte bezeichnet – bis hin zu einer komplexen Entscheidungsfindung und Einflussnahme auf den (produkteigenen) Fertigungs- bzw. Produktionsprozess [MFH09]. Letztere werden dabei entsprechend ihres (Kommunikations-)Verhaltens auch als aktive Produkte bezeichnet. Wie sich im Zuge einer praktischen Umsetzung intelligenter Produkte in der industriellen Praxis zeigt, ergeben sich hierdurch alternative Möglichkeiten für die Nachverfolgbarkeit und Diagnosefähigkeit von Produkten [LRB⁺15]. Während passive Produkte in der industriellen Praxis bereits eingesetzt werden, ist die Anwendung aktiver intelligenter Produkte noch Gegenstand der Forschung. Nachweise für Vorteile des Einsatzes eines aktiven Produktes im Vergleich zu etwaigen Lösungsalternativen konnten noch nicht endgültig erbracht werden.

Im Folgenden wird (unabhängig etwaiger konzeptioneller Aspekte und Anwendungsfelder) die Anwendung automatischer Planungsverfahren untersucht. Dabei werden in Abschnitt 3.2.1 unterschiedliche Softwarearchitekturen, welche für die Realisierung intelligenter technischer Systeme – d.h. cyber-physischer Systeme – als wesentliche Umsetzungstechnologien betrachtet werden, im Hinblick auf die Anwendung automatischer Planungsverfahren untersucht. Anschließend werden in Abschnitt 3.2.2 automatische Planungsverfahren, welche bereits in der Anlagenautomatisierung eingesetzt werden untersucht und anhand der in dieser Arbeit gestellten Anforderungen gegenübergestellt.

3.2.1. Anwendung von Planungsverfahren in unterschiedlichen Software Architekturen

In der Literatur werden insbesondere zwei Softwarearchitekturen als mögliche Umsetzungstechnologien für die Realisierung cyber-physischer Produktionssysteme genannt [LCK16, VHLL15]: Die Service-orientierte Architektur sowie Multi-Agentensysteme. Aus diesem Grund werden diese beiden Architekturen (sowie deren Kombinationsmöglichkeiten) im Folgenden näher betrachtet, ihre Anwendung in der Maschinen- und Anlagenautomatisierung beleuchtet und dabei die Nutzung automatischer Planungsverfahren in diesen Architekturen im Hinblick auf die in dieser Arbeit gestellten Anforderungen untersucht.

²⁶Diese Art der Ausprägung smarterer Produkte findet sich häufig im Kontext der Anwendung von AutoID Technologien [MSC⁺03], wie beispielsweise RFID.

3.2.1.1. Dienstorientierte Architektur

Die Service-orientierte Architektur (SOA), oft auch als dienstorientierte Architektur bezeichnet, ist ein Architekturmuster zur Strukturierung und Nutzung von Funktionalitäten innerhalb verteilter (Software-)Systeme [MLM⁺06]. Im Bereich der Softwareentwicklung ist die service-orientierte Architektur ein akzeptiertes und etabliertes Mittel zur Kapselung und Kopplung von Softwarefunktionalität, um beispielsweise der zunehmenden Komplexität betrieblicher Anwendungen zu begegnen und die Agilität von Unternehmen zu steigern [AS08].

Es existieren eine Vielzahl von Grundprinzipien dienstorientierten Designs [Erl08]: Zu nennen sind im Zusammenhang dieser Arbeit insbesondere eine geringe Kopplung (engl. loose coupling) und Abhängigkeit zwischen Diensten, deren Wiederverwendbarkeit und Auffindbarkeit (engl. discoverability) sowie Komponierbarkeit (engl. composability). Diese Eigenschaften bilden die Grundlage, um mittels zur Verfügung stehenden Diensten bzw. mittels der durch Dienste bereitgestellten Funktionalitäten komplexere Funktionalitäten zu realisieren. Ein service-orientiertes Softwaresystem besteht aus einer Menge von Diensten sowie einem Ablaufmodell (engl. workflow model). Ein solches Softwaresystem bestehend aus Diensten und Ablaufmodell kann wiederum als Dienst angeboten werden. Ein solcher Dienst wird auch als zusammengesetzter Dienst (engl. composite service) bezeichnet. Hierdurch können hierarchische Softwarestrukturen erstellt werden. Eine sogenannte Orchestration Engine interpretiert dabei ein gegebenes Ablaufmodell und realisiert die Funktionalität des Softwaresystems. Ablaufmodelle werden genutzt, um Ausführungslogik (wie z.B. Reihenfolge der Aufrufe von Diensten und damit verbundene Abhängigkeiten wie Parameterwerte) zu spezifizieren. Die Funktionalität eines service-orientierten Softwaresystems ist durch das Ablaufmodell sowie die darin orchestrierten Dienste definiert und wird durch einer entsprechende Orchestration Engine realisiert.

Häufig werden sog. Webdienste (engl. Web Services) mit SOA synonym gesetzt. Tatsächlich stellen Webdienste lediglich eine spezifische, aber stark verbreitete Weise dar, eine service-orientierte Architektur insbesondere im Kontext des Internets zu implementieren [NL05]. Im Zuge der Bemühungen hinsichtlich Webdiensten wurden und werden eine Vielzahl von Standards entwickelt. Die Basis bilden dabei SOAP (Simple Object Access Protocol) als Zugriffsprotokoll für Dienste sowie WSDL (Web Service Description Language) zur Beschreibung der Schnittstelle von Diensten. Im Kontext dieser Webdienste sind mittlerweile mehr als 150 weitere Spezifikationen für unterschiedliche Aspekte und Anwendungsfelder verfügbar [WCL⁺05] – diese werden auf Grund ihrer Benennung mittels dem Prefix WS auch als WS-* Spezifikationen bezeichnet. Zu nennen ist in diesem Zusammenhang beispielsweise WS-BPEL zur Beschreibung von Workflows.

Die inhärenten Eigenschaften einer dienstorientierten Architektur, wie z.B. die lose Kopplung zwischen Funktionseinheiten (Diensten) und die damit verbundene, strikte Anwendung von Schnittstellendefinitionen, stellen eine Lösungsmöglichkeiten dar, um Flexibilität und Anpassbarkeit innerhalb eines Softwaresystems zu gewährleisten. Dieser Zielstellung entspricht auch die zunehmende Forderung nach Anpassbarkeit der Produktion und damit auch der Steuerungssoftware. Eine Überführung entwickelter Konzepte dienstorientierter Architekturen aus der IT für die Automatisierungstechnik ist jedoch auf Grund teils unterschiedlicher Charakteristika nicht ohne Anpassungen und etwaige Einschränkungen möglich [OSH11]. Aus diesem Grund befindet sich das SOA Paradigma in der industriellen Automatisierungstechnik des Maschinen- und Anlagenbaus noch im aktiven Stand der Forschung. Eine Diskussion der besonderen Charakteristika

und Unterschiede zwischen Diensten in der IT im Vergleich mit Diensten in der Automatisierungstechnik können [OSH11] entnommen werden. Dienstorientierte Softwaresysteme operieren typischer Weise ereignisorientiert auf Basis von (synchronen oder asynchronen) Dienstaufrufen. Im Gegensatz dazu steht beispielsweise das Paradigma zyklisch operierender Steuerungssoftware nach IEC 61131. So können für den Einsatz dienstorientierter Architekturen für IEC 61131 basierte Speicherprogrammierbare Steuerungen nicht alle Eigenschaften übernommen werden, wie in [LVH14b, LVH15] beschrieben.

Im Zuge des SIRENA Projektes²⁷ [BBG06, JS05] wurde die Anwendung der SOA in der Automatisierung von Produktionssystemen fokussiert. Im Rahmen dieses Projektes wurde mittels der Entwicklung von DPWS (Device Profile for Web Services) — einer Spezifikation für Webdienste auf eingebettete, ressourcenarme Geräte wie Sensoren und Aktoren — die Grundlage für die Anwendung der dienstorientierten Architektur auf ressourcenarme Geräte wie Sensoren, Aktoren und Feldsteuergeräte gelegt. Mittlerweile ist DPWS ein durch die Institution OASIS verwalteter Standard [OAS09]. So wurde beispielsweise darauf aufbauend im Rahmen des Forschungsprojektes SOCRADES²⁸ [SSG⁺08, CGT08, DSSG⁺08] die Anwendung der dienstorientierten Architektur für die Feldgeräte und die Steuerungsebene näher untersucht und ein Konzept entwickelt, um auf Basis der WS-* Standards die Funktionalität der Steuerungsebene auf Basis von Diensten und deren geeigneter Zusammensetzung zu modellieren und höheren Schichten der Steuerungsarchitektur den Zugriff auf Informationen der Feldebene zu ermöglichen (z.B. Abfrage von Sensorwerten durch ein Leitsystem mittels Dienstaufruf). Damit legte das Projekt SOCRADES die Grundlage für weitere Forschungen zur Anwendung dienstorientierter Architekturen für die Automatisierung der Produktion, die durch eine Vielzahl nachfolgender Forschungen für unterschiedliche Anwendungsgebiete weitergeführt wurde, z.B. um die Entwicklung und Agilität cyber-physischer Systeme zu verbessern [ZO12, TOJ12]. Die für diese Arbeiten essentielle Eigenschaft dienstorientierter Architekturen – die Möglichkeit der Flexibilisierung von Abläufen – wurde bereits im Jahr 2005 durch Jammes et al. [JSLD05] als ein wesentlicher Vorteil der SOA für die Produktionsautomatisierung identifiziert.

In diesem Zusammenhang gewinnt die Zielsetzung eines vollständig automatisierten Prozesses zum Auffinden geeigneter Dienste, deren Zusammenfügen und Ausführen zunehmend an Bedeutung. Der sog. Dienstplanungsprozess wird von Blau et al. [BNWL08] als die Analyse von Dienstbeschreibungen und dem Erstellung von geeigneten Vorschlägen zu deren Kombination für den Anwender beschrieben. Die Analyse von Dienstbeschreibungen wird dabei einerseits für das Auffinden geeigneter Prozesse als auch die Identifikation der Kombinierbarkeit von Diensten angewandt. Beide Aufgaben lassen sich technisch auf den Abgleich von Dienstbeschreibungen, dem sog. Matchmaking, zurückführen. Das Ergebnis des Matchmakings stellt letztendlich eine Beschreibung aller möglichen Kombinationen bekannter Dienste dar. Im Anschluss an das Matchmaking wird eine Auswahl der geeigneten und ggf. besten Kombination der Dienste ausgeführt. In Abhängigkeit des genutzten Verfahrens wird das Modell des Matchmakings ggf. nicht vollständig erstellt bzw. materialisiert – insbesondere ist dies in einer offenen Dienstumgebung mit weltweit

²⁷Das Projekt "Service Infrastructure for Real-time Embedded Networked Applications" (SIRENA) wurde von Januar 2003 bis September 2005 durchgeführt und im Rahmen der ITEA3 Initiative gefördert.

²⁸Das Projekt SOCRADES ("Service-Oriented Cross-layer Infrastructure for Distributed Smart Embedded Devices") wurde von 2006 bis 2009 durch die Europäische Union im Kontext des Rahmenprogrammes FP6 gefördert.

3. Stand der Wissenschaft und Technik

verfügbaren Softwarediensten kaum effizient lösbar – sondern beide Schritte (Matchmaking und Auswahl) in Kombination ausgeführt.

Eine Vielzahl von Forschungsarbeiten wurden bereits im Zusammenhang mit der automatisierten Dienstaussführung durchgeführt, um die Herausforderung der automatischen Dienstplanung zu adressieren (vgl. beispielsweise [RS05, SPAS03]). Die meisten Ansätze basieren dabei auf Planungstechniken und Algorithmen der Künstlichen Intelligenz [Pee05], insbesondere da sich das grundlegende Dienstplanungsproblem vollständig und eindeutig als Start-Ziel-Planungsproblem formulieren lässt. In Sirin et al. [SPW⁺04] wird beispielsweise HTN mittels SHOP2 zur Dienstplanung verwendet während Zeng et al. [ZBN⁺04] Lineare Programmierung zur Auswahl des optimalen Ausführungsplans nutzt. Die Komplexität der Beschreibung von Diensten sowie die damit verbundenen Matchmaking-Algorithmen beeinflussen häufig signifikant die Performanz des Gesamtsystems [SPAS03]. Eine detaillierte Übersicht und Gegenüberstellung von Forschungsergebnisse zur automatischen Dienstplanung können beispielsweise [SP04, KSKR05] entnommen werden. Die Fragestellung einer automatischen Dienstplanung und Ausführung ist dementsprechend eng mit der in dieser Arbeit gestellten Forschungsfrage der automatischen Handlungsplanung verbunden.

So verfolgt das iLAND Projekt²⁹ einen dienstorientierten Ansatz, um auf Basis einer Middleware vernetzte, eingebettete Systeme hinsichtlich ihrer Funktionalität dynamisch zu koppeln. Hierbei wird ein dienstorientierter Ansatz verfolgt [GVLV13]. Verfügbare, ausführbare Dienste und deren mögliche Kompositionen werden durch einen sog. Applikationsgraph dargestellt. Die Planung einer geeigneten Komposition wird als Identifikation eines geeigneten Pfades innerhalb des Applikationsgraph interpretiert. Hierfür werden in [EABVGV⁺09] spezielle Algorithmen vorgeschlagen, um eine geeignete Komposition unter besonderer Berücksichtigung von Qualitätsmerkmalen in Echtzeit zu bestimmen. Im Kontext dieser Arbeiten wird von echtzeitfähiger Rekonfiguration gesprochen, da die Rekonfigurationsplanung (i.e. Kompositionsplanung) innerhalb zeitlich gegebener, deterministischer Obergrenzen erfolgen kann [EAGVABV08]. Ein modellbasierter Ansatz für die vereinfachte Entwicklung von Diensten wurde auf Basis des MARTE UML Profils vorgeschlagen [MEJK11]. Eine Berücksichtigung produktionstechnischer Anwendungsfälle und deren spezifischen Randbedingungen erfolgten im Rahmen des Projektes nicht. Im Rahmen des Projektes eSCOP³⁰ [RGI⁺14, GFLML13] wird die Anwendung der dienstorientierten Architektur für die Flexibilisierung der Feldebene von Fertigungssystemen untersucht und stellt eines der zuvor bereits angesprochenen Forschungsprojekte dar, welche konzeptionell auf den Ergebnissen des SOCRADES Projektes aufbauen. Die Flexibilisierung wird dabei durch einen zentralen Dienst zur Auffindung anderer Dienste ermöglicht, welcher Beschreibungen verfügbarer Dienste innerhalb eines Fertigungssystems enthält und legt damit die Grundlage für eine automatische Dienstplanung auf Basis eines mathematisch logischen Modells [PLML13]. In [LSH⁺11, LHSS12, Los13] wird ein Konzept für die Anwendung der dienstorientierten Architektur für die Steuerung von Fertigungssystemen vorgestellt, in dessen Kern eine Methodik für das Zusammenfügen logik-basierter Beschreibung der Funktionalität von Feldgeräten genutzt wird. Die Anwendung dieses Konzeptes zur Steigerung der Agilität von Fertigungssystemen wird

²⁹Das Projekt „Middleware for Deterministic Dynamically Reconfigurable Networked Embedded Systems“ (iLAND) wurde im Rahmen des EU ARTEMIS im Zeitraum 2009 – 2012 gefördert.

³⁰Das Projekt „Embedded Systems for Service-based Control of Open Manufacturing and Process Automation“ (eSCOP) wurde von 20013 bis 2016 im Rahmen von EU ARTEMIS gefördert.

in [FLR⁺13] diskutiert. Ein Verfahren für die automatische Bestimmung von Ablaufreihenfolgen der Dienste wird nicht betrachtet. Auch Mendes et al. [MLRC10, MLCR12] adressieren die Beschreibung von Diensten für Fertigungssysteme und schlagen hierzu hierarchische S/T-Netze vor. Dabei wird der Formalismus um einen Mechanismus für das Zusammenfügen von Diensten und der Erstellung einer Beschreibungen der daraus resultierenden Komposition erweitert. Dieser Ansatz schlägt die Brücke zwischen dem Konzept der Dienste für die Steuerungssoftware fertigungstechnischer Systeme und dem Formalismus der S/T-Netze. Dabei wird aufgezeigt, dass eine Komposition dieser Dienste sich ebenfalls innerhalb eines S/T-Netzes geeignet abbilden lässt. Alle zuvor beschriebenen Arbeiten adressieren primär den integrative Aspekt von Diensten und deren geeignete Beschreibung für das Auffinden und sequentielles Zusammenfügen zweier Dienste bzw. Dienstbeschreibungen; Verfahren zur automatischen Dienstplanung finden hierbei jedoch keine Anwendung.

Im Rahmen dieses Abschnitts wurde gezeigt, dass die Fragestellung der automatischen Dienstplanung für IT Systeme mittels automatischen Planungsverfahren in Form einer klassischen, zielbasierten Formulierung des Planungsproblems adressiert werden kann. Zahlreiche Forschungsarbeiten adressieren dabei mittlerweile auch die Anwendung der dienstorientierten Architektur für die Steuerungssoftware von Fertigungssystemen. Es existieren jedoch noch keine bekannten Arbeiten, welche eine automatische Dienstplanung für die Steuerungsebene des Maschinen und Anlagenbaus adressieren, insbesondere nicht mittels der Konzepte der automatischen Handlungsplanung, wie sie im Rahmen dieser Dissertation untersucht wird.

3.2.1.2. Multi-Agenten Systeme und (Selbst-)Adaptive Systeme

Um den Flexibilitätsanforderungen in der Produktion zu begegnen, wird seit geraumer Zeit die Anwendung von Techniken aus dem Bereich der Künstlichen Intelligenz untersucht [BBB⁺11, Par96]. In diesem Zusammenhang ist insbesondere das Konzept der Agenten bzw. der Agentenorientierung zu nennen. Agenten sind intelligente, kollaborierende Softwareentitäten, die für eine verteilte Problemlösung eingesetzt werden können [Woo09, RN95, Wei99]. Die Anwendung dieses Softwareparadigmas in der industriellen Produktionstechnik wurde bereits für eine Vielzahl unterschiedlicher Anwendungsfelder von der verteilten Produktionsplanung [SWH06, ZX07] bis zur Echtzeitsteuerung auf der Feldebene [SSF⁺11, VHGL15] vorgeschlagen. Darüber hinaus werden Agenten als adäquate Möglichkeit betrachtet, Cyber-physische Produktionssysteme umzusetzen [VHLL15]. Einen Überblick über verschiedene Anwendungsfelder geben beispielsweise [LMV13, MP11, VM10, ML07, SHYN06]. Konkret, kann ein Agent als eine autonome Komponente betrachtet werden, welche physische oder logische Einheiten eines Systems repräsentiert und in der Lage ist, anderen Agenten sowie seine Umwelt zu beobachten, mit beiden zu interagieren und Entscheidungen zu treffen, um seine Ziele zu erreichen [VDI10, Lei09]. Ein Multi-Agenten System (MAS) ist eine Softwarearchitekturmuster, welches eine Menge von interagierenden Agenten enthält. Agenten arbeiten gemeinschaftlich zusammen, um ein gemeinsames oder eine Vielzahl (möglicherweise widersprechender) Ziele zu verfolgen. Multi-Agenten Systeme zeichnen sich durch ein kollaboratives, dynamisches Verhalten aus, welches ermöglicht, flexibel durch Verhaltensanpassung auf sich ändernde, externe Bedingungen zu reagieren und dabei auf wenige externe Eingriffe angewiesen sind [Woo09].

3. Stand der Wissenschaft und Technik

Um Agenten in der industriellen Produktionstechnik anzuwenden, gibt es verschiedene konzeptuelle Möglichkeiten [VHLL15, VHGL15, FLW⁺12]:

- Ein (**entkoppelter**) **Softwareagent** agiert als (informationelle) Softwareentität, ohne physikalische Produktionsressourcen explizit zu berücksichtigen, wie z.B. in [LKP⁺05] beschrieben.
- Physikalische Hardwarekomponenten der Produktionsanlage werden als Module mit integrierter Echtzeitsteuerung betrachtet. Durch (**gekoppelte**) **Softwareagenten**, welche mit der Steuerung des Hardwaremoduls interagieren, wird ein gewisses Maß an Intelligenz ermöglicht. Durch die Kopplung wird mittels des Softwareagenten das Verhalten der Echtzeitsteuerung, beispielsweise durch Parametrierung beeinflusst, wie z.B. in [VTM⁺11, LZVM11] beschrieben. Eine geeignete Kopplung ist aktuell intensiver Gegenstand der Forschung, da beispielsweise bis dato keine standardisierten Schnittstellen existieren [SZ15].
- Wenn der informationelle Teil, d.h. der Agent, mit den physikalischen Hardwarekomponenten integriert betrachtet wird, spricht man von **Software/Hardware Agenten**. Dabei ist der Agent vollständig innerhalb eines physikalischen Moduls aggregiert, um das Modul innerhalb überlagerter Steuerungslogik informationell zu repräsentieren und das Modul zu steuern, d.h. den (vom Modul realisierbare Teil des) technischen Prozesses zu realisieren. Hierbei müssen die für Software der Feldebene geltenden, harten Echtzeitbedingungen berücksichtigt werden.

Im Zuge der Bemühungen von Industrie 4.0, stets flexiblere Softwaresysteme und Steuerungssysteme zu entwickeln, werden zusehends intelligente, sich selbst verwaltende Systeme postuliert, welche ausreichend Freiheiten besitzen, um sich dynamisch an verändernde externe und interne Bedingungen ohne oder lediglich eingeschränkte manuelle Eingriffe anpassen zu können [Sch05, Lei08]. In diesem Zusammenhang stellen Autonomie und Selbstorganisation wesentliche Verhaltenseigenschaften der Systeme dar (vgl. Abschnitt 1). Allgemein wird in diesem Kontext von sogenannten self-x Eigenschaften (auch als self-* Eigenschaften bezeichnet) gesprochen, wobei der Platzhalter für eine gewünschte Eigenschaft des Systems steht [OSL11]. Zwei in diesem Zusammenhang bedeutende Eigenschaften stellen die Selbstheilung (engl. self-healing) sowie die Selbstoptimierung (engl. self-optimization) dar. Psaiar und Distdar [PD11] definieren die Eigenschaft der Selbstheilung als die autonome Entdeckung, Diagnose und Reaktion auf Störungen. Im Gegensatz dazu ist ein selbstoptimierendes System der eigene, ideale Leistungszugstand bekannt, kann aktuelle Leistungswerte messen und ist versucht, durch geeignete Strategien die eigene Leistung zu verbessern [Ste05]. Eine allgemeine Übersicht über verschiedene Forschungen im Bereich solcher sich selbst anpassender IT Systemen kann z.B. [ST09, BDR07, PD11] entnommen werden. Eine Übersicht über die Anwendung von Self-x Eigenschaften in Automatisierungssystemen finden sich beispielsweise in [FS11, BT02, Lei08]. Brachman [Bra02] identifiziert die Notwendigkeit, dass solch intelligente Systeme ein Verständnis benötigen, welche Tätigkeiten aktuell ausgeführt werden sowie welche Möglichkeiten zu agieren einem System zur Verfügung stehen (self-awareness), um intelligente self-x Eigenschaften letztendlich realisieren zu können. In diesem Kontext stellen automatische Planungsverfahren einen essentiellen, technologischen Beitrag dar [BBW07, BBB⁺11].

Die Notwendigkeit einer Planungskomponente für die Feldebeneensteuerung im Maschinen- und Anlagenbau wurde bereits in [Wan10, SSF⁺11] identifiziert. Eine Anwendung von Planungsverfahren im Sinne dieser Arbeit erfolgte hierbei jedoch bisher nicht. Um diese Verfahren in eine Agentenarchitektur geeignet einzubetten, existieren unterschiedliche Modellkonzepte. Im Folgenden wird das belief-desire-intention (BDI) Modell exemplarisch für die Entwicklung von Softwareagenten diskutiert.

Ein Agent, welcher das BDI Modell anwendet, wird auch vereinfacht BDI Agent genannt. Das BDI Modell unterscheidet, entsprechend seinem Namen, zwischen dem Weltwissen (*belief*), dem Begehren³¹ (*desires*) sowie den Absichten (*intentions*) eines Agenten. Das "Weltwissen" stellt die Gesamtmenge der einem Agenten zur Verfügung stehenden Informationen – also die Vorstellung des Agenten – über seine Umwelt dar (engl. *beliefset*). Das "Begehren" eines Agenten repräsentiert die übergeordnete Zielsetzung oder auch Situation, die ein Agent erreichen möchte. In diesem Zusammenhang wird von Zielen (engl. *Goals*) gesprochen, falls ein Agent dieses aktuell aktiv verfolgt. Die Absichten eines Agenten stellen die aktuelle Zielsetzung dar. Um dies zu konkretisieren, wird in typischen BDI System von Plänen gesprochen, die ein Agent ausführt, um ein Ziel zu erreichen. Die Absicht eines Agenten kann dann als Ausführung des Plans im Sinne der automatischen Planung verstanden werden [RG95]. Während "Begehren" repräsentieren, was ein BDI Agenten tun möchte, stellen "Absichten" dar, was ein Agent ausgewählt hat, zu tun. Im Kontext dieser Arbeit ist insbesondere die Entscheidungsfindung eines BDI Agenten von Bedeutung, in der zu verfolgende Absichten identifiziert werden, um ein gegebenes Ziel zu erreichen. Eine mögliche Lösung, um diese Entscheidungsfindung für BDI Agenten zu realisieren, stellen automatische Planungsverfahren dar. Auf Grund der vorgegebenen BDI Struktur lässt sich dabei eine Handlungsplanung und die damit verbundene Entscheidungsfindung meist auf eine zielbasierte Planung zurückführen [Bra87]. Daher finden für die typische Handlungsplanung von BDI Agenten die in Abschnitt 3.1.1 beschriebenen formalen Planungskonzepte wie logik-basierende Planungsansätze [MZCM04, DSSP09] ebenso Anwendung wie hierarchische Planungsverfahren [TMS13, SSP06]. Die allgemeine Problemformulierungen der zielbasierten Planung widerspricht jedoch der gestellten Anforderung A1.2 und Anforderung A1.3. Daher sind klassische BDI Agenten für die hier vorliegende Problemstellung nicht anwendbar³².

Um eine Koordination zwischen Entitäten und deren Zielen innerhalb eines Multi-Agenten Systems zu erreichen, werden Koordinationsmechanismen benötigt [RC10]. Der Grad der notwendigen Koordination zwischen Agenten ist dabei stark vom Grad der Kopplung (und damit der Abhängigkeit von Aktionen) zwischen Agenten abhängig [BD08]. Die Koordination innerhalb eines Multi-Agenten Systems weist dabei inhaltlich einen starken Planungsaspekt auf. Aus diesem Grund wird in diesem Zusammenhang auch oft von einem Multi-Agenten-Planungsproblem (engl. *multi-agent planning problem*) gesprochen. Zumeist wird von einer vollständig verteilten Problemlösung (engl. *distributed problem solving*) ausgegangen, wobei unter anderem auch

³¹Häufig werden *desires* fälschlicher Weise mit Zielen übersetzt. Ziele (engl. *goals*) haben jedoch in diesem Zusammenhang eine gesonderte Bedeutung.

³²Auf Grund des hohen Abstraktionsgrades des BDI Modells können grundsätzlich eine Vielzahl unterschiedlicher, automatischer Planungsverfahren darin subsumiert werden. Aus diesem Grund wurde bisher ausschließlich von typischer Handlungsplanung im Kontext von BDI gesprochen. Das im Rahmen dieser Dissertation entwickelte, automatische Planungsverfahren kann konzeptionell ebenfalls in das BDI Modell — jedoch auf einer abstrakteren Detaillierungsstufe als vorgeschlagen — eingeordnet werden, da das in dieser Arbeit vorgeschlagene Planungskonzept auf dem Paradigma der zielbasierten Planung aufbaut.

3. Stand der Wissenschaft und Technik

das zur Verfügung stehende Planungswissen als verteilt angenommen wird. Jennings [Jen96] beschreibt das Multi-Agenten-Planungsproblem als den Prozess, in dem ein Agent eigene sowie (antizipierten) Aktionen anderer Agenten beurteilt, um sicherzustellen, dass alle Agenten innerhalb eines Agentensystems stimmig agieren. Multiagenten-Planung verbindet somit die klassische Planung mit einem koordinativen Aspekt (siehe [DWTMW05] für eine detaillierte Diskussion). Ein Ansatz, welcher ein klassisches Planungsproblem für die Lösung des Multi-Agenten-Planungsproblems anwendet, wird von Dimopoulos und Moraitis [DM06] beschrieben. In [BD08, NBD10] wird ein Verfahren zur verteilten Multi-Agenten-Planung auf Basis einer Erweiterung von STRIPS um Agenten und deren Menge möglicher Aktionen vorgestellt. Dieses MA-STRIPS genannte Verfahren wurde dabei durch ein (verteilt) CSP realisiert. Es existieren jedoch auch Ansätze, welche innerhalb eines verteilten Systems zusätzlich eine hierarchische Planung zu Grunde legen. Auf Basis dieser Idee wurde von Lesser et al. [DL91, DL92] eine partielle, globale Planung vorgeschlagen (engl. partial global planning). Dieses PGPlan genannte Konzept ermöglicht den Austausch von Zielen zwischen Agenten, welche von einem Agenten lokal als Zwischenziele berücksichtigt werden können. Planungstheoretisch kann dies auf ein zielbasiertes Planungsverfahren zurückgeführt werden, welches bei der Planerstellung Zwischenziele berücksichtigt. Auf diese Weise könnten Anforderung A1.1 und Anforderung A1.3 möglicherweise berücksichtigt werden; Anforderung A1.2 impliziert für die Massenfertigung jedoch die Berücksichtigung von wiederkehrenden Aktivitäten. Dies erfordert im Kontext von PGPlan die mehrfache Berücksichtigung zumindest eines Zwischenziels innerhalb eines einzelnen Planes. Dies ist in PGPlan nicht vorgesehen. In manchen wissenschaftlichen Arbeiten zur Realisierung und Anwendung von Multi-Agenten-Systemen wird der Begriff "Selbstorganisation" stellvertretend für eine verteilte Planung genutzt, falls einzelne Agenten jeweils nur über ein klar abgegrenztes Wissen über mögliche Zustände und Aktionen verfügen [Fre10, S. 22]. Ein solches selbstorganisierendes Multi-Agenten-System für Montagesysteme wird von Frei et al. [FS11, Fre10] beschrieben. Auf Grund der verteilten Entscheidungsfindung kann durch diese Verfahren jedoch die Identifikation eines optimalen Handlungsplanes kaum gewährleistet werden (Anforderung A2).

Auch im Bereich der Produktionstechnik sind bereits eine Vielzahl von Ansätzen verfügbar, welche Multi-Agenten Systeme erfolgreich für unterschiedliche Anwendungsfelder einsetzen. Der überwiegende Teil der Anwendung von Planungstechniken im Kontext von Multi-Agenten-Systemen ist dabei im Bereich der Produktions(fein)planung und -scheduling zu finden [SHYN06, SWH06]. Einerseits existieren zahlreiche Ansätze, welche flexible Abläufe und deren verteilte Steuerung in Form eines Multi-Agenten-Systems, jedoch keine automatische Handlungsplanung zur Bestimmung dieser Abläufe vorsehen, wie dies z.B. in den Arbeiten von Possel-Dölken et al. [BFPD08, PD06] oder Lüder et al. [LPS⁺04, LKP⁺05] vorgeschlagen wird. Andererseits ermöglichen eine Vielzahl von Ansätzen ein automatisches Scheduling oder Sequenzieren von Produktionsaufträgen. Hierbei steht die Bestimmung von Reihenfolgen der Produkte und die (zeitliche) Zuordnung notwendiger Bearbeitungsschritte auf Produktionsressourcen im Mittelpunkt (vgl. hierzu auch Abschnitt 2.1.3). Hoffmann et al. [HTS⁺16, Hof17] schlagen ein Konzept vor, um eine Kopplung von Agenten auf der Feldebene sowie deren Informationsbereitstellung für Produktionsplanungssystemen in standardisierter Form zu ermöglichen. Um auf Grund der kombinatorischen Herausforderung durch unterschiedliche Produktionsaufträge eine Planung durchführen zu können, werden in Produktionsplanungssystemen typischer Weise Einschränkungen möglicher Reihenfolgen, welche sich durch die Steuerungsebene ergeben, abstra-

hiert. Eine explizite Berücksichtigung und detaillierte Planung feingranularer Softwarefunktionen der Steuerungsebene (d.h. Automatisierungsfunktionen) und deren Abhängigkeiten in Form von Sensorwerten erfolgt in diesen Arbeiten nicht.

Zusammenfassend existieren eine Vielzahl von Ansätzen im Bereich der Multi-Agenten-Systeme. Planungstechniken stellen insbesondere in Ansätzen aus dem Bereich der Künstlichen Intelligenz und reinen Softwareagenten einen wesentlichen und häufig adressierten Aspekt dar. Als ein exemplarisches Beispiel für eine Agentenarchitektur, welche explizit den Einsatz von Planungstechniken voraussetzt, wurde das BDI Paradigma in diesem Abschnitt eingeführt. Hierfür können unterschiedliche, in Abschnitt 3.1 bereits detailliert beschriebene, formale Planungskonzepte eingesetzt werden. Keiner der betrachteten Ansätze kann jedoch die gestellten Anforderungen vollständig adressieren. Insbesondere kann durch eine Verteilung des Planungsproblems zumeist keine Garantie für die Identifikation einer optimalen Lösung (Anforderung A2) gewährleistet werden. Während eine Vielzahl ausgereifter Verfahren im Bereich der Produktionstechnik existieren um Fragestellungen der Produktionsplanung bzw. Produktionsfeinplanung zu adressieren, berücksichtigen diese Ansätze die Anforderungen hinsichtlich der Bestimmung von Ablaufreihenfolgen feingranularer Automatisierungsfunktionen nicht ausreichend. Ferner gibt es nur wenige Arbeiten, welche eine Implementierung von Agenten für SPS-basierte Steuerungen berücksichtigen (Anforderung A4). Die Anwendung eines Planungsverfahrens zur Bestimmung entsprechender Abläufe der Aktionen bzw. Funktionen solcher Agenten für die Feldebene existiert noch nicht.

3.2.1.3. Kombination von Dienstorientierter Architektur und Multi-Agenten Systemen

In den vorangegangenen Abschnitten wurden detailliert sowohl dienstorientierte Architekturen als auch Multi-Agenten Systeme charakterisiert und im Kontext automatischer Planungsverfahren diskutiert. Durch eine Gegenüberstellung der architektonischen Eigenschaften von SOA und MAS schlussfolgern Ribeiro et al. [RBM08], dass es sich dabei um teils gegensätzliche, jedoch auch potentiell positiv ergänzende Paradigmen handelt. Erste Anwendungen dieses Konzepts aus dem Bereich der robotergestützten Montage [RBC08, BGIR09] sowie flexiblen Fertigungssystemen [HBL⁺08, HRL12] konnte dabei die Erwartungen hinsichtlich positiver Effekte bestärken.

Mendes e al. [MLRC09] fassen drei mögliche Kombinationen von SOA und MAS wie folgt zusammen: (a) Eine Middleware bietet mittels Gateways oder Proxys die Anbindung und Übersetzung zwischen beiden „Welten“. In diesem Fall stellen SOA und MAS vollständig separierte Technologien dar. Dieser Ansatz wird beispielsweise von Nguyen and Kowalczyk [NK05] verfolgt, um Dienste mittels eines Agentensystems zu verwalten. (b) Agenten werden als Dienste in einem service-orientierten Softwaresystem gekapselt. Von außen sind diese Dienste nicht von anderen Diensten im Softwaresystem zu unterscheiden (außer beispielsweise durch ein ggf. „intelligenteres“ Handeln) und können so direkt integriert werden. In diesem Fall kann SOA als vorherrschende Basistechnologie angesehen werden. Ein entsprechender Ansatz wird beispielsweise von Shen et al. [SGL06] beschrieben. Huhns [Huh03] begründet die Vorteilhaftigkeit dieser Kombinationsart dadurch, dass SOA prädestiniert ist, Softwarefunktionalität unabhängig zu entwickeln und zu betreiben, während im Gegensatz dazu der organisatorische Aspekt und die Kollaboration in MAS zu höheren Abhängigkeiten zwischen einzelnen Softwareelementen führt. (c) Eine integrative Kombination von Diensten innerhalb eines MAS, wobei Agenten als vorherrschende Basistechnologie betrachtet werden, wird in [MLRC09] vorgeschlagen. Dabei werden verfügbare

3. *Stand der Wissenschaft und Technik*

Dienste im System durch die Agenten genutzt; die Funktionalität von Agenten steht wiederum als Dienst zur Verfügung.

Die ersten beiden Kombinationsmöglichkeiten stellen die klassische, dienstorientierte Architektur in den Vordergrund. Entsprechend finden die in Abschnitt 3.2.1.1 beschriebenen Ansätze zur automatischen Planung in dienstorientierten Architekturen direkte Anwendung. Im Gegensatz dazu stellt die dritte Kombinationsmöglichkeit den Agenten in den Vordergrund. In diesem Fall kann durch Agenten eine automatische Planung und Dienstkomposition durchgeführt werden, um komplexe Funktionalität zu realisieren. Wie in Abschnitt 3.2.1.1 beschrieben, kann für die Dienstkomposition auf klassische Planungsalgorithmen, wie die zielbasierte Planung, zurückgegriffen werden. In diesem Fall werden Dienste als Aktionen im Sinne des Planungsverfahrens betrachtet. Andererseits kann die automatische Handlungsplanung von Agenten im Kontext von BDI betrachtet werden, wie im vorangegangenen Kapitel beschrieben. Dienste stellen in diesem Zusammenhang die für die Erreichung der Begehren eines Agenten zur Verfügung stehenden Aktionen dar. Hierfür können und werden ebenfalls die entsprechenden Planungsverfahren eingesetzt. Insgesamt kann dies auf einen triangulären Zusammenhang zwischen automatischer Dienstplanung, Handlungsplanung von Agenten und der technologischen Basis der Planungsverfahren zurückgeführt werden. Folglich kann, bei geeigneter Dienstdefinition bzw. Dienstimplementierung (als entsprechende Aktionen eines Agenten), die Handlungsplanung von Agenten und die Dienstplanung gegenseitig konzeptuell substituiert werden. Grundlage hierfür bilden jedoch stets die grundlegenden, theoretischen Planungsverfahren.

3.2.2. Anwendung von Planungsverfahren in der Anlagenautomatisierung

Während im vorangegangenen Abschnitt die Anwendung automatischer Planungsverfahren in unterschiedlichen, für die Realisierung von Industrie 4.0 bzw. cyber-physischer (Produktions-)Systeme als grundlegend eingeschätzte Arbeiten betrachtet wurde, wird in diesem Abschnitt die Anwendung von Planungsverfahren (unabhängig konkret zu Grunde gelegter Softwarearchitekturen) in der Anlagenautomatisierung untersucht. Die Untersuchung ist dabei in zwei Abschnitte gegliedert: In Abschnitt 3.2.2.1 wird eingangs die Handlungsplanung für Werkzeugmaschinen und die robotergestützte Montage betrachtet während in Abschnitt 3.2.2.2 die Bestimmung von Ausführungsreihenfolgen von Softwarefunktionen für SPS-basierte Steuerungssoftware (d.h. Automatisierungsfunktionen) näher betrachtet wird. Die Zuordnung der Arbeiten erfolgt dabei auf Basis ihrer Hauptausrichtung und ist nicht vollständig disjunkt. Da jedoch für die Untersuchung der verwandten Arbeiten stets auf die eingangs in dieser Arbeit beschriebenen Anforderungen zurückgegriffen wird, ist die Bewertung des Ansatzes unabhängig von dieser Kategorisierung.

3.2.2.1. Handlungsplanung für Werkzeugmaschinen und robotergestützte Montage

Im Zuge der Bestrebungen hinsichtlich einer computerintegrierten Fertigung (engl. computer integrated manufacturing) [Sch12] und der damit verbundenen, digitalen Kopplung von Entwicklungswerkzeuge wurde insbesondere in den Bereichen der Fertigung mechanischer Teile und der Montage die automatische Ableitung von Handlungsempfehlungen und (teils in neueren Ansätzen soweit möglich) die Synthese von entsprechendem Steuerungsprogrammen untersucht. Dabei werden (kontinuierlich) numerisch gesteuerte Werkzeugmaschinen adressiert; in der Montage werden robotergestützte Systeme und deren Steuerungssoftware betrachtet. In diesem Kontext

wurde auch die gewünschte Ableitung von Produkteigenschaften, beispielsweise auf Basis von Daten aus dem computergestützten Design (engl. computer-aided design) untersucht. Für die Anwendung dieser Produkteigenschaften (engl. features) in der Fertigungs- und Montageplanung wurde der Begriff des feature-based manufacturing [MNS96, CWH00] geprägt; aber auch der Begriff computerintegrierte Prozessplanung (engl. computer-aided process planning, CAPP) findet hierbei Anwendung [EN14].

Es existieren eine Vielzahl unterschiedlicher Ansätze für die Handlungsplanung von Werkzeugmaschinen und Montageroboter (siehe [XWN11, MGG98] für ausführliche Beschreibung und Diskussionen dieser Ansätze). Diese Ansätze basieren zumeist auf Algorithmen zur Trajektorienplanung, z.B. für Steuerung von Fräsmaschinen oder die Bewegungsplanung von Robotern. Für robotergestützte Montagesysteme wird in diesem Zusammenhang eine zusätzliche, abstraktere, diskrete Betrachtung der zur Verfügung stehenden Funktionalitäten – sogenannte Skills – genutzt, um gewünschten Produkteigenschaften bzw. deren gewünschten Zusammenbau zu erreichen [MW01, WBB⁺10]. Eine universelle Betrachtung diskreter Fertigungsprozesse mit logischer, diskreter Betrachtung der Steuerung und deren Realisierung mittels Speicherprogrammierbarer Steuerungen wird dabei zumeist nicht berücksichtigt. Folglich wird durch diese Ansätze insbesondere die Anforderung A4 nicht geeignet adressiert. Nichtsdestotrotz stellen einige Arbeiten Grundlagen im Bezug auf das in dieser Arbeit vorgeschlagene Planungsverfahren dar. Aus diesem Grund wird im Folgenden auf eine detaillierte Diskussion dieser Ansätze verzichtet, jedoch einige ausgewählte Ansätze diskutiert, welche für die hier vorliegende Arbeit von besonderer Relevanz sind.

Andersson et al. [ALFF11, ALF10, And06] beschreibt ein Verfahren zur automatischen Ableitung von Handlungssequenzen für den Wiederanlauf der Montage nach einem fehlerbedingtem Betriebsstillstand. Hierbei steht insbesondere die Synchronisation des internen Zustandes der Steuerungssoftware mit dem durch menschliche Manipulation geänderten Zustand des technischen Systems (zur Fehlerbehebung) im Vordergrund. Der technische Prozess findet dabei keine Berücksichtigung (Anforderung A1.1 und Anforderung A1.2). Auf Basis eines erweiterten, deterministischen, endlichen Automaten wird ausgehend von einem gegebenen Initialzustand mittels Controller Synthese [RW89] eine geeignete Wiederanlaufsequenz abgeleitet. Ziel des Ansatzes war dabei die Möglichkeit der Implementierung des Algorithmus auf Speicherprogrammierbaren Steuerungen, welche in der industriellen Praxis für die übergeordnete Steuerung (engl. Supervision) der Montageroboter eingesetzt werden. Entsprechend ist Anforderung A4 nicht geeignet adressiert.

Bergagård et al. [BFF15, BF14, BF13, Ber15] adressieren ebenfalls die Berechnung von Schaltsequenzen für den Wiederanlauf von Fertigungssystemen auf Basis endlicher Automaten. Der Ansatz basiert dabei auf den von Andersson et al. entwickelten Konzepten, verallgemeinert diese jedoch, um zusätzliche Funktionalitäten, wie z.B. eine automatische Ableitung alternativer Handlungssequenzen und den Wiederanlauf mehrerer Ressourcen. Ferner können durch die vorgenommene Verallgemeinerung beliebige, insbesondere effizientere Synthesealgorithmen eingesetzt werden. Durch das in der Arbeit von Bergagård et al. zu Grunde gelegte Ziel des Wiederanlaufs wird Anforderung A1.3 vollständig erfüllt. Eine integrierte Betrachtung des technischen Prozesses für die Fertigung von Produkten in der Massenfertigung wird jedoch nicht betrachtet (Anforderung A1.1 und Anforderung A1.2). Eine Anbindung an die Steuerungsebene des Maschinen- und

3. *Stand der Wissenschaft und Technik*

Anlagenbaus ist jedoch durch die genutzte, formale Grundlage der Supervisory Control Theory prinzipiell möglich (Anforderung A4).

Kempf et al. [KHB09, BKH08, Kem10] schlagen ebenfalls ein modellbasiertes Verfahren zur automatischen Bestimmung von Handlungsabläufen für die robotergestützte Montage vor. Auf Basis einer Beschreibung diskreter Handhabungsoperationen der Roboter wird hierbei entsprechend dem Grundgedanken der Montageplanung ein Suchalgorithmus vorgeschlagen, um geeignete Montagesequenzen aller in der Montagezelle verfügbaren Roboter und deren notwendige Koordination für die Montage eines einzelnen Produktes zu identifizieren. Unterschiedliche Teilfunktionen innerhalb der identifizierten Handlungspläne entsprechend Anforderung A1 werden hierbei jedoch nicht berücksichtigt. Das für die Handlungsplanung genutzte Modell ist in Form logischer Axiome beschrieben. Eine automatische Generierung dieses Modells auf Basis von Entwicklungsdaten (CAD) ist berücksichtigt (Anforderung A5).

Ein zweistufiger Ansatz für die robotergestützte Montageplanung wird in Ewert et al. [ETK⁺13, ESJ14, Ewe14] verfolgt. Auf Basis von CAD Daten wird ein Planungsgraph abgeleitet, welcher mögliche Montagesequenzen und deren Abhängigkeiten repräsentiert. Zur Laufzeit eines Montagesystems wird dieser Graph für eine dynamische Ableitung auszuführender Montagesequenz genutzt, um auf unvorhergesehene Zustände reagieren zu können. Das durch Ewert et al. beschriebene Verfahren berücksichtigt dabei, wie auch zuvor beschriebene Ansätze, jedoch nicht die spezifische Struktur von Handlungsplänen entsprechend Anforderung A1 – so wird ausschließlich die Zielerreichung für die Montage eines Gutes berücksichtigt; eine gleichzeitige Berücksichtigung etwaiger Startzustände oder einer Grundstellungsfahrt erfolgen nicht. Eine Anwendung des Konzeptes für Steuerungssoftware nach IEC 61131-3 (Anforderung A4) wurde ebenfalls nicht betrachtet.

Backhaus et al. [Bac16, BR15, BR13] schlagen ein adaptierbares, aufgabenorientiertes Programmiersystem für Montagesysteme vor, um den Aufwand für Veränderung von Abläufen gering zu halten (analog zu Anwendungsszenario 1; vgl. Abschnitt 1.1.1). Hierbei wird auf das Konzept der sogenannten Skills zurückgegriffen. Das vorgeschlagene Programmiersystem ist dabei vollständig modellbasiert (Anforderung A5). Ein zu Grunde gelegtes Vorgehensmodell ermöglicht einerseits die schrittweise Verfeinerung von Skills bis diese letztendlich durch eine Programmiersprache implementiert werden können und andererseits die manuelle Definitionen deren Ablaufreihenfolgen [Bac16, BR15]. Dabei können unterschiedliche Teilfunktionen im Sinne der Anforderung A1 abgebildet und realisiert werden; eine automatisierte Bestimmung notwendiger Ablaufreihenfolgen erfolgt nicht, weshalb Anforderung A1 lediglich partiell erfüllt ist. Wie in [Bac16, BR13] beschrieben, kann das vorgeschlagene Konzept für SPS-basierte Montagemodule – insbesondere auch IEC 61131-3 basierte Steuerungen – genutzt werden (Anforderung A4).

Zusammenfassend lässt sich feststellen, dass im Bereich der Werkzeugmaschinen sowie der robotergestützten Montage zahlreiche Ansätze für eine automatische Handlungsplanung existieren. Diese Ansätze setzen dabei häufig graph-basierten Modelle auf Basis diskreter Aktionen ein und zeigen dadurch auf, dass sich kontinuierliche, kinematisch und regelungstechnisch komplexen Aufgabenstellungen in geeigneter Weise für die Anwendung einer (klassische) Handlungsplanung diskretisieren lassen. Die Planung von Handlungsabläufen von Softwarefunktionen SPS-basierte Steuerungen von Maschinen und Anlagen wird in diesen Arbeiten jedoch nicht betrachtet. Forschungsarbeiten mit entsprechenden Fokus werden im nachfolgenden Kapitel den gestellten Anforderungen gegenübergestellt.

3.2.2.2. Ansätze zur Bestimmung der Ausführungsreihenfolge von Softwarefunktionen SPS-basierter Steuerungssoftware

Neben den bereits zuvor beschriebenen Anwendungen zur Flexibilisierung and Steigerung der Anpassbarkeit von Produktionssystemen existieren einige wenige Ansätze, welche diskrete Steuerungssysteme für die Fertigungsautomatisierung betrachten. Diese werden im Folgenden beschrieben und den gestellten Anforderungen gegenübergestellt.

Im Rahmen des Projektes OntoReA³³ [LZVM11, VMLK11] wurde ein Konzept zur formalen Darstellung von Wissen für Agenten erarbeitet, welches explizit die Steuerungsebene sowie Sensorik und Aktorik eines Fertigungssystems berücksichtigt. In diesem Wissensmodells werden mögliche Situationen (d.h. Zustände des Systems und logistischen Prozesses) beschrieben. Auf Basis dieser Daten wird ein geeignetes Planungsproblem abgeleitet (beschrieben durch eine logikbasierte Aktionssprache), um eine Sequenz zur Herstellung eines gegebenen Zustandes (d.h. zielbasierte Planung) automatisch abzuleiten. Dabei wurde auf bereits existierende Planungsalgorithmen zurückgegriffen. Für die Steuerungssoftware wird der IEC 61499 Standard zu Grunde gelegt, weshalb Anforderung A4 durch diesen Ansatz nicht geeignet adressiert wird. Ferner wird durch die im Projekt adressierten Anwendungsfälle der zu realisierende Produktionsprozess nicht explizit berücksichtigt (Anforderung A1.1); auch werden durch den innerhalb dieses Projektes entwickelten Ansatz für die Massenfertigung spezifische Charakteristika (Anforderung A1.2) ebenfalls nicht geeignet adressiert.

In Mechs et al. [MMLP12, MLPM13a, Mec13] wird ein Ansatz vorgeschlagen, um Fertigungssysteme systematisch in einen energieeffizienten Zustand (eine Art Standby-by) zu versetzen. Geeignete Schaltsequenzen werden dabei auf Basis eines vernetzten, kostenannotierten, zeitbehafteten Automaten (engl. priced timed automata) [BLR05] automatisch abgeleitet. Der Ansatz unterliegt der Annahme, dass sich ein betrachtetes Fertigungssystem bereits in Stillstand (sog. nicht-produktive Phase) befindet. Folglich müssen bei der Bestimmung der notwendigen Schaltsequenz keine Aspekte des technischen Prozesses berücksichtigt werden, weshalb Anforderung A1 – insbesondere Anforderung A1.1 und Anforderung A1.2 – nicht geeignet adressiert werden.

Im Projekt SkillPro³⁴ [PSA⁺14, PSB13, PSA⁺15] wird ein Konzept für eine vollständig modellbasierte Produktionsfeinplanung erarbeitet. Zielsetzung ist eine Berücksichtigung etwaiger Veränderungen des Produktionssystems innerhalb der Produktionsfeinplanung ausschließlich auf Basis des zu Grunde gelegten Modells. Es werden dabei ausschließlich „flexible“ Komponenten und deren Steuerungssysteme, wie Roboter und NC-gesteuerte Systeme betrachtet. Für die Modellierung der Fähigkeiten dieser Komponenten wird auf existierende Konzepte aus der Handlungsplanung in der Robotik und der Montage (siehe auch Kapitel 3.2.2.2), den Skills, zurückgegriffen. Das Granularitätslevel der Skills entspricht dabei der Produktionsfeinplanung auf Abstraktionsstufe des MES [PSB13]. Wie durch Pfrommer et al. [PSA⁺14] beschrieben, erfolgt die Formulierung des zielbasierten Planungsproblems auf theoretischer Grundlage des Situation Calculus. Für die automatische Planung wird auf ein bereits existierendes Planungswerkzeug („Fast Downward

³³Das Projekt „Ontology-based Reflective World Model for Autonomous Agents“ (OntoReA) wurde im Rahmen der FIT-IT Initiative unter der Fördernummer FFG 815132 im Zeitraum 2008 – 2011 gefördert. Träger der FIT-IT-Initiative (Forschung, Innovation, Technologie – Informationstechnologie) war das österreichische Bundesministeriums für Verkehr, Innovation und Technologie (BMVIT).

³⁴Das Projekt SkillPro (*Skill-based Propagation of „Plug&Produce“-Devices in Reconfigurable Productions Systems by AML*) wird im Rahmen des Framework Programme 7 der Europäischen Union im Zuge der Bestrebungen *Factory of the Future* gefördert. Projektlaufzeit 10/2012 – 09/2015.

3. Stand der Wissenschaft und Technik

Planning System“ [Hel06]) zurückgegriffen, weshalb unter anderem auch Anforderung A2 durch diesen Ansatz nicht adressiert wird.

Zhang et al. [ZKL⁺13, ZKL⁺15] schlagen ein Verfahren zur Modellierung und Verifikation dynamisch konfigurierbarer, diskreter Steuerungssysteme vor. Für die Modellbildung des Steuerungsverhaltens und dessen dynamische Rekonfigurationsmöglichkeiten werden spezialisierte, zeitbehaftete Bedingungs-/Ereignisnetze vorgeschlagen. Hierbei wird ein explizites Modell aller Steuerungsabläufe erstellt. Die automatische Prüfung notwendiger Verhaltensanforderungen erfolgt mittels automatischer Modellprüfung auf Basis des Erreichbarkeitsgraphen. Aktuell wird das Verfahren nicht für die Ableitung automatischer Handlungsabläufe genutzt. Dies wäre jedoch auf Basis einer Reformulierung der zu verifizierenden Eigenschaft denkbar (Handlungsplanung als Erfüllbarkeitsproblem). Zwar ist die Anwendung des Verfahrens für Speicherprogrammierbare Steuerungen nach IEC 61131 nicht explizit beschrieben, auf Grund der Charakteristiken des Modells liegt jedoch eine mögliche Eignung nahe. Die Erfüllung von Anforderung A1.2 und Anforderung A1.3 werden durch den Ansatz jedoch nicht adressiert.

Castillo et al. [CFOG00b, CFOG00a] adressieren explizit die Planung der Steuerungsabläufe von Produktionssystemen mittels POP. Sie setzen dabei auf eine adaptierte STRIPS Notation hinsichtlich des genutzten Aktionsmodells. Für die Anbindung an die Steuerungssoftware wird auf GRAFCET³⁵ zurückgegriffen. Castillo et al. setzen in ihrem Ansatz auf eine vollständig zielbasierte Formulierung des Planungsproblems. Zwar berücksichtigt der vorgeschlagene Ansatz die für die Fertigung eines Produktes notwendigen Schritte, wobei die Zieldefinition durch den Zielzustand erfolgt (Anforderung A1.1), es wird jedoch angenommen, dass der Initialzustand des Systems Bestandteil des für die Fertigung eines Gutes notwendigen Prozesses darstellt. Somit ist Anforderung A1.3 lediglich teilweise erfüllt. Auch werden keine zusätzlichen Funktionalitäten, wie sie für die Massenfertigung notwendig sind, innerhalb der Planung berücksichtigt, weshalb Anforderung A1.2 nicht erfüllt ist. Innerhalb der Spezifikation von GRAFCET [DIN02] wird die Ablaufsprache explizit als eine mögliche Implementierung angeführt, weshalb Anforderung A4 als erfüllt betrachtet wird. Eine Integration in einen modellbasierten Entwicklungsansatz (Anforderung A5) wird nicht betrachtet.

Klein et al. [KL93, KB91] beschreiben einen Ansatz zur Planung ausschließlich sequentieller Steuerungsabläufe. Analog zu dem Ansatz von Castillo et al. wird GRAFCET für die Implementierung des identifizierten Plans angewandt (Anforderung A4 erfüllt), beschränken sich dabei jedoch auf eine spezielle Klasse von Planungsproblemen mit einer Vielzahl von Restriktionen, wie u.a. der Einschränkung auf ausschließlich boolesche Zustandsvariablen sowie der Tatsache, dass Aktionen lediglich eine einzelne Zustandsvariable beeinflussen. Dies schränkt die Anwendbarkeit im Maschinen- und Anlagenbau signifikant ein³⁶. Der Ansatz adressiert insbesondere Sequenzen um ein System vom Startzustand in einen Zielzustand für den Beginn der Produktion zu versetzen. Entsprechend erfolgt die Beschreibung des Planungsproblems in klassischer Form für eine zielbasierte Planung. Während hierdurch Anforderung A1.3 geeignet adressiert wird,

³⁵GRAFCET stellt eine standardisierte Spezifikationssprache für die Darstellung von Ablaufbeschreibungen für Steuerungen dar [DIN02].

³⁶Betrachtet man beispielsweise das Ausfahren eines pneumatischen Zylinders mit zwei binären Endlagensensoren, welche den Zustand des Zylinders beschreiben, als mögliche Aktion innerhalb eines Planungssystems. Die Veränderung zweier Zustandsvariablen, wie es im Falle des Ausfahrens eines pneumatischen Zylinders durch die Endlagensensoren der Fall wäre, kann mittels dieses Planungssystems nicht bzw. nur auf Umwegen abgebildet werden.

werden Anforderung A1.2 und Anforderung A1.1 nicht ausreichend adressiert. Eine Integration in einen modellbasierten Entwicklungsansatz, wie durch Anforderung A5 gefordert, wird nicht vorgeschlagen.

Innerhalb dieses Abschnittes konnte gezeigt werden, dass bereits zahlreiche Ansätze existieren, welche speziell die Steuerungsebene des Maschinen- und Anlagenbaus adressieren. Dabei wird eine Anbindung der Verfahren zur Ableitung von Steuerungsabläufen an Entwicklungswerkzeuge und Sprachen wie GRAFCET bereits umgesetzt und dadurch die gestellte Anforderung A4 (zumindest teilweise) adressiert. Alle Ansätze setzen dabei, unabhängig von der gewählten formalen Basis, auf die Suche von geeigneten Abläufen auf Basis gegebener Start- und Zielzustände, weshalb diese der Anforderung A1 und dessen kombinierte Forderung einerseits nach der Berücksichtigung von initialen Zuständen (Anforderung A1.3), aber auch des notwendigen Fertigungsprozesses (Anforderung A1.1) sowie etwaiger, zusätzlicher Funktionalität für die Massenfertigung eines Gutes (Anforderung A1.2) nicht gerecht werden.

3.3. Zusammenfassung

Eingangs in diesem Kapitel wurden unterschiedliche Ansätze zur Formulierung des Planungsproblems und darauf basierende algorithmische Lösungsansätze beschrieben. Die vorgestellten Ansätze unterscheiden sich insbesondere in der zu Grunde gelegten Abstraktionsebene sowie deren formalen Basis. Im Rahmen der Diskussion unterschiedlicher Ansätze wurde letztendlich aufgezeigt, dass alle Ansätze zur Formulierung von Planungsproblemen auf eine Suche innerhalb eines Graphen zurückgeführt werden können. Entsprechend stellen graph-basierte Ansätze zur Formulierung des Planungsproblems, wie dies bei S/T-Netzen der Fall ist, die grundlegende Basis dar, während alternative (abstraktere) Ansätze darauf abgebildet werden können.

Im Anschluss daran wurde gezeigt, dass automatische Planungsverfahren bereits eine breite Anwendung in unterschiedlichen Einsatzgebieten, Anwendungsfällen und zu Grunde gelegten Softwarearchitekturen, auch im Bereich der Produktionsautomatisierung, gefunden haben. Es wurde gezeigt, dass Ansätze aus dem Bereich der Multi-Agenten-Systeme und dienstorientierten Architekturen, welche automatische Planungsverfahren einsetzen, die gestellten Anforderungen hinsichtlich der Anwendbarkeit für die Entwicklung (z.B. Anforderung A5) und/oder den Betrieb (z.B. Anforderung A1) der Steuerungsebene für die Maschinen- und Anlagenautomatisierung zumeist nicht geeignet adressieren können. Nichtsdestotrotz konnte bei dieser Untersuchung identifiziert werden, dass klassische Planungsverfahren ein mögliches Mittel darstellen, um die zunehmend steigende Forderung nach Anpassbarkeit und Flexibilität der Produktionsautomatisierung und damit letztendlich der Steuerungsebene zu adressieren. Ferner wurde aufgezeigt, dass eine Vielzahl sehr weit fortgeschrittener Forschungsansätze für den Einsatz automatischer Planungsverfahren existieren, wie z.B. für die robotergestützten Montage oder im Bereich der Werkzeugmaschinen. Die Forderung nach Anwendbarkeit der Ansätze für diskrete Steuerungssoftware, welche auf Speicherprogrammierbarer Steuerungen nach IEC 61131-3 ausführbar ist, wird dabei jedoch nicht ausreichend adressiert (vgl. Abschnitt 3.2.2.1). Innerhalb dieses Kapitels wurden abschließend die Analyse zahlreicher weiterer Ansätze beschrieben, welche explizit auf die Steuerungsebene für die Maschinen- und Anlagenautomatisierung adressieren. Die Anbindung bzw. Einbettung des Konzeptes an die Steuerungssoftware ist dabei zumeist explizit berücksichtigt (Anforderung A4). Diese Ansätze zeichnen sich dabei auch durch eine Anbindung

3. *Stand der Wissenschaft und Technik*

an entsprechende Entwicklungswerkzeuge für Steuerungssoftware (z.B. für GRAFCET) aus. Es zeigt sich jedoch, dass insbesondere die Kombination der Teilanforderungen von Anforderung A1 d.h. Anforderung A1.1 bis Anforderung A1.3, nicht erfüllt werden können, weil eine klassische, zielbasierte Formulierung und Lösung des Planungsproblems hierbei nicht ausreichend ist. Leider werden in den wenigsten Beiträgen Untersuchungen hinsichtlich der Leistungsfähigkeit – insbesondere der Skalierbarkeit – der vorgeschlagenen Ansätze und Algorithmen beschrieben. Daher ist eine Bewertung der Anforderung A3 nur sehr beschränkt möglich. Tabelle 3.1 fasst diese Sachverhalte visuell zusammen, indem es die der Zielsetzung dieser Arbeit im Hinblick auf die Erfüllung der Anforderungen nahestehenden Arbeiten gegenüberstellt. Abschließend kann zusammengefasst werden, dass kein bekannter Ansatz existiert, welcher eine automatische Handlungsplanung ermöglicht und dabei die domänenspezifischen Charakteristika der Steuerungsebene für die Maschinen- und Anlagenautomatisierung, wie sie im Zuge der Anforderungsermittlung in Abschnitt 2.2 identifiziert wurden, geeignet adressiert.

Tabelle 3.1.: Zusammenfassung der Analyseergebnisse verwandter Arbeiten zur automatischer Planung von Handlungsabläufen im Maschinen- und Anlagenbau.

Referenz	Erfüllungsgrad der Anforderungen						
	A1.1	A1.2	A1.3	A2	A3	A4	A5
	Automatische Identifikation des technischen (Fertigungs-)Prozesses	Automatische Identifikation der Grundstellungsfahrt	Berücksichtigung des Initialzustandes der Fertigungsanlage	Selektion eines Ablaufplanes mittels Gütekriterium (Optimalität)	Zeit- und Skalierbarkeitsverhalten des Planungsalgorithmus	Eignung für Steuerungssoftware und Steuerungen nach IEC 61131-3	Möglichkeit der Integration in einen modellbasierten Entwicklungsansatz
Anderson et al. [ALFF11, ALF10, And06]	-	-	+	-	x	+	-
Bergagård et al. [BFF15, BF14, BF13, Ber15]	-	-	+	-	x	+	-
Castillo et al. [CFOG00b, CFOG00a]	+	-	o	-	x	o	-
Ewert et al. [ETK ⁺ 13, ESJ14, Ewe14]	+	-	+	+	o	-	+
Frei et al. [FS11, Fre10]	x	+	+	o	x	-	+
Kempf et al. [KHB09, BKH08, Kem10]	+	+	-	o	o	-	+
Klein et al. [KL93, KB91]	+	-	+	-	x	o	-
Mechs et al. [MMLP12, MLPM13a, Mec13]	-	-	+	+	+	-	+
OntoReA Projekt [LZVM11, VMLK11]	-	-	+	-	x	-	x
SkillPro Projekt [PSA ⁺ 14, PSB13, PSA ⁺ 15]	+	-	x	-	x	-	+
Zhang et al. [ZKL ⁺ 13, ZKL ⁺ 15]	+	-	-	-	x	o	x
Backhaus et al. [Bac16, BR15, BR13]	o	o	o	x	x	+	+

Bewertungsskala: + erfüllt, o teilweise erfüllt, - nicht erfüllt, x keine Angabe

4. Bestimmung der Ablaufreihenfolge von Automatisierungsfunktionen

Um die spezifischen Anforderungen an ein Verfahren zur Planung der Ablaufreihenfolge von Automatisierungsfunktionen geeignet zu adressieren, wird in diesem Abschnitt ein neues Planungsverfahren namens HiTraP-AT (**H**ierarchical **T**ransition **P**lanner for **A**utomation **T**echnology) vorgeschlagen. Abbildung 4.1 gibt einen Überblick über die konzeptuellen Bausteine des HiTraP-AT Planungsverfahrens.

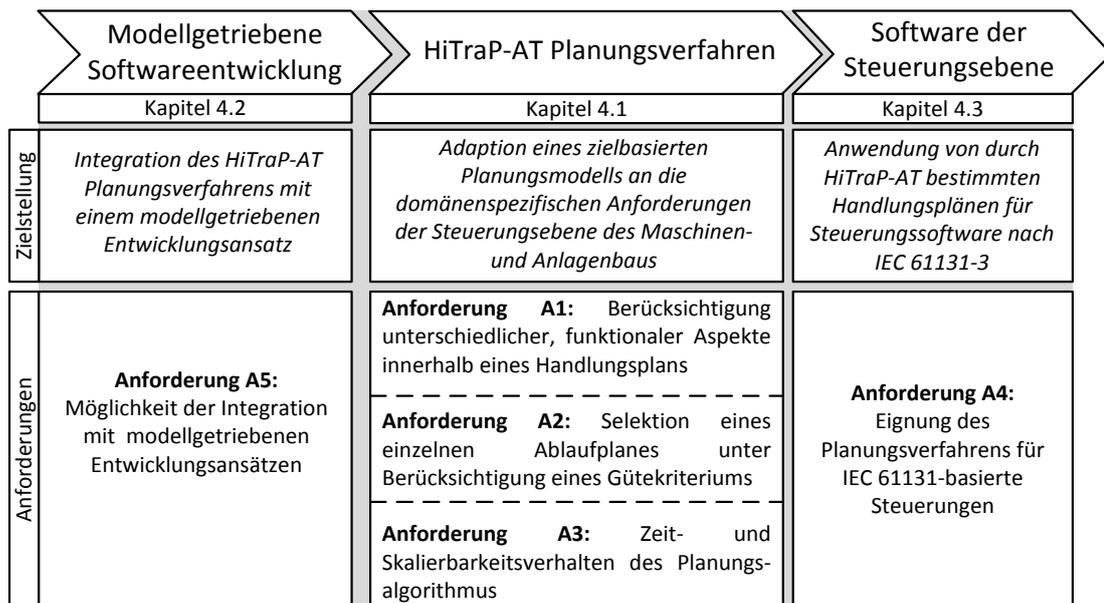


Abbildung 4.1.: Gegenüberstellung von Anforderungen und konzeptionellen Lösungsbausteinen für die Bestimmung von Ablaufreihenfolgen von Steuerungsfunktionen für die Maschinen- und Anlagenautomatisierung.

Das HiTraP-AT Planungsverfahren zur Bestimmung der Ablaufreihenfolgen von Automatisierungsfunktionen wird in Abschnitt 4.1.1 beschrieben. Neben spezifischen funktionalen Anforderungen an das Planungsverfahren sowie dessen Leistungsfähigkeit sind für ein geeignetes Verfahren zur Bestimmung von Ablaufreihenfolgen weitere Anforderungen zu berücksichtigen (siehe Abschnitt 2.2): zum einen ist die Möglichkeit der Integration des Planungsverfahrens in modellgetriebene Entwicklungsansätze gefordert und zum anderen dessen Anwendbarkeit für IEC 61131-3 basierte Maschinen- und Anlagensteuerungen. Diese hierfür notwendigen konzeptuellen Bestandteile werden entsprechend in Abschnitt 4.2 und Abschnitt 4.3 beschrieben.

4.1. Das HiTraP-AT Planungsverfahren (Anforderungen A1 bis A3)

Gegenstand dieser Dissertation stellt ein konfigurierbares Planungsverfahren für die Steuerungsebene des Maschinen- und Anlagenbaus – das HiTraP-AT Planungsverfahren – dar, welche im

4. Bestimmung der Ablaufreihenfolge von Automatisierungsfunktionen

Rahmen dieses Abschnitts beschrieben wird. Um gestellte, spezifische Anforderungen geeignet adressieren zu können, ist ein adäquates Modell notwendig, welches die Bestimmung entsprechender Ablaufreihenfolgen ermöglicht. Dieses interne Planungsmodell, welches in Abschnitt 4.1.1 detailliert beschrieben wird, berücksichtigt dabei zum einen die durch Anforderung A1 geforderte Charakteristik eines validen Handlungsplanes; adressiert zum anderen aber auch die Anforderung an eine gute Leistungsfähigkeit hinsichtlich Zeit- und Skalierbarkeitsverhalten des Planungsverfahrens (Anforderung A3) durch eine spezifische Repräsentation von Anlagenzuständen. Dabei ermöglicht das interne Planungsmodell auch die Repräsentation möglicher paralleler Ausführungen von Automatisierungsfunktionen und stellt dadurch die Grundlage bereit, um einen guten Handlungsplan zu identifizieren (Anforderung A2). Wie bei klassischen Planungsverfahren üblich, wird die Definition eines Planungsproblem genutzt, um den Planungsgegenstand – das betrachtete Fertigungssystem – sowie notwendige Informationen über die in der Planung zu berücksichtigenden Funktionalität des Fertigungssystems an das Planungsverfahren zu übergeben. In Abschnitt 4.1.2 wird daher eine (an die domänenspezifischen Anforderungen der Steuerungsebene des Maschinen und Anlagenbaus angepasste) Beschreibung des Planungsproblem vorgeschlagen, welche die für das interne Planungsmodell des HiTraP-AT Planungsverfahren notwendigen Informationen enthält. Um zum einen die Anforderung an einen guten bzw. optimalen Handlungsplan (Anforderung A2) zu adressieren und zum anderen geeignete zeitliche Leistungswerte zu erreichen, wie dies in Anforderung A3 gefordert ist, wird das in dieser Arbeit vorgeschlagene HiTraP-AT Planungsverfahren als Lineares Optimierungsproblem aufgefasst. Die Reformulierung des Planungsmodells sowie die Formulierung des Planungsproblems als lineares Programm mit Optimierungscharakter wird in Abschnitt 4.1.3 detailliert beschrieben.

4.1.1. Das interne HiTraP-AT Planungsmodell

Wie bei modellbasierten Planungsverfahren üblich, wird eine geeignete interne Repräsentation valider und damit möglicher Handlungspläne für die Bestimmung der Ablaufreihenfolgen von Aktionen (hier Automatisierungsfunktionen) genutzt (vgl. Abschnitt 3.1). Dieses interne HiTraP-AT Planungsmodell repräsentiert dabei alle korrekten Handlungspläne in einer Form, so dass eine automatische Handlungsplanung effizient – d.h. entsprechenden geforderter Leistungswerten – ermöglicht wird (vgl. Anforderung A3). Die spezifische Struktur von Handlungsplänen, wie dies im Rahmen von Anforderung A1 beschrieben wurde, ist dabei maßgeblich um die Korrektheit der Handlungspläne sicherzustellen. Hierzu wird durch HiTraP-AT das Konzept zusammenhängender Handlungsräume genutzt, welches das von zielbasierten Planungsverfahren zu Grunde gelegte Konzept des Handlungsraumes entsprechend Anforderung A1 erweitert. Dies wird detailliert im nachfolgenden Abschnitt 4.1.1.1 beschrieben. Hierbei wird von einer expliziten Repräsentation aller validen Handlungsräume ausgegangen, was die Herausforderung einer Zustandsraumexplosion bei einer im Maschinen- und Anlagenbau üblichen, großen Anzahl von Aktionen und Zuständen, verschärft. Um geforderte Leistungswerte des Planungsverfahrens adressieren zu können sowie die Repräsentation parallel ausführbarer Automatisierungsfunktionen zu ermöglichen – was die Grundlage für die Identifikation guter bzw. optimaler Handlungspläne darstellt (Anforderung A2) – wird für das interne HiTraP-AT Planungsmodell ferner das Konzept der konditionalen Hierarchisierung vorgeschlagen. Im Gegensatz zu einer kombinatorischen Betrachtung des Handlungsraumes wird hierbei eine domänenspezifische Repräsentation partiell-geordneter Pläne

ermöglicht. Das Konzept der konditionalen Hierarchisierung wird in Abschnitt 4.1.1.2 detailliert beschrieben. Die Kombination dieser beiden Konzepte, um letztendlich alle Anforderungen A1 bis A3 durch das interne HiTraP-AT Planungsmodell zu adressieren, wird in Abschnitt 4.1.1.3 beschrieben.

4.1.1.1. Zusammenhängende Handlungsräume

Wie in Kapitel 3 identifiziert, stellt insbesondere die Adressierung der Anforderung A1 hinsichtlich spezifischer, nicht sequentieller funktionaler Bestandteile valider Handlungspläne eine noch nicht (hinreichend) gelöste Herausforderung dar. Dies ist insbesondere darauf zurückzuführen, dass die Identifikation von Handlungsplänen, welche die Anforderungen A1.1 bis A1.3 innerhalb eines einzelnen Handlungsraumes erfüllen, schwer adressiert werden kann. Klassische (zielbasierte) Planungsverfahren setzen jedoch eine Formulierung des Planungsproblems auf Basis eines einzelnen Handlungsraumes und einzelner Zielsetzungen. Nichtsdestotrotz sind diese Verfahren wissenschaftlich für unterschiedliche Anwendungsfelder bereits intensiv untersucht worden (vgl. insbesondere Abschnitt 3.2). Um die gestellte Anforderung A1 geeignet zu adressieren und dabei auf die konzeptionellen Vorarbeiten klassischer, zielbasierter Verfahren zurückgreifen zu können, adaptiert HiTraP-AT das Konzept des Handlungsraumes, wie es durch zielbasierte Planungsverfahren bereits erfolgreich angewandt wird und erweitert dieses Konzept derart, dass unterschiedliche, charakteristische Merkmale der Steuerungsebene (entsprechend Anforderung A1) berücksichtigt werden können. Ausgehend von existierenden, zielbasierten Verfahren, welche auf einem einzelnen Handlungsraum basieren, werden in HiTraP-AT separate, miteinander spezifisch verknüpfte Handlungsräume genutzt. Auf Grund des Zusammenhangs zwischen Handlungsraum und Plan ergibt sich daraus analog eine spezifische Struktur von mit einander verknüpften Handlungsplänen. Abbildung 4.2 stellt die grundlegenden Elemente zusammenhängender Handlungspläne schematisch dar.

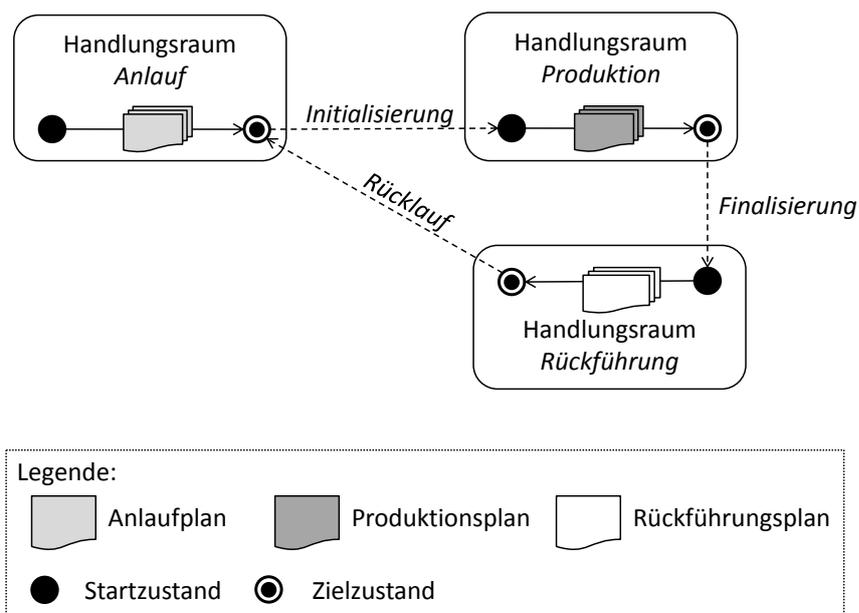


Abbildung 4.2.: Schematische Darstellung der Grundstruktur des HiTraP-AT Modells.

4. Bestimmung der Ablaufreihenfolge von Automatisierungsfunktionen

Um Anforderung A1.1 adressieren können, beschreibt der *Produktionsplan* die Aktionen sowie deren notwendige Reihenfolge zur Bearbeitung eines Werkstücks. Ausgehend von einer (durch den Planer ebenfalls zu bestimmenden) Grundstellung realisiert der Produktionsplan den technischen Prozess zur Fertigung eines einzelnen gewünschten Gutes. Um den Handlungsplan Produktion identifizieren zu können, wird ein entsprechendes Modell benötigt, welche durch das betrachtete System mögliche Handlungspläne beschreibt und letztendlich zur Identifikation bzw. Selektion eines geeigneten Handlungsraumes dient. Dies wird in HiTraP-AT durch den *Handlungsraum Produktion* adressiert, welcher ausschließlich Aktionen und deren mögliche Reihenfolgen zur Fertigung von Werkstücken enthält und dabei Anforderung A1.1 adressiert. Da innerhalb dieses Handlungsraumes eine Bearbeitung des Werkstückes stattfindet, müssen neben Einschränkungen bedingt durch das technische System auch solche Einschränkungen berücksichtigt werden, die durch den zu realisierenden technischen Prozess bzw. das Werkstück gegeben sind (vgl. Anforderung A1.1). Der Zielzustand des Handlungsraumes Produktion berücksichtigt dabei insbesondere den Zielzustand des zu fertigenden Werkstücks, da nach Ausführung eines durch einen Planer bestimmten Produktionsplanes der technische Prozess zur Fertigung eines einzelnen Gutes realisiert ist und damit das Werkstück sich in seinem Zielzustand befindet.

Wie in Anforderung A1.2 beschrieben, ist für die Massenfertigung von Produkten ggf. die Ausführung weiterer bzw. zusätzlicher Funktionalitäten notwendig, um nach Ausführung des Produktionsplanes eine Fahrt in die Grundstellung zu realisieren. Um dies zu adressieren und ein technisches System ausgehend von dem durch Ausführung des Produktionsplans erreichten Systemzustand wieder in den Zustand zu versetzen, um einen Produktionsplan erneut ausführen zu können (d.h. die Grundstellung), wird der *Rückführungsplan* benötigt. Der zur Bestimmung des Rückführungsplans notwendige *Handlungsraum Rückführung* berücksichtigt dabei ausschließlich Aktionen, welche ohne Werkstück durchgeführt werden können. Der Übergang vom Zielzustand des technischen Systems nach Ausführung des Produktionsplans hin zum Startzustand der Rückführung wird als *Finalisierung*³⁷ bezeichnet. Ausgehend vom Zielzustand des Rückführungsplans kann der Produktionsplan (re-)initialisiert werden.

Klassische, zielbasierte Planungsverfahren, wie sie im vorangegangenen Kapitel ausführlich diskutiert wurden, ermöglichen die Ableitung eines einzelnen Handlungsplans, um ein gegebenes System vom Initial- in den Zielzustand zu versetzen. Auf Basis dieser Formulierung ist es mit existierenden Planungsverfahren grundsätzlich möglich, einen Handlungsplan zu bestimmen, um ausgehend von einem gegebenen Initialzustand (siehe Anforderung A1.3) des technischen Systems und Produktes, den Zielzustand eines Produktes zu erreichen (vgl. beispielsweise Castillo et al. [CFOG00b, CFOG00a], siehe hierzu auch Abschnitt 3.2.2.2). Unter den zuvor beschriebenen Sachverhalten, insbesondere der nach Anforderung A1.2 notwendigen Rückführung zur Realisierung der Grundstellungsfahrt nach Fertigung eines Werkstücks ist eine solche Formulierung nicht ausreichend, da der Initialzustand eines Systems nicht notwendiger Weise Bestandteil eines Produktionsplanes sein muss. Folglich ist ein zusätzlicher Handlungsplan notwendig. Der *Handlungsraum Anlauf* dient dabei der Adressierung der gestellten Anforderung A1.3 hinsichtlich der Berücksichtigung von Initialzuständen von Maschinen. Ein *Anlaufplan* stellt dabei einen Handlungsplan dar, der den Initialzustand eines betrachteten (technischen) System so manipuliert, dass ein optimaler Startzustand für die Produktion (d.h. Grundstellung) erreicht werden kann.

³⁷Der Terminus *Finalisieren* wurde gewählt um die Beendigung des Produktionsprozesses und der Handhabung/-Manipulation des Werkstücks hervorzuheben.

Der Handlungsraum Anlauf berücksichtigt dabei ausschließlich Aktionen, die ohne Handhabung von Material durchgeführt werden können.

Auf Grund der Forderung nach einer Grundstellung, die sowohl nach dem Anlauf des Systems als auch vor dem erneuten Fertigen eines Produktes erreicht wird, entspricht der Zielzustand des Rücklaufplanes dem Zielzustand des Anlaufplanes. Um eine strikte Trennung zwischen den Handlungsräumen Anlauf- und Rücklauf zu erreichen, stellt der *Rücklauf* eine Äquivalenzbeziehung zwischen identischen Zuständen in beiden Handlungsräumen dar.

An dieser Stelle sei zusätzlich bereits vorab angemerkt, dass eine Planung auf Basis eines solchen beschriebenen, zusammenhängenden Zustandsraum nicht der mehrfachen Ausführung eines klassischen zielbasierten Planungsverfahrens entspricht. Grund hierfür ist die Tatsache, dass nicht alle Zielzustände a priori bekannt sind, um eine separate Betrachtung und damit verbundene Trennung in getrennte, zielbasierte Planungen zu ermöglichen. So sind zwar der initiale Zustand des Systems und des Produktes bekannt, nicht jedoch der Zielzustand des Produktionsplans (d.h. auch nicht der Startzustand des Rückführungsplanes) sowie der Zielzustand des Anlauf- bzw. Rückführungsplanes (d.h. auch nicht der Startzustand des Produktionsplanes). Nur im Falle einer gemeinsamen Betrachtung aller Handlungsräume kann die Korrektheit (und ggf. Optimalität) eines Handlungsplans gewährleistet werden. Neben der Möglichkeit der Formulierung als getrennte, zielbasierte Planungsprobleme wäre aus diesem Grund eine mögliche Anwendung von Planungsverfahren denkbar, die eine integrierte Berücksichtigung von Zwischenzielen ermöglichen (vgl. beispielsweise der partial global planning Ansatz von Lesser et al. [DL91, DL92] oder der Planer STeLLa 2.0 [SOM02]). Diese setzen jedoch ebenfalls eine explizite Formulierung der Zwischenziele voraus, weshalb diese keine Anwendung finden können. Aus diesem Grund ist ein neues Verfahren zur Formulierung und Lösung des automatischen Planungsproblems unter Berücksichtigung zusammenhängender Zustandsräume notwendig.

4.1.1.2. Hierarchisierung des Zustandsraumes

Automatische Planungsalgorithmen stellen ganz allgemein Suchalgorithmen innerhalb eines gegebenen Lösungsraumes — dem Handlungsraum — dar. Die Leistungsfähigkeit von Suchalgorithmen ist dabei von der Größe des Suchraums, also des Handlungsraumes, abhängig. Die Größe des Handlungsraumes wiederum ist insbesondere von der Menge der Aktionen sowie dem zu Grunde gelegten Zustandsmodell abhängig³⁸. Dementsprechend ist das Zeitverhalten (also Leistungsfähigkeit entsprechend Anforderung A3) direkt von der Wahl einer geeigneten Modellierung des Zustandsraumes abhängig.

Eine mögliche Definition des Zustandsmodells stellt die Betrachtung von Zuständen als Tupel aller zustandsbeschreibenden Variablen des betrachteten Systems dar (d.h. im Maschinen- und Anlagenbau insbesondere die Berücksichtigung aller installierten Sensoren sowie ggf. zusätzlicher weiterer Parameter wie interner Variablen der Steuerung). Durch diese kombinatorische Betrachtung steigt jedoch die Anzahl der zu betrachtenden, möglichen Pläne exponentiell an. Dieser, als sogenannte Zustandsraumexplosion bekannte, Sachverhalt führt bereits für kleine Maschi-

³⁸Diese Aussage ist für Handlungsräume von zielbasierten Planungsverfahren für die Identifikation eines einzelnen Pfades von Start- zu Zielzustand vollständig. Zusätzliche Anforderungen hinsichtlich der Basisstruktur des Handlungsraumes, wie beispielsweise durch das in dieser Arbeit beschriebene Planungsverfahren mittels zusammenhängender Handlungsräume beschrieben, beeinflussen die Größe des gesamten Suchraumes zusätzlich. Die Korrektheit der Aussage bleibt unter Annahme einer festgelegten Basisstruktur erhalten.

4. Bestimmung der Ablaufreihenfolge von Automatisierungsfunktionen

nen und Anlagen zu schlechter Leistungsfähigkeit eines automatischen Planungsverfahrens und verhindert auf Grund der ungünstigen Skalierbarkeitseigenschaft die praktische Anwendbarkeit. Dies konnte im Rahmen eigener Voruntersuchungen [LSVH12] bereits für die oben beschriebene Struktur der verbundenen Zustandsräume quantitativ belegt werden (für Details siehe Abschnitt 5.2.2). Der hinsichtlich der Anwendbarkeit negative Effekt der Zustandsraumexplosion ist letztendlich auf die Eigenschaft des Zustandsraummodells zurückzuführen³⁹. Um die Herausforderung der Zustandsraumexplosion und damit Anforderung A3 nach einer adäquaten Leistungsfähigkeit des Planungsverfahrens bereits durch das interne Planungsmodell geeignet zu adressieren, ist ein alternativer Ansatz zur Repräsentation des Handlungsraumes notwendig.

Im Kontext der automatischen Modellprüfung und -verifikation wird zur Adressierung des Problems der Zustandsraumexplosion häufig die sog. partial order reduction angewandt [CGMP99]. Hierbei wird die Forderung nach einer vollständigen Sequenzierung gelockert, um keine Aussagen über nebenläufige Aktionen treffen zu müssen. Die Herausforderung besteht für Verfahren der automatischen Modellprüfung darin, eine durch Abstraktion des Zustandsraums resultierende, repräsentative Zustandsmenge abzuleiten, welche eigenschaftserhaltend ist und somit keine zu verifizierende Eigenschaft im Modell durch Abstraktion verloren geht [Pel93]. Betrachtet man strukturelle Eigenschaften von Fertigungsanlagen, stellt die vollständige Sequenzierung aller Aktionen einer Anlage in einer Totalordnung eine potentiell zu hohe Restriktion mit im Hinblick der Anwendbarkeit signifikanten, negativen Effekten dar. Die integrierte, verschränkte Betrachtung aller Aktionen führt dazu, dass durch den automatischen Planungsalgorithmus eine Entscheidung über die sequentielle Reihenfolge potentiell parallelisierbarer Aktionen getroffen wird. Durch eine vollständige Sequenzierung kann daher Anforderung A2 nach einem guten bzw. optimalen Handlungsplan kaum adressiert werden, da hierbei die Parallelisierung von Aktionen eine essentielle Rolle spielt. Ein weiterer, wesentlicher Beitrag dieser Dissertation stellt neben der grundlegenden Struktur der verbundenen Handlungsräume zur Adressierung von Anforderung A1, eine für den Maschinen- und Anlagenbau spezifische Lösung zur Adressierung der zuvor beschriebenen Herausforderung der Zustandsraumexplosion durch Einführung einer partiellen Ordnung in der Handlungsplanung dar. Um einen der wesentlichen Kritikpunkte von Planung partiell geordneter Pläne – die häufig negativen Eigenschaften hinsichtlich Skalierbarkeit und Zeitverhalten (vgl. hierzu Argumentationen von Rintanen et al. [RHN04] oder Nguyen und Kambhampati [NK01]) – zu adressieren, wird hier auf eine neuartige, spezifische Hierarchisierung des Zustandsraumes zurückgegriffen, welche inhärent eine partielle Sequenzierung ermöglicht und demzufolge sowohl Anforderung A2 als auch Anforderung A3 kombiniert adressiert.

Die eingangs in diesem Abschnitt beschriebene Betrachtung von Systemzuständen als Wertetupel zustandsgebender Variablen führt zu einer Berücksichtigung aller möglichen Reihenfolgen von Aktionen innerhalb eines Handlungsraumes und damit auch zu einem entsprechenden, automatischen Planungsalgorithmus. Für eine Bestimmung von Handlungssequenzen für die Steuerungssoftware im Maschinen- und Anlagenbau ist eine Forderung nach einem lediglich partiell geordneten Plan ausreichend und hinreichend. Ein solcher Plan enthält dabei notwendige sequen-

³⁹Eine Verkleinerung des Zustandsraumes durch Reduktion der Aktionsmenge einer Maschine oder Anlage ist lediglich durch eine gröbere Kapselung der Funktionalität (d.h. Abstraktion) möglich. Dies schränkt die Flexibilität, welche durch die Anwendung eines automatischen Planungsverfahrens möglich wäre, wiederum ein. Die Menge der durch eine Maschine bzw. Anlage verfügbaren Aktionen eines zu Grunde gelegten Planungsproblems wird als gegeben und unveränderlich betrachtet.

tielle Abhängigkeiten, trifft jedoch keine (unnötigen) Aussagen über die Reihenfolge paralleler bzw. parallelisierbarer Aktionen.

Die Hierarchisierung des Aktionsraumes wird von einigen Ansätzen im Zuge der HTN Planung bereits verfolgt (siehe Kapitel 3.1.1), wie beispielsweise der HTN Planer SHOP2 [NAI⁺03]. Erste Ansätze für eine Kombination der HTN Planung und Partial Order Planung existieren, z.B. Hi-PoP [BBI⁺14]. Die Hierarchisierung des Aktionsraums setzt jedoch zusätzliche, explizit gegebene Informationen über die Zerlegung von Aktionen (die Hierarchieebenen) voraus, wodurch ein potentiell zusätzlicher Aufwand bei der Modellbildung entsteht⁴⁰. Im Rahmen dieser Arbeit wird daher eine Hierarchisierung des Zustandsraumes auf Basis bereits im Aktionsmodell gegebener Informationen unter teilweisem Rückgriff auf Domänenwissen verfolgt.

Während im Forschungsbereich der automatischen Planung paralleler Aktionen (dem sog. parallel planning) der Begriff der Abhängigkeit bzw. Unabhängigkeit zwischen Aktionen aus dem Blickwinkel der Parallelsierbarkeit von Aktionen betrachtet wird [BF97, RHN06], wird beispielsweise in Arbeiten von Chrupa et al. [CMO12, CB08] ein Unabhängigkeitsbegriff von Aktionen speziell im Hinblick auf die Vertauschbarkeit benachbarter Aktionen in sequenziellen Handlungsplänen untersucht. Der Unterschied dieser beiden Betrachtungsweisen besteht dabei im Wesentlichen darin, dass im Bereich des Parallel Plannings die Unabhängigkeit nicht von der Reihenfolge abhängt, während dies für Chrupa et al. essentiell ist [CMO12]. Beide Betrachtungsweisen haben gemein, dass die Unabhängigkeit von Aktionen auf Basis der Vor- und Nachbedingung (bzw. Effekte) definiert wird. Im Rahmen dieser Arbeit findet ein von beiden Betrachtungsweisen inspirierter Unabhängigkeitsbegriff Verwendung: Aktionen sind dann unabhängig, wenn die Folgen einer parallelen Ausführung identisch mit ihrer Ausführung in beliebiger Sequenz ist. Zwei Mengen von Aktionen sind unabhängig, wenn jede Aktion einer Menge zu allen Aktionen der anderen Menge unabhängig ist und dies auch umgekehrt für alle Aktionen der anderen Menge gilt⁴¹. Diese Definition auf Basis der Parallelisierbarkeit legt eine komplementäre Betrachtung des Abhängigkeitsbegriffs zu Grunde: sind zwei Aktionen nicht abhängig, werden sie als Unabhängig betrachtet; sind Aktionen nicht unabhängig, können sie als abhängig betrachtet werden. Betrachtet man die Abhängigkeit/Unabhängigkeit jedoch aus Sicht der Vertauschbarkeit sequentiell benachbarter Aktionen, stellt die Unabhängigkeit von Aktionen nicht stets das komplementäre Gegenstück zur Abhängigkeit von Aktionen dar [CMO12]. Grund hierfür ist die Tatsache, dass durch eine Aktion Variablenwerte einer sequentiell nachfolgenden Aktionen manipuliert werden, welche die Ausführbarkeit der nachfolgenden Aktion ermöglicht. Eine solche (sequentielle) Abhängigkeit muss nicht auch umgekehrt für die sequentiell nachfolgende Aktion gelten und ist folglich weder kommutativ noch komplementär.

Im Rahmen der in Abschnitt 3 beschriebenen Untersuchung konnte bereits gezeigt werden, dass sich ohne Einschränkung der Universalität und grundsätzlichen Anwendbarkeit auf Grund der Überführbarkeit der unterschiedlichen Formalismen, S/T-Netze als grundlegender Formalismus für die Modellbildung eines automatischen Planungsverfahrens fungieren können. Aus diesem Grund finden S/T-Netze im Folgenden Anwendung, um Handlungsräume zu beschreiben. Mit-

⁴⁰Betrachtet man beispielsweise die Hierarchisierung des Aktionsraumes für eine HTN Planung, kann die Hierarchisierung analog zu dem Softwareentwurfsmuster Kompositum (engl. composite pattern) [GHJV94] betrachtet werden. Aktionen können dabei zu (potentiell wiederverwendbaren) Plänen zusammengesetzt werden, welche wiederum Aktionen einer höheren Abstraktionsebene darstellen. Hierfür sind eine Vielzahl zusätzlicher Informationen notwendig.

⁴¹Eine Unabhängigkeit von Aktionen innerhalb einer Sequenz ist dabei nicht vorausgesetzt.

4. Bestimmung der Ablaufreihenfolge von Automatisierungsfunktionen

tels S-/T-Netzen können parallelisierbare, unabhängige Aktionsmengen direkt abgebildet werden. Existierende Formen der Hierarchisierung von S/T-Netzen erhalten dabei die grundlegende Struktur des Netzes und stellen eine Abstraktion bzw. Verfeinerung des Netzes dar (siehe Kapitel 3.1.3). Wie oben beschrieben, ist eine kombinatorische Betrachtung der Zustände für die Anwendbarkeit eines Planungsverfahrens als nachteilig einzuschätzen. Eine Veränderung des Zustandsraummodells zur Vermeidung einer kombinatorischen Betrachtung legt dabei eine vollständige Überführung eines S/T-Netzes voraus, weshalb für die dieser Arbeit zu Grunde gelegte Problemstellung der automatischen Handlungsplanung existierende, klassische Hierarchisierungsmethoden von S/T-Netzen nur schwer vorteilhaft einsetzbar sind. Aus diesem Grund wird im Rahmen dieser Dissertation ein alternativer Ansatz zur Hierarchisierung verfolgt, die sog. konditionale Hierarchisierung.

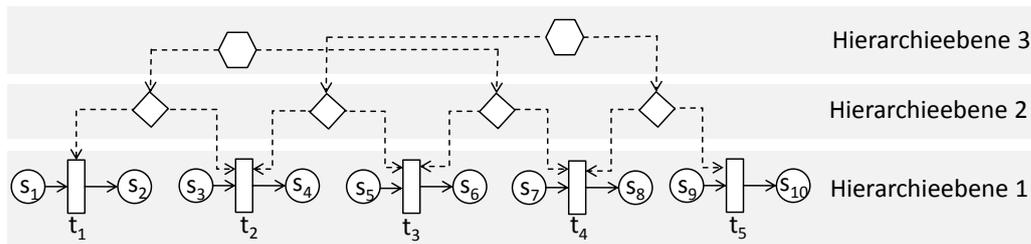


Abbildung 4.3.: Schematische Darstellung konditionale hierarchischer S/T-Netze.

Abbildung 4.3 zeigt beispielhaft das Konzept der konditionalen Hierarchisierung von S/T-Netzen mit drei Hierarchieebenen. Die unterste Ebene wird durch ggf. nicht zusammenhängende S/T-Netze gebildet, wie sie in Kapitel 3.1.3 beschrieben wurden. Die nächst höhere Hierarchiestufe wird durch zusätzliche, spezielle Knotentypen – im Folgenden auch als Kompositionsknoten bezeichnet und visuell durch Rauten in Abbildung 4.3 dargestellt – gebildet, welche mittels zusätzlicher Kanten mit Transitionen der Hierarchieebene 1 verbunden sind. Transitionsknoten, die durch einen gemeinsamen Knoten der Hierarchieebene 2 (Kompositionsknoten) referenziert werden, können ausschließlich gemeinsam und parallel entsprechend einer nebenläufigen Schaltsemantik ausgeführt werden. Hierarchieebenen bilden folglich zusätzliche Bedingungen hinsichtlich der Ausführbarkeit, weshalb auch von konditionaler Hierarchisierung gesprochen wird. Analog gilt die konditionale Ausführbarkeit für Knoten der dritten Hierarchieebene. Knoten auf dieser Hierarchieebene werden auch als Ereignisknoten bezeichnet und sind visuell durch ein reguläres 6-Eck dargestellt. Kanten zwischen unterschiedlichen Hierarchieebenen werden im Folgenden auch als Hierarchiekanten bezeichnet. Konditionale Beziehungen via Hierarchiekanten kann als transitiv betrachtet werden, da Knoten der dritten Hierarchieebene die konditionale Ausführbarkeit der Transitionen der Hierarchieebene 1 auf Basis der Knoten aus Hierarchieebene 2 definieren. Ebenfalls in Abbildung 4.3 ist die sich aus der konditionalen Hierarchie ergebende Möglichkeit der Verschränkung der Ausführbarkeit von Transitionen dargestellt. So werden beispielsweise Transitionen t_2 , t_3 und t_4 durch zwei unterschiedliche Knoten der Hierarchieebene 3 referenziert. Durch Glättung des in Abbildung 4.3 dargestellten konditional hierarchisierten S/T-Netzes mittels Fusion von Transitionsknoten ergibt sich das in Abbildung 4.4 visualisierte S/T-Netz. Grundlage der Fusion bilden die Knoten der höchsten Hierarchiestufe; die Anzahl dieser Knoten entspricht dabei der Anzahl der Transitionen in einem geglätteten S/T-Netz. Da der in Abbildung 4.3 dargestellte Ausschnitt des konditional hierarchisierten S/T-Netz aus zwei

Hierarchieknoten der Ebene 3 besteht, existieren zwei entsprechende Transitionen im geglätteten Netz: Transition t_6 ist dabei durch Zusammenfassung der Transitionen t_1 bis t_4 des in Abbildung 4.3 dargestellten Netzes entstanden; Transition t_7 aus den Transitionen t_2 bis t_5 .

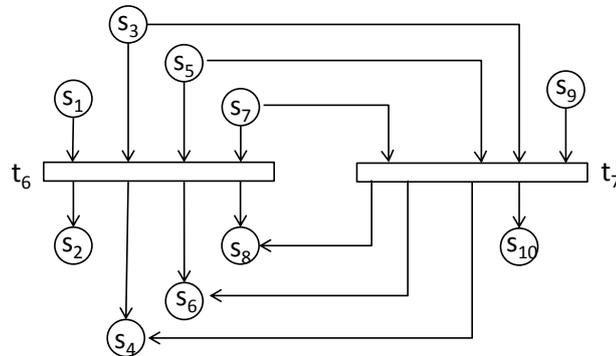


Abbildung 4.4.: Exemplarisches geglättetes S/T-Netz des konditional hierarchisierten S/T-Netzes aus Abbildung 4.3.

Die Ausführbarkeit von Aktionen ist, analog zu den Konzepten der zielbasierten Planungsverfahren und üblicher Aktionssprachen (siehe Kapitel 3.1), durch Vor- und Nachbedingungen beschrieben. Im Kontext des Maschinen- und Anlagenbau kann die Ausführbarkeit von Aktionen als Wertebelegung zustandsgebender Variablen aufgefasst werden, welche durch logische Aussagen geeignet in Vor- und Nachbedingungen verknüpft sind. Variablen, die innerhalb der Vor- bzw. Nachbedingung einer Aktion referenziert werden, werden im Folgenden auch als Variablenmenge der Aktionen bezeichnet. Wie in Aktionssprachen üblich, werden alle Auswirkungen von Aktionenausführungen – also die Menge der beeinflussten Variablen – vollständig durch Vor- und Nachbedingungen spezifiziert. Folglich sind Aktionen, deren Variablenmenge disjunkt ist, als unabhängig zu betrachten. Auf Grund dieser Tatsache wird als niedrigste Ebene eines hierarchischen Zustandsraumes für die Maschinen- und Anlagenautomatisierung die Variablenebene gewählt.

Die Anhängigkeit der Ausführung von Funktionalitäten, d.h. Aktionen, unterschiedlicher Komponenten, ist innerhalb fertigungstechnischer Systeme auf konstruktive Einschränkungen zurückzuführen, welche durch den technischen Prozess gegeben sind. Konstruktive Einschränkungen ergeben sich insbesondere durch räumlich angrenzende oder überlappende Arbeitsräumen. Während überlappende Arbeitsräume insbesondere bei Montage- oder Handhabungssystemen eine Herausforderung darstellen, ist ein angrenzender Arbeitsbereich bei modularen Fertigungssystemen notwendig, um den Materialfluss sicherzustellen; die Funktionen sind jedoch mit Ausnahme der Materialübergabe unabhängig bzw. als solche im Folgenden betrachtet. Folglich stellt die Modul- bzw. Komponentenebene einen geeigneten Kandidaten für eine weitere Hierarchisierungsebene dar.

Das übergeordnete Ziel in der Anlagenautomatisierung des Maschinen- und Anlagenbaus ist die automatisierte Herstellung (hier insbesondere Fertigung) von Gütern. Betrachtet man den Fertigungsprozess aus Sicht der klassischen Handlungsplanung, kann vereinfacht die Zielsetzung eines Fertigungssystems als die Transformation (hier diskrete Abfolge von Transformationsschritten) des Produktes von einem Initialzustand in einen Endzustand aufgefasst werden. Diese diskreten Transformationsschritte stellen einzelne Zustandsübergänge des Produktes dar. Aus diesem

4. Bestimmung der Ablaufreihenfolge von Automatisierungsfunktionen

Grund wird neben Zuständen und Zustandsübergängen des Automatisierungssystems ebenfalls eine Beschreibung der Auswirkungen der Ausführung von Aktionen auf das Produkt benötigt, da dies dem zuvor genannten Transformationsschritt entspricht. Im Folgenden wird das Konzept der attributiven Beschreibung von Produkten – auch als Produkteigenschaften oder -merkmale – aufgegriffen, wie es beispielsweise im Bereich der Handlungsplanung für Werkzeugmaschinen und Montageroboter bereits eingesetzt wird (vgl. Abschnitt 3.2.2.1). Teils wird in diesem Zusammenhang auch der englische Begriff *feature* verwendet (vgl. beispielsweise Nau et al. [NGR95]) und bezieht sich hierbei insbesondere auf Produkteigenschaften aus Sicht des Kunden. Im Folgenden wird als Produktmerkmal bzw. Produkteigenschaft eine beliebige, attributive Charakterisierung des Produktes verstanden, wobei insbesondere solche Merkmale relevant sind, welche während der Fertigung verändert werden und folglich den Produktionsprozess – die Transformation hin zu einem spezifischen Produkt – geeignet beschreiben. Entsprechend wird ein Produkt durch zwei Hierarchieebenen beschrieben: einer Variablenebene (analog zu Hierarchieebene 1 des Automatisierungssystems) sowie die Ebene des Produktes selbst, welche analog der zuvor beschriebenen Hierarchieebene 2 des Automatisierungssystems aufgefasst werden kann.

In den bisherigen Ausführungen wurden bereits je zwei Hierarchieebenen für das Automatisierungssystem (Variablen- und Modulebene) sowie für das Produkt (Variablen- und Produktebene) eingeführt. Während diese zuvor beschriebene Hierarchieebenen Abhängigkeiten zwischen Variablen und innerhalb einzelner Module bzw. des Produktes betrachten, existieren weitere, bisher nicht berücksichtigte Abhängigkeiten. So existieren einerseits Abhängigkeiten zwischen Modulen eines Fertigungssystems, beispielsweise im Falle der Materialübergabe zwischen den Modulen. Andererseits existiert aktuell innerhalb des beschriebenen Modells noch kein Zusammenhang zwischen Funktionsausführungen des Fertigungsautomatisierungssystems und Zustandsübergängen des Produktes. Aus diesem Grund wird eine weitere, dritte Hierarchieebene eingeführt, welche diese Abhängigkeiten berücksichtigt. Innerhalb dieser, auch als *Anlagenebene* bezeichneten, Hierarchieebene werden einerseits Abhängigkeiten zwischen Modulen unter Bezugnahme auf Modellelemente (Knoten) der entsprechenden Modulebene sowie die durch Ausführung der Aktionen sich verändernde Produkteigenschaften unter Rückgriff auf Modellelemente der zweiten Hierarchieebene des Produktes (Produktebene) berücksichtigt.

4.1.1.3. Hierarchische, zusammenhängende Zustandsräume

Das HiTraP-AT Planungsverfahren bzw. das interne HiTraP-AT Planungsmodell basiert auf beiden zuvor eingeführten, charakteristischen Konzepten, um die Kombination der Anforderungen A1 bis A3 zu adressieren: Zum einen dem Konzept zusammenhängender Zustandsräume, welche aufbauend auf dem klassischen Konzept der Zustandsräume eine entsprechend Anforderung A1 angepasste Konzept darstellt und zum anderen dem Konzept der konditionalen Hierarchisierung. Die konditionale Hierarchisierung wird genutzt, um innerhalb des Modells die Unabhängigkeit von Aktionen berücksichtigen zu können und dadurch sowohl eine Explosion des zu modellierenden Zustandsraumes zu vermeiden (Anforderung A3), aber andererseits letztendlich auch eine Planung mit partiell geordneten Plänen zu ermöglichen (Anforderung A2). Die Integration beider Konzepte ist in Abbildung 4.5 dargestellt.

Für jedes Modul eines betrachteten Systems wird die Struktur zusammenhängender Handlungsräume angewandt. Somit besteht der gesamte Handlungsraum eines Moduls aus den Hand-

4.1. Das HiTraP-AT Planungsverfahren (Anforderungen A1 bis A3)

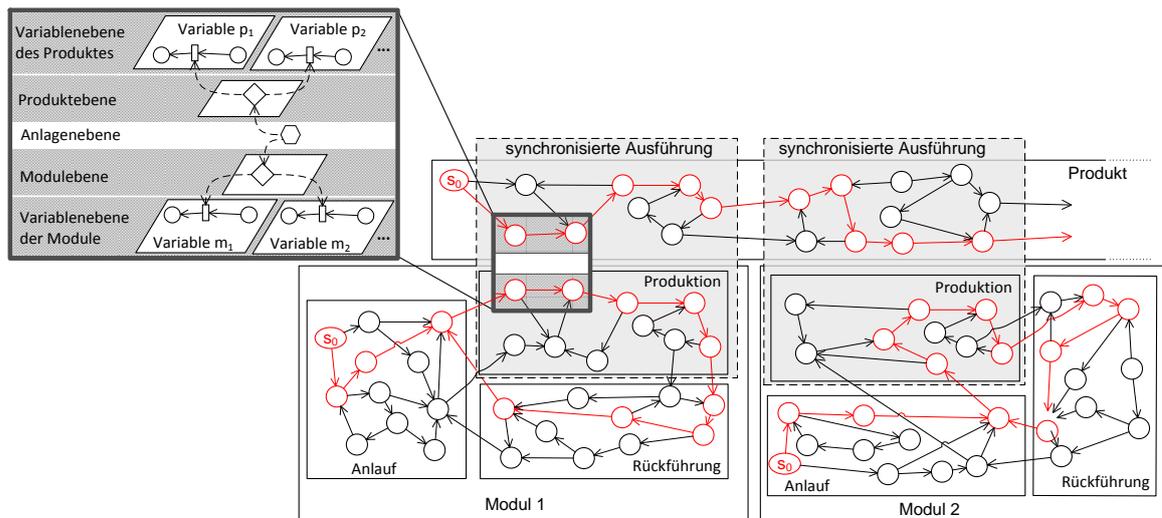


Abbildung 4.5.: Übersicht der Gesamtstruktur von HiTraP-AT Modellen (Darstellung in Anlehnung an [LSVH13]).

lungsräumen "Anlauf", "Produktion" und "Rücklauf". Hierbei finden innerhalb der Handlungsräume eines einzelnen Moduls ausschließlich Aktionen Anwendung, welche dem Modul zugeordnet sind. Diese Zuordnung kann sowohl logisch (eine Automatisierungsfunktion ist logisch einem spezifischen Modul zugeordnet) als auch mechanisch aufgefasst werden (notwendige Aktorik sind Bauteile des Moduls). Entsprechend der in Abschnitt 4.1.1.1 eingeführten Klassifikation der Handlungsräume findet innerhalb der Handlungsräume "Anlauf" und "Rückführung" keine Verarbeitung des Produktes statt. Folglich finden innerhalb dieser Handlungsräume ausschließlich Aktionen Verwendung, die keine produktiven Aktivitäten darstellen oder eine Handhabung des Produktes bedürfen. Auch sind daher für diese Handlungsräume lediglich zwei Hierarchieebenen notwendig (da weder eine Abhängigkeit zwischen anderen Modulen existiert⁴² noch eine Verarbeitung des Produktes erfolgt). Die Variablenebene berücksichtigt dabei zustandsbeschreibende Variablen des Moduls wie beispielsweise Sensorwerte aber auch, für die Modellierung der Aktionsvorbereitungen und Effekte notwendige, interne Zustände der Steuerungssoftware, sofern die Ausführbarkeit oder der Effekt einer Aktion hiervon abhängig ist. Auf dieser Ebene kommen klassische S/T-Netze zur Anwendung, wie dies im vorangegangenen Abschnitt 4.1.1.2 bereits beschrieben wurde. Durch die Hierarchisierung repräsentieren die S/T-Netze die Zustandsübergänge einzelner Variablen. Aus diesem Grund werden sie auch als variablenspezifische S/T-Netze bezeichnet. Die Modulebene fasst entsprechend dem Konzept der konditionalen Hierarchisierung Zustandsübergänge der Variablenebene derart zusammen, dass Vorbereitungen und Effekte durch die entsprechend mittels Variablenebene beschriebenen Zustände repräsentiert sind.

Im Gegensatz zu den vorher beschriebenen Handlungsräumen werden innerhalb des Handlungsraumes "Produktion" eines einzelnen Moduls wertschöpfende Aktionen im Hinblick auf die Fertigung eines Produktes durchgeführt. Entsprechend werden innerhalb dieses Handlungsraumes im Gegensatz zu den anderen Handlungsräumen des Moduls, Aktionen eines Moduls be-

⁴²Diese Annahme wurde bereits in Abschnitt 4.1.1.2 eingeführt.

4. Bestimmung der Ablaufreihenfolge von Automatisierungsfunktionen

trachtet, welche für die Fertigung des Produktes notwendig sind und dabei die charakteristischen Eigenschaften des Produktes verändern oder zumindest bei der Bestimmung der Ausführbarkeit einer Aktion, der aktuelle Zustand des Produktes Berücksichtigung finden muss. Analog zu den weiteren Handlungsräumen des Moduls besteht der Handlungsraum "Produktion" ebenfalls aus zwei Hierarchieebenen (Variablen- und Modulebene). Knoten der zweiten Hierarchieebene dieses Handlungsraumes werden dabei zusätzlich durch eine weitere Hierarchieebene – der Anlagenebene – hinsichtlich der Ausführbarkeit der Aktionen beeinflusst. Dabei finden insbesondere die Knoten der zweiten Hierarchieebene des Produktes (Produktebene) Berücksichtigung, um hinsichtlich Ausführbarkeit den Produktzustand zu berücksichtigen und Einflüsse der Aktionsausführung auf die charakteristischen Merkmale des Produktes (gegeben als Zustände mittels der Variablenebene) zu modellieren.

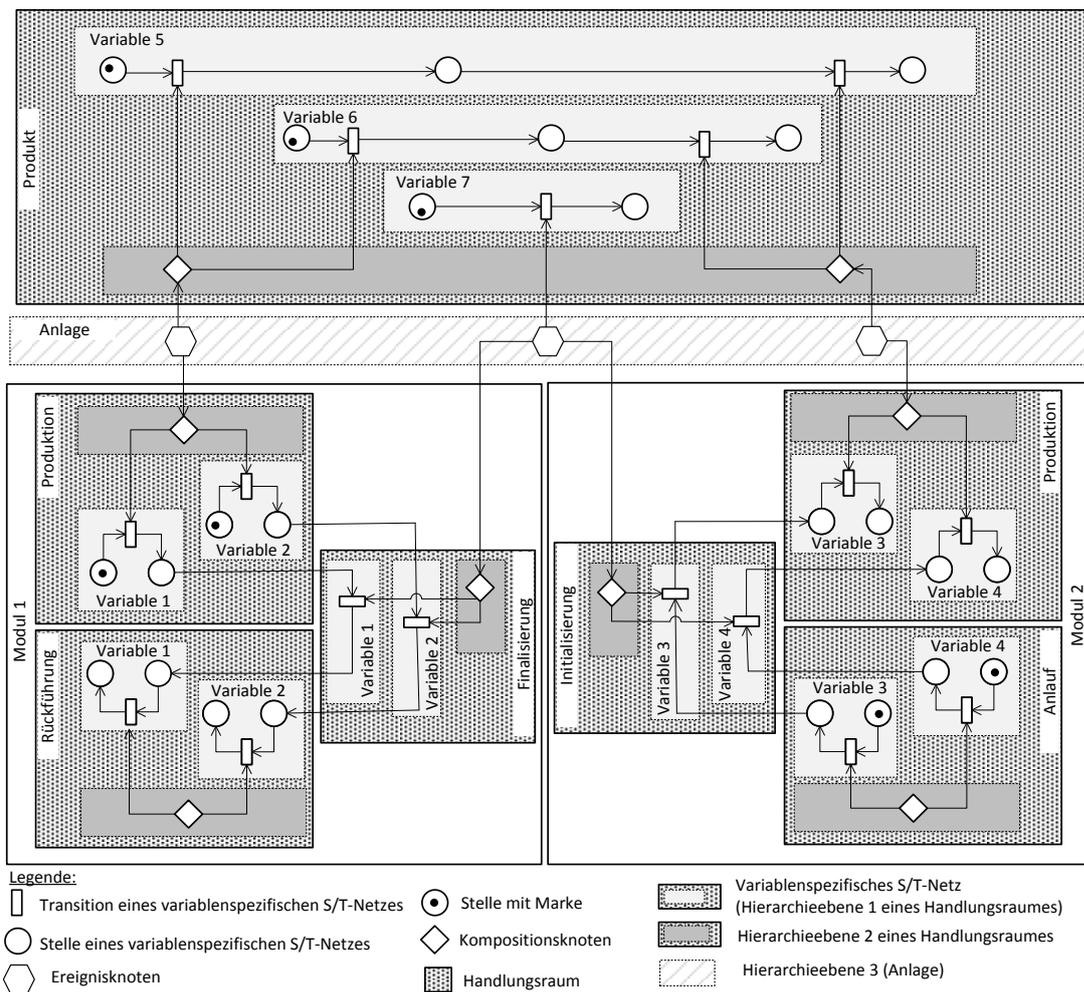


Abbildung 4.6.: Beispielhafter Ausschnitt eines HiTraP-AT Modell mit Übergabe eines Produktes zwischen Modulen.

Der Übergang zwischen den einzelnen Handlungsräumen war bisher nicht Gegenstand näherer Betrachtung. Ein abstraktes Beispiel der Realisierung solcher Übergänge stellt Abbildung 4.6

graphisch dar. Jeder Zustandsübergang innerhalb eines Handlungsraumes wird durch die Ausführung einer Aktion hervorgerufen. Demzufolge werden Initialisierung wie auch Finalisierung durch entsprechende Aktionen durchgeführt, welche sowohl Einschränkungen hinsichtlich ihrer Ausführbarkeit unter Rückgriff auf die Variablen des entsprechenden Moduls unterworfen sind (Vorbedingung) wie auch eine Veränderung dieser bewirken können (Effekt). Aus diesem Grund werden im Folgenden Übergänge zwischen Handlungsräumen ebenfalls durch mehrere Hierarchieebenen beschrieben. Betrachtet man dabei die Semantik der Übergänge zwischen Handlungsräumen, erfolgen diese entweder, falls ein Produkt ein Modul erreicht (Initialisierung) oder die Bearbeitung eines Produktes durch ein Modul abgeschlossen ist und daher das Modul verlässt (Finalisierung). Aus diesem Grund existieren innerhalb der Übergänge der Handlungsräume drei Hierarchieebenen. Ferner werden die zwei möglichen Verbindungen zwischen den das Produkt betreffenden Handlungsräumen, von Handlungsraum "Anlauf" zu "Produktion" (d.h. Initialisierung) sowie von Handlungsraum "Produktion" zu "Rückführung" (Finalisierung), ebenfalls als spezifische Handlungsräume aufgefasst. Im Gegensatz zu den drei (auch als primäre Handlungsräume) bezeichneten Handlungsräumen "Anlauf", "Produktion" und "Rücklauf" enthalten diese verbindenden Handlungsräume keine Stellen bzw. Zustände, die nicht auch den primären Handlungsräumen zugeordnet sind. Die Übergabe von Werkstücken (dem Produkt) zwischen Modulen erfolgt ebenfalls im Zuge des Wechsels zwischen Handlungsräumen beider beteiligter Module. Während ein Modul, welches die Bearbeitung eines Produktes abgeschlossen hat, im Zuge einer Werkstückübergabe von dessen Handlungsraum "Produktion" zu "Rückführung" wechselt, wird das Modul, an welches das Werkstück übergeben wird, gleichzeitig von dessen Handlungsraum "Anlauf" oder "Rückführung" (je nach dem ob bereits zuvor ein Werkstück bearbeitet wurde) in den Handlungsraum "Produktion" überführt.

Wie bereits in Abschnitt 4.1.1.1 beschrieben, stellt der Rücklauf (im Gegensatz zur Initialisierung und Finalisierung) eine Äquivalenzbeziehung zwischen gleichen Zuständen in unterschiedlichen Handlungsräumen dar. Somit entspricht dem Rücklauf keine explizit auszuführende Aktion bzw. Automatisierungsfunktion; es handelt sich vielmehr um eine fiktive Aktion, welche für die Korrektheit des internen HiTraP-AT Planungsmodells notwendig ist. Zwei Zustände von Variablen werden als gleich im Sinne der Rückführung betrachtet, falls sie den selben Variablenwert repräsentieren. Dementsprechend sind innerhalb eines internen HiTraP-AT Planungsmodells jeweils Rücklauftransitionen zwischen zwei Knoten enthalten, welche in den Handlungsräumen "Anlauf" und "Rückführung" einen gleichen Variablenwert repräsentieren. Analog zur Modellierung von Initialisierung und Finalisierung wird der Rücklauf ebenfalls durch mehrere Hierarchieebenen repräsentiert. Da jedoch beide Handlungsräume – Anlauf und Rücklauf – keine wertschöpfenden, das Produkt betreffenden Aktionen enthält und dementsprechend lediglich zwei Hierarchieebenen umfassen, sind für die Modellierung des Rücklaufs ausschließlich zwei Hierarchieebenen notwendig. Hinsichtlich der Ausführung der Rücklauftransitionen auf Variablenebene existieren keine Einschränkungen, da es sich um eine fiktive Aktion handelt. Daher existieren für jede mögliche Kombination von Variablenbelegungen innerhalb des Handlungsraumes "Rückführung" entsprechende Kompositionsknoten um Markierungen aus variablenspezifischen S/T-Netzen des Handlungsraum "Rückführung" zu den gleichen Zustand repräsentierenden Stellen im Handlungsraum "Anlauf" zu überführen. Dieser Sachverhalt ist an Hand eines abstrakten Beispiels visuell in Abbildung 4.7 dargestellt.

4. Bestimmung der Ablaufreihenfolge von Automatisierungsfunktionen

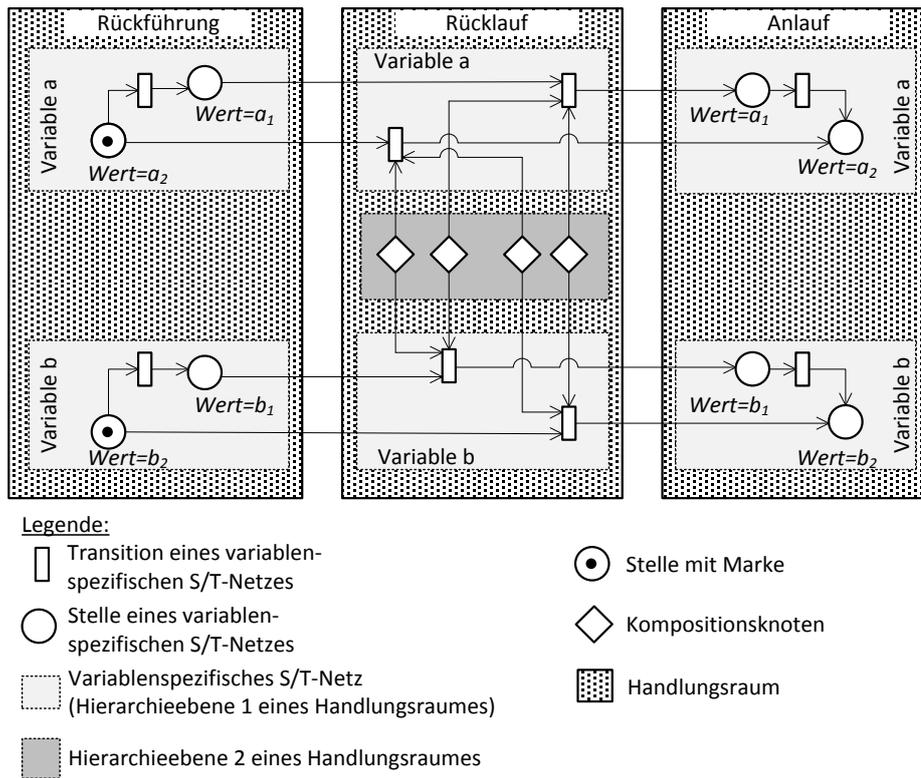


Abbildung 4.7.: Ausschnitt einer exemplarischen Darstellung der Modellierung eines Rücklaufs für ein Modul mit zwei Variablen.

Das in diesem Abschnitt beschriebene interne HiTraP-AT Planungsmodell wird durch einen entsprechenden Planungsalgorithmus genutzt, um ein gegebenes Planungsproblem geeignet zu lösen und einen partiell geordneten Handlungsplan für die Steuerung des Maschinen- und Anlagenbaus abzuleiten. Das spezifische HiTraP-AT Planungsproblem wird im nachfolgenden Abschnitt 4.1.2 beschrieben; die automatische Bestimmung eines, wie bereits zuvor beschrieben, Handlungsplanes wird im darauf folgenden Abschnitt 4.1.3 beschrieben.

4.1.2. Das HiTraP-AT Planungsproblem

Die Anwendung der beiden spezifischen strukturellen Konzepte für die Handlungsplanung – zusammenhängende Zustandsräume und konditionale Hierarchisierung – wie sie im vorangehenden Abschnitt 4.1.1 beschrieben wurden, sowie allgemein die Anwendung einer automatischen Planung für die Steuerungsebene des Maschinen- und Anlagenbaus bedingen zusätzliche anwendungs- bzw. domänenspezifische Informationen im Vergleich zur klassischen Formulierung von Planungsproblemen (vgl. hierzu auch Abschnitt 2.1.1). Abbildung 4.8 gibt eine graphische Übersicht der Konzepte innerhalb des HiTraP-AT Planungsproblems. Im Folgenden werden die darin enthaltenen Konzepte näher beschrieben.

4.1. Das HiTraP-AT Planungsverfahren (Anforderungen A1 bis A3)

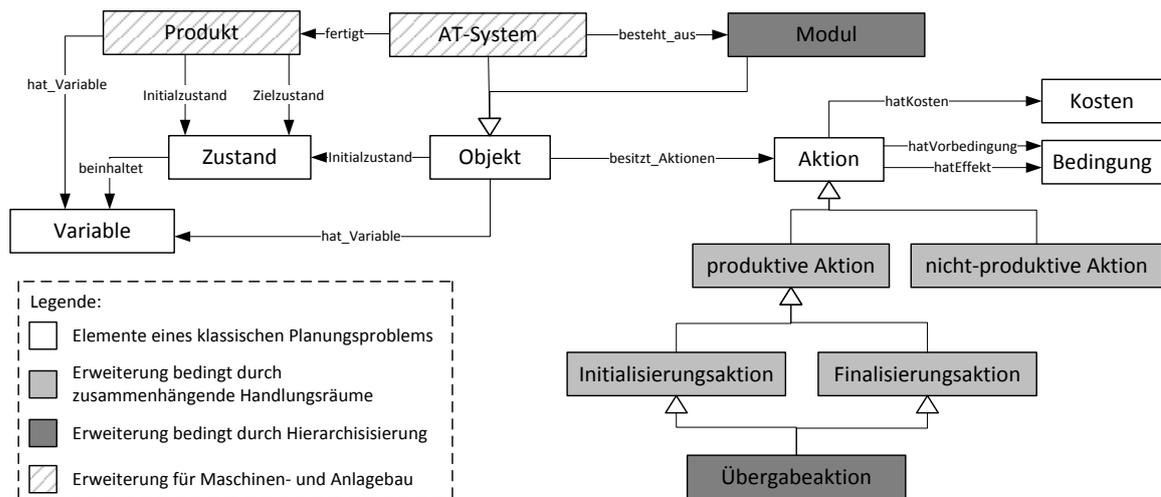


Abbildung 4.8.: Graphische Darstellung der Konzepte eines HiTraP-AT Planungsproblems in Form eines UML Klassendiagramms.

Ein klassisches Planungsproblem (vgl. Abschnitt 2.1.1) ist ausgehend von einem *Objekt* formuliert, welches für die Planung von Interesse ist (d.h. Gegenstand des Planungsproblems ist). Dieses Objekt besitzt eine Menge ausführbarer *Aktionen*, welche durch *Vorbedingungen*, *Effekte* und zugeordnete *Kosten* – eine (fiktive) numerische Quantifizierung von Kosten der Ausführung einer Aktion – definiert sind. Diese stellen Bedingungen (Wahrheitsaussagen) dar, welche auf Basis von *Variablen* formuliert sind. Variablen definieren in diesem Zusammenhang Zustände des Systems, welche durch Aktionsausführung verändert werden (Effekte). Ferner wird ein *Initialzustand* für das Objekt angegeben, welcher eine spezifische Wertebelegung der Variablen darstellt. Dieser Formulierungsansatz wird innerhalb des HiTraP-AT Modells aufgegriffen und spezifisch verfeinert bzw. erweitert.

Spezifisch für den hier vorliegenden Anwendungsfall, der Planung für die Steuerungsebene des Maschinen- und Anlagebaus, wird das *AT-System* als spezifisches Objekt eingeführt, welches ausführbare Aktionen anbieten kann. Im Gegensatz dazu stellt ein zu fertigendes Produkt kein solches Objekt dar, da es keine ausführbaren Aktionen besitzt. Um das von HiTraP-AT verfolgte Konzept der Hierarchisierung des Zustandsraumes zu ermöglichen, ist die Definition von *Modulen* notwendig. Jede Aktion eines AT-Systems ist dabei gleichzeitig mindestens einem Modul zugeordnet. In diesem Zusammenhang ist ein AT-System als logische Strukturierung einer Menge von Modulen zu verstehen. Analog ist jede Variable eines AT-Systems genau einem Modul zugeordnet. Diese zusätzlichen Objekte – also AT-System und Modul – stellen dabei folglich eine logische Strukturierung der Zuordnung von Aktionen bzw. Variablen zu Objekten dar.

Analog zur Formulierung klassischer Planungsprobleme werden in HiTraP-AT Initialzustände angegeben, sowohl für ein AT-System als auch für Module. Da ein AT-System die identische Variablenmenge wie die Menge aller zugeordneten Module besitzt, spiegelt sich ein analoger Sachverhalt für den Startzustand wieder. Der Startzustand des Systems ist durch die Startzustände der zugeordneten Module definiert. Im Gegensatz zu Initialzuständen wird dem allgemeinen Objekt jedoch kein Zielzustand angegeben, sondern, entsprechend der Charakteristik eines Fer-

4. Bestimmung der Ablaufreihenfolge von Automatisierungsfunktionen

tigungsautomatisierungssystem, wird die Zielsetzung ausschließlich durch den *Zielzustand des Produktes* definiert.

Ausgehend von der Definition von Aktionen klassischer Planungsverfahren werden im Rahmen dieser Arbeit unter Aktionen eine diskrete, durch eine Steuerung ausführbare Funktionalität verstanden, welche eine in endlicher Zeit beobachtbare Veränderung charakteristischer Merkmale von Produkt und/oder Automatisierungssystem (repräsentiert durch Variablen) bedingt. Vereinfacht wird dabei im Folgenden angenommen, dass relevante Eigenschaften dieser Aktionen ausschließlich durch Vor- und Nachbedingungen beschrieben werden können und alle weiteren, zu beachtenden Aspekte innerhalb der Aktionen geeignet berücksichtigt werden. Diese Definition legt dabei eine für die Anwendung klassischer Planungsmethoden notwendige Diskretisierung von Aktionen und deren Charakterisierung mittels Vor- und Nachbedingungen zu Grunde. Eine solche Modellierung der Funktionalität auf Basis diskreter (Software-)Funktionen wurde bereits in den dieser Dissertationen thematisch naheliegenden Arbeiten, wie sie insbesondere in Abschnitt 3.2.2.1 und Abschnitt 3.2.2.2 beschrieben wurden, bereits erfolgreich eingesetzt. Diese Arbeiten bildet daher diesbezüglich die konzeptionelle Basis. Durch die ausschließliche Berücksichtigung von Vor- und Nachbedingungen, stehen innerhalb des Modells keine Informationen über das Verhalten einzelner Prozesswerte zur Verfügung während der Ausführung einer Aktion. Demzufolge können beispielsweise keine Abhängigkeiten zwischen Aktionen auf Basis von Zustandswerten beschrieben werden, welche während der Ausführung von Aktionen erreicht werden. Zu beachten ist hierbei insbesondere auch, dass Nachbedingungen als Zusicherung eines spezifischen Zustands nach Aktionsausführung betrachtet werden. Dabei wird ebenfalls angenommen, dass eine Aktion genau dann beendet ist, wenn die Nachbedingung der Aktion erfüllt ist. Dementsprechend wird nicht berücksichtigt, falls während der Ausführung einer Aktion zwischenzeitlich Zustände erreicht werden, die die Nachbedingung einer Aktion erfüllen, die Ausführung jedoch noch nicht abgeschlossen ist. Dies stellt jedoch keine Einschränkung dar, da in einem solchen Fall durch zusätzliche Variablen die Eindeutigkeit der Nachbedingung hergestellt werden kann. Das dieser Arbeit zu Grunde gelegte Konzept der Aktion kann dabei einerseits als Zusammenfassung bzw. Verallgemeinerung existierender Konzepte sowie andererseits deren spezifische Anwendung für die Steuerungsebene des Maschinen- und Anlagenbaus verstanden werden. Während im Bereich der Handlungsplanung für die robotergestützte Montage und Werkzeugmaschinen als Aktionen bezeichnet werden, sind diese Aktionen im Kontext dienstorientierter Architekturen als Dienste (vgl. Abschnitt 3.2.1.1) bezeichnet. Letztere wurden dabei im Zuge von Forschungsarbeiten bereits geeignet in die Automatisierungstechnik und auch teils für die Steuerung im Maschinen- und Anlagenbau überführt (vgl. insbesondere Abschnitt 3.2.1.1). Wie bereits in der im vorangegangenen Kapitel beschriebenen Untersuchung verwandter Arbeiten identifiziert werden konnte, lassen sich unterschiedliche Anwendungen und zu Grunde gelegte theoretische bzw. architektonische Ansätze aufeinander abbilden (vgl. beispielsweise Abschnitt 3.2.1.3). Auch im Kontext intelligenter (selbst-)adaptiver Systeme und Multi-Agenten-Systemen (vgl. Abschnitt 3.2.1.2) findet das klassische Konzept diskreter Aktionen für die Handlungsplanung Anwendung. Das Konzept der Multiagentensysteme für die Steuerungsebene des Maschinen- und Anlagenbaus stellt dabei jedoch ein vergleichsweise junges Forschungsfeld dar. In den Arbeiten von Andreas Wannagat [Wan10] und Daniel Schütz [Sch14] wird die erfolgreiche Anwendung von Agenten basierten Ansätzen für die Steuerungsebene des Maschinen- und Anlagenbaus erfolgreich evaluiert und ein Ansatz zur modellbasierten Entwicklung und automatischen Generierung von Programmcode für

die Steuerung beschrieben. Aufbauend auf diesen Arbeiten, in denen die durch einen Agenten ausführbaren, diskreten Aktionen bereits erfolgreich eingesetzt und für die Steuerungsebene des Maschinen- und Anlagenbaus realisiert wurden, wird in dieser Arbeit dieses Konzept zu Grunde gelegt und für eine Anwendung zur automatischen Handlungsplanung für die Steuerungsebene des Maschinen- und Anlagenbaus erweitert. Die hier vorliegende Arbeit ist folglich an diverse Forschungsarbeiten aus unterschiedlichen Bereichen angelehnt und bedient sich deren Konzepten. Dementsprechend kann diese Arbeit als konzeptuelle Brücke zwischen existierenden Ansätzen für die Steuerungsebene (wie z.B. den Arbeiten von Wannagat und Schütz), der automatischen Handlungsplanung im Bereich der robotergestützten Montage sowie theoretischen Ansätzen der klassischen Handlungsplanung verstanden werden.

Auf Basis dieser zuvor unter Rückgriff auf unterschiedliche, konzeptuelle Arbeiten definierte Aktionen wird im Folgenden die Erweiterung dieses Aktionskonzepts für die Adressierung domänenspezifischer Anforderungen der Handlungsplanung von Automatisierungsfunktionen beschrieben. Die Anwendung zusammenhängender Handlungsräume bedingt die Einführung einer Taxonomie möglicher Aktionen⁴³, da für Aktionen der Zustandsräume "Anlauf" und "Rückführung" ausschließlich der Zustand des technischen Systems relevant ist, deren Einschränkungen der Ausführung – d.h. im Kontext klassischer Planungsverfahren Vorbedingung und Effekte der Aktionen – sich ausschließlich auf das technische System beziehen. Diese Aktionen werden im Folgenden auch als *nicht-produktive Aktionen* bezeichnet. Zusätzlich ist in der Menge der nicht-produktiven Aktionen auch eine Rücklaufaktion enthalten. Sie unterliegt keiner Einschränkung im Hinblick auf die Ausführbarkeit, d.h. die Vorbedingung ist stets erfüllt. Zielsetzung dieser Aktionen ist die Überführung eines Moduls vom Handlungsraum "Rückführung" zurück in dessen Grundstellung, wie dies im vorangegangenen Abschnitt bereits beschrieben wurde. Im Gegensatz zu den Handlungsräumen "Anlauf" und "Rückführung" impliziert die Zielsetzung des Handlungsraumes "Produktion", dass dort ausgeführte Aktionen neben den Einschränkungen durch das technische System auch solche des zu realisierenden, technischen Prozesses zur Fertigung eines Produktes unterliegen. Im Terminus klassischer Planungsverfahren formuliert, sind folglich für Aktionen des Handlungsraumes "Produktion" die Vorbedingungs- und Effektdefinitionen auf Basis der Variablen des AT-Systems sowie auch des Produktes formuliert. Diese Aktionen werden nachfolgend auch als *produktive Aktionen* bezeichnet.

Die Einführung zusammenhängender Handlungsräume impliziert auch die Notwendigkeit der Verbindung von Handlungsräumen. Wie in Abschnitt 4.1.1.1 beschrieben, kann zwischen Initialisierung und Finalisierung unterschieden werden. Diese Funktionalitäten werden dabei ebenfalls durch Aktionen repräsentiert, die durch das Automatisierungssystem ausgeführt werden müssen. Sogenannte *Initialisierungsaktionen* beginnen einen Produktionsprozess, weshalb deren Ausführbarkeit beispielsweise von der Verfügbarkeit notwendiger Werkstücke an entsprechender Örtlichkeit innerhalb des Automatisierungssystems abhängig ist. Ferner können bereits Werkstückeigenschaften durch diese Initialisierungsaktionen beeinflusst werden. Folglich können sich Vorbedingungen und Effekte sowohl auf Variablen des Systems wie auch des Produktes beziehen. Analog gilt dies für *Finalisierungsaktionen*, da diese die letzte Aktion innerhalb des Pro-

⁴³Die Einführung einer Taxonomie von Aktionen widerspricht nicht den Anmerkungen in Abschnitt 4.1.1.2, da diese Erweiterung im Vergleich zu einer Hierarchisierung, wie sie beispielsweise in der HTN Planung Verwendung findet, keine vollständig neuen Konzepte hinsichtlich des Aktionsraumes einführt, sondern lediglich eine Klassifikation und damit verbundener Aufteilung des Aktionsraumes darstellt.

4. Bestimmung der Ablaufreihenfolge von Automatisierungsfunktionen

duktionsplanes darstellen, und letztendlich das Produkt in den Zielzustand versetzen müssen. Entsprechend müssen auch für diese Aktionen Variablen des AT-Systems sowie des Produktes in Vorbedingung und Effekt berücksichtigt werden. Aus diesem Grund sind beide Arten von Aktionen, Initialisierungs- und Finalisierungsaktionen, als spezifische, produktive Aktionen zu klassifizieren.

Übergabeaktionen stellen wiederum spezifische Aktionen dar, deren Notwendigkeit aus der Kombination der beiden konzeptuellen Ansätze – zusammenhängende Handlungsräume und Hierarchisierung – resultiert. Die Anwendung der Modulebene führt dazu, dass ein Produkt von einem Modul zu einem Anderen weitergegeben bzw. übergeben wird. Diese Tatsache wird mittels Übergabeaktionen dargestellt. Da bei der Übergabe zwischen Modulen für ein Modul die Bearbeitung des Produktes abgeschlossen ist und entsprechend eine Finalisierungsaktion durchführt, beginnt für ein anderes (das das Produkt übernehmende) Modul die Bearbeitung des Produktes. Entsprechend führt letzteres Modul eine Initialisierungsaktion aus. Folglich stellt eine Übergabeaktion innerhalb des HiTraP-AT Planungsproblems sowohl eine Finalisierungs- als auch eine Initialisierungsaktion dar.

Die Kombination des Konzeptes zusammenhängender Zustandsräume sowie der konditionalen Hierarchisierung innerhalb des internen HiTraP-AT Planungsmodells (vgl. Abschnitt 4.1.1) resultiert in einen Handlungsplan, welcher die durch Anforderung A1 entsprechenden Teilfunktionalitäten eines Handlungsplanes für jeweils einzelne Module realisiert. Hierdurch ergibt sich eine spezifische Struktur eines HiTraP-AT Handlungsplanes, wie er exemplarisch Abbildung 4.9 darstellt ist. Nicht-produktive Aktionen der Module werden in den Handlungsräumen "Anlauf" und "Rücklauf" genutzt, wohingegen die produktiven Aktionen im Handlungsraum "Produktion" sowie für den Übergang zwischen den Handlungsräumen (in Form von Initialisierungs-, Finalisierungs- bzw. Übergabeaktionen) genutzt werden. Durch Ausführung einer Initialisierungsaktion (welche keine Übergabeaktion darstellt) wird die Fertigung eines einzelnen Werkstücks im Kontext der gesamten Anlage begonnen. Wie in Abbildung 4.9 dargestellt, wird die Unabhängigkeit nicht-produktiver Aktionen unterschiedlicher Module für eine (mögliche) Parallelisierung der Aktionsausführung (im Hinblick auf Anforderung A2) genutzt, weshalb letztendlich der gesamte HiTraP-AT Handlungsplan durch eine partielle Ordnung der Aktionen charakterisiert ist.

Zusammenfassend bedingt das Konzept zusammenhängender Handlungsräume die Einführung einer Taxonomie von Aktionen, wobei zwischen produktiven und nicht-produktiven Aktionen unterschieden wird. *Produktive Aktionen* finden innerhalb des Handlungsraumes "Produktion" Anwendung und berücksichtigen Variablen des AT-Systems und Produktes in Vorbedingung und Effekt. *Nicht-produktiven Aktionen* werden in den Handlungsräumen "Anlauf" und "Rückführung" genutzt und berücksichtigen lediglich die Variablen des AT-Systems in Vorbedingung und Effekt. *Initialisierungsaktionen* und *Finalisierungsaktionen* stellen spezifische produktive Aktionen dar, welche den Übergang zwischen den Handlungsräumen ermöglichen bzw. realisieren. Die Übergabe von Werkstücken zwischen Modulen eines Automatisierungssystems erfolgt durch *Übergabeaktionen*. Durch die Hierarchisierung des Zustandsraumes findet das Konzept zusammenhängender Zustandsräume in HiTraP-AT für jedes Modul Anwendung, weshalb Übergabeaktionen sowohl Initialisierungsaktionen eines Moduls wie auch Finalisierungsaktionen eines anderen Moduls darstellen. Diese unterschiedlichen Aktionen finden sich entsprechend auch innerhalb eines charakteristischen HiTraP-AT Handlungsplanes wieder. Die Kombination der Konzepte des

4.1. Das HiTraP-AT Planungsverfahren (Anforderungen A1 bis A3)

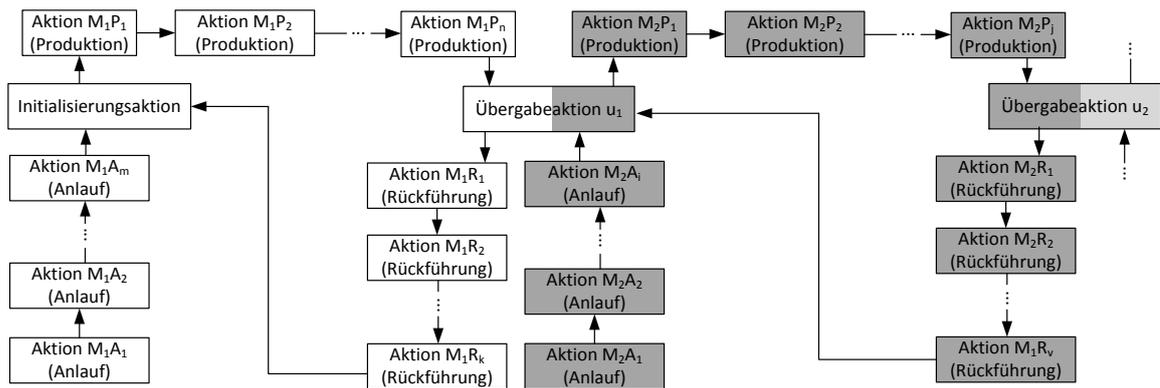


Abbildung 4.9.: Exemplarische Ausschnitt eines partiell geordneten HiTraP-AT Handlungsplanes. Die Zuordnung von Aktionen zu Modulen ist durch unterschiedliche Hintergrüdfarben gekennzeichnet; M_bP_c bezieht sich auf eine Aktion P_c des Handlungsraumes "Produktion" von Modul M_b ; analog beziehen sich Aktionen M_bA_c bzw. M_bR_c auf Aktionen des Handlungsraumes "Anlauf" A_c bzw. "Rücklauf" R_c des Moduls M_b .

zusammenhängenden Handlungsraumes sowie der konditionalen Hierarchisierung innerhalb des internen HiTraP-AT Planungsmodells führen zu einer partiellen Ordnung potentiell parallelisierbarer Aktionen.

4.1.3. Formulierung und Lösung des HiTraP-AT Planungsproblems mittels Linearer Programmierung

In beiden vorangegangenen Abschnitten wurde das interne HiTraP-AT Planungsmodell zur Repräsentation aller valider Handlungspläne in Form eines zusammenhängenden, hierarchischen Handlungsraummodells sowie das HiTraP-AT Planungsproblem und die charakteristische Struktur des sich aus dem internen HiTraP-AT Planungsmodell ergebenden HiTraP-AT Handlungsplans beschrieben. Auf Basis dieser Konzepte wird im Rahmen dieses Abschnitts nun die automatische Ableitung guter bzw. optimaler Handlungspläne auf Basis des internen HiTraP-AT Planungsmodells beschrieben. Um einerseits die Zielstellung dieser Dissertation, die Entwicklung eines Verfahrens zur automatischen Bestimmung von Ablaufreihenfolgen von Softwarefunktionen, zu adressieren und andererseits die entsprechenden Anforderungen – insbesondere die Forderung nach der Bestimmung eines guten bzw. optimalen Handlungsplans (Anforderung A2) sowie nach geeigneten Leistungswerten im Hinblick auf das zeitliche Verhalten und die Skalierbarkeit (Anforderung A3) – zu berücksichtigen, greift das HiTraP-AT Planungsverfahren auf die Lineare Programmierung zurück.

Es existieren bereits einige Arbeiten, die eine Kodierung bzw. Lösung des klassischen Planungsproblems als Lineares Programm anwenden (vgl. Abschnitt 3.1.2.2). Die Lineare Programmierung ermöglicht insbesondere die Definition eines Optimierungskriteriums, wodurch nicht nur ein valider, sondern vielmehr ein im Hinblick auf das gegebene Optimierungskriterium optimaler Handlungsplan bestimmt werden kann (Anforderung A2). Im Gegensatz zu existierenden Arbeiten, die eine klassische Problemdefinition direkt als Lineares Programm formulieren, re-

4. Bestimmung der Ablaufreihenfolge von Automatisierungsfunktionen

präsentiert das interne HiTraP-AT Planungsmodell in Form eines konditional hierarchischen, zusammenhängenden S/T-Netzes bereits alle Abläufe. Aus diesem Grund stellen existierenden Ansätze zur Planung auf Basis der Linearen Programmierung einen Ausgangspunkt dar, können jedoch nicht direkt für die Lösung des HiTraP-AT Planungsproblems genutzt werden. Einen weiteren Ausgangspunkt stellen die Arbeiten dar, welche eine Formulierung der Schaltsemantik von S/T-Netzen mittels Linearer Programme beschreiben. Details zu diesen Arbeiten sind in Abschnitt 3.1.2.2 sowie Abschnitt 3.1.3 zu finden. Ausgehend von diesen Arbeiten wird in diesem Abschnitt die Formulierung des zuvor bereits eingeführten internen HiTraP-AT Planungsmodells als Lineares Programm beschrieben.

Im folgenden Abschnitt 4.1.3.1 wird im Hinblick auf eine geeignete Adressierung der Anforderung nach Optimalität des Handlungsplanes eingangs eine allgemeine Betrachtung durchgeführt, welches Optimalitätskriterium für die hier vorliegenden Anforderungen genutzt werden kann. Im Anschluss in Abschnitt 4.1.3.2 wird die Formulierung des internen HiTraP-AT Planungsmodells in Form eines Linearen Programms detailliert beschrieben. Darauf Aufbauend wird in Abschnitt 4.1.3.3 erläutert, wie durch Lösung des Linearen Programms ein (optimaler und partiell geordneter) Handlungsplan für die Steuerungssoftware des Maschinen- und Anlagenbaus abgeleitet werden kann.

4.1.3.1. Bestimmung des Optimalitätskriterium

Übergeordnetes Ziel der Optimierung der Massenproduktion in der fertigen Industrie stellt die Maximierung des Anlagendurchsatzes dar, da hierbei die pro Produkt anrechenbaren Produktionskosten minimiert werden (vgl. Anforderung A2). Aus diesem Grund wird im Folgenden der Durchsatz, d.h. die Anzahl Produkte, die in einer bestimmten Zeitspanne gefertigt werden, zu Grunde gelegt. Wie in Abschnitt 3 beschrieben, stellt die Suche nach einer guten oder gar optimalen Lösung eines gegebenen Planungsproblems eine Erweiterung klassischer Planungsprobleme dar. Da für die Maschinen- und Anlagenautomatisierung jedoch gerade die Maximierung des Durchsatzes von essentieller Bedeutung ist, wird im Zuge dieser Arbeit auf ein, um ein Optimierungskriterium erweitertes Planungsverfahren vorgeschlagen.

Das Gesetz von John D.C. Little [Lit61, Lit11] (auch als Little's Law bekannt) führt den Durchsatz auf zwei Basisfaktoren, den Umlaufbestand und die Durchlaufzeit, wie folgt zurück:

$$\text{Durchsatz} = \frac{\text{Umlaufbestand}}{\text{Durchlaufzeit}}$$

Unter *Umlaufbestand* wird die durchschnittliche Anzahl aller gleichzeitig in der Fertigung befindlichen Produkte verstanden; die Durchlaufzeit stellt die Zeit dar, die die vollständige Fertigung eines einzelnen Produktes benötigt. Eine detaillierte Diskussion dieser Zusammenhänge findet sich beispielsweise in [Vah08, CF10]. Um durchsatzoptimiert fertigen zu können, sollte entsprechend dem Gesetz von Little, ein möglichst großer Umlaufbestand sowie eine möglichst geringe Durchlaufzeit realisiert werden.

Die Durchlaufzeit eines Produktes ergibt sich entsprechend dem zuvor vorgestellten Modell durch die Summe der Zeitdauern von Funktionen, die durch einen entsprechenden Planungsalgorithmus ausgewählt werden bzw. wurden, sowie deren Anwendung innerhalb des Handlungsplans. Eine geringe Durchlaufzeit pro Werkstück bzw. Produkt wirkt sich auf Grund der Rolle

4.1. Das HiTraP-AT Planungsverfahren (Anforderungen A1 bis A3)

der Durchlaufzeit als Divisor innerhalb des Gesetzes von Little positiv auf den Gesamtdurchsatz aus. Im Kontext des HiTraP-AT Planungsverfahrens wird die Durchlaufzeit ausschließlich durch den Handlungsraum "Produktion" definiert, da dieser entsprechend der HiTraP-AT Modellspezifikation alle für die Fertigung eines Produktes notwendigen Funktionalitäten enthält. Entsprechend charakterisieren die Ausführungszeiten der durch das Planungsverfahren ausgewählten Funktionen die Durchlaufzeit eines Produktes innerhalb eines durch HiTraP-AT bestimmten Handlungsplanes.

Neben der Durchlaufzeit lässt sich der Durchsatz entsprechend dem Gesetz von Little positiv durch die Anzahl der innerhalb einer Maschine oder Anlage gleichzeitig in der Fertigung befindlichen Produkte, d.h. den Umlaufbestand, beeinflussen. Der Umlaufbestand ist dabei von einer Vielzahl von Faktoren wie beispielsweise dem Lagerbestand und aktuell vorliegenden Kundenaufträgen abhängig. Daher stellt der Umlaufbestand typischer Weise eine Fragestellung aus der Produktionsplanung dar (vgl. Abschnitt 2), welche mit Methoden des Scheduling adressiert werden. Aus diesem Grund wird im Rahmen dieser Arbeit die Maximierung des Umlaufbestandes nicht fokussiert.

Betrachtet man jedoch das in dieser Arbeit als beeinflussbar angenommene Verhalten eines gegebenen, betrachteten Automatisierungssystems und alle weiteren Randbedingungen als konstant, wird eine Maximierung des Umlaufbestandes durch die Minimierung des zeitlichen Abstands zwischen einzelnen Produkten charakterisiert. Dies stellt wiederum einen Einflussfaktor dar, welcher durch ein klassisches Planungssystem mit Optimierungsfunktion Berücksichtigung finden kann. Im Kontext des HiTraP-AT Planungssystems sowie dessen Definition der Unabhängigkeit von Funktionen des Automatisierungssystems, wird der Abstand zwischen zwei Produkten durch den Rückführungsplan maßgeblich beeinflusst.

Entsprechend obiger Ausführungen kann die Selektion eines optimalen Handlungsplans – also die Lösung des Planungsproblem – auf ein Optimierungsproblem, konkret ein Minimierungsproblem, auf Basis der Ausführungszeiten der durch das Planungsverfahren ausgewählten Funktionalitäten in den Handlungsräumen "Produktion" und "Rücklauf" zurückgeführt werden. Aus diesem Grund verfolgt das Planungsverfahren HiTraP-AT eine Formulierung des internen HiTraP-AT Planungsmodells als Lineares Optimierungsproblem.

4.1.3.2. Formulierung des HiTraP-AT Planungsverfahrens als Lineares Programm

Ein Lineares Programm kann formal als 4-Tupel $\langle X, D, \Phi, B \rangle$ betrachtet werden (siehe hierzu auch Abschnitt 3.1.2.2), mit einer Menge an Variablen $x_i \in X$, den Definitionsmengen der Variablen D sowie einer Menge an (Un-)Gleichungen Φ und einer (linearen) Bewertungsfunktion B . Die Formulierung eines Planungsproblems in Form eines Linearen Programmes erfolgt durch deklarative Beschreibung notwendiger Charakteristika in Form von (Un-)Gleichungen, welche als Bedingungen Φ des Linearen Programms gegeben sind. Auf diese Weise wird dem Planer der Entscheidungsraum spezifiziert, innerhalb dessen eine bzw. die beste Lösung zu identifizieren ist.

Im Folgenden wird eingangs abgeleitet, welche Knotentypen des internen HiTraP-AT Planungsmodells innerhalb des Linearen Programms durch Variablen repräsentiert werden müssen. Anschließend werden die für die vollständige Formulierung des HiTraP-AT Modells und eine damit verbundene korrekte Bestimmung eines Handlungsplanes notwendigen Bedingungen beschrieben. Wie in Tabelle 4.1 zusammengefasst, sind für die Repräsentation der beiden grundlegenden

4. Bestimmung der Ablaufreihenfolge von Automatisierungsfunktionen

Konzeptbausteine des HiTraP-AT Modells unterschiedliche Bedingungsarten notwendig. Im Zusammenspiel mit diesen Bedingungen wird am Ende dieses Abschnitts die Bewertungsfunktion und dessen spezifische Ausprägung für die Bestimmung durchsatzoptimierender Handlungspläne erläutert.

Tabelle 4.1.: Übersicht genutzter Elemente (d.h. Bedingungen und Bewertungsfunktion) für die Formulierung des HiTraP-AT Modells als Lineares Programm.

Konzeptbaustein	LP Element	Beschreibung
konditionale Hierarchisierung	Schaltbedingung	Sicherstellung der korrekten Schaltsemantik
	Bewertungsfunktion	Auswahl einer minimalen Menge von Aktionen zur Fertigung eines gewünschten Produktes sowie Berücksichtigung eines Optimalitätskriteriums
zusammenhängende Handlungsräume	Flussbedingung	graph-basierte Beschreibung der grundlegenden Merkmale der unterschiedlichen Stellentypen (Start- und Zielstellen, innere Stellen) innerhalb variablen-spezifischer S/T-Netze
	Kurzzyklusbedingung	Vermeidung der Partitionierung des Handlungsplanes durch den Planer
	Integritätsbedingung	Sicherstellung korrekter Übergänge zwischen den einzelnen, zusammenhängenden Handlungsräumen

Variablen: Das dem HiTraP-AT Planungsverfahren zu Grunde gelegte interne Planungsmodell stellt einen spezifischen Graphen dar, innerhalb dessen der Planer einen geeigneten bzw. besten Handlungsplan bestimmen muss. Hierbei ist ein geeigneter Handlungsraum durch die entsprechende Auswahl von Aktionen, welche innerhalb des Graphmodells durch unterschiedliche Knotentypen repräsentiert werden, gekennzeichnet. Diese Modellelemente werden innerhalb des Linearen Programms durch Variablen repräsentiert, für die durch den Planer – hier ein Löser für Lineare Programme – eine Entscheidung getroffen werden muss. Im Folgenden wird ein Lineares Programm als binäres, ganzzahliges Programm (BIP) aufgefasst. Daher gilt für die Definitionsmengen aller Variablen $D = \{0, 1\}$. Im Falle eines BIP wird die Auswahl von Modellelementen durch den binären Wert der entsprechenden Variablen gekennzeichnet: Wird ein Modellelement ausgewählt, weist die entsprechende Variable einen Wert von Eins auf; andernfalls den Wert Null. Variablen des Lineare Programms stellen folglich den Entscheidungsgegenstand der automatischen Handlungsplanung in Form eines Linearen Programms dar. Aktionen des Planungsproblems werden innerhalb des internen HiTraP-AT Planungsmodell durch unterschiedliche Knotentypen repräsentiert: Ereignis-, Kompositions- und Transitions-knoten. Für jeden Ereignis-

und Kompositionsknoten existiert innerhalb des Linearen Programms eine entsprechende binäre Variable. Da innerhalb des Modells ein Transitionsknoten von unterschiedlichen Kompositionsknoten referenziert werden kann, ist eine Eindeutigkeit des Kompositionskontexts in diesen Fällen nicht gegeben. Diese Eindeutigkeit wird jedoch für die Formulierung der Bedingungen innerhalb des Linearen Programms benötigt. Aus diesem Grund sind nicht für Transitionsknoten sondern stellvertretend für jede Bindungskante zwischen Kompositions- und Transitionsknoten entsprechende binäre Variablen innerhalb des Linearen Programms enthalten.

Modulare Fertigungssysteme, wie sie durch HiTraP-AT vorausgesetzt werden, können Module enthalten, welche innerhalb eines Handlungsplans nicht genutzt werden. Dies kann beispielsweise durch die Redundanz von Modulen begründet sein. Sollten mehrere, unterschiedliche Produkte mit Hilfe eines Produktionssystems gefertigt werden, können ebenfalls Module existieren, welche für die Fertigung eines spezifischen Produktes keine Verwendung finden. Wie bereits oben beschrieben, erfolgt die Formulierung des Planungsproblems als Lineares Programm auf Basis der deklarativen Beschreibung der Lösungseigenschaften. Etwaige Charakteristika des Handlungsplanes eines Modules (vgl. auch Abschnitt 4.1.2) müssen dabei durch den Planer (d.h. den Löser des Linearen Programms) lediglich berücksichtigt werden, falls ein Modul Bestandteil der Lösung (d.h. des durch den Planer bestimmten Handlungsplanes) ist. Um diesen Sachverhalt geeignet innerhalb des Linearen Programms darstellen zu können, wird für jedes Modul eine zusätzliche binäre Variable eingeführt (auch als Modulvariable bezeichnet), welche die Nutzung des Moduls innerhalb eines Handlungsplans bestimmt.

Schaltbedingungen: Um sicherzustellen, dass durch die konditionale Hierarchisierung abhängige Transitionen gemeinsam ausgeführt werden, wird mittels *Schaltbedingungen* ein Zusammenhang zwischen den entsprechenden Variablen in Form von Gleichungen des BIP hergestellt. Wie in Abschnitt 4.1.1.3 beschrieben, kann die Ausführbarkeit von Aktionen, die durch entsprechende Knoten der verschiedenen Hierarchieebenen repräsentiert werden, ausschließlich auf Basis der Knoten der darunterliegenden Hierarchieebene erfolgen. Entsprechend bedingt das Schalten eines Kompositionsknotens das gleichzeitige Schalten aller mittels Bindungskanten referenzierter (variablenspezifischer) Transitionen. Für die Variablen eines BIP gilt dementsprechend folgende, deklarative Charakteristika: Sollte ein Kompositionsknoten zur Ausführung durch den Planer ausgewählt werden (d.h. für die entsprechende binäre Variable wird ein Wert von Eins gewählt), müssen ebenfalls alle mit diesem Kompositionsknoten mittels Bindungskanten verbundenen Transitionen ausgeführt werden. Daher müssen in diesem Fall alle Variablen, welche mit diesem Kompositionsknoten verbundene Bindungskanten repräsentieren, ebenfalls einen Wert von Eins aufweisen. Wird ein Kompositionsknoten nicht zur Ausführung ausgewählt (d.h. Variablenwert Null), müssen die Variablen der entsprechenden Bindungskanten den selben Wert Null aufweisen. Es wird daher innerhalb des BIP gefordert, dass der Variablenwert der Kompositionsknoten stets dem Variablenwert jedes, mit diesem Knoten verbundenen Kompositionsknotens entspricht. Die Anzahl der notwendigen Schaltbedingungen für einen spezifischen Kompositionsknoten entspricht demzufolge der Anzahl der Bindungskanten dieses Kompositionsknotens. Durch die Tatsache, dass eine Transition durch mehrere Kompositionsknoten referenziert werden kann, wäre eine Variable, welche Transitionen repräsentiert, innerhalb mehreren Schaltbedingungen widersprüchlich genutzt. Die Nutzung von Variablen für Bindungskanten, welche jeweils eindeutig im Kontext eines einzelnen Kompositionsknotens genutzt werden und dabei (ebenfalls

4. Bestimmung der Ablaufreihenfolge von Automatisierungsfunktionen

eindeutig) eine spezifische Transition referenzierten, ermöglichen die vorangegangene (im Sinne des BIP widerspruchsfreie) Formulierung der Schaltsemantik.

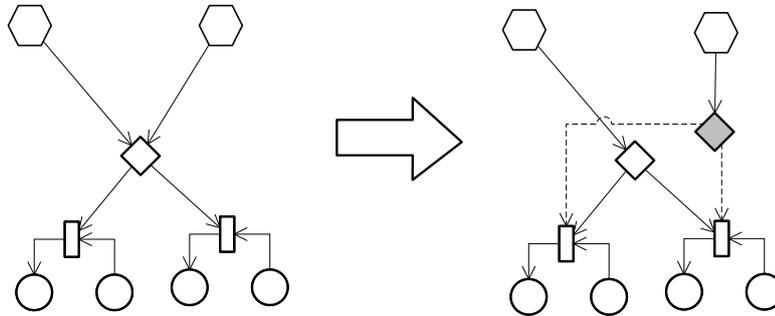


Abbildung 4.10.: Restrukturierung eines beispielhaften, internen HiTraP-AT Planungsmodells zur Erreichung der Eindeutigkeit des Kontextes von Kompositionsknoten hinsichtlich Ereignisknoten (grau hinterlegter Knoten und gestrichelte Kanten werden hinzugefügt).

Die Formulierung der Ereignisknoten kann analog zu der Schaltsemantik von Kompositionsknoten erfolgen: Wird ein Ereignisknoten durch den Planer ausgewählt (Variablenwert Eins), müssen entsprechend der Schaltsemantik der konditionalen Hierarchisierung die mittels Verbindungskanten referenzierten Kompositionsknoten ebenfalls ausgewählt werden. Wird ein Ereignisknoten nicht ausgewählt, darf ein Kompositionsknoten nicht ausgewählt werden, außer dessen Selektion wird durch die Auswahl eines anderen Ereignisknotens bedingt, welcher ebenfalls diesen Kompositionsknoten referenziert. Demzufolge ergibt sich, analog zur Schaltsemantik von Kompositionsknoten, die Notwendigkeit der Eindeutigkeit des Kontextes der Variablen von Kompositionsknoten, um eine analoge, vereinfachte Formulierung zu erreichen. Um diese Eindeutigkeit zu erreichen, ist eine partielle Erweiterung eines gegebenen internen HiTraP-AT Planungsmodells notwendig, so dass Kompositionsknoten, welche mittels mehrerer Verbindungsrelationen an unterschiedliche Ereignisknoten gebunden sind, anschließend nur eine eingehende Verbindungsrelation aufweisen (wie beispielhaft in Abbildung 4.10 dargestellt). Sind innerhalb eines Planungsmodells für einen Kompositionsknoten $n \in \mathbb{N}, n \geq 2$ eingehende Verbindungskanten enthalten, werden entsprechend $n - 1$ zusätzliche Kompositionsknoten und Bindungskanten erzeugt. Die Menge der Verbindungskanten wird dabei derart angepasst, dass $n - 1$ der n eingehende Verbindungsrelationen eines Kompositionsknotens jeweils auf einen der neuen Kompositionsknoten zeigen. Diese Transformation wird in Abbildung 4.10 exemplarisch für einen einzelnen Kompositionsknoten mit zwei eingehenden Verbindungsrelationen und zwei Bindungskanten dargestellt. Kompositionsknoten, welche im Zuge dieser Anpassung neu erzeugt wurden, müssen innerhalb der Schaltbedingung für Kompositionsknoten ebenfalls, wie oben beschrieben, berücksichtigt werden. Auf Basis dieser Transformation kann die Formulierung der Schaltbedingung für Ereignisknoten in deklarativer Form analog zur Schaltbedingung für Kompositionsknoten erfolgen: Der Wert einer Variablen, welche einen Ereignisknoten repräsentiert, muss durch den Planer

derart gewählt werden, dass er dem Wert aller Variablen entspricht, welche Kompositionsknoten repräsentieren, die mittels Verbindungskanten mit diesem Ereignisknoten verbunden sind.

Mittels dieser beiden Schaltbedingungen für Kompositions- und Ereignisknoten wird die Korrektheit der Schaltsemantik des HiTraP-AT Modells sichergestellt, welcher maßgeblich durch dessen konditionale Hierarchisierung charakterisiert ist.

Flussbedingungen: Im Zuge der Formulierung des Planungsproblems als BIP, werden charakteristische Merkmale der zu bestimmenden Handlungspläne von Modulen (vgl. Abschnitt 4.1.2) in Form von (Un-)Gleichungen beschrieben. Die Grundlage der deklarativen Formulierung der geforderten Struktur zusammenhängender Handlungspläne erfolgt mittels den hier beschriebenen *Flussbedingungen*. Die vollständige Formulierung der gesamten Semantik zusammenhängender Handlungspläne erfolgt durch Flussbedingungen, sowie den nachfolgend beschriebenen Kurzzyklus- und Integritätsbedingungen.

Durch Flussbedingungen werden die wesentlichen, strukturellen Merkmale von Handlungsräumen einzelner Module beschrieben. Dabei wird ausschließlich auf Stellen- und Transitionsknoten der untersten Hierarchieebene des internen HiTraP-AT Planungsmodells, d.h der Variablenebene, zurückgegriffen. Innerhalb variablenspezifischer S/T-Netze werden drei grundlegende Arten von Stellen genutzt: *Initialstellen*, welchen eine Anfangsmarkierung zugewiesen ist und entsprechend einen Anfangszustand darstellen (sowohl für Variablen des Automatisierungssystems als auch des Produktes), *Zielstellen*, welche den Zielzustand einer Variablen des Produktes beschreiben sowie *innere Stellen* von System oder Produkt, welche alle verbleibenden Stellen darstellen, die keiner der zuvor genannten Arten zugehörig sind.

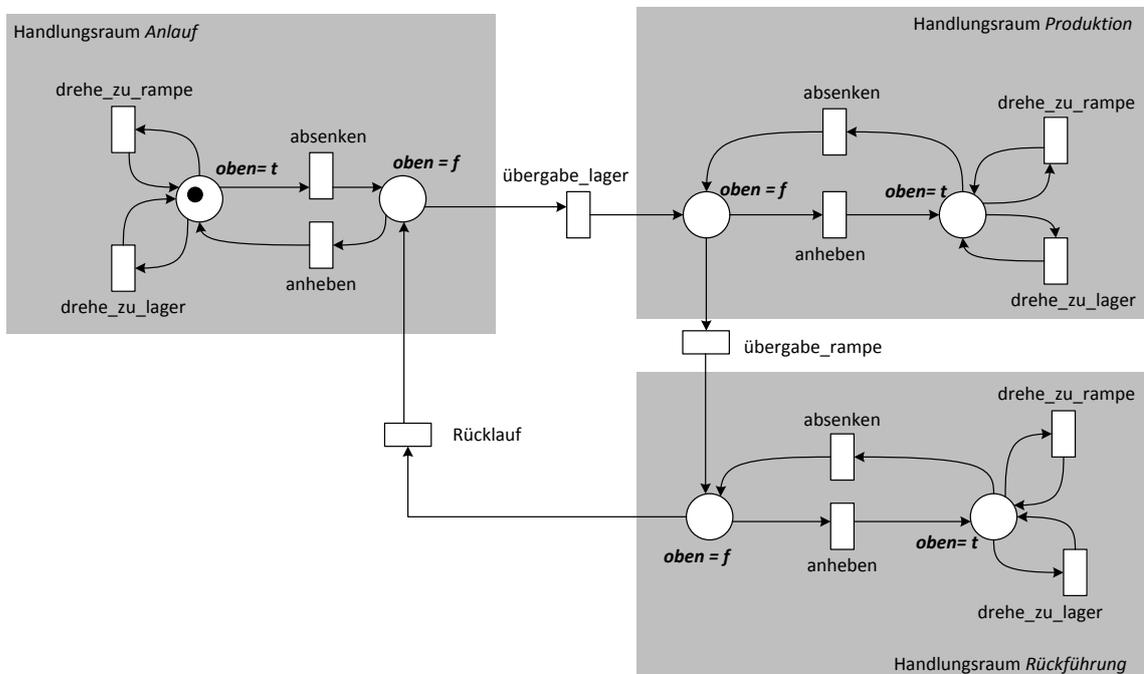


Abbildung 4.11.: Variablenspezifisches S/T-Netz der Variablen *oben* des Kranmoduls.

4. Bestimmung der Ablaufreihenfolge von Automatisierungsfunktionen

Wie in Abbildung 4.11 exemplarisch visualisiert, weisen einige Stellen im Handlungsplan "Anlauf" eine weitere, besondere Charakteristik auf: Während in ihrem Nachbereich Initialisierungstransitionen enthalten sind, existieren entsprechende Rücklauftransitionen in ihrem Vorbereich. Um eine einfache Formulierung der Flussbedingungen für diese Stellen zu ermöglichen, werden Rücklauftransitionen aus dem Vorbereich der Stellen im Handlungsplan "Anlauf" nicht berücksichtigt. Auf diese Weise ist eine gesonderte Formulierung dieser Initialisierungsstellen im Rahmen der Flussbedingungen nicht notwendig; ihre besonderen Merkmale werden jedoch im Rahmen der Integritätsbedingungen erneut aufgegriffen und geeignet berücksichtigt.

Um die Dynamik des Systems und die Korrektheit des zu bestimmenden Handlungsplanes sicherzustellen, sollte ein Modul nicht in der initialen Markenbelegung verweilen, falls dieses Modul zur Nutzung innerhalb des Handlungsplanes durch den Planer selektiert wird. In diesem Fall muss daher stets eine Transition im Nachbereich der Initialstellen durch den Planer ausgewählt werden. Trotz der strikten Trennung der Handlungsräume "Anlauf", "Produktion" und "Rücklauf" – wodurch ggf. mehrere Stellen eine identische Variablenbelegung in den unterschiedlichen Kontexten repräsentieren – kann die Auswahl eines zyklischen Ablaufs notwendig sein (vgl. Abbildung 4.11). Aus diesem Grund wird innerhalb der *Flussbedingungen für Initialstellen der Module* diese Bedingung derart erweitert, so dass zyklische Abläufe innerhalb eines Handlungsraumes, in dem eine Initialstelle enthalten ist, berücksichtigt werden. Entsprechend muss für Initialstellen der Module gelten, dass die Anzahl der durch den Planer ausgewählten Transition im Nachbereich der Initialstellen die Anzahl der selektierten Transitionen im Vorbereich stets um eins übersteigen muss. Wird ein Modul jedoch nicht genutzt, ist eine Auswahl von Transitionen dieses Moduls nicht notwendig und sollte vermieden werden, um eine unnötige Ausführung von Funktionen zu verhindern. Letzteres wird durch eine minimierende Bewertungsfunktion sichergestellt, wohingegen die Anzahl der zu selektierenden Transitionen in den Flussbedingungen festgelegt ist. Dies ist in bisher beschriebenen Überlegungen noch nicht ausreichend berücksichtigt. Daher wird in den Flussbedingungen für Initialstellen der Module auf die entsprechende Modulvariable zurückgegriffen: Die Differenz der Anzahl selektierter Transitionen im Nach- und Vorbereich von Initialstellen der Module muss dem durch die Modulvariablen gegebenen Wert entsprechen. Wird ein Modul durch den Planer selektiert, d.h. die Modulvariable weist den Wert Eins auf, gilt eingangs beschriebener Sachverhalt. Wird ein Modul nicht gewählt (Wert der Modulvariable ist Null), wird gefordert, dass die Anzahl der selektierten Transitionen in Vor- und Nachbereich identisch sind. Dies führt letztendlich im Zusammenhang mit einer Minimierungsbedingung dazu, dass keine Transition gewählt wird, falls das Modul nicht innerhalb eines Handlungsraumes enthalten ist. Für *Flussbedingungen der Initialstellen des Produktes* gelten analoge Überlegungen. Im Gegensatz zu Flussbedingungen der Initialstellen der Module ist jedoch keine Bezugnahme auf eine spezifische Modulvariable notwendig, da Marken die Initialstellen von Variablen des Produktes verlassen müssen, um die Zielstellen zu erreichen⁴⁴, wie exemplarisch in Abbildung 4.12 am Beispiel der Variablen *Ort* des Produktes dargestellt. Entsprechend wird innerhalb des BIP für Initialstellen des Produktes gefordert, dass exakt eine Transitionen mehr im Nachbereich dieser Stellen durch den Planer selektiert wird als in dessen Vorbereich.

Im Gegensatz zu Initialstellen bedingt ein korrekter Handlungsplan, dass Marken innere Stellen ggf. passieren, diese jedoch in diesem Fall stets wieder verlassen. Betrachtet man das in Abbil-

⁴⁴Hierbei wird auf die dem Problemmodell zu Grunde gelegte Annahme Bezug genommen, dass Start- und Zieldefinition einer Variablen des Produktes (d.h. dessen Wert) nicht identisch sind.

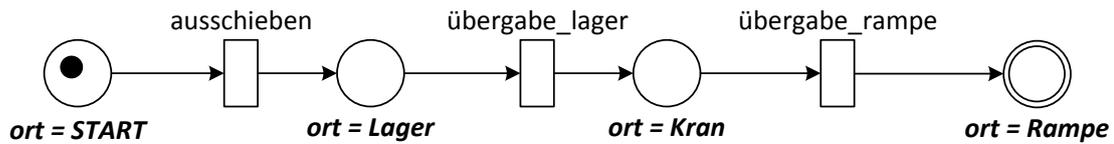


Abbildung 4.12.: Variablenspezifisches S/T-Netz der Variablen *Ort* des Produktes in der Stempelanlage.

Abbildung 4.11 gegebene Beispiel für eine Variable des Kranmoduls, so muss für die Stelle "oben=f" im Handlungsplan "Produktion" jeweils zwei Transitionen in dessen Vor- und Nachbereich selektiert werden, um einen korrekten Handlungsplan zu bestimmen, während für die Transition "oben=t" im selben Handlungsraum jeweils nur eine Transition ausgewählt werden muss. Auch für Variablen des Produktes gelten analoge Überlegungen. Aus diesem Grund stellen *Flussbedingungen für innere Stellen* der Module und des Produktes sicher, dass die Anzahl der durch den Planer ausgewählten Transitionen im Vorbereich der Anzahl der selektierten Transitionen im Nachbereich einer inneren Stellen entspricht. Durch diese Formulierung werden einerseits etwaige zyklische Abläufe unterstützt sowie andererseits ermöglicht, dass Stellen nicht genutzt werden. In letzterem Fall wird weder eine Transition im Vor- noch im Nachbereich ausgewählt. Eine gesonderte Betrachtung für Stellen des Moduls unter Bezugnahme auf dessen Nutzung, wie beispielsweise im Falle der Flussbedingungen für Initialstellen, ist hierbei aus diesem Grund nicht notwendig. Wird ein Modul nicht genutzt, werden auf Grund der minimierenden Bewertungsbedingung sowie der entsprechenden Flussbedingung für Initialstellen keine Transitionen für innere Stellen durch den Planer selektiert. Anzumerken ist an dieser Stelle, dass Rücklauftransitionen im Nachbereich entsprechender Stellen enthalten sind⁴⁵ und folglich diese Stellen im Handlungsraum "Rücklauf" im Rahmen der Flussbedingungen für innere Stellen Berücksichtigung finden.

Durch die zuvor beschriebenen Flussbedingungen werden alle Anforderungen hinsichtlich der Korrektheit des zu bestimmenden Handlungsplanes für die Module des AT-Systems in Form entsprechender Gleichungen spezifiziert. Des Weiteren gelten die Flussbedingungen für Initialstellen sowie innere Stellen ebenfalls für entsprechende Stellen der Produktvariablen, da diese identische Charakteristik aufweisen. Im Gegensatz zu dem für Module des AT-Systems zu bestimmenden Handlungsplan, ist für Variablen des Produktes eine Zieldefinition gegeben, d.h. für jede Variable des Produktes ist eine Zielstelle spezifiziert. Für diese Zielstellen sollte der Handlungsplan durch den Planer derart bestimmt werden, dass Marken des S/T-Netzes diese Stellen letztendlich nicht mehr verlassen (vgl. Abbildung 4.12). Auch hier sind grundsätzlich zyklische Abläufe denkbar, so dass eine Marke eine Zielstelle nochmals verlässt, bevor sie letztendlich final dort verweilt. Aus diesem Grund wird innerhalb der *Flussbedingungen für Zielstellen* des Produktes (ähnlich den Flussbedingungen der Initialstellen) gefordert, dass die Anzahl der durch den Planer ausgewählten Transitionen im Vorbereich einer Zielstelle die Anzahl der selektierten Transitionen im Nachbereich um exakt eins übersteigt.

⁴⁵Rücklauftransitionen finden im Nachbereich der Stellen Berücksichtigung, nicht jedoch im Vorbereich.

4. Bestimmung der Ablaufreihenfolge von Automatisierungsfunktionen

Integritätsbedingungen: Um die geforderte, spezifische Struktur eines Handlungsplanes bestehend aus "Anlauf", "Produktion" und "Rücklauf", und die damit verbundenen Eigenschaften eines zu bestimmenden Handlungsplanes sicherzustellen, sind weitere Bedingungen innerhalb eines BIP notwendig. Darin wird fixiert, dass für ein Modul, falls dieses vom Planer zur Verwendung ausgewählt wird (gegeben durch die entsprechende Modulvariable), jeweils exakt eine Initialisierungs-, Finalisierungs- sowie Rücklaufaktion eines Moduls selektiert werden muss. Da durch die Schaltbedingung bereits das gemeinsame Schalten notwendiger Transitionen sichergestellt wurde, kann die Definitionen der Integritätsbedingungen auf Basis einer einzelnen Knotenmenge (Transitions-, Kompositions- oder Ereignisknoten) erfolgen. Auf Grund der kausalen Abhängigkeit der Integritätsbedingung von der Nutzung eines Moduls, was wiederum Teil der durch den Planer zu bestimmenden Lösung ist, erfolgt eine Definition der Integritätsbedingungen auf Basis der entsprechenden Kompositionsknoten. Dementsprechend werden innerhalb des BIP für jedes Modul drei zusätzliche Integritätsbedingungen (jeweils für Initialisierung, Finalisierung und Rücklauf) spezifiziert. Diese stellen dabei sicher, dass die Summe der Variablen, welche entsprechende Kompositionsknoten repräsentierenden (entweder Initialisierungs-, Finalisierungs- oder Rücklaufaktionen) dem Wert der Modulvariable entspricht. Wird ein Modul durch den Planer selektiert (d.h. Wert von Eins der entsprechenden Modulvariablen), wird der Planer durch das deklarative Modell in Form von Integritätsbedingungen gezwungen, ebenfalls exakt eine der entsprechenden Variablen für Initialisierung, Finalisierung und Rücklaufinitialisierung auszuwählen. Wird ein Modul innerhalb einer Lösung nicht genutzt (d.h. Wert Null der entsprechenden Modulvariablen), wird durch die Integritätsbedingungen sichergestellt, dass keine dieser Aktionen ausgewählt wird. Um die geforderte zyklische Struktur des Handlungsplanes bestehend aus Teilplänen für "Produktion" und "Rücklauf" sicherzustellen, muss ferner mittels Integritätsbedingungen innerhalb des BIP sichergestellt werden, dass für eine Stelle, für die eine Initialisierungstransitionen im Nachbereich ausgewählt wurde ebenfalls eine Rücklauftransition in dessen Vorbereich durch den Planer ausgewählt wird. Hierzu wird innerhalb des BIP ergänzend gefordert, dass für jede Stelle (welche im Nachbereich eine Initialisierungstransition enthält) die Anzahl der selektierten Initialisierungstransitionen der Anzahl der ausgewählten Rücklauftransitionen entspricht. In Zusammenhang mit den bereits zuvor beschriebenen Integritätsbedingungen, welche verhindern, dass mehr als eine dieser Transitionen durch den Planer ausgewählt wird, wird die geforderte zyklische Struktur des zu bestimmenden Handlungsplanes sichergestellt.

Kurzzyklusbedingungen: In den bisher beschriebenen Bedingungen wurde die gewünschte Struktur des Handlungsplanes ausschließlich unter Berücksichtigung einzelner Knoten des HiTraP-AT Modells durchgeführt. Die deklarative Beschreibung struktureller Anforderungen an Graphen (hier in diesem Fall des Handlungsplanes) in Form eines Linearen Programms wurde in Theorie und Praxis bereits intensiv untersucht. Dabei konnte gezeigt werden, dass diese Formulierung ggf. nicht ausreichend ist um sicherzustellen, dass selektierte Transitionen einen geschlossenen Weg innerhalb des Graphen beschreiben. Wie in Abbildung 4.13 dargestellt, können obige Bedingungen auch in eine Partitionierung variablen spezifischer S/T-Netze resultieren, welche durch sogenann-

te Kurzzyklen verursacht wird⁴⁶. Eine Auswahl zusätzlicher Transitionen ist zwar durch eine minimierende Bewertungsfunktion ausgeschlossen. Wird jedoch durch eine entsprechende konditionale Hierarchisierung eine Auswahl dieser Knoten gefordert (Schaltbedingung), kann eine Partitionierung mit den bisherigen Bedingungen nicht ausgeschlossen werden.

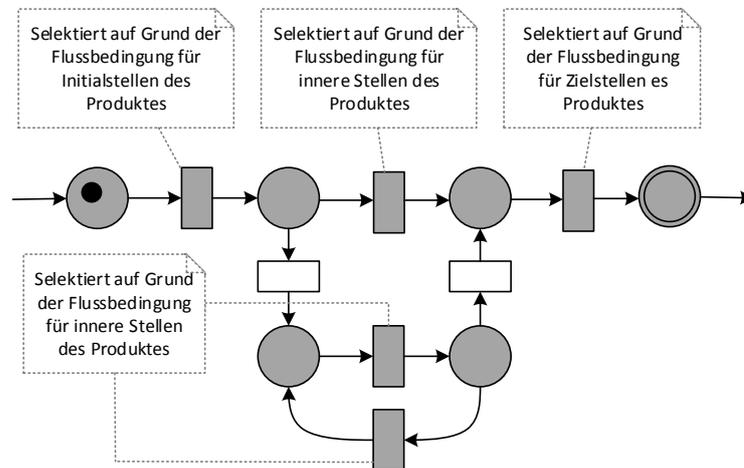


Abbildung 4.13.: Beispiel eines variablen-spezifischen S/T-Netzes mit Kurzzyklus (selektierte Stellen und Transitionen sind grau ausgefüllt).

Kurzzyklen und die daraus resultierende Partitionierung von Graphen stellt in unterschiedlichen Anwendungsfällen bei einer Formulierung des Planungsproblems auf Basis struktureller Eigenschaften als Lineares Programm eine Herausforderung dar, wie z.B. in der Wege- und Routenplanung in der Logistik [BCGL07]. Die Herausforderung von Kurzzyklen wurde jedoch bereits intensiv theoretisch insbesondere im Bereich des sogenannten Problems des Handlungsreisenden (engl. Traveling Salesman Problem) untersucht⁴⁷. Dabei konnten bereits unterschiedliche, korrekte Lösungen zur Vermeidung der Kurzzyklen identifiziert werden, welche letztendlich auf das Hinzufügen unterschiedlicher, zusätzlicher Bedingungen (hier auch Subtour-Eliminationsbedingungen genannt) zurückgeführt werden [Lap86, HPR13]. Variablen-spezifische S/T-Netze, welche von einer potentiellen Partitionierung betroffen wären, können als Graph im Sinne dieser etablierten Lösungen betrachtet werden. Für die Formulierung des internen HiTraP-AT Planungsmodells kann daher vollständig auf diese etablierten Ansätze zurückgegriffen werden.

Bewertungsfunktion: Mittels der zuvor in diesem Abschnitt eingeführten Bedingungen in Form von (Un-)Gleichungen werden die Bedingungen eines Linearen Programms vollständig ebenso gegeben wie die Menge der Variablen und deren binäre Definitionsmenge. In dem vorherigen

⁴⁶Die Abbildung zeigt lediglich den Ausschnitt eines fiktiven S/T-Netzes einer Produktvariablen, da eine Partitionierung ausschließlich bei Variablen mit größeren Definitionsmengen auftreten kann als dies im Beispiel der Stempelanlage der Fall ist. Für kleinere Definitionsmengen z.B. bei binären Variablen, ist das Vorkommen von Kurzzyklen bereits durch Integritätsbedingung sowie Flussbedingungen für Initialstellen und ggf. Zielstellen ausgeschlossen.

⁴⁷Das Problem des Handlungsreisenden fordert, dass in einem gegebenen Graphen alle Knoten im Zuge eines geschlossenen, zyklischen Weges besucht werden müssen [HPR13]

4. Bestimmung der Ablaufreihenfolge von Automatisierungsfunktionen

Abschnitt 4.1.3.1 wurde bereits identifiziert und erläutert, dass eine Handlungsplanung für die Steuerungsebene des Maschinen- und Anlagenbaus die Anwendung eines Optimierungskriteriums bedingt (siehe auch Anforderung A2). Die Integration eines Optimierungskriteriums mittels einer Bewertungsfunktion B innerhalb eines Linearen Programms ist dabei bereits vorgesehen und kann für die hier vorgeschlagene Formulierung der HiTraP-AT Handlungsplanung als LP genutzt werden.

Eine gegebene Kostenfunktion weist innerhalb der Definition des Planungsproblems jeder Aktion einen spezifischen Kostenwert in Form einer natürlichen Zahl zu (vgl. Abschnitt 4.1.2). Innerhalb eines HiTraP-AT Modells repräsentieren die jeweils höchsten Knoten der konditionalen Hierarchie, d.h. für Handlungsräume "Anlauf" und "Rücklauf" die Kompositionsknoten bzw. für Handlungsräume "Produktion" die Ereignisknoten, vollständig die Ausführung einer Aktion. Die Bewertungsfunktion stellt dabei eine Minimierung der Summe der durch die Kostenfunktion definierten Kosten dar, wobei die Skalare der Summe die die Kompositionsknoten repräsentierenden Variablen darstellen. Da es sich hier um ein binäres Lineares Programm handelt, werden durch Bestimmung eines Wertes für Variablen durch das Lösungsprogramm entsprechend die Kosten innerhalb der Bewertungsfunktion berücksichtigt. Als optimal wird hierbei durch die Bewertungsfunktion die Minimierung der Kosten des sequenzierten partiell geordneten Planes betrachtet⁴⁸. Innerhalb der Bewertungsfunktion können ferner mittels Koeffizienten die Kosten der Ausführung von Aktionen innerhalb der verschiedenen Handlungsräume eines Moduls unterschiedlich gewichtet werden. Die Wahl der konkreten Werte für die Koeffizienten ist dabei sowohl von dem Optimierungsziel selbst als auch von der innerhalb der Kostenfunktion für Aktionen gegebenen Kosten abhängig.

Wie in Anforderung A2 gefordert und in Abschnitt 4.1.3.1 konkretisiert, stellt die Maximierung des Durchsatzes ein essentielles Optimierungskriterium in der Massenfertigung dar. Um eine Minimierung der Durchlaufzeit eines Produktes zu erreichen, ist eine Auswahl möglichst weniger Aktionen innerhalb der Handlungsräume "Produktion" zielführend. Entsprechend müssen diese Kosten innerhalb des Linearen Programms proportional deutlich höher gewichtet werden, als die Ausführung von Aktionen an anderer Stelle (d.h. in den Handlungsräumen "Anlauf" und "Rücklauf"). Entsprechend dem Gesetz von Little ist ferner, wie in Abschnitt 4.1.3.1 bereits erläutert, der Umlaufbestand von essentieller Bedeutung im Hinblick auf eine durchsatzoptimierte Produktion. Der minimal mögliche Abstand zweier Produkte ist dabei davon abhängig, wann durch ein Modul erneut ein Produkt aufgenommen werden kann. Dies ist durch die vom Planer selektierten Aktionen des Handlungsraumes "Rücklauf" der Module bestimmt. Unter ggf. vorhanden validen Lösungen des Gleichungssystems sollte demzufolge die Lösung ausgewählt werden, welche neben der bereits zuvor (und insgesamt bedeutendsten Gewichtung) minimale Kosten im Hinblick auf die selektierten Aktionen des Handlungsraumes "Rücklauf" erreicht. Folglich gilt in diesem Zusammenhang im Hinblick auf die Gewichtungsfaktoren zusätzlich, dass der Koeffizient für die Gewichtung der Kosten der selektierten Aktionen im Handlungsraum "Rücklauf" höher gewichtet werden sollte, als die des Handlungsraumes "Anlauf". Zusammenfassend sind die Koeffizienten für eine Durchlaufoptimierung also derart zu wählen, dass Aktionen des Handlungsraumes "Produktion" signifikant höher gewichtet werden, als die des Handlungsraumes "Rücklauf", welche wiederum deutlich höher als Aktionen des Handlungsraumes "Anlauf" gewichtet werden.

⁴⁸Auf Grund der notwendigen Linearität der Bewertungsfunktion ist eine vollständige Optimalität im Hinblick auf partiell geordnete und dementsprechende parallel ausführbare Aktionen nicht berücksichtigt.

Der konkrete, minimal sinnvollen Wert der Gewichtungskoeffizienten richtet sich dabei nach den durch die Kostenfunktion definierten Werten. Wird beispielsweise der Koeffizient für den Handlungsraum "Produktion" im Vergleich zu den anderen Koeffizienten zu gering gewählt, könnte der Planer (d.h. der Löser des Linearen Programms) dazu veranlasst werden, zusätzliche Aktionen innerhalb des Handlungsraumes "Produktion" zu selektieren, da dies zu einer Minimierung der gesamten Bewertungsfunktion führen würde, falls hierdurch eine vergleichsweise sehr kostenintensive Aktion in dem Handlungsraum "Anlauf" eingespart werden würde. Hierdurch wird jedoch nicht das notwendige Optimierungskriterium erfüllt, da dies zu einer unerwünschten Steigerung der Durchlaufzeit führt. Eine Wahl sehr großer Werte der Koeffizienten hat jedoch keinen Einfluss auf die algorithmische Laufzeit des Lösungsverfahrens linearer Programme, weshalb heuristisch beliebig hohe Werte gewählt werden, um die geforderte Optimalität sicherzustellen und letztendlich die unterschiedlichen Gewichtungen innerhalb der verschiedenen Handlungsräume durch den Planer selektierten Aktionen deutlich zu machen⁴⁹.

4.1.3.3. Bestimmung des Handlungsplans durch Lösung des Linearen Programms

Auf Basis der BIP Formulierung von HiTraP-AT, wie sie im vorangegangenen Abschnitt detailliert beschrieben wurde, kann durch ein beliebiges Lösungsprogramm für diese Art Linearer Programme (d.h. BIP) eine automatische Handlungsplanung erfolgen. Für dieses BIP werden durch einen solchen Löser geeignete Belegungen der Variablen bestimmt. Diese durch den Löser identifizierte Lösung wird in der Terminologie der Constraint Programmierung auch als Konfiguration bezeichnet (vgl. Abschnitt 3.1.2). Wird für eine Variable der Wert Eins innerhalb einer als Lösung bestimmten Konfiguration durch den Löser gewählt, kann dies als Auswahl des dieser Variable entsprechenden Modellelements innerhalb des HiTraP-AT Modells aufgefasst werden (wie dies bereits im vorangegangenen Abschnitt detailliert beschrieben wurde). Während eine Vielzahl der genutzten Variablen ausschließlich der korrekten Abbildung des durch HiTraP-AT vorgegebenen Modells und dessen Schaltsemantik dienen, sind für die Bestimmung des Handlungsplans auf Basis der Lösung des Linearen Programms primär Variablen von Bedeutung, welche Kompositions- und Ereignisknoten repräsentieren. Grund hierfür ist die Tatsache, dass diese Knotentypen einzelne Aktionen des Planungsproblems vollständig beschreiben (wie dies bereits für die Bewertungsfunktion ebenfalls beschrieben wurde). Auf diese Weise können selektierte Aktionen und ihre Reihenfolge bzw. Abhängigkeiten identifiziert werden. Die Ableitung eines Handlungsplanes bedingt zusätzliche Nachbearbeitung.

Ausgehend vom Initialzustand eines betrachteten AT-Systems und dessen Repräsentation innerhalb des HiTraP-AT Modells durch die initiale Markenbelegung, kann der Handlungsplan für jedes einzelne Modul bestimmt werden, in dem durch den Planer das HiTraP-AT Modell mittels selektierter Aktion traversiert wird bis der Zustand erreicht ist, welcher lediglich eine Rücklauftransition als ausgehende Transition ausweist. Dieser Zustand kann auch als Zielzustand des Handlungsplans eines Moduls verstanden werden. Innerhalb des HiTraP-AT Modells existieren auf Grund der konditionalen Hierarchisierung jedoch keine direkten Kanten zwischen Ereignis- bzw. Kompositionsknoten. Die Abhängigkeit der Ausführbarkeit zwischen durch diese Knoten innerhalb des HiTraP-AT Modells repräsentierten Aktionen lässt sich mittels der in Ab-

⁴⁹In den durchgeführten Planungen für unterschiedliche Systemfallstudien hat sich eine Differenz der Gewichtungskoeffizienten jeweils von 10^4 (zwischen den beiden Koeffizient für "Produktion" und "Rücklauf", sowie wiederum für "Rücklauf" und "Anlauf") als zielführend erwiesen.

4. Bestimmung der Ablaufreihenfolge von Automatisierungsfunktionen

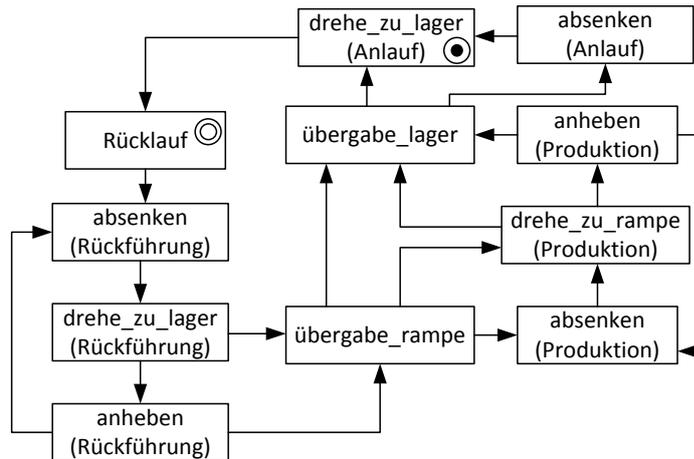


Abbildung 4.14.: Exemplarischer Abhängigkeitsgraphen des Kranmoduls der Stempelanlage.

schnitt 4.1.1.2 beschriebenen Abhängigkeitsdefinition auf Basis variablenspezifischer S/T-Netze bestimmen: die Ausführung einer Aktion A ist dabei von einer anderen Aktion B abhängig, falls für Aktion A innerhalb mindestens eines variablenspezifischen S/T-Netzes eine Transition existiert, die Aktion A repräsentiert und deren Vorbereich eine Stelle enthält, welche sich im Nachbereich der Transition befindet und die Aktion B repräsentiert. Auf Basis der durch den Planer selektierten Aktionen (gegeben durch die Variablen der Ereignis- und Kompositionsknoten mit Variablenwert Eins), kann mittels dieser Abhängigkeitsdefinition ein entsprechender Abhängigkeitsgraph bestimmt werden. Abbildung 4.14 zeigt den Abhängigkeitsgraph für das Kranmodul der Stempelanlage nach einer durchgeführten Planung. Die gegebenen Knoten des Abhängigkeitsgraphen entsprechen dabei den ausgewählten Aktionen (abgeleitet durch die von einem Planer ausgewählten Ereignis- bzw. Kompositionsknoten). Die Abhängigkeit zwischen den Aktionen, welche durch Kanten innerhalb des Graphen dargestellt sind, ergeben sich hierbei vollständig aus dem variablenspezifischen S/T-Netz der Variablen "oben" des Krans (vgl. Abbildung 4.11). Vollständig unabhängige Aktionen sind innerhalb des Abhängigkeitsgraphen dadurch charakterisiert, dass zwischen diesen Aktionen kein korrekten Weg innerhalb des Graphen existiert.

Die Ableitung der (partiell geordneten) Handlungspläne kann auf Basis solcher Abhängigkeitsgraphen als Graphensuchproblem aufgefasst werden: Zu Bestimmen ist ein Weg von Zielknoten zu Startknoten des Abhängigkeitsgraphen bei dem jeder Knoten genau einmal besucht wird und keine Kante mehrfach Verwendung findet⁵⁰. Da der Abhängigkeitsgraph im Hinblick auf die Ausführungsreihenfolge von Aktionen invers gerichtete Kanten enthält, wird der Startknoten durch die Rücklauftransition gebildet, wohingegen der Zielknoten durch die erste Aktion gebildet wird, welche dadurch charakterisiert ist, dass diese innerhalb des Graphen ausschließlich eingehende Kanten aufweist. Die Bestimmung des Handlungsplanes eines Moduls lässt sich dabei

⁵⁰Wird eine Aktion mehrfach innerhalb eines einzelnen Handlungsplanes ausgeführt, wird die Aktion ebenfalls mehrfach innerhalb des Abhängigkeitsgraphen aufgeführt. Grund hierfür ist, dass die Abhängigkeit auf Basis der Zustände vor bzw. nach Aktionsausführung bestimmt wird. Eine Aktion, welche mehrfach ausgeführt wird, wird dabei in unterschiedlichen Zuständen ausgeführt und weist daher ggf. auch entsprechend unterschiedliche Abhängigkeiten innerhalb der variablenspezifischen S/T-Netze auf.

direkt auf das theoretische Problem des Handlungsreisenden abbilden: jede Aktion, welche in einem Abhängigkeitsgraphen gegeben ist, muss innerhalb eines Handlungsplanes enthalten sein (d.h. alle Knoten innerhalb des Graphen müssen besucht werden); die im Handlungsplan definierte Reihenfolge von Aktionen muss dabei jedoch im Hinblick auf die Abhängigkeiten der Aktionen korrekt sein (d.h. lediglich Kanten innerhalb des Abhängigkeitsgraphen können für die Bestimmung von Reihenfolgen genutzt werden). Wie zuvor beschrieben, existieren zwischen vollständig unabhängigen Aktionen keine Kanten innerhalb eines Abhängigkeitsgraphen, da zwischen diesen Aktionen keine Abhängigkeit auf Basis von Variablen des variablen-spezifischen S/T-Netzes existiert. Um die Bestimmung eines Handlungsplanes auch in diesem Fall auf das Problem des Handlungsreisenden zurückführen zu können, müssen für unabhängige Aktionen zusätzliche bidirektionale Kanten (bzw. zwei gegenläufige, unidirektionale Kanten) eingefügt werden. Hierdurch können die Kanten, wie für eine Formulierung als Problem des Handlungsreisenden notwendig, semantisch korrekt als ordnungsgebendes Element des Abhängigkeitsgraphen für die Bestimmung des Handlungsplans interpretiert werden. Fügt man alle auf diese Weise bestimmten, modul-spezifischen Handlungspläne auf Basis gemeinsamer enthaltener Aktionen zusammen, ergibt sich ein vollständiger Handlungsplan der Anlage. Abbildung 4.15 stellt einen Ausschnitt des gesamten Handlungsplanes der Stempelanlage dar, welcher die bereits in Abschnitt 4.1.2 beschriebenen Charakteristiken des HiTraP-AT Handlungsplanes aufweist.

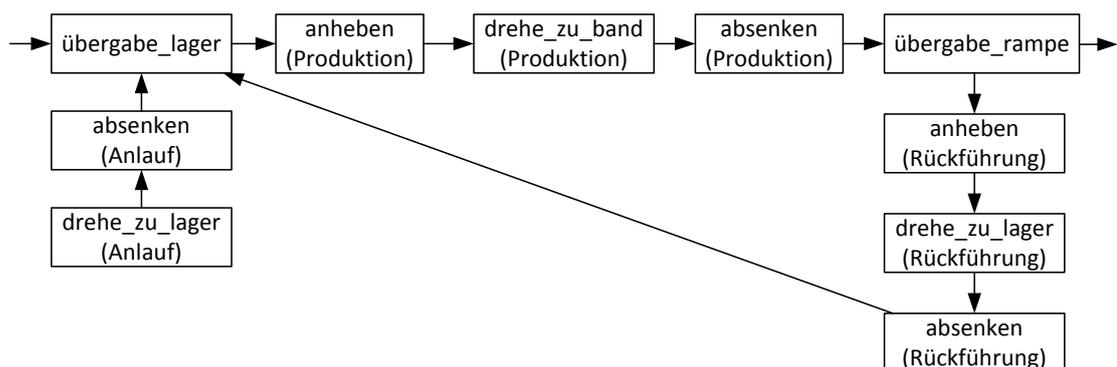


Abbildung 4.15.: Ausschnitt eines charakteristischen HiTraP-AT Handlungsplanes der Stempelanlage entsprechend den in Abbildung 4.14 dargestellten Informationen.

4.2. Integration von HiTraP-AT in einen modellgetriebenen Entwicklungsansatz (Anforderung A5)

Im Rahmen der Anforderungsermittlung (vgl. Abschnitt 2.2) wurde identifiziert, dass – insbesondere im Hinblick auf die Anwendbarkeit und Benutzerfreundlichkeit – eine Integration des Planungsverfahrens in modellgetriebene Entwicklungsansätze eine notwendige Anforderung darstellt (Anforderung A5). Grund hierfür ist unter anderem der große (ggf. im Entwicklungsprozess zusätzliche) manuelle Aufwand für die Erstellung der Eingabe- bzw. Planungsmodelle. Um der Anforderung an ein für den Maschinen- und Anlagenbau geeignetes Planungsverfahren gerecht

4. Bestimmung der Ablaufreihenfolge von Automatisierungsfunktionen

zu werden, wird im Rahmen dieses Kapitels die Möglichkeit der Integration des HiTraP-AT Ansatzes mit einem modellgetriebenen Entwicklungsansatz vorgeschlagen.

Im Rahmen gemeinsamer Vorarbeiten mit Schütz et al. [LSVH13, LSF⁺13, LSVH14, SLVH12] wurde ein Modellierungskonzept entwickelt, um den Zustandsraum SPS-basierter Software in konsolidierter Weise zu beschreiben. Hierzu wurde für diesen konzeptuellen Ansatz ebenfalls ein geeignetes UML Profil für die modellgetriebene Entwicklung in der Dissertation von Daniel Schütz vorgeschlagen [Sch14]: Die sogenannte UML-OP. Die Namensgebung des UML-OP Profils ist dabei auf die Zentrierung des Modellierungsansatzes auf Operationen eines Fertigungssystems – UML für Operationen – zurückzuführen [Sch14, S.130]. Im weiteren Verlauf dieses Abschnitts werden die charakteristischen Merkmale der UML-OP (soweit für das Verständnis notwendig) eingeführt, um anschließend die Abbildung zwischen UML-OP und dem HiTraP-AT Modell beschreiben zu können.

4.2.1. Die UML-OP

Zur Steigerung der Wiederverwendung von Modellierungselementen wird in der UML-OP ein konsequent objektorientierter Ansatz verfolgt, in dem einzelne Komponenten und deren Funktionalitäten in einem Komponentenmodell beschrieben werden. Dabei wird stets die Prämisse verfolgt, eine vollständige Beschreibung der Funktionalität der Komponente zu erreichen, unabhängig etwaiger Einschränkungen, die sich durch deren Einsatz in einem Gesamtsystem ergeben. Ein Gesamtsystem wird in einem Systemmodell unter Rückgriff auf bereits modellierte Komponenten beschrieben, ermöglicht aber die Einschränkung der Funktionalität von Komponenten, die sich durch deren Einsatz innerhalb des Systems ergeben. Ferner werden Anforderungen bzgl. des Fertigungsprozesses festgelegt und damit verbundene Charakteristika, die ein Werkstück vor und nach Ausführung einzelner Funktionalitäten aufweist, in einem Produktmodell beschrieben. Im weiteren Verlauf dieses Abschnitts werden die Modellierungskonzepte und die UML-OP eingeführt, um die Abbildung auf HiTraP-AT, welche im nachfolgenden Abschnitt beschrieben ist, nachvollziehen zu können. Für eine präzise Spezifikation des UML-Profiles sei speziell auf die Arbeiten von Daniel Schütz – insbesondere [Sch14, SLVH12] – verwiesen.

4.2.1.1. Das Komponentenmodell der UML-OP

Mit dem Ziel, die Wiederverwendung entwickelter Modelle und Lösungen zu steigern [LSF⁺13, LSF⁺14], stellen Anlagenkomponenten voneinander unabhängige, mehrfach wiederverwendbare Einheiten (Klassen) dar. Neben den installierten Sensoren (Variablen), werden Operationen (Methoden) der Komponenten beschrieben. Da die UML-OP die Zielsetzung verfolgt, eine konsolidierte Beschreibung des Zustandsraumes einer Maschine zu ermöglichen [SLVH12], wird das (diskrete) Verhalten gegebener Operationen näher spezifiziert. UML-OP ermöglicht dabei die Modellierung, wann eine Operation ausführbar ist (Vorbedingungen) und in welchem Zustand die Ausführung der Operation resultiert (Effekte). Die Vorbedingungen und Effekte werden in Abhängigkeit zu den Variablen, d.h. Sensoren, der Komponentenklasse definiert. Diese Sachverhalte werden im Folgenden näher betrachtet.

Um alle diese Informationen geeignet anzugeben, wurde eine tabellarische Notation gewählt. In Abbildung 4.16 ist exemplarisch die Beschreibung der Krankomponente der Stempelanlage gegeben.

Kran		
<i>attributes:</i>		
B3, B4, B5: boolean B6: integer		
<i>operations:</i>		
dreheLinks() dreheRechts() anheben() absenken() ansaugen() loslassen()		

Kran		
B3: boolean ; B4: boolean ; B5: boolean ; B6: integer		
Operationname	Vorbedingung	Effekt
dreheLinks()	$B6 + \Delta < 360$	$B6 \rightarrow B6@pre + \Delta$
dreheRechts()	$B6 - \Delta > 0$	$B6 \rightarrow B6@pre - \Delta$
anheben()	NOT B3 AND B4	$B3 \rightarrow true$ AND $B4 \rightarrow false$
absenken()	B3 AND NOT B4	$B3 \rightarrow false$ AND $B4 \rightarrow true$
ansaugen()	NOT B5	$B5 \rightarrow true$
loslassen()	B5	$B5 \rightarrow false$

Abbildung 4.16.: Tabellarische Notation der Funktionalität von Modulen/Komponenten (in Anlehnung an [LSF⁺13, LSVH14]).

Die Krankomponente der Stempelanlage (vgl. Kapitel A.1), besteht wie in Abbildung 4.16 dargestellt, aus vier Sensoren, die durch entsprechende Variablen $B3$ bis $B6$ im Modell repräsentiert sind. Die binären Variablen $B3$ und $B4$ stellen die binäre Positionssensoren des für eine vertikale Bewegung des Krans genutzten, pneumatischen Zylinders dar. Variable $B5$ repräsentiert, ebenfalls binär, die erfolgreiche Aufnahme eines Werkstücks mittels Vakuumbreifer. Die Bestimmung der horizontalen Position des Krans erfolgt mittels Potentiometer. Dessen Information wird innerhalb des Modells als Winkelposition aufgefasst und durch die Variablen $B6$ – einen ganzzahligen Datenwert – dargestellt.

Die Funktionalität von Komponenten ist durch diskrete Operationen beschrieben, welche als mechatronische Funktionen der Komponente verstanden werden können. Dem Modellierungskonzept liegen dabei einige Annahmen hinsichtlich den Operationen zu Grunde. Operationen werden in UML-OP lediglich zeitdiskret (vor und nach) der Ausführung der Operationen betrachtet. Alle funktionalen Aspekte der Operationen werden auf Basis von Variablenwerten sowie deren diskrete Veränderungen (Vorbedingung und Effekt) ausgedrückt. Die Veränderung von Variablen während der Ausführung von Operationen wird nicht explizit berücksichtigt. Ferner wird angenommen, dass keine Abhängigkeiten zwischen Operationen existieren, die nicht durch Variable ausgedrückt werden können. Dürfen beispielsweise zwei Operationen nicht parallel ausgeführt werden, so muss dies durch die Einführung zusätzlicher Variablen (in diesem Fall einer Mutex Variable) gelöst werden. Analog gilt dies, falls explizit auf Veränderungen von Variablen während der Ausführung von Operationen Bezug genommen werden soll. Dies lässt sich ebenfalls über die geeignete Definition einer Mutex Variablen abbilden.

Die Funktionalitäten des Kranmoduls werden mittels Operationen modelliert. Die Manipulation der horizontalen Position erfolgt beispielsweise mittels der Operationen *dreheLinks()* bzw. *dreheRechts()*. Aus mechanischen Gründen kann der Kran keine vollständige, horizontale Umdrehung durchführen [VHLFF14]. Dieser Sachverhalt findet durch die Vorbedingung der beiden Operationen Berücksichtigung: eine Linksdrehung ist nur möglich, wenn die durch die Drehoperation erreichte Winkel noch den technischen Einschränkungen entspricht. Dies ist mittels der Vorbedingung unter Rückgriff auf die aktuelle Winkelposition (gegeben durch Variable $B6$) zuzüglich des mittels Drehoperationen manipulierten Winkel (hier durch den Platzhalter δ angegeben) berücksichtigt. Analog gilt dies für eine Rechtsdrehung (vgl. Abbildung 4.16).

4. Bestimmung der Ablaufreihenfolge von Automatisierungsfunktionen

Vorbedingungen werden in diesem Modellierungsansatz mittels einer impliziten Mengendefinition unter Rückgriff auf die (Zustands-)variablen der Komponente beschrieben. Die implizite Notation aller Zustände, in denen eine Operation ausgeführt werden kann, ist dabei wesentlich kompakter als eine explizite Aufzählung. Dies lässt sich leicht am Beispiel der Operation *ansaugen()* der Krankkomponente veranschaulichen. Die Operation *ansaugen* repräsentiert die Funktionalität des Krans, Werkstücke mittels eines Vakuumgreifers aufzunehmen. Da hier von einer Einzelkomponente ausgegangen wird, kann unter Berücksichtigung technischer Randbedingungen eine Werkstückaufnahme stets erfolgen, wenn die Krankkomponente nicht bereits ein Werkstück aufgenommen hat. Folglich ist die Ausführbarkeit der Operation *ansaugen()* ausschließlich von der Variablen *B5* abhängig. Eine explizite Mengendefinition der Vorbedingung würde in diesem Fall die Aufzählung von $2^2 \cdot 2^{16} = 2^{18}$ Wertekombinationen bedingen (es wird hier exemplarisch davon ausgegangen, dass *B6* eine 16-bit Integer ist).

Eine implizite Mengendefinition ermöglicht die Beschreibung aller zu einer Menge zugehörigen Elemente auf Basis der Eigenschaften der Elemente (d.h. Werte). Unter Rückgriff auf Vergleichsoperatoren, die für entsprechende Datentypen einer Variablen definiert sind, wird im Falle einer impliziten Mengendefinition der mögliche Wertebereich auf gültige Werte eingeschränkt. Die Einschränkung einer Variablen auf einen einzelnen, konkreten Wert ist hierbei ebenfalls möglich. Somit ist die Ausdrucksmächtigkeit der impliziten Mengendefinition äquivalent zur deren expliziten Definition, ist jedoch deutlich kompakter möglich.

Betrachtet man die zuvor beschriebenen Operation *ansaugen()*, ist deren Ausführbarkeit lediglich von dem Ladezustand gegeben durch Variable *B5* abhängig. Daher ist in diesem Fall die Angabe des für diese Variable gültigen Wertes hinreichend ($\neg B5$) und somit signifikant kompakter als eine explizite Mengendarstellung.

Neben der Einschränkung der Funktionsausführung durch Vorbedingungen werden Informationen über die Auswirkungen der Ausführung von Operationen auf das technische System angegeben. Dies stellt eine wesentliche Eingabeinformation für ein Planungsverfahren jeglicher Art dar. Wird beispielsweise ein pneumatischer Zylinder ausgefahren, werden Informationen über beobachtbare Auswirkung dieser Operationsausführung notwendig. In anderen Worten, wird eine Angabe des resultierenden Zustands des technischen Systems benötigt, welcher im Rahmen der UML-OP auf Basis von Variablen beschrieben ist. Die Beschreibung dieses resultierenden Zustands wird in UML-OP durch sogenannte Effekte realisiert. In der Softwareentwicklung werden hierzu Nachbedingungen (engl. *postcondition*) verwendet, um Zustände nach der Ausführung von Methoden oder Diensten zu beschreiben. Eine Nachbedingung stellt hierbei eine Zusage dar, welche die Einhaltung einer bestimmten Bedingung nach Ausführung entsprechender Programmteile fordert. Häufig werden hierzu logische Prädikate verwendet.

Betrachtet man beispielsweise den Effekt der Funktion *freigeben()* der Krankkomponente, welche ein loslösen des durch den Vakuumgreifer angesaugten Werkstücks zur Folge hat, weisen Sensor und Variable *B5* nach Ausführung der Operation stets den Wert *false* auf. Alle anderen Sensor- bzw. Variablenwerte bleiben von dieser Operationsausführung unberührt und somit konstant. Dieser Sachverhalt wird mittels des Effekts $B5 \rightarrow false$ modelliert. Für die Effektdefinition werden Pfeile als Zuweisungsoperator genutzt, um diese hier vereinfachend visuell von dem logischen Äquivalenzkomperator „ \rightarrow “ abzugrenzen, sind semantisch jedoch äquivalent. Die Zuweisung ausschließlich konstanter Werte in der Effektdefinition wird im Folgenden auch als konstanter Effekt bezeichnet.

Die ausschließliche Angabe von durch Operationsausführung resultierenden Zuständen mittels konstanter Effekte zeigt jedoch Grenzen. Dies lässt sich an der horizontalen Drehfunktionalität des Kranmoduls veranschaulichen. Der Kran wird durch `dreheLinks` bzw. `dreheRechts` um einen konstanten Winkel Δ gedreht und kann alle Winkelpositionen annehmen. Demzufolge existieren $\frac{360}{\Delta}$ mögliche resultierende Winkelpositionen, weshalb diese Operation mehrmals angeführt werden müssten, falls ausschließlich konstante Effekte beschrieben werden können.

Um dies zu umgehen, werden in der UML-OP neben konstanten Effekten auch sog. komplexe (abbildende) Effekte genutzt. Die Effekte ermöglichen einerseits zur Beschreibung resultierender Zustände einer Operationsausführung die Angabe von Berechnungsvorschriften, um resultierende Zustände, d.h. die erwartete Wertebelegung von Variablen, zu beschreiben. Andererseits kann auf den Wert der Variable vor Operationsausführung Bezug genommen werden, um zeitdiskretes Übergangsverhalten der Variablenwerte zu beschreiben. Dies wird mittels des OCL Operators `@pre` ermöglicht und beispielsweise für die bereits eingeführten horizontalen Drehoperationen des Krans genutzt. Hierdurch ist die konsolidierte Angabe des diskreten Verhaltens von Operationen möglich.

Zusammenfassend ermöglicht die vorgeschlagene Notation eine kompakte Modellierung der Vorbedingungen und Effekte von Operationen einzelner Komponenten.

4.2.1.2. Das Systemmodell der UML-OP

Die Modellierung des Gesamtsystems erfolgt in UML-OP ebenfalls durch eine Klasse, deren *Variablen* durch *Objekte* der zuvor modellierten Komponenten des Systems gebildet werden. Bei der Modellierung des Gesamtsystems ergeben sich weitere Einschränkungen an die Ausführbarkeit von Operationen, z.B. durch die Zusammenstellung der Anlagenkomponenten. Diese werden als ergänzende Vorbedingungen und Effekten der Operationen im Systemmodell angegeben. Abbildung 4.17 zeigt einen Auszug des Systemmodells der Stempelanlage. Im Rahmen der Komponentenmodellierung werden lediglich Einschränkungen der Funktionalität hinsichtlich der Eigenschaften der Komponente selbst berücksichtigt wurden (z.B. konstruktive Einschränkung der Drehoperationen des Krans). Aus dem Einsatz der Komponente innerhalb eines konkreten Systems können sich weitere Einschränkungen ergeben. So ist es beispielsweise notwendig, dass ein mit Material beladener Kran keine rotatorische Bewegung durchführen sollte, falls er abgeseht ist. Grund hierfür ist die mechanische Konstruktion der Übergabepositionen zwischen den Komponenten. Diese Einschränkungen können im Systemmodell als zusätzliche Vorbedingungen angegeben werden. Analog können für Operationen der Komponenten ebenfalls zusätzliche Effekte angegeben werden, die sich aus der Integration der Komponente im Systemkontext ergeben.

Die Stempelanlage ist dabei hinsichtlich des Layouts wie folgt konfiguriert: Befindet sich der Kran an der Übergabeposition des Stempels, weist der die Positionsvariable B_6 des Krans den Wert 270 auf; an der Position der Rampe den Wert 180 und an der Position des Lagers den Wert 0. Wie in der Tabelle von Abbildung 4.17 angegeben, gilt für die Drehoperationen der Krankomponente – d.h. `dreheLinks()` und `dreheRechts()` – dass sich der Kran an den Positionen der weiteren Komponenten des Systems (Warenlager, Stempelmodul und Rampe) nur in angehobenem Zustand (gegeben durch die Variablen B_3 und B_4) drehen darf. In diesem Fall ist die weitere Einschränkung der Funktionalität der Krankomponente ausschließlich von den Sensoren bzw. Variablen des Krans abhängig. Eine Abhängigkeit zwischen Variablen unterschiedlicher Kom-

4. Bestimmung der Ablaufreihenfolge von Automatisierungsfunktionen

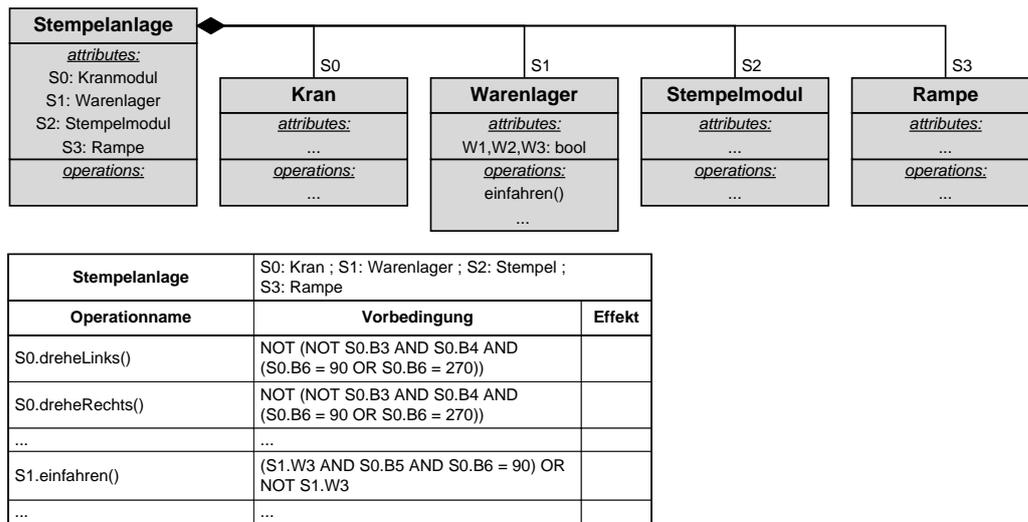


Abbildung 4.17.: Ausschnitt des UML-OP Modells der Stempelanlage (in Anlehnung an [LSF⁺13, LSVH14]).

ponenten ergibt sich beispielsweise durch den Übergabemechanismus der Werkstücke zwischen Warenlager und Kran. Konstruktionsbedingt ist es notwendig, dass der Zylinder des Warenlagers nach Vereinzelung eines Werkstücks ausgefahren bleibt. Grund hierfür ist, dass durch den Widerstand des Mikroschalters (Variable $W3$) die Werkstückposition bei Einfahren des Zylinders so verändert werden würde, dass eine Werkstückaufnahme durch den Kran nicht mehr möglich ist. Ein erneutes Fixieren durch den Zylinder ist ferner nach Einfahren nur noch möglich, wenn zuvor das letzte Werkstück vereinzelte wurde, welches sich im Lager befand. Andernfalls befindet nach Einfahren des Zylinders ein weiteres Werkstück an der Position vor dem Zylinderkopf. Dieser Sachverhalt kann durch Einschränkung der Funktionalität mittels Vorbedingung der Operation *einfahren()* des Warenlagers berücksichtigt werden. Diese Operation darf lediglich ausgeführt werden, wenn entweder ein Werkstück fixiert wird (d.h. Variable $W3$ des Warenlagers den Wert *true* liefert) und durch den Kran bereits erfolgreich angesaugt wurde (d.h. Variable $B5$ des Krans den Wert *true* liefert) oder kein Werkstück durch das Warenlager vereinzelte wurde (d.h. Variable $W3$ des Warenlagers den Wert *false* liefert). Letzteres ist genau dann der Fall, wenn sich kein Werkstück mehr im Warenlager befindet.

Zusammenfassend ermöglicht die vorgestellte Notation mittels des Systemmodells die Modellierung der Zusammenstellung einzelner Komponenten und der daraus für die Funktionalitäten der Komponenten ergebenden Einschränkungen.

4.2.1.3. Das Produktmodell der UML-OP

Um Anforderungen hinsichtlich der Bearbeitung von Werkstücken (d.h. dem Fertigungsprozess) darstellen zu können, wird eine Klasse zur Beschreibung von Produkteigenschaften verwendet. Die Modellierung der Vorbedingungen und Effekte erfolgt hierbei analog zur Komponenten- und Systemmodellierung. Zusätzlich werden der Initialzustand (Initial) sowie der Zielzustand (Goal) eines Produktes beschrieben. Dieser spiegelt die Zielsetzung der UML-OP wieder, den Zustands-

raum eines Systems kompakt und modellgetrieben zu Modellieren, da Start- und Zielzustand als Minimalanforderungen an den Zustandsraum des Werkstücks/Produktes gestellt werden.

Ein Auszug einer beispielhaften Modellierung von in der Stempelanlage gefertigten Werkstücken ist in Abbildung 4.18 dargestellt. Im vorliegenden Fall sollen gestempelte Werkstücke nach Vereinzeln im Warenlager an die Rampe übergeben werden. Den Einfluss auf die charakteristischen Merkmale des Produktes, ist durch die Variablen des Produktes gegeben. Einerseits wird die boolesche Variable *Bearbeitet* genutzt, welche eine erfolgreiche Durchführung der Operation *stempeln()* des Stempelmoduls anzeigt (vgl. gegebenen Effekt der Operation in Abbildung 4.18). Eine weitere Variable des Produktes gibt den *Ort* eines Produktes an, an dem sich ein Produkt jeweils befindet. Hierfür kann der Datentyp ENUM der UML genutzt werden. Neben Werten für die einzelnen Komponenten der Stempelanlage enthält die durch die Variable *Ort* gegebene Aufzählung zusätzliche Werte für den initialen Ort des Produktes (innerhalb des Magazins des Warenlagers) sowie dem Zielort, welcher eine beliebige Endposition des Werkstücks auf der Rampe repräsentiert. Wie anhand des in der Abbildung gegebenen Beispiels zu erkennen ist, wird beispielsweise der Wert der Variablen *Ort* durch die Operation *ausfahren()* des Warenlagers derart manipuliert, dass nach Operationsausführung sich das Produkt nicht mehr am initialen Ort, sondern innerhalb des Warenlagers befindet. Analog werde durch andere Operationen die Variablen des Produktes beeinflusst.

Produkt			
<i>attributes:</i> Ort: ENUM Bearbeitet: boolean			Initial: Ort = INITIAL AND NOT Bearbeitet Goal: Ort = ZIEL AND Bearbeitet
<i>operations:</i>	Operationsname	Vorbedingung	Effekt
	S1.ausfahren()	P1.Ort = INITIAL	P1.Ort → Warenlager
	S2.stempeln()	NOT P1.Bearbeitet	P1.Bearbeitet → true

Abbildung 4.18.: Ausschnitt eines beispielhaften Produktmodells für die Stempelanlage (in Anlehnung an [LSF⁺13, LSVH14]).

4.2.2. Gegenüberstellung der UML-OP und des HiTraP-AT Planungsproblems

Um eine Integration des HiTraP-AT Planungsverfahrens in einen modellgetriebenen Entwicklungsansatz für den Maschinen- und Anlagenbau und damit letztendlich eine automatische Ableitung von Handlungsplänen auf Basis von im Rahmen der modellgetriebenen Software- bzw. Systementwicklung genutzter Modelle zu erreichen, ist es notwendig, dass alle für die Durchführung der Planung notwendigen Informationen verfügbar sind. Dementsprechend wird im Folgenden die zuvor eingeführte UML-OP dem Eingabemodell des HiTraP-AT Planungsverfahrens – dem HiTraP-AT Planungsproblem (vgl. Abschnitt 4.1.2) – gegenübergestellt und eine entsprechende Abbildung zwischen beiden identifiziert.

Die Beschreibung der strukturellen Zusammenhänge einzelner Teilelemente innerhalb eines Gesamtsystem wird in UML-OP mittels des Komponenten- und Systemmodells realisiert. Analog wird ein Planungsproblem nach HiTraP-AT auf Basis eines AT-Systems beschrieben, welches aus einzelnen Modulen besteht. Ferner wird innerhalb der UML-OP ein zu fertigendes Produkt

4. Bestimmung der Ablaufreihenfolge von Automatisierungsfunktionen

auf Basis des Produktmodells beschrieben. Analog zur Modellierung des Systems und Moduls wird innerhalb des Produktmodells der UML-OP die Charakterisierung des Produktes auf Basis von Attributen einer Klasse vorgenommen und ist damit auf die Problembeschreibung nach HiTraP-AT, in der ein Produkt auf Basis von Variablen beschrieben ist, direkt abbildbar. Dementsprechend kann zusammengefasst werden, dass für die Kernelemente eines HiTraP-AT Modells (Produkt, Modul und AT-System) entsprechende Konzepte innerhalb der UML-OP Modelle existieren und entsprechend das Fundament für eine Abbildung zwischen UML-OP und HiTraP-AT Problemmodell existiert.

HiTraP-AT sieht im Gegensatz zur UML-OP keine Trennung von Funktionalität der Module und des Gesamtsystems vor. Die Aktionen des HiTraP-AT Modells entsprechen dabei jedoch in UML-OP Semantik der Menge aller *Operationen* der Komponenten eines Systems unter Berücksichtigung der im Systemmodell definierten, zusätzlichen Einschränkungen. Entsprechend gilt für die Ausführbarkeit einer HiTraP-AT Aktion, dass sowohl die im Systemmodell der UML-OP angegebenen Vorbedingungen wie auch die Vorbedingung der Komponentenoperation erfüllt sein müssen – beide Vorbedingungen also als logische Konjunktion betrachtet werden müssen. Ist keine Vorbedingung für eine Komponentenoperation angegeben, ist eine Operation unabhängig einer Variablenbelegung – d.h. eines Zustandes – ausführbar. Daher kann eine leere Vorbedingung als stets wahre Bedingung verstanden werden und stellvertretend der konstante Wert *true* eingesetzt werden. Analog gilt dies, falls keine einschränkende Vorbedingung im Systemmodell für eine Operation angegeben ist. Aus diesem Grund kann stets die logische Konjunktion Anwendung finden, ist jedoch für eine stets ausführbare Operation in trivialer Weise durch "true AND true" gegeben.

Die Effektdefinitionen von Operationen des UML-OP Komponentenmodells unterscheiden sich von der Effektbeschreibung der Aktionen des HiTraP-AT Modells signifikant: Während innerhalb des HiTraP-AT Modells ausschließlich von Nachbedingungen ausgegangen wird (d.h. eine durch logische Bedingung nach Aktionsausführung entsprechend gültige Belegung der Zustandsvariablen), werden Effekte verallgemeinert als Abbildungsvorschrift zwischen Zuständen aufgefasst. Diese beiden Semantiken lassen sich jedoch vollständig aufeinander abbilden. Effekte der UML-OP, welche lediglich einen oder mehreren Variablen konstante Werte zuweisen, können direkt als logische Bedingung aufgefasst und formuliert werden. Betrachtet man beispielhaft die Operation *loslassen()*, welche die Freigabe von Werkstücken durch das Kranmodul repräsentiert (vgl. Abschnitt 4.2.1.1), gilt für die Effektdefinition der Operation stets $B5 \rightarrow false$. Dies kann direkt in eine mathematisch logische Aussage der Art $NOT B5$ (alternativ auch $B5 = false$) transformiert werden. Effekte, deren Definition Rückgriff auf Werte vor Operationsausführung (durch den Operator *@pre*) nehmen, müssen gesondert behandelt werden, um eine Eindeutigkeit zwischen Vorbedingung und Effekt (wie in HiTraP-AT Planungsproblem gefordert) zu erreichen. Um dies zu erreichen, wird jede einzelne solche Operation mit (nicht eindeutig) abbildender Effektdefinition durch eine Menge von (künstlich erzeugten) Operationen mit konstantem Effekt ersetzt⁵¹. Dabei wird für jede mögliche Wertebelegung der durch den Operator *@pre* angesprochenen Variablen einer UML-OP Operation eine neue Operation erstellt, welche diese Wertebelegung als zusätzlich

⁵¹ Würde die UML-OP Semantik der Effektdefinition direkt innerhalb des HiTraP-AT Planungsproblems aufgegriffen, könnte diese Auffaltung auf Ebene der Aktionen entfallen. Dabei würde zwar die Anzahl der Aktionen beeinflusst, da keine zusätzlichen Aktionen für diese Art Effekte künstlich erzeugt werden müssten, die Größe des HiTraP-AT Modells selbst bleibt dabei jedoch unverändert.

konjugierte Bedingung innerhalb der Vorbedingung besitzt und als Nachbedingung, die sich daraus ergebende Wertebelegung in Form eines konstanten Effekts enthält. So wird beispielsweise die Operation *dreheLinks()* des Krans in $\lceil \frac{360}{\Delta} \rceil$ Operationen überführt. Eine dieser Operationen ist dabei beispielsweise ausführbar, falls die Variable *B6* des Krans den Wert Null aufweist. Die innerhalb des UML-OP Modells gegebene Vorbedingung der Operation mit *@pre* Operator in der Effektdefinition wird entsprechend der zuvor beschriebenen Transformation derart ergänzt, so dass die Vorbedingung der neu erzeugten Operation gegeben ist durch $(B6 + \Delta < 360) \text{ AND } (B6 = 0)$. Für die entsprechende Nachbedingung, in der durch die Vorbedingung zugesicherte Wert der durch den Operator *@pre* referenzierten Variablen *B6* eingesetzt wurde, ergibt sich in diesem Fall $B6 \rightarrow \Delta + 0$ (dieser konstante Effekt kann, wie oben beschrieben, transformiert werden). Analog wird für alle andere Operationen verfahren. Hierdurch kann die geforderte Eindeutigkeit hinsichtlich der Abbildung zwischen Vor- und Nachfolgezuständen erreicht werden und folglich gegebene Effektdefinitionen von Operationen innerhalb eines UML-OP Modells geeignet in die Beschreibung von Effekten entsprechend dem HiTraP-AT Planungsproblems erfolgen.

Für Effektdefinitionen von Operationen innerhalb eines UML-OP Systemmodells gilt ein zu den Vorbedingungen von Operationen analoges Vorgehen. Dabei wird für Effektdefinitionen des Systemmodells, welche keine konstanten Abbildungsvorschriften darstellen, die zuvor beschriebene Transformation durchgeführt. Wie ebenfalls bereits zuvor erläutert, können Effektdefinitionen, welche einer oder mehreren Variablen konstante Werte zuweisen, direkt in logische Bedingungen überführt werden, welche die Effektdefinition des HiTraP-AT Modells darstellen. Dementsprechend kann der Bezug zwischen gegebenen Effektdefinitionen des System- und Komponentenmodells analog zu Vorbedingungen, als logische Konjunktion der Effektdefinitionen innerhalb eines HiTraP-AT Problemmodells verstanden werden.

Operationen der UML-OP unterliegen stets der Annahme, dass deren Ausführbarkeit, analog zu HiTraP-AT, vollständig auf Basis von Variablen spezifiziert wird⁵². Entsprechend sind funktionale Einschränkungen des Systemmodells als vollständig zu betrachten. Ist für die Ausführung beispielsweise die Anwesenheit eines Werkstücks an einer spezifischen Stelle des Moduls notwendig, wird diese Information durch geeignete Sensoren geliefert. Andernfalls wäre die Implementierung einer geeigneten Steuerungssoftware nicht möglich. Aus diesen Gründen können die zuvor beschriebenen Aktionen, welche aus Komponenten- und Systemmodell der UML-OP abgeleitet sind, als nicht-produktive Aktionen im Sinne des HiTraP-AT Planungsproblems verstanden werden. Bisher nicht adressiert wurde dabei einerseits der Einfluss der Ausführung von Aktionen auf das Produkt sowie die durch das HiTraP-AT Planungsproblem geforderte Klassifikation von Aktionen (alle Typen produktiver Aktionen). Im Folgenden werden die noch verbleibenden Aspekte innerhalb der Abbildung eines UML-OP Modells auf ein HiTraP-AT Planungsproblem erläutert.

Neben den Spezifikationen der Vorbedingungen und Effekte in Komponenten- und Systemmodell existieren in der UML-OP ferner analoge Spezifikationen im Produktmodell der UML-OP. Diese beschreiben notwendige Abhängigkeiten der Ausführbarkeit von Operationen unter Berücksichtigung der charakteristischen Merkmale (d.h. Variablen) des Produktes ebenso wie die

⁵²Installierte Sensorik innerhalb eines Automatisierungssystems ist dabei insofern vollständig, dass die Anwesenheit von Werkstücken durch die Steuerungssoftware (soweit notwendig) identifiziert werden kann, z.B. durch Mikroschalter oder Lichtschranken. Dies ist insbesondere in den in dieser Arbeit betrachteten, modularen Fertigungsautomatisierungssystemen gegeben.

4. Bestimmung der Ablaufreihenfolge von Automatisierungsfunktionen

Effekte auf diese Merkmale durch die Ausführung der Operation. Diese Informationen sind innerhalb des HiTraP-AT Modells für den Handlungsraum "Produktion" essentiell notwendig. Für jede innerhalb eines Produktmodells der UML-OP gegebene Operation wird (zusätzlich zu der die Operation repräsentierenden, nicht-produktiven Aktion, welche auf Basis des Komponenten- und Systemmodells wie zuvor beschrieben erzeugt wurde) eine als produktiv zu klassifizierende Aktion innerhalb eines HiTraP-AT Planungsproblems erzeugt. Die Vorbedingung dieser Aktionen wird dabei durch die logische Konjunktion aus den innerhalb des Komponenten- und Systemmodells gegebenen Vorbedingungen (analog zu entsprechenden nicht-produktiven Aktionen) sowie der innerhalb des Produktmodells für diese Operation gegebenen Vorbedingung abgeleitet. Sind innerhalb des Produktmodells Effektdefinitionen gegeben, welche nicht ausschließlich eine konstante Wertzuweisung realisieren, sondern, wie in UML-OP spezifisch möglich, eine komplexe Abbildungsvorschrift unter Rückgriff auf Variablenwerte vor Operationsausführung (durch den Operator *@pre*) enthalten, werden diese analog wie zuvor für System- und Komponenteneffekte beschrieben, in eine entsprechende Menge von Operationen mit konstanten Effekten transformiert. Konstante Effekte können, wie oben erläutert, direkt als logische Bedingungen im Sinne des HiTraP-AT Planungsproblems transformiert werden. Entsprechend werden die (ggf. durch Transformation zur Behandlung nicht-konstanter Effekte resultierende zusätzliche Operationen) innerhalb eines Produktmodells gegebenen Effektdefinitionen in eine logische Bedingung entsprechend des HiTraP-AT Planungsproblems transformiert und mit den sich aus System- und Komponentenmodell ergebenden, logischen Bedingungen konjugiert. Hierdurch ist nun die Menge aller produktiven Aktionen eines HiTraP-AT Planungsproblems vollständig auf Basis der in UML-OP gegebenen Informationen abgeleitet.

Die Startzustände von Produkt und AT-System, wie dies durch das HiTraP-AT Planungsproblem gefordert ist, finden sich an unterschiedlichen Stellen des UML-OP Modells. Die Startzustände von Komponenten sind im Systemmodell der UML-OP gegeben, da einem System Instanzen der als Klasse definierten Komponenten zugeordnet werden. Die Menge aller Startzustände der Komponenten eines in UML-OP modellierten Systems ergibt dabei die Startzustände des AT-Systems innerhalb des HiTraP-AT Planungsproblems. Die initialen Werte der Variablen zu fertigender Werkstücke sind im Rahmen des Produktmodells angegeben.

Informationen darüber, welche dieser Operationen als Initialisierungs-, Finalisierungs- und Übergabeaktion entsprechend der durch das HiTraP-AT Planungsproblem gegebenen Taxonomie klassifiziert werden kann, ist innerhalb der UML-OP nicht explizit vorhanden. Eine automatische Transformation zwischen UML-OP Modellen und HiTraP-AT ist jedoch durch eine kleine Konvention für die Verwendung der UML-OP möglich. Gefordert wird in diesem Zusammenhang, dass stets eine Variable *Ort* (wie dies bereits in dem in Abschnitt 4.2.1.3 gegebenen Beispiel erfolgt ist) innerhalb eines Produktmodells gegeben ist. Die Definitionsmenge dieser Variablen *Ort* ist eine Aufzählung möglicher Werte. Diese ergeben sich dabei einerseits aus den Namen der innerhalb des Systemmodells angegebenen Komponenten sowie zwei zusätzlichen (ggf. nicht real existierenden) Orten, welche einen Start- und Zielort repräsentieren. Wechselt ein Modul nach Operationsausführung das Modul, muss innerhalb des Produktmodells dies auf Basis der Effektdefinition und der Variable *Ort* beschrieben sein. Ferner gibt es in einem (oder ggf. betrachteten Ausschnitt eines) Fertigungsautomatisierungssystem stets Operationen, welche allgemein betrachtet die Bearbeitung eines Produktes initialisieren oder finalisieren. So existieren beispielsweise in der zuvor beschriebenen Stempelanlage die Operation *ausfahren()* des Waren-

lagers, welches gelagerte Werkstücke vereinzelt, um nachgelagerten Module eine Bearbeitung des Werkstücks zu ermöglichen. Solche Operationen manipulieren zusätzlich die Variable *Ort*. Im Falle einer Initialisierung ist in der Vorbedingung gefordert, dass der *Ort* dem gegebenen, initialen Ort entspricht. Die Effektdefinition bestimmt dabei, dass nach Ausführung dieser Operation der *Ort* des Produktes dem der Komponente entspricht, welche die Operation ausgeführt hat (vgl. die Beschreibung der Operation *ausfahren()* in Abbildung 4.18). Analog hierzu gilt für Operationen mit Finalisierungsfunktion, dass ein zusätzlicher Effekt bzgl. der Variablen *Ort* existiert, welche den Wert dieser Variablen auf den Wert des Zielortes setzt. Enthält eine innerhalb eines Produktmodells gegebene Operation eines Moduls eine Effektdefinition, welche der Variablen *Ort* einen Wert zuweist, welcher nicht dieses Modul repräsentiert, stellt diese Operation eine Übergabeaktion im Sinne des HiTraP-AT Planungsproblems dar. Ist innerhalb der Vorbedingung der initiale Ort für eine Operation gefordert, stellt die entsprechende Aktion eine Initialisierungsaktion dar, welche keine Übergabeaktion ist. Wird innerhalb einer Effektdefinition eine Zuweisung der Variablen *Ort* auf den Zielort angegeben, werden Aktionen, welche diese Operation repräsentieren, nicht als Übergabe- sondern als Finalisierungsaktion aufgefasst. Die Konvention der Einführung einer spezifischen (zusätzlichen) Variablen innerhalb des Produktmodells für UML-OP schränkt weder den Anwendungsbereich der UML-OP noch die Allgemeingültigkeit der hier beschriebenen Abbildung zwischen UML-OP und HiTraP-AT ein. Wird diese vorgeschlagene Konvention nicht eingehalten, ist lediglich während der Transformation die manuelle Klassifikation der sich aus einem gegebenen UML-OP Modell resultierenden, produktiven Aktionen als Initialisierungs-, Finalisierungs- und Übergabeaktionen notwendig.

Zusammenfassend konnte in diesem Abschnitt gezeigt werden, dass UML-OP Modelle in das HiTraP-AT Planungsproblem transformiert werden können und entsprechend eine geeignete Abbildung existiert. Somit existiert folglich zum einen mit UML-OP eine Modellierungsnotation, welche eine konsolidierte Beschreibung des Handlungsraumes für die Steuerungsebene des Maschinen- und Anlagenbaus erlaubt. Zum Anderen erlaubt dies eine nahtlose Integration des in dieser Arbeit vorgeschlagenen HiTraP-AT Planungsverfahrens in einen modellgetriebenen Entwicklungsansatz des Maschinen- und Anlagenbaus.

4.3. Anwendung von HiTraP-AT für IEC 61131-3 basierte Steuerungssoftware (Anforderung A4)

Eine der wesentlichen Anforderungen an das HiTraP-AT Planungswerkzeug stellt dessen Anwendbarkeit für IEC 61131-3 basierte Steuerungssoftware dar (vgl. Anforderung A4). Im Verlaufe dieses Abschnitts wird auf Basis zuvor beschriebener Konzepte die entsprechende Anwendbarkeit des HiTraP-AT Planungsverfahrens dargelegt. Ein Planungswerkzeug wird dabei entsprechend Anforderung A4 als anwendbar für IEC 61131-3 basierte Steuerungen betrachtet, wenn gezeigt werden kann, dass die durch den Handlungsplan definierten Informationen (Reihenfolgen und genutzte Funktionalitäten), für die Generierung von Programmcode für Speicherprogrammierbare Steuerungen genutzt werden können.

Entsprechend den beiden in Abschnitt 1.1 beschriebenen Anwendungsfällen ergeben sich jeweils unterschiedliche Anforderungen im Hinblick auf eine Anwendung des HiTraP-AT Verfahrens für IEC 61131-3 basierte Steuerungssoftware (vgl. Anforderung A4): Während in Anwendungsfall 1 (Konfiguration der Anlage) die Annahme getroffen wird, dass der Steuerungscode während

4. Bestimmung der Ablaufreihenfolge von Automatisierungsfunktionen

des Betriebs der Maschine bzw. Anlage unverändert bleibt⁵³ (d.h. im Kontext von HiTraP-AT bleiben durch die Steuerungssoftware realisierte Handlungssequenzen statisch) wird für die Realisierung von Anwendungsfall 2 (dynamische Rekonfiguration) eine Steuerungssoftware benötigt, deren Verhalten geeignet während des Betriebes verändert werden kann (d.h. Handlungssequenzen konfiguriert werden können ohne den Steuerungscode selbst zu verändern). Dementsprechend werden im Folgenden zwei mögliche Ansätze zur Anwendung des HiTraP-AT Verfahrens beschrieben: Die Realisierung und Generierung von Steuerungscode für einen a priori gegebenen Handlungsplan wird in Abschnitt 4.3.1 beschrieben, wohingegen die Anforderung nach konfigurierbarer Steuerungssoftware in Abschnitt 4.3.2 beschrieben wird.

4.3.1. Realisierung von Steuerungscode mit statischen Handlungsplänen

In gemeinsamen Vorarbeiten mit Daniel Schütz [LSVH13, LSF⁺13, LSVH14] konnte bereits gezeigt werden, dass durch HiTraP-AT abgeleitete Handlungssequenzen mittels eines Aktivitätsdiagramms der UML dargestellt werden können. Durch HiTraP-AT selektierte Aktionen (welche, wie in Abschnitt 5.1 bereits erläutert, Operationen des UML-OP Modells repräsentieren) werden dabei durch Aktionen im Aktivitätsdiagramm dargestellt. Die durch HiTraP-AT bestimmte Reihenfolge ist im Aktivitätsdiagramm durch Objektflüsse repräsentiert. Um die Komplexität der generierten Handlungspläne zu reduzieren, wurden dabei zum einen mehrfach aufeinander folgende Aufrufe derselben Operation zu einer einzelnen Operation im Aktivitätsdiagramm zusammengefasst und zum anderen eine spezifische Sequenz von Aktionen der durch einen HiTraP-AT Handlungsplan beschriebenen Menge (total geordneter) Handlungspläne ausgewählt. Vor- und Nachbedingungen, die durch die Zusammenführung des Komponenten-, System- und Produktmodells bestimmt wurden, werden dem Anwender durch Kommentare im Aktivitätsdiagramm visualisiert.

Die Generierung von Steuerungscode nach IEC 61131-3 auf Basis von UML-Aktivitätsdiagrammen wurde bereits in einigen früheren Forschungsarbeiten, z.B. [Pö11, SW09, WSVH08], beschrieben. Da nähere Details für das weitere Verständnis nicht notwendig sind, wird an dieser Stelle auf die entsprechenden Arbeiten verwiesen. Hervorzuheben sind in diesem Zusammenhang allerdings die Dissertationen von Witsch [Wit12] und Schütz [Sch14]. Daniel Witsch beschreibt im Rahmen seiner Dissertation [Wit12] einen modellgetriebenen Ansatz für die Entwicklung von Steuerungssoftware im Maschinen- und Anlagenbau, welcher zur Generierung von IEC 61131-3 Software genutzt werden kann. In dem durch Witsch vorgeschlagenen plcML Profil wird für die Metaklasse „Operation“ der UML eine Transformation in Methoden der IEC 61131 definiert. Anzumerken ist in diesem Zusammenhang, dass das Aktivitätsdiagramm ausschließlich Informationen über die (partielle) Ordnung von Aktionen beinhaltet und folglich auf dessen Basis IEC 61131-3 konformer Steuerungscode generiert werden kann, welcher die Reihenfolge der Aktionen realisiert. Im Rahmen eigener Vorarbeiten [BLVH15] und der studentischen Arbeit von Ulrich Bühner [Bü14] konnte positiv evaluiert werden, dass die UML-OP eine automatische Erzeugen von Prozessabläufen für Speicherprogrammierbare Steuerungen ermöglicht.

Für die Generierung des Programmcodes einzelner Aktionen werden weitere Informationen benötigt, welche in der UML-OP nicht enthalten sind. Im Rahmen der Dissertation von Da-

⁵³Mit unverändert ist in diesem Zusammenhang die Tatsache referenziert, dass eine automatische Anpassung der Reihenfolge von Softwarefunktionen auf einer Speicherprogrammierbaren Steuerung ohne manuelle Intervention nicht gefordert ist.

niel Schütz [Sch14] wurde ein modellgetriebener Ansatz für die automatische Generierung von Softwareagenten und deren Wissensbasen vorgeschlagen. Als Ausgangspunkt für die Generierung von IEC 61131-3 konformen Programmcode für die Steuerungsebene des Maschinen- und Anlagenbaus dient hierbei das von Daniel Schütz entwickelte SysML-AT Profil genutzt. Dieses ermöglicht eine automatische Generierung der Implementierung nach IEC 61131-3 von Multi-Agenten Systemen [FPS11] und erweitert dabei das von Daniel Witsch bereits erarbeitete Konzept. Insbesondere wurde im Kontext der Arbeit von Daniel Schütz das Parameterdiagramm der SysML erweitert, um eine vollständige Generierung des Steuerungscode zu ermöglichen. Das Entwicklungskonzept der SysML-AT wurde in eine industrielle Entwicklungs- und Laufzeitumgebung integriert, was eine modellbasierte Entwicklung von Softwareagenten, der Generierung ausführbaren Programmcodes sowie ein integriertes Debugging innerhalb der Modelle ermöglicht [SLVH14]. Im Rahmen der Dissertation von Daniel Schütz wurde ferner gezeigt, dass sich die UML-OP (sowie dessen Anwendung für eine etwaige automatische Handlungsplanung) mit marginalen Anpassungen nahtlos in das Konzept des SysML-AT integrieren lässt.

Es lässt sich zusammenfassen, dass durch Abbildung von HiTraP-AT Handlungsplänen auf das UML-Aktivitätsdiagramm eine Generierung von IEC 61131-3 möglich ist, sich dabei jedoch auf die diagrammspezifischen Informationen und Umfang beschränkt. Die Integration der UML-OP in die durch Daniel Schütz entwickelte SysML-AT, welche die vollständige Generierung von entsprechendem IEC 61131-3 Steuerungscode ermöglicht sowie die Abbildungen zwischen HiTraP-AT Planungsproblem und UML-OP (vgl. Abschnitt 4.2 bzw. von HiTraP-AT Handlungsplänen und UML-Aktivitätsdiagrammen ermöglichen direkte Anwendung des HiTraP-AT Planungsverfahrens für IEC 61131-3 basierte Speicherprogrammierbare Steuerungen im Kontext einer modellbasierten Entwicklungsprozesses für Maschinen und Anlagen (Anforderung A4). Dabei handelt es sich bei dem durch dieses Vorgehen generierten Programmcode für die Steuerungsebene um Softwareprogramme deren Ablauf, d.h. die durch ein Planungsverfahren bestimmte Ablauf der Aktionen, für ein gegebenes Modell stets identisch ist. Dieser Ansatz der Generierung von Programmcode ist daher lediglich für die modellgetriebene Softwareentwicklung, wie sie in Anwendungsbeschreibung 1 (vgl. Abschnitt 1.1.1) zu Grunde gelegt ist, geeignet.

4.3.2. Realisierung von Steuerungscode für konfigurierbare Handlungssequenzen

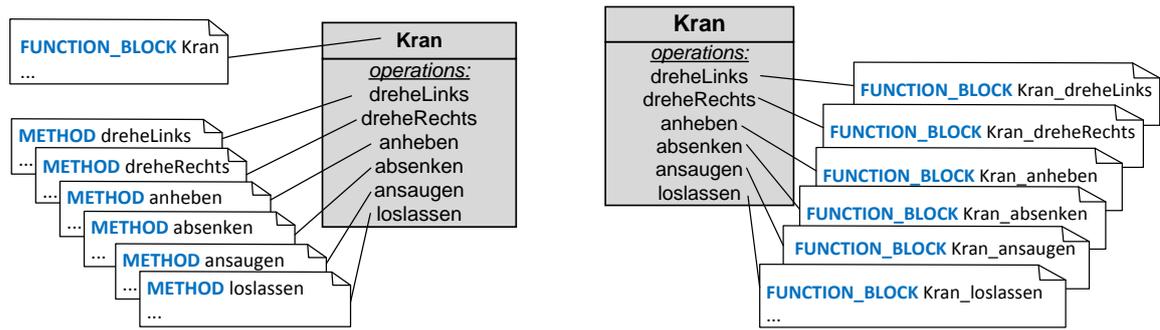
Im vorangegangenen Abschnitt wurde beschrieben, dass sich mittels durch HiTraP-AT identifizierten Handlungsplänen und deren Darstellung als UML-Aktivitätsdiagramm, Programmcode für IEC 61131-3 basierte Speicherprogrammierbare Steuerungen erzeugen lässt. Hierdurch wird jedoch exakt ein konkreter Handlungsplan realisiert. Um das Verhalten eines Fertigungssystems jedoch anzupassen, wie dies beispielsweise für die Realisierung des Anwendungsfalls 2 (siehe Abschnitt 1.1.2) notwendig ist, ist ein solches Verfahren nur bedingt geeignet. Grund hierfür ist, dass Steuerungssysteme bei Änderung von Softwarecode zumeist einen Neustart erfordern und dadurch für das Automatisierungssystem kurzzeitig keine Steuerung verfügbar ist. Dies kann letztendlich dazu führen, dass ggf. notwendige Variablenbelegungen von Aktoren nicht gesetzt werden können. Daher ist es für die durch Anwendungsfall 2 repräsentierten Ziele notwendig, parametrisierbaren Programmcode zu generieren (siehe hierzu auch die Diskussion in Abschnitt 1.1.2.2). Aus diesem Grund wurde in eigenen Vorarbeiten [LVH13, LVH14a] sowie [LVH14b, LVH15], aber auch in gemeinsamen Vorarbeiten mit Ulrich Bühner [BLVH15] und dessen studentischen Arbeit

4. Bestimmung der Ablaufreihenfolge von Automatisierungsfunktionen

[Bü14] untersucht, ob das HiTraP-AT Planungsverfahren zur Generierung von IEC 61131-3 konformen Programmcode genutzt werden kann, welcher eine dynamische Rekonfiguration – d.h. eine dynamische Anpassung und letztendlich konfigurierbare Handlungssequenzen zur Laufzeit – ermöglicht.

Um eine Anpassung des Steuerungsverhaltens entsprechend vorgegebener Handlungsplänen zur Laufzeit einer Maschine oder Anlage zu ermöglichen, wurde in den Vorarbeiten [LVH14b, LVH15] ein Konzept entwickelt um – analog dem Konzept dienstorientierter Architekturen – eine Trennung der implementierten Funktionalität und deren möglicher Abläufe – auf IEC 61131-3 basierten Speicherprogrammierbaren Steuerungen zu ermöglichen. Ein durch HiTraP-AT identifizierter Handlungsplan stellt dabei einen spezifischen Ablauf von in der Steuerung implementierten Funktionalitäten analog den in UML-OP modellierbaren Operationen (vgl. Abschnitt 4.2.1) dar. Im Kontext dienstorientierter Architekturen wird dies auch als Workflow Model bezeichnet und definiert die Komposition im Sinne eines Ablaufs existierender – d.h. im (Steuerungs-)System implementierter – Funktionalitäten. Dieses Workflow Model wird dabei als explizites Datenmodell innerhalb des IEC 61131-3 Steuerungsprogramms repräsentiert. Im Gegensatz zu einer Implementierung, bei der Ablaufsteuerung und Funktionalität integriert implementiert werden (wie es beispielsweise bei der Generierung von Steuerungscode entsprechend dem im vorangegangenen Abschnitt beschriebenen Verfahren der Fall ist), kann durch die Trennung einzelner Funktionalitäten und deren Ablauf unter anderem durch ein externes Programm Einfluss auf den Programmablauf genommen werden. Ein innerhalb der Speicherprogrammierbaren Steuerung implementiertes Softwaresystem interpretiert dabei das Workflow Model und führt die darin angegebenen Funktionalitäten in der entsprechenden Reihenfolge aus. Im Kontext dienstorientierter Architekturen wird ein solches Softwaresystem auch als Orchestration Engine bezeichnet. Wie bereits in Abschnitt 3.2.1.1 beschrieben, ist das Konzept dienstorientierter Architekturen bereits für automatisierungstechnische Systeme – insbesondere auch für Maschinen und Anlagen – umgesetzt; diese Ansätze basieren jedoch auf ereignisbasierten Steuerungssystemen – im Gegensatz zu dem zyklisch operierenden IEC 61131 Standard. In eigenen Vorarbeiten [LVH14b, LVH15] konnte jedoch gezeigt werden, dass eine Orchestration Engine auch auf IEC 61131 konformen Steuerungen implementiert werden kann. In [Bü14, BLVH15] wurde ferner gezeigt, dass eine flexible Anpassung des Workflow Models – d.h. verschiedene Handlungspläne werden durch HiTraP-AT bestimmt – während dem Betrieb einer Maschine und Anlage am Beispiel der Stempelanlage möglich ist.

Die Implementierung einer solchen Orchestration Engine kann sowohl in der klassischen (nicht objekt-orientierten) IEC 61131 als auch in der objekt-orientierten Version erfolgen. Auf Grund der Abwärtskompatibilität der objektorientierten Version des IEC 61131 Standards ergeben sich lediglich kleine Unterschiede in der Implementierung, welche ausschließlich architektonischer, nicht jedoch algorithmischer Natur sind. Während durch Objektorientierung einzelne Operationen von Modulen (in UML-OP als Komponenten bezeichnet) als Methoden einer, das Modul repräsentierenden, Klasse (d.h. entsprechend IEC 61131 einem Funktionsbaustein) realisiert werden können (vgl. Abbildung 4.19(a)), ist dies ohne Objektorientierung lediglich durch individuelle Konventionen implizit möglich. So kann beispielsweise jede Funktionalität bzw. Operation eines Moduls als



(a) Beispielhafte Deklarationen des *Kran*-moduls der Stempelanlage in objektorientierter IEC 61131. (b) Beispielhafte Deklarationen in klassischer IEC 61131 des *Kran*-moduls der Stempelanlage.

Abbildung 4.19.: Unterschiede der Deklarationen in klassischer und objektorientierter IEC 61131 am Beispiel des *Kran*-moduls der Stempelanlage.

separater Funktionsbaustein⁵⁴ realisiert werden, wobei mittels Namenskonvention die Zuordnung zu einem spezifischen Modul gekennzeichnet wird (siehe Abbildung 4.19(b)). Auf Basis der innerhalb der HiTraP-AT Planungsproblems bzw. der UML-OP gegebenen Informationen ist dabei lediglich die Generierung der Rahmen – d.h. die Definition bzw. Signatur der jeweiligen Klassen und Methoden – möglich. Die Implementierung der Funktionen, d.h. der Methodenrumpfe kann in diesem Fall wahlweise manuell erfolgen oder mittels einer automatischen Generierung durch Informationen anderer Modelle, wie im Falle der von Daniel Schütz vorgeschlagene SysML-AT.

Neben den Funktionsbausteinen, welche Module bzw. deren Operationen implementieren, wird mindestens ein weiterer Funktionsbaustein benötigt, um die Gesamtfunktionalität einer Anlage zu implementieren und letztendlich die Ablaufsteuerung auf Basis bereits implementierter Operationen zu realisieren. Im Wesentlichen erfolgt dies, unabhängig ob klassische oder objektorientierte IEC 61131, durch zwei konzeptionelle Bausteine: die explizite Repräsentation des Handlungsplans in Form (veränderbarer) Variablen (d.h. dem Workflow Model) und dessen Interpretation in Form der Aufrufe von Operationen entsprechend der gegebenen Reihenfolge (d.h. der Funktionalität, welche einer Orchestration Engine entspricht).

Abbildung 4.20 zeigt den Deklarationsteil zur Implementierung eines vollständig sequentiellen, konfigurierbaren Handlungsplans. Die auszuführende Sequenz von Operationen ist innerhalb des Arrays *plan* beginnend mit allen Anlaufoperationen der Module gefolgt von den Operationen der Handlungsräume "Produktion" und "Rücklauf" (siehe auch Abschnitt 4.1.3.3) definiert. Die einzelnen Operationen sind mittels Identifikator des spezifischen Datentyps *E_Operations* repräsentiert (siehe hierzu auch nachfolgend in Abbildung 4.21). Im Rahmen des IEC 61131 Standards wird keine dynamische Speicherzuordnung unterstützt. Aus diesem Grund wird durch die Konstante *MAX_PLAN_LENGTH* eine Länge des Plans definiert, welche groß genug gewählt wird, um beliebige Pläne innerhalb der betrachteten Anlage zu repräsentieren. Die Länge des aktu-

⁵⁴Eine Funktionalität kann auch als Funktion realisiert werden, sofern die Implementierung zustandslos, d.h. ohne interne Zustände, erfolgen kann. Um eine einheitliche Implementierung der Funktionalitäten zu erreichen, werden im Folgenden ausschließlich Funktionsbausteine genutzt. Kann eine zustandlose Implementierung erfolgen, werden keine Variablen innerhalb des Funktionsbausteins verwendet.

4. Bestimmung der Ablaufreihenfolge von Automatisierungsfunktionen

```
VAR CONSTANT
  MAX_PLAN_LENGTH: INT := MY_MAX_PLAN_LENGTH;
...
END_VAR
VAR
  plan: ARRAY[0..MAX_PLAN_LENGTH] OF INT := [E_Operations.ANLAUF_OP_1
                                             E_Operations.ANLAUF_OP_2,
                                             E_Operations.PRODUKTION_OP_1,
                                             E_Operations.PRODUKTION_OP_2,
                                             ...,
                                             E_Operations.RUECKFUEHRUNG_OP_1,
                                             E_Operations.RUECKFUEHRUNG_OP_2,
                                             ...];

  currentOperation: INT := 0;
  rampUpPlanLength: INT := 2;
  planLength: INT := MY_PLAN_LENGTH;
...
END_VAR
```

Abbildung 4.20.: Auszug des Deklarationsteils für die Repräsentation sequentieller Handlungspläne in IEC 61131-3 ST.

ell auszuführenden Plans ist durch die Variable *planLength* definiert; die Länge des gesamten Anlaufplans durch die Variable *RampUpPlanLength*.

Die wesentlichen Programmbestandteile der Ablaufsteuerung (Orchestration Engine) sind ebenfalls in Abbildung 4.21 rechts in Strukturiertem Text dargestellt. Wie bereits angeführt, werden die einzelnen Operationen innerhalb des Arrays, welches den Handlungsplan repräsentiert, durch Identifikatoren repräsentiert. Dieser sogenannte nutzerdefinierte Datentyp stellt eine Aufzählung aller innerhalb des Planungsproblems definierten Aktionen bzw. innerhalb der Komponentenmodelle eines UML-OP Modells definierten Operationen dar. Abbildung 4.21 stellt diesen Zusammenhang am Beispiel der Krankkomponente dar. Im oberen Teil des gegebenen Programmcodes der Ablaufsteuerung wird entsprechend der innerhalb des Handlungsplans angegebenen Operation (mittels *plan[currentOperation]* gegeben) die entsprechende Operation aufgerufen⁵⁵. Effektdefinitionen bzw. Nachbedingungen (wie sie aus der in Abschnitt 4.2 beschriebenen Abbildung zwischen UML-OP und HiTraP-AT Planungsproblem hervorgehen) werden innerhalb der Ablaufsteuerung genutzt, um die erfolgreiche Ausführung der jeweiligen Operation zu prüfen – d.h. eine Operation ist vollständig ausgeführt und der erwartete Zustand nach Ausführung ist eingetreten – und innerhalb der booleschen Variablen *success* zu speichern. Im unteren Bereich des gegebenen Programmausschnitts der Ablaufsteuerung wird, falls eine Operation erfolgreich ausgeführt würde, durch Inkrementierung des Zählers (Variable *currentOperation*) die nächste, auszuführende Operation ausgewählt. Ist das Ende eines Plans erreicht, wird dessen Ausführung (ohne Berücksichtigung der Operationen des Handlungsplans "Anlauf") erneut gestartet. Durch den explizit, lediglich innerhalb von Variablen spezifizierten Handlungsplan ist dessen Veränderung durch externe Software (beispielsweise mittels standardisierter Schnittstellen wie

⁵⁵Im Rahmen dieser Programmierung wird von einer objektorientierten Implementierung ausgegangen und Operationen als Methoden aufgerufen. Im Falle der Realisierung mit klassischer IEC 61131-3 müssen diese Aufrufe durch entsprechende Aufrufe der die Operation repräsentierenden Funktionsbausteine ersetzt werden.

4.3. Anwendung von HiTraP-AT für IEC 61131-3 basierte Steuerungssoftware

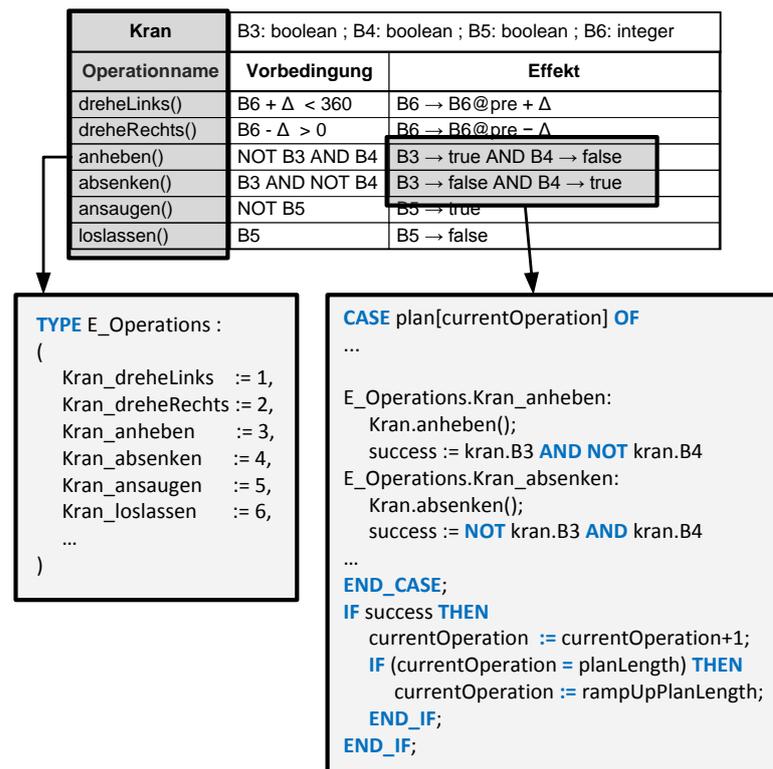


Abbildung 4.21.: Ableitung der Identifikatoren der Operationen (links) und Ausschnitt der entsprechenden Ablaufsteuerung (rechts) jeweils in IEC 61131-3 ST am Beispiel des Kranmoduls der Stempelanlage.

OPC bzw. OPC UA [MLD09]) möglich. Hierdurch kann entsprechend dem Planungsergebnis des HiTraP-AT Planungswerkzeugs folglich während dem Betrieb eine Anpassung des Handlungsplans vorgenommen werden, um z.B. die in User Story 2 beschriebene Zielsetzung zu erreichen.

Zusammenfassend wurde in diesem Abschnitt gezeigt, wie IEC 61131-3 konformer, konfigurierbarer Programmcode realisiert werden kann, welcher durch andere Softwareprogramme innerhalb und außerhalb der Steuerungsebene veränderbare Handlungssequenzen ausführen kann. Ferner wurde gezeigt, wie und in welcher Weise dieser Programmcode mittels Informationen aus dem HiTraP-AT Planungsproblem bzw. der UML-OP generiert werden kann. Hierdurch wird Anforderung A4 nach Anwendbarkeit des HiTraP-AT Planungsverfahrens für IEC 61131-3 basierte Steuerungssoftware im Kontext von Anwendungsfall 1 (dynamische Rekonfiguration zur Störungskompensation) adressiert.

5. Untersuchung der Anwendbarkeit und Evaluation von HiTraP-AT

Im vorangegangenen Kapitel wurde ein Verfahren für die Bestimmung der Abläufen von Automatisierungsfunktionen vorgeschlagen, welches die im Rahmen der Anforderungsermittlung (vgl. Abschnitt 2.2) ermittelten, spezifischen Anforderungen der Steuerungsebene der Maschinen- und Anlagenautomatisierung adressiert: der HiTraP-AT Ansatz. Um zu evaluieren, inwiefern das vorgeschlagene HiTraP-AT Planungsverfahren die gestellten Anforderungen tatsächlich erfüllt und dadurch letztendlich die Anwendbarkeit und Vorteilhaftigkeit des in dieser Arbeit vorgeschlagenen Konzeptes aufzuzeigen, wurden verschiedene Untersuchungen durchgeführt und ausgewertet. Die Beschreibung dieser Untersuchungen, deren Ergebnisse und Bewertung insbesondere im Hinblick auf die gestellten Anforderungen sind Gegenstand dieses Kapitel.

Eingangs wird untersucht, inwieweit die Integration des vorgeschlagenen HiTraP-AT Planungsverfahrens in modellgetriebene Entwicklungsansätze (Anforderung A5) adressiert werden konnte. Im Anschluss hierzu wird in Abschnitt 5.2 eine ausführliche Untersuchung der Leistungsfähigkeit des HiTraP-AT Planungsalgorithmus im Hinblick auf dessen zeitliches Verhalten sowie Skalierbarkeit (Anforderung A3) beschrieben. In Abschnitt 5.3 wird anschließend untersucht, ob sich das HiTraP-AT Planungsverfahren für die Bestimmung von Ablaufplänen für IEC 61131-3 basierte Steuerungssoftware eignet (Anforderung A4). Eine Untersuchung, ob durch das im Rahmen dieser Arbeit beschriebene Konzept zur automatischen Bestimmung von Ablaufreihenfolgen von Automatisierungsfunktionen die weiteren gestellten Anforderungen A1 und A2 adressiert werden, wird in Abschnitt 5.4 untersucht. Im Anschluss an die Beschreibung der Evaluation, werden die Ergebnisse und Grenzen im Hinblick auf die Erfüllung der Anforderungen in Abschnitt 5.5 zusammengefasst. Im Rahmen der durchgeführten Untersuchungen und damit verbundenen Anwendung des HiTraP-AT Planungsverfahrens für unterschiedliche fertigungstechnische Laboranlagen des Lehrstuhls für Automatisierung und Informationssysteme (AIS) konnten Grenzen des Ansatzes, welche über die gestellten Anforderungen hinausgehen, bereits identifiziert werden. Diese sind in Abschnitt 5.6 beschrieben. Abschließend werden in Abschnitt 5.7 die Ergebnisse dieser Arbeit im Kontext der eingangs in Abschnitt 1.1 beschriebenen Anwendungsbeispiele reflektiert und der sich hierdurch ergebende anwendungsspezifische Beitrag dieser Dissertation beschrieben.

5.1. Untersuchung der Integration in einen modellgetriebenen Entwicklungsansatz (Anforderung A5)

Dieser Abschnitt untersucht die Integration des im Rahmen dieser Arbeit entwickelten Planungsverfahrens HiTraP-AT mit modellgetriebenen Entwicklungsansätzen des Maschinen- und Anlagenbaus. Mit dieser Untersuchung soll evaluiert werden, ob die entsprechend Anforderung A5 geforderte Möglichkeit der Integration des im Rahmen dieser Arbeit entwickelten Planungsverfahrens in modellgetriebene Entwicklungsansätze erfüllt werden konnte. Dabei wird das in Abschnitt 4.2 beschriebene Integrationskonzept zwischen HiTraP-AT und der von Daniel Schütz entwickelten UML-OP genutzt. Die Struktur dieses Abschnitts spiegelt dabei die zwei wesentlichen, sich aus Anforderung A5 ergebenden Fragestellungen wieder: Zum einen sollte sichergestellt

5. Untersuchung der Anwendbarkeit und Evaluation von HiTraP-AT

sein, dass mittels der Integration von HiTraP-AT und UML-OP die Anforderung an eine Integration des Planungsansatzes in modellgetriebene Entwicklungsansätze erfolgreich adressiert werden kann. Zum anderen muss untersucht werden, ob auf Basis von in Form von UML-OP gegebenen Modellen eine erfolgreiche, korrekte Planung der Abläufe der Steuerungssoftware durchgeführt werden kann. Können diese beiden Fragestellungen positiv evaluiert werden, kann auf die Erfüllung von Anforderung A5 geschlossen werden.

In der Dissertation von Daniel Schütz [Sch14] wird ein Modellierungsansatz für die Architektur von Softwareagenten und deren Wissensbasen vorgeschlagen: das sogenannte SysML-AT Profil. Die Arbeit adaptiert dabei sowohl die SysML (wie in [VHSFL14a] beschrieben) als auch die UML (siehe [SLVH12]). Ferner wird in der Dissertation von Daniel Schütz die Erweiterung des SysML-AT Profils um die Konzepte der UML-OP beschrieben; ebenso wie die Integration des SysML-AT Profils mit weiteren Vorgehensmodellen und Entwicklungsansätzen, wie z.B. dem von Timo Frank in seiner Dissertationen beschriebenen Ansatz für den modellbasierten Architektorentwurf verteilter Automatisierungssysteme [Fra14]. Auf Basis der sich aus den Arbeiten von Daniel Schütz ergebenden, nahtlosen Integration der UML-OP in die durch die von ihm ebenfalls entwickelte, modellgetriebene Vorgehensweise für die Softwareentwicklung der Steuerungsebene des Maschinen- und Anlagenbaus sowie dessen Einbettung innerhalb weiterer Ansätze für die modellgetriebene Software- und Systementwicklung kann zusammengefasst werden, dass die UML-OP ein repräsentatives Beispiel für einen modellgetriebenen Entwicklungsansatz darstellt, um als Basis für die Adressierung von Anforderung A5 – der Integration des entwickelten Planungsverfahrens in modellgetriebene Entwicklungsansätze – genutzt werden zu können.

Switch	pos: integer; loaded: bool	
Operations	Precondition	Effect
turnLeft()	true	pos \rightarrow (pos@pre + Δ) mod 360
turnRight()	true	pos \rightarrow (pos@pre - Δ) mod 360

Conveyor	lb1:bool; lb2: bool; speed:int; loaded: bool	
Operations	Precondition	Effect
stop()	speed \neq 0	speed \rightarrow 0
startForward()	speed = 0	speed \rightarrow (> 0)
transportForward()	speed > 0 and loaded	lb2 \rightarrow true
intakeFront()	speed > 0 and not loaded	lb1 \rightarrow true AND loaded \rightarrow true

Plant	C1:Conveyor; ... ; C13: Conveyor; C14:Conveyor; ... C24: Conveyor; S1: Switch; ... ; S8: Switch; S9:Switch;...S12:Switch; F1: Filling1; F2: Filling2	
Operations	Precondition	Effect
C1.intakeFront()	S1.pos = 0	
C2.intakeFront()	S1.pos = 90	
C14.intakeFront()	S1.pos = 270	
...

Product	loc: {C1,...,C13,C14,...,C24 S1,...,S8,S9,...,S12, F1,F2}; red: {0,...,5}; green:{0,...,5}; blue:{0,...,5}; yellow: {0,...,5}	
Operations	Precondition	Effect
C1.intakeFront()	loc = C1 and red = 0 and green = 0 and blue = 0 and yellow = 0	loc \rightarrow C1
F2.toggleYellow()	yellow < 4	yellow \rightarrow yellow@pre + 1
...

Abbildung 5.1.: Ausschnitt des UML-OP Modells des Hybriden Prozessmodells (in Anlehnung an [LSVH13]). Helle Textelemente visualisieren den Unterschied zwischen Ausbaustufe 1 und Ausbaustufe 2 der Laboranlage; die Anzahl abzufüllender Granulate ist durch Variablen x_1, \dots, x_4 gegeben.

In gemeinsamen Vorarbeiten mit Daniel Schütz et al. [LSVH13] wurde ein UML-OP Modell für die Systemfallstudie "Hybrides Prozessmodell" (vgl. Abschnitt A.2) erstellt. Ein Auszug des entsprechenden UML-OP Modells in tabellarischer Notation⁵⁶ ist in Abbildung 5.1 dargestellt. Dieses Modell wurde unter Rückgriff auf die in Abschnitt 4.2.2 beschriebene Abbildung zwischen UML-OP und HiTraP-AT für die Bestimmung der Ablaufreihenfolge der einzelnen Operatio-

⁵⁶Die tabellarische Notation der UML-OP wurde in Abschnitt 4.2.1 eingeführt.

5.2. Untersuchung von Zeit- und Skalierbarkeitsverhalten des HiTraP-AT Planungsalgorithmus

nen der Komponenten mittels des in dieser Arbeit entwickelten Planungsverfahrens genutzt. Für die Untersuchung wurde dabei auf die in Abschnitt 5.2.1 beschriebene Implementierung des HiTraP-AT Planungsverfahrens zurückgegriffen. Das Eingabemodell des Planungswerkzeugs wurde auf Basis gegebener UML-OP Modelle entsprechend der definierten Abbildung und ausschließlich mittels der darin enthaltenen Informationen manuell erstellt. Dabei wurden beide in Abschnitt 1.1 beschriebenen Anwendungsfälle berücksichtigt. Zum einen wurde das entwickelte UML-OP Modell für eine automatische Bestimmung der optimalen Reihenfolge der Automatisierungsfunktionen beider Ausbaustufen der Laboranlage genutzt (siehe hell gekennzeichnete Modellunterschiede der Ausbaustufen in Abbildung 5.1) und spiegelt damit die in User Story 1 (vgl. Abschnitt 1.1.1) geforderte Funktionalität des Planungsverfahrens für die Generierung von Vorschlägen möglicher Abläufe wieder. Zum anderen wurde bereits im Rahmen der User Story 2 in Abschnitt 1.1.2 identifiziert, dass ein automatisches Planungsverfahren für die Bestimmung alternativer Abläufe im Fehlerfall genutzt werden kann. Dieser Aspekt wurde auf Basis der hier vorliegenden UML-OP Modelle aufgegriffen und simulativ untersucht. Um den Einfluss von Störungen auf die Funktionalität der Maschine bzw. Anlage zu repräsentieren, wurden entsprechende Operationen bzw. ganze Komponenten aus dem Modell manuell entfernt. So wurde beispielsweise der Ausfall einer Weiche durch Löschen der entsprechenden Variable im Systemmodell simuliert. Die Prüfung der identifizierten Abläufe des Planungsverfahrens erfolgte mittels eines manuellen Vergleichs mit zuvor (ebenfalls manuell) erstellten, als Ausgabe erwarteter Abläufe. Dabei entsprachen die durch HiTraP-AT automatisch bestimmten Abläufe zumeist den manuell spezifizierten Abläufen. Alle Unterschiede zwischen automatischen und manuell bestimmten Abläufen stellten Variationen der Reihenfolge von Operationen dar und waren nach detaillierter Prüfung vollständig auf unterschiedliche Sequenzierungen der partiell geordneten Pläne zurückzuführen. Somit wurden auf Basis der in Form von UML-OP gegebenen Modellen stets korrekte Lösungen bestimmt. Demzufolge können durch UML-OP gegebene Modelle genutzt werden, um korrekte Abläufe der darin beschriebenen Operationen der Steuerungsebene geeignet zu bestimmen.

Wie zuvor dargelegt, stellt das von Daniel Schütz entwickelte UML-OP Profil ein repräsentatives Beispiel für einen modellgetriebenen Entwicklungsansatz im Maschinen- und Anlagenbau dar. Ferner wurde die Anwendung der UML-OP für die Systemfallstudie "Hybrides Prozessmodell" beschrieben. Mittels HiTraP-AT wurden auf Basis unterschiedlicher UML-OP Modelle des Hybriden Prozessmodells automatisch geeignete Abläufe der innerhalb der UML-OP Modelle beschriebenen Operationen bestimmt. Im Rahmen der durchgeführten und in diesem Abschnitt beschriebenen Untersuchungen konnte identifiziert werden, dass auf Basis gegebener UML-OP Modelle des Hybriden Prozessmodells stets ein korrekter Ablauf bestimmt wurde und sich dementsprechend das HiTraP-AT Planungsverfahren in den durch UML-OP repräsentierten Entwicklungsansatz geeignet integrieren lässt. Somit ist die Möglichkeit der Integration des im Rahmen dieser Arbeit beschriebenen HiTraP-AT Planungswerkzeugs in modellgetriebene Entwicklungsansätze für den Maschinen- und Anlagenbau entsprechend der gestellten Anforderung A5 gegeben.

5.2. Untersuchung von Zeit- und Skalierbarkeitsverhalten des HiTraP-AT Planungsalgorithmus (Anforderung A3)

Ein geeignetes Zeit- und Skalierbarkeitsverhalten des Planungsverfahrens stellt eine wesentliche Anforderung im Hinblick auf dessen Anwendbarkeit für die Steuerungsebene des Maschinen- und

5. Untersuchung der Anwendbarkeit und Evaluation von HiTraP-AT

Anlagenbaus dar (vgl. Anforderung A3). So wird insbesondere eine gute Skalierbarkeit des Ansatzes im Hinblick auf die Größe des technischen Systems gefordert. Auch sollte das implementierte Planungswerkzeug ein adäquates, zeitliches Verhalten aufweisen um domänenspezifische Anwendungen, wie die in Abschnitt 1.1 beschriebenen Anwendungsbeispiele, geeignet zu adressieren.

Die Messung dieser Leistungsfähigkeit automatischer Planungsverfahren – d.h. insbesondere Zeit- und Skalierbarkeitsverhalten – stellt jedoch eine komplexe Herausforderung dar, insbesondere für domänenspezifische Planungssysteme [NG04]. Für domänenunabhängige Planungswerkzeuge existiert in Form des internationalen Planungswettbewerbs⁵⁷ eine allgemein verfügbare Auswahl unterschiedlicher Problemformulierungen. Im Rahmen der Aktivitäten des internationalen Planungswettbewerbs wurde ferner für domänenunabhängige Planungsverfahren ein einheitliches Eingabemodell – die PDDL – spezifiziert (siehe hierzu auch Abschnitt 3.1.1), wodurch eine Gegenüberstellung der Leistungsfähigkeit in Form von Laufzeiten unterschiedlicher Planungsverfahren und Planungswerkzeuge für die Lösung spezifischer Planungsprobleme letztendlich ermöglicht wird. Im Gegensatz dazu wird in dieser Arbeit ein konfigurierbarer Planungsansatz verfolgt, um domänenspezifische Anforderungen für die Maschinen- und Anlagenautomatisierung zu adressieren. Für solche Planungsansätze existieren zumeist weder allgemein verfügbare Problemformulierungen noch alternative Werkzeuge zur Lösung dieser spezifischen Herausforderung, welche als Basisszenario für einen direkten Vergleich der Leistungsfähigkeit dienen können. Daher wird im Rahmen dieser Arbeit auf die im Anhang A beschriebenen Systemfallstudien als exemplarische Formulierung von Planungsproblemen zurückgegriffen, um die Untersuchungen des Zeit- und Skalierbarkeitsverhaltens des HiTraP-AT Planungsalgorithmus durchzuführen.

Die Anforderung hinsichtlich der Leistungsfähigkeit an ein Planungsverfahren für die Feldebene des Maschinen- und Anlagenbaus (vgl. Anforderung A3) besteht, wie eingangs in diesem Abschnitt bereits angeführt, aus zwei Teilaspekten: (a) Anforderung an die Skalierbarkeit des Ansatzes, um (b) letztendlich eine akzeptable Laufzeit einer Planung realisieren zu können und letztendlich die in Abschnitt 1.1 beschriebenen Anwendungsbeispiele geeignet adressieren zu können. Die geforderte Leistungsfähigkeit graph-basierter Suchalgorithmen, wie in dieser Arbeit entwickelt, ist im Allgemeinen von der Größe des Suchraumes, d.h. des zu Grunde gelegten Graphen, abhängig (siehe hierzu auch die entsprechende Argumentation in Abschnitt 4.1.1.2). Die Explosion des Zustandsraumes stellt dabei die wesentliche, konzeptionelle Herausforderung dar. Um diese Herausforderung zu adressieren, wurde die Hierarchisierung des internen HiTraP-AT Planungsmodells vorgeschlagen. Als Basisszenario dient in diesem Zusammenhang ein Planungsmodell ohne Hierarchisierung, d.h. ein flacher Zustandsraum. Innerhalb des Planungsmodells spiegelt das Konzept zusammenhängender Zustandsräume (wie in Abschnitt 4.1.1.1 beschrieben) weitere, domänenspezifische Anforderungen an das Planungsverfahren wieder. Um die Vergleichbarkeit mit dem HiTraP-AT Planungsverfahren zu gewährleisten, wird innerhalb des Basisszenarios daher das Konzept zusammenhängender Zustandsräume berücksichtigt. Die Untersuchung der Größe flacher, zusammenhängender Zustandsräume und die daraus resultierenden Skalierbarkeitseigenschaften sind in Abschnitt 5.2.2 beschrieben; die entsprechende Untersuchung des HiTraP-AT Planungsverfahrens mit Hierarchisierung des Zustandsraumes – d.h. hierarchischer, zusammenhängende Zustandsräume – kann Abschnitt 5.2.3 entnommen werden. Der Vergleich dieser Untersuchungsergebnisse und deren Bewertung im Hinblick auf die Skalierbarkeit wird

⁵⁷Weitere Informationen hinsichtlich des internationale Planungswettbewerb IPC kann der Fußnote auf Seite 37 entnommen werden.

in Abschnitt 5.2.4 beschrieben. Der zweite Teilaspekt der Anforderung A3 wird im darauf folgenden Abschnitt 5.2.5 in Form der Untersuchung und Bewertung der Laufzeitergebnisse unter Berücksichtigung der Anwendungsbeispiele aus Abschnitt 1.1 durchgeführt. Da für alle diese Untersuchungen die Implementierung des HiTraP-AT Planungsverfahrens genutzt wird, wird dessen Realisierung in Abschnitt 5.2.1 im Folgenden vorab kurz beschrieben.

5.2.1. Das HiTraP-AT Planungswerkzeug

Das im vorangegangenen Kapitel beschriebene Planungsverfahren wurde in Form des HiTraP-AT Planungswerkzeugs mittels der Programmiersprache Java implementiert. Die interne Repräsentation des HiTraP-AT Planungsmodells erfolgt durch ein spezifisches Klassenmodell, welches Zustände und Transitionen, die drei verschiedenen Handlungsräume – Anlauf, Produktion und Rücklauf – sowie die drei Hierarchiestufen enthält. Ein spezifisches Planungsmodell wird als ein konkretes Objektmodell repräsentiert.

Für die Lösung des Planungsproblems wird die in Abschnitt 4.1.3 beschriebene Abbildung auf ein Lineares Programm verwendet. Zur Übergabe an ein externes Lösungsprogramm für Lineare Programme wird durch das HiTraP-AT Planungswerkzeug eine Datei im LP-Dateiformat⁵⁸ erstellt, welches das Planungsproblem als lineares Programm enthält. Dabei werden binäre Variablen $x_i \in \{0, 1\}$, welche Transitionsknoten aller Hierarchiestufen des HiTraP-AT Modells darstellen sowie die notwendigen Constraints auf Basis des vorliegenden Objektmodells erstellt. Als LP Lösungsprogramm wird auf das kommerzielle Werkzeug IBM ILOG CPLEX⁵⁹ zurückgegriffen. Wie in Abschnitt 4.1.3 beschrieben, repräsentieren binäre Variablen, deren Wert durch das LP Lösungsprogramm ausgewählt wurden – d.h. $x_i = 1$ bestimmt wurde – selektierte Transitionen im Handlungsraum. Knoten auf der jeweils höchsten Hierarchiestufe eines Handlungsraumes – Hierarchiestufe 2 für Handlungsräume Anlauf und Rücklauf sowie Hierarchiestufe 3 für Handlungsraum Produktion – bestimmen die als auszuführen gewählte Transition. Diese Transitionen sind Elemente des partiellen Handlungsplanes und werden – ebenfalls mittels eines entsprechenden Objektmodells – repräsentiert.

5.2.2. Untersuchung flacher, zusammenhängender Handlungsräume

Im Rahmen eigener Vorarbeiten [LSVH12] wurde das Zeit- und Skalierbarkeitsverhalten eines Zustandsraummodells auf Basis flacher, zusammenhängender Handlungsräume untersucht. Dabei wurde ausschließlich die in dieser Dissertation vorgeschlagene grundlegende Struktur zusammenhängender Handlungsräume (vgl. Abschnitt 4.1.1), nicht jedoch das ebenfalls vorgeschlagene Konzept der Hierarchisierung berücksichtigt. Zustände sind in diesem Fall als Tupel aller zustandsbeschreibender Variablen im betrachteten System zu verstehen. Die Semantik dieser Variablen bleibt davon unberührt. Abbildung 5.2 visualisiert das sich daraus ergebende, flache Planungsmodell bestehend aus Zuständen (dargestellt als Kreise) und Zustandsübergänge bedingt

⁵⁸LP ist ein für ILOG CPLEX spezifisches Dateiformat, welches die algebraische Formulierung des linearen Problems in einer, der natürlichen Formulierung linearer Optimierungsprobleme mittels Variablen, deren Wertebereich, (Un-)Gleichungen sowie Optimierungsfunktion (vgl. Abschnitt 3.1.2.2) sehr ähnlichen Form ermöglicht. Das Format wird nicht nur von IBM ILOG CPLEX, sondern auch von einer Vielzahl weiterer Lösungsprogramme unterstützt.

⁵⁹In der Implementierung wird das IBM ILOG CPLEX Optimization Studio V12.2 und dessen CPLEX Implementierung genutzt.

5. Untersuchung der Anwendbarkeit und Evaluation von HiTraP-AT

durch Ausführung von Aktionen (als Pfeile zwischen Zuständen dargestellt); ein exemplarischer Handlungsplan ist hell hervorgehoben. Eine automatische Planung auf Basis dieses Planungsmodells resultiert direkt in einem vollständig sequenzierten (d.h. total geordneten) Handlungsplan, welcher aus drei vollständig getrennten Handlungsteilplänen (Anlauf-, Produktions- und Rückführungsplan) besteht.

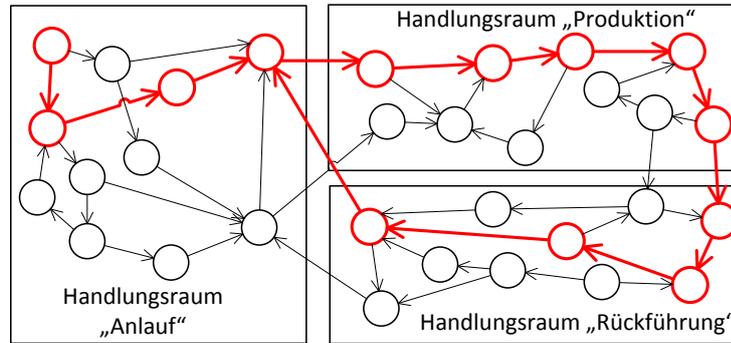


Abbildung 5.2.: Darstellung eines exemplarischen, flachen Zustandsraumes (Darstellung in Anlehnung an [LSVH12]; Zustände werden durch Kreise und Zustandsübergänge durch Pfeile visualisiert).

Analog zum HiTraP-AT Planungswerkzeug wurde die Implementierung mittels der Programmiersprache Java realisiert. Für die Lösung des Planungsproblems wird ebenfalls auf IBM ILOG CPLEX zurückgegriffen. Die Untersuchung des Zeit- und Skalierbarkeitsverhaltens flacher, zusammenhängender Zustandsräume erfolgte mittels der Systemfallstudie "Stempelanlage" (vgl. Anhang A.1), da sie eine geeignete Fallstudie für erste Untersuchungen neuer Forschungskonzepte im Maschinen- und Anlagenbau darstellt [VHFL13]. Für die Untersuchung wird dabei auf das bereits in Abschnitt 4.2 beschriebene Modell der Stempelanlage zurückgegriffen. Eine Übersicht der genutzten Aktionen der Krankomponente kann Abbildung 4.16 entnommen werden. Wie in Anforderung A3 beschrieben, stellen Sensoren mit nicht-binären Wertemengen und fein-granulare Aktionen eine große Herausforderung für Planungsverfahren dar, da diese eine wesentliche Ursache für die Explosion des Zustandsraumes darstellen. Um den Einfluss dieses Sachverhalts und damit die Skalierbarkeit wie in Anforderung A3 gefordert zu untersuchen, wird innerhalb der betrachteten Systemfallstudie exemplarisch eine (Modell-)Variable genutzt und verändert: Die Schrittweite Δ der Aktionen des Kranmoduls, welche die rotatorische Drehung beschreibt (vgl. Abschnitt 4.2.1.1). Die kombinatorische Betrachtung des Zustandes führt dabei dazu, dass die Anzahl der Knoten und Kanten mit reduzierter Schrittweite proportional ansteigt. Analog verhält es sich mit der Laufzeit, die durch das Planungswerkzeug benötigt wird, um einen Handlungsplan abzuleiten. Da die Stempelanlage im Verhältnis zu realen Systemen eine vergleichsweise geringe Komplexität und Größe — insbesondere im Hinblick auf die Anzahl der installierten Sensoren und Funktionalität — aufweist [VHFL13], und dies die grundlegenden Komplexitätsparameter des Planungsverfahrens darstellen (vgl. Anforderung A3) muss festgehalten werden, dass für reale Anlagen die Skalierbarkeitseigenschaften des Planungswerkzeugs auf Basis flacher, zusammenhängender Zustandsräume nicht ausreichend gegeben sind.

5.2. Untersuchung von Zeit- und Skalierbarkeitsverhalten des HiTraP-AT Planungsalgorithmus

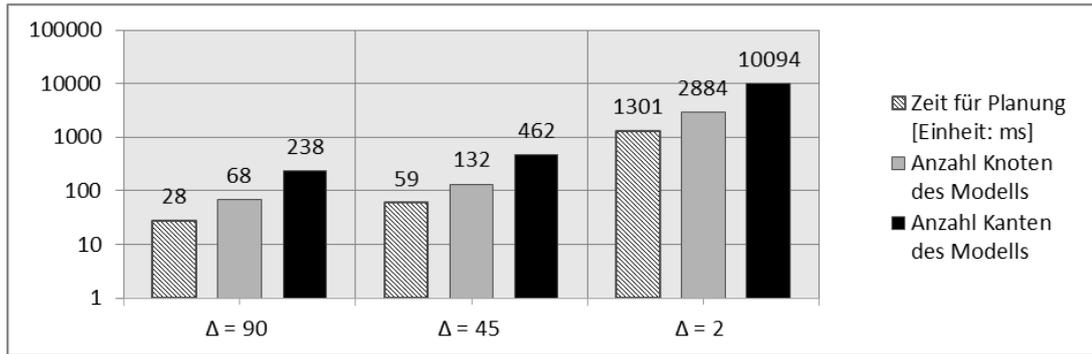


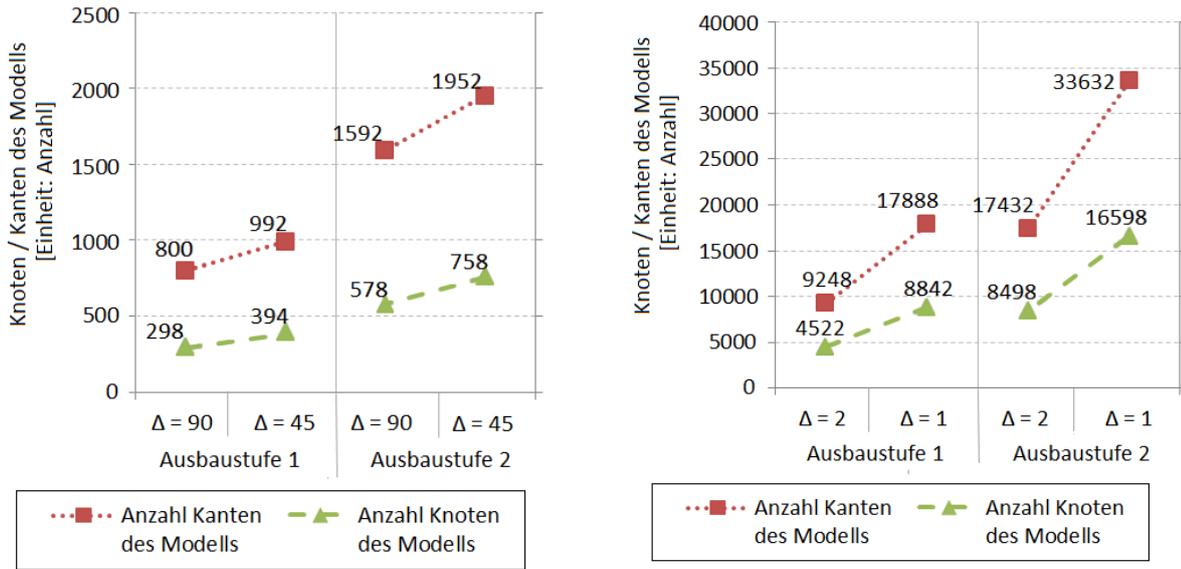
Abbildung 5.3.: Ergebnis der Anwendung flacher, zusammenhängender Handlungsräume für die Systemfallstudie "Stempelanlage".

5.2.3. Untersuchung und Bewertung der Skalierbarkeitseigenschaft des HiTraP-AT Ansatzes

Für die Untersuchung des Skalierbarkeitsverhaltens des vorgeschlagenen HiTraP-AT Planungsverfahrens (basierend auf hierarchischen, zusammenhängenden Zustandsräumen) wurde auf das in Abschnitt 5.2.1 beschriebene, implementierte HiTraP-AT Planungswerkzeug zurückgegriffen. Die Untersuchung erfolgte dabei im Zuge von Vorarbeiten [LSVH13]. Für die in diesem Abschnitt beschriebene Untersuchung der Leistungsfähigkeit des HiTraP-AT Planungsverfahrens, wurde auf das Hybride Prozessmodell als Systemfallstudie zurückgegriffen, da diese auf Grund der wesentlich größeren Anzahl an Komponenten im Vergleich zur Stempelanlage eine größere Aussagekraft im Hinblick auf die Skalierbarkeit des Ansatzes ermöglicht. Um hinsichtlich der Leistungsfähigkeit zusätzlich Trendaussagen treffen zu können, wurde das Hybride Prozessmodell in zwei sogenannte Ausbaustufen eingeteilt (siehe Anhang A.2 für Details). Dies ermöglicht eine Untersuchung der Skalierbarkeit im Hinblick auf die Anzahl der Komponenten, da Ausbaustufe 1 knapp die Hälfte der Komponenten der Gesamtanlage (als Ausbaustufe 2 bezeichnet) beinhaltet. Ferner ist die hinzugekommene Menge an Komponenten zwischen Ausbaustufe 1 und Ausbaustufe 2 homogen – d.h. es wird die Anzahl der Komponenten erhöht, wobei jedoch identische Komponenten genutzt werden. Analog zur Untersuchung der Skalierbarkeit flacher, zusammenhängender Zustandsräume werden zur Skalierung innerhalb der Ausbaustufen unterschiedliche Schrittweiten $\Delta \in \{90, 45, 2, 1\}$ der rotatorischen Weichen des Transportsystems genutzt. Hierdurch kann die Größe des Planungsmodells im Hinblick auf Knoten und Kanten erhöht werden, ohne die Ergebnisse durch Änderung weiterer Variablen des Experiments zu verfälschen (Seiteneffekt). In Abbildung 5.4 sind die experimentell ermittelten Ergebnisse dargestellt.

Im Gegensatz zu flachen, zusammenhängenden Zustandsräumen steigt bei Änderung des Drehwinkels die Anzahl der Knoten und Kanten deutlich geringer an. Zu beachten ist in diesem Zusammenhang auch, dass im Falle der Systemfallstudie der Stempelanlage durch eine Änderung der rotatorischen Schrittweite lediglich ein einzelnes Modul (Kranmodul) beeinflusst wird. Im Falle des Hybriden Prozessmodells betrifft dies alle Weichen der jeweiligen Ausbaustufe, wobei sieben (in Ausbaustufe 1) bzw. zehn (Ausbaustufe 2) rotatorische Weichen genutzt werden. Betrachtet man den durch Anpassung des Drehwinkels ergebenden Anstieg der Knoten- bzw. Kantenanzahl im Verhältnis zur Anzahl der dadurch beeinflussten Module, skaliert das HiTraP-AT Modell be-

5. Untersuchung der Anwendbarkeit und Evaluation von HiTraP-AT



(a) Modellgröße der Planung für die Schrittweiten $\Delta \in \{90, 45\}$ der Drehoperationen der rotatorischen Weichen.

(b) Modellgröße der Planung für die Schrittweiten $\Delta \in \{2, 1\}$ der Drehoperationen der rotatorischen Weichen.

Abbildung 5.4.: Ergebnis der Untersuchung der Anwendung des HiTraP-AT Planungswerkzeugs für das Hybride Prozessmodell mit unterschiedlichen Schrittweiten $\Delta \in \{90, 45, 2, 1\}$ der Drehoperationen der rotatorischen Weichen.

zätzlich der Anzahl Knoten/Kanten signifikant besser als eine rein kombinatorische Betrachtung mittels flacher (zusammenhängender) Zustandsräume.

Im Falle einer Erhöhung der Anzahl an Modulen und einer damit verbundenen Steigerung der Funktionalität des Hybriden Prozessmodells (Vergleich zwischen Ausbaustufe 1 und Ausbaustufe 2) weist das in dieser Dissertation vorgeschlagene Planungsmodell und somit das damit einhergehende Planungsverfahren positive Leistungseigenschaften auf. Durch die Erweiterung des Hybriden Prozessmodells wird die Flexibilität der Laboranlage signifikant gesteigert, da hierdurch beispielsweise eine Vielzahl zusätzlicher, alternativer Routen innerhalb des logistischen Teils möglich sind. Im Falle einer kombinatorischen Betrachtungsweise des Planungsmodells würde dies zu einem exponentiellen Anstieg der Größe im Bezug auf die Anzahl Knoten/Kanten des (zusammenhängenden) Zustandsraumes führen. Wie in Abbildung 5.4 dargestellt, zeigt im Gegensatz dazu das HiTraP-AT Planungsmodell einen nahezu linearen Anstieg der Anzahl Knoten bzw. Kanten.

5.2.4. Bewertung des Skalierungsverhaltens des HiTraP-AT Planungsverfahrens

In den beiden vorangegangenen Abschnitten wurden bereits die Ergebnisse der Untersuchungen der beiden unterschiedlichen Planungsmodelle – zusammenhängende, flache sowie zusammenhängende, hierarchische Zustandsräume – beschrieben. Im Rahmen der Anforderung A3 wird im Hinblick auf die Anwendbarkeit des Planungsverfahrens für die Feldebene der Maschinen- und Anlagenautomatisierung eine gute Skalierbarkeit gefordert, welche in diesem Abschnitt nochmals näher betrachtet wird. Wie bereits in vorangegangenen Abschnitten argumentiert und erläutert,

5.2. Untersuchung von Zeit- und Skalierbarkeitsverhalten des HiTraP-AT Planungsalgorithmus

stellt die Veränderung der Modellgröße – d.h. die Anzahl Knoten und Kanten innerhalb eines graph-basierten Modells – ein wesentliches Maß für die Skalierbarkeit eines Planungsverfahrens dar. Die Schrittweite rotatorisch drehender Komponenten – das Kranmodul im Falle der Systemfallstudie der Stempelanlage oder die Weichen im Falle des Hybriden Prozessmodells – bzw. dessen Änderung wurde in diesem Zusammenhang in der vorangegangenen Betrachtung als stellvertretende Variable der Untersuchung genutzt. Der entsprechende Skalierungsfaktor ergibt sich aus der Division der Modellgrößen zweier Szenarien (d.h. unterschiedlicher Schrittweiten) unter Berücksichtigung der Unterschiede zwischen hierarchischen und flachen (jeweils zusammenhängenden) Zustandsräumen⁶⁰. Eine Gegenüberstellung dieser Skalierungsfaktoren der beiden unterschiedlichen zusammenhängenden Zustandsräume mit und ohne Hierarchisierung ist in Abbildung 5.5 dargestellt.

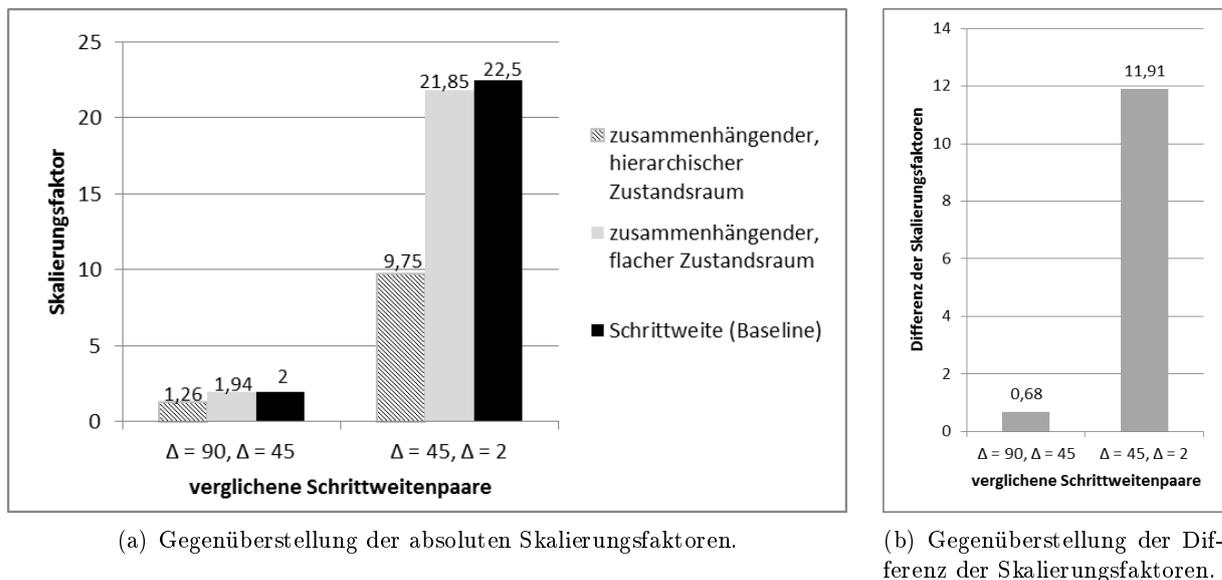


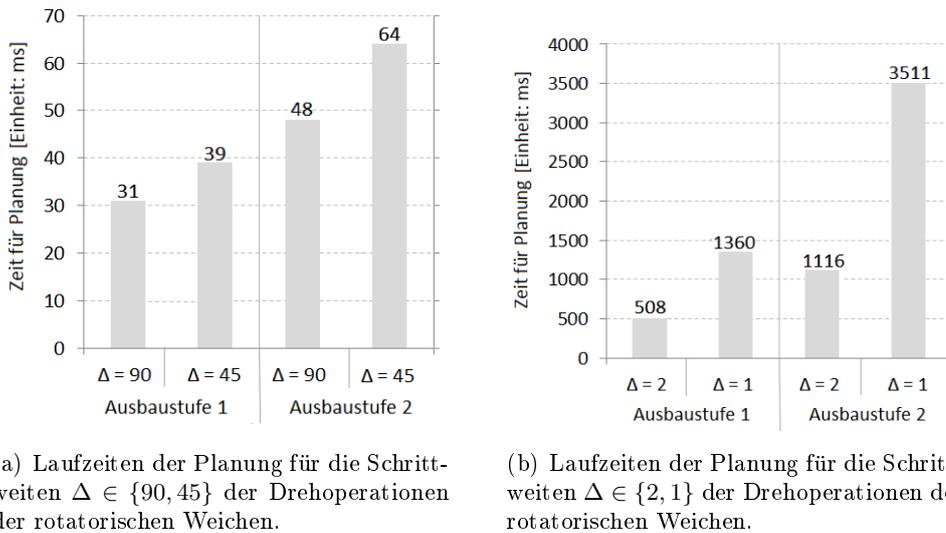
Abbildung 5.5.: Gegenüberstellung der Skalierungsfaktoren von flachen und hierarchischen, zusammenhängenden Zustandsräumen⁶⁰.

In Abbildung 5.5(a) sind die absoluten Werte der Skalierungsfaktoren gegenübergestellt. In diesem Zusammenhang kann die Skalierung der Schrittweite selbst als Basisszenario (Baseline) betrachtet werden. Während sich der Skalierungsfaktor flacher, zusammenhängender Zustandsräume direkt proportional zur Baseline verhält, weisen hierarchische, zusammenhängende Zustandsräume eine signifikant bessere Skalierung auf. Wie in Abbildung 5.5(b) quantitativ dargestellt, ist durch Anwendung des in dieser Arbeit vorgeschlagenen Hierarchisierungskonzeptes von Zustandsräumen eine deutliche Verbesserung des Skalierungsfaktors – in dem betrachteten Fall um mehr als Faktor 2 – erreicht. Demzufolge kann zusammenfassend festgehalten werden, dass durch die Hierarchisierung als konzeptueller Ansatz eine Verbesserung der Skalierbarkeitseigenschaft eines Planungsmodells erreicht werden kann.

⁶⁰Der Skalierungsfaktor stellt die Gesamtmenge aller Modellelemente – d.h. Knoten und Kanten des Graphen – gegenüber. Die Anzahl der Modellelemente des erstgenannten Modells wird für den Skalierungsfaktor als Dividend, die des zweitgenannten Modells als Divisor genutzt. Für die Gegenüberstellung der Differenz in Abbildung 5.5(b) wird der Skalierungsfaktor zusammenhängender, flacher Zustandsräume als Minuend und der Skalierungsfaktor zusammenhängender, hierarchischer Zustandsräume als Subtrahend genutzt.

5.2.5. Bewertung des Zeitverhaltens des HiTraP-AT Planungsverfahrens

Während im vorangegangenen Abschnitt die Skalierbarkeit entsprechend fokussiert wurde, wird in diesem Abschnitt eine Bewertung des durch das HiTraP-AT Planungsverfahren bzw. dessen Implementierung in Form des HiTraP-AT Planungswerkzeugs realisierte, zeitliche Leistungsverhalten des Planungsverfahrens betrachtet. Um eine Bewertung dieses Leistungsverhaltens durchführen zu können, wurde als Maßstab bereits in der Beschreibung von Anforderung A3 auf die in Abschnitt 1.1 beschrieben, exemplarischen Anwendungsfallbeschreibungen zurückgegriffen. Hierbei wurde identifiziert, dass im Hinblick auf ein zeitliches Verhalten für die gesamte Planung ein Zeithorizont von wenigen Minuten für die hier zu Grunde gelegten Anwendungen ausreichend ist. Im Folgenden wird das zeitliche Verhalten des HiTraP-AT Planungswerkzeugs entsprechend näher betrachtet.



(a) Laufzeiten der Planung für die Schrittweiten $\Delta \in \{90, 45\}$ der Drehoperationen der rotatorischen Weichen.

(b) Laufzeiten der Planung für die Schrittweiten $\Delta \in \{2, 1\}$ der Drehoperationen der rotatorischen Weichen.

Abbildung 5.6.: Zeitverhalten des HiTraP-AT Planungswerkzeugs für das Hybride Prozessmodell mit unterschiedlichen Schrittweiten $\Delta \in \{90, 45, 2, 1\}$ der Drehoperationen der rotatorischen Weichen.

In Abbildung 5.6 werden Laufzeiten für eine Bestimmung von Ablaufplänen durch HiTraP-AT in unterschiedlichen, bereits in Abschnitt 5.2.3 genutzten Szenarien visualisiert: Für beide Ausbaustufen der Systemfallstudie "Hybrides Prozessmodell" werden unterschiedliche Schrittweiten $\Delta \in \{90, 45, 2, 1\}$ für die Drehoperationen der rotatorischen Weichen genutzt. Die angegebenen Zeiten stellen den Mittelwert von jeweils zehn Planungsläufen auf einem handelsüblichen Business Notebook⁶¹ dar. Zu erkennen ist hierbei, dass die absoluten Zeiten selbst im komplexesten Szenario (Ausbaustufe 2 mit $\Delta = 1$) lediglich wenige Sekunden umfasst. Die gemessenen Laufzeiten des Planungsalgorithmus hängen jedoch signifikant von der genutzten Plattform, d.h. Leistungswerte der Hardware des verwendeten Rechner- bzw. Computersystems aber auch von dessen Betriebssystem, sowie der eigentlichen Implementierung des Planers, ab. Zumeist lassen sich Seiteneffekte wie bei dem hier genutzten, handelsüblichen Laptop kaum verhindern bzw. deterministisch innerhalb der Untersuchungen berücksichtigen. Ein Vergleich der absoluten Messwerte ist daher noch aussagekräftiger.

⁶¹Für die Testläufe wurde ein Lenovo ThinkPad T81 genutzt.

5.2. Untersuchung von Zeit- und Skalierbarkeitsverhalten des HiTraP-AT Planungsalgorithmus

Das zeitliche Verhalten des Planungswerkzeugs im Bezug auf die Modellgröße (unabhängig von der betrachteten Ausbaustufe) ist in Abbildung 5.7 dargestellt. Das Planungswerkzeug weist schwach exponentiell wachsendes Laufzeitverhalten auf, wie die in der Abbildung dargestellte Trendlinie⁶² zeigt. Selbst für größere Planungsmodelle mit mehr als 50000 Modellelementen (vgl. $\Delta = 1$ in Ausbaustufe 1 des Hybriden Prozessmodells) erfolgt eine performante Lösung des Planungsproblems.

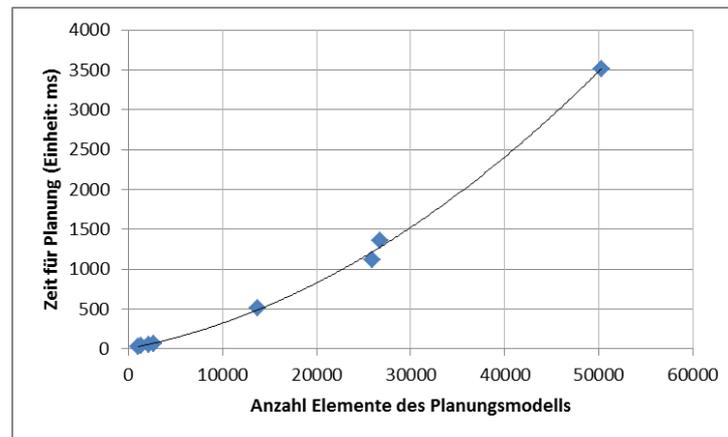


Abbildung 5.7.: Experimentelles Laufzeitverhalten des HiTraP-AT Planungswerkzeugs im Bezug zur Größe des Planungsmodells.

Im Rahmen dieses Abschnitts wurde das zeitliche Verhalten des HiTraP-AT Planungswerkzeugs insbesondere unter Berücksichtigung der in Abschnitt 1 beschriebenen Anwendungsfälle betrachtet und untersucht. Die auf einfachem Rechensystem gemessenen absoluten Zeiten sowie das im Hinblick auf die Modellgröße lediglich schwach exponentiell steigende Zeitverhalten lassen hierbei darauf schließen, dass der notwendige zeitliche Rahmen (auch für größere Maschinen und Anlagen) erfüllt werden kann.

5.2.6. Zusammenfassung

Im Rahmen dieses Abschnitts wurde die Leistungsfähigkeit – d.h. das zeitliche Verhalten sowie die Skalierbarkeitseigenschaften – des HiTraP-AT Planungsalgorithmus evaluiert. Entsprechend der gestellten Anforderung A3 wurden hierbei zwei Teilaspekte untersucht: Die Auswirkungen der Anwendung des Konzeptes der konditionalen Hierarchisierung auf die Größe und Skalierbarkeit des Planungsmodells sowie das zeitliche Verhalten des implementierten Planungsverfahrens. Die Vorteilhaftigkeit der Hierarchisierung zusammenhängender Zustandsräume wurde durch einen Vergleich von Größen der Planungsmodell mit und ohne Anwendung des Hierarchisierungskonzeptes erfolgreich dargelegt. Eine Anwendung des HiTraP-AT Planungsmodells führt dabei zu einer deutlichen Reduktion der Modellgröße mit verbesserter Eigenschaft hinsichtlich Skalierbarkeit. Dies spiegelt sich letztendlich auch im Laufzeitverhalten des Planungswerkzeugs wieder. Die Identifikation von optimalen Abläufen konnte dabei in Zeiten erfolgen, welche entsprechend der ebenfalls im Rahmen dieses Kapitels durchgeführten Quantifizierung und Bewertung der zeitlichen Anforderung für die in Abschnitt 1 beschriebenen, exemplarischen Anwendungsbe-

⁶²Die dargestellte Trendlinie wurde als Polynom zweiten Grades berechnet.

spiele ausreichend sind. Dementsprechend weist das HiTraP-AT Planungsverfahren bzw. dessen Implementierung die notwendigen Leistungswerte auf, um die in Anforderung A3 geforderten Eigenschaften entsprechend zu reflektieren und damit diese Anforderung geeignet zu adressieren.

5.3. Untersuchung der Eignung des HiTraP-AT Planungsverfahrens für IEC 61131-3 basierte Steuerungen (Anforderung A4)

Die Evaluation der Anwendbarkeit des HiTraP-AT Planungsverfahrens für Speicherprogrammierbarer Steuerungen nach IEC 61131-3 und damit die Erfüllung der gestellten Anforderung A4 wird in diesem Abschnitt beschrieben. Eignung des in dieser Arbeit vorgeschlagenen HiTraP-AT Planungsverfahrens für IEC 61131-3-basierte Steuerungssoftware liegt vor, wenn zum einen das in HiTraP-AT genutzte Aktionsmodell in IEC 61131-3 umgesetzt werden kann und zum anderen die durch das Planungsverfahren identifizierten Ablaufreihenfolgen genutzt werden können, um den technischen Prozess der Maschinen und Anlagen korrekt zu realisieren. Im Rahmen von Vorarbeiten sowie einigen studentischen Arbeiten wurden diverse Laboranlagen des Lehrstuhls AIS durch IEC 61131-3 basierte Steuerungssoftware automatisiert, wobei stets Konzepte mit Bezug zu dem in dieser Arbeit vorgeschlagenen HiTraP-AT Planungsverfahren zu Grunde gelegt wurden. Dabei wurde teils die Funktionalität eines konkreten, gegebenen Handlungsplanes realisiert oder das in Abschnitt 4.3.2 beschriebene Implementierungskonzept für konfigurierbare Handlungssequenzen berücksichtigt.

Im Rahmen der Bachelorarbeit von Max Hujber [Huj12] wurde unter der wissenschaftlichen Betreuung von Daniel Schütz der UML-OP Modellierungsansatz auf eine Laboranlage des Lehrstuhls AIS angewandt. Die Laboranlage wird in der Lehre für studentische Praktika im Bereich Automatisierungstechnik eingesetzt und realisiert mittels vier von einander unabhängigen Anlagenmodulen einen Abfüllungsprozess von Schüttgut in Flaschen und deren Kommissionierung. Im Rahmen der Abschlussarbeit von Herrn Hujber wurde auf Basis eines UML-OP Modells dieser Praktikumsanlage – insbesondere auf Basis der Definition der Operationen einzelner Module sowie deren Vor- und Nachbedingungen – ein entsprechendes Steuerungsprogramm nach IEC 61131-3 manuell mittels des Programmierwerkzeugs SIMATIC STEP7 der Firma Siemens implementiert. Durch die erfolgreiche Umsetzung des korrekten Modellprozesses der Praktikumsanlage konnte letztendlich erfolgreich evaluiert werden, dass der in der UML-OP zu Grunde gelegte Modellierungsansatz – insbesondere die Kapselung von einzelnen Funktionen der Module und deren entsprechende Sequenzierung an dieser Anlage angewandt werden können. Hierdurch kann auf Grund der Abbildbarkeit zwischen der UML-OP und HiTraP-AT, wie sie in Abschnitt 5.1 bereits positiv evaluiert wurde, ferner geschlossen werden, dass auch das Konzept der Aktionen, wie sie im HiTraP-AT Planungsverfahren Verwendung finden, für die Anwendung statischer Steuerungssoftware nach IEC 61131-3 für diese Anlage anwendbar ist.

Der Modellierungsansatz der UML-OP für die Systemfallstudie "Stempelanlage" wurde im Rahmen gemeinsamer Vorarbeiten mit Daniel Schütz et al. [LSVH12, SLVH12, LSVH13] angewandt und untersucht. Dabei wurde durch Daniel Schütz et al. [SLVH12] insbesondere positiv evaluiert, dass die UML-OP zur Beschreibung der Operationen von Maschinen und Anlagen – d.h. der Automatisierungsfunktionen – geeignet ist. Die in diesem Modell enthaltenen Informationen werden in [LSVH12] genutzt, um automatisch einen geeigneten Handlungsplan abzuleiten. Im Rahmen der Evaluation des Gesamtansatzes wurde ferner eine Implementierung der Stempel-

5.3. Untersuchung der Eignung von HiTraP-AT für IEC 61131-3 basierte Steuerungen

anlage mit der Programmierumgebung CODESYS der Firma 3S umgesetzt. Dabei wurden zum einen die in der UML-OP spezifizierten Operationen und deren Vor- und Nachbedingungen sowie deren Reihenfolge entsprechend des durch das Planungsverfahren identifizierten Handlungsplanes manuell implementiert. Bei der in [LSVH12] genutzten Implementierung handelte es sich um die Implementierung des Planungsverfahrens auf Basis flacher, zusammenhängender Zustandsräume (wie in Abschnitt 5.2.2 beschrieben) und nicht um das vollständige HiTraP-AT Planungsverfahren. Beide Implementierungen nutzen jedoch eine identische Definition des Planungsproblems selbst (siehe hierzu auch [LSVH13]). Ferner lassen sich total geordnete Handlungspläne, wie sie durch diese Implementierung auf Basis zusammenhängender, flacher Handlungsräume bestimmt werden, auch auf Basis partiell-geordneter HiTraP-AT Handlungspläne erstellen (siehe hierzu auch Abschnitt 4.1.3.3). Dementsprechend kann hierdurch positiv evaluiert werden, dass das HiTraP-AT Planungsverfahren für die Automatisierung der Stempelanlage mittels IEC 61131-3 Steuerungssoftware geeignet ist.

Im Zuge der Evaluation weiterer gemeinsamen Vorarbeit mit Daniel Schütz et al. [LSVH13] wurde das in Abschnitt 4.3.2 beschriebene Konzept zur Realisierung konfigurierbarer Handlungspläne für die Automatisierung des Hybriden Prozessmodells (siehe Abschnitt A.2) umgesetzt. Als Automatisierungsfunktionen wurden die in Abbildung 5.1 aus Abschnitt 5.1 dargestellten Operationen genutzt. Für die manuelle Implementierung des Steuerungsprogramms mit konfigurierbaren Handlungsplänen in klassischer IEC 61131-3 kam das Programmierwerkzeug SIMATIC STEP7 zum Einsatz. Wie bereits im Zuge der Evaluation von Anforderung A5 beschrieben, wurden zahlreiche unterschiedlicher HiTraP-AT Planungsprobleme für das Hybride Prozessmodell bestimmt (vgl. Abschnitt 5.1). Diese wurden als Konfiguration der Steuerungssoftware genutzt. Dabei konnte beobachtet werden, dass die unterschiedlichen, durch das HiTraP-AT Planungsverfahren automatisch bestimmten Handlungspläne zur Realisierung der erwarteten technischen Prozesse erfolgreich ausgeführt werden konnten.

Auf Basis der Erkenntnisse aus der Implementierung der Steuerungssoftware für konfigurierbare Handlungspläne für das Hybride Prozessmodell wurde im Rahmen der Masterarbeit von Ulrich Bühler [Bü14] die automatische Generierung von IEC 61131-3 konformen Steuerungscode für konfigurierbare Handlungspläne untersucht. Ausgangspunkt der Generierung des Programmcodes stellt dabei die UML-OP von Daniel Schütz dar. Die Generierung des Steuerungsprogramms wurde in der Arbeit von Herrn Bühler mittels Model-to-Text (M2T) Transformation umgesetzt. Bei der Implementierung des M2T Transformation wurde dabei auf das Werkzeug Aceleo zurückgegriffen. Der durch die Transformation automatisch generierte Programmcode stellte die Implementierung des in Abschnitt 4.3 beschriebenen Konzeptes in klassischer IEC 61131-3 dar. Der Steuerungscode wurde anschließend manuell um die Implementierung der UML-OP Operationen in der Programmierumgebung CODESYS ergänzt. Für die Evaluation der automatischen Generierung wurde die Stempelanlage genutzt. Automatisch generierter Steuerungscode (zzgl. der manuellen Implementierung der Softwarefunktionen) wurde mit unterschiedlichen, manuell definierten Handlungsplänen ausgeführt. Dabei konnten unterschiedliche, korrekte technische Prozesse der Stempelanlage ausschließlich durch Änderung der Werte des Arrays *plan* (siehe Abschnitt 4.3.2) realisiert werden. Dies zeigt zum einen, dass eine automatische Generierung von Steuerungscode für konfigurierbaren Handlungspläne der Stempelanlage möglich ist und zum anderen, dass entsprechende Handlungspläne, welche die erwarteten, technischen Prozesse kor-

5. Untersuchung der Anwendbarkeit und Evaluation von HiTraP-AT

rekt umsetzen, ausschließlich durch eine geeignete Anpassung der Variablenwerte, welche den Handlungsplan innerhalb der Steuerungssoftware repräsentieren, realisiert werden können.

Im Zuge unterschiedlicher, zuvor beschriebener Arbeiten konnte gezeigt werden, dass das HiTraP-AT Planungsverfahren für IEC 61131-3 basierte Steuerungen unter Anwendung der in Abschnitt 4.3 vorgeschlagenen Konzepte eingesetzt werden kann. Dabei können durch das Planungsverfahren identifizierte Abläufe von Automatisierungsfunktionen sowohl für eine manuelle Implementierung genutzt werden (wie in Abschnitt 4.3.1 vorgeschlagen) als auch für die Konfiguration entsprechenden Steuerungscode, welcher das in Abschnitt 4.3.2 beschriebene Konzept für konfigurierbaren Steuerungscode umsetzt. Ferner konnte gezeigt werden, dass Steuerungscode nach IEC 61131-3 für drei verschiedene Laboranlagen sowie mittels zwei unterschiedlichen Programmierwerkzeugen implementiert und durch entsprechende Speicherprogrammierbare Steuerungen ausgeführt werden kann. Auch eine automatische Generierung des entsprechenden Steuerungscode ist durch die Integration des HiTraP-AT Planungsverfahrens mit der UML-OP von Daniel Schütz möglich. Dabei kann sowohl IEC 61131-3 konformer Steuerungscode generiert werden, welcher exakt einen zuvor gegebenen Handlungsplan realisiert als auch Steuerungsprogramme, welche ohne Änderung des Quellcodes unterschiedliche Handlungspläne ausführen können. Zusammenfassend konnte, wie im Rahmen dieses Abschnitts beschrieben, positiv evaluiert werden, dass die in Abschnitt 4.3 vorgeschlagenen Konzepte genutzt werden können, um das HiTraP-AT Planungsverfahren mit Steuerungssoftware entsprechend dem IEC 61131-3 Standards anzubinden und dabei entsprechende Aspekte der beiden in Abschnitt 1.1 beschriebenen Anwendungsfälle zu adressieren. Demzufolge kann das HiTraP-AT Planungsverfahren für die Planung von Ablaufreihenfolgen der Softwarefunktionen von IEC 61131-3 basierter Steuerungssoftware eingesetzt werden, weshalb Anforderung A4 adressiert ist.

5.4. Evaluation der Anforderungen A1 und A2

In den vorangegangenen Abschnitten wurden Anforderungen hinsichtlich des Zeit- und Skalierbarkeitsverhaltens des HiTraP-AT Planungsalgorithmus ebenso untersucht wie Möglichkeit der Integration des vorgeschlagenen Planungsverfahrens mit modellgetriebenen Entwicklungsansätzen. Auch die Anwendbarkeit des HiTraP-AT Planungsverfahrens für den aktuell am weitesten verbreiteten Standard zur Implementierung von Steuerungssoftware auf der Feldebene des Maschinen- und Anlagenbaus, der IEC 61131-3 wurde evaluiert. Neben diesen Anforderungen wurden in Abschnitt 2.2 weitere Anforderungen an den Lösungsansatz gestellt: Zum einen hinsichtlich der Berücksichtigung unterschiedlicher, funktionaler Aspekte innerhalb eines Handlungsplanes (Anforderung A1) und zum anderen hinsichtlich der Bestimmung eines optimalen Handlungsplanes (Anforderung A2). Die Erfüllung dieser Anforderungen durch das vorgeschlagene HiTraP-AT Planungsverfahren werden im Rahmen dieses Abschnitts betrachtet.

Eine der konzeptionell grundlegenden Anforderungen an HiTraP-AT stellt Anforderung A1 dar: ein valider Handlungsplan für die Steuerungsebene für die Automatisierung des Maschinen- und Anlagenbaus umfasst eine Menge von Teilplänen, die nicht sequentiell sondern spezifisch (zyklisch) angeordnet sind, um die in dieser Arbeit betrachteten Teilfunktionen einer Steuerungssoftware – Anlauf zur Grundstellung (Anforderung A1.3), Fertigung eines Werkstücks (Anforderung A1.1), Rückkehr zur Grundstellung (Anforderung A1.2) – geeignet zu repräsentieren. Die spezifische Struktur des HiTraP-AT Modells spiegelt dabei diese gestellten Anforderungen an ein

modellbasiertes Planungsverfahren wieder. Die Fertigung von Stückgut und die damit verbundene Berücksichtigung des Produktionsprozesses innerhalb der Planung (Anforderung A1.1) wird insbesondere mittels des Zustandsraumes "Produktion" sowie des Zustandsraumes des Produktes adressiert. Um die Anforderung hinsichtlich der automatischen Planung für massenfertigende, industrielle Automatisierungssysteme zu adressieren, findet innerhalb des Zustandsraumes "Rücklauf" die notwendige Ausführung weiterer, nicht-produktiver Funktionalitäten des Systems Berücksichtigung (Anforderung A1.2). Ferner wird durch HiTraP-AT, wie dies in Anforderung A1.3 gefordert wird, der Startzustand sowie etwaige, notwendige Ausführung von Funktionalitäten berücksichtigt, um eine Maschine bzw. Anlage in die Lage zu versetzen, Produkte (ausgehend von einer ebenfalls zu bestimmenden Grundstellung) geeignet zu fertigen. Dies ist innerhalb des HiTraP-AT Modells durch Integration des Zustandsraumes "Anlauf" erfolgt. Zusammenfassend kann von einer grundsätzlichen Erfüllung der in Anforderung A1 gestellten Eigenschaften gesprochen werden.

Durch das zusätzliche Konzept der (konditionalen) Hierarchisierung, welches insbesondere für die Erreichung eines geeigneten Zeit- und Skalierbarkeitsverhalten des Planungsalgorithmus eingeführt wurde, können entsprechende Teilpläne "Anlauf", "Produktion" und "Rückführung" für jedes Modul einer Anlage effizient bestimmt werden. Wie in Abschnitt 4.1.3.3 lassen sich aus diesen modulspezifischen Teilplänen wieder nicht-hierarchische Handlungspläne generieren, welche die im Rahmen der Anforderungen beschriebene funktionale Trennung der Teilpläne exakt widerspiegelt. Im Rahmen der Arbeiten hinsichtlich der Integration des Planungsverfahrens mit der Steuerungsebene des Maschinen- und Anlagenbaus und die damit verbundene Abbildung des Handlungsplanes für IEC 61131-3 basierte Steuerungen (vgl. Abschnitt 5.3) konnte ferner identifiziert werden, dass mittels dieser Teilpläne die grundlegende Funktionalität von Maschinen und Anlagen geeignet realisiert werden kann.

Neben den zuvor diskutierten Anforderungen wurde in Abschnitt 2.2 identifiziert, dass für ein geeignetes Planungsverfahren zur Bestimmung von Ablaufreihenfolgen der Steuerungsebene des Maschinen- und Anlagenbau notwendig ist, einen guten bzw. optimalen Handlungsplan zu bestimmen (Anforderung A2). Diese Anforderung wurde im Rahmen des Konzeptes durch die Reformulierung des HiTraP-AT Planungsproblems als lineares Optimierungsproblem adressiert. Durch entsprechend geeignete Gewichtungen von Aktionen ist es hierdurch möglich, nicht nur einen korrekten, sondern wie gefordert, einen optimalen Plan unter Berücksichtigung der Gewichtungen von Aktionen zu bestimmen. Neben der allgemeinen Möglichkeit, optimale Handlungspläne zu bestimmen, wurde im Rahmen von Anforderung A2 ferner gefordert, dass (wie für die Massenproduktion üblich) ein den Durchsatz maximierender Handlungsplan bestimmt werden kann. Im Zuge der Untersuchungen hinsichtlich der Integration mit modellgetriebenen Entwicklungsansätzen und einem angenommenen Ausfall einer Komponente zur Laufzeit wurden eine Vielzahl unterschiedlicher Anlagenmodelle einer automatischen Handlungsplanung unterzogen. Wie bereits in Abschnitt 5.1 beschrieben, wurde hierbei stets ein erwarteter (optimaler) Handlungsplan durch das HiTraP-AT Planungsverfahren bestimmt. Die automatisch abgeleiteten Handlungspläne enthielten dabei insbesondere keine Aktionen oder Aktionsfolgen, die für eine korrekte Realisierung des technischen Prozesses eingespart hätten werden können. Durch die charakteristische, partielle Ordnung von HiTraP-AT Handlungspläne ergibt sich ferner die Möglichkeit der Parallelisierung, welche sich zusätzlich positiv auf die Güte des Handlungspla-

5. Untersuchung der Anwendbarkeit und Evaluation von HiTraP-AT

nes auswirken kann. Insofern wurde Anforderung A2 hinsichtlich der Bestimmung eines guten Handlungsplanes erfüllt.

HiTraP-AT betrachtet jedoch ausschließlich die Fertigung eines einzelnen Produktes. Zwar werden relevante Aspekte hinsichtlich der Güte eines einzelnen Handlungsplanes im Bezug auf dessen mehrfache Ausführung für die Fertigung großer Stückzahlen berücksichtigt (siehe hierzu Abschnitt 4.1.3.1), eine Optimierung der Ausführung mehrerer, paralleler Handlungspläne erfolgt jedoch nicht. Demzufolge bleibt diese Herausforderung durch das im Rahmen dieser Arbeit vorgeschlagene HiTraP-AT Planungsverfahren offen, wurde jedoch durch Anforderung A2 auch nicht explizit gefordert.

Auf Grund der zuvor beschriebenen Sachverhalte und Untersuchungsergebnisse kann zusammengefasst werden, dass sowohl Anforderung A1 wie auch Anforderung A2 durch das HiTraP-AT Planungsverfahren geeignet adressiert wird.

5.5. Zusammenfassung der Untersuchungsergebnisse

In den vorangegangenen Abschnitten in diesem Kapitel wurde die Erfüllung der gestellten Anforderungen des im Rahmen dieser Dissertation vorgeschlagenen Ansatzes untersucht. Hierbei wurde eine Analyse der Leistungsfähigkeit im Hinblick auf das zeitliche Verhalten und der Skalierbarkeit der Implementierung des vorgeschlagenen HiTraP-AT Planungsverfahrens vorgenommen. Ferner wurde die im Rahmen dieser Dissertation vorgeschlagene Integration von HiTraP-AT in einen modellgetriebenen Entwicklungsansatz sowie die Anbindung an die Steuerungssoftware nach IEC 61131-3 anhand unterschiedlicher Fallstudien evaluiert.

Im Rahmen der Anforderungsermittlung wurden spezifische Charakteristika von Ablaufplänen identifiziert, welche die Notwendigkeit verschiedener Teilfunktionen der Steuerungssoftware für die Maschinen- und Anlagenautomatisierung widerspiegeln (Anforderung A1). Diese Anforderung wird in dem in dieser Arbeit vorgeschlagenen Konzept maßgeblich durch die Struktur des internen HiTraP-AT Planungsmodells adressiert. Im Rahmen der Anwendung des HiTraP-AT Planungsverfahrens für unterschiedliche Laboranlagen zeigte sich, dass einerseits die gestellte Anforderung die Charakteristika der Steuerungsebene geeignet reflektiert und andererseits, dass die durch das im Rahmen dieser Arbeit vorgeschlagene Planungskonzept geeignete Ablaufpläne für diese Laboranlagen bestimmt werden. Im Zusammenspiel mit der vorgeschlagenen Integration von HiTraP-AT mit SPS-basierter Steuerungssoftware konnten automatisch identifizierte Abläufe genutzt werden, um korrekte technische Prozesse dieser Laboranlagen zu realisieren. Im Hinblick auf die Erfüllung von Anforderung A1 konnte im Rahmen der durchgeführten Untersuchungen letztendlich gezeigt werden, dass das Konzept für strikt modulare Laboranlagen erfolgreich Anwendung finden kann. Eine Anwendung des vorgeschlagenen Verfahrens darüber hinaus konnte nicht gezeigt werden.

Im Rahmen der Anwendung des HiTraP-AT Planungsverfahrens für unterschiedliche Laboranlagen wurde ebenfalls die Güte der durch HiTraP-AT bestimmten Handlungspläne untersucht, um die Erfüllung von Anforderung A2 zu beurteilen, wie dies in Abschnitt 5.4 zusammengefasst wurde. Dabei konnte identifiziert werden, dass durch HiTraP-AT einerseits die gestellte Anforderung A2 adressiert wurde, so dass von einer Erfüllung der Anforderung gesprochen werden kann, andererseits durch HiTraP-AT jedoch ausschließlich die Güte eines einzelnen Handlungsplanes adressiert wird. Eine vollständige Optimierung der Handlungspläne mehrerer (potentiell paralle-

ler) Produkte innerhalb der Anlage erfolgt nicht. Dementsprechend kann durch HiTraP-AT nicht von einem optimalen Handlungsplan im Hinblick auf die gesamten, zu fertigenden Produktionsaufträge gesprochen werden.

Im Rahmen der Evaluation des HiTraP-AT Planungsverfahrens wurde eine intensive Untersuchung der Leistungsfähigkeit hinsichtlich dem zeitliche Verhalten sowie der Skalierbarkeit entsprechend Anforderung A3 durchgeführt. Wie bereits die Analyse verwandter Arbeiten in Kapitel 3 zeigte, existiert kein Verfahren, welches die spezifische Problemformulierung der Steuerungsebene für die Maschinen- und Anlagenautomatisierung (insbesondere bedingt durch Anforderung A1) adressieren kann. Auch sind Informationen zur Leistungsfähigkeit dieser Ansätze kaum verfügbar. Auf Grund der fehlenden Vergleichsmöglichkeiten mit anderen Planungsansätzen wurde daher in Abschnitt 5.2 evaluiert, wie sich unterschiedliche konzeptionelle Teile des HiTraP-AT Planungsverfahrens auf die Laufzeit und Skalierbarkeit des Verfahrens im Hinblick auf die Erfüllung der Anforderung A3 auswirken. Hierbei zeigte sich, dass das HiTraP-AT Planungsverfahren im Hinblick auf das zeitliche Verhalten die gestellte Anforderung vollständig erfüllt. Auf Grund der guten Skalierbarkeitseigenschaft des Planungsverfahrens erfüllt das in dieser Arbeit vorgeschlagene HiTraP-AT Planungsverfahren die gestellte Anforderung A3.

In Abschnitt 5.1 wurde die vorgeschlagene Integration des HiTraP-AT Planungsverfahrens mit der von Daniel Schütz entwickelten UML-OP [Sch14, VHSFL14b] – einen modellgetriebenen Ansatz für Entwicklung von Steuerungssoftware für die Maschinen- und Anlagenautomatisierung – im Hinblick auf die Erfüllung von Anforderung A5 hin untersucht. Dabei wurde die UML-OP für die Beschreibung zweier Laboranlagen des Lehrstuhls AIS angewandt und auf Basis der in diesen Modellen enthaltenen Informationen eine Planung mittels HiTraP-AT durchgeführt. Hierdurch konnte positiv evaluiert werden, dass die in Abschnitt 4.2.2 vorgeschlagene Abbildung zwischen UML-OP und HiTraP-AT Planungsproblem genutzt werden kann, um eine automatische Bestimmung der Ablaufreihenfolgen der in UML-OP definierten Operationen durchzuführen. Demzufolge ist eine Integration entsprechend Anforderung A5 des in dieser Arbeit vorgeschlagenen HiTraP-AT Planungsverfahrens mit einem modellgetriebenen Entwicklungsansatz für den Maschinen- und Anlagenbau adressiert.

Die in Abschnitt 4.3 vorgeschlagenen Möglichkeiten der Anbindung des HiTraP-AT Planungsmodells und deren konkrete Umsetzung mittels Programmiersprachen sowie Programmierkonzepten entsprechend IEC 61131-3 Standards zur Adressierung der entsprechenden Anforderung A4 wurde im Rahmen eigener Vorarbeiten sowie unterschiedlicher studentischer Arbeiten für verschiedene fertigungstechnische Demonstrationsanlagen des Lehrstuhls AIS evaluiert (vgl. Abschnitt 5.3). Die nahtlosen Integration des HiTraP-AT Planungsverfahrens in einen modellgetriebenen Entwicklungsprozess für Maschinen bzw. Anlagen und dessen positive Evaluation stellt die Grundlage für die Erfüllung der gestellten Anforderung A4 nach der Anwendbarkeit des Planungsansatzes für IEC 61131-3 konforme Speicherprogrammierbare Steuerungen dar. Dabei konnte für unterschiedliche Konfigurationen⁶³ von drei verschiedenen Laboranlagen insbesondere gezeigt werden, dass die durch das HiTraP-AT Planungsverfahren automatisch bestimmten Ablaufreihenfolgen korrekte technische Prozesse für die Fertigung eines gewünschten Produktes realisieren. Unter den bereits im Kontext von Anforderung A2 beschriebenen Prämissen konnte dadurch auf eine Erfüllung von Anforderung A4 durch das vorgeschlagene HiTraP-AT Planungsver-

⁶³Unter Konfiguration wird hier unterschiedliche Anordnung bzw. Nutzung von Modulen verstanden.

5. Untersuchung der Anwendbarkeit und Evaluation von HiTraP-AT

fahren und dessen Integrationskonzept mit IEC 61131-3 basierte Steuerungssoftware geschlossen werden.

Zusammenfassend lässt sich von einer grundsätzlichen Adressierung sowie weitestgehenden Erfüllung aller gestellten Anforderungen durch das in dieser Arbeit vorgeschlagene HiTraP-AT Planungsverfahren sprechen. Im Rahmen der durchgeführten Untersuchungen und damit verbundenen Anwendung des Konzeptes an unterschiedlichen Fallstudien konnten dabei bereits auch Grenzen des Konzeptes sowie der Anforderungserfüllung identifiziert werden. Durch den identifizierten Erfüllungsgrad der Anforderungen des in dieser Arbeit vorgeschlagenen Konzeptes konnte eine grundsätzliche Anwendbarkeit für fertigungstechnische, modulare Anlagen gezeigt werden, welche in der industriellen Praxis mehr und mehr eingesetzt werden. Auf Basis der Erfahrungen bei der Anwendung des vorgeschlagenen Konzeptes konnten ferner einige Grenzen des Ansatzes – insbesondere im Hinblick auf die Anwendbarkeit das Konzept in der industriellen Praxis identifiziert werden. Diese werden im nachfolgenden Abschnitt beschrieben.

5.6. Grenzen und Erweiterungspotentiale des Ansatzes

Im vorangegangenen Abschnitt wurden bereits einige Grenzen des Ansatzes im Hinblick auf die Erfüllung der gestellten Anforderung angemerkt. Darüber hinaus wurden im Rahmen der durchgeführten Untersuchungen zahlreiche über die gestellten Anforderungen hinausgehende Grenzen und potentiellen Erweiterungsmöglichkeiten identifiziert, welche im Folgenden beschrieben werden. Die durchgeführte Evaluation des in dieser Arbeit vorgeschlagenen Ansatzes erfolgte dabei ausschließlich auf Basis von fertigungstechnischen Demonstratoren mit Laborcharakter. Fokus der Betrachtungen in diesem Kapitel sind daher insbesondere auch Grenzen des Ansatzes im Hinblick auf einen Einsatz in der industriellen Praxis für reale Fertigungssysteme sowie weiteren Einsatzszenarien.

Einschränkungen durch das gewählte Aktionsmodell

Im Rahmen dieser Arbeit werden ausschließlich vollständig diskrete, technische Prozesse (wie in Abschnitt 2.1.2 beschrieben) zu Grunde gelegt. Eine klare Unterscheidung der Prozesse ist jedoch nicht immer möglich [LG99]. Dies spiegelt sich beispielsweise innerhalb der genutzten Systemfallstudien "Hybrides Prozessmodell" bei der Modellbildung für Transportbänder und deren Funktion "Transport" wieder. Grund hierfür ist, dass sich der Effekt der Bewegung des Bandes (beispielsweise bedingt durch den Start eines Antriebsmotors) konstant solange auf ein auf dem Band befindliches Werkstück auswirkt, bis das entsprechende Band wieder gestoppt wird (oder das Produkt das Förderband verlässt). Entsprechende Funktionalität kann durch das in dieser Arbeit genutzte Aktionsmodell – d.h. die Beschreibung diskreter Automatisierungsfunktionen ausschließlich anhand von Vor- und Nachbedingungen – prinzipiell derart abgebildet werden, so dass korrekte Handlungspläne abgeleitet werden können (siehe hierzu beispielsweise das in Abschnitt 5.1 beschriebene UML-OP Modell der Fallstudie "Hybriden Prozessmodells" und dessen Anwendung für eine automatische Handlungsplanung mittels HiTraP). Hierbei ist jedoch die Definition von Aktionen notwendig, welche lediglich für die Korrektheit der Planung notwendig sind, aber keine spezifische Funktionalität aus Sicht der Steuerungstechnik darstellen, wie dies für die Aktion "transportieren" eines Förderbandes der Fall ist: Nach Ausführung für den Start des

Antriebsmotors des Förderbandes wird die Aktion "transport" dieser Komponente ausgeführt, sobald sich ein Werkstück auf dem Band befindet. Andere Aktionen, wie z.B. die Drehung einer rotatorischen Weiche oder das Ausschieben eines Werkstücks durch einen pneumatischen Zylinder werden im Gegensatz dazu erst ausgeführt, falls spezifische Aktorstellwerte innerhalb der Implementierung der Aktion nach Beginn der Aktionsausführung gesetzt werden. Diese Modellbildung ist aktuell auf die spezifischen Aspekte des HiTraP-AT Planungsverfahrens ausgerichtet. Das Aktionsmodell von HiTraP-AT bietet dabei keine Möglichkeiten der Formulierung von Wirkzusammenhängen, wie sie hierfür potentiell benötigt würden. Eine mögliche Lösung für diese Herausforderung wäre eine Anpassung des Aktionsmodells, wie sie beispielsweise durch Castillo et. al [CFOG00b, CFOG00a] vorgeschlagen wird. Hierzu wäre eine geeignete Erweiterung des Planungsproblems notwendig ebenso wie eine entsprechende Anpassung des internen HiTraP-AT Planungsmodells und dessen Formulierung als Lineares Programm.

Eine weitere, sich aus dem verwendeten Aktionsmodell resultierende Einschränkung, ergibt sich im Hinblick auf die Selektion eines optimalen Handlungsplanes. Im Rahmen der Evaluation konnte gezeigt werden, dass im Hinblick auf die gestellte Anforderung A2, durch HiTraP-AT die Bestimmung eines guten Handlungsplanes im Hinblick auf den Durchsatz von Maschinen und Anlagen erfolgt. Hierbei wird die Formulierung des Optimierungskriteriums im Wesentlichen auf Basis der Zeit- und Kostenbewertung von Aktionen, welche für die Fertigung eines Werkstücks benötigt werden (vgl. Abschnitt 4.1.3.1) zu Grunde gelegt. Von Bedeutung ist hierbei lediglich, ob die Auswahl und Ausführung von Aktionen zum Starten bzw. Stoppen eines Förderbandes einen negativen Einfluss auf das Optimierungskriterium hätte. Im Gegensatz dazu werden für eine Berechnung energieeffizienter Handlungspläne Informationen darüber benötigt, dass Aktionen fortwährend ausgeführt werden (wie z.B. der fortwährende Antrieb eines Förderbandes). Diese Informationen sind in dem durch HiTraP-AT verwendeten Aktionsmodell nicht enthalten. Auch ist in diesem Zusammenhang das durch die Definition des HiTraP-AT Planungsproblems zu Grunde gelegte Zeitmodell für die Ausführung von Aktionen nicht geeignet. In HiTraP-AT kann die Zeit (wie für die Berechnung durchsatzoptimierter Handlungspläne in dieser Arbeit genutzt) in Form konstanter Kosten der Aktionsausführung berücksichtigt werden. Eine Berücksichtigung der Länge der Ausführung einer Aktion und entstehende Formulierung eines Optimierungsproblems unter Berücksichtigung variablen Kosten in Abhängigkeit der Länge der Ausführung einer Aktion sind durch das in dieser Arbeit beschriebene Konzept nicht möglich.

Einschränkungen durch die Beschreibung von Anforderungen an den zu realisierenden, technischen Prozesses

Im Rahmen der durchgeführten Evaluation des HiTraP-AT Planungsverfahrens und dessen Anwendung für unterschiedliche, fertigungstechnische Laboranlagen des Lehrstuhls AIS zeigte sich, dass die vergleichsweise einfache Beschreibung von Anforderungen an den zu realisierende technische Prozess auf Basis einer Definition von Start- und Zielzustand des Produktes für die hier genutzten Systemfallstudien ausreichend ist. In manchen Fällen kann es jedoch notwendig sein, zusätzliche Anforderungen an den zu realisierenden Prozess zu stellen, wie z.B. die Ausführung einer spezifischen Aktion oder gar die Ausführung einer spezifischen Sequenz von Aktionen. Dies wird aktuell weder durch das HiTraP-AT Planungsproblem noch durch die UML-OP explizit unterstützt. Die Auswahl spezifischer Aktionen und deren Nutzung innerhalb eines Handlungs-

5. Untersuchung der Anwendbarkeit und Evaluation von HiTraP-AT

planes durch HiTraP-AT sowie ggf. auch spezifische Reihenfolgen zwischen Aktionen kann jedoch durch die Nutzung zusätzlicher, spezifischer Variablen innerhalb des Planungsproblems erreicht werden. Um die Auswahl einer gewünschten Aktion zu forcieren, ist das Hinzufügen einer einzelnen Variablen des Produktes in das Planungsproblem ausreichend, wobei ausschließlich die Aktion, welche ausgewählt werden soll, den Zustand dieser Variable verändert. Der veränderte Wert dieser Variablen wird ferner in die Definition des Zielzustandes des Produktes integriert. Eine Sequenzierung von Aktionen kann auf analoge Weise erfolgen. Es hat sich jedoch gezeigt, dass hierdurch die Erstellung des Planungsproblems sowie eines entsprechenden UML-OP Modells komplexer und fehleranfälliger wird. Eine spezifische Erweiterung der UML-OP zur vereinfachten Spezifikation solcher Abhängigkeiten und eine damit verbundene Erweiterung der Abbildung zwischen UML-OP und HiTraP-AT würde diese Umgehungslösung obsolet machen. Obwohl innerhalb der Formulierung des internen HiTraP-AT Planungsmodells keine Berücksichtigung von Zwischenzielen erfolgt, können entsprechende, zusätzliche Anforderungen an den technischen Prozess bei der Planung berücksichtigt werden. Weitere, darüber hinausgehende Spezifikation von Anforderung an den zu realisierenden, technischen Prozess werden durch das in dieser Arbeit beschriebene HiTraP-AT Planungsverfahren nicht unterstützt.

Während der Anwendung des HiTraP-AT Planungsverfahrens für die Systemfallstudien zeigte sich beispielsweise, dass die in dieser Arbeit zu Grunde gelegte Annahme eines stets völlig identischen technischen Prozesses, welcher für jedes einzelne Produkt im Falle einer Massenfertigung durch eine Maschine und Anlage durchgeführt wird, sehr strikt ist. Trotz Massenfertigung eines einzelnen Produktes kann es zu notwendigen Fallunterscheidungen innerhalb des Produktionsprozesses kommen, um beispielsweise im Anschluss an eine Qualitätskontrolle Werkstücke entsprechend des Ergebnisses unterschiedlich zu bearbeiten. Diese Aspekte können mittels HiTraP-AT zwar (zum Teil) dadurch adressiert werden, dass mehrere separate Planungen durchgeführt werden (z.B. ein Handlungsplan mit Zielsetzung Qualitätsprüfung, zwei weitere Handlungspläne deren Startzustand jeweils ein binäres Ergebnis der Qualitätskontrolle darstellt). Die Möglichkeit solch notwendige, prozessbedingte Anforderungen an Fallunterscheidungen direkt innerhalb des Planungsproblems zu spezifizieren und durch mittels HiTraP-AT berücksichtigen zu können, würde dabei eine einzelne, vollständige Planung und Optimierung der Steuerungsabläufe ermöglichen. Ferner wäre eine integrierte Betrachtung dieser Aspekte für den Anwender deutlich benutzerfreundlicher. Dies wird durch das aktuelle Konzept nicht unterstützt.

Einschränkungen durch das interne HiTraP-AT Planungsmodell

Neben den zuvor bereits beschriebenen Einschränkungen des in dieser Arbeit vorgeschlagenen HiTraP-AT Planungsverfahrens ergeben sich weitere Einschränkungen durch das spezifische interne HiTraP-AT Planungsmodell sowie dessen in dieser Arbeit vorgeschlagene Formulierung in Form eines Linearen Programms.

Die in dieser Arbeit vorgeschlagenen Formulierung und Lösung des Planungsproblems als lineares Optimierungsproblem basiert auf der Prämisse, dass jeder der zusammenhängenden Zustandsräume "Anlauf", "Produktion" und "Rücklauf" exakt einmal innerhalb eines HiTraP-AT Handlungsplanes durchlaufen wird (wie durch die Bedingungen in Abschnitt 4.1.3.2 formuliert). Bedingt hierdurch, können von HiTraP-AT ausschließlich Handlungspläne bestimmt werden, in denen alle produktiven Aktionen, die durch ein Modul ausgeführt werden müssen, chronologisch

vor der Ausführung einer Finalisierungs- bzw. Übergabeaktion angeordnet sind – d.h. bevor ein Werkstück ein Modul wieder verlässt. Dabei können durch das interne HiTraP-AT Planungsmodell grundsätzlich zwar entsprechend flexible Materialflüsse abgebildet werden und finden auch bei der Bestimmung des Handlungsplanes in Form alternativer Reihenfolgen von Modulen Berücksichtigung, identifizierte Handlungspläne weisen jedoch im Hinblick auf den Materialfluss stets eine lineare Charakteristik auf. Das heißt insbesondere auch, dass ein Werkstück durch ein Modul nicht mehrfach bearbeitet werden, sofern zwischenzeitlich andere Module zur Bearbeitung notwendig sind. Während dies für eine Linienfertigung keine Einschränkungen darstellt, können für andere Fertigungssysteme mit größerer Flexibilität im Hinblick auf mögliche Materialflüsse, die sich durch HiTraP-AT ergebende Flexibilität nicht genutzt werden.

Wie bereits zuvor beschrieben, ergeben sich durch das in dieser Arbeit zu Grunde gelegte Aktionsmodell Einschränkungen im Hinblick auf die Bestimmung optimaler Handlungspläne, falls Kosten der Aktionsausführung von der Dauer der Ausführung einer Aktion abhängig sind. Durch das vorgeschlagene, interne Planungsmodell ergeben sich diesbezüglich weitere Einschränkungen im Bezug auf die Bestimmung durchsatzoptimierter Handlungspläne: Während das in dieser Arbeit vorgeschlagene Konzept zwar den Umlaufbestand und damit auch den zeitlichen Abstand zwischen einzelnen Produkten berücksichtigt, ist der maximal mögliche Umlaufbestand durch die Prämisse des Konzeptes eingeschränkt, dass lediglich ein einzelnes Produkt bzw. Werkstück durch ein Modul bearbeitet wird. So ist zwar Anforderung A2 durch das zu Grunde gelegte Aktionsmodell erfüllt, darüber hinaus existiert jedoch Verbesserungspotential. So ist es denkbar, dass ein Modul mehrere Produkte parallel bearbeiten kann, wie z.B. im Falle eines Bandförderers, durch den der Transport mehrerer Produkte gleichzeitig in die selbe Richtung möglich wäre. Um diesen Sachverhalt innerhalb des HiTraP-AT Planungsverfahrens zu adressieren, wäre eine Erweiterung der Unabhängigkeitsnotation von Aktionen notwendig, um die parallel Ausführung von Aktionen an unterschiedlichen Produkten durch ein Modul zu berücksichtigen. Dies bedingt dabei auch eine vollständige Überarbeitung des internen Planungsmodells und somit des Gesamtkonzeptes.

5.7. Bewertung des Ansatzes im Kontext der Anwendungsbeispiele

Im vorangegangenen Abschnitt wurde der in dieser Arbeit vorgeschlagene HiTraP-AT Ansatz im Hinblick auf die gestellten Anforderungen untersucht. Im Rahmen dieses Abschnitts werden nun Vorteile und Möglichkeiten der Anwendung des HiTraP-AT Verfahrens im Kontext der eingangs in dieser Arbeit beschriebenen und für die Ableitung der Anforderung genutzten Anwendungsbeispiele beschrieben. Hierdurch soll untersucht werden, was durch die Anwendung der im Rahmen dieser Arbeit beschriebenen Konzepte und Entwicklungen im Kontext der Anwendungsbeispiele erreicht werden konnte und welche Potentiale sich hieraus ergeben. Dementsprechend werden in den nachfolgenden Abschnitten 5.7.1 sowie 5.7.2 die dieser Arbeit zu Grunde gelegten Anwendungsbeispiele betrachtet, bevor in Abschnitt 5.7.3 eine Zusammenfassung gegeben wird.

5.7.1. Anwendungsbeispiel 1: Konfigurationsassistenz

Ausgangspunkt dieses Anwendungsszenarios stellt, wie in Abschnitt 1.1.1 beschrieben, die Herausforderung dar, die Veränderbarkeit von Fertigungssystemen im Kontext von Industrie 4.0

5. Untersuchung der Anwendbarkeit und Evaluation von HiTraP-AT

zunehmend sicherstellen zu müssen. Kernidee ist dabei, diese Herausforderung zu adressieren, in dem modulare Anlagen eingesetzt werden, deren Module (d.h. im Kontext von Industrie 4.0 cyber-physische Systeme) schnell und einfach ausgetauscht werden können sowie die Anordnung dieser Module (in Abhängigkeit verfügbarer Materialflussschnittstellen der Module) angepasst werden kann [ABD⁺16]. Angenommen werden im Kontext dieser Arbeit funktional abgeschlossene Fertigungsmodule, wie sie z.B. in [Zue10, BHH⁺16, ReC16] beschrieben werden und als einzelne Maschinen mit geeigneter Materialflussschnittstelle verstanden werden können. Zielsetzung im Kontext dieser Arbeit ist dabei die Assistenz bei der Entwicklung neuer, kundenspezifischer Konfiguration der Steuerungssoftware, um Veränderungen hinsichtlich genutzter Module und deren Anordnung steuerungstechnisch geeignet zu adressieren. Dabei liegt die Annahme zu Grunde, dass entsprechende Fertigungsmodule ebenso bereits existieren wie die Implementierung einzelner Steuerungsfunktionen. Die Ausführung der Automatisierungsfunktionen kann dabei dann im Feld zentral – wie in einer klassischen, hierarchischen Steuerungsarchitektur – oder dezentral verteilt auf unterschiedliche Steuerungen – entsprechend der Vision von Industrie 4.0 – erfolgen. Eine autonome oder verteilte Entscheidungsfindung einzelner cyber-physischer Systeme wird hierbei nicht betrachtet.

Eine prototypische Umsetzung eines Konfigurationsassistenten wurde in Zusammenarbeit mit Daniel Schütz et al. entwickelt und umgesetzt [LSVH13, LSF⁺13]. Dabei wird ein modellgetriebener Ansatz verfolgt: Für die Beschreibung der Komponenten und deren Automatisierungsfunktionen wird das von Daniel Schütz vorgeschlagene SysML-AT Profil genutzt; die Generierung von Vorschlägen für Ablaufreihenfolgen der Automatisierungsfunktionen erfolgt durch das in dieser Arbeit beschriebene HiTraP-AT Planungsverfahren. Mittels des im Rahmen dieser Arbeit vorgeschlagenen Konzepts für die Umsetzung auf IEC 61131-3 basierten Steuerungen (siehe Abschnitt 4.3) kann ferner eine Konfiguration des Softwareprojekts mittels der identifizierten Ablaufreihenfolgen erfolgen. Existierende Ansätze sehen hierfür eine manuelle Spezifikation der auszuführenden Prozesse bzw. deren Realisierung mittels der Spezifikation geeigneter Abläufe von Automatisierungsfunktionen vor. Durch das in dieser Arbeit vorgeschlagene Verfahren zur automatischen Identifikation von Ablaufreihenfolgen und dessen Integration in zuvor beschriebenen Prototypen für eine modellgetriebene Softwareentwicklung für die Maschinen- und Anlagenautomatisierung können nun Vorschläge hinsichtlich der Reihenfolge von Automatisierungsfunktionen unterbreitet werden. Diese Vorschläge können entweder direkt übernommen oder auf Wunsch manuell angepasst werden. Hierdurch ergibt sich als sofortige Konsequenz eine Assistenzfunktion für den Steuerungsentwickler.

Während diese Arbeit der Prämisse unterliegt, dass geeignete fertigungstechnische Module bereits entwickelt, für eine spezifische Anlage ausgewählt und angeordnet wurden, existieren darüber hinaus eine Vielzahl weitführende Herausforderungen bei der Entwicklung cyber-physischer Systeme [HVHB⁺16]. Neben der Assistenz für die Identifikation des durchzuführenden Prozesses, kann das HiTraP-AT Verfahren, wie im Kontext von Anwendungsbeispiel 2 nachfolgend detailliert beschrieben, für die Bestimmung von alternativen Abläufen im Falle von Komponentenausfällen genutzt werden. Denkbar ist, diese Möglichkeit bereits im Engineering von Maschinen und Anlagen für eine Analyse von alternativem Verhalten von Komponenten zu nutzen und hierdurch bei der Durchführung klassischer Verfahren wie Fehlermöglichkeits- und –influssanalysen zu assistieren.

Um eine breite Anwendung des Konzept in der industriellen Praxis zu ermöglichen, ist eine Erweiterung des Konzeptes und Adressierung der in Abschnitt 5.6 beschriebenen Einschränkungen notwendig. Insbesondere bedarf es einer Erweiterung der durch HiTraP-AT unterstützten, technischen Prozesse und damit verbunden eine erweiterte Formulierung komplexerer Anforderungen an den technischen Prozess, welcher über eine Definition des gewünschten Zielzustandes des Produktes hinausgeht.

Zusammenfassend kann die Integration des in dieser Arbeit vorgeschlagen HiTraP-AT Verfahrens und dessen exemplarische Integration in ein modellgetriebenes Entwicklungswerkzeug als erster Schritt hin zu weiterführenden Assistenzfunktionen für die Entwicklung cyber-physischer Systeme betrachtet werden. Im Gegensatz zu klassischen Ansätzen, welche eine explizite Definition des durchzuführenden technischen Prozesses zu Grunde legen und damit eine Spezifikation fordern, „was eine Maschine tun soll“, stellt die in dieser Arbeit zu Grunde gelegte Beschreibung „was eine Maschine tun kann“ einen wesentlichen Schritt im Kontext von Industrie 4.0 dar, um die geforderte Flexibilität hinsichtlich unterschiedlicher, technischer Prozesse (z.B. für eine auftragsbezogene Fertigung) zu ermöglichen. Die Verfügbarkeit eines solchen Modells sowie eines geeigneten Verarbeitungsmechanismus – dem HiTraP-AT Planungsverfahren – ermöglicht erstmals eine automatisierte Identifikation der Abläufen von Automatisierungsfunktionen, d.h. Softwarefunktionen, welche auf Speicherprogrammierbaren Steuerungen nach IEC 61131-3 ausgeführt werden können.

5.7.2. Anwendungsbeispiel 2: Dynamische Rekonfiguration zur Störkompensation

Anwendungsbeispiel 2 ist im Bereich des Alarm- und Störmanagements angesiedelt und fokussiert auf die Steigerung der Verfügbarkeit automatisierter Fertigungsanlagen. Im Kontext von Industrie 4.0 wird hierbei häufig von autonomer Fehlerkompensation oder Selbstheilung der Maschinen und Anlagen gesprochen. Ausgangspunkt dieser Arbeit war dabei die Fragestellung, ob ein geeignetes Konzept für Automatisierungsfunktionen und der Einsatz eines geeigneten Verfahrens zu Bestimmung von Ablaufreihenfolgen dieser Automantisierungsfunktionen einen Beitrag zur Verbesserungen der Verfügbarkeit von Maschinen und Anlagen leisten kann. Während bereits vielversprechende Arbeiten existieren, um beispielsweise Sensorstörungen geeignet adressieren und kompensieren zu können (siehe unter anderem [Sch14, SWLVH13]), stellen Störungen von Aktoren nach wie vor eine Herausforderung dar. Mögliche Störungen und deren Folgen werden zwar durch komplexe Analysemethoden identifiziert und geeignet in der Steuerungssoftware berücksichtigt. Wird jedoch eine Störung nicht berücksichtigt oder kein Kompensationsmechanismus realisiert, ist ein Stillstand der Anlage unumgänglich.

In dieser Arbeit wird daher das Ziel verfolgt, diese ungeplanten Stillstände zu vermeiden und den Betrieb im Fehlerfall aufrechterhalten zu können. Hierzu werden Module bzw. Maschinen einer Anlage aus funktionaler Sicht im Hinblick auf die durch sie ausführbaren Automatisierungsfunktionen betrachtet, welche den Zustand eines Moduls und damit der Gesamtanlage sowie ggf. eines Werkstückes manipulieren. Adressiert wird die Zielsetzung der Aufrechterhaltung der Fertigung im Fehlerfall durch die Bestimmung eines Ablaufs unter Rückgriff auf durch eine Störung nicht betroffener Funktionen einer Anlage, welcher ebenfalls den gewünschten technischen Prozess für die Fertigung eines Produktes realisiert. Hierfür wurden auf Basis des beschriebenen Verfahrens zur Bestimmung von Ablaufreihenfolgen der Automatisierungsfunktionen – dem

5. Untersuchung der Anwendbarkeit und Evaluation von HiTraP-AT

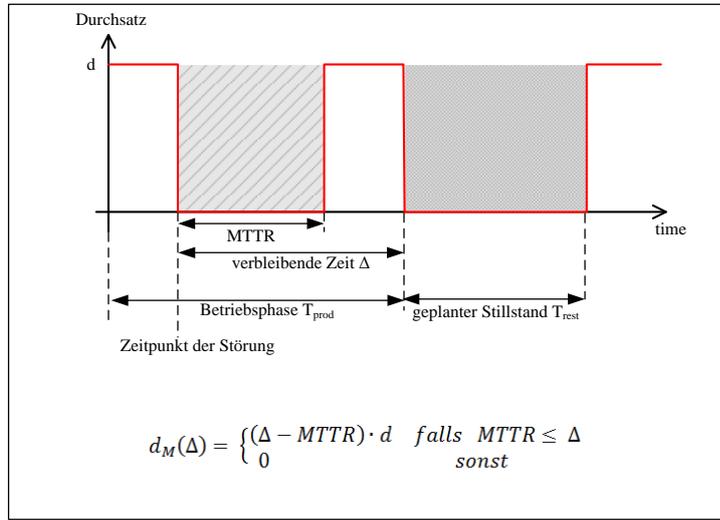
HiTraP-AT Planungsverfahren – mehrere Prototypen realisiert, welche für unterschiedliche Laboranlagen eine Rekonfiguration der SPS-basierten Steuerungssoftware exemplarisch umsetzen. Dabei wurden, wie in Abschnitt 5.3 beschrieben, unterschiedliche Funktionsstörungen simuliert und mittels des beschriebenen Verfahrens erfolgreich kompensiert.

Die Bewertung von Ansätzen für eine dynamische Adaption von Software stellt allgemein eine Herausforderung dar. Es existieren bereits einige Arbeiten mit Vorschlägen für einheitliche Maßstäbe, um die Leistungsfähigkeit unterschiedlicher Ansätze vergleichen und bewerten zu können [TPB⁺13, VHRF⁺16]. Bis dato existiert jedoch noch keine allgemeingültige und vollständige Bewertungsgrundlage für adaptive Software. Raibulet und Masciadri [RM09] führt dies insbesondere darauf zurück, dass Adaptivität von einer Vielzahl unterschiedlicher Aspekte abhängig ist und sich auf zahlreiche unterschiedliche Leistungskennzahlen auswirken kann. Hierdurch sind Bewertungen und Kennzahl zumeist nur für spezifische Anwendungsfälle anwendbar. Eine häufig genutzte Leistungskennzahl für die allgemeine Bewertung der Wertschöpfung einer Produktionsanlage stellt die Gesamtanlageneffektivität (engl. *overall equipment effectiveness*, OEE) dar, welche Verfügbarkeit, Leistung und Qualität sowie explizit ungeplante Verluste berücksichtigt [MS09]. Wie in [VHRF⁺16] beschrieben, kann hierdurch jedoch keine Aussagen über eine (teilweise) Aufrechterhaltung der Produktion im Falle eines Fehlers und etwaige damit verbundene Folgenden auf den Durchsatz oder die Produktqualität getroffen werden. In [RM10] wird eine Reihe von Metriken für adaptive Software definiert. Die Definition erfolgt dabei jedoch aus Sicht der Softwaretechnik ohne Berücksichtigung spezifischer Aspekte der Maschinen- und Anlagenautomatisierung bzw. cyber-physischer Systeme wie beispielsweise Produktqualität oder Produktionsdurchsatz. [VHRF⁺16, VHWF15] schlagen eine Verfeinerung dieser Metriken für den Wiederanlauf und die Selbstkonfiguration von Maschinen- und Anlagen vor. Die entsprechenden Metriken werden dabei unter Rückgriff auf spezifische Merkmale SPS-basierte Steuerungssoftware, wie beispielsweise Funktionsbausteine und Zykluszeit, definiert und ermöglichen hierdurch eine Kennzahlberechnung für Adaptionsmechanismen SPS-basierter Software. Kern der hier vorliegenden Dissertation ist, wie zuvor bereits beschrieben, das Verfahren zur Bestimmung von Ablaufreihenfolgen von Automatisierungsfunktionen, welche durch Speicherprogrammierbare Steuerungen ausgeführt werden. Hierbei erfolgt keine Betrachtung auf Basis einzelner Funktionsbausteine bzw. deren Ausführungszeit sondern vielmehr auf Basis der Funktionen, welche durch einen oder mehrere Funktionsbausteine realisiert wird. Auch werden keine Anforderungen hinsichtlich harter Echtzeit an das Planungsverfahren gestellt (vgl. Abschnitt 1.1.2 sowie Anforderung A3 in Abschnitt 2.2), weshalb die Planung alternativer Abläufe im Fehlerfall außerhalb der Steuerungssoftware der Feldebene durchgeführt wird. Dementsprechend ist eine Anwendung von Metriken, welche eine zyklische Abarbeitung zu Grunde legen, nicht zielführend. Obwohl die hier vorliegende Arbeit, analog zu [VHRF⁺16], explizit Software für Speicherprogrammierbare Steuerungen nach IEC 61131 adressiert, können die dort beschriebenen Metriken deshalb nicht direkt für die Bewertung des hier vorliegenden Ansatzes angewandt werden. Aus diesem Grund wird im Folgenden ein alternativer Ansatz für die Bewertung verfolgt: Unter der Prämisse, dass durch jede Rekonfiguration eine gleichbleibende Qualität des Produktes erreicht werden kann, wird als Leistungskennzahl das dieser Arbeit bereits in den Abschnitten 2.2 und Anforderung A2 motivierte Optimierungskriterium des tatsächlich realisierten Durchsatzes innerhalb eines wohldefinierten zeitlichen Horizonts genutzt.

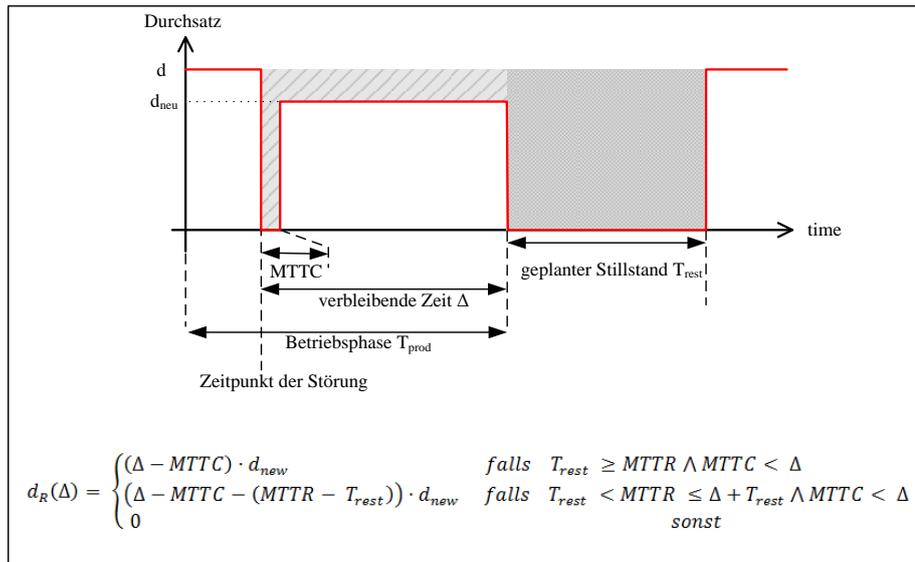
Gerade, wenn eine Störung bisher nicht explizit in der Entwicklung berücksichtigt wurde, ist ein sofortiger Wartungsstopp zumeist die einzig mögliche Reaktion, um weiteren Schaden an Mensch und/oder Maschine zu verhindern. Eine dynamische Rekonfiguration der Steuerungssoftware kann hierzu als Alternative betrachtet werden, um ungeplante Stillstandszeiten (wie in diesem Fall ein sofortiger Wartungsstopp) zu vermeiden. Wie in eigenen Vorarbeiten [LVH14a] aufgezeigt wurde, kann der Effekt einer Störung im Falle der Anwendung unterschiedlicher Strategien – d.h. hier insbesondere einer sofortigen, ungeplanten Wartungspause oder einer dynamischen Rekonfiguration – mathematisch berechnet und gegenübergestellt werden. Dieser Sachverhalt ist in Abbildung 5.8 dargestellt. Ausgehend von dem Zeitpunkt der Erkennung einer Störung wird hierbei die verbleibende Produktionszeit Δ bis zu der nächsten, geplanten Stillstandszeit bestimmt. Dies dient wiederum als Grundlage für die Berechnung des möglichen Durchsatzes unter der Prämisse eines konstanten Durchsatzes d im fehlerfreien Betrieb bzw. eines Durchsatzes d_{neu} im Anschluss an eine durchgeführte Rekonfiguration. Eine ungeplante Stillstandszeit lässt sich in beiden Fällen – weder bei einem ungeplanten Wartungsstopp noch im Falle einer wie in dieser Arbeit zu Grunde gelegten dynamischen Rekonfiguration – vollständig vermeiden. Während sich im Falle eines Wartungsstopps ein vollständiger Durchsatzverlust für die gesamte Reparaturzeit – hier repräsentiert durch MTTR (engl. *mean time to repair*) – manifestiert, ist dies im Falle einer dynamischen Rekonfiguration lediglich für die Zeitdauer der Planung und Anpassung der Abläufe – hier als MTTC (engl. *mean time to configure*) bezeichnet – der Fall. Die entsprechenden mathematischen Funktionen für die Berechnung des Durchsatzes in beiden betrachteten Fällen können Abbildung 5.8 entnommen werden.

Die Gegenüberstellung des Durchsatzes in beiden Fällen ermöglicht das Fazit, dass eine dynamische Rekonfiguration des Fertigungssystems eine vielversprechende Alternative darstellt, um die Verfügbarkeit – insbesondere im Falle von Störungen, welche a priori nicht explizit berücksichtigt wurden und dementsprechend keine adäquate Reaktion implementiert wurde – zu steigern. Von der Vorteilhaftigkeit einer dynamischen Rekonfiguration im Vergleich zu einem sofortigen Wartungsstopp kann jedoch nicht allgemein gesprochen werden, da diese nicht in jedem Fall gültig ist. In einigen Fällen können durch eine sofortige Behebung bessere Ergebnisse hinsichtlich des möglichen, zu realisierenden Durchsatzes erzielt werden. Ein solcher Fall ergibt sich beispielsweise, wenn bei einer vergleichsweise langen, noch verbleibenden Produktionszeit bis zu einem geplanten Stillstand, eine auftretende Störung schnell und problemlos behoben werden kann, dem Gegenüber jedoch eine geeignete Rekonfiguration lediglich ein vergleichsweise geringen Durchsatz erreicht werden kann. Wird eine potentielle Störung nicht a priori berücksichtigt, ist anzunehmen, dass (falls verfügbar) ein alternativer Prozess höchstens den gleichen – im allgemeinen jedoch eher einen signifikant schlechteren – Durchsatz realisieren wird. Demzufolge gilt allgemein, dass je länger die verbleibende Zeit bis zu einem geplanten Stillstand ist, an dem ohne weiteren Durchsatzverlust eine Behebung der Störung durchgeführt werden kann, umso besser ist ein sofortiger Wartungsstopp im Hinblick auf den realisierten Durchsatz. Dabei gilt jedoch auch, dass je geringer der Durchsatzverlust im Falle einer Rekonfiguration ist, umso vorteilhafter ist potentiell der Einsatz einer dynamischen Rekonfiguration. Allgemein sind bei dieser Entscheidung hinsichtlich der geeigneten Kompensationsstrategie noch weitere Faktoren zu berücksichtigen, wie beispielsweise die Verfügbarkeit notwendiger Ersatzteile und Wartungspersonal, ggf. ungenutzten Einsatzkosten manueller Arbeitskräfte oder die aktuelle Auftragslage sowie etwaige Vertragsstrafen.

5. Untersuchung der Anwendbarkeit und Evaluation von HiTraP-AT



(a) Visuelle Darstellung und mathematische Berechnung des Durchsatzes im Falle eines durch Störung bedingten, sofortigen Wartungsstopps.



(b) Visuelle Darstellung und mathematische Berechnung des Durchsatzes im Falle einer möglichen Rekonfiguration zur Störkompensation.

Abbildung 5.8.: Gegenüberstellung des erreichbaren Durchsatzes einer Anlage im Falle einer Störung bei Durchführung eines sofortigen Wartungsstopps sowie einer dynamischen Rekonfiguration (in Anlehnung an [LVH14a]).

Während in den im Rahmen von Vorarbeiten entwickelten Prototypen die grundsätzliche Anwendbarkeit des Konzeptes für die zu Grunde gelegten Anforderungen gezeigt und damit Anwendungsbeispiel 2 adressiert werden konnte, erweist sich HiTraP-AT als vielfältig einsetzbar und bildet dabei einen Schritt hin zur Realisierung der Vision sich selbstheilender cyber-physischer Systeme. So wurde bereits durch andere Wissenschaftler das hier beschriebene HiTraP-AT Planungsverfahren für weiterführende Forschungsarbeiten genutzt. Priego et al. [PAE⁺15, PSVHM15, Rem17] schlagen beispielsweise eine Middleware vor, die es ermöglicht, Programmcode von Steuerungen (auch während dem Betrieb) flexibel auszutauschen. Dabei wird das hier vorgeschlagene Planungsverfahren konzeptionell in deren Arbeit aufgegriffen und integriert: Durch HiTraP-AT identifizierte Handlungspläne und daraus resultierender Steuerungsprogramme bilden den Ausgangspunkt des von Priego et al. vorgeschlagenen Konzeptes. Bareiß et al. [BSP⁺16] schlagen ein Verfahren für den Wiederanlauf von automatisierten Produktionssystemen im Fehlerfall unter Rückgriff auf das in dieser Arbeit beschriebene HiTraP-AT Verfahren vor. Der Ansatz von Bareiß et al. geht dabei über den in dieser Arbeit zu Grunde gelegten Anwendungsfall zur Störkompensation hinaus: Unter Rückgriff auf industrielle Standards aus der Verpackungsindustrie wird ein hierarchisches Zustandsmodell sowie (analog zu einigen Aktionssprachen) innerhalb des Modells eine zusätzliche Beziehung zwischen Aktionen eingeführt, welche beschreibt, ob zwei Aktionen invers zueinander sind (d.h. eine Aktion ein System wieder in den Zustand vor der Ausführung der inversen Aktion versetzt). Der Wiederanlaufplan im Fehlerfall wird auf Basis des durch HiTraP-AT ursprünglich bestimmten Planes sowie entsprechend inverser Aktionen bestimmt.

Zusammenfassend ermöglicht das in dieser Arbeit vorgeschlagene HiTraP-AT Verfahren zur Ablaufplanung von Automatisierungsfunktionen einen weiteren Schritt hin zu selbstkonfigurierenden bzw. selbstheilenden Maschinen und Anlagen. Im Kontext von Anwendungsbeispiel 2 ermöglicht der Einsatz von HiTraP-AT eine alternative Reaktion auf a priori nicht explizit berücksichtigte Störungen. Die Nutzung von HiTraP-AT im Kontext weiterführender Forschungen zeigt weitere Anwendungsmöglichkeiten und Potentiale auf.

5.7.3. Zusammenfassung

Während in vielen Arbeiten ausschließlich eine spezifische Anwendung im Vordergrund steht, wurden die im Zuge der in dieser Dissertation beschriebenen Entwicklungen und Untersuchungen auf Basis zweier vergleichsweise heterogener Anwendungsbeispielen motiviert. Hierdurch konnte eine Fokussierung auf funktionale Aspekte des Algorithmus sowie dessen Leistungsfähigkeit im Hinblick auf Zeit- und Skalierbarkeitsverhalten erreicht werden. Ferner konnte durch die zu Grunde gelegte, verallgemeinerte Fragestellung ein Verfahren zur automatischen Ableitung von Automatisierungsfunktionen – d.h. Softwarefunktionen der Feldebene für die Maschinen- und Anlagenautomatisierung – entwickelt werden, welches sich für unterschiedliche Anwendungsbereiche und Anwendungsfälle eignet. Dies spiegelt sich auch in der Tatsache wider, dass neben den zu Grunde gelegten Anwendungsbeispielen, das HiTraP-AT Verfahren für weiterführende Arbeiten und weitere Anwendungsfälle bereits Anwendung findet [PAE⁺15, PSVHM15, BSP⁺16].

Ferner konnte durch die in dieser Dissertation beschriebenen Forschungsarbeiten und Ergebnisse für den Stand der Wissenschaft in den beiden zu Grunde gelegten Anwendungsbereichen ein Beitrag geleistet werden. So wurde in Abschnitt 5.7.1 dargelegt, wie durch den Einsatz von HiTraP-AT ein Assistenzsystem für die Konfiguration modularer Anlagen umgesetzt werden kann

5. Untersuchung der Anwendbarkeit und Evaluation von HiTraP-AT

(Anwendungsbeispiel 1 „Konfigurationsassistenz“). Während zuvor eine vollständige Spezifikation notwendiger Ausführungsreihenfolgen von Automatisierungsfunktionen für die Realisierung eines technischen Prozesses notwendig war, konnte im Rahmen dieser Arbeit exemplarisch gezeigt werden, dass eine automatisierte Ableitung notwendiger Reihenfolgen möglich ist und hierdurch eine Assistenzfunktionen für diese Aufgabe realisiert werden kann. Es zeigte sich in den durchgeführten Untersuchungen jedoch auch, dass für eine weitere Anwendung des Verfahrens insbesondere die aktuell genutzte Spezifikation der Anforderungen an einen durchzuführenden technischen Prozess mittels Start- und gewünschtem Zielzustand des Produktes einschränkend ist. Somit stellt diese Arbeit und das darin beschriebene Konzept zur Ableitung von Ablaufreihenfolgen von Steuerungsfunktionen im Kontext von Anwendungsbeispiel 1 einen einzelnen Schritt hin zu Assistenzfunktionen bei der Entwicklung zukünftiger Anlagen dar.

In Abschnitt 5.7.2 wurde beschrieben, wie durch den Einsatz von HiTraP-AT für eine dynamischer Rekonfiguration eine Verbesserung der Verfügbarkeit im Falle unvorhergesehener Störungen erreicht werden kann (Anwendungsbeispiel 2 „Dynamische Rekonfiguration zur Störkompensation“). Dabei konnte identifiziert werden, dass keine allgemeine gültige Aussage über die Vorteilhaftigkeit einer dynamischen Rekonfiguration im Hinblick auf den erreichbaren Durchsatz im Fehlerfall getroffen werden kann, sondern dies vielmehr von einer Vielzahl von Faktoren – insbesondere dem Fehlerzeitpunkt bzw. der verbleibenden Zeit bis zu einem geplanten Stillstand – abhängig ist. Im Rahmen dieser Arbeit konnte im Hinblick auf die dynamische Konfiguration von Steuerungssoftware die Anwendbarkeit des vorgeschlagenen Verfahrens für unterschiedliche Funktionsstörungen, aber auch unterschiedliche Laboranlagen gezeigt werden. Für einen Einsatz des Verfahrens in der industriellen Praxis sind jedoch auch für dieses Anwendungsbeispiel weitere Forschungen notwendig. Eine dieser potentiell einschränkenden Prämissen dieser Arbeit – die notwendige, manuelle Entnahme aller Werkstücke im Falle einer Rekonfiguration – wurde bereits in der auf dieser Arbeit aufbauenden Arbeit von Bareiß et al. [BSP⁺16] weiter verfolgt. Auch ist anzumerken, dass der Einsatz von HiTraP-AT für eine dynamische Rekonfiguration, eine spezifische Struktur der Steuerungssoftware voraussetzt, welche für bereits existierende Anlagen nicht vorliegt. Eine Anwendung für bereits existierende Maschinen und Anlagen ist ohne vollständige Überarbeitung der eingesetzten SPS-basierten Steuerungssoftware nicht möglich.

Zusammenfassend konnte gezeigt werden, dass der Einsatz von HiTraP-AT in beiden betrachteten Anwendungsfeldern vorteilhaft ist und einen soliden Ausgangspunkt für weitere Forschungen im Kontext von Industrie 4.0 bzw. cyber-physischer Systeme bietet. Es konnte dabei ebenfalls identifiziert werden, dass weiterer Forschungsbedarf im Hinblick auf einen praktischen Einsatz des vorgeschlagenen Planungsverfahrens existiert.

6. Zusammenfassung und Ausblick

Im Rahmen dieser Arbeit wurde ein Verfahren zur automatischen Ableitung von partiell geordneten Handlungsplänen für die Steuerungsebene des Maschinen- und Anlagenbaus vorgestellt: das HiTraP-AT Planungsverfahren. Dieses Verfahren berücksichtigt einerseits domänenspezifische Anforderungen, um die Anwendbarkeit für IEC 61131-3 basierte Steuerungen zu gewährleisten, sowie andererseits den Einsatz des Planungsverfahrens innerhalb der Entwicklung von Maschinen- und Anlagen zu ermöglichen. Letzteres wurde durch die nahtlose Integration des Planungsverfahrens in einen existierenden, modellgetriebenen Entwicklungsansatz für den Maschinen- und Anlagenbau adressiert. Das HiTraP-AT Planungsverfahren setzt dabei auf eine Erweiterung der klassischen Formulierung von Planungsproblemen. Konkret werden durch HiTraP-AT in dessen internem Planungsmodell zwei wesentliche, konzeptionelle Ansätze kombiniert: Zum einen wird das Konzept zusammenhängender Handlungsräume vorgeschlagen, um die entsprechend der gestellten Anforderung spezifischen Charakteristika von Ablaufplänen von Automatisierungsfunktionen – d.h. feingranularer Softwarefunktionen der Steuerungsebene für die Maschinen- und Anlagenautomatisierung – unter der Prämisse der Massenproduktion von Gütern zu adressieren und dabei die konzeptionelle Brücke zur klassischen, zielbasierten Formulierung von Planungsproblemen zu bilden. Zum anderen wird eine zustandsbasierte Hierarchisierung des Handlungsraumes auf Basis der Zustände vorgeschlagen. Hierdurch wird unter Rückgriff auf eine domänenspezifische Strukturierung von Anlagen eine inhärente partielle Ordnung der Handlungspläne erreicht, wodurch letztendlich ein partiell geordneter Handlungsplan automatisch bestimmt werden kann. Die Lösung des zu Grunde gelegten Planungsproblems erfolgt in HiTraP-AT durch eine, ebenfalls im Rahmen dieser Dissertation vorgeschlagene, Formulierung des internen HiTraP-AT Planungsmodells als binäres, lineares Optimierungsproblem. Diese Formulierung erweist sich im Zuge der Evaluation als gut skalierbar im Hinblick auf die Größe des internen Planungsmodells, wodurch letztendlich die Bestimmung von Abläufen sehr effizient im Hinblick auf die Laufzeit der Planung, auch für größere und (bezüglich der Menge möglicher Automatisierungsfunktionen) flexible Anlagen, erfolgen kann. Die Erfüllung eingangs in dieser Dissertation gestellten Anforderungen an ein geeignetes Verfahren für die Ableitung von Ablaufreihenfolgen von Automatisierungsfunktionen und die damit verbundene Anwendbarkeit des entwickelten HiTraP-AT Planungsverfahrens wurde mittels Untersuchungen anhand unterschiedlicher Fallstudien gezeigt. Dabei konnte identifiziert werden, dass die Implementierung des HiTraP-AT Planungsverfahrens ein geeignetes zeitliches Antwortverhalten bei der Bestimmung von Ablaufplänen aufweist, um die initial in dieser Dissertation beschriebenen Anwendungsbeispiele unter den ebenfalls im Zuge der Beschreibung der Anwendungen genannten Prämissen geeignet adressieren zu können. Um die Anwendbarkeit des Planungsverfahrens für die Domäne des Maschinen- und Anlagenbaus sicherzustellen, wurde eine Integration des HiTraP-AT Planungsverfahrens mit einem modellgetriebenen Entwicklungsansatz – der von Daniel Schütz in seiner Dissertation [Sch14] beschriebenen UML-OP – vorgeschlagen. Im Zuge der durchgeführten Untersuchungen zur Evaluation des HiTraP-AT Planungsverfahrens konnte eine erfolgreiche Integration beider Ansätze gezeigt werden: Eine automatische Bestimmung von Abläufen kann auf Basis der durch die UML-OP gegebenen Informationen erfolgen. Hierdurch wird letztendlich die Grundlage für einen Einsatz des Planungsverfahrens in der Entwicklung von Maschinen und Anlagen gelegt, da eine manuelle Erstellung notwendiger Modelle für die Planung einen zusätzlichen Aufwand bedingen

6. Zusammenfassung und Ausblick

würde. Das HiTraP-AT Konzept umfasst ferner einen Ansatz zur Implementierung und Ausführung der automatisch bestimmten Abläufe durch Speicherprogrammierbare Steuerungen, welche dem aktuell in der industriellen Praxis am weitesten verbreiteten Standard – dem IEC 61131 Standard – entsprechen. Hierdurch wird letztendlich die essentielle Zielsetzung dieser Dissertation – ein Planungsverfahren für die Steuerungsebene der Maschinen- und Anlagenautomatisierung – adressiert. Die Anwendbarkeit des Planungsverfahrens für, mittels Sprachen im IEC 61131-3 programmierbaren Steuerungen, wurde durch Anwendung des Konzeptes für drei verschiedene Laboranlagen des Lehrstuhls AIS positiv evaluiert. Im Zuge der Evaluation konnten ferner Grenzen des Ansatzes identifiziert werden, wodurch sich konkrete Einschränkungen der Anwendbarkeit ergeben: Das HiTraP-AT Planungsverfahren kann ausschließlich für modulare Fertigungssysteme mit IEC 61131-3 basierter Steuerungssoftware eingesetzt werden. Ferner ergibt sich eine wesentliche Einschränkungen hinsichtlich des durch HiTraP-AT unterstützten, realisierbaren technischen Prozesses und Materialflusses innerhalb der Anlage: Es werden ausschließlich Abläufe durch HiTraP-AT geplant, bei denen ein einzelnes Werkstück einmalig für die Bearbeitung an einem Modul der Anlage bereitsteht. Dadurch können keine technischen Prozesse durch das vorgeschlagene Planungsverfahren identifiziert werden, welche zyklische Materialflüsse bedingen. Die Evaluation des Ansatzes beschränkte sich auf fertigungstechnische Demonstratoren mit Laborcharakter. Die Anwendbarkeit des Ansatzes für reale, industrielle Anlagen bleibt offen.

Nichtsdestotrotz zeigte sich, dass der in dieser Arbeit vorgeschlagene Ansatz zur automatischen Bestimmung der Ablaufreihenfolgen von Automatisierungsfunktionen einen möglichen Ausgangspunkt für weitere Forschungsarbeiten sowohl für die Entwicklung als auch den Betrieb rekonfigurierbarer, wandlungsfähiger Maschinen- und Anlagen darstellt. Teile des in dieser Arbeit vorgeschlagenen Konzeptes finden bereits in verschiedenen Lehrveranstaltungen des Lehrstuhls für Automatisierung und Informationstechnik der Technischen Universität München Anwendung: So wird im Rahmen verschiedener, durch den Lehrstuhl angebotener Vorlesungen für ingenieurwissenschaftliche Masterstudiengänge beispielsweise die Integration des HiTraP-AT Planungsverfahrens mit der UML-OP von Daniel Schütz als exemplarisches Beispiel für die weitere Anwendung modellgetriebener Software- und Systementwicklung beschrieben. Auch das Konzept der diskreten Aktionen der Steuerungssoftware und deren Beschreibung in Form von Vor- und Nachbedingungen, was die gemeinsame Basis von UML-OP und HiTraP-AT bildet, wird in unterschiedlichen Vorlesungen genutzt. Im Zuge eines als Lehrveranstaltung des Lehrstuhls angebotenen Praktikums, bei dem die Entwicklung intelligenter, verteilter Systeme in der Mechatronik im Fokus steht, wird den Teilnehmern ebenfalls die Anwendung von Teilaspekten des in dieser Arbeit beschriebenen Konzeptes, wie z.B. die modellgetriebene Entwicklung diskreter Steuerungsfunktionen (d.h. Automatisierungsfunktionen) sowie die in dieser Arbeit beschriebenen Implementierung konfigurierbarer Abläufe der Softwarefunktionen und dessen automatische Generierung, näher gebracht. Des Weiteren wurden und werden aufbauend auf dem in dieser Arbeit vorgeschlagene HiTraP-AT Planungsverfahren bereits weitere internationale Forschungsarbeiten durchgeführt, wie z.B. durch Priego et al. [PAE⁺15, PSVHM15] oder Bareiß et al. [BSP⁺16].

Industrie 4.0 bedingt die operative und strategische Wandlung der Geschäftsprozesse und eine damit einhergehende, technologische Transformation hin zu einem sozio-technischem Ökosystem auf Basis intensiver Digitalisierung und nahtloser Vernetzung. Aus diesem Grund wurden

und werden in diesem Kontext viele (häufig auch sehr praxisnahe und anwendungsorientierte) Forschungs- und Entwicklungsarbeiten durchgeführt [BTW⁺15, BVZ15, VDM16]. Die in dieser Arbeit beschriebenen Konzepte und Lösungswege können hierbei für unterschiedliche Forschungsprojekte und weiterführende Arbeiten als Startpunkt betrachtet werden. Im Zuge dieser Entwicklungen ist beispielsweise ein zunehmender Trend hin zu modularen Maschinen und Anlagen beobachtbar und entspricht dabei dem in dieser Arbeit als Grundlage genutzte Modularitätskonzept von Maschinen- und Anlagen. Nichtsdestotrotz stellen Anwendungsfelder, welche im Zuge von Industrie 4.0 nochmals an Bedeutung gewinnen (siehe hierzu [BVZ15, ABD⁺16]), wie beispielsweise wandlungsfähige Produktionssysteme, bei denen ggf. während der Laufzeit Maschinen ausgetauscht werden können (sog. Plug-and-Produce), oder die Fertigung kundenspezifischer Güter (sog. Mass Customization), weitere oder andere Anforderungen an ein geeignetes Lösungskonzept. Daher wird es notwendig sein, die in dieser Arbeit zu Grunde gelegten Anforderungen zu lockern oder anzupassen, um diese Anwendungen im Kontext von Industrie 4.0 geeignet adressieren zu können. Aus der starken Anlehnung des in dieser Arbeit entwickelten Konzeptes an eingangs in dieser Arbeit gestellte Anforderungen resultiert ein vergleichsweise starres Konzept im Hinblick auf die Änderung von Anforderungen. So wurde bereits im Zuge der Untersuchungen im vorangegangenen Abschnitt beschrieben, dass je nach Anpassung der Anforderungen, eine mehr oder weniger große und ggf. weitreichende Anpassung des Konzeptes notwendig ist.

Maschinen und Anlagen, welche ein Verständnis darüber besitzen, was sie tun und tun können, stellen die Grundlage für intelligenten, technischen Systemen dar, wie dies bereits von [Bra02, BBW07] dargelegt wurde. Das in dieser Arbeit vorgeschlagene Konzept und die aus der Untersuchung des Ansatzes gewonnen Erkenntnisse können als soliden Schritt hin zu einer flexibleren Steuerungssoftware für die Maschinen- und Anlagenautomatisierung betrachtet werden, zeigen jedoch auch deutlich die Grenzen des Ansatzes und daraus resultierende Einschränkungen der Anwendbarkeit auf. Nichtsdestotrotz kann das in dieser Arbeit beschriebene Planungsverfahren zur automatischen Identifikation geeigneter Handlungspläne für die Steuerungsebene des Maschinen- und Anlagenbaus als solide Grundlage und Ausgangspunkt weiterer Forschungen aufgefasst werden, um Zukunftsvisionen und zukünftige Anforderungen an Maschinen und Anlagen von Morgen wie Self-x Eigenschaften, Plug-and-Produce sowie autonome, intelligenter, cyber-physischer Produktionssysteme zu realisieren.

Abbildungsverzeichnis

1.1.	Konzeptionelle Übersicht eines Anlagenkonfigurators.	6
1.2.	Konzeptionelle Übersicht eines Assistenzsystems für die Rekonfiguration des Steuerungsverhaltens im Fehlerfall.	8
1.3.	Grafischer Überblick über den Aufbau dieser Dissertation.	11
2.1.	Konzeptuelles Modell automatischer Planungsverfahren.	14
2.2.	Zusammenhang zwischen technischem System und technischem Prozess.	17
2.3.	Hierarische Steuerungsstruktur industrieller Produktionssysteme.	18
2.4.	Unterschiedliche Weiterentwicklungen der Automatisierungspyramide.	20
2.5.	Zyklische Abarbeitung eines SPS Programms.	21
2.6.	Überblick der Programmiersprachen nach IEC 61131-3.	22
2.7.	Übersicht der Anforderungen an das Planungsverfahren.	24
2.8.	Mechanischer Aufbau eines pneumatischen Greifers.	25
2.9.	Schematische Darstellung des exemplarischen Greifers.	27
2.10.	Schematische Darstellung eines Intralogistiksystems mittels schienengeführter Wagenträger.	29
2.11.	Gegenüberstellung des Anlagendurchsatzes im Störfall mit unterschiedlichen Reaktionsstrategien.	32
4.1.	Gegenüberstellung von Anforderungen und konzeptionellen Lösungen.	67
4.2.	Schematische Darstellung der HiTraP-AT Grundstruktur.	69
4.3.	Schematische Darstellung konditionaler hierarchischer S/T-Netze.	74
4.4.	Beispiel für ein geglättetes, konditional hierarchisches S/T-Netz.	75
4.5.	Gesamtstruktur des HiTraP-AT Planungsmodells.	77
4.6.	Beispiel des Übergabemechanismus eines Produktes zwischen zwei Modulen in HiTraP-AT.	78
4.7.	Ausschnitt einer exemplarischen Darstellung der Modellierung eines Rücklaufs für ein Modul mit zwei Variablen.	80
4.8.	Graphische Darstellung der Konzepte eines HiTraP-AT Planungsproblems.	81
4.9.	Exemplarische Darstellung eines partiell geordneten HiTraP-AT Handlungsplanes (Ausschnitt).	85
4.10.	Exemplarische Restrukturierung eines internen HiTraP-AT Planungsmodells für die vereinfachte Formulierung als Lineares Programm.	90
4.11.	Variablenspezifisches S/T-Netz der Variablen <i>oben</i> des Kranmoduls.	91
4.12.	Variablenspezifisches S/T-Netz der Variablen <i>Ort</i> des Produktes in der Stempelanlage.	93
4.13.	Beispiel eines Kurzzyklus im S/T-Netz einer Produktvariablen.	95
4.14.	Exemplarischer Abhängigkeitsgraphen des Kranmoduls der Stempelanlage.	98
4.15.	Ausschnitt eines charakteristischen HiTraP-AT Handlungsplanes der Stempelanlage.	99
4.16.	Tabellarische Notation der Funktionalität von Modulen/Komponenten.	101
4.17.	Ausschnitt des UML-OP Modells der Stempelanlage.	104
4.18.	Ausschnitt eines Produktmodells für die Stempelanlage.	105

4.19. Unterschiede der Deklarationen in klassischer und objektorientierter IEC 61131 am Beispiel des Kranmoduls der Stempelanlage.	113
4.20. Auszug des Deklarationsteils für die Repräsentation sequentieller Handlungspläne in IEC 61131-3 ST.	114
4.21. Zusammenhang zwischen UML-OP und IEC 61131-3 ST.	115
5.1. Ausschnitt eines UML-OP Modells des Hybriden Prozessmodells.	118
5.2. Darstellung eines exemplarischen, flachen Zustandsraumes.	122
5.3. Ergebnis der Anwendung flacher, zusammenhängender Handlungsräume für die Systemfallstudie "Stempelanlage".	123
5.4. Ergebnis der Untersuchung des HiTraP-AT Planungswerkzeugs für das Hybride Prozessmodell.	124
5.5. Gegenüberstellung der Skalierungsfaktoren flacher und hierarchischer (jeweils zusammenhängender) Zustandsräume.	125
5.6. Zeitverhalten des HiTraP-AT Planungswerkzeugs für das Hybride Prozessmodell.	126
5.7. Laufzeitverhalten des HiTraP-AT Planungswerkzeugs.	127
5.8. Gegenüberstellung des erreichbaren Durchsatzes bei sofortigem Wartungsstopps und dynamischen Rekonfiguration.	142
A.1. Überblick der installierter Sensoren der Stempelanlage.	190
A.2. Gesamtübersicht des mechanischen Aufbaus der Stempelanlage.	191
A.3. Überblick des machanischen Aufbaus des Hybriden Prozessmodells.	192

Tabellenverzeichnis

3.1. Zusammenfassung der Analyseergebnisse verwandter Arbeiten zur automatischer Planung von Handlungsabläufen im Maschinen- und Anlagenbau.	65
4.1. Übersicht der Bedingungen zur Formulierung des internen HiTraP-AT Planungsmodells als Lineares Programm.	88

Literaturverzeichnis

- [ABD⁺16] ANDERL, Reiner ; BAUER, Klaus ; DIEGNER, Bernhard ; DIEMER, Johannes ; FAY, Alexander ; FRITZ, Jessica ; GOERICKE, Dietmar ; GROTEPASS, Jürgen ; HILGER, Claus ; JASPERNEITE, Jürgen ; KALHOFF, Johannes ; KUBACH, Uwe ; LÖWEN, Ulrich ; MENGES, Georg ; MICHELS, Jan S. ; MÜNCH, Wolfram ; PREISS, Helena ; REUS, Olaf ; SCHMIDT, Fabian ; SCHMOLZE-KRAHN, Raimund ; STEFFENS, Ernst-Joachim ; STIEDL, Thomas ; HOMPEL, Michael ten ; ZEIDLER, Christian: Aspekte der Forschungsroadmap in den Anwendungsszenarien / Bundesministerium für Wirtschaft und Energie. 2016. – Ergebnispapier der Plattform Industrie 4.0
- [Abo16] ABOWD, Gregory D.: Beyond weiser: From ubiquitous to collective computing. In: *Computer* 49 (2016), Nr. 1, S. 17–23
- [AIM10] ATZORI, Luigi ; IERA, Antonio ; MORABITO, Giacomo: The internet of things: A survey. In: *Computer networks* 54 (2010), Nr. 15, S. 2787–2805
- [ALF10] ANDERSSON, Kristin ; LENNARTSON, Bengt ; FABIAN, Martin: Restarting manufacturing systems; restart states and restartability. In: *IEEE Transactions on Automation Science and Engineering* 7 (2010), Juli, Nr. 3, S. 486 – 499. <http://dx.doi.org/10.1109/TASE.2009.2034136>. – DOI 10.1109/TASE.2009.2034136
- [ALFF11] ANDERSSON, Kristin ; LENNARTSON, Bengt ; FALKMAN, Petter ; FABIAN, Martin: Generation of restart states for manufacturing cell controllers. In: *Control Engineering Practice* 19 (2011), September, Nr. 9, S. 1014 – 1022. <http://dx.doi.org/doi:10.1016/j.conengprac.2011.05.013>. – DOI doi:10.1016/j.conengprac.2011.05.013
- [And06] ANDERSSON, Kristin: *Hierarchical Control and Restart of Flexible Manufacturing Systems*, Department of Signals and Systems, Automation, Chalmers University of Technology, Diss., 2006
- [And15] ANDERL, Reiner: Industrie 4.0 – Technological approaches, use cases, and implementations. In: *at – Automatisierungstechnik* 63 (2015), Nr. 10, S. 737 – 756. <http://dx.doi.org/10.1515/auto-2015-0025>. – DOI 10.1515/auto-2015-0025
- [Apt03] APT, Krzysztof R.: *Principles of constraint programming*. Cambridge University Press, Cambridge, UK, 2003
- [AS08] AIER, Stephan ; SCHELP, Joachim: EAI und SOA – Was bleibt nach dem Hype? In: BICHLER, Martin (Hrsg.) ; HESS, Thomas (Hrsg.) ; KRCMAR, Helmut (Hrsg.) ; LECHNER, Ulrike (Hrsg.) ; MATTHES, Florian (Hrsg.) ; PICOT, Arnold (Hrsg.) ; SPEITKAMP, Benjamin (Hrsg.) ; WOLF, Petra (Hrsg.): *Multikonferenz Wirtschaftsinformatik*, 2008, S. 1469 – 1480

- [Aug07] AUGUSTO, Juan C.: Ambient intelligence: the confluence of ubiquitous/pervasive computing and artificial intelligence. In: *Intelligent Computing Everywhere*. Springer, 2007, S. 213–234
- [AZ02] ALEXANDER, Ian ; ZINK, Thomas: Introduction to systems engineering with use cases. In: *Computing & Control Engineering Journal* 13 (2002), Nr. 6, S. 289–297
- [Bü14] BÜHRER, Ulrich: *Automatische Generierung anwendungsspezifischer, konfigurierbarer Echtzeitscheduler für Speicherprogrammierbare Steuerungen*, Lehrstuhl für Automatisierung und Informationssysteme, Technische Universität München, Diplomarbeit, 2014
- [BA97] BRESLOW, Leonard A. ; AHA, David W.: Simplifying decision trees: A survey. In: *The Knowledge Engineering Review* 12 (1997), Nr. 01, S. 1–40
- [BAB+03] BIUNDO, Susanne ; AYLETT, Ruth ; BEETZ, Michael ; BORRAJO, Daniel ; CESTA, Amedeo ; GRANT, Tim ; MCCCLUSKEY, L ; MILANI, Alfredo ; VERFAILLE, Gérard: Technological roadmap on AI planning and scheduling / Planet, the European Network of Excellence in AI Planning. Citeseer, 2003. – Forschungsbericht
- [Bac16] BACKHAUS, Julian Christoph S.: *Adaptierbares aufgabenorientiertes Programmiersystem für Montagesysteme*. München, Technische Universität München, Dissertation, 2016
- [Bay14] BAYRAK, Gülden: *Vergleich und Evaluation von Beschreibungsmitteln für die Automatisierung hybrider Prozesse*, Technische Universität München, Diss., 2014
- [BBB+11] BANNAT, Alexander ; BAUTZE, Thibault ; BEETZ, Michael ; BLUME, Juergen ; DIEPOLD, Klaus ; ERTELT, Christoph ; GEIGER, Florian ; GMEINER, Thomas ; GYGER, Tobias ; KNOLL, Alois u. a.: Artificial cognition in production systems. In: *IEEE Transactions on Automation Science and Engineering* 8 (2011), Dezember, Nr. 1, S. 148–174. <http://dx.doi.org/10.1109/TASE.2010.2053534>. – DOI 10.1109/TASE.2010.2053534
- [BBG06] BOHN, Hendrik ; BOBEK, Andreas ; GOLATOWSKI, Frank: SIRENA – Service Infrastructure for Real-time Embedded Networked Devices: A service oriented framework for different domains. In: *International Conference on Networking, International Conference on Systems, and International Conference on mobile Communication and Learning Technologies*. Morne, Mauritius : IEEE, April 2006, S. 43 – 48
- [BBH13] BRADLEY, Joseph ; BARBIER, Joel ; HANDLER, Doug: Embracing the Internet of everything to capture your share of \$14.4 trillion / Cisco. 2013. – White Paper

- [BBI⁺14] BECHON, Patrick ; BARBIER, Magali ; INFANTES, Guillaume ; LESIRE, Charles ; VIDAL, Vincent: HiPOP: Hierarchical Partial-Order Planning. In: ENDRISS, Ulle (Hrsg.) ; LEITE, Jo ao (Hrsg.): *STAIRS 2014: Proceedings of the 7th European Starting AI Researcher Symposium* Bd. 264. Prag, Tschechische Republik : IOS Press, August 2014 (Frontiers in Artificial Intelligence and Applications), S. 51
- [BBW07] BEETZ, Michael ; BUSS, Martin ; WOLLHERR, Dirk: Cognitive technical systems—what is the role of artificial intelligence? In: HERTZBERG, Joachim (Hrsg.) ; BEETZ, Michael (Hrsg.) ; ENGLERT, Roman (Hrsg.): *KI 2007: Advances in Artificial Intelligence* Bd. 4667. Osnabrück, Deutschland : Springer Berlin Heidelberg, September 2007 (Lecture Notes in Computer Science), S. 19 – 42
- [BCGL07] BERBEGLIA, Gerardo ; CORDEAU, Jean-François ; GRIBKOVSKAIA, Irina ; LAPORTE, Gilbert: Static pickup and delivery problems: a classification scheme and survey. In: *Top 15* (2007), Nr. 1, S. 1–31
- [BCM11] BROWN, Brad ; CHUI, Michael ; MANYIKA, James: Are you ready for the era of ‘big data’. In: *McKinsey Quarterly* 4 (2011), Nr. 1, S. 24–35
- [BD00] BOCKMAYR, Alexander ; DIMOPOULOS, Yannis: Integer Programs and Valid Inequalities for Planning Problems. In: BIUNDO, Susanne (Hrsg.) ; FOX, Maria (Hrsg.): *Recent Advances in AI Planning – 5th European Conference on Planning* Bd. 1809. Durham, UK : Springer Berlin Heidelberg, September 2000 (Lecture Notes in Computer Science), S. 239 – 251
- [BD08] BRAFMAN, Ronen I. ; DOMSHLAK, Carmel: From One to Many: Planning for Loosely Coupled Multi-Agent Systems. In: RINTANEN, Jussi (Hrsg.) ; NEBEL, Bernhard (Hrsg.) ; BECK, J. C. (Hrsg.) ; HANSEN, Eric (Hrsg.): *Proceedings of the 18th International Conference on Automated Planning and Scheduling*. Sydney, Australien : AAI, 2008, S. 28–35
- [BDR07] BALDAUF, Matthias ; DUSTDAR, Schahram ; ROSENBERG, Florian: A survey on context-aware systems. In: *International Journal of Ad Hoc and Ubiquitous Computing* 2 (2007), Nr. 4, S. 263 – 277. <http://dx.doi.org/10.1504/IJAHUC.2007.014070>. – DOI 10.1504/IJAHUC.2007.014070
- [BEP⁺13] BŁAŻEWICZ, Jacek ; ECKER, Klaus H. ; PESCH, Erwin ; SCHMIDT, Günter ; WEGLARZ, Jan: *Scheduling computer and manufacturing processes*. Springer-Verlag Berlin Heidelberg, 2013
- [Ber15] BERGAGÅRD, Patrik: *On restart of automated manufacturing systems*, Department of Signals and Systems, Automation, Chalmers University of Technology, Diss., 2015. – 192
- [BF97] BLUM, Avrim L. ; FURST, Merrick L.: Fast planning through planning graph analysis. In: *Artificial intelligence* 90 (1997), Februar, Nr. 1-2, S. 281 – 300. [http://dx.doi.org/10.1016/S0004-3702\(96\)00047-1](http://dx.doi.org/10.1016/S0004-3702(96)00047-1). – DOI 10.1016/S0004-3702(96)00047-1

- [BF13] BERGAGÅRD, Patrik ; FABIAN, Martin: Calculating restart states for systems modeled by operations using supervisory control theory. In: *Machines* 1 (2013), Dezember, Nr. 3, S. 116 – 141. <http://dx.doi.org/10.3390/machines1030116>. – DOI 10.3390/machines1030116
- [BF14] BERGAGÅRD, Patrik ; FABIAN, Matthias: Calculating restart states using reset transitions. In: *2014 IEEE International Conference on Robotics and Automation*. Hong Kong, China : IEEE, Mai 2014, S. 3345 – 3350
- [BFF15] BERGAGÅRD, Patrik ; FALKMAN, Petter ; FABIAN, Martin: Modeling and automatic calculation of restart states for an industrial windscreen mounting station. In: DOLGUI, Alexandre (Hrsg.) ; SASIADEK, Jurek (Hrsg.) ; ZAREMBA, Marek (Hrsg.) ; IFAC (Veranst.): *5th IFAC Symposium on Information Control Problems in Manufacturing* Bd. 48. Ottawa, Kanada : Elsevier, Mai 2015 (IFAC-PapersOnLine 3)
- [BFKR14] BRETTEL, Malte ; FRIEDERICHSEN, Niklas ; KELLER, Michael ; ROSENBERG, Marius: How virtualization, decentralization and network building change the manufacturing landscape: An industry 4.0 perspective. In: *International Journal of Mechanical, Industrial Science and Engineering* 8 (2014), Nr. 1, S. 37–44
- [BFPD08] BRECHER, Christian ; FAYZULLIN, K ; POSSEL-DÖLKEN, Frank: Intelligent operations control: Architecture for seamless integration of scheduling and execution. In: *Production Engineering* 2 (2008), Nr. 3, S. 293–301
- [BGIR09] BORANGIU, Theodor ; GILBERT, Pascal ; IVANESCU, Nick-Andrei ; ROSU, Andrei: An implementing framework for holonic manufacturing control with multiple robot-vision stations. In: *Engineering Applications of Artificial Intelligence* 22 (2009), Juni, Nr. 4-5, S. 505 – 521. <http://dx.doi.org/10.1016/j.engappai.2009.03.001>. – DOI 10.1016/j.engappai.2009.03.001
- [BGT81] BLAND, Robert G. ; GOLDFARB, Donald ; TODD, Michael J.: The ellipsoid method: A survey. In: *Operations Research* 29 (1981), Dezember, Nr. 6, S. 1039 – 1091. <http://dx.doi.org/10.1287/opre.29.6.1039>. – DOI 10.1287/opre.29.6.1039
- [BHH⁺16] BERNSHAUSEN, Jens ; HALLER, Axel ; HOLM, Thomas ; HOERNICKE, Mario ; OBST, Michael ; LADIGES, Jan: Namur Modul Type Package – Definition. In: *atp edition* 58 (2016), Nr. 01-02, S. 72–81. <http://dx.doi.org/10.17560/atp.v58i01-02.554>. – DOI 10.17560/atp.v58i01-02.554
- [Bix02] BIXBY, Robert E.: Solving real-world linear programs: A decade and more of progress. In: *Operations Research* 50 (2002), Februar, Nr. 1, S. 3 – 15. <http://dx.doi.org/10.1287/opre.50.1.3.17780>. – DOI 10.1287/opre.50.1.3.17780
- [BJL86] BLAIR, Charles E. ; JEROSLOW, Robert G. ; LOWE, James K.: Some results and experiments in programming techniques for propositional logic. In: *Computers & operations research* 13 (1986), Nr. 5, S. 633 – 645. [http://dx.doi.org/10.1016/0305-0548\(86\)90056-0](http://dx.doi.org/10.1016/0305-0548(86)90056-0). – DOI 10.1016/0305-0548(86)90056-0

- [BJMR05] BRESINA, John L. ; JÓNSSON, Ari K. ; MORRIS, Paul H. ; RAJAN, Kanna: Mixed-Initiative Activity Planning for Mars Rovers. In: *IJCAI Citeseer*, 2005, S. 1709–1710
- [BK05] BRIEL, Menkes van d. ; KAMBHAMPATI, Subbarao: Optiplan: Unifying IP-based and Graph-based Planning. In: *J. Artif. Intell. Res. (JAIR)* 24 (2005), S. 919–931
- [BKH08] BRECHER, Christian ; KEMPF, Tobias ; HERFS, Werner: Cognitive control technology for a self-optimizing robot based assembly cell. In: *ASME 2008 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* American Society of Mechanical Engineers, 2008, S. 1423–1431
- [BLE⁺07] BAINES, Tim S. ; LIGHTFOOT, Howard W. ; EVANS, Steve ; NEELY, Andy ; GREENOUGH, Richard ; PEPPARD, Joe ; ROY, Rajkumar ; SHEHAB, Essam ; BRAGANZA, Ashley ; TIWARI, Ashutosh u. a.: State-of-the-art in product-service systems. In: *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 221 (2007), Nr. 10, S. 1543–1552
- [BLR05] BEHRMANN, Gerd ; LARSEN, Kim G. ; RASMUSSEN, Jacob I.: Priced timed automata: Algorithms and Applications. In: BOER, Frank S. (Hrsg.) ; BONSANGUE, Marcello M. (Hrsg.) ; GRAF, Susanne (Hrsg.) ; ROEVER, Willem-Paul de (Hrsg.): *Formal Methods for Components and Objects* Bd. 3657, Springer Berlin Heidelberg, 2005 (Lecture Notes in Computer Science), S. 162 – 182
- [BLVH15] BÜHRER, Ulrich ; LEGAT, Christoph ; VOGEL-HEUSER, Birgit: Changeability of Manufacturing Automation Systems using an Orchestration Engine for Programmable Logic Controllers. In: DOLGUI, Alexandre (Hrsg.) ; SASIADEK, Jurek (Hrsg.) ; ZAREMBA, Marek (Hrsg.) ; IFAC (Veranst.): *15th IFAC Symposium on Information Control Problems in Manufacturing – INCOM 2015* Bd. 48. Ottawa, Kanada : Elsevier, Mai 2015 (IFAC-PapersOnLine 3), S. 1573 – 1579
- [BMUVH12] BAYRAK, Gokay ; MURR, Patrik J. ; ULEWICZ, Sebastian ; VOGEL-HEUSER, Birgit: Comparison of a transformed Matlab/Simulink model into the programming language CFC on different IEC 61131-3 PLC environments. In: *Emerging Technologies & Factory Automation (ETFA), 2012 IEEE 17th Conference on IEEE*, 2012, S. 1–8
- [BN95] BÄCKSTRÖM, Christer ; NEBEL, Bernhard: Complexity results for SAS+ planning. In: *Computational Intelligence* 11 (1995), November, Nr. 4, S. 625 – 655. <http://dx.doi.org/10.1111/j.1467-8640.1995.tb00052.x> – DOI 10.1111/j.1467-8640.1995.tb00052.x
- [BNWL08] BLAU, Benjamin ; NEUMANN, Dirk ; WEINHARDT, Christof ; LAMPARTER, Steffen: Planning and pricing of service mashups. In: *10th IEEE Conference on E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services*. Washington, DC, USA : IEEE, Juli 2008, S. 19–26

- [Bor14] BORGIA, Eleonora: The Internet of Things vision: Key features, applications and open issues. In: *Computer Communications* 54 (2014), S. 1–31
- [BR13] BACKHAUS, Julian ; REINHART, Gunther: Efficient application of task-oriented programming for assembly systems. In: *IEEE/ASME International Conference on Advanced Intelligent Mechatronics* IEEE, 2013, S. 750–755
- [BR15] BACKHAUS, J ; REINHART, G: Adaptive and device independent planning module for task-oriented programming of assembly systems. In: *Procedia CIRP* 33 (2015), S. 545–550. <http://dx.doi.org/10.1016/j.procir.2015.06.073>. – DOI 10.1016/j.procir.2015.06.073
- [Bra87] BRATMAN, Michael: Intention, plans, and practical reason. (1987)
- [Bra02] BRACHMAN, Ronald J.: Systems that know what they’re doing. In: *Intelligent Systems, IEEE* 17 (2002), Nr. 6, S. 67–71
- [BRV04] BERTHOMIEU, Bernard ; RIBET, P.-O. ; VERNADAT, François: The tool TINA – construction of abstract state spaces for Petri nets and time Petri nets. In: *International Journal of Production Research* 42 (2004), Februar, Nr. 14, S. 2741 – 2756. <http://dx.doi.org/10.1080/00207540412331312688>. – DOI 10.1080/00207540412331312688
- [BSP+16] BAREISS, Patrick ; SCHÜTZ, Daniel ; PRIEGO, Rafael ; MARCOS, Marga ; VOGEL-HEUSER, Birgit: A Model-based Failure Recovery Approach for automated Production Systems combining SysML and Industrial Standards. In: *21st IEEE Conference on Emerging Technologies and Factory Automation*, 2016
- [BT02] BOUSBIA, Salah ; TRENTESAUX, Damien: Self-organization in distributed manufacturing control: state-of-the-art and future trends. In: *Systems, Man and Cybernetics, 2002 IEEE International Conference on* Bd. 5 IEEE, 2002, S. 6–pp
- [BTW+15] BISCHOFF, Jürgen ; TAPHORN, Christoph ; WOLTER, Denise ; BRAUN, Nomo ; FELLBAUM, Manfred ; GOLOVEROV, Alexander ; LUDWIG, Stefan ; HEGMANN, Tobias ; PRASSE, Christian ; HENKE, Michael ; HOMPEL, Michael ten ; DÖBBELER, Frederik ; FUSS, Emanuel ; KIRSCH, Christopher ; MÄTTIG, Ben ; BRAUN, Stefan ; GUTH, Michael ; KASPERS, Mark ; SCHEFFLER, Doris: *Erschließen der Potenziale der Anwendung von "Industrie 4.0" im Mittelstand*. Juni 2015. – Studie im Auftrag des Bundesministeriums für Wirtschaft und Energie (BMWi)
- [Bun14] BUNDESMINISTERIUM FÜR BILDUNG UND FORSCHUNG: Innovationen für Deutschland: Die neue Hightech-Strategie Innovationen für Deutschland / Bundesministerium für Bildung und Forschung –Referat Grundsatzfragen der Innovationspolitik . 2014. – Broschüre
- [Bun15] BUNDESMINISTERIUM FÜR ARBEIT UND SOZIALES – ABTEILUNG GRUNDSATZFRAGEN DES SOZIALSTAATS, DER ARBEITSWELT UND DER SOZIALEN MARKTWIRTSCHAFT (Hrsg.): *Grünbuch – Arbeit weiter denken: Arbeiten 4.0*. 2015 www.arbeitenviernull.de

- [BVZ15] BITKOM E.V. (Hrsg.) ; VDMA E.V. (Hrsg.) ; ZVEI E.V. (Hrsg.): *Umsetzungsstrategie Industrie 4.0: Ergebnisbericht der Plattform Industrie 4.0*. 2015
- [Byl94] BYLANDER, Tom: The computational complexity of propositional STRIPS planning. In: *Artificial Intelligence* 69 (1994), Nr. 1, S. 165–204
- [CA95] CORBETT, James C. ; AVRUNIN, George S.: Using integer programming to verify general safety and liveness properties. In: *Formal Methods in System Design* 6 (1995), Nr. 1, S. 97–123
- [CB08] CHRPA, Lukás ; BARTÁK, Roman: Towards Getting Domain Knowledge: Plans Analysis through Investigation of Actions Dependencies. In: *FLAIRS Conference*, 2008, S. 531–536
- [CCO⁺12] COLES, Amanda ; COLES, Andrew ; OLAYA, Angel G. ; JIMÉNEZ, Sergio ; LÓPEZ, Carlos L. ; SANNER, Scott ; YOON, Sungwook: A survey of the seventh international planning competition. In: *AI Magazine* 33 (2012), Nr. 1, S. 83–88
- [CF10] CURRY, Guy L. ; FELDMAN, Richard M.: *Manufacturing systems modeling and Analysis*. Springer Science & Business Media, 2010
- [CFOG00a] CASTILLO, L ; FDEZ-OLIVARES, J ; GONZÁLEZ, A: Intelligent planning of Gifcet charts. In: *Robotics and Computer-Integrated Manufacturing* 16 (2000), Nr. 4, S. 225–239
- [CFOG00b] CASTILLO, L ; FDEZ-OLIVARES, J ; GONZÁLEZ, Antonio: Automatic generation of control sequences for manufacturing systems based on partial order planning techniques. In: *Artificial Intelligence in Engineering* 14 (2000), Nr. 1, S. 15–30
- [CGMP99] CLARKE, Edmund M. ; GRUMBERG, Orna ; MINEA, Marius ; PELED, Doron: State space reduction using partial order techniques. In: *International Journal on Software Tools for Technology Transfer* 2 (1999), Nr. 3, S. 279–287
- [CGT08] CANNATA, A ; GEROSA, M ; TAISCH, M: SOCRADES: A framework for developing intelligent systems in manufacturing. In: *Industrial Engineering and Engineering Management, 2008. IEEM 2008. IEEE International Conference on IEEE*, 2008, S. 1904–1908
- [CL94] CADOLI, Marco ; LENZERINI, Maurizio: The complexity of propositional closed world reasoning and circumscription. In: *Journal of Computer and System Sciences* 48 (1994), Nr. 2, S. 255–310
- [CMO12] CHRPA, Lukás ; MCCLUSKEY, Thomas L. ; OSBORNE, Hugh: Optimizing Plans through Analysis of Action Dependencies and Independencies. In: *ICAPS*, 2012
- [Coh04] COHN, Mike: *User stories applied: For agile software development*. Addison-Wesley Professional, 2004

- [CW85] CARDELLI, Luca ; WEGNER, Peter: On understanding types, data abstraction, and polymorphism. In: *ACM Computing Surveys (CSUR)* 17 (1985), Nr. 4, S. 471–523
- [CWH00] CASE, Keith ; WAN HARUN, WA: Feature-based representation for manufacturing planning. In: *International Journal of Production Research* 38 (2000), Nr. 17, S. 4285–4300
- [Dan90] DANTZIG, George B.: *Origins of the simplex method*. ACM, 1990
- [DDD⁺15] DIEGNER, Bernhard ; DIEMER, Johannes ; DÜMMEL, Mathias ; ERKER, Stefan ; FAY, Alexander ; HERFS, Werner ; HILGER, Claus ; JASPERNEITE, Jürgen ; KALHOFF, Johannes ; KUBACH, Uwe ; LÖWEN, Ulrich ; MENGES, Georg ; MICHELS, Jan S. ; MILDNER, Frank ; QUETSCHLICH, Mathias ; STEFFENS, Ernst-Joachim ; STIEDL, Thomas: *Industrie 4.0: Whitepaper FuE-Themen / Plattform Industrie 4.0*. 2015. – Forschungsbericht
- [Dec03] DECHTER, Rina: *Constraint processing*. Morgan Kaufmann, 2003
- [DH14] DRATH, Rainer ; HORCH, Alexander: Industrie 4.0: Hit or hype? In: *IEEE industrial electronics magazine* 8 (2014), Nr. 2, S. 56–58. <http://dx.doi.org/10.1109/MIE.2014.2312079>. – DOI 10.1109/MIE.2014.2312079
- [DIN87] DIN – DEUTSCHE INSTITUT FÜR NORMUNG E. V.: *DIN19226-1: Leittechnik, Regelungstechnik und Steuerungstechnik – Teil 1: Allgemeine Begriffe*. 1987
- [DIN98] DIN – DEUTSCHE INSTITUT FÜR NORMUNG E. V.: *DIN V 19233: Automatisierung mit Prozessrechenystemen – Begriffe*. 1998
- [DIN02] DIN – DEUTSCHE INSTITUT FÜR NORMUNG E. V.: *DIN EN 60848: GRAFCET, Spezifikationssprache für Funktionspläne der Ablaufsteuerung*. 2002
- [DJ01] DESEL, Jörg ; JUHÁS, Gabriel: What Is a Petri Net? Informal Answers for the Informed Reader. In: EHRIG, H. (Hrsg.) ; JUHÁS, J. G. and P. G. and Padberg (Hrsg.) ; ROZENBERG, G. (Hrsg.): *Unifying Petri Nets* Bd. 2128. Springer, 2001, S. 1–25
- [DK01] DO, Minh B. ; KAMBHAMPATI, Subbarao: Planning as constraint satisfaction: Solving the planning graph by compiling it into CSP. In: *Artificial Intelligence* 132 (2001), Nr. 2, S. 151–182
- [DK04] DEAN, Thomas ; KAMBHAMPATI, Subbarao: Planning and Scheduling. In: *Computer Science Handbook*. CRC Press, 2004
- [DL91] DURFEE, Edmund H. ; LESSER, Victor R.: Partial global planning: A coordination framework for distributed hypothesis formation. In: *Systems, Man and Cybernetics, IEEE Transactions on* 21 (1991), Nr. 5, S. 1167–1183

- [DL92] DECKER, Keith S. ; LESSER, Victor R.: Generalizing the partial global planning algorithm. In: *International Journal of Intelligent and Cooperative Information Systems* 1 (1992), Nr. 02, S. 319–346
- [DM06] DIMOPOULOS, Yannis ; MORAITIS, Pavlos: Multi-agent coordination and cooperation through classical planning. In: *Intelligent Agent Technology, 2006. IAT'06. IEEE/WIC/ACM International Conference on IEEE*, 2006, S. 398–402
- [DSSG⁺08] DE SOUZA, Luciana Moreira S. ; SPIESS, Patrik ; GUINARD, Dominique ; KÖHLER, Moritz ; KARNOUSKOS, Stamatis ; SAVIO, Domnic: Socrades: A web service based shop floor integration infrastructure. In: *The internet of things*. Springer, 2008, S. 50–67
- [DSSP09] DE SILVA, Lavindra ; SARDINA, Sebastian ; PADGHAM, Lin: First principles planning in BDI systems. In: *8th International Conference on Autonomous Agents and Multiagent Systems* Bd. 2 International Foundation for Autonomous Agents and Multiagent Systems, 2009, S. 1105–1112
- [DWTMW05] DE WEERDT, Mathijs ; TER MORS, Adriaan ; WITTEVEEN, Cees: Multi-agent planning: An introduction to planning and coordination. In: *In: Handouts of the European Agent Summer* Citeseer, 2005
- [DXHL14] DA XU, Li ; HE, Wu ; LI, Shancang: Internet of things in industries: A survey. In: *IEEE Transactions on Industrial Informatics* 10 (2014), Nr. 4, S. 2233–2243
- [EA12] EVANS, Peter C. ; ANNUNZIATA, Marco: Industrial Internet: Pushing the Boundaries of Minds and Machines. In: *World Economic Forum*, 2012
- [EABVGV⁺09] ESTÉVEZ-AYRES, Iria ; BASANTA-VAL, Pablo ; GARCÍA-VALLS, Marisol ; FISTEUS, Jesús A ; ALMEIDA, Luís: QoS-aware real-time composition algorithms for service-based applications. In: *Industrial Informatics, IEEE Transactions on* 5 (2009), Nr. 3, S. 278–288
- [EAGVABV08] ESTÉVEZ-AYRES, Iria ; GARCIA-VAILS, M ; ALMEIDA, Luís ; BASANTA-VAL, Pablo: Solutions for supporting composition of service-based real-time applications. In: *Object Oriented Real-Time Distributed Computing (ISORC), 2008 11th IEEE International Symposium on IEEE*, 2008, S. 42–49
- [EDB⁺15] EMMRICH, Volkhard ; DÖBELE, Mathias ; BAUERNHANSL, Thomas ; PAULUS-ROHMER, Dominik ; SCHATZ, Anja ; WESKAMP, Markus: Geschäftsmodell-Innovation durch Industrie 4.0 Chancen und Risiken für den Maschinen- und Anlagenbau / Dr. Wieselhuber & Partner GmbH und Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA. 2015. – Studie
- [EIPS08] EBEL, F. ; IDLER, S. ; PREDE, G. ; SCHOLZ, D.: *Grundlagen der Automatisierungstechnik*. Festo Didactic GmbH & Co. KG, Januar 2008
- [EJ06] EDELKAMP, Stefan ; JABBAR, Shahid: Action planning for directed model checking of Petri nets. In: *Electronic Notes in Theoretical Computer Science* 149 (2006), Nr. 2, S. 3–18

- [EKS11] EDELKAMP, Stefan ; KELLERSHOFF, Mark ; SULEWSKI, Damian: Program model checking via action planning. In: *Model Checking and Artificial Intelligence*. Springer, 2011, S. 32–51
- [EMO07] ESTÉVEZ, Elisabet ; MARCOS, Marga ; ORIVE, Darío: Automatic generation of PLC automation projects from component-based models. In: *The International Journal of Advanced Manufacturing Technology* 35 (2007), Nr. 5-6, S. 527–540
- [EN14] ELMARAGHY, Hoda ; NASSEHI, Aydin: Computer-Aided Process Planning. In: *CIRP Encyclopedia of Production Engineering*. Springer, 2014, S. 266–271
- [EP10] EBEL, Frank ; PANY, Markus: *Station Handhaben – Handbuch*. Festo Didactic GmbH & Co. KG, Juni 2010
- [Erl08] ERL, Thomas: *Soa: principles of service design*. Bd. 1. Prentice Hall Upper Saddle River, 2008
- [ERV02] ESPARZA, Javier ; RÖMER, Stefan ; VOGLER, Walter: An improvement of McMillan’s unfolding algorithm. In: *Formal Methods in System Design* 20 (2002), Nr. 3, S. 285–310
- [ES01] ESPARZA, Javier ; SCHRÖTER, Claus: Unfolding based algorithms for the reachability problem. In: *Fundamenta Informaticae* 47 (2001), Nr. 3-4, S. 231–245
- [ESJ14] EWERT, Daniel ; SCHILBERG, Daniel ; JESCHKE, Sabina: Selfoptimized assembly planning for a ROS based robot cell. In: *Automation, Communication and Cybernetics in Science and Engineering 2013/2014*. Springer, 2014, S. 609–618
- [Esp94] ESPARZA, Javier: Model checking using net unfoldings. In: *Science of Computer Programming* 23 (1994), Nr. 2, S. 151–195
- [ETK⁺13] EWERT, Daniel ; THELEN, Sebastian ; KUNZE, Ralph ; MAYER, Marcel ; SCHILBERG, Daniel ; JESCHKE, Sabina: A graph based hybrid approach of offline pre-planning and online re-planning for efficient assembly under realtime constraints. In: *Automation, Communication and Cybernetics in Science and Engineering 2011/2012*. Springer, 2013, S. 623–635
- [Ewe14] EWERT, Daniel: *Adaptive Ablaufplanung für die Fertigung in der Factory of the Future*, Rheinisch-Westfälische Technische Hochschule Aachen, Diss., 2014
- [Feh93] FEHLING, Rainer: A concept of hierarchical Petri nets with building blocks. In: *Advances in Petri Nets 1993*. Springer, 1993, S. 148–168
- [FL03] FOX, Maria ; LONG, Derek: PDDL2. 1: An Extension to PDDL for Expressing Temporal Planning Domains. In: *J. Artif. Intell. Res.(JAIR)* 20 (2003), S. 61–124
- [FLR⁺13] FELDMANN, Stefan ; LOSKYLL, Matthias ; RÖSCH, Susanne ; SCHLICK, Jochen ; ZÜHLKE, Detlef ; VOGEL-HEUSER, Birgit: Increasing Agility in Engineering

- and Runtime of Automated Manufacturing Systems. In: *IEEE International Conference on Industrial Technology*. Kapstadt, Südafrika : IEEE, Februar 2013, S. 1303 – 1308
- [FLW⁺12] FOEHR, Matthias ; LEITÃO, Paulo ; WAGNER, Thomas ; JAGER, T ; LÜDER, A: Integrating mechatronic thinking and multi-agent approaches. In: *Emerging Technologies & Factory Automation (ETFA), 2012 IEEE 17th Conference on IEEE*, 2012, S. 1–8
- [FM06] FREUDER, Eugene C. ; MACKWORTH, Alan K.: Constraint satisfaction: An emerging paradigm. In: ROSSI, Francesca (Hrsg.) ; VAN BEEK, Peter (Hrsg.) ; WALSH, Toby (Hrsg.): *Handbook of Constraint Programming*. Elsevier, 2006, Kapitel 2, S. 13–28
- [FMS14] FRÜH, K.F. ; MAIER, Uwe ; SCHAUDEL, Dieter: *Handbuch der Prozessautomatisierung*. 5. Deutscher Industrieverlag, 2014
- [FN72] FIKES, Richard E. ; NILSSON, Nils J.: STRIPS: A new approach to the application of theorem proving to problem solving. In: *Artificial intelligence 2 (1972)*, Nr. 3, S. 189–208
- [FPS11] FRANK, Ursula ; PAPENFORT, Josef ; SCHÜTZ, Daniel: Real-time capable software agents on IEC 61131 systems—Developing a tool supported method. In: *Proc. of the 18th IFAC World Congress. Mailand, Italien*, 2011
- [Fra14] FRANK, Timo: *Entwicklung und Evaluation einer Modellierungssprache für den Architekturf Entwurf von verteilten Automatisierungsanlagen auf Basis der Systems Modeling Language (SysML)*, Technische Universität München, Diss., 2014
- [Fre10] FREI, Regina: *Self-organisation in evolvable assembly systems*, New University of Lisbon, Diss., 2010
- [Fri09] FRIEDRICH, Andreas D.: *Anwendbarkeit von Methoden und Werkzeugen des konventionellen Softwareengineering zur Modellierung und Programmierung von Steuerungssystemen*. Bd. 2. kassel university press GmbH, 2009
- [FS11] FREI, Regina ; SERUGENDO, Giovanna Di M.: Self-organizing assembly systems. In: *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 41 (2011), Nr. 6, S. 885–897
- [GFLML13] GARETTI, Marco ; FUMAGALLI, Luca ; LOBOV, Andrei ; MARTINEZ LASTRA, Jose L.: Open automation of manufacturing systems through integration of ontology and web services. In: *Manufacturing Modelling, Management, and Control* Bd. 7, 2013, S. 198–203
- [GHJV94] GAMMA, Erich ; HELM, Richard ; JOHNSON, Ralph ; VLISSIDES, John: *Design patterns: elements of reusable object-oriented software*. Pearson Education, 1994

- [GHL⁺09] GEREVINI, Alfonso E. ; HASLUM, Patrik ; LONG, Derek ; SAETTI, Alessandro ; DIMOPOULOS, Yannis: Deterministic planning in the fifth international planning competition: PDDL3 and experimental evaluation of the planners. In: *Artificial Intelligence* 173 (2009), Nr. 5, S. 619–668
- [GL98] GELFOND, Michael ; LIFSCHITZ, Vladimir: Action languages. In: *Electronic Transactions on AI* 3 (1998), Nr. 16
- [GL05] GEREVINI, Alfonso ; LONG, Derek: Plan constraints and preferences in PDDL3. In: *The Language of the Fifth International Planning Competition. Tech. Rep. Technical Report, Department of Electronics for Automation, University of Brescia, Italy* 75 (2005)
- [GLS81] GRÖTSCH, Martin ; LOVÁSZ, László ; SCHRIJVER, Alexander: The ellipsoid method and its consequences in combinatorial optimization. In: *Combinatorica* 1 (1981), Nr. 2, S. 169–197
- [GNT04] GHALLAB, Malik ; NAU, Dana ; TRAVERSO, Paolo: *Automated planning: theory & practice*. Elsevier, 2004
- [GR83a] GOLTZ, Ursula ; REISIG, Wolfgang: The non-sequential behaviour of Petri nets. In: *Information and Control* 57 (1983), Nr. 2, S. 125–147
- [GR83b] GOLTZ, Ursula ; REISIG, Wolfgang: Processes of place/transition-nets. In: *Automata, Languages and Programming*. Springer, 1983, S. 264–277
- [Grö04] GRÖTSCH, Eberhard: *SPS-speicherprogrammierbare Steuerungen als Bausteine verteilter Automatisierung*. Oldenbourg Industrieverlag, 2004
- [Gro11] GROUP, ARC A.: PLC & PLC-based PAC Worldwide Outlook: Five year market analysis and technology forecast through 2016 / ARC Advisory Group. 2011. – Forschungsbericht
- [GS03] GEREVINI, Alfonso ; SERINA, Ivan: Planning as propositional CSP: from Walksat to local search techniques for action graphs. In: *Constraints* 8 (2003), Nr. 4, S. 389–413
- [GSLZ14] GORECKY, Dominic ; SCHMITT, Mathias ; LOSKYLL, Matthias ; ZÜHLKE, Detlef: Human-machine-interaction in the industry 4.0 era. In: *2014 12th IEEE International Conference on Industrial Informatics (INDIN)* IEEE, 2014, S. 289–294
- [GT96] GONDZIO, Jacek ; TERLAKY, Tamás: A computational view of interior point methods. In: *Advances in Linear and Integer Programming, Oxford Lecture Series* (1996), Nr. 4, S. 103–144
- [GVLV13] GARCIA VALLS, Marisol ; LÓPEZ, Iago R. ; VILLAR, L F.: iLAND: An enhanced middleware for real-time reconfiguration of service oriented distributed real-time systems. In: *Industrial Informatics, IEEE Transactions on* 9 (2013), Nr. 1, S. 228–236

- [HBL⁺08] HERRERA, V V. ; BEPPERLING, A ; LOBOV, Andrei ; SMIT, H ; COLOMBO, Armando W. ; LASTRA, J: Integration of multi-agent systems and service-oriented architecture for industrial automation. In: *Industrial Informatics, 2008. INDIN 2008. 6th IEEE International Conference on IEEE*, 2008, S. 768–773
- [Hel06] HELMERT, Malte: The Fast Downward Planning System. In: *J. Artif. Intell. Res.(JAIR)* 26 (2006), S. 191–246
- [HKB12] HEITMANN, Frank ; KÖHLER-BUSSMEIER, Michael: P-and T-Systems in the Nets-within-Nets-Formalism. In: *Application and Theory of Petri Nets*. Springer, 2012, S. 368–387
- [Hof17] HOFFMANN, Max: *Adaptive and Scalable Information Modeling to Enable Autonomous Decision Making for Real-Time Interoperable Factories*, RWTH Aachen University, Diss., 2017
- [Hoo88] HOOKER, John N.: A quantitative approach to logical inference. In: *Decision Support Systems* 4 (1988), Nr. 1, S. 45–69
- [HPE⁺10] HEINEN, T ; PETER, K ; ERLACH, K ; NYHUIS, P ; LANZA, G ; WESTKÄMPER, E: Zukunftsthemen der Fabrikplanung. Wohin geht der Trend? - Ergebnisse einer umfangreichen Online-Umfrage des VDI-Fachausschusses "Fabrikplanung". In: *Zeitschrift für wirtschaftlichen Fabrikbetrieb* 105 (2010), Nr. 5, S. 405–409
- [HPO16] HERMANN, Mario ; PENTEK, Tobias ; OTTO, Boris: Design Principles for Industrie 4.0 Scenarios. In: *2016 49th Hawaii International Conference on System Sciences (HICSS) IEEE*, 2016, S. 3928–3937
- [HPR13] HOFFMAN, Karla L. ; PADBERG, Manfred ; RINALDI, Giovanni: Traveling salesman problem. In: *Encyclopedia of Operations Research and Management Science*. Springer, 2013, S. 1573–1578
- [HRL12] HERRERA, Vladimir V. ; RAMOS, Axel V. ; LASTRA, José L M.: An agent-based system for orchestration support of web service-enabled devices in discrete manufacturing systems. In: *Journal of Intelligent Manufacturing* 23 (2012), Nr. 6, S. 2681–2702
- [HRTW07] HICKMOTT, Sarah L. ; RINTANEN, Jussi ; THIÉBAUX, Sylvie ; WHITE, Langford B.: Planning via Petri Net Unfolding. In: *IJCAI Bd. 7*, 2007, S. 1904–1911
- [HS09] HICKMOTT, Sarah L. ; SARDINA, Sebastian: Optimality Properties of Planning Via Petri Net Unfolding: A Formal Analysis. In: *ICAPS Citeseer*, 2009
- [HTM⁺14] HOLLER, Jan ; TSIATSI, Vlasios ; MULLIGAN, Catherine ; AVESAND, Stefan ; KARNOUSKOS, Stamatis ; BOYLE, David: *From Machine-to-machine to the Internet of Things: Introduction to a New Age of Intelligence*. Academic Press, 2014

- [HTS⁺16] HOFFMANN, Max ; THOMAS, Philipp ; SCHÜTZ, Daniel ; VOGEL-HEUSER, Birgit ; MEISEN, Tobias ; JESCHKE, Sabina: Semantic integration of multi-agent systems using an OPC UA information modeling approach. In: *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)* IEEE, 2016, S. 744–747
- [Huh03] HUHNS, Michael N.: Software agents: the future of web services. In: *Agent Technologies, Infrastructures, Tools, and Applications for E-Services*. Springer, 2003, S. 1–18
- [Huj12] HUIJBER, Max: *Modellierung und Implementierung eines neuartigen Softwarekonzepts für Speicherprogrammierbare Steuerungen (SPS)*, Lehrstuhl für Automatisierung und Informationssysteme, Technische Universität München, Bachelorarbeit, 2012
- [HVHB⁺16] HEHENBERGER, Peter ; VOGEL-HEUSER, Birgit ; BRADLEY, David ; EYNARD, B. ; TOMIYAMAE, T ; ACHICHEF, S: Design, modelling, simulation and integration of cyber physical systems: Methods and applications. In: *Computers in Industry* 82 (2016), Oktober, S. 273–289. <http://dx.doi.org/10.1016/j.compind.2016.05.006>. – DOI 10.1016/j.compind.2016.05.006
- [IEC09] IEC – INTERNATIONAL ELECTROTECHNICAL COMMISSION: *IEC 61131-3: Programmable Logic Controllers Part 3: Programming Languages*. 2009
- [IEC12] IEC – INTERNATIONAL ELECTROTECHNICAL COMMISSION: *IEC 61499: Function blocks*. 2012
- [IEC13a] IEC – INTERNATIONAL ELECTROTECHNICAL COMMISSION: *IEC 61131-3: Programmable Logic Controllers Part 3: Programming Languages* International Electrotechnical Commission, 2013
- [IEC13b] IEC – INTERNATIONAL ELECTROTECHNICAL COMMISSION: *Integration von Unternehmensführungs- und Leitsystemen — Teil 1: Modelle und Terminologie (IEC 62264-1:2013)*. 2013
- [IEC16] IEC – INTERNATIONAL ELECTROTECHNICAL COMMISSION: *IEC 62264-3: Enterprise-control system integration – Part 3: Activity models of manufacturing operations management*. 2016
- [JCJ92] JACOBSON, Ivar ; CHRISTERSON, Magnus ; JONSSON, Patrik: *Object-Oriented Software Engineering: A Use case Approach*. Addison-Wesley Longman, Amsterdam, 1992
- [Jen96] JENNINGS, Nick R.: Coordination techniques for distributed artificial intelligence. In: *Foundations of distributed artificial intelligence* (1996), S. 187–210
- [JLRM09] JAEHNE, DM ; LI, Miao ; RIEDEL, Ralph ; MUELLER, Egon: Configuring and operating global production networks. In: *International Journal of Production Research* 47 (2009), Nr. 8, S. 2013–2030

- [JM94] JAFFAR, Joxan ; MAHER, Michael J.: Constraint logic programming: A survey. In: *The journal of logic programming* 19 (1994), S. 503–581
- [JS05] JAMMES, François ; SMIT, Harm: Service-oriented architectures for devices—the SIRENA view. In: *Industrial Informatics, 2005. INDIN'05. 2005 3rd IEEE International Conference on IEEE*, 2005, S. 140–147
- [JSB11] JACOBSON, Ivar ; SPENCE, Ian ; BITTNER, Kurt: *USE-CASE 2.0: The guide to succeeding with Use Cases*. online verfügbar via <http://www.ivarjacobson.com>, Dezember 2011. – White paper
- [JSLD05] JAMMES, François ; SMIT, Harm ; LASTRA, Jose L M. ; DELAMER, Ivan M.: Orchestration of service-oriented manufacturing processes. In: *ETFA*, 2005
- [KAA⁺11] KOCH, Thorsten ; ACHTERBERG, Tobias ; ANDERSEN, Erling ; BASTERT, Oliver ; BERTHOLD, Timo ; BIXBY, Robert E. ; DANNA, Emilie ; GAMRATH, Gerald ; GLEIXNER, Ambros M. ; HEINZ, Stefan ; LODI, Andrea ; MITTELMANN, Hans ; RALPHS, Ted ; SALVAGNIN, Domenico ; STEFFY, Daniel E. ; WOLTER, Kati: MIPLIB 2010. In: *Mathematical Programming Computation* 3 (2011), Nr. 2, S. 103–163. <http://dx.doi.org/10.1007/s12532-011-0025-9>. – DOI 10.1007/s12532-011-0025-9
- [Kag15] KAGERMANN, Henning: Change Through Digitization – Value Creation in the Age of Industry 4.0. Version: 2015. http://dx.doi.org/10.1007/978-3-658-05014-6_2. In: *Management of Permanent Change*. Springer, 2015. – DOI 10.1007/978-3-658-05014-6_2, S. 23–45
- [Kam00] KAMBHAMPATI, Subbarao: Planning Graph as a (dynamic) CSP: Exploiting EBL, DDB and other CSP search techniques in Graphplan. In: *J. Artif. Intell. Res. (JAIR)* 12 (2000), S. 1–34
- [KB91] KLEIN, Inger ; BÄCKSTRÖM, Christer: On the planning problem in sequential control. In: *Decision and Control, 1991., Proceedings of the 30th IEEE Conference on IEEE*, 1991, S. 1819–1823
- [KB14] KÖHLER-BUSSMEIER, Michael: A survey of decidability results for elementary object systems. In: *Fundamenta Informaticae* 130 (2014), Nr. 1, S. 99–123
- [KBK07] KELLY, John-Paul ; BOTEÁ, Adi ; KOENIG, Sven: Planning with hierarchical task networks in video games. In: *Proceedings of the ICAPS-07 Workshop on Planning in Games*, 2007
- [Kem10] KEMPF, Tobias A.: *Ein kognitives Steuerungsframework für robotergestützte Handhabungsaufgaben*, Rheinisch-Westfälische Technische Hochschule Aachen, Diss., 2010
- [KHB09] KEMPF, Tobias ; HERFS, Werner ; BRECHER, Christian: SOAR-based sequence control for a flexible assembly cell. In: *Emerging Technologies & Factory Automation, 2009. ETFA 2009. IEEE Conference on IEEE*, 2009, S. 1–9

- [Kho03] KHOMENKO, Victor: *Model checking based on prefixes of petri net unfoldings*, Newcastle University, Diss., 2003
- [KL93] KLEIN, Inger ; LINDSKOG, Peter: Automatic creation of sequential control schemes in polynomial time. In: *Decision and Control, 1993., Proceedings of the 32nd IEEE Conference on IEEE*, 1993, S. 211–216
- [KMS96] KAUTZ, Henry ; MCALLESTER, David ; SELMAN, Bart: Encoding plans in propositional logic. In: *KR 96 (1996)*, S. 374–384
- [Kno94] KNOBLOCK, Craig A.: Generating parallel execution plans with a partial order Planner. In: *Proceedings of the 2nd International Conference on Artificial Intelligence Planning Systems*, 1994, S. 98 – 103
- [KS89] KOWALSKI, Robert ; SERGOT, Marek: A logic-based calculus of events. In: *Foundations of knowledge base management*. Springer, 1989, S. 23–55
- [KS92] KAUTZ, Henry A. ; SELMAN, Bart: Planning as Satisfiability. In: *ECAI Bd. 92*, 1992, S. 359–363
- [KSKR05] KÜSTER, Ulrich ; STERN, Mirco ; KÖNIG-RIES, Birgitta: A classification of issues and approaches in automatic service composition. In: *Proc. 1st Intl. Workshop on Engineering Service Compositions (WESC'05)*, 2005, S. 25–33
- [KV08] KLAAS, Volker ; VOLLMUTH, Jan: Funktechnologie in der Industrie – Flexible Produktionsabläufe durch den Einsatz von RFID. In: *Elektronik Praxis* (2008)
- [KW99] KAUTZ, Henry ; WALSER, Joachim P.: State-space planning by integer optimization. In: *AAAI/IAAI*, 1999, S. 526–533
- [Lap86] LAPORTE, Gilbert: Generalized subtour elimination constraints and connectivity constraints. In: *Journal of the Operational Research Society* (1986), S. 509–514
- [LaV06] LAVALLE, Steven M.: *Planning algorithms*. Cambridge university press, 2006
- [LBK15] LEE, Jay ; BAGHERI, Behrad ; KAO, Hung-An: A cyber-physical systems architecture for industry 4.0-based manufacturing systems. In: *Manufacturing Letters* 3 (2015), S. 18–23
- [LCK16] LEITÃO, Paulo ; COLOMBO, Armando W. ; KARNOUSKOS, Stamatis: Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges. In: *Computers in Industry* 81 (2016), 11–25. <http://dx.doi.org/10.1016/j.compind.2015.08.004>. – DOI 10.1016/j.compind.2015.08.004. – ISSN 01663615
- [LDXZ15] LI, Shancang ; DA XU, Li ; ZHAO, Shanshan: The internet of things: a survey. In: *Information Systems Frontiers* 17 (2015), Nr. 2, S. 243–259
- [Lee08] LEE, Edward A.: Cyber physical systems: Design challenges. In: *11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)* IEEE, 2008, S. 363–369

- [Lei08] LEITÃO, Paulo: Self-organization in manufacturing systems: challenges and opportunities. In: *Self-Adaptive and Self-Organizing Systems Workshops, 2008. SASOW 2008. Second IEEE International Conference on IEEE*, 2008, S. 174–179
- [Lei09] LEITÃO, Paulo: Agent-based distributed manufacturing control: A state-of-the-art survey. In: *Engineering Applications of Artificial Intelligence* 22 (2009), Nr. 7, S. 979–991
- [LFVH13] LEGAT, Christoph ; FOLMER, Jens ; VOGEL-HEUSER, Birgit: Evolution in Industrial Plant Automation: A Case Study. In: DIETRICH, Dietmar (Hrsg.) ; HUNG, John Y. (Hrsg.) ; LUO, Ren C. (Hrsg.) ; IEEE (Veranst.): *39th Annual Conference of the IEEE Industrial Electronics Society*. Wien, Österreich : IEEE, November 2013, S. 4386 – 4391
- [LG99] LAUBER, Rudolf ; GÖHNER, Peter: *Prozessautomatisierung 1*. Springer-Verlag, 1999
- [LHSS12] LOSKYLL, Matthias ; HECK, Ines ; SCHLICK, Jochen ; SCHWARZ, Michael: Context-based orchestration for control of resource-efficient manufacturing processes. In: *Future Internet* 4 (2012), Nr. 3, S. 737–761
- [Lit61] LITTLE, John D.: A proof for the queuing formula: $L = \lambda W$. In: *Operations research* 9 (1961), Nr. 3, S. 383–387
- [Lit11] LITTLE, John D.: Little’s Law as Viewed on Its 50th Anniversary. In: *Operations Research* 59 (2011), Nr. 3, S. 536–549
- [LKP⁺05] LÜDER, A ; KLOSTERMEYER, Axel ; PESCHKE, Jörn ; BRATOUKHINE, Alexei ; SAUTER, Thilo: Distributed automation: PABADIS versus HMS. In: *Industrial Informatics, IEEE Transactions on* 1 (2005), Nr. 1, S. 31–38
- [LKR⁺16] LEITÃO, Paulo ; KARNOUSKOS, Stamatis ; RIBEIRO, Luis ; LEE, Jay ; STRASSER, Thomas ; COLOMBO, Armando W.: Smart Agents in Industrial Cyber-Physical Systems. In: *Proceedings of the IEEE* 104 (2016), Mai, Nr. 5, 1086–1101. <http://dx.doi.org/10.1109/JPROC.2016.2521931>. – DOI 10.1109/JPROC.2016.2521931. – ISSN 0018–9219, 1558–2256
- [LKS⁺00] LONG, Derek ; KAUTZ, Henry ; SELMAN, Bart ; BONET, Blai ; GEFFNER, Hector ; KOEHLER, Jana ; BRENNER, Michael ; HOFFMANN, Jörg ; RITTINGER, Frank ; ANDERSON, Corin R. u. a.: The AIPS-98 planning competition. In: *AI magazine* 21 (2000), Nr. 2, S. 13
- [LMV13] LEITÃO, Paulo ; MARIK, Vladimir ; VRBA, Pavel: Past, present, and future of industrial agent applications. In: *Industrial Informatics, IEEE Transactions on* 9 (2013), Nr. 4, S. 2360–2372
- [Los13] LOSKYLL, M.: *Entwicklung einer Methodik zur dynamischen kontextbasierten Orchestrierung semantischer Feldgerätefunktionalitäten*, Fachbereich Maschinenbau und Verfahrenstechnik, Technische Universität Kaiserslautern, Diss., 2013

- [LPS⁺04] LÜDER, Arndt ; PESCHKE, Jörn ; SAUTER, Thilo ; DETER, Steffen ; DIEP, Daniel: Distributed intelligence for plant automation based on multi-agent systems: the PABADIS approach. In: *Production Planning & Control* 15 (2004), Nr. 2, S. 201–212
- [LRB⁺15] LEITÃO, Paulo ; RODRIGUES, Nelson ; BARBOSA, José ; TURRIN, Claudio ; PAGANI, Arnaldo: Intelligent products: The GRACE experience. In: *Control Engineering Practice* 42 (2015), September, 95–105. <http://dx.doi.org/10.1016/j.conengprac.2015.05.001>. – DOI 10.1016/j.conengprac.2015.05.001. – ISSN 09670661
- [LSF⁺13] LEGAT, Christoph ; SCHÜTZ, Daniel ; FELDMANN, Stefan ; LAMPARTER, Steffen ; SEITZ, Christian ; VOGEL-HEUSER, Birgit: Wandlungsfähige Automation auf Knopfdruck. In: *atp edition - Automatisierungstechnische Praxis* (2013), Nr. 5, S. 30 – 39
- [LSF⁺14] LEGAT, Christoph ; STEDEN, Frank ; FELDMANN, Stefan ; WEYRICH, Michael ; VOGEL-HEUSER, Birgit: Co-Evolution and Reuse of Automation Control and Simulation Software: Identification and Definition of Modification Actions and Strategies. In: *40th Annual Conference of the IEEE Industrial Electronics Society (IECON 2014)*. Dallas, USA, November 2014, S. 2525–2531
- [LSH⁺11] LOSKYLL, Matthias ; SCHLICK, Jochen ; HODEK, Stefan ; OLLINGER, Lisa ; GERBER, Tobias ; PIRVU, Bogdan: Semantic Service Discovery and Orchestration for Manufacturing Processes. In: *Proceedings of the 14th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2011)* (2011)
- [LSVH12] LEGAT, Christoph ; SCHÜTZ, Daniel ; VOGEL-HEUSER, Birgit: Automatic Generation of Field Control Strategies for Supporting (Re-)Engineering of Manufacturing Systems. In: BORANGIU, Theodor (Hrsg.) ; DOLGUI, Alexandre (Hrsg.) ; DUMITRACHE, Ioan (Hrsg.) ; FILIP, Florin G. (Hrsg.) ; IFAC (Veranst.): *14th IFAC Symposium on Information Control Problems in Manufacturing (INCOM 2012)* Bd. 14. Bukarest, Romänien : IFAC, Mai 2012 (Information Control Problems in Manufacturing 1), S. 1574 – 1579
- [LSVH13] LEGAT, Christoph ; SCHÜTZ, Daniel ; VOGEL-HEUSER, Birgit: Automatic Generation of Field Control Strategies fir Supporting (Re-)Engineering of Manufacturing Systems. In: *Journal of Intelligent Manufacturing* (2013), March. – published online
- [LSVH14] LEGAT, Christoph ; SCHÜTZ, Daniel ; VOGEL-HEUSER, Birgit: Wandlungsfähige Automation auf Knopfdruck. In: *gwi – Gaswärme international* 63 (2014), S. 63 – 72
- [Lun03] LUNZE, Jan ; (Hrsg.): *Automatisierungstechnik – Methoden für die Überwachung und Steuerung kontinuierlicher und ereignisdiskreter Systeme*. Oldenbourg Wissenschaftsverlag, 2003

- [LVH13] LEGAT, Christoph ; VOGEL-HEUSER, Birgit: A Multi-agent Architecture for Compensating Unforeseen Failures on Field Control Level. In: *3rd International Workshop on Service Orientation in Holonic and Multi Agent Manufacturing and Robotics*, 2013
- [LVH14a] LEGAT, Christoph ; VOGEL-HEUSER, Birgit: A Multi-agent Architecture for Compensating Unforeseen Failures on Field Control Level. In: BORANGIU, Theodor (Hrsg.) ; THOMAS, André (Hrsg.) ; TRENTESAUX, Damien (Hrsg.): *Service Orientation in Holonic and Multi-agent Manufacturing and Robotics* Bd. 544. Springer, 2014, S. 195 – 208
- [LVH14b] LEGAT, Christoph ; VOGEL-HEUSER, Birgit: An Orchestration Engine for Service-oriented Field Level Automation Software. In: *4th Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing (Sohoma 2014)*. Nancy, France, November 2014
- [LVH15] In: LEGAT, Christoph ; VOGEL-HEUSER, Birgit: *An Orchestration Engine for Services-Oriented Field Level Automation Software*. Berlin, Germany : Springer, Berlin, Germany, 2015, S. 71 – 80
- [LW94] LISKOV, Barbara H. ; WING, Jeannette M.: A behavioral notion of subtyping. In: *ACM Transactions on Programming Languages and Systems (TOPLAS)* 16 (1994), Nr. 6, S. 1811–1841
- [LZVM11] LEPUSCHITZ, Wilfried ; ZOITL, Alois ; VALLÉE, Mathieu ; MERDAN, Munir: Toward Self-Reconfiguration of Manufacturing Systems Using Automation Agents. In: *IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews* 41 (2011), January, Nr. 1, S. 52 – 69
- [Mal99] MALONE, Thomas W.: Is' empowerment'just a fad? Control, decision-making, and information technology. In: *BT Technology Journal* 17 (1999), Nr. 4, S. 141–144. <http://dx.doi.org/10.1023/A:1009663512936>. – DOI 10.1023/A:1009663512936
- [Mau06] MAUSER, Sebastian: *Semantiken von Petrinetzen – ein algebraischer Ansatz, der zwischen nebenläufigem und synchronem Verhalten unterscheidet*, Katholische Universität Eichstätt-Ingolstadt, Diplomarbeit, 2006
- [McD00] MCDERMOTT, Drew M.: The 1998 AI planning systems competition. In: *AI magazine* 21 (2000), Nr. 2, S. 35
- [McM93] MCMILLAN, Kenneth L.: Using unfoldings to avoid the state explosion problem in the verification of asynchronous circuits. In: *Computer Aided Verification* Springer, 1993, S. 164–177
- [Mec13] MECHS, Sebastian: *Model-based Engineering for Energy-Efficient Operation of Factory Automation Systems within Unproductive Phases*, Institut für Informatik, Technische Universität Clausthal, Diss., 2013

- [MEJK11] MARCOS, Marga ; ESTÉVEZ, Elisabet ; JOUVRAY, Christophe ; KUNG, Antonio: An Approach to use MDE in Dynamically Reconfigurable Networked Embedded SOAs. In: *Proceedings of the 18th IFAC World Congress*, 2011, S. 14946–14951
- [MF90] MITTAL, Sanjay ; FALKENHAINER, Brian: Dynamic constraint satisfaction. In: *Proceedings Eighth National Conference on Artificial Intelligence*, 1990, S. 25–32
- [MFH09] MEYER, Gerben G. ; FRÄMLING, Kary ; HOLMSTRÖM, Jan: Intelligent products: A survey. In: *Computers in industry* 60 (2009), Nr. 3, S. 137–148
- [MGG98] MARRI, HB ; GUNASEKARAN, A ; GRIEVE, RJ: Computer-aided process planning: a state of art. In: *The International Journal of Advanced Manufacturing Technology* 14 (1998), Nr. 4, S. 261–268
- [MGWH13] MCFARLANE, Duncan ; GIANNIKAS, Vaggelis ; WONG, Alex C. ; HARRISON, Mark: Product intelligence in industrial control: Theory and practice. In: *Annual Reviews in Control* 37 (2013), Nr. 1, S. 69–88. <http://dx.doi.org/10.1016/j.arcontrol.2013.03.003>. – DOI 10.1016/j.arcontrol.2013.03.003
- [MH69] MCCARTHY, John ; HAYES, Patrick J.: Some philosophical problems from the standpoint of artificial intelligence. In: *Readings in artificial intelligence* (1969), S. 431–450
- [MKB⁺16] MONOSTORI, L ; KADAR, B ; BAUERNHANSL, T ; KONDOH, S ; KUMARA, S ; REINHART, G ; SAUER, O ; SCHUH, G ; SIHN, W ; UEDA, K: Cyber-physical systems in manufacturing. In: *CIRP Annals-Manufacturing Technology* (2016)
- [MKMO11] MARRON, Pedro J. ; KARNOUSKOS, Stamatis ; MINDER, Daniel ; OLLERO, Aníbal: *The emerging domain of Cooperating Objects*. Springer Science & Business Media, 2011
- [ML07] MAŘÍK, Vladimír ; LAŽANSKÝ, Jiří: Industrial applications of agent technologies. In: *Control Engineering Practice* 15 (2007), Nr. 11, S. 1364–1380
- [MLCR12] MENDES, J. M. ; LEITAO, Paulo ; COLOMBO, Armando W. ; RESTIVO, Francisco: High-level Petri nets for the process description and Control in service-oriented manufacturing systems. In: *International Journal of Production Research* 50 (2012), S. 1650 – 1665
- [MLD09] MAHNKE, Wolfgang ; LEITNER, Stefan-Helmut ; DAMM, Matthias: *OPC unified architecture*. Springer Science & Business Media, 2009
- [MLM⁺06] MACKENZIE, C M. ; LASKEY, Ken ; MCCABE, Francis ; BROWN, Peter F. ; METZ, Rebekah ; HAMILTON, Booz A.: Reference model for service oriented architecture 1.0. In: *OASIS Standard* 12 (2006)
- [MLPM13a] MECHS, Sebastian ; LAMPARTER, Steffen ; PESCHKE, Jörn ; MÜLLER, Jörg P: Efficient Identification of Energy-Optimal Switching and Operating Sequences for Modular Factory Automation Systems. In: *Recent Trends in Applied Artificial Intelligence*. Springer, 2013, S. 202–211

- [MLPM13b] MECHS, Sebastian ; LAMPARTER, Steffen ; PESCHKE, Jörn ; MÜLLER, Jörg P: Start-Stopp-Automatik für Nicht-Produktivphasen: Höhere Energieeffizienz in Automatisierungssystemen. In: *atp edition-Automatisierungstechnische Praxis* 55 (2013), Nr. 06, S. 32–39
- [MLRC09] MENDES, J M. ; LEITÃO, Paulo ; RESTIVO, Francisco ; COLOMBO, Armando W.: Service-oriented agents for collaborative industrial automation and production systems. In: *Holonic and Multi-Agent Systems for Manufacturing*. Springer, 2009, S. 13–24
- [MLRC10] MENDES, J M. ; LEITÃO, Paulo ; RESTIVO, Francisco ; COLOMBO, Armando W.: Composition of Petri nets models in service-oriented industrial automation. In: *Industrial Informatics (INDIN), 2010 8th IEEE International Conference on IEEE*, 2010, S. 578–583
- [MMLP12] MECHS, Sebastian ; MULLER, JP ; LAMPARTER, Steffen ; PESCHKE, Jörn: Networked priced timed automata for energy-efficient factory automation. In: *American Control Conference (ACC), 2012 IEEE*, 2012, S. 5310–5317
- [MNS96] MÄNTYLÄ, Martti ; NAU, Dana ; SHAH, Jami: Challenges in feature-based manufacturing research. In: *Communications of the ACM* 39 (1996), Nr. 2, S. 77–85
- [Mon14] MONOSTORI, László: Cyber-physical production systems: roots, expectations and R&D challenges. In: *Procedia CIRP* 17 (2014), S. 9–13. <http://dx.doi.org/10.1016/j.procir.2014.03.115>. – DOI 10.1016/j.procir.2014.03.115
- [MP95] MCMILLAN, Kenneth L. ; PROBST, David K.: A technique of state space search based on unfolding. In: *Formal Methods in System Design* 6 (1995), Nr. 1, S. 45–65
- [MP11] METZGER, Mieczyslaw ; POLAKOW, Grzegorz: A survey on applications of agent technology in industrial process control. In: *Industrial Informatics, IEEE Transactions on* 7 (2011), Nr. 4, S. 570–581
- [MS09] MAY, Constantin ; SCHIMEK, Peter: *Total productive management: Grundlagen und Einführung von TPM-oder wie Sie Operational Excellence erreichen*. CETPM Publ., 2009
- [MSC+03] MCFARLANE, Duncan ; SARMA, Sanjay ; CHIRN, Jin L. ; WONG, ChienYaw ; ASHTON, Kevin: Auto ID systems and intelligent manufacturing control. In: *Engineering Applications of Artificial Intelligence* 16 (2003), Nr. 4, S. 365–376. [http://dx.doi.org/10.1016/S0952-1976\(03\)00077-0](http://dx.doi.org/10.1016/S0952-1976(03)00077-0). – DOI 10.1016/S0952-1976(03)00077-0
- [MSDPC12] MIORANDI, Daniele ; SICARI, Sabrina ; DE PELLEGRINI, Francesco ; CHLAMTAC, Imrich: Internet of things: Vision, applications and research challenges. In: *Ad Hoc Networks* 10 (2012), Nr. 7, S. 1497–1516

- [MUK00] MEHRABI, Mostafa G. ; ULSOY, A G. ; KOREN, Yoram: Reconfigurable manufacturing systems: key to future manufacturing. In: *Journal of Intelligent manufacturing* 11 (2000), Nr. 4, S. 403–419
- [MW01] MOSEMANN, Heiko ; WAHL, Friedrich M.: Automatic decomposition of planned assembly sequences into skill primitives. In: *Robotics and Automation, IEEE Transactions on* 17 (2001), Nr. 5, S. 709–718
- [MZCM04] MENEGUZZI, Felipe R. ; ZORZO, Avelino F. ; COSTA MÓRA, Michael da: Propositional planning in BDI agents. In: *Proceedings of the 2004 ACM symposium on Applied computing* ACM, 2004, S. 58–63
- [NAI⁺03] NAU, Dana S. ; AU, Tsz-Chiu ; ILGHAMI, Okhtay ; KUTER, Ugur ; MURDOCK, J W. ; WU, Dan ; YAMAN, Fusun: SHOP2: An HTN planning system. In: *J. Artif. Intell. Res.(JAIR)* 20 (2003), S. 379–404
- [NBD10] NISSIM, Raz ; BRAFMAN, Ronen I. ; DOMSHLAK, Carmel: A general, fully distributed multi-agent planning algorithm. In: *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1* International Foundation for Autonomous Agents and Multiagent Systems, 2010, S. 1323–1330
- [NG04] NAU, Dana ; GHALLAB, Malik: Measuring the Performance of Automated Planning Systems. In: *Performance Metrics for Intelligent Systems Workshop (PerMIS)*, 2004
- [NGR95] NAU, Dana S. ; GUPTA, Satyandra K. ; REGLI, William C.: Manufacturing-operation planning versus AI Planning. In: *AAAI Spring Symposium, AAAI*, 1995. – Technical Report SS-95-04
- [NHR⁺08] NYHUIS, P ; HEINEN, T ; REINHART, G ; RIMPAU, C ; ABELE, E ; WÖRN, A: Wandlungsfähige Produktionssysteme – Theoretischer Hintergrund zur Wandlungsfähigkeit von Produktionssystemen. In: *wt Werkstattstechnik online* 98 (2008), Nr. 9, S. 8 – 10
- [NHS⁺15] NIGGEMANN, Oliver ; HENNING, Steffen ; SCHRIEGEL, Sebastian ; OTTO, Jens ; ANIS, Anas: Models for adaptable automation software-an overview of plug-and-produce in industrial automation. In: *Modellbasierte Entwicklung eingebetteter Systeme (MBEES)*. Dagstuhl, Deutschland, 2015
- [NK01] NGUYEN, XuanLong ; KAMBHAMPATI, Subbarao: Reviving partial order planning. In: *IJCAI* Bd. 1, 2001, S. 459–464
- [NK05] NGUYEN, Xuan T. ; KOWALCZYK, Ryszard: Enabling agent-based management of Web services with WS2JADE. In: *Quality Software, 2005.(QSIC 2005). Fifth International Conference on IEEE*, 2005, S. 407–412
- [NKN02] NGUYEN, XuanLong ; KAMBHAMPATI, Subbarao ; NIGENDA, Romeo S.: Planning graph as the basis for deriving heuristics for plan synthesis by state space and CSP search. In: *Artificial Intelligence* 135 (2002), Nr. 1, S. 73–123

- [NL05] NEWCOMER, Eric ; LOMOW, Greg: *Understanding SOA with Web services*. Addison-Wesley, 2005
- [OAS09] OASIS – ORGANIZATION FOR THE ADVANCEMENT OF STRUCTURED INFORMATION STANDARDS ; DRISCOLL, Dan (Hrsg.) ; MENSCH, Antoine (Hrsg.): *Devices Profile for Web Services Version 1.1*. Juli 2009
- [OK15] O’HALLORAN, Derek ; KVOCHKO, Elena: Industrial Internet of Things: Unleashing the Potential of Connected Products and Services / World Economic Forum. 2015 (REF 020315). – Research Agenda
- [OSH11] OLLINGER, Lisa ; SCHLICK, Jochen ; HODEK, Stefan: Leveraging the agility of manufacturing chains by combining process-oriented production planning and service-oriented manufacturing automation. In: *Proceedings of the 18th IFAC World Congress*, 2011
- [OSL11] ONORI, Mauro ; SEMERE, Daniel ; LINDBERG, Bengt: Evolvable systems: an approach to self-X production. In: *International Journal of Computer Integrated Manufacturing* 24 (2011), Nr. 5, S. 506–516
- [Pö11] PÖRNBACHER, C.: *Modellgetriebene Entwicklung der Steuerungssoftware automatisierter Fertigungssysteme*, Technische Universität München, Diss., 2011
- [PAE⁺15] PRIEGO, R ; AGIRRE, A ; ESTÉVEZ, E ; ORIVE, D ; MARCOS, M: Middleware-based Support for Reconfigurable Mechatronic Systems. In: *IFAC-PapersOnLine* 48 (2015), Nr. 10, S. 81–86
- [Par96] PARUNAK, H Van D.: Applications of distributed artificial intelligence in industry. In: *Foundations of distributed artificial intelligence* 2 (1996)
- [PD06] POSSEL-DÖLKEN, Frank: *Projektierbares Multiagentensystem für die Ablaufsteuerung in der flexiblen automatisierten Fertigung*, Rheinisch-Westfälische Technische Hochschule Aachen, Diss., 2006
- [PD11] PSAIER, Harald ; DUSTDAR, Schahram: A survey on self-healing systems: approaches and systems. In: *Computing* 91 (2011), Nr. 1, S. 43–73
- [PE01] PADBERG, Julia ; EHRIG, Hartmut: Parameterized Net Classes: A Uniform Approach to Petri Net Classes. In: *Unifying Petri Nets*. Springer, 2001, S. 173–229
- [Ped86] PEDNAULT, Edwin P.: Formulating multiagent, dynamic-world problems in the classical planning framework. In: *Reasoning about actions and plans* (1986), S. 47–82
- [Ped89] PEDNAULT, Edwin: ADL: Exploring the middle ground between STRIPS and the situation calculus. In: *Proceedings of the first international conference on Principles of knowledge representation and reasoning* Morgan Kaufmann Publishers Inc., 1989, S. 324–332

- [Ped94] PEDNAULT, Edwin P.: ADL and the state-transition model of action. In: *Journal of logic and computation* 4 (1994), Nr. 5, S. 467–512
- [Pee05] PEER, Joachim: Web service composition as AI planning—a survey. In: *University of St. Gallen* (2005)
- [Pel93] PELED, Doron: All from one, one for all: on model checking using representatives. In: *Computer Aided Verification* Springer, 1993, S. 409–423
- [Pet62] PETRI, Carl A.: *Kommunikation mit Automaten*, Universität Bonn, Diss., 1962
- [Pin99] PINE, B J.: *Mass customization: the new frontier in business competition*. Harvard Business Press, 1999
- [PLML13] PUTTONEN, Juha ; LOBOV, Andrei ; MARTINEZ LASTRA, JL: Semantics-based composition of factory automation processes encapsulated by web services. In: *IEEE Transactions on Industrial Informatics* 9 (2013), Nr. 4, S. 2349–2359
- [PRK⁺15] PAELKE, Volker ; RÖCKER, Carsten ; KOCH, Nils ; FLATT, Holger ; BÜTTNER, Sebastian: User interfaces for cyber-physical systems. In: *at - Automatisierungstechnik* 63 (2015), Nr. 10, S. 833–843. <http://dx.doi.org/10.1515/auto-2015-0016>. – DOI 10.1515/auto-2015-0016
- [Pro12] PROJEKTGRUPPE "M2M INITIATIVE DEUTSCHLAND" DER AG2 "DIGITALE INFRASTRUKTUREN ALS ENABLER FÜR INNOVATIVE ANWENDUNGEN": Machine-to-Machine-Kommunikation – eine Chance für die deutsche Industrie / Bundesministerium für Wirtschaft und Technologie. 2012. – Forschungsbericht
- [PS05] PANKRATIUS, Victor ; STUCKY, Wolffried: A formal foundation for workflow composition, workflow view definition, and workflow normalization based on petri nets. In: *Proceedings of the 2nd Asia-Pacific conference on Conceptual modelling - Volume 43* Australian Computer Society, Inc., 2005, S. 79–88
- [PSA⁺14] PFROMMER, Julius ; STOGL, Denis ; ALEKSANDROV, Kiril ; SCHUBERT, Viktor ; HEIN, Bjorn: Modelling and orchestration of service-based manufacturing systems via skills. In: *Emerging Technology and Factory Automation (ETFA), 2014 IEEE* IEEE, 2014, S. 1–4
- [PSA⁺15] PFROMMER, Julius ; STOGL, Denis ; ALEKSANDROV, Kiril ; NAVARRO, Stefan E. ; HEIN, Björn ; BEYERER, Jürgen: Plug & produce by modelling skills and service-oriented orchestration of reconfigurable manufacturing systems. In: *at - Automatisierungstechnik* (2015), S. 790–800
- [PSB13] PFROMMER, Julius ; SCHLEIPEN, Miriam ; BEYERER, Jürgen: PPRS: Production skills and their relation to product, process, and resource. In: *Emerging Technologies & Factory Automation (ETFA), 2013 IEEE 18th Conference on IEEE*, 2013, S. 1–4

- [PSVHM15] PRIEGO, Rafael ; SCHUTZ, Daniel ; VOGEL-HEUSER, Birgit ; MARCOS, Marga: Reconfiguration architecture for updates of automation systems during operation. In: *Emerging Technologies & Factory Automation (ETFA), 2015 IEEE 20th Conference on IEEE*, 2015, S. 1–8
- [PW92] PENBERTHY, J S. ; WELD, Daniel S.: UCPOP: A Sound, Complete, Partial Order Planner for ADL. In: *Kr 92 (1992)*, S. 103–114
- [RAS08] RAMOS, Carlos ; AUGUSTO, Juan C. ; SHAPIRO, Daniel: Ambient intelligence – the next step for artificial intelligence. In: *IEEE Intelligent Systems 23 (2008)*, Nr. 2, S. 15–18
- [RBC08] RIBEIRO, Luis ; BARATA, Jose ; COLOMBO, Armando: MAS and SOA: A case study exploring principles and technologies to support self-properties in assembly systems. In: *Self-Adaptive and Self-Organizing Systems Workshops, 2008. SASOW 2008. Second IEEE International Conference on IEEE*, 2008, S. 192–197
- [RBM08] RIBEIRO, Luís ; BARATA, José ; MENDES, Pedro: MAS and SOA: complementary automation paradigms. In: *Innovation in manufacturing networks*. Springer, 2008, S. 259–268
- [RBP⁺91] RUMBAUGH, James ; BLAHA, Michael ; PREMERLANI, William ; EDDY, Frederick ; LORENSEN, William E. u. a.: *Object-oriented modeling and design*. Prentice-hall Englewood Cliffs, NJ, 1991
- [RC10] REN, Wei ; CAO, Yongcan: *Distributed coordination of multi-agent networks: emergent problems, models, and issues*. Springer Science & Business Media, 2010
- [RCD⁺04] RAMANI, Karthik ; CUNNINGHAM, Robert ; DEVANATHAN, Srikanth ; SUBRAMANIAM, Jayanti ; PATWARDHAN, Harshal: Technology review of mass customization. In: *International Conference on Economic, Technical and Organisational aspects of Product Configuration Systems, June Citeseer*, 2004, S. 28–29
- [ReC16] RECAM KONSORTIUM: ReCaM – Rapid reconfiguration of flexible Production Systems. Version: September 2016. <http://www.recam-project.eu>. 2016 (1). – Newsletter
- [Rei01] REITER, Raymond: *Knowledge in action: logical foundations for specifying and implementing dynamical systems*. MIT press, 2001
- [Rem17] REMENTERIA, Rafael P.: *A Model-based Approach for supporting Flexible Automation Production Systems and an Agent-based Implementation*, University of the Basque Country, Diss., 2017
- [RG95] RAO, Anand S. ; GEORGEFF, Michael P.: BDI agents: From theory to practice. In: *ICMAS Bd. 95, 1995*, S. 312–319

- [RGI⁺14] RAMIS, Borja ; GONZALEZ, Luis ; IAROVYI, Sergii ; LOBOV, Andrei ; MARTINEZ LASTRA, Jose L. ; VYATKIN, Valeriy ; DAI, William: Knowledge-based web service integration for industrial automation. In: *Industrial Informatics (INDIN), 2014 12th IEEE International Conference on IEEE*, 2014, S. 733–739
- [RHN04] RINTANEN, Jussi ; HELJANKO, Keijo ; NIEMELÄ, Ilkka: Parallel encodings of classical planning as satisfiability. In: *Logics in Artificial Intelligence*. Springer, 2004, S. 307–319
- [RHN06] RINTANEN, Jussi ; HELJANKO, Keijo ; NIEMELÄ, Ilkka: Planning as satisfiability: parallel plans and algorithms for plan search. In: *Artificial Intelligence* 170 (2006), Nr. 12, S. 1031–1080
- [RLSS10] RAJKUMAR, Raguathan R. ; LEE, Insup ; SHA, Lui ; STANKOVIC, John: Cyber-physical systems: the next computing revolution. In: *47th Design Automation Conference ACM*, 2010, S. 731 – 736
- [RM09] RAIBULET, Claudia ; MASCIADRI, Laura: Evaluation of dynamic adaptivity through metrics: an achievable target? In: *Joint Working IEEE/IFIP Conference on Software Architecture & European Conference on Software Architecture IEEE*, 2009, S. 341–344
- [RM10] RAIBULET, Claudia ; MASCIADRI, Laura: Metrics for the Evaluation of Adaptivity Aspects in Software Systems. In: *International Journal on Advances in Software* 3 (2010), Nr. 1 & 2, S. 238–251
- [RN95] RUSSELL, Stuart ; NORVIG, Peter: *Artificial intelligence: a modern approach*. Prentice-Hall, Englewood Cliffs, 1995
- [RS05] RAO, Jinghai ; SU, Xiaomeng: A survey of automated web service composition methods. In: *Semantic Web Services and Web Process Composition*. Springer, 2005, S. 43–54
- [RVBW06] ROSSI, Francesca ; VAN BEEK, Peter ; WALSH, Toby: *Handbook of constraint programming*. Elsevier, 2006
- [RW89] RAMADGE, Peter J. ; WONHAM, W M.: The control of discrete event systems. In: *Proceedings of the IEEE* 77 (1989), Nr. 1, S. 81–98
- [Rze03] RZEWSKI, George: On conceptual design of intelligent mechatronic systems. In: *Mechatronics* 13 (2003), Nr. 10, S. 1029–1044
- [San10] SANNER, Scott: Relational dynamic influence diagram language (rddl): Language description. In: *Unpublished ms. Australian National University* (2010)
- [SBF07] SECCHI, Cristian ; BONFE, Marcello ; FANTUZZI, Cesare: On the use of UML for modeling mechatronic systems. In: *Automation Science and Engineering, IEEE Transactions on* 4 (2007), Nr. 1, S. 105–113

- [Sch05] SCHMECK, Hartmut: Organic computing—a new vision for distributed embedded systems. In: *Object-Oriented Real-Time Distributed Computing, 2005. ISORC 2005. Eighth IEEE International Symposium on IEEE*, 2005, S. 201–203
- [Sch12] SCHEER, August-Wilhelm: *CIM. Computer Integrated Manufacturing: Towards the Factory of the Future*. Springer Science & Business Media, 2012
- [Sch14] SCHÜTZ, Daniel: *Automatische Generierung von Softwareagenten für die industrielle Automatisierungstechnik der Steuerungsebene des Maschinen- und Anlagenbaus auf Basis der Systems Modeling Language*, Technische Universität München, Diss., 2014
- [SCK00] SILVA, Fabiano ; CASTILHO, Marcos A. ; KÜNZLE, Luis A.: Petriplan: a new algorithm for plan generation (preliminary report). In: *Advances in Artificial Intelligence*. Springer, 2000, S. 86–95
- [SCSK12] SCHREINER, Marcos A. ; CASTILHO, Marcos A. ; SILVA, Fabiano ; KUNZLE, Luis A.: The Planning Net: Exploring the Petri Net Flow to Improve Planning Solvers. In: *Advances in Artificial Intelligence—IBERAMIA 2012*. Springer, 2012, S. 581–590
- [SF99] In: SCHUH, Günther ; FRIEDLI, Thomas: *Die Virtuelle Fabrik Konzepte, Erfahrungen, Grenzen*. Wiesbaden : Gabler Verlag, 1999. – ISBN 978–3–322–89482–3, S. 217–242
- [SFJ00] SMITH, David E. ; FRANK, Jeremy ; JÓNSSON, Ari K.: Bridging the gap between planning and scheduling. In: *The Knowledge Engineering Review* 15 (2000), Nr. 01, S. 47–83
- [SGL06] SHEN, Weiming ; GHENNIWA, Hamada ; LI, Yinsheng: Agent-based service-oriented computing and applications. In: *Pervasive Computing and Applications, 2006 1st International Symposium on IEEE*, 2006, S. 8–9
- [Sha99] SHANAHAN, Murray: The event calculus explained. In: *Artificial intelligence today*. Springer, 1999, S. 409–430
- [SHYN06] SHEN, Weiming ; HAO, Qi ; YOON, Hyun J. ; NORRIE, Douglas H.: Applications of agent-based systems in intelligent manufacturing: An updated review. In: *Advanced engineering INFORMATICS* 20 (2006), Nr. 4, S. 415–431
- [Sie01] SIERKSMA, Gerard: *Linear and integer programming: theory and practice*. CRC Press, 2001
- [SLVH12] SCHÜTZ, Daniel ; LEGAT, Christoph ; VOGEL-HEUSER, Birgit: On Modeling the State-Space of Manufacturing Systems with UML. In: BORANGIU, Theodor (Hrsg.) ; DOLGUI, Alexandre (Hrsg.) ; DUMITRACHE, Ioan (Hrsg.) ; FILIP, Florin G. (Hrsg.) ; IFAC (Veranst.): *Proc. of the 14th IFAC Symposium on Information Control Problems in Manufacturing (INCOM 2012)*. Bukarest, Romania : IFAC, May 2012, 469-474

- [SLVH14] SCHÜTZ, Daniel ; LEGAT, Christoph ; VOGEL-HEUSER, Birgit: MDE of Manufacturing Automation Software – Integrating SysML and Standard Development Tools. In: *12th IEEE International Conference on Industrial Informatics (INDIN)*. Porto Alegre, Brazil, Juli 2014, S. 267–273
- [SM83] SUZUKI, Ichiro ; MURATA, Tadao: A method for stepwise refinement and abstraction of Petri nets. In: *Journal of computer and system sciences* 27 (1983), Nr. 1, S. 51–76
- [SMC⁺14] SHERWOOD, Robert ; MISHKIN, Andrew ; CHIEN, Steve ; ESTLIN, Tara ; BACKES, Paul ; COOPER, Brian ; RABIDEAU, Gregg ; ENGELHARDT, Barbara: An integrated planning and scheduling prototype for automated Mars rover command generation. In: *Sixth European Conference on Planning*, 2014
- [SNT98] SMITH, Stephen J. ; NAU, Dana ; THROOP, Tom: Computer bridge: A big win for AI planning. In: *Ai magazine* 19 (1998), Nr. 2, S. 93
- [SOM02] SEBASTIA, Laura ; ONAINDIA, Eva ; MARZAL, Eliseo: STeLLa v2. 0: Planning with intermediate goals. In: *Advances in Artificial Intelligence—IBERAMIA 2002*. Springer, 2002, S. 805–814
- [SP04] SIRIN, Evren ; PARSIA, Bijan: Planning for semantic web services. In: *Semantic Web Services Workshop at 3rd International Semantic Web Conference*, 2004, S. 33–40
- [SPAS03] SYCARA, Katia ; PAOLUCCI, Massimo ; ANKOLEKAR, Anupriya ; SRINIVASAN, Naveen: Automated discovery, interaction and composition of semantic web services. In: *Web Semantics: Science, Services and Agents on the World Wide Web* 1 (2003), Nr. 1, S. 27–46
- [SPW⁺04] SIRIN, Evren ; PARSIA, Bijan ; WU, Dan ; HENDLER, James ; NAU, Dana: HTN planning for web service composition using SHOP2. In: *Web Semantics: Science, Services and Agents on the World Wide Web* 1 (2004), Nr. 4, S. 377–396
- [SRHD15] SCHUH, Günther ; REUTER, Christina ; HAUPTVOGEL, Annika ; DÖLLE, Christian: Hypotheses for a Theory of Production in the Context of Industrie 4.0. Version: 2015. http://dx.doi.org/10.1007/978-3-319-12304-2_2. In: BRECHER, Christian (Hrsg.): *Advances in Production Technology*. Springer, 2015 (Lecture Notes in Production Engineering). – DOI 10.1007/978-3-319-12304-2_2, S. 11–23
- [SSC⁺13] SCHREINER, Marcos A. ; SILVA, Fabiano ; CASTILHO, Marcos A. ; MONTAÑO, Razer A. ; KUNZLE, Luis A.: Planning Net with ordination of propositions: an improvement to planning solvers with Petri Nets. In: *2nd Brazilian Conference on Intelligent Systems*, 2013
- [SSF⁺11] SCHÜTZ, D ; SCHRAUFSTETTER, Markus ; FOLMER, Jens ; VOGEL-HEUSER, Birgit ; GMEINER, Thomas ; SHEA, Kristina: Highly reconfigurable production

- systems controlled by real-time agents. In: *Emerging Technologies & Factory Automation (ETFA), 2011 IEEE 16th Conference on IEEE*, 2011, S. 1–8
- [SSG⁺08] SOUZA, Luciana Moreira S. ; SPIESS, Patrik ; GUINARD, Dominique ; KÖHLER, Moritz ; KARNOUSKOS, Stamatis ; SAVIO, Domnic: SOCRADES: A Web Service based Shop Floor Integration Infrastructure. In: *The internet of things*, 2008
- [SSP06] SARDINA, Sebastian ; SILVA, Lavindra de ; PADGHAM, Lin: Hierarchical planning in BDI agent programming languages: A formal approach. In: *fifth international joint conference on Autonomous agents and multiagent systems ACM*, 2006, S. 1001–1008
- [SSV08] STRASSER, Thomas ; SÜNDER, Christoph ; VALENTINI, Antonio: Model-driven embedded systems design environment for the industrial automation sector. In: *Industrial Informatics, 2008. INDIN 2008. 6th IEEE International Conference on IEEE*, 2008, S. 1120–1125
- [ST09] SALEHIE, Mazeiar ; TAHVILDARI, Ladan: Self-adaptive software: Landscape and research challenges. In: *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* 4 (2009), Nr. 2, S. 14
- [Sta16a] STATISTISCHES BUNDESAMT: 43,6 Millionen Erwerbstätige im August 2016 / Statistisches Bundesamt. 2016 (347/16). – Pressemitteilung
- [Sta16b] STATISTISCHES BUNDESAMT: Bruttoinlandsprodukt 2015 für Deutschland / Statistisches Bundesamt. 2016. – Begleitmaterial zur Pressekonferenz am 14. Januar 2016 in Berlin
- [Sta16c] STATISTISCHES BUNDESAMT: Produzierendes Gewerbe: Beschäftigung und Umsatz der Betriebe des Verarbeitenden Gewerbes sowie des Bergbaus und der Gewinnung von Steinen und Erden / Statistisches Bundesamt. 2016 (2040411161084). – Fachserie 4 Reihe 4.1.1
- [STC98] SILVA, Manuel ; TERUE, Enrique ; COLOM, José M.: Linear algebraic and linear programming techniques for the analysis of place/transition net systems. In: *Lectures on Petri Nets I: Basic Models*. Springer, 1998, S. 309–373
- [Ste05] STERRITT, Roy: Autonomic computing. In: *Innovations in systems and software engineering* 1 (2005), Nr. 1, S. 79–88
- [SW09] SCHÜNEMANN, U. ; WITSCH, D.: UML zur technologieorientierten Spezifikation von Steuerungssoftware – Integration von Aktivitäts- und Zustandsdiagrammen in CoDeSys V3. In: *Automatisierungstechnische Praxis (atp)* 51 (2009), Nr. 6, S. 50–56
- [SWH06] SHEN, Weiming ; WANG, Lihui ; HAO, Qi: Agent-based distributed manufacturing process planning and scheduling: a state-of-the-art survey. In: *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 36 (2006), Nr. 4, S. 563–577

- [SWLVH13] SCHÜTZ, Daniel ; WANNAGAT, Andreas ; LEGAT, Christoph ; VOGEL-HEUSER, Birgit: Development of PLC-based Software for Increasing the Dependability of Production Automation Systems. In: *IEEE Transactions on Industrial Informatics* 9 (2013), November, Nr. 4, S. 2397 – 2406
- [SZ15] STRASSER, Thomas ; ZOITL, Alois: Distributed Real-time Automation and Control Reactive Control Layer for Industrial Agents. In: *Industrial Agents: Emerging Applications of Software Agents in Industry*. Elsevier, 2015
- [TF⁺11] THRAMBOULIDIS, Kleantlis ; FREY, Georg u. a.: Towards a model-driven IEC 61131-based development process in industrial automation. In: *Journal of Software Engineering and Applications* 4 (2011), Nr. 04, S. 217
- [TK00] TAN, Wei ; KHOSHNEVIS, Behrokh: Integration of process planning and scheduling—a review. In: *Journal of Intelligent Manufacturing* 11 (2000), Nr. 1, S. 51–63
- [TMS13] TELANG, Pankaj R. ; MENEGUZZI, Felipe ; SINGH, Munindar P.: Hierarchical planning about goals and commitments. In: *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems* International Foundation for Autonomous Agents and Multiagent Systems, 2013, S. 877–884
- [TOJ12] THEORIN, Alfred ; OLLINGER, Lisa ; JOHANSSON, Charlotta: Service-oriented Process Control with Grafchart and the Devices Profile for Web Services. In: *14th IFAC Symposium on Information Control Problems in Manufacturing* (2012)
- [TPB⁺13] TRENTESAUX, Damien ; PACH, Cyrille ; BEKRAR, Abdelghani ; SALLEZ, Yves ; BERGER, Thierry ; BONTE, Thérèse ; LEITÃO, Paulo ; BARBOSA, José: Benchmarking flexible job-shop scheduling and control systems. In: *Control Engineering Practice* 21 (2013), Nr. 9, 1204–1225. <http://dx.doi.org/10.1016/j.conengprac.2013.05.004>. – DOI 10.1016/j.conengprac.2013.05.004. – ISSN 09670661
- [Vah08] VAHRENKAMP, Richard: *Produktionsmanagement*. Oldenbourg Verlag, 2008
- [Val79] VALETTE, Robert: Analysis of Petri nets by stepwise refinements. In: *Journal of Computer and System Sciences* 18 (1979), Nr. 1, S. 35–46
- [VBC99] VAN BEEK, Peter ; CHEN, Xinguang: CPlan: A constraint programming approach to planning. In: *AAAI/IAAI, 1999*, S. 585–590
- [VBLN99] VOSSEN, Thomas ; BALL, Michael O. ; LOTEM, Amnon ; NAU, Dana: On the use of integer programming models in AI planning. (1999)
- [VDBBKV07] VAN DEN BRIEL, Menkes ; BENTON, J ; KAMBHAMPATI, Subbarao ; VOSSEN, Thomas: An LP-based heuristic for optimal planning. In: BESSIÈRE, Christian (Hrsg.): *Principles and Practice of Constraint Programming – CP 2007* Bd. 4741. Providence, RI, USA : Springer Berlin Heidelberg, September 2007 (Lecture Notes in Computer Science), S. 651–665

- [VDBVK05] VAN DEN BRIEL, Menkes ; VOSSEN, Thomas ; KAMBHAMPATI, Subbarao: Reviving Integer Programming Approaches for AI Planning: A Branch-and-Cut Framework. In: *ICAPS*, 2005, S. 310–319
- [VDI10] VDI/VDE: *VDI/VDE 2653: Multi agent systems in industrial automation – Part 1: Fundamentals*. 2010
- [VDM16] VDMA FORUM INDUSTRIE 4.0: Industrie-4.0-Forschung an deutschen Forschungsinstituten: Ein Überblick / VDMA e.V. 2016. – Forschungsbericht
- [VH09] VOGEL-HEUSER, Birgit: Objektorientierung im Engineering der Automatisierungstechnik: Fluch oder Segen? In: VOGEL-HEUSER, Birgit (Hrsg.): *Automation & Embedded Systems: Effizienzsteigerung im Engineering*. Kassel University Press, 2009
- [VH14] VOGEL-HEUSER, Birgit: Usability Experiments to Evaluate UML/SysML-Based Model Driven Software Engineering Notations for Logic Control in Manufacturing Automation. In: *Journal of Software Engineering and Applications* 7 (2014), Nr. 11, S. 943
- [VHDB13] VOGEL-HEUSER, Birgit ; DIEDRICH, Christian ; BROY, Manfred: Anforderungen an CPS aus Sicht der Automatisierungstechnik. In: *at-Automatisierungstechnik at-Automatisierungstechnik* 61 (2013), Nr. 10, S. 669–676. <http://dx.doi.org/10.1515/auto.2013.0061>. – DOI 10.1515/auto.2013.0061
- [VHDF⁺14] VOGEL-HEUSER, Birgit ; DIEDRICH, Christian ; FAY, Alexander ; JESCHKE, Sabine ; KOWALEWSKI, Stefan ; WOLLSCHLAEGER, Martin u. a.: Challenges for software engineering in automation. In: *Journal of Software Engineering and Applications* 2014 (2014)
- [VHFL13] VOGEL-HEUSER, Birgit ; FOLMER, Jens ; LEGAT, Christoph: Anforderungen an die Softwareevolution in der Automatisierung des Maschinen und Anlagenbaus. In: *at - Automatisierungstechnik* 62 (2013), Nr. 3, S. 163 – 174. – accepted
- [VHFST15] VOGEL-HEUSER, Birgit ; FAY, Alexander ; SCHAEFER, Ina ; TICHY, Matthias: Evolution of software in automated production systems Challenges and Research Directions. In: *Journal of Systems and Software* 110 (2015), Dezember, S. 54 – 84. <http://dx.doi.org/10.1016/j.jss.2015.08.026>. – DOI 10.1016/j.jss.2015.08.026
- [VHGL15] VOGEL-HEUSER, Birgit ; GÖHNER, Peter ; LÜDER, Arndt: Agent based control of production systems – and its architectural challenges. In: *Industrial Agents: Emerging Applications of Software Agents in Industry*. 2015
- [VHH16] VOGEL-HEUSER, Birgit ; HESS, Dieter: Industry 4.0 – Prerequisites and Visions. In: *IEEE Transactions on Automation Science and Engineering* 13 (2016), Nr. 2, S. 411–413. – Guest Editorial

- [VhKBW09] VOGEL-HEUSER, Birgit ; KEGEL, Gunther ; BENDER, Klaus ; WUCHERER, Klaus: Global information architecture for industrial automation. In: *atp edition international -Automatisierungstechnische Praxis International* 51 (2009), Nr. 01-02, S. 108–115
- [VHLFF14] VOGEL-HEUSER, B. ; LEGAT, C. ; FOLMER, J. ; FELDMANN, S.: Researching Evolution in Industrial Plant Automation: Scenarios and Documentation of the Pick and Place Unit / Technische Universität München. 2014 (TUM-AIS-TR-01-14-02). – Forschungsbericht
- [VHLFR14] VOGEL-HEUSER, Birgit ; LEGAT, Christoph ; FOLMER, Jens ; RÖSCH, Susanne: Challenges of Parallel Evolution in Production Automation Focusing on Requirements Specification and Fault Handling. In: *Automatisierungstechnik (at)* 11 (2014), November, Nr. 62, S. 755–826. <http://dx.doi.org/10.1515/auto-2014-1111>. – DOI 10.1515/auto-2014-1111
- [VHLL15] VOGEL-HEUSER, B. ; LEITAO, P. ; LEE, J.: Agents enabling Cyber-Physical Production Systems. In: *Automatisierungstechnik (at)* (2015). <http://dx.doi.org/10.1515/auto-2014-1153>. – DOI 10.1515/auto-2014-1153. – to appear
- [VHRF⁺16] VOGEL-HEUSER, Birgit ; RÖSCH, Susanne ; FISCHER, Juliane ; SIMON, Thomas ; ULEWICZ, Sebastian ; FOLMER, Jens: Fault handling in PLC-based industry 4.0 automated production systems as a basis for restart and self-configuration and its evaluation. In: *Journal of Software Engineering and Applications* 9 (2016), Nr. 1, S. 1
- [VHSF15] VOGEL-HEUSER, Birgit ; SCHÜTZ, Daniel ; FOLMER, Jens: Criteria-based alarm flood pattern recognition using historical data from automated production systems (aPS). In: *Mechatronics* 31 (2015), S. 89–100
- [VHSFL14a] VOGEL-HEUSER, Birgit ; SCHÜTZ, Daniel ; FOLMER, Jens ; LEGAT, Christoph: An assessment of the potentials and challenges in future approaches for automation software. In: *Industrial Agents: Emerging Applications of Software Agents in Industry*. Munich, Germany : Elsevier, Munich, Germany, Oktober 2014
- [VHSFL14b] VOGEL-HEUSER, Birgit ; SCHÜTZ, Daniel ; FRANK, Timo ; LEGAT, Christoph: Model-driven engineering of Manufacturing Automation Software Projects–A SysML-based approach. In: *Mechatronics* 24 (2014), Nr. 7, S. 883–897
- [VHW07] VOGEL-HEUSER, Birgit ; WANNAGAT, Andreas: *Modulares Engineering und Wiederverwendung mit CoDeSys V3: Für Automatisierungslösungen mit objektorientiertem Ansatz*. Deutscher Industrieverlag, 2007
- [VHWF15] VOGEL-HEUSER, Birgit ; WEBER, Jan ; FOLMER, Jens: Evaluating reconfiguration abilities of automated production systems in Industrie 4.0 with metrics. In: *2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)* IEEE, 2015, S. 1–6

- [VM10] VRBA, Pavel ; MARIK, Vladimír: Capabilities of dynamic reconfiguration of multiagent-based industrial control systems. In: *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 40 (2010), Nr. 2, S. 213–223
- [VMLK11] VALLÉE, Mathieu ; MERDAN, Munir ; LEPUSCHITZ, Wilfried ; KOPPENSTEINER, Gottfried: Decentralized Reconfiguration of a Flexible Transportation System. In: *IEEE Transactions on Industrial Informatics* 7 (2011), August, Nr. 3, S. 505 – 516
- [VTM⁺11] VRBA, Pavel ; TICHY, Pavel ; MARIK, Vladimír ; HALL, Kenwood H. ; STARON, Raymond J. ; MATURANA, Francisco P. ; KADERA, Petr: Rockwell Automation’s Holonic and Multiagent Control Systems Compendium. In: *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 41 (2011), Nr. 1, S. 14–30
- [Vya13] VYATKIN, Valeriy: Software engineering in industrial automation: State-of-the-art review. In: *Industrial Informatics, IEEE Transactions on* 9 (2013), Nr. 3, S. 1234–1249
- [WADX15] WHITMORE, Andrew ; AGARWAL, Anurag ; DA XU, Li: The Internet of Things – A survey of topics and trends. In: *Information Systems Frontiers* 17 (2015), Nr. 2, S. 261–274
- [Wan10] WANNAGAT, Andreas: *Entwicklung und Evaluation agentenorientierter Automatisierungssysteme zur Erhöhung der Flexibilität und Zuverlässigkeit von Produktionsanlagen*, Technische Universität München, Diss., 2010
- [WBB⁺10] WALLHOFF, Frank ; BLUME, Jürgen ; BANNAT, Alexander ; RÖSEL, Wolfgang ; LENZ, Claus ; KNOLL, Alois: A skill-based approach towards hybrid assembly. In: *Advanced Engineering Informatics* 24 (2010), Nr. 3, S. 329–339
- [WCL⁺05] WEERAWARANA, Sanjiva ; CURBERA, Francisco ; LEYMAN, Frank ; STOREY, Tony ; FERGUSON, Donald F.: *Web services platform architecture: SOAP, WSDL, WS-policy, WS-addressing, WS-BPEL, WS-reliable messaging and more*. Prentice Hall PTR, 2005
- [Wei91] WEISER, Mark: The computer for the 21st century. In: *Scientific american* 265 (1991), Nr. 3, S. 94–104
- [Wei99] WEISS, Gerhard: *Multiagent systems: a modern approach to distributed artificial intelligence*. MIT press, 1999
- [Wel94] WELD, Daniel S.: An introduction to least commitment planning. In: *AI magazine* 15 (1994), Nr. 4, S. 27
- [WEN⁺07] WIENDAHL, H-P ; ELMARAGHY, Hoda A. ; NYHUIS, Peter ; ZÄH, Michael F. ; WIENDAHL, H-H ; DUFFIE, Neil ; BRIEKE, Michael: Changeable manufacturing-classification, design and operation. In: *CIRP Annals-Manufacturing Technology* 56 (2007), Nr. 2, S. 783–809

- [Wer09] WERNER, Bernhard: Object-oriented extensions for IEC 61131-3. In: *Industrial Electronics Magazine, IEEE* 3 (2009), Nr. 4, S. 36–39
- [Wil99] WILKINS, David E.: Using the SIPE-2 Planning System: A Manual for SIPE-2, Version 6.1 / Artificial Intelligence Center, SRI International, Menlo Park, CA. 1999. – Forschungsbericht
- [Wit12] WITSCH, D.: *Modellgetriebene Entwicklung von Steuerungssoftware auf Basis der UML unter Berücksichtigung der domänenspezifischen Anforderungen des Maschinen- und Anlagenbaus*, Technische Universität München, Diss., 2012
- [WKC09] WALTER, Lutz ; KARTSOUNIS, George-Alexander ; CAROSIO, Stefano: Transforming clothing production into a demand-driven, knowledge-based, high-tech industry. In: *Transforming Clothing Production into a Demand-driven, Knowledge-based, High-tech Industry: The Leapfrog Paradigm, ISBN 978-1-84882-607-6. Springer London, 2009* 1 (2009)
- [Woo09] WOOLDRIDGE, Michael: *An introduction to multiagent systems*. John Wiley & Sons, 2009
- [WSVH08] WITSCH, D. ; SCHÜNEMANN, U. ; VOGEL-HEUSER, B.: Steigerung der Effizienz und Qualität von Steuerungsprogrammen durch Objektorientierung und UML. In: *Automatisierungstechnische Praxis (atp)* 50 (2008), Nr. 11, S. 42–47
- [WVH09] WITSCH, Daniel ; VOGEL-HEUSER, Birgit: Close integration between UML and IEC 61131-3: New possibilities through object-oriented extensions. In: *Emerging Technologies & Factory Automation, 2009. ETFA 2009. IEEE Conference on IEEE*, 2009, S. 1–6
- [WYL+13] WAN, Jiafu ; YAN, Hehua ; LIU, Qiang ; ZHOU, Keliang ; LU, Rongshuang ; LI, Di: Enabling cyber-physical systems with machine-to-machine technologies. In: *International Journal of Ad Hoc and Ubiquitous Computing* 13 (2013), Nr. 3-4, S. 187–196. <http://dx.doi.org/10.1504/IJAHUC.2013.055454>. – DOI 10.1504/IJAHUC.2013.055454
- [WZT00] WESTKÄMPER, E. ; ZAHN, E. ; TILEBEIN, M: Ansätze zur Wandlungsfähigkeit von Produktionsunternehmen – Ein Bezugsrahmen für die Unternehmensentwicklung im Turbulenten Umfeld. In: *wt – Werkstatttechnik* 90 (2000), Nr. 1/2, S. 22 – 26
- [XWN11] XU, Xun ; WANG, Lihui ; NEWMAN, Stephen T.: Computer-aided process planning—A critical review of recent developments and future trends. In: *International Journal of Computer Integrated Manufacturing* 24 (2011), Nr. 1, S. 1–31
- [YL04] YOUNES, Håkan LS ; LITTMAN, Michael L.: PPDDL1. 0: The language for the probabilistic part of IPC-4. In: *Proc. International Planning Competition, 2004*

- [ZBN⁺04] ZENG, Liangzhao ; BENATALLAH, Boualem ; NGU, Anne H. ; DUMAS, Marlon ; KALAGNANAM, Jayant ; CHANG, Henry: Qos-aware middleware for web services composition. In: *Software Engineering, IEEE Transactions on* 30 (2004), Nr. 5, S. 311–327
- [ZI06] ZIPARO, Vittorio A. ; IOCCHI, Luca: Petri net plans. In: *Proceedings of Fourth International Workshop on Modelling of Objects, Components, and Agents (MOCA)*, 2006, S. 267–290
- [ZIL⁺11] ZIPARO, Vittorio A. ; IOCCHI, Luca ; LIMA, Pedro U. ; NARDI, Daniele ; PALAMARA, Pier F.: Petri net plans: A framework for collaboration and coordination in multi-robot systems. In: *Autonomous Agents and Multi-Agent Systems* 23 (2011), Nr. 3, S. 344–383
- [ZKL⁺13] ZHANG, Jiafeng ; KHALGUI, Mohamed ; LI, Zhiwu ; MOSBAHI, Olfa ; AL-AHMARI, Abdulrahman M.: R-TNCES: A novel formalism for reconfigurable discrete event control systems. In: *Systems, Man, and Cybernetics: Systems, IEEE Transactions on* 43 (2013), Nr. 4, S. 757–772
- [ZKL⁺15] ZHANG, Jiafeng ; KHALGUI, Mohamed ; LI, Zhiwu ; FREY, Georg ; MOSBAHI, Olfa ; BEN SALAH, Hela: Reconfigurable coordination of distributed discrete event control systems. In: *Control Systems Technology, IEEE Transactions on* 23 (2015), Nr. 1, S. 323–330
- [ZO12] ZÜHLKE, Detlef ; OLLINGER, Lisa: Agile Automation Systems Based on Cyber-Physical Systems and Service-Oriented Architectures. In: LEE, Gary (Hrsg.): *Advances in Automation and Robotics* Bd. 122. Springer, 2012, S. 567–574
- [Zue10] ZUEHLKE, Detlef: SmartFactory—Towards a factory-of-things. In: *Annual Reviews in Control* 34 (2010), Nr. 1, S. 129–138. <http://dx.doi.org/10.1016/j.arcontrol.2010.02.008>. – DOI 10.1016/j.arcontrol.2010.02.008
- [ZX07] ZHANG, WJ ; XIE, SQ: Agent technology for collaborative process planning: a review. In: *The International Journal of Advanced Manufacturing Technology* 32 (2007), Nr. 3-4, S. 315–325

A. Anhang – Beschreibung verwendeter Systemfallstudien

A.1. Die Stempelanlage (PPU)

Die Stempelanlage – auch als Pick-and-Place Unit (PPU) bezeichnet - ist eine Lehr- und Forschungsanlage des Lehrstuhls für Automatisierung und Informationssysteme⁶⁴ der Technischen Universität München. Die PPU stellt dabei einen Demonstrator für die Fertigungsautomatisierung dar, um unterschiedliche Forschungsaspekte zu untersuchen. Trotz der eingeschränkten Größe und Komplexität der PPU, stellt sie eine geeignete Basis für erste, grundlegende Untersuchungen entwickelter Konzepte dar [VHFL13]. Für die Stempelanlage wurden bereits mehr als 15 unterschiedliche Ausführungen⁶⁵ (auch als Szenarien bezeichnet) detailliert beschrieben und dokumentiert [LFVH13, VHLFF14, VHLFR14]. Die unterschiedlichen Szenarien unterscheiden sich dabei hinsichtlich des mechanischen Aufbaus (d.h. genutzter Bauteile oder deren räumliche Anordnung) aber auch hinsichtlich der implementierten Funktionalität der Steuerungssoftware. Die Dokumentation, die teils frei verfügbar ist, umfasst für eine Vielzahl beschriebener Szenarien unter anderem Struktur- und Verhaltensbeschreibungen in Form von SysML Modellen, CAD Modelle, Simulationsmodelle und Programmcode für die Steuerung durch IEC 61131-3 basierte SPSen. Die im Rahmen dieser Dissertation genutzte Ausführung bzw. Szenario wird im Folgenden erläutert. Wird im Rahmen dieser Arbeit von Stempelanlage oder PPU gesprochen, wird dabei auf dieses Szenario Bezug genommen.

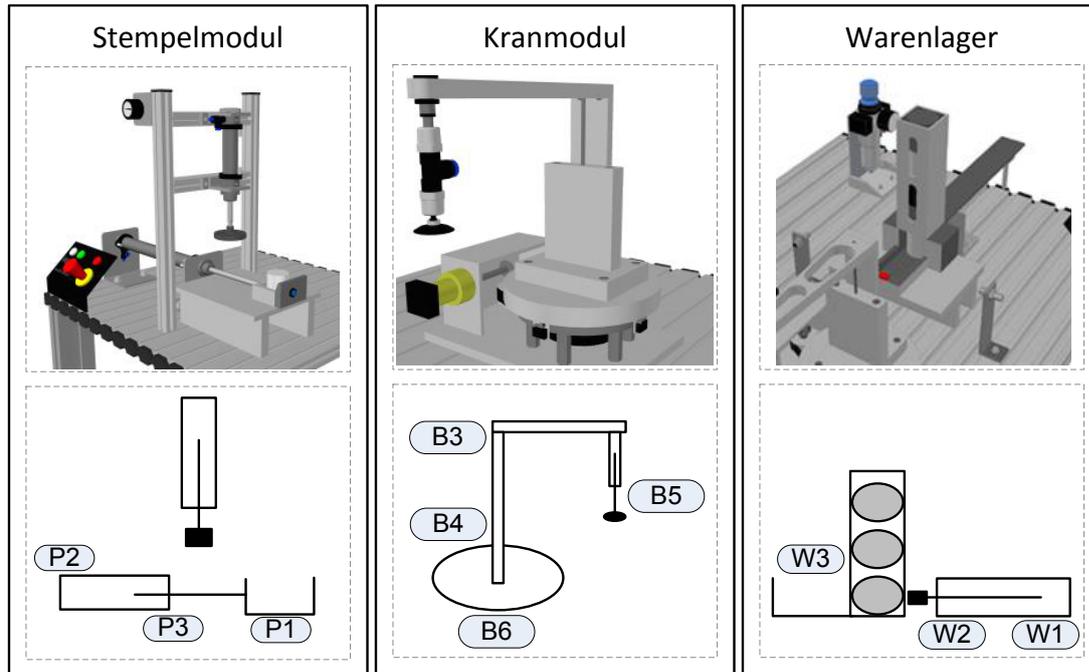
Die Stempelanlage verarbeitet zylindrische Werkstücke und besteht aus drei mit Sensoren und Aktoren ausgestatteten (d.h. mechatronischen) Einheiten (siehe Abbildung A.1) – einem Stempelmodul, einem Kranmodul und einem Warenlager – sowie einer rein mechanischen Rampe.

Innerhalb des *Warenlagers* werden zylindrische Werkstücke gelagert und durch Ausfahren eines pneumatischen Zylinders vereinzelt. Zur Identifikation der Zylinderposition werden zwei binäre Sensoren $W1$ und $W2$ genutzt. Der Sensor $W3$ – durch einen Mikroschalter realisiert – ist an der Ausstoßposition der Werkstücke positioniert und ermöglicht die Erkennung der Anwesenheit von Werkstücken. Durch die Information dieses Sensors kann auf eine erfolgreiche Vereinzlung geschlossen werden, falls der Sensor den Wert *true* liefert. Liefert der Sensor $W3$ trotz ausgefahrenem Zylinder den Wert *false*, lag kein Werkstück mehr für die Vereinzlung vor und das Warenlager ist leer.

Das *Kranmodul* ist auf einer drehbaren Kranbasis montiert, dessen Position mittels des Positionsgebers $B6$ bestimmt werden kann. Konstruktionsbedingt kann sich das Kranmodul ausschließlich zwischen den Werten 0° und 359° und somit nicht vollständig um seine eigene Achse drehen. Vertikale Bewegungen, d.h. anheben und absenken, des Krans werden durch einen pneumatischen Zylinder erreicht. Die binären Sensoren $B3$ und $B4$ ermöglichen die Bestimmung der vertikalen Kranposition. Weißt $B3$ den Wert *true* und $B4$ den Wert *false* auf, befindet sich der Kran in angehobener Position. Im Falle einer hierzu invertierten Wertebelegung der Sensoren $B3$ und $B4$ befindet sich der Kran in abgesenkter Position. Für die Aufnahme bzw. Abgabe von

⁶⁴<http://www.ais.mw.tum.de>

⁶⁵Es wird hier gezielt nicht von Varianten oder Versionen gesprochen, da dies eine zeitliche Reihenfolge hinsichtlich der Existenz der unterschiedlichen Ausführungen voraussetzen würde.



Variable	Beschreibung	Datentyp
W2	Zylinder ausgefahren	Boolean
W1	Zylinder eingefahren	Boolean
W3	Werkstück vereinzelt	Boolean
B3	Hebezyylinder an oberer Position	Boolean
B4	Hebezyylinder an unterer Position	Boolean
B5	Vakuumbreifer beladen	Boolean
B6	Positionsgeber der Kranbasis	Integer
P1	Werkstückerkennung	Boolean
P2	Einspannzylinder eingefahren	Boolean
P3	Einspannzylinder ausgefahren	Boolean

Abbildung A.1.: Überblick der installierten Sensoren der Stempelanlage.

Werkstücken ist am äußeren Ende des Auslegers ein Vakuumbreifer installiert. Der binäre Sensor *B5* ermöglicht die Identifikation des Beladezustandes des Krans.

Das *Stempelmodul* ermöglicht das Einspannen von Werkstücken mit Hilfe eines pneumatischen Zylinders unter einer Presseinheit. Hierzu sind, analog zu dem im Kran verwendeten Zylinder, zwei Sensoren *P2* und *P3* für die Bestimmung der Zylinderposition integriert. Ferner ist ein Mikroschalter *P1* in der Aufnahme position des Stempelmoduls angebracht, um die Anwesenheit von Werkstücken in der Ladevorrichtung zu identifizieren.

Abbildung A.2 zeigt eine Übersicht der Stempelanlage (d.h. des genutzten Szenarios der Stempelanlage) und die damit verbundene Anordnung der Module. Ziel der Stempelanlage ist die Bearbeitung zylindrischer Werkstücken im Stempelmodul und anschließender Materialabgabe an der Rampe. Hierzu müssen unbearbeiteter Werkstücken aus dem Warenlager vereinzelt werden. Auf Grund der Beschaffenheit der Werkstücke sowie der Konstruktion des Vakuumbreifers

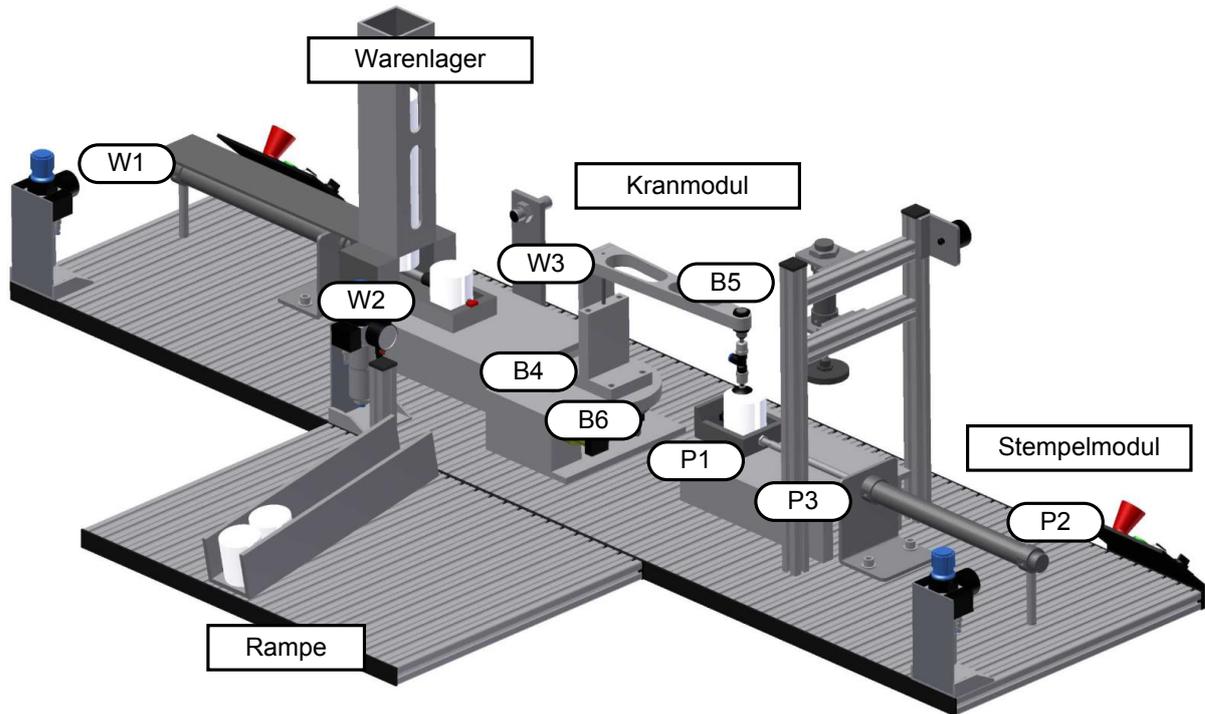


Abbildung A.2.: Gesamtübersicht des mechanischen Aufbaus der Stempelanlage.

des Kranmoduls ist eine rotatorische Bewegung des Krans in abgesenkter Position nicht möglich. Zur Aufnahme der Werkstücke am Warenlager ist folglich sowohl das Anheben sowie eine Drehung des Kranmoduls zur Position des Warenlagers (Position 90° des Positionsgebers $B6$) notwendig. Das Kranmodul nimmt durch das Warenlager vereinzelte Werkstücke durch Ansaugen des Werkstücks mittels des Vakuumschneppers auf. Die Fixierung für die Aufnahme des Werkstücks durch das Kranmodul erfolgt hierbei bis zur Übernahme des Werkstücks mittels des pneumatischen Zylinders des Warenlagers. Im Anschluss an eine erfolgreiche Werkstückaufnahme durch das Kranmodul wird das Werkstück zum Stempelmodul transportiert, das sich auf Kranposition 270° (Sensor $B6$) befindet. Um eine Aufnahme des Werkstücks durch das Stempelmodul zu ermöglichen, muss der Zylinder des Stempelmoduls ausgefahren sein, so dass sich die Aufnahmevorrichtung des Stempelmoduls unterhalb des Kranmoduls – genauer dessen Auslegers und etwaiger, daran befindlicher Werkstücke – befindet, falls dieser an die entsprechende Position gedreht wurde. Nun kann das Werkstück vom Kranmodul an das Stempelmodul zur Bearbeitung übergeben werden. Bevor eine Handhabung des Werkstücks durch das Stempelmodul erfolgen kann, muss der Kran angehoben werden, um die Aufnahmevorrichtung nicht zu beschädigen. Nun kann mittels der Zylinder des Stempelmoduls samt Werkstück eingefahren, das Werkstück durch die Presseinheit verarbeitet, d.h. gestempelt, und der Zylinder wieder ausgefahren werden. Abschließend werden die bearbeiteten Werkstücke vom Kran zur Rampe (Kranposition 180° des Sensors $B6$) transportiert und abgeladen, um eine Entnahme der Produkte für eine weitere, manuelle Weiterverarbeitung am Ende der Rampe zu ermöglichen.

A.2. Das Hybride Prozessmodell

Das sogenannte Hybride Prozessmodell (HPM) ist, wie auch die zuvor beschriebene Stempelanlage, eine Demonstrationsanlage des Lehrstuhls für Automatisierung und Informationssysteme der Technischen Universität München. Durch das Hybride Prozessmodell werden Glasflaschen mit Granulaten unterschiedlicher Farbe befüllt. Im Rahmen dieser Arbeit werden zwei als Ausbaustufen bezeichnete Versionen des HPM betrachtet (siehe Abbildung A.3).

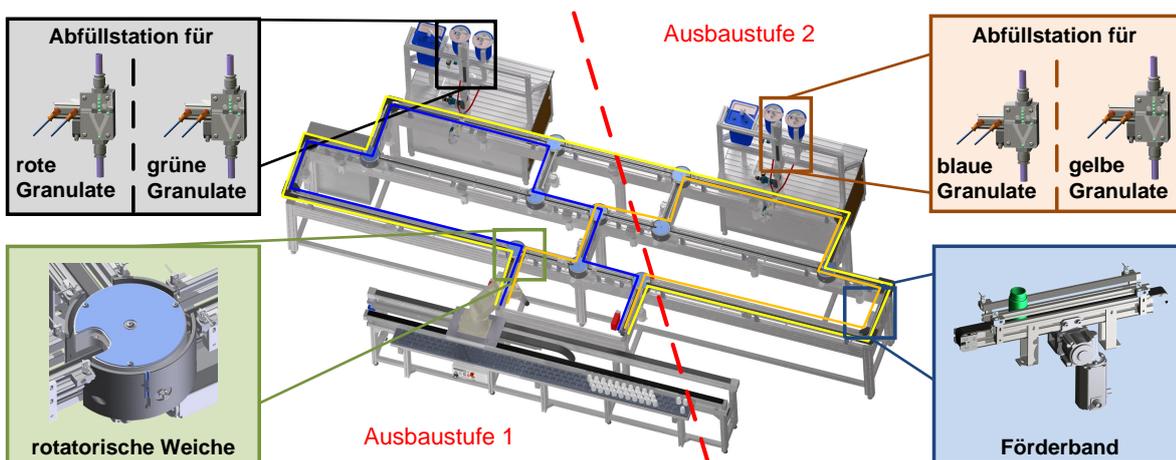


Abbildung A.3.: Überblick des mechanischen Aufbaus des Hybriden Prozessmodells [LSVH13].

In der auch real existierenden *Ausbaustufe 2* stehen in der Laboranlage zwei räumlich getrennten Stationen, in welchen jeweils zwei unterschiedlich farbige Granulate abgefüllt werden können (vgl. Abbildung A.3). Der logistische Teil besteht aus einem Raster von mehr als 20 Transportbändern und 10 rotatorischen Weichen, die jeweils getrennt von einander durch einen bidirektionalen Motor angetrieben werden. Zur Lagerung von Flaschen verfügt das HPM über einen 5-achsigen Roboter, welcher Flaschen zwischen einer Ablagefläche und zwei Förderbändern des logistischen Anlagenteils kommissionieren kann.

In der im Verhältnis zu *Ausbaustufe 2* verkleinerten *Ausbaustufe 1* der HPM steht lediglich eine Teilmenge der logistischen Komponenten, d.h. Transportbänder und Weichen, sowie nur eine Abfüllstation zur Verfügung (siehe Abbildung A.3). Hierdurch ist die Menge der möglichen technischen Prozesse und Produkte ebenfalls eingeschränkt, da nur grüne und rote Granulate in der Abfüllstation zur Verfügung stehen. Die Funktionalität des Kommissionierroboters ist dabei in beiden Ausbaustufen identisch.

Drei unterschiedlichen Routen der Flaschen, welche das Produkt der Anlage bilden, sind in Abbildung A.3 farblich gekennzeichnet. Die blaue Route kann in beiden Ausbaustufen realisiert werden und ermöglicht die Abfüllung einer beliebigen Anzahl roter und grüner Granulate in Abhängigkeit des gewünschten Produktes. Im Gegensatz dazu kann die gelbe Route ausschließlich mit *Ausbaustufe 2* realisiert werden und reflektiert alle Produkte bei denen mindestens ein rotes oder grünes Granulat sowie mindestens ein blaues oder gelbes Granulat abgefüllt werden müssen. Die orange gekennzeichnete Route kann ebenfalls nur in *Ausbaustufe 2* realisiert werden und spiegelt die Abfüllung von Flaschen wieder, die ausschließlich blaue und/oder gelbe Granulate enthalten.

