

Wolfgang Wolf

*Entwicklung eines agentenbasierten  
Steuerungssystems zur Materialflussorganisation im  
wandelbaren Produktionsumfeld*





Wolfgang Wolf

*Entwicklung eines agentenbasierten  
Steuerungssystems zur Materialflussorganisation im  
wandelbaren Produktionsumfeld*

Herausgegeben von  
Professor Dr.-Ing. Klaus Feldmann,  
Lehrstuhl für  
Fertigungsautomatisierung und Produktionssystematik

**FAPS**



Meisenbach Verlag Bamberg

Als Dissertation genehmigt von der Technischen Fakultät  
der Friedrich-Alexander-Universität Erlangen-Nürnberg

Tag der Einreichung:	26. September 2008
Tag der Promotion:	19. Februar 2009
Dekan:	Prof. Dr.-Ing. habil. J. Huber
Berichterstatter:	Prof. Dr.-Ing. K. Feldmann
	Prof. Dr.-Ing. habil. W. Schröder-Preikschat

### **Bibliografische Information Der Deutschen Bibliothek**

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

ISSN 1431-6226  
ISBN 978-3-87525-293-4

Dieses Werk ist urheberrechtlich geschützt.  
Alle Rechte, auch die der Übersetzung, des Nachdrucks  
und der Vervielfältigung des Buches oder Teilen daraus,  
vorbehalten.  
Kein Teil des Werkes darf ohne schriftliche Genehmigung des  
Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder ein  
anderes Verfahren), auch nicht für Zwecke der Unterrichts-  
gestaltung - mit Ausnahme der in den §§ 53, 54 URG ausdrücklich  
genannten Sonderfälle -, reproduziert oder unter Verwendung  
elektronischer Systeme verarbeitet, vervielfältigt oder  
verbreitet werden.

© Meisenbach Verlag Bamberg 2009  
Herstellung: Gruner Druck GmbH, Erlangen-Eltersdorf  
Printed in Germany

# Vorwort und Danksagung

Die vorliegende Dissertation entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik der Friedrich-Alexander-Universität Erlangen-Nürnberg.

Herrn Prof. Dr.-Ing. Klaus Feldmann, Leiter dieses Lehrstuhls im Department Maschinenbau, gilt mein Dank für die engagierte Förderung bei der Durchführung meiner Arbeit, die vielfältigen wissenschaftlichen Freiräume und das mir entgegengebrachte Vertrauen.

Herrn Prof. Dr.-Ing. Wolfgang Schröder-Preikschat, Leiter des Lehrstuhls für verteilte Systeme und Betriebssysteme der Friedrich-Alexander-Universität Erlangen-Nürnberg, danke ich für die Übernahme des Korreferats und die konstruktiven Gespräche. Ferner gilt mein Dank Herrn Prof. Dr.-Ing. Harald Meerkamm für die Übernahme des Vorsitzes bei meiner Promotionsprüfung sowie Herrn Prof. Dr.-Ing. Günter Roppenecker, für die Teilnahme als weiterer Prüfer und die Unterstützung seines Instituts bei unseren Forschungsvorhaben.

All meinen Kolleginnen und Kollegen sei an dieser Stelle herzlich für die stets sehr gute und kollegiale Zusammenarbeit im Team, die zahlreichen fachlichen Diskussionen und die stets angenehme Atmosphäre gedankt. Hervorheben möchte ich in diesem Zusammenhang besonders meinen langjährigen Bürokollegen Herrn Dr.-Ing. Matthias Weber, dem ich für die intensive und freundschaftliche Zusammenarbeit, begleitet von fachlich anregenden Diskussionen, danke. Darüber hinaus danke ich Herrn Dr.-Ing. Stefan Junker und Herrn Dr.-Ing. Stefan Lang für die fachlichen Gespräche und Beantwortung von Fragen zu Themen des Maschinenbaus.

Ferner möchte ich mich bei allen Studenten bedanken, die mich während meiner Zeit am Lehrstuhl tatkräftig im Rahmen verschiedener Projekte unterstützt und so einen wichtigen Beitrag zur Realisierung von Forschungsvorhaben und zur Erstellung der vorliegenden Arbeit geleistet haben.

Besonders bedanken möchte ich mich bei meinen Eltern, die mich seit meiner Kindheit stets gefördert und beim Erreichen meiner eigenen Ziele unterstützt haben. Mein größter Dank gilt meiner Frau Anja für die ursprüngliche Motivation zur Aufnahme der Stelle als wissenschaftlicher Mitarbeiter, das zeitintensive Korrekturlesen der Arbeit und die Geduld und Liebe, die sie mir während der Anfertigung der Arbeit entgegengebracht hat.

Kühnhofen, im März 2009

Wolfgang Wolf

*Das schönste Glück des denkenden Menschen ist,  
das Erforschliche erforscht zu haben und  
das Unerforschliche zu verehren.*

Johann Wolfgang von Goethe

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Grundlagen zu Materialfluss- und Agentensystemen</b>	<b>4</b>
2.1	Steigerung der Flexibilität in der Produktion . . . . .	4
2.2	Materialflusssteuerung in der Produktion . . . . .	6
2.3	Stand der Technik Fahrerloser Transportsysteme . . . . .	9
2.3.1	Zentraler Steuerungsansatz für Fahrerlose Transportsysteme . . . . .	9
2.3.2	Defizite kommerziell verfügbarer Fahrerloser Transportsysteme . . . . .	12
2.3.3	Entwicklungs- und Inbetriebnahmeaufwände . . . . .	14
2.4	Der Multi-Agentenansatz . . . . .	17
2.4.1	Definition des Agenten-Begriffs . . . . .	17
2.4.2	Interaktionen im Multiagentensystem . . . . .	19
2.4.3	Modellierungsformen von Agenten . . . . .	21
2.4.4	Agententypen . . . . .	22
2.4.5	Zielsetzungen des Agentenansatzes . . . . .	23
2.5	Einsatzbereiche und Konzepte für Agententechnologien in der Produktion . . . . .	24
2.5.1	Agenteneinsatz in der Planung . . . . .	24
2.5.2	Agentenbasierte Produktions- und Materialflusssteuerungssysteme . . . . .	25
2.5.3	FIPA-Agentenplattform . . . . .	27
2.5.4	Wandelbare Produktion . . . . .	29
<b>3</b>	<b>Agentenbasiertes Systemmodell für wandelbare Produktionsanlagen</b>	<b>31</b>
3.1	Identifikation der Komponenten einer Produktionsanlage aus Sicht des Materialflusses	31
3.2	Modellierung von Produktionsanlagen mittels standardisierter Komponenten . . . . .	34
3.3	Differenzierung verschiedener Lagertypen . . . . .	37
3.4	Klassifikation von Transfersystemen . . . . .	40
3.5	Modellierung von Reibschlussfördersystemen . . . . .	41
3.6	Modellierung von fahrzeugbasierten Transportsystemen . . . . .	42
3.7	Auftragseinlastung zur Organisation der Produktion innerhalb eines Produktionssystems	43
3.8	Agentenbasiertes Modell zur Organisation der Produktionsabläufe . . . . .	43
3.8.1	Definition der Agentenklassen . . . . .	44
3.8.2	Spezifikation der Teilnehmer im Agentenverbund wandelbarer Produktionsanlagen . . . . .	45
3.8.3	Funktion und Aufgaben der Werkstückagenten . . . . .	47
3.8.4	Übergabebahnhof als Schnittstelle zwischen Transportsystem und Bearbeitungsstationen . . . . .	50
3.8.5	Vermittlerrolle des Übergabebahnhofs . . . . .	52
3.8.6	Lageragent zur Verwaltung der Ein- und Auslagerungsvorgänge . . . . .	55

<b>4</b>	<b>Verteilte Steuerungsarchitektur als Grundlage für den effektiven Einsatz in der Produktion</b>	<b>58</b>
4.1	Dezentraler Steuerungsansatz für wandelbare Produktionsumgebungen . . . . .	58
4.2	Hierarchische Modellierung und Anordnung der Anlagenkomponenten . . . . .	60
4.3	Selbsterkennung und Selbstkonfiguration der Entitäten . . . . .	62
4.3.1	Verfahren zur Anwesenheitserkennung . . . . .	63
4.3.2	Weiterleitung der Nachbarschaftsdaten an alle Entitäten . . . . .	64
4.3.3	Algorithmus zur Strukturermittlung . . . . .	65
4.3.4	Automatisierte, verteilte Konfiguration der IP-Adressen und IP-Routen-Tabellen . . . . .	65
4.4	Wandelbare Softwarearchitektur für verteilte Steuerungssysteme . . . . .	67
4.4.1	Knoten zum strukturellen Aufbau . . . . .	67
4.4.2	Facette als Träger der Funktionalität . . . . .	68
4.5	Modularer Aufbau der Steuerungsarchitektur . . . . .	70
4.5.1	Net-O-Matic zur Nachbarschaftserkennung und Selbstkonfiguration . . . . .	71
4.5.2	Plant-O-Matic - Referenzimplementierung der Softwarearchitektur . . . . .	72
4.5.3	Includ-O-Matic - Integration peripherer Geräte . . . . .	76
4.6	Graphische Entwicklungsumgebung (GIDE) zur Ablaufspezifikation . . . . .	77
<b>5</b>	<b>Konzepte und Algorithmen zur Realisierung autonom navigierender Fahrerloser Transportsysteme</b>	<b>82</b>
5.1	Anforderungen in wandelbaren Produktionsumgebungen . . . . .	83
5.2	Neuer Steuerungsansatz für Fahrerlose Transportsysteme . . . . .	84
5.3	Kartographierung mittels integriertem Vermessungswerkzeug . . . . .	86
5.4	Bekannte Methoden zur Umgebungsmodellierung . . . . .	90
5.5	Übersicht der bekanntesten Bahnplanungsalgorithmen . . . . .	92
5.5.1	Bahnplanung mittels topologischer Karten . . . . .	93
5.5.2	Zellenmethode . . . . .	94
5.5.3	Potentialfeldmethode . . . . .	95
5.6	Flexible Bahnplanung anhand von Umgebungsinformationen . . . . .	96
5.6.1	Füllen der Rasterkarte mit Zonendaten . . . . .	96
5.6.2	Berechnung der Andock- und Abdockpunkte . . . . .	98
5.6.3	Ermittlung des Pfades . . . . .	99
5.6.4	Optimierung des Pfades . . . . .	102
<b>6</b>	<b>Agentenbasierter Materialfluss zur Koordinierung der Bearbeitungs- und Transportvorgänge</b>	<b>105</b>
6.1	Koordinierung der Werkstückbearbeitung durch den Werkstückagenten . . . . .	105
6.1.1	Aufgaben des Werkstückagenten . . . . .	106
6.1.2	Verhandlungsaufgaben . . . . .	108
6.1.3	Interaktion zwischen Werkstückagenten und Bearbeitungsstation . . . . .	110
6.2	Verarbeitung von Produkt- und Anlageninformationen aus Primärquellen . . . . .	112
6.2.1	Spezifikation der Produktherstellungsschritte mittels Produktionsvorranggraph . . . . .	113
6.2.2	Spezifikation der Technologien . . . . .	114
6.2.3	Spezifikation einer Produktvariante . . . . .	115
6.2.4	Ableitung von Materialflusssentscheidungen . . . . .	117

6.3	Maschinen- und Betriebsdatenerfassung . . . . .	120
6.4	Übergabebahnhof als Schnittstelle zwischen Stationen und Transportsystemen . . . .	124
6.4.1	Konzeption eines allgemeinen Bahnhofmodells . . . . .	124
6.4.2	Anbindung des Eingangs-Bahnhofs an die Bearbeitungsstation . . . . .	125
6.4.3	Anbindung des Ausgangs-Bahnhofs an die Bearbeitungsstation . . . . .	127
6.4.4	Interaktionen zwischen Bahnhof und Transportsystem . . . . .	128
6.5	Verhandlungsmechanismus zwischen Werkstück und den Ressourcen . . . . .	130
<b>7</b>	<b>Prototypische Implementierung und Validierung</b>	<b>133</b>
7.1	Aufbau des Labors für verteilte Steuerungssysteme . . . . .	133
7.2	Evaluierung der Steuerungsarchitektur im IT-Labor . . . . .	135
7.3	Mechanischer Aufbau des Demonstrationsfahrzeugs . . . . .	138
7.3.1	Konstruktion eines flexiblen Fahrzeugs mit hoher Bewegungsfreiheit . . . .	138
7.3.2	Sensorik zur Positionsbestimmung und Erfassung der Umgebung . . . . .	139
7.3.3	Verteilte Steuerungshardware zur aufgabenbezogenen Steuerung und Regelung	140
7.4	Referenzimplementierung im Labor zur Evaluierung der agentenbasierten Material-	
	flussteuerung . . . . .	145
7.4.1	Vermessung der Halle . . . . .	146
7.4.2	Autonome Fahrzeugsteuerung . . . . .	148
7.4.3	Anbindung der Bahnhöfe an die bestehende Zellensteuerung . . . . .	151
<b>8</b>	<b>Zusammenfassung</b>	<b>153</b>
	<b>Summary</b>	<b>157</b>
	<b>Literaturverzeichnis</b>	<b>160</b>





# Kapitel 1

## Einleitung

Die zunehmende Globalisierung und Öffnung der Wirtschaftsräume für Produkte aus Niedriglohnländern und die daraus resultierende leichtere Vergleichbarkeit von Angeboten führt zu gravierenden Veränderungen der Wettbewerbssituation, verstärkt durch das Streben der Unternehmen nach Wachstum [1]. Dabei werden Produzenten aus Hochlohnländern vor die große Herausforderung gestellt, Segmente des Marktes zu erschließen, in denen sie der Konkurrenz aus Niedriglohnländern begegnen. Sie sehen sich mit einer sinkenden Markentreue, neuen Anforderungen der Kunden bezüglich der Erfüllung individueller Produktwünsche und schwankenden Bedarfe konfrontiert. Entscheidende Wettbewerbsfaktoren sind eine schnelle Reaktionsfähigkeit und die Fertigkeit, Produkte kundenspezifisch auslegen zu können. Die daraus resultierende hohe Produkt- und Variantenvielfalt stellt die Hersteller vor die Problematik, wirtschaftlich zu produzieren und dabei die Produkte kostengünstig anbieten zu müssen. Zudem sinkt durch den schnellen Innovationsfortschritt zunehmend der Produktlebenszyklus, der die Anlagensysteme beeinflusst, da neue Produktvarianten Anpassungen an den Produktionsmitteln bedingen. Verstärkt wird dieser Effekt durch zunehmend sinkende Losgrößen, die vom Kunden bedarfsgerecht in Auftrag gegeben werden. Um gleichzeitig Lagerkosten zu minimieren, werden Produkte zunehmend Just-In-Time oder Just-In-Sequence hergestellt, wodurch die Produktion der Herausforderung ständig wechselnder Produktmixe begegnen muss. In diesem turbulenten Umfeld kann eine hohe Wirtschaftlichkeit nur durch die Maximierung des Auslastungsgrades der Anlagensysteme bei gleichzeitig hoher Flexibilität erreicht werden. Dabei stehen die Hersteller vor zunehmenden logistischen und produktionstechnischen Problemstellungen, die sie mit geeigneten Organisationsstrukturen und mit flexiblen Technologien bewältigen müssen. Wandlungsfähige Fabrikstrukturen bieten hierzu geeignete Technologien und Verfahren, die auf verschiedenen Ebenen der Fabrikorganisation eine hohe Anpassungsfähigkeit erlauben bzw. die technologische Voraussetzung bieten, aufwandsarm die Anlagensysteme umbauen bzw. umrüsten zu können. [2, 3, 4]

Der Materialfluss stellt in der Produktion das zentrale Bindeglied zwischen allen Bearbeitungsstationen dar und ist wesentlich für den reibungsfreien Betrieb. Neben einer hohen Ausfallsicherheit muss das Materialflusssystem den stetig neuen Anforderungen und den wechselnden Produktionsbedingungen angepasst werden. Jedoch weisen heutige Systeme nur eine geringe Dynamik auf und basieren auf inflexiblen Verfahren und vielfach auf starren Regeln. Um Abweichungen vom standardisierten Ablauf, beispielsweise Störungen oder Anpassung an produkt- oder organisationsbedingte Veränderungen, kompensieren zu können, ist der Eingriff in die Materialflussteuerung oder -organisation oft nur über manuell zu integrierende Sonderregeln möglich. Da zur Entwicklungszeit nicht alle Ausnahmefälle und gegenseitigen Einflüsse erfasst werden können, liefern die manuell erstellten Regeln nur bedingt eine Lösung zur Kompensation von Störungen oder von Veränderungen in der Liefer- und Produktionskette. Zudem sind die bestehenden Systeme oftmals nicht in der Lage, den Betrieb bei Teilausfällen möglichst aufrechtzuerhalten, da zumeist die durchgängige technologische und organisatorische Entkopplung modular und gleichzeitig autonom agierender Einheiten nicht gegeben ist.

Motiviert aus den steigenden Forderungen an die Flexibilität in der Erzeugung von Produkten mit hoher Variantenvielfalt bei gleichzeitig geringen Losgrößen und sinkenden Lieferzeiten, sehen sich die Unternehmen zunehmend zu einer Ablösung bestehender starrer Systeme durch flexible und zugleich robuste Lösungen gezwungen. Ändern sich die Anforderungen an die Produktion derart, dass diesen durch elastische Flexibilität allein nicht entsprochen werden kann, muss ein struktureller Umbau erfolgen [5, 6]. Geeignete Plug&Produce-Mechanismen bieten dem Anlagenbetreiber wirtschaftliche und effektive Möglichkeiten, u.a. Materialflusssysteme mit der notwendigen Flexibilität zu versehen, um Anlagen ad hoc abhängig vom Bedarf umbauen und den aktuellen Anforderungen an die Produktion anpassen zu können. Der Betreiber ist somit in der Lage, Linien und Anlagen gemäß der derzeit aktuellen Auftragslage aufzubauen oder zu erweitern, um die Nachfrage decken zu können. Analog können Produktionskapazitäten reduziert werden, um die entnommenen wiederverwendbaren Komponenten an anderer Stelle in der Produktion erneut einzusetzen. Dabei sind nicht nur mechanische Lösungen ausschlaggebend, sondern insbesondere die flexible Anpassbarkeit auf Seiten der Informationsverarbeitung ist ein wesentliches Kriterium für einen schnellen Umbau. In diesen Vorstellungen spiegelt sich der Ansatz des Plug&Produce wider, unter dem das Verknüpfen von Produktionselementen zu einer betriebsbereiten Linie ohne bzw. nur mit minimalem manuellen Aufwand verstanden wird [7]. Voraussetzung hierfür sind geeignete Mechanismen und Dienste in der Steuerungstechnik mit erweiterten, intelligenten Algorithmen. Diese sind erforderlich, um die Inbetriebnahme und spätere Umbauphasen erheblich zu verkürzen und eine möglichst hohe Wirtschaftlichkeit erzielen zu können.

### **Ziele und Aufbau der Arbeit**

Das Ziel der vorliegenden Arbeit besteht darin, für die Werkstattsteuerung im strukturell wandelbaren und organisatorisch dynamischen Produktionsumfeld ein selbst-adaptierendes Steuerungssystem zu entwickeln. Dieses bietet einerseits die Eigenschaft der Selbstkonfiguration zur aufwandsarmen Anlageninbetriebnahme und zum Anlagenumbau und andererseits der Selbstorganisation zur flexiblen Materialflussorganisation während der Laufzeit mit autonom agierenden Einheiten. Das Steuerungssystem muss dabei insbesondere den Anforderungen einer vereinfachten Integration verteilter Einheiten genügen und hierbei den Entwickler mit Werkzeugen zur Entwicklungszeit und Mechanismen zur Laufzeit unterstützen, um bisher manuell erforderliche Aufwände zu automatisieren.

Eine Voraussetzung ist zunächst die systematische Erfassung der Anlagenkomponenten und ihrer strukturellen Beziehungen. Hierzu wird in Kapitel 3 ein agentenbasiertes Systemmodell vorgestellt, mit dem funktional abgeschlossene und modular abgrenzbare Anlagenkomponenten entworfen werden können. Die systematische Identifizierung und funktionale Abgrenzung von Anlagenkomponenten auf Materialflussebene bieten den Vorteil, Komponentenklassen mit einheitlichem Verhalten zu identifizieren, um daraus im nächsten Schritt wiederverwendbare Programmmodule abzuleiten. Dabei konzentriert sich die Modellerfassung gezielt auf deren Verhalten zur Organisation der Werkstück- und Bauteilflüsse im Materialflussverbund und zur Organisation der Tätigkeiten. Basierend auf den Ansätzen der Agententechnologie werden diesen zudem autonomes und autarkes Verhalten zugewiesen, um kollaborativ im Verbund eine gemeinsame Aufgabenstellung erfüllen zu können. Als Ergebnis steht ein agentenbasiertes Modell mit normierten und autonom agierenden Teilnehmern zur Verfügung, die sich insbesondere durch ihre Rolle im Umfeld mit anderen Teilnehmern auszeichnen.

---

Das Modell lässt sich auf beliebige Anwendungen in der Produktion anwenden und ist zudem an verschiedene Anforderungen flexibel anpassbar.

Grundlage für eine hohe Wandelbarkeit ist ein flexibles Steuerungssystem, das dem Automatisierer manuelle Konfigurationsschritte mit automatisierten Mechanismen sowohl während der Entwicklungszeit mit geeigneten Werkzeugen als auch während der Laufzeit mit Automatismen zur Selbstkonfiguration abnimmt. Die in Kapitel 4 vorgestellte wandelbare Steuerungsarchitektur unterstützt den Entwickler maßgeblich in der Umsetzung verteilter Komponenten, die eine technologische Grundlage zur transparenten Integration von Anwendungen über die Rechnergrenzen hinaus erfordern. Die Architektur zielt darauf ab, die Steuerungssoftware durch Kopplung von wiederverwendbaren Softwarebausteinen, die in einem Baukastensystem hinterlegt werden können, aufzubauen. Zudem unterstützt sie die Wandelbarkeit von Anlagensystemen durch Mechanismen zur automatisierten Ableitung der Anlagenstruktur als Voraussetzung für die Ableitung von Materialflussregeln.

Die Wandelbarkeit von Anlagensystemen wirkt sich insbesondere auf den Materialfluss aus, der sich flexibel an die dynamischen Anforderungen anpassen muss. Aus dieser Motivation heraus werden im Folgenden neuartige Steuerungslösungen sowohl für spurgeführte Gurt- oder Rollenfördersysteme als auch für freinavigierende Fahrerlose Transportsysteme vorgestellt. Freinavigierende Flurförderfahrzeuge bieten prinzipiell eine hohe Flexibilität an, da sie zunächst keine spurbezogenen Installationen im Boden benötigen und sich somit frei, ohne mechanische Umbaumaßnahmen, im Raum bewegen können. Bei den Fahrerlosen Transportsystemen stellt nach derzeitigem Stand der Technik jedoch das zugehörige Steuerungssystem nicht die erforderliche Flexibilität zur Verfügung. Das in Kapitel 5 vorgestellte Steuerungskonzept für freinavigierende Fahrzeuge betrachtet dabei einerseits eine autonome Bahnplanung zur Laufzeit und die Ausstattung der Fahrzeuge mit einer autonomen Entscheidungsgehalt gemäß dem Agentenansatz.

Die genauen Abläufe zur Materialflussorganisation im Verbund mit verteilt und autonom agierenden Agenten, die auf dem agentenbasierten Systemmodell basieren (Kapitel 3), werden in Kapitel 6 beschrieben. Die Realisierung fußt auf der Umsetzung von Steuerungsagenten, die physisch an den Geräten vor Ort zur Ausführung gelangen. Dabei wird auf die Interaktionen zwischen allen Teilnehmern im Materialflussverbund und die erforderlichen Dienste und Mechanismen im Detail eingegangen, die für die Gewährleistung einer flexiblen Produktion zuständig sind. Zudem werden Verfahren aufgezeigt, wie Daten aus Primärquellen, d.h. Anlagen- und Produktdaten, direkt automatisiert verarbeitet werden können. Als Ergebnis werden daraus Arbeitspläne als Grundlage für den Materialfluss sowie die Prozesskoordination und -steuerung abgeleitet.

Im Rahmen dieser Arbeit wurde für den agentenbasierten Steuerungsansatz zur aufwandsarmen Integration und dynamischen Selbstorganisation eine Referenzimplementierung realisiert, die in Kapitel 7 vorgestellt wird. Die Strategien wurden in zwei Laboren des Lehrstuhls für Fertigungsautomatisierung und Produktionssystematik der Universität Erlangen-Nürnberg evaluiert. Zudem wurde der neuartige Steuerungsansatz für autonom navigierende Fahrerlose Transportsysteme an einem eigens entwickelten Fahrerlosen Transportfahrzeug implementiert und im Verbund mit den verteilt agierenden Systemmodulen validiert.

## Kapitel 2

### Grundlagen zu Materialfluss- und Agentensystemen

Die Produktion stellt im Wesentlichen ein Flusssystem mit angebundenen Produktionseinheiten dar, die neben stofflichen Objekten auch mit Energie, Signalen und Informationen versorgt werden müssen [8]. Die logistische Verkettung der Produktionseinheiten wird insbesondere durch den Materialfluss bestimmt, der für den termingerechten Transport von Gütern verantwortlich ist [5, 9, 10, 11]. In Abhängigkeit von der Produkt- und Variantenvielfalt sowie der zu erreichenden Zielgrößen (z.B. Ausbringung und Liefertreue), stehen unterschiedliche Konzepte zur anwendungsspezifischen Auslegung der Materialflusssysteme und der Produktionsanlage zur Verfügung [5].

Um jedoch zukünftigen Anforderungen des Marktes, wie die Herstellung Just-in-Time in geringen Losgrößen und hoher Produktvariantenvielfalt gerecht zu werden, sind zunehmend flexible und modulare Produktionssysteme [6] erforderlich. Die Produktionssysteme haben hierzu in den letzten 20 Jahren eine umfassende Entwicklung von starr gekoppelten Systemen zu autonomen, lose gekoppelten Produktionszentren erfahren (Kapitel 2.1). Zur logistischen Vernetzung dieser autonomen Produktionszentren sowohl in der Fertigung als auch in der Montage, ist ein leistungsfähiges Materialflusssystem erforderlich, das termingerecht Werkstücke und Bauteile anliefern. Heutige Steuerungssysteme zur Organisation des Materialflusses sind geprägt von starren Transportflüssen und bieten nicht die Voraussetzungen für den Einsatz in wandlungsfähigen Anlagen (Kapitel 2.2). Die gleichen Defizite weisen insbesondere die Fahrerlosen Transportfahrzeuge auf, die trotz hoher Bewegungsfreiheit kaum Flexibilität und Freiheit in der Neuorganisation der Flüsse und Transportwege besitzen (Kapitel 2.3). Hierzu sind umfassende, neuartige Steuerungskonzepte erforderlich, die neben einer mechanischen und elektrischen Modularität aufwandsarme Umbaumaßnahmen ermöglichen. Diese und weitere Aspekte werden mit dem Agenten-Ansatz, autonom agierender und kooperierender Teilnehmer im Verbund, unterstützt. Hierzu werden in Kapitel 2.4 zunächst das allgemeine Konzept des Multiagenten-Ansatzes erläutert und, bezogen auf die Produktion in Kapitel 2.5, verschiedene Anwendungsgebiete und Entwicklungen vorgestellt, die sich zumeist noch im Forschungsstadium befinden, bzw. nur bedingt die erforderliche Flexibilität bieten.

#### 2.1 Steigerung der Flexibilität in der Produktion

Der zunehmende Kostendruck in der Produktion führte in den vergangenen Jahren selbst in der Mittel- und Kleinserienfertigung zu einer zunehmenden Automatisierung der Fertigungsprozesse [12, 13]. Entgegen früherer Anlagensysteme, die nur eine starre Verkettung der Fertigungsschritte in einer Linie vorsahen, zielen heutige Ansätze auf die lose Kopplung autonomer Fertigungszellen, die über einen gerichteten oder ungerichteten Werkstückfluss miteinander gekoppelt sind. Hieraus ist in den 90er Jahren der Ansatz der flexiblen Fertigungssysteme (FFS) entstanden, der insbesondere für den Einsatz in der Produktion mit unterschiedlichen Losgrößen und hoher Variantenvielfalt geeignet ist [14]. Die Idee fußt auf dem Einsatz flexibler, voll- und teilautomatisierter Bearbeitungssysteme,

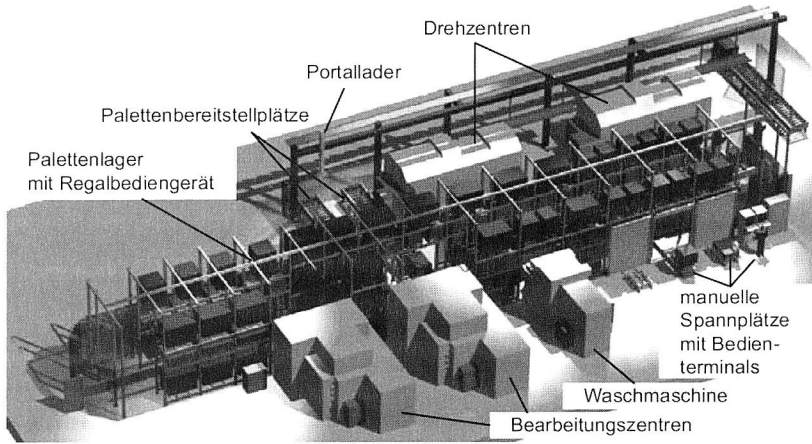


Bild 2.1: Flexibles Fertigungssystem für die Zerspangung, bestehend aus voll- oder teilautomatisierten Bearbeitungssystemen, die sich wiederum aus einer oder mehreren Bearbeitungsmaschinen, einem automatisierten Material- bzw. Teilelager und einer oder mehreren Be- und Entladestationen zusammensetzen. [17]

die aus einer oder mehreren Bearbeitungsmaschinen, einem automatisierten Material- bzw. Teilelager und einer oder mehreren Be- und Entladestationen bestehen. Der Aufbau entspricht i.d.R. einer Nebenschluss- oder Netzwerkstruktur, in deren Zentrum sich ein Lagersystem mit einem häufig schienengebundenen Transportgerät bzw. Regalbediengerät befindet. An den Längsseiten des Lagersystems sind die Maschinen angebunden und können vom Transportgerät automatisiert beschickt werden [15]. Diese Fertigungssysteme sind typischerweise in der Fräs-, Dreh- und der Blechbearbeitung sowie dem Sägen von Stangenmaterial vorzufinden [16]. Bild 2.1 zeigt ein Beispiel eines solch flexiblen Fertigungssystems für die Zerspangung.

Zudem sollte eine zunehmende Automatisierung der Fertigungsprozesse und des Materialflusses zu einer Senkung der Werkzeugkosten beitragen. Bedingt durch die hohe Komplexität der Systemlösungen wird jedoch genau das Gegenteil verursacht. Der hohe Automatisierungsgrad führt zu hohem Investitions- und Planungsaufwand, der selbst Bedienungs- und Servicepersonal überfordert [18, 19]. Somit ist die Produktivität flexibel automatisierter Fertigungssysteme dennoch geringer als bei starr automatisierten.

Laut einer Studie des Fraunhofer Instituts für Systemtechnik und Innovationsforschung (ISI) beklagen deutsche Unternehmen die unzureichende Leistungsfähigkeit der automatisierten Fertigungs- und Montagesysteme bei der Herstellung kleiner Losgrößen [20]. Diese Ergebnisse werden durch eine weitere Studie in [21] unterstrichen, die die mangelnde Leistungsfähigkeit in der eingesetzten Steuerungsoftware begründet sieht. Sie hat einen wesentlichen Einfluss auf die Flexibilität in der Fertigung und konnte den wechselnden Anforderungen aus wirtschaftlichen Gesichtspunkten nicht genügen. Dies

liegt u.a. an der hohen Unübersichtlichkeit im Bereich der Logistik gegenüber starr automatisierten Anlagen. Zur Bewältigung der erforderlichen Aufgaben sind komplexe Steuerungssysteme zur Materialflusssteuerung, Werkstückverfolgung, Ressourcen- und Werkzeugverwaltung erforderlich [22].

## 2.2 Materialflusssteuerung in der Produktion

Die in Kapitel 2.1 angesprochenen Defizite bei bestehenden Steuerungssystemen zur Materialflusssteuerung und Werkstückverfolgung resultieren in heutigen Systemen in verschiedenen Bereichen der Produktionssteuerung aus einem hohen manuellen Aufwand zur Aufrechterhaltung des Betriebes bzw. zur Anpassung der Systeme an neue Produktionsbedingungen [23, 24, 25]. Dabei hängt die Komplexität des Materialflusses in erster Linie vom Produktionsplan ab. Im einfachsten Fall wird nur ein Produkt gefertigt. Hierbei müssen Prozesse in der richtigen Reihenfolge aneinandergehängt, d.h. die Arbeitsstationen zusammen mit dem Transportsystem in eine gewöhnliche Linie überführt werden. Die Komplexität erhöht sich, wenn ein oder mehrere Prozesse mehrfach an unterschiedlichen Stellen innerhalb der Prozessfolge für das gleiche Produkt wiederholt werden müssen. In der Anlagenplanung muss daher entschieden werden, ob die mehrfach wiederholten Prozessschritte von mehreren oder nur einer Bearbeitungsstation durchgeführt werden sollen. Im letzten Fall müssen mehrere zyklische Wege im Materialfluss vorgesehen werden, die eine umfangreiche Materialflusssteuerung zur Folge haben.

Darüber hinaus nimmt die Komplexität zu, wenn mehrere Produkte oder unterschiedliche Varianten mit unterschiedlichen Prozessschritten bzw. -folgen gefertigt werden müssen. Das Ergebnis sind Produktionsanlagen mit verzweigten oder zyklischen Strukturen, die ebenso ineinander geschachtelt sein können. Für Anlagen mit Verzweigungen ist es erforderlich für jedes Produkt die Pfade durch die Produktionsanlage festzulegen. Dies erfolgt durch die Spezifikation der produktbezogenen Routing-Regeln an den Materialflusskreuzungen, die im Materialflusssystem hinterlegt sind. Die Methodik zur Ableitung der Routentabellen ist in Bild 2.2 veranschaulicht. Die Routentabellen müssen manuell vom Entwickler aus den Arbeitsplänen der verschiedenen Produktvarianten und der Anlagenstruktur abgeleitet werden, bevor dieser die Daten im Materialflusssystem eingeben kann. Hierzu ist für jede Kreuzung innerhalb des Materialflusssystems eine eigene Routentabelle zu erzeugen, die bezogen auf die bestehenden Produkte bzw. Varianten die Richtung zur Weiterleitung des zu bearbeitenden Werkstücks bestimmt. Zur Laufzeit liest die Kreuzung den Bearbeitungszustand des Werkstücks aus oder entnimmt die erforderlichen Daten anhand einer eindeutigen ID aus einer zentralen Datenbank und entscheidet, basierend auf dem Produkt bzw. der Variante und dem Bearbeitungszustand, in welche Richtung das Werkstück zum Erreichen der nächsten Bearbeitungsstation weitergeleitet werden muss. Mit der manuellen Erstellung der Routentabellen ist zudem eine hohe Gefahr menschlicher Fehler verbunden, die nur durch aufwändige Tests oder in Simulationen erkennbar wird. Folglich ist ein hoher Engineering- und Zeitaufwand mit den zahlreichen Validierungs- und daran anschließenden Korrekturphasen verbunden.

Die zur Entwicklungszeit abgeleiteten Routentabellen gehen von einem Idealzustand der Anlage aus, d.h. alle Systeme sind einsatzbereit, es liegen keine Staus vor und alle Voraussetzungen zur Produktherstellung, wie Betriebsmittel oder Bauteile, sind an den Bearbeitungsstationen termingerecht vorrätig. Jedoch sind während der Laufzeit Materialflusssysteme einer großen Zahl dynamischer Einflüsse

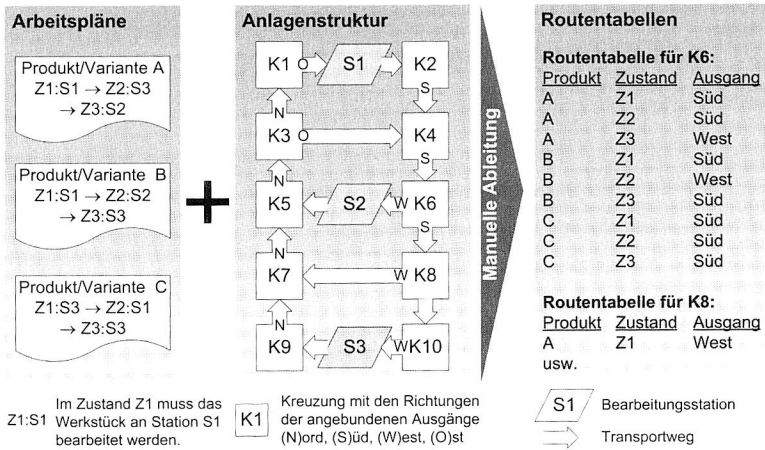


Bild 2.2: Die manuelle Ableitung der Routentabellen, basierend auf den produktbezogenen Arbeitsplänen und der Anlagenstruktur für jede einzelne Kreuzung, erfordert einen hohen Konfigurationsaufwand.

ausgesetzt. Dies ist durch notwendige Veränderungen im Produktionsablauf bedingt. Die Störungen innerhalb des Materialflusssystems müssen erfasst werden, um die Materialflussorganisation durch Nachbesserungen zielgerichtet zu optimieren. In der Konsequenz ergeben sich hieraus zahlreiche Regeln, die nur schwer nachvollziehbar und wartbar sind. Wenn nachträgliche Anlagenmodifikationen erforderlich werden, müssen die Regeln im Nachhinein ebenso manuell adaptiert werden. Ein weiteres Problem liegt in der fehlenden Möglichkeit zur Validierung vorab, wodurch Tests an der realen Anlage die Inbetriebnahmezeit maßgeblich verlängern. Zwar können Simulationssysteme eingesetzt werden, jedoch ist dies ein aufwendiges Unterfangen, das aus Gründen der Wirtschaftlichkeit die Erfassung und Erstellung der Simulationsmodelle mit ihrem vollständigen dynamischen Verhalten nicht zulässt. Zuletzt stehen während der Planung nicht alle Informationen zur Erstellung des Simulationsmodells zur Verfügung, da sich die realen Anlagen noch im Aufbau befinden und somit die für das Simulationsmodell erforderlichen Bearbeitungsdaten nicht erfasst werden können. Letztendlich werden aus wirtschaftlichen Gründen die Regeln manuell erfasst und diese an der realen Anlage validiert. Dies führt, wie in Kapitel 2.1 beschrieben, zu hohem Aufwand bei Anlagenmodifikationen und somit zu Wirtschaftlichkeitsproblemen.

Nach heutigem Stand der Technik werden Materialflusssysteme meist von zentralen Steuerungskomponenten koordiniert und gesteuert [26]. Hierbei ist das zentrale Steuerungssystem mit den Sensoren des Transportsystems gekoppelt, um alle Teilstände innerhalb des Systems zu erfassen. Basierend auf den oben beschriebenen Routingregeln trifft die zentrale Steuereinheit die Richtungsentscheidung zur Weiterleitung des Werkstücks. Zur physischen Durchführung der Werkstückweiterleitung muss die zentrale Steuereinheit zudem die entfernten Aktuatoren an den Kreuzungen des Materialflusssystems ansteuern. Zur Veranschaulichung des Mechanismus wird im Folgenden ein spurgeführtes, gurt- oder rollengetriebenes Transportsystem angenommen, das aus Transportstrecken und Kreuzungen be-

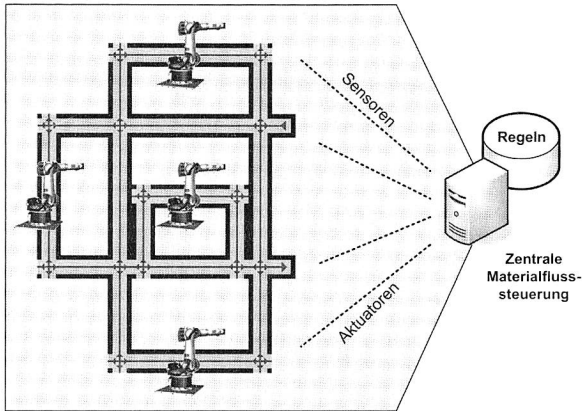


Bild 2.3: Zentraler Steuerungsansatz zur Koordinierung des Materialflusses. Der zentrale Rechner besitzt Zugriff auf alle Sensoren und Aktuatoren, die i.d.R. über ein Feldbussystem angebunden sind. Mit den global hinterlegten Regeln zum Routen der Werkstücke entscheidet der zentrale Steuerrechner für jede Kreuzung im Materialfluss, in welche Richtung die Werkstücke zur Erreichung ihres Zieles weitergeleitet werden müssen.

steht (Bild 2.3). Die zentrale Steuerung erfasst zunächst mittels Sensorik die Ankunft des Werkstücks an jeder Kreuzung. Wenn die Kreuzung frei ist, ermittelt die zentrale Steuerung für die Kreuzung, welches wartende Werkstück an den Eingängen als nächstes geroutet wird. Daraufhin sendet sie an die entsprechende Einlasskontrolle ein Signal zur Aufforderung das nächste, in der Warteschlange befindliche Werkstück einzulassen. Ist das Werkstück auf der Kreuzung positionsgenau platziert, erfolgt in den meisten Implementierungen erst auf der Kreuzung die Ermittlung der Werkstückidentifikation, die meist über einen RFID-Tag oder einen Barcode erfolgt. Die ID des Werkstücks ermöglicht der zentralen Steuerung, abhängig von deren aktuellen Bearbeitungszustand, die Entscheidung über die nächste Richtungsweiterleitung zu treffen, um auf dem kürzesten, verfügbaren Weg die nächste Bearbeitungsstation zu erreichen. Hierzu koordiniert die zentrale Steuerung alle vorhandenen Aktuatoren, um das Werkstück freizugeben und in die gewünschte Richtung zu befördern.

Das zentrale Steuerungssystem birgt das Risiko eines Single-Point-of-Failures, d.h. dem vollständigen Stillstand des Materialflusssystems, wenn die zentrale Steuerung ausfällt. Zudem besteht der Nachteil, dass sich das System nur schlecht skalieren lässt, d.h. die Größe des Materialflusssystems ist durch die maximalen Erweiterungsmöglichkeiten der zentralen Steuerung begrenzt. Dieser Aspekt ist insbesondere für den Einsatz in wandelbaren Produktionsumgebungen von entscheidender Bedeutung, da beispielsweise bei Einführung eines Produktes nicht bekannt ist, welche Kapazitäten in der Hochphase des Produktabsatzes erforderlich sind [27]. Folglich ist auch nicht bekannt, in welche Modelle für Steuerungs- und Kommunikationssysteme investiert werden soll. Umgekehrt sind die Betreiber aus Gründen der Kostenreduktion gezwungen, lediglich Anlagenmodule anzuschaffen, die funktional und kapazitiv auf die minimalen Anforderungen abgestimmt sind.



Zusammenfassend lässt sich aus den beschriebenen hohen Aufwänden zur Inbetriebnahme bestehender Steuerungssysteme für den Materialfluss ableiten, dass sich derzeitige Systeme für den wirtschaftlichen Einsatz in wandelbaren Produktionssystemen nicht eignen. Für den effektiven Einsatz müssen diese geeignete Mechanismen und Dienste bereitstellen, die Informationen automatisiert ableiten bzw. diese automatisiert aus den Primärquellen, wie Produktherstellungsdaten oder Anlagenstrukturdaten, verarbeiten können, um somit manuelle Aufwände zu reduzieren. Dies betrifft insbesondere die Ableitung der Materialfluss- bzw. Routingregeln, die selbst bei nur wenigen Bearbeitungsstationen und verschiedenen Produktvarianten mit unterschiedlichen Bearbeitungsfolgen eine hohe, nur schwer zu durchschauende Komplexität aufweisen.

## 2.3 Stand der Technik Fahrerloser Transportsysteme

Die in Kapitel 2.2 beschriebenen Defizite der Materialflusssysteme bestehen ebenso bei Fahrerlosen Transportsystemen. Trotz der hohen Bewegungsfreiheit als Flurförderfahrzeuge, sind diese ebenso starren Fahrwegen und Navigationsregeln unterworfen und können kaum ihre eigentliche Flexibilität effektiv nutzen. Im Folgenden wird der Stand der Technik kommerziell verfügbarer Fahrerloser Transportsysteme vorgestellt. Diese werden ebenso von einem zentralen Leitrechner gesteuert und koordiniert, der alle Transportaufträge verwaltet und einlastet. Zudem ist er für die Navigation jedes einzelnen Fahrzeugs verantwortlich, um diesem den Weg zum Erreichen des nächsten Zielbahnhofes vorzugeben. Diese und weitere Aspekte werden in Kapitel 2.3.1 beschrieben. Der zentrale Steuerungsansatz und die starren Navigationsstrategien erfordern hohe Entwicklungs- und Inbetriebnahmeaufwände, die schnelle Anpassungen an neue Produktionsbedingungen, wie neue Transportabläufe oder neue Transportwege, nicht erlauben. Diese Aufwände bzw. die einzelnen Schritte zum Aufbau eines Fahrerlosen Transportsystems in der Produktion vor Ort werden in Kapitel 2.3.3 eingehend erläutert.

### 2.3.1 Zentraler Steuerungsansatz für Fahrerlose Transportsysteme

#### Zentrale Steuerung

Nach heutigem Stand der Technik werden Fahrerlose Transportsysteme von zentraler Stelle koordiniert und überwacht. Analog zu allgemeinen Steuerungsstrukturen für Anlagen werden alle Vorgänge im Fahrerlosen Transportsystem und jedes einzelne Fahrerlose Transportfahrzeug von einem zentralen übergeordneten Leitrechner gesteuert [28, 29]. Dieser übernimmt dabei alle administrativen und steuerungstechnischen Aufgaben, zur Erfüllung der globalen Ziele, wie die Maximierung des Durchsatzes. Zu seinen Aufgaben zählen neben einem optimierten Scheduling aller Transportaufträge auf die verfügbaren Fahrzeuge [30] selbst die Koordinierung der lokalen Vorgänge jedes einzelnen Fahrzeugs. Somit ist der größte Teil des steuerungstechnischen Funktionsumfangs eines Fahrerlosen Transportsystems im zentralen Leitrechner konzentriert. Die Fahrzeuge agieren lediglich als Befehlsempfänger und sind weder über die Gesamtabläufe und den Zustand des Gesamtsystems noch über ihre eigenen Transportaufgaben im vollständigen Umfang informiert. Dies bedeutet, dass der Leitrechner einen Transportauftrag in verschiedene Einzel- und Teilschritte zerlegt und diese dem Fahrzeug nur sukzessive nach Abarbeitung jedes einzelnen Schrittes vorgibt (vgl. Bild 2.4).

Zunächst besteht die Aufgabe des zentralen Leitrechners darin, mit den Bearbeitungsstationen und Bahnhöfen über das Netzwerk zu kommunizieren, um Transportaufträge entgegenzunehmen und die-

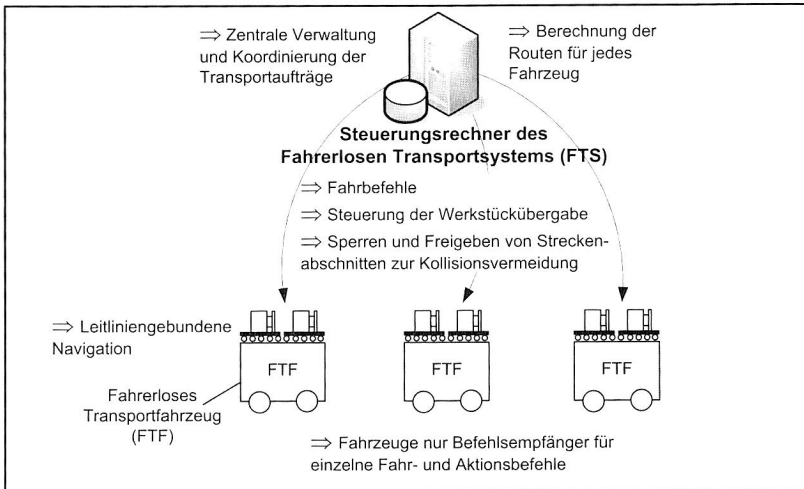


Bild 2.4: Klassische Steuerungsarchitektur mit zentralem Steuerungsrechner - eingesetzt bei kommerziell verfügbaren Fahrerlosen Transportsystemen.

se einzulasten. Hierzu werden die Transportaufträge zentral in einer Auftragsdatenbank gespeichert und nach bestimmten Prioritätskriterien, wie dem Auftragsingang oder der spätesten Lieferfrist, sortiert. Zudem müssen die Aufträge in optimierter Weise den verfügbaren Fahrzeugen zugeordnet werden. Die hierbei angewandten Verfahren bauen oftmals auf Heuristiken und bekannten Optimierungsstrategien auf, die allgemeine Problemstellungen zur durchgängigen Belegung der Ressourcen behandeln.

Generell verfügen die Fahrzeuge nur über eine geringe Entscheidungsgewalt und sind nur zu Handlungen berechtigt, die vom zentralen Leitreechner vorgegeben bzw. autorisiert werden. Ihre Aufgabe beschränkt sich auf das Verfolgen der Bahnen, Erkennen von Hindernissen, die zum Stop des Fahrzeugs führen, und der Aufnahme bzw. Abgabe von Werkstücken. Somit sind sie zur Laufzeit ständig vom zentralen Leitstand abhängig und müssen von diesem alle durchzuführenden Aktionen, die Tourenpläne und die Freigaben zur Befahrung von Streckenabschnitten einholen. Letztere dienen zur Vermeidung von Kollisionen zwischen den Fahrzeugen. Hierzu werden Streckenabschnitte vom zentralen Leitreechner für jedes Fahrzeug gesperrt. Bevor ein Fahrzeug in einen neuen Abschnitt einfahren darf, muss dieses die Berechtigung einholen und ggf. warten, bis diese erteilt wird. Am Bahnhof angekommen muss der Leitreechner dem Fahrzeug mitteilen, welche Aktion durchzuführen ist, d.h. ob ein Werkstück aufgenommen oder abgegeben werden soll. Zudem muss der Leitreechner im Falle mehrerer Aufnahmeplätze auf dem Fahrzeug ebenso die Position vorgeben, die Richtlinien zum optimierten Transport genügen muss. Die Koordination der Werkstückübergabe erfolgt nach heutigem Stand der Technik noch ausschließlich über Sensorik. Das bedeutet, dass der Leitstand nicht mit dem Bahnhof über das Netzwerk kommuniziert, sondern der Bahnhof über eine Anwesenheitssensorik verfügt, die es ihm erlaubt zu erkennen, wenn das Fahrzeug angekommen ist. Erst dann erfolgt der Austausch der

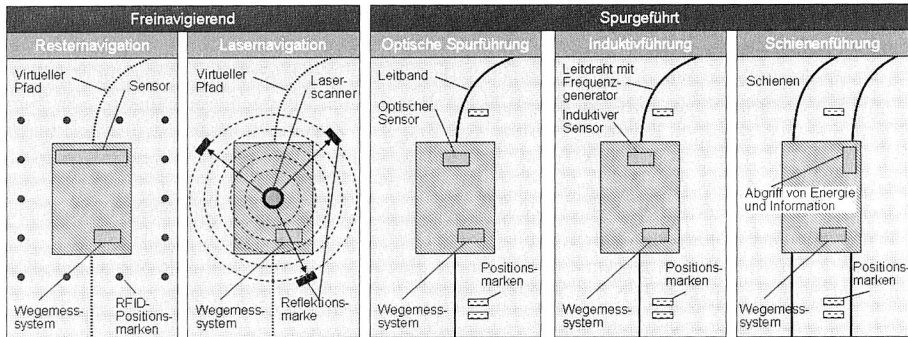


Bild 2.5: Freinavigierende und spurgebundene Spurführungskonzepte - in beiden Fällen folgen die Fahrzeuge einer fest vorgegebenen Spur: freinavigierende Systeme basieren auf einem virtuellen Pfad, spurgeführte auf einer physischen Leitspur.

Werkstücke zwischen dem Bahnhof und dem Fahrzeug. Diese Vorgehensweise ist in der Minimierung des Entwicklungs- und Integrationsaufwands zur softwaretechnischen Einbindung der Bahnhöfe bzw. Bearbeitungsstationen und Leitrechner begründet.

## Navigation

Bei den industriell verfügbaren Lösungen ist die Navigation Fahrerloser Transportfahrzeuge nur durch vorgegebene Bahnen möglich. Dabei bewegen sich die Fahrzeuge entlang vorgegebener Leitlinien zwischen den Übergabebahnhöfen. Die Linien bzw. Pfade können mittels physischer Markierungen im Boden festgelegt sein, z.B. in Form von optischen Markierungen, elektrischen Leitern oder mechanischen Schienen (vgl. Bild 2.5). Diese Form der Navigation bedarf eines hohen Installationsaufwands bei der Erstinbetriebnahme und im Falle von Routenänderungen. Aus diesem Grund sind diese Formen der Navigation nicht für den Einsatz in wandelbaren Produktionssystemen geeignet.

Im Gegensatz dazu bieten freinavigierende Lösungen einen geringeren Aufwand bei Änderungen an den Transportwegen. Hierbei ermittelt jedes Fahrzeug die Position im Raum mittels Laserscanner oder gerasterten Markierungen, z.B. Magneten oder RFID-Tags, im Boden [31]. Dennoch können sich die Fahrzeuge nach heutigem Stand der Technik in industriell verfügbaren Lösungen nicht frei bewegen, sondern folgen immer einer vorgegebenen virtuellen Spur, die niemals verlassen werden darf. Zwar entfällt der mechanische Aufwand zur Änderung der Routen in wandelbaren Produktionsumgebungen, allerdings ist hierbei ein umfangreicher Konfigurationsaufwand mit anschließenden Tests und Nachkorrekturen erforderlich. Neben der Notwendigkeit von spezifischem Fachwissen, haben diese lange Inbetriebnahme- und Konfigurationszeiten zur Folge.

Die dabei manuell definierten Streckenabschnitte bestehen aus Geraden, Kurven, Kreuzungen und Verzweigungen (vgl. Bild 2.6). Die Fahrzeuge, die sich entlang dieser Streckenabschnitte bewegen, kennen nicht den gesamten Pfad zum Ziel, sondern nur den lokalen Bereich zum nächsten Verzweigungspunkt. Sie verfügen weder über Informationen zum Transportauftrag noch über die kürzeste

bzw. geeignete Route zwischen zwei Übergabebahnhöfen. Hierbei wird die Planung der gesamten Route zwischen den Bahnhöfen für einen Transportauftrag vom zentralen Leitrechner übernommen, der den gesamten Wegeplan und die Position der Bahnhöfe zur Werkstückübergabe vorhält [32]. Dieser gibt den Transportfahrzeugen die einzelnen Fahrbefehle zum Befahren einer Strecke oder zum Abbiegen an vordefinierten Kreuzungspunkten vor (Bild 2.4). Zudem ist dieser über die Positionen und Zustände aller Fahrzeuge informiert und kann mit aktuellen Zustandsdaten für jeden Transportauftrag eines jeden Fahrzeugs den geeigneten kürzesten Weg unter Berücksichtigung von Staus und blockierten Wegen ermitteln. Im Falle von freinavigierenden Fahrzeugen sind die Streckenabschnitte in den Fahrzeugen hinterlegt, um die Regelung zwischen Soll- und Ist-Position zu ermöglichen. Bei physisch gebundenen Navigationsstrategien benötigen die Fahrzeuge keinerlei Informationen über die Streckenabschnitte, da diese im Boden markiert sind.

Starre, spurgeführte Systeme haben den Nachteil, dass Hindernisse im Fahrweg der Fahrzeuge zum Stillstand des Fahrzeugs bzw. zum Blockieren der Fahrtstrecke führen. Befindet sich im Fahrweg ein Hindernis, müssen die Fahrzeuge stehen bleiben, die Störung der Leitsteuerung melden und solange warten, bis ein Mitarbeiter das Hindernis entfernt hat. Dadurch können Staus entstehen, die nur dann umgangen werden können, wenn vordefinierte Ausweichrouten existieren. Service-Fahrzeuge in öffentlichen Einrichtungen oder im privaten Bereich sind mit Mechanismen zur Abtastung der Umgebung und zum automatischen Ausweichen von Hindernissen ausgestattet [28]. Allerdings sind in Produktionen solche Mechanismen noch nicht umgesetzt worden.

### 2.3.2 Defizite kommerziell verfügbarer Fahrerloser Transportsysteme

Die unmittelbare und schnelle Anpassung der Fahrerlosen Transportfahrzeuge an die veränderten Produktionsbedingungen bzw. das veränderte Produktionsumfeld ist mit bisherigen Steuerungslösungen nicht zu erzielen. Dies liegt zum einen an den zahlreichen manuellen Konfigurationsschritten, die aus einem hohen zeitlichen Konfigurations- und Validierungsaufwand resultieren. Zum anderen bieten die derzeitigen Steuerungssysteme und Mechanismen keinen ausreichenden Funktionsumfang an, mit dem flexibel auf Störungen oder unmittelbare Veränderungen im Ablauf reagiert werden kann. Die Nachteile werden im Folgenden erläutert:

- **Vordefinierte Spurführung:** das starre Spurführungskonzept, das nur vordefinierte Wege zulässt, birgt den Nachteil einer höheren Störempfindlichkeit. Im Falle von Staus, i.d.R. hervorgerufen durch Hindernisse auf den Transportwegen, können diese nicht automatisiert kompensiert werden. Die Fahrwege sind i.d.R. nur unidirektional befahrbar, wodurch Staus nur mit manuellem Eingriff aufgelöst werden können. Darüber hinaus bietet diese Technologie keine Möglichkeit zur Laufzeit Wege neu zu bestimmen und diese gemäß den aktuellen Anforderungen umzugestalten, um somit bessere Leistungsmerkmale, wie einen erhöhten Durchsatz, erzielen zu können.
- **Fachpersonal:** Die Inbetriebnahme und Konfiguration eines Fahrerlosen Transportsystems kann ausschließlich von Fachpersonal vorgenommen werden. Dies umfasst eine Installation der Fahrzeuge vor Ort, die Einrichtung des Orientierungssystems bei freinavigierenden Lösungen bzw. des liniengeführten Spurführungssystems, verbunden mit den erforderlichen Bodeninstallationen und die Konfiguration der Steuerungssoftware. Bei freinavigierenden Strategien müssen

die Wege, inkl. der Verzweigungen, Kreuzungen und Übergabestellen zu den Bearbeitungsstationen oder Lägern, am PC in einer Karte der Produktionshalle manuell definiert werden. Für Änderungen an der Anlage, wie die Einbindung eines neuen Fahrzeugs oder umfassende Änderungen an den Transportwegen, ist Fachpersonal erforderlich. Somit sind diese Systeme nicht für den Einsatz in flexiblen, wandelbaren Produktionsbedingungen geeignet, die ad hoc Veränderungen der Anlagenstruktur und den Abläufen vorsehen.

- **Vollständige Vermessung der Produktionshalle:** Zur Erfassung der Fahrwege in der Produktionsumgebung in freinavigierenden Fahrerlosen Transportsystemen ist das Kartenmaterial des bestehenden Anlagenaufbaus erforderlich, in dem manuell die Fahrwege eingezeichnet werden können. Kartenmaterial einer jeden Produktion steht immer aus der Phase der Planung zur Verfügung, jedoch stimmen diese Daten oftmals nicht mit den realen Aufbauten überein, da beim Anlagenaufbau oftmals kurzfristige Änderungen vorgenommen werden, ohne diese im Kartenmaterial zu vermerken. Folglich werden zur Inbetriebnahme von Fahrerlosen Transportsystemen im Vorfeld Vermessungsfirmen beauftragt, um die Produktionshallen nochmals kartographisch mit hoher Genauigkeit zu erfassen. Dieses detaillierte und exakte Kartenmaterial stellt die Grundlage zur Planung bzw. Spezifikation der Fahrwege und der Übergabestellen dar.
- **Aufwändige Validierung der Wege und Positionen:** Zwar kann die Vermessung der Halle von den Vermessungsfirmen millimetergenau vorgenommen werden, jedoch ist die Validierung der Fahrwege und erfassten Positionen der Übergabebahnhöfe unerlässlich. Hierzu muss ein Mitarbeiter während der ersten Tests dem fahrenden Fahrzeug folgen, alle möglichen Kollisionen mit Hindernissen überprüfen und die Warenübergabe an allen erfassten Bahnhöfen validieren. Dies ist mit einem umfassenden Korrektur-Aufwand verbunden, da Probleme, die sich während der Validierungs-Phase ergeben, weitere manuelle Korrekturen im Datenbestand und erneute Tests zur Folge haben. Der Zeitaufwand für die Validierungs- und Korrektur-Phase ist sehr groß und maßgeblich für den langen Inbetriebnahmeaufwand vor Ort sowie den ebenso damit verbundenen großen Änderungsaufwänden im Falle von Umbaumaßnahmen.
- **Manuelle Festlegung der Routen und Alternativwege:** Um Störungen in den Transferstrecken kompensieren zu können, müssen im Streckennetz Alternativwege zur Verfügung stehen, die manuell definiert werden müssen. Dies bedeutet, dass für jedes möglich gestörte Streckensegment mindestens ein Alternativweg berücksichtigt werden muss. Zudem verfügen heutige Fahrerlose Transportsysteme nicht immer über die Möglichkeit in einem Streckennetz den kürzesten Weg zum Ziel automatisch abzuleiten, sondern es werden noch immer vielfach die Routen zwischen den Bahnhöfen manuell hinterlegt. Dies betrifft auch die Alternativrouten im Falle von Störungen.
- **Kompensation von Störungen nur über explizit definierte Regeln:** Neben der Erfassung der Fahrtstrecken müssen ebenso Ablaufregeln definiert werden. Diese beinhalten hauptsächlich die Steuerung der Aufnahme von Werkstücken, den Transport und die Abgabe. Um im Falle einer Störung den Materialfluss umorganisieren zu können, sind ebenso Alternativregeln erforderlich, die den Betrieb soweit wie möglich aufrechterhalten. Bisherige Systeme sind nicht in der Lage automatisiert Alternativen zu ermitteln bzw. die Abläufe zur Laufzeit den Bedürfnissen anzupassen, sondern ermöglichen lediglich die manuelle Definition und Eingabe fest definierter Regeln. In komplexen, unüberschaubaren Materialflusssystemen können die vielfältigen

Abhängigkeiten und möglichen Kettenreaktionen im Störfall im Vorfeld nur bedingt vom Mitarbeiter erfasst und überblickt werden. Somit wird im Regelfall lediglich eine Untermenge möglicher Kompensationsstrategien analysiert, erfasst und in explizite Kompensationsregeln überführt.

- Stilllegung der Anlage bei Änderungen: Im Falle von Änderungen am Fahrerlosen Transportsystem muss - abhängig von der Art der Änderung - die Gesamtanlage oder Teile hiervon stillgelegt werden. Dadurch muss der Materialfluss unterbrochen werden, wodurch sich die unproduktive Zeit der Gesamtanlage erhöht. Kommerziell verfügbare Systeme bieten i.d.R. keine Mechanismen an, um zur Laufzeit Änderungen ohne Unterbrechung einzupflegen.

Zusammenfassend ist festzustellen, dass der Inbetriebnahmeaufwand von Fahrerlosen Transportsystemen nach dem bisherigen Stand der Technik sehr hoch ist und Expertenwissen sowie ausgebildete Fachkräfte benötigt werden. Somit ist der Einsatz dieser führerlosen Fahrzeuge in wandelbaren Produktionsumgebungen, die von ad hoc Umbaumaßnahmen oder Veränderungen des Materialflusses zur Laufzeit geprägt sind, nicht geeignet.

### 2.3.3 Entwicklungs- und Inbetriebnahmeaufwände

Aufbauend auf den in Kapitel 2.3.2 erläuterten Defiziten derzeit kommerziell verfügbarer Fahrerloser Transportsysteme werden im Folgenden die Schritte zur Inbetriebnahme detailliert beschrieben. Im späteren Verlauf der Arbeit werden auf diese zur Motivation der neuen Konzepte und Ansätze Bezug genommen.

#### Zentraler Leitreechner

Fahrerlose Transportsysteme bestehen nach dem heutigen Stand der Technik aus einem Leitreechner und den Fahrzeugen. Da der zentrale Leitreechner alle Vorgänge innerhalb des Transportsystems koordiniert und überwacht, ist an dieser Stelle eine hohe Verfügbarkeit erforderlich, nachdem dieser einen Single-Point-of-Failure darstellt, d.h. fällt der Leitreechner aus, steht das gesamte Fahrerlose Transportsystem still. Die Verfügbarkeit kann nur durch die Errichtung eines redundant ausgelegten Systems erreicht werden, das einen Ausfall zur Laufzeit kompensieren kann. Hierzu wird neben dem zentralen Leitreechner ein zweiter Slave-Rechner redundant installiert, der analog zum Master-Leitreechner die Zustände des gesamten Transportsystems mitverfolgt. Dies erfordert bereits bei der Entwicklung der Steuerungssoftware und der Auslegung eines redundant ausgelegten Leitstands einen erheblichen Mehraufwand. Die zusätzliche Sicherheit wirkt sich ebenso nachhaltig auf die Anwendung dieses Systems beim Kunden aus, da die Hardware zusammen mit der Software doppelt installiert und dementsprechend konfiguriert werden müssen. Zudem erfordert der zentrale Leitstand Stromunterbrechungsversorgungen, um selbst Stromschwankungen und kurzzeitige Stromausfälle kompensieren zu können. Zur Gewährleistung der unmittelbaren Funktionsbereitschaft nach unvorhergesehenen Neustarts des Leitrechners, ist es erforderlich, den Zustand der Gesamtanlage nach jedem Wechsel der Teilstände persistent zu speichern. Dies erfordert zusätzliche transaktionale Mechanismen bei der Verwaltung und Speicherung der Auftrags- und fahrzeugbezogenen Zustandsdaten in der lokalen Datenbank, die analog zur Steuerungssoftware redundant, sowohl auf dem Master- als auch auf dem Slave-Rechner, ausgelegt sein muss.

## **Fahrerlose Transportfahrzeuge**

Die fahrerlosen Transportfahrzeuge verfügen, wie in Kapitel 2.3.1 beschrieben, über einen geringen Funktionsumfang. Da sie Befehlsempfänger des Leitstands sind, müssen sie nur geringfügige Koordinierungsaufgaben übernehmen, die sich auf die genannten Prozesse konzentrieren. Somit ist der Einrichtungsaufwand der Steuerungssysteme in den Fahrzeugen gering, jedoch müssen diese inklusive der Parameter und ihrer Type im Leitrechner manuell erfasst werden. Zwar besteht hierbei die Möglichkeit Fahrzeugtypen zu hinterlegen, jedoch werden Fahrerlose Transportfahrzeuge in den meisten Fällen speziell für den Kunden und die Anwendung angefertigt, die dennoch eine manuelle Erfassung am Leitstand erfordern. Zudem muss ggf. das Rollenverhalten der Fahrzeuge manuell erfasst werden, wenn es sich um unterschiedliche Fahrzeuge mit unterschiedlichen Funktionen oder verschiedenen Aufgaben handelt.

## **Erfassung der Fahrwege**

Nach dem Aufbau des Leitrechners und dem Aufbau der Fahrzeuge sowie der Integration dieser in der zentralen FTS-Software, müssen die Fahrwege und Übergabebahnhöfe applikationsspezifisch im zweidimensionalen Raum der Fertigungshalle festgelegt werden. Hierzu ist detailliertes Kartenmaterial erforderlich, das aus unterschiedlichen Quellen herangezogen werden kann (Kapitel 2.3.2).

Das vorliegende Kartenmaterial muss zunächst in ein geeignetes Format transformiert werden, damit es von der Konfigurationssoftware für das Fahrerlose Transportsystem eingelesen werden kann. Da die Pläne jedoch oftmals in unterschiedlichen Softwareapplikationen mit unterschiedlichen Datenformaten vorliegen und selbst standardisierte Formate herstellerspezifische Unterschiede aufweisen, sind hierbei umfassende Import-Schnittstellen erforderlich. Zudem muss das verfügbare Kartenmaterial manuell in das bestehende Koordinatensystem des Fahrerlosen Transportsystems ausgerichtet und relativ zum Ursprung verschoben werden. Diese Tätigkeit kann dem Fachpersonal nicht mittels Automatismen abgenommen werden, da Referenzmarkierungen im Kartenmaterial fehlen und diese manuell ausgewählt bzw. zur manuellen Berechnung herangezogen werden müssen.

Dem Fachpersonal zur Einrichtung des Fahrerlosen Transportsystems wird das Kartenmaterial nach der Übernahme in einer Oberfläche dargestellt (vgl. Bild 2.6). Mit dieser Software ist es möglich, graphisch zweidimensional die Wege einzuzeichnen. Hierzu stehen verschiedene Standardelemente, wie gerade Streckenabschnitte, Kurven oder Kreuzungen zur Verfügung, die mit Mausbedienung ausgewählt und aneinandergefügt werden können. Zudem muss für jede Strecke u.a. die Fahrtrichtung spezifiziert werden. Voraussetzung zur Festlegung der Fahrwege ist die Analyse und konzeptionelle Erfassung aller Materialflüsse. Daraus muss das Fachpersonal ein vollständiges Streckennetz ableiten, das den Kapazitätsanforderungen genügt. Dieses muss verschiedenen Kriterien genügen, wie die Minimierung der Nutzfläche, ausreichende Kapazität zur Bewerkstellung des Materialflusses, Festlegung von Alternativwegen zur Störungskompensation usw.

Diese Vorgänge verursachen eine lange Konfigurationsphase, die hohe Personalkosten zur Folge haben. Zudem sind einfache und schnelle Änderungen am Wegenetz und dem Materialfluss nicht möglich, da hierzu zumindest geschultes Personal erforderlich ist, das die Software bedienen kann. Zudem erhöht jede manuelle Anpassung das Fehlerrisiko in der Konfiguration.

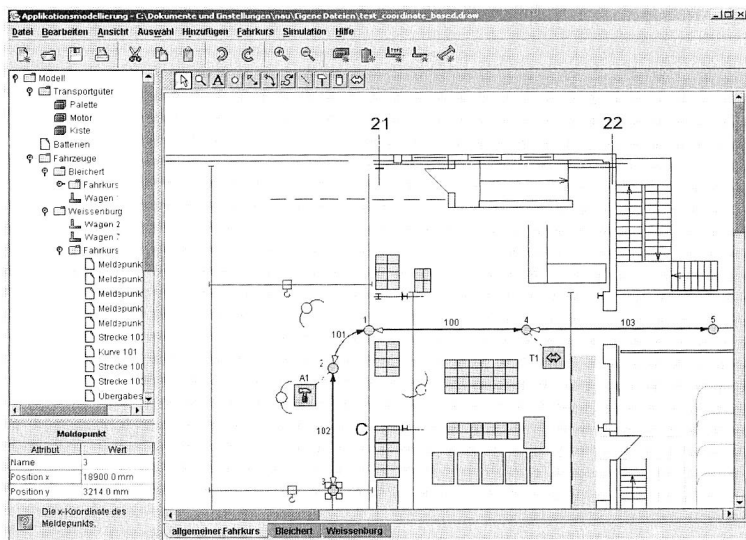


Bild 2.6: Bildschirmauszug der OpenTCS-Anwendung zur manuellen Spezifikation der Fahrwege inklusive der Kreuzungen und der Übergabestellen. [29]

## Einbindung der Übergabebahnhöfe und Bearbeitungsstationen

Auf technischer Seite sind neben der Inbetriebnahme der Fahrzeuge auch die softwaretechnische Integration der Übergabebahnhöfe erforderlich. Diese müssen über das Netzwerk mit dem zentralen Leitrechner kommunizieren können, um anstehende Transportaufträge mitzuteilen. Der zentrale Leitrechner nimmt diese Transportaufträge entgegen und verteilt sie auf die Fahrzeuge. Voraussetzung sind bestehende Softwareschnittstellen zwischen dem Leitrechner und den Bearbeitungsstationen, die teilweise individuell definiert werden müssen - abhängig davon, welches Steuerungsgerät bzw. welche Steuerungssoftware auf Seiten der Bearbeitungsstation bzw. Übergabebahnhöfe zu Grunde liegt. Hierzu muss der Softwareentwickler entweder auf Seiten der Bahnhöfe oder auf Seiten des FTS-Leitrechners geeignete Softwareschnittstellen entwerfen und umsetzen. Dies erfordert in einem nicht homogenen Verbund von Teilnehmern mit unterschiedlichen Steuerungssystemen einen erheblichen manuellen Mehraufwand. Zudem muss die Interaktion zwischen dem Leitrechner und jedem Bahnhof einzeln auf mögliche Implementierungsfehler validiert werden, was einen erheblichen Zeitaufwand verursacht.

## Validierung der Fahrwege

Nach erfolgreicher Eingabe der Fahrwege und der Positionen der Übergabebahnhöfe in der Karte, werden diese aus der Konfigurationssoftware an den zentralen FTS-Leitrechner übertragen. Der Leitrechner besitzt lediglich die Positionsangaben der Strecken und der Bahnhöfe - die Hindernisse sind für den laufenden Betrieb unerheblich. Erst jetzt sind alle Voraussetzungen gegeben, die Steuerungssoft-



ware und Fahrzeuge mit dem vollständigen Datenmaterial im Produktionsverbund zu starten. Jedoch ist es erforderlich die Fahrwege und Übergabestellen zu prüfen. Hierzu wird ein Fahrzeug in Betrieb genommen, das, begleitet von einem Mitarbeiter, die vollständigen Wege abfährt. Dabei werden die Positionen der Übergabebahnhöfe kontrolliert, mögliche Kollisionen mit Hindernissen und eventuelle Gefahrensituationen mit anderen vom Menschen geführten Fahrzeugen oder Mitarbeitern geprüft. Zudem werden alle Sicherheitsmechanismen, wie das Erkennen von neuen Hindernissen im Fahrweg oder das Registrieren einer Störung am Fahrzeug, überprüft und getestet. Sollten Änderungen, insbesondere an den Fahrwegen oder Positionsdaten, erforderlich sein, muss wiederum das Fachpersonal mit der Konfigurationssoftware diese Änderungen vornehmen und den Validierungsprozess zumindest an der betroffenen Stelle wiederholen. Diese Notwendigkeit hat einen aufwendigen und zeitbehafteten Prozess zur Folge, der selbst bei Änderungen an der Organisation des Materialflusses teilweise erforderlich ist. Erst nach mehrmaligen Durchläufen mit einem Fahrzeug wird die Anlage für den Probetrieb eingesetzt und hierbei der komplette Materialfluss, die Interaktion mit den Bahnhöfen und die Auslastung der Fahrzeuge bzw. der Streckenabschnitte vollständig getestet. Erst nach Erfüllung aller Randbedingungen und Zielvorgaben kann die Anlage für den Betrieb freigegeben werden.

## 2.4 Der Multi-Agentenansatz

Eine Lösung der in Kapitel 2.1 und 2.2 angesprochenen Defizite bestehender Steuerungssysteme im wandelbaren Umfeld liegt im Ansatz, autonome und abgeschlossene physische und steuerungstechnische Komponenten zu entwickeln, die zudem softwaretechnisch autarke Entscheidungsgewalt besitzen. Dabei ist es von Vorteil, wenn die physischen Komponenten und ihr zugehöriges Steuerungssystem eine Einheit bilden. Diese Konzepte werden vom Agenten-Ansatz unterstützt, der eine kollaborative Zusammenarbeit mehrerer, verteilter Einheiten zur Bewältigung einer gemeinsamen Aufgabe, in diesem Fall zur Herstellung der angeforderten Produkte, vorsieht. Im Folgenden werden der Agentenbegriff und das zugrundeliegende Konzept zunächst definiert (Kapitel 2.4.1). Zur Auslegung eines umfassenden Agentensystems existieren verschiedene Modellierungsformen, um das kooperierende Verhalten zwischen den Agenten spezifizieren zu können (Kapitel 2.4.3). Im Verbund kann den Agenten unterschiedliches Verhalten zugewiesen werden, die sich, wie in Kapitel 2.4.4 beschrieben, über Agententypen klassifizieren lassen. Dabei lässt sich der Agentenansatz auf zahlreiche Bereiche der Produktion anwenden. Ein Auszug aus möglichen und weit verbreiteten Anwendungsgebieten wird in Kapitel 2.4.5 vorgestellt.

### 2.4.1 Definition des Agenten-Begriffs

Der Agenten-Begriff ist nicht klar definiert, da dieser im Speziellen unterschiedlich ausgelegt wird. Somit konnte sich keine allgemeingültige Definition etablieren. Ein Problem liegt in der klaren formalen Spezifikation des Agentenverhaltens und seiner besonderen Aufgabe und Funktionalität. Dies liegt darin begründet, dass die Grenze zwischen der klassischen Software und dem Agenten nicht klar gezogen werden kann. Als Beispiel wird im Folgenden eine Definition des Agentenbegriffes für den Einsatz in der Produktion von Wooldridge zitiert:

An agent is considered a software entity situated in a flexible production environment, with enough intelligence that is capable of autonomous control actions in this environ-

ment and of co-operation relationships by participating in associations agreements with other entities in order to meet its design objectives. [33]

Ritter definiert den Agentenbegriff wie folgt:

Ganz allgemein kann als Agent jemand gelten, der im Auftrag eines anderen handelt. Auf technische Systeme übertragen, können Agenten als autonome, kooperierende Entitäten in verteilten, dezentralen Systemen verstanden werden. [34]

Idealerweise lässt sich ein Agent mittels folgender Charakteristika beschreiben [35, 36]:

- Autonomie: Ein Agent handelt unabhängig von Benutzereingaben oder direkten Eingriffen anderer Agenten. Sie kennen ihre Fähigkeiten und kontrollieren autark ihren inneren Zustand.
- Proaktiv: Agenten können gemäß einer zielgerichteten Verhaltensstrategie, die in Form von Regeln hinterlegt ist, initiative Entscheidungen treffen.
- Reaktiv: Sie können die Umgebung wahrnehmen und auf diese reagieren.
- Sozial: Agenten interagieren mit anderen Agenten in einer Organisationseinheit zur Erfüllung einer gemeinsamen Aufgabe [37].
- Anpassungsfähig: Besitzen die Fähigkeit von Experimenten und Beobachtungen zu lernen. [38].
- Verklemmungsfrei: Die hinterlegten Regeln müssen Konfliktsituationen erkennen und Verklemmungen vermeiden bzw. auflösen.
- Fair: Die Agenten dürfen keinen benachbarten Agenten unterdrücken oder nicht nur ihr eigenes Ziel ohne Rücksicht auf Verluste durchsetzen.
- Lernfähig: Sie besitzen ein gewisses Maß an Lernvermögen, um aus Beobachtungen oder Handlungen Erfahrungen abzuleiten [39].
- Mobil: Eine Unterklasse der Agenten kann sich in einer Umgebung selbständig bewegen.

Agenten agieren immer als Repräsentanten für ein physisches oder ein Informations-Objekt und handeln im Namen und im Sinne bzw. Interesse des zugeordneten Objektes. Hierfür muss der Agent die Ausprägung und Fähigkeiten des Objektes kennen, für das er stellvertretend handelt. Ein physisches Objekt kann eine Bearbeitungsstation, ein Werkzeug oder ein Werkstück sein. Im Gegensatz dazu stellt ein Fertigungsauftrag ein virtuelles, d.h. ein informationelles Objekt, dar. Der FIPA-Standard [40], der bislang einzige Standard für Agentensysteme, unterteilt diese in Softwareagenten und technische Agenten [41]. Um die Rolle eines Repräsentanten übernehmen zu können, benötigt der Agent ein Modell des Objektes, das die Struktur, Funktionanlilität und seine Zustände beschreibt. Dies ist notwendig, um mit anderen Agenten und Objekten, wie dem Steuerungssystem einer Maschine, zu interagieren.

### 2.4.2 Interaktionen im Multiagentensystem

Die Agenten agieren immer in einer Gemeinschaft mit anderen Agenten. Mehrere Agenten, die in einer Organisationseinheit zusammengefasst sind, werden als *Multiagentensysteme* [42] bezeichnet. In der Gemeinschaft sind sie für die gemeinsame Lösung einer komplexen Aufgabenstellung verantwortlich [43]. Zudem werden im Verbund die Rollenverteilungen, d.h. die Aufgaben und das Verhalten der Agenten zueinander, festgelegt. Voraussetzung für das Multiagentensystem ist eine ausreichende Infrastruktur, bestehend aus einem Kommunikationsnetzwerk, das die flexible Interaktion zwischen Agenten unterstützt, sowie Verwaltungsdienstleistungen anbietet, z.B. zur Registrierung der vorhandenen Agenten, die selbst zur Laufzeit angebunden oder entfernt werden können [44, 45]. Diese erforderliche Infrastruktur wird als Agentenplattform [46] bezeichnet und umfasst sowohl die Hardware als auch Basissoftware, die zur Erzeugung und Verwaltung der einzelnen Softwareagenten notwendig ist [34].

Das Multiagentensystem ist geprägt von Interaktionen zwischen den gleichberechtigten Agenten in Form von Nachrichtenaustausch. Im Agentenmodell sind prinzipiell keine übergeordneten Agenten vorgesehen, die untergeordneten Agenten Befehle oder Anweisungen vorgeben. Vielmehr besteht die Absicht, Informationen auszutauschen, damit jeder Agent für sich und zum Erreichen des gemeinschaftlichen Gesamtzieles die beste Entscheidung treffen kann. Zur gemeinsamen Strategiefindung im Verbund autonomer Agenten kommen als Interaktionsmechanismen zwischen den Agenten überwiegend Verhandlungsmechanismen zum Einsatz. Mit diesen können die Agenten zur Laufzeit eine gemeinsam getragene Übereinkunft zur weiteren Vorgehensweise finden. Hierbei werden Vorgänge aus dem Leben des Menschen in das Multiagentensystem abgebildet. Ein weit verbreitetes Rollenverhalten aus der menschlichen Geschäftswelt stellen Dienstgeber, die eine Dienstleistung anbieten, und Dienstnehmer, die eine Dienstleistung bei den Dienstgebern in Anspruch nehmen, dar. Das Multiagentensystem kann hierzu einen virtuellen Marktplatz zur Verfügung stellen, auf dem sich die Agenten finden und Dienstleistungen vermitteln können. Wie dieser Marktplatz eingerichtet wird und welche Voraussetzungen dazu erforderlich sind, hängt von den Interaktionsmechanismen ab. [40]

In der Produktion kann der Agentenansatz generell an allen Stellen eingesetzt werden, an denen zur Laufzeit Entscheidungen zu treffen sind. Zunächst werden die möglichen Anwendungsgebiete in folgende zwei Problemstellungen, die vom Verhandlungsgegenstand abhängig sind, unterteilt: die Zuordnungsproblematik und die Einigungsproblematik.

Bei der *Zuordnungsproblematik* sind die Verhandlungsgegenstände Stoffe oder Informations-Objekte. Beispiele für physische Objekte sind Werkstückträger mit zu bearbeitenden Bauteilen oder Betriebsmittel, die auf Ressourcen verteilt werden müssen. Ein Informations-Objekt kann ein Zugriffsobjekt sein, das dem Besitzer einen exklusiven Zugriff auf einen bestimmten Gegenstand oder eine Informationsmenge ermöglicht. Im Gegensatz dazu behandelt die *Einigungsproblematik* Fragestellungen, wie eine Tätigkeit ausgeführt werden kann. Hierbei einigen sich zwei oder mehr Teilnehmer über eine Fertigungsgröße, z.B. die Temperatur, oder die optimale Festlegung der Reihenfolge. Hierzu sind, abgestimmt auf die Anwendung, geeignete Verhandlungsverfahren auszuwählen oder zu entwickeln, die, angepasst auf den Verhandlungsgegenstand, die gewünschte Fragestellung beantworten.

Der Vorzug des Agentenansatzes liegt in der Realisierung verteilter, disjunkter Funktionen mit abge-

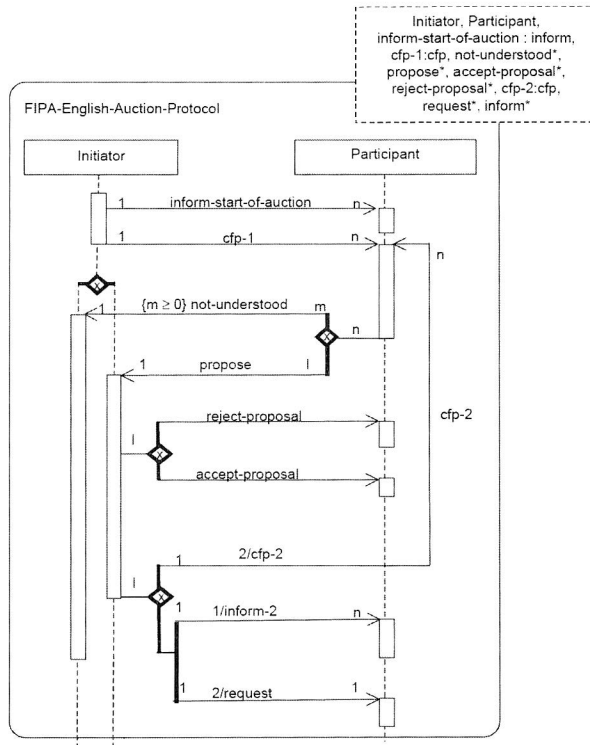


Bild 2.7: Interaktionsprotokoll der Englischen Auktion gemäß dem FIPA-Standard. Mittels Verhandlungen kann ein Initiator bei anderen Teilnehmern ein Gebot anfragen. Die eingegangenen Gebote werden als Entscheidungsgrundlage, z.B. zur Wahl der nächsten Ressource, herangezogen. [40]

geschlossenem und überschaubarem Funktionsumfang. Somit werden die Strukturen und Vorgänge für den Menschen viel besser nachvollziehbar. Gleichzeitig bietet die Entscheidungsfindung zur Laufzeit eine hohe Flexibilität, die dynamische Schwankungen oder Veränderungen unmittelbar abfangen oder kompensieren kann. Die Verhandlungsmechanismen bieten dabei in vielen Fällen eine implizite Optimierung der Abläufe an, ohne dabei zusätzliche Optimierungsstrategien implementieren zu müssen. Ein Beispiel eines der bekanntesten und am meisten eingesetzten Verhandlungsmechanismen ist die englische Auktion (Bild 2.7). Der Initiator, z.B. ein Dienstnehmer, der ein Werkstück repräsentiert, sendet dabei eine Anfrage an alle relevanten Teilnehmer, in diesem Beispiel den Bearbeitungsstationen als Dienstgeber. In dieser Anfrage informiert der Dienstnehmer die Dienstgeber über den Verhandlungsgegenstand, in diesem Fall die Bearbeitung eines Werkstücks, und fordert die Dienstgeber auf, ein Gebot abzugeben. Die Dienstgeber antworten mit einem Gebot und schätzen dabei die anfallenden Kosten oder die Zeit. Nachdem der Dienstnehmer alle Gebote gesammelt hat, entscheidet

er sich immer für das beste Gebot. Somit reguliert sich das System selbständig und die Last wird gleichmäßig und implizit, d.h. ohne weitere Optimierungsregeln, auf alle Teilnehmer, d.h. hier auf die Bearbeitungsstationen, verteilt.

### 2.4.3 Modellierungsformen von Agenten

Agenten bilden zunehmend das Verhalten der menschlichen Welt nach, indem sie mit artverwandten Fähigkeiten, wie verschiedenen Formen der Wahrnehmung und der Handlung im menschlichen Umfeld, ausgestattet werden. Die Problematik besteht insbesondere in der formalen Beschreibung dieser Verhaltensweisen und der Klassifikation der Agenten für die verschiedenen Einsatzgebiete [33]. Dabei soll die Modellierung so allgemein wie möglich gehalten werden. Im Folgenden wird ein kurzer Überblick verschiedener, grundlegender Ansätze der funktionalen Modellierung gegeben. [47]

#### Zustandsorientierte Modellierung

Die Zustandsmodellierung ordnet jedem Agenten eine bestimmte Menge von möglichen, internen Zuständen zu. Die Zustandsänderungen werden entweder durch Ereignisse oder durch Aktionen ausgelöst. Zur Modellierung der inneren Zustände und ihrer möglichen Abfolge bieten sich verschiedene Formalismen, wie nicht-endliche Automaten oder Petri-Netze [48], an. Die Möglichkeiten der zustandsorientierten Modellierung sind begrenzt, da jedem Agenten nur eine endliche Anzahl definierter Zustände zugeordnet werden kann. Dennoch bietet diese Modellierungsform dem Entwickler die Möglichkeit, die vorgesehenen Abläufe und Vorgänge zu überblicken bzw. diese im Nachhinein im Falle von Änderungen nachzuvollziehen.

#### Objektorientierte Modellierung

Gemäß dem objektorientierten Ansatz, wird in diese Modellierungsform das Prinzip des Information-Hiding übernommen, wonach der innere Zustand jedes Agenten nach außen hin abgeschirmt ist. Er kann nur durch eine, von außen zugängliche Eingangs-Operation verändert werden. Diese ist nicht weiter spezifiziert. Hierbei kann es sich um eine per Kommunikationsmedium übermittelte Information oder ein Sensorsignal handeln. Nach der Änderung des internen Zustands wird die Operation *Abgeben* initiiert. Jedoch eignet sich diese Modellierungsart nur als Basis-Modellierung, da sie sehr abstrakt und wenig differenzierend ist.

#### Absichtenorientierte Modellierung (Intentions)

Das absichtenorientierte Modellieren stellt eine der ausdrucksstärksten Modellierungsmethoden dar, bei der davon ausgegangen wird, dass die Agenten zu motiviertem Handeln fähig sind. Sie werden mit Absichten ausgestattet, die Pläne, Ziele oder Wünsche sein können. Zur Verwirklichung der Absichten muss der Agent initiative Handlungen eingehen. Die absichtenorientierte Modellierung ist sehr umfangreich, aber ebenfalls nur für theoretische Fragestellungen geeignet, da die praktische Umsetzung aufgrund der Komplexität des Modells kaum möglich ist.

## Rollenorientierte Modellierung

Das Prinzip des Agentenansatzes besagt, dass die Agenten soziales Verhalten besitzen und sich mit anderen Agenten koordinieren müssen (Kapitel 2.4.1). Hierbei muss die Aufgabenverteilung mehrerer Agenten im Verbund klar geregelt sein. Die rollenorientierte Modellierung sieht insbesondere vor, jedem Agenten eine bestimmte Rolle zuzuweisen, die er zur Laufzeit erfüllen muss. Sie muss nicht zwingend fest zur Entwicklungszeit spezifiziert werden, sondern kann sich selbst zur Laufzeit ändern. Die Rolle legt sein Verhalten, seine Attribute und seine Ziele fest. Ein Beispiel ist das Dienstnehmer-Dienstgeber-Modell, in dem ein Agent die Rolle des Dienstgebers übernimmt und seine Dienste anbietet auf die der Dienstnehmer-Agent zugreifen kann. Diese Dienste bzw. Rollen können beispielsweise Datendienste zur Erfassung von Maschinen- oder Betriebsdaten sein, die im Falle von Ausfällen oder Umbaumaßnahmen auf andere Teilnehmer übergehen. Ein weiteres Beispiel sind kooperierende Maschineneinheiten, die im Falle von Ausfällen die Rollenverteilung der koordinierenden Einheit neu bestimmen. Zwar ist dieses Modell für den praktischen Einsatz gut geeignet, dennoch ist es erforderlich, einen operationsbasierten Formalismus zur formalen Erfassung der Rollen anzuwenden.

### 2.4.4 Agententypen

Neben den Modellierungsformen werden in der Literatur grundsätzlich drei verschiedene Agententypen unterschieden. Diese charakterisieren das grundsätzliche Verhalten der Agenten, abhängig davon, wie sie auf Reizeinflüsse aus der Umgebung reagieren, bzw. inwieweit sie zukünftige Aktionen planen. Im Folgenden werden diese Typen, die *proaktiven Agenten*, *reaktiven Agenten* und *hybriden Agenten* vorgestellt.

#### Proaktive Agenten

*Proaktive Agenten* planen vorausschauend, wann eine Aktion durchzuführen ist, oder reagieren auf Ereignisse in Form von äußeren Einflüssen. Sie zeichnen sich dadurch aus, dass sie ein symbolisches Modell der Welt besitzen, in der sie sich befinden. Das Vorgehen zur Lösung einer Aufgabe wird bereits zur Entwicklungszeit eindeutig festgelegt. Hierbei kann eine horizontale Architektur zum Einsatz kommen, in der die Planungskomponente Informationen von einer vorgelagerten Wahrnehmungskomponente erhält, diese regelbasiert verarbeitet und die daraus abgeleiteten Planungen einer nachgelagerten Handlungskomponente zur Durchführung der Aktionen weiterleitet [47]. Somit können mit diesem Ansatz selbst komplexe Steuerungsaufgaben durchgeführt werden.

#### Reaktive Agenten

Die *reaktiven Agenten*, auch als Stimulus-Response-Agenten bezeichnet [42], erfassen ihre Umgebung und können mit Hilfe ihres internen Zustands und vorgegebener Entscheidungsregeln die erforderlichen Aktionen ableiten. Im Gegensatz zu den proaktiven, d.h. planenden Agenten, besitzen die reaktiven kein Weltmodell und können somit auch keine komplexen logischen Schlussfolgerungen treffen [33]. Das Auslösen von Aktionen erfolgt bei reaktiven Agenten immer unmittelbar, ohne signifikante zeitliche Verzögerung, als eine Folge von empfangenen Eingangssignalen. Die Trennung von Planung und Ausführung der Aktionen führt im laufenden Betrieb zu höherer Störanfälligkeit und Inflexibilität. Zur Bewältigung komplexer Steuerungsprobleme ist die Integration von Planungs- und Steuerungsfunktionen erforderlich [49].

## Hybride Agenten

*Hybride Agenten* sind eine Kombination aus planenden und reaktiven Architekturen, die die Vorzüge beider Ansätze vereinen sollen. Dies beinhaltet insbesondere die schnelle Reaktionszeit und hohe Flexibilität bei den reaktiven Agenten einerseits und andererseits die Fähigkeiten komplexe Handlungen auszuführen. Darüber hinaus überwachen sie die Ausführung der Aktionen, um Abweichungen von den Zielvorgaben zu erkennen und frühzeitig darauf reagieren zu können. [50]

### 2.4.5 Zielsetzungen des Agentenansatzes

Analog der Weiterentwicklung der Computersysteme, die immer komplexer und leistungstärker werden, sind umfassendere Konzepte und Strukturen erforderlich, um die Beherrschung dieser Systeme auf Seiten der Entwicklung und Handhabung zu gewährleisten. Hierzu haben sich immer weiterführende Abstraktionsebenen etabliert, die ausgehend von den prozeduralen, funktionalen oder objektorientierten Programmierparadigmen, sich immer mehr in Richtung Absichten- und Verhaltensformen bewegen. Hierbei bieten bereits jetzt Agentensysteme geeignete Modellierungsformen an, um diesen Absichten, Ziele und Meinungen vorzugeben, nach denen sie automatisiert Verhaltensregeln ableiten. Diese sogenannte *bewusste Haltung* ist eine Abstraktionsebene, die dem Entwickler auf einfachere Art und Weise dazu verhilft, komplexe Systeme zu beschreiben und ihr Verhalten vorherzusehen. Somit bieten Agentensysteme eine nachvollziehbare Abstraktionsebene, um Strukturen und Abläufe aus der menschlich geprägten Welt in geeignete Softwaresysteme zu überführen. [51]

Die *Componentware* [52, 53] bietet bereits heute geeignete Mechanismen und Werkzeuge zur Modularisierung und Beherrschung komplexer Systeme an. Die Agententechnologie ist die stetige Weiterentwicklung der Componentware, die eine Kapselung komplexer Softwareeinheiten in modulare Komponenten vorsieht, um höhere Überschaubarkeit, Austauschbarkeit und Wartbarkeit zu erzielen [54]. Der Agent an sich beansprucht diese gleichen Ansätze auch für sich und erweitert sie hinsichtlich einer höheren Autonomie des Agenten, der insbesondere von dynamischen ad hoc Verbindungen mit anderen Teilnehmern im Agentenverbund zur Laufzeit geprägt ist. Gleichzeitig verfügt er über ein eigenständiges Verhalten, abgelegt in einem Algorithmus, der zudem die Fähigkeit besitzt die Umgebung abzutasten und zu verstehen, um eigenständig sich in der vordefinierten Umgebung zurechtzufinden und agieren zu können. Die Agententechnologie hilft dabei folgende Ziele zu erreichen:

- Die *Reduzierung der Komplexität* wird erzielt durch eine zunehmende Kapselung und das darauf angewandte Prinzip des Information-Hiding [55], das lediglich den Zugriff über die extern verfügbaren Schnittstellen ermöglicht und dabei die interne Implementierung nach außen verbirgt. Somit können komplexe Probleme in Teilprobleme zerlegt und diese unabhängig voneinander entwickelt und getestet werden.
- Eine *erhöhte Flexibilität und Skalierbarkeit* wird mit einer zu Grunde liegenden Softwarearchitektur erzielt, die neben einem universellen Kommunikationsprotokoll weiterführende Dienstleistungen, wie Verzeichnisdienste, Datenerfassungs- und Verarbeitungsdienste, anbietet, um die dynamische Integrierbarkeit und flexiblen Interaktionen zwischen beliebigen Agenten zu gewährleisten. Hierbei kommen u.a. die Ansätze der Middlewaresysteme zum Einsatz [56, 57], die eine Infrastruktur für wandelbare Steuerungssysteme zur ortstransparenten und dynamischen

Interaktion der verteilten Softwarekomponenten über Rechnergrenzen hinweg bieten. Jedoch fehlen diesen weiterführende Konzepte und Mechanismen, die den Teilnehmern im Bereich der Organisation eigenständige Handlungsfreiheit zuweisen und sie als autonom agierende und kooperierende Einheiten in einem kollektiven und kollaborativen Verbund vorsehen.

- Durch die Auslegung der Agenten an den realen Objekten, kann eine *Erhöhung der Transparenz* erzielt werden. Dies zeichnet sich insbesondere durch die einfachere Nachvollziehbarkeit der Abläufe in jedem Agenten aus. Darüber hinaus vereinfacht sich die Entwicklungsphase, da der Entwickler nicht zwischen der realen Welt und dem Steuerungsmodell umdenken muss und Verhaltensformen aus der realen Welt nahezu identisch in ein Softwaremodell übergeführt werden.
- Auf Grund der Autonomie jedes Agenten kann insgesamt eine *Erhöhung der Gesamtausfallsicherheit* erzielt werden, da es keinen Single-Point-of-Failure gibt und sich somit Einzelausfälle nicht auf das Gesamtsystem auswirken. Mit diesem Ansatz können Fehler besser ausgeglichen werden.

## 2.5 Einsatzbereiche und Konzepte für Agententechnologien in der Produktion

Für die Produktion werden im Folgenden exemplarisch die wichtigsten Bereiche, in denen der Agentenansatz eingesetzt werden kann, genannt und ihre Funktionsweise in bestehenden Implementierungen kurz beschrieben. Die Umsetzungen befinden sich zumeist noch im Forschungsstadium.

### 2.5.1 Agenteneinsatz in der Planung

Der Agentenansatz kann bereits beim Engineeringprozess und der Prozessplanung eingesetzt werden. Hierbei geht es insbesondere um gemeinschaftliche Entwicklungsaufgaben, in denen Teilentwicklungen zusammengebracht bzw. vorab für alle gleichermaßen einheitliche Vorgaben oder Schnittstellen ausgehandelt werden müssen. Der Ansatz der *Computer-Aided Process Planning (CAPP)* [58] sieht dabei einerseits die strukturelle Dekomposition der realen Welt bzw. der real aufzubauenden Objekte in sogenannte Mikro-Welten vor. Andererseits werden darauf aufbauend Probleme in Teilprobleme, wie Prozess- und Ressourcenauswahl, Planung der Inbetriebnahme, Reihenfolgeeinhaltung, Materialflussplanung oder Steuerungsprogrammierung, zerlegt. Hierzu gibt es verschiedene Ansätze zur Standardisierung der Koordinierungsaufgaben zwischen den unterschiedlichen Fachbereichen, die innerhalb eines Projektes zusammenarbeiten [59, 60, 61].

Ein weit verbreitetes Feld, in dem bereits der Einsatz der Agententechnologie untersucht und angewandt wird, ist die Ressourcenzuordnung und -reservierung. Dies umfasst zunächst die Auftragseinkaufsplanung in der Produktion. Hierbei müssen auf verschiedenen Ebenen die Kapazitäten der Anlagen geprüft und erforderliche Bauteile reserviert werden [62]. Auf der strategischen Ebene müssen Manager Ressourcen für langfristige Planungen anhand ihrer Erfahrung und Abschätzung einplanen. Auf



der Operationsebene erfolgen kurzfristige Planungen und das zeitliche Scheduling aller auftragsrelevanten Ressourcen zur Erfüllung aller Vorgaben. Dabei muss die Reihenfolge der Aufträge nach den Zielvorgaben, wie termingerechte Lieferung, Maximierung des Durchsatzes und Reduktion der Kapitalbindungskosten, bestimmt werden. Damit ein Produkt hergestellt werden kann, müssen alle dafür erforderlichen Werkzeuge, Bearbeitungsstationen und Betriebsmittel zur Verfügung stehen. Analog dazu müssen Mitarbeiter für die Produktion eingeplant werden, die an manuellen oder hybriden Arbeitsplätzen zum Einsatz kommen. Der Agentenmechanismus kann bei der langfristigen Planung der Ressourcen helfen. Fehlen bestimmte Betriebsmittel oder Bauteile zur Herstellung, müssen diese angefordert werden. Hierzu ist es erforderlich, eine Anbindung an den Wareneingang zu realisieren, der selbst repräsentiert durch einen Agenten, über Verhandlungen entscheidet, an welche Ressourcen bzw. für welche Aufträge die Betriebsmittel oder die Bauteile reserviert werden müssen. Hierbei ist ebenso der Transport und der Einbau der Geräte sicherzustellen, die i.d.R. durch den Menschen vorgenommen werden. Zur genauen Bewertung der Produktionskosten für einen Auftrag, ist es erforderlich den aktuellen Zustand und die derzeitige Auslastung der Bearbeitungsstationen in die Kostenberechnung einfließen zu lassen. Konflikte im Zeitplan und in der Einhaltung der Fertigstellungszeiten müssen erkannt und nach Möglichkeit selbständig aufgelöst werden.

Zur Erfüllung dieser Anforderungen gibt es ein Modell, das die Vereinigung aller menschlichen und maschinellen Unternehmensressourcen aus den planenden Abteilungen einerseits und der Produktion andererseits in einem verteilt agierenden Verbund anstrebt [63]. Dies wird durch eine Bibliothek generischer Ressourcen bewerkstelligt, die einzelne Unternehmensressourcen beschreiben. Ein marktbasierter Verhandlungsalgorithmus mit dem Namen *P-TATO* sieht Agenten mit unterschiedlichen Rollen vor [64]. Ein Projekt-Manager verwaltet die Projekt-Meilensteine und die den Aufträgen zugeordneten Ressourcen. Der Task-Agent ist für einen einzelnen Auftrag zuständig. Der Ressource-Manager überwacht und koordiniert eine bestimmte Ressourcen-Menge. Der Koordinator-Agent ist für die Einteilung der Ressourcen im virtuellen Markt zuständig. Eine Anwendung basierend auf diesem Modell wurde von Schneider Electric und der Daimler AG für eine flexible Transferstraße zur Zylinderkopf-fertigung entwickelt [65, 66].

### 2.5.2 Agentenbasierte Produktions- und Materialflusssteuerungssysteme

Die einem Auftrag zugehörigen Bauteile müssen in optimierter Weise auf die Bearbeitungsstationen aufgeteilt und ebenso über geeignete Wege dorthin transportiert werden. Dabei handelt es sich um ein Zuordnungsproblem, das durch die Agententechnologie gelöst werden kann. Die Agenten entscheiden zur Laufzeit nach jedem Bearbeitungsschritt, welche Bearbeitungsstation den nächsten Schritt durchführt. Ebenso muss auf diese Weise der Transport geregelt werden. Hierbei stehen alle Transportsysteme, die Quell- und Zielstation eines Transportauftrags verbinden können, in direkter Konkurrenz zueinander. Um jedoch gezielt Staus zu vermeiden, die sich durch die Handlung einzelner, autonom agierender Agenten bilden können, da diese oftmals nur lokale Sicht auf die Umgebung besitzen, wurde ein spezielles *time-out-Protokoll* [67] entwickelt. Dieses überwacht die Zuordnung der Aufträge zu den Ressourcen und greift ein, falls eine Ressource zu lange durch einen Auftrag belegt wurde. Innerhalb einer Bearbeitungsstation muss entschieden werden, welches Prozess- und Werkzeugmodul den nächsten anstehenden Prozessschritt übernimmt. Hierbei können u.a. redundante Prozess- und Werkzeugmodule in der Zelle existieren, die Schritte gleichermaßen übernehmen können. Ein agentenbasierter Mechanismus bringt zudem einen Optimierungsvorteil mit sich, da zur

Laufzeit entschieden wird, welches Prozessmodul die Bearbeitung übernimmt.

Diese vielfachen Entscheidungsprozesse konnten jedoch in der Implementierung von Schneider Electric zur Zylinderkopffertigung nicht flexibel eingebracht werden. Zwar wurde die Steuerungslogik auf einzelne Agenten verteilt, jedoch wurden diese nach festgelegten Kommunikationsabläufen gesteuert [68, 69]. Das Steuerungsverhalten des Systems ist allerdings analog zu herkömmlichen Leitsystemen vollständig deterministisch im Programmcode fixiert. Folglich konnten die Vorzüge einer entkoppelten, nicht-deterministischen Steuerungslogik und die damit verbundene hohe Flexibilität nicht genutzt werden. Zudem wurde auf kommerziell verfügbare Steuerungssysteme aufgebaut, die den Einsatz von Steuerungsagenten, die in das Steuerungssystem der Teilnehmer vor Ort integriert sind, nicht unterstützen. Die Agenten wurden über zentrale Rechner simuliert und logisch über Datenkopplungen mit den Anlagenkomponenten verbunden.

In einer weiteren prototypischen Implementierung einer agentenbasierten Fertigungssteuerung an der Technischen Hochschule Lyngby in Dänemark, wurde ebenso auf ein vorgegebenes Ablaufszenario aufgebaut, das nicht an realen Fertigungssystemen evaluiert wurde [70]. Die Projektierung von Steuerungsagenten ist auch hier nicht vorgesehen.

Im Rahmen des EU-Projektes *Plant Automation on Distributed Systems (PABADIS)* wurde eine agentenbasierte Plattform für die Leitebene geschaffen [71, 72]. Die Plattform zielt primär auf den Einsatz in der kurzfristigen Fertigungsplanung, d.h. der Zuweisung der Werkstücke zu den Maschinen. Für jedes Fertigungslos wird im ERP-System der PABADIS-Plattform ein Produktagent erzeugt, der Informationen über die notwendigen Fertigungsschritte, Steuerungsprogramme, historischen und aktuellen Daten der Fertigung hält. Die Maschinen auf der anderen Seite werden ebenso durch Agenten, sogenannte *Cooperative Manufacturing Units (CMU)* repräsentiert, die an die Geräteschnittstelle gekoppelt sind. Die Maschinenfähigkeiten können durch eine hierarchische Datenstruktur beschrieben werden, die jedoch sehr allgemein gehalten ist. Zudem ist auch die Verhaltenssteuerung der Agenten festgelegt und somit ist diese nicht auf unterschiedliche Einsatzgebiete, wie z.B. an die speziellen Anforderungen der NC-gestützten Fertigung, flexibel anpassbar.

Im Bereich der Produktionsleitsysteme ist bereits *ProVis.Agent* kommerziell erhältlich, das auf einem bestehenden, bewährten Leitstandssystem *ProVis.NT* aufbaut [73, 74]. Dieses System zielt insbesondere auf den Einsatz im turbulenten Umfeld der Fertigung und Montage ab, um in kurzer Zeit zielgerichtet Entscheidungen treffen zu können und die relevanten Informationen an Abteilungen im Unternehmen oder Zulieferer weiterzuleiten. Seine Funktionsfähigkeit wurde im Simulator erwiesen. Auch hier ist der Einsatz von integrierten Steuerungsagenten nicht vorgesehen. Dies bedeutet, dass das Leitsystem an zentraler Stelle mit den virtuellen Agenten die Planung übernimmt und über die Ferne die Geräte mit der klassischen Steuerungstechnik ansteuert. Bedingt durch aufwändige Umbaumaßnahmen und -arbeiten im Falle erforderlicher Anlagenmodifikationen, können die Vorzüge der Agententechnologie, sich schnell und einfach in das Gesamtsystem einbinden zu können, nicht vollständig genutzt werden. Somit kann auch dieser Ansatz nur bedingt für wandelbare Produktionsumgebungen genutzt werden.

Im Open-ID-Center des Fraunhofer Instituts IML in Dortmund steht für Forschungszwecke eine Testanlage für dezentral organisierte Materialflusssysteme zur Verfügung [75]. Ein Teilprojekt beschäftigt

sich mit der Entwicklung eines Multishuttle-Lagersystems, bei dem eine Vielzahl autonomer, schienegebundener Fahrzeuge mit aktiven Lastaufnahmemitteln operieren. Der Vorteil des Systems liegt in der hohen Flexibilität die Anzahl von Fahrzeugen während des Betriebs bedarfsgerecht anzupassen und somit die Kapazität gemäß den Anforderungen auslegen zu können. Die gewählte Modellierung als Multiagentensystem ordnet jedem Multishuttle-Fahrzeug einen Agenten zu, der sich um Transportaufträge bewerben kann, und einen Steuerungsagenten, der die Steuerung des Fahrzeugs übernimmt. Daneben gibt es weitere Agenten zur Auftragszuteilung, zur Steuerung der Betriebsmittel und zur Kompensation von Störungen. Dennoch laufen diese Agenten auf einem zentralen Steuerungsrechner, der simulativ die Agenten als Repräsentanten für die Komponenten hält und ihnen über Funk-Netzwerkverbindungen von zentraler Stelle Befehle vorgibt. Als Erweiterung soll das Konzept *Internet-der-Dinge* [76] umgesetzt werden, bei dem die Behälter mit RFID-Tags ausgestattet werden und somit die Behälter die Entscheidungsgewalt über die Wegewahl haben. Der Mechanismus wird verglichen mit dem bewährten E-Mail-System im Internet, bei dem ein Behälter genauso selbständig über den Weg entscheiden soll, wie eine E-Mail im Internet. In Wahrheit jedoch entscheiden nicht die E-Mails über die Weiterleitung, sondern die E-Mail-Server, die Routentabellen halten bzw. über die hierarchisch organisierte DNS-Adresse die E-Mail-Server adressieren können. Nach diesem skizzierten Ansatz müssen die Behälter entweder den Aufbau des Gesamtsystems kennen, der nach aktuellem Stand der Technik manuell bzw. aus Plandaten eingegeben werden muss oder die Gesamtroute wird vorab dem Behälter vorgegeben. Dies erschwert zudem die Lösungsfindung, um selbst Störungen, wie Staus, mit einer schnellen Reaktionszeit begegnen zu können. Die Steuerung der Kreuzungspunkte im reibschlussbasierten Stetigfördersystem wurde mit konventioneller Steuerungstechnik und herkömmlichen Konzepten umgesetzt, die hinsichtlich der flexiblen, aufwandsarmen Skalierbarkeit Nachteile mit sich bringen. Um diesen Nachteilen zu begegnen, werden in dieser Arbeit neuartige Konzepte zur automatisierten Konfiguration und Ableitung von Materialflussregeln sowie dynamischen Selbstorganisation des Materialflusses vorgestellt.

### 2.5.3 FIPA-Agentenplattform

Agentenbasierte Steuerungsansätze autonomer, mobiler Roboter wurden in den Forschungsprojekten *Agentec* [77] und *Holonic Manufacturing Systems* [78, 79] entwickelt. Der Schwerpunkt lag in der Entwicklung geeigneter Interaktionsprotokolle für die verteilte Steuerung mobiler Robotersysteme. Aus dem *Holonic Manufacturing System* wurde die FIPA-Agentenplattform [40] abgeleitet, die zum Ziel hat, geräte- und anwendungsunabhängig eine Plattform bereitzustellen, um verteilte, autonome Komponenten, selbst unterschiedlicher Hersteller einzubinden.

Die FIPA-Spezifikationen lassen sich in folgende fünf Kategorien einteilen. Die Agentenkommunikation, das Management von Agenten, der Nachrichtentransport und die Abstrakte Architektur, die die ersten drei logisch miteinander verknüpft. Die fünfte Kategorie umfasst die Anwendungen, in der u.a. die Integration von Agenten mit Nicht-Agenten-Software [80], wie Legacysoftware [81] oder Datenbanken, festgelegt ist. Somit soll die Einbindung bestehender Softwaresysteme in die Agentenplattform gewährleistet werden, indem diesen ein Agent in der Rolle eines Brokers oder Wrapper zur Seite gestellt wird. In dieser Kategorie werden auch spezielle Dienste, Ontologien und Use Cases beschrieben, um die Entwicklungsprozesse und Anwendungsauslegungen zu formalisieren.

Der Vorzug der FIPA-Agenten liegt in der offenen Auslegung der Spezifikationen, um die Akzep-

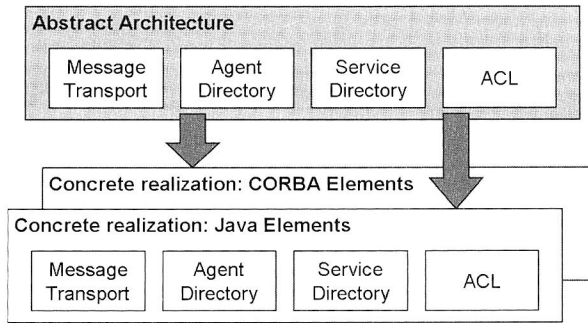


Bild 2.8: Ableitung konkreter Implementierungen von der abstrakten FIPA-Architektur, bestehend aus den Modulen: Nachrichtentransport, Agentenverzeichnis, Dienstleistungsverzeichnis und Agentensprache. [40]

tanz und die Möglichkeit zur Entwicklung eines Quasi-Standards zu erhöhen. Die FIPA-Plattform bietet zudem eine Abstraktionsebene (Bild 2.8) von der die konkreten Implementierungen abgeleitet bzw. die Schnittstellen und Softwarearchitektur übernommen werden. Darüber hinaus bietet die FIPA-Spezifikation geeignete Schemata zur einheitlichen Auslegung der Agenten und ihres Kommunikationsverhaltens. Zudem abstrahiert der Standard von der zugrundeliegenden Systemarchitektur, damit sich Entwickler nicht mit den technischen Details, wie der eingesetzten Hardware oder dem Betriebssystem, auseinandersetzen müssen. [82]

Als Nachteil erweist sich hingegen die konzeptionelle Auslegung des Kommunikationsprotokolls, das nicht verpflichtet ist, Antworten auf Anfragen oder Nachrichten zurückzusenden. Somit ist nicht erkennbar, ob der Kommunikationsteilnehmer noch nicht geantwortet hat, überhaupt nicht antworten wird oder sogar ausgefallen ist. Der Einsatz von Timeouts ist keine Lösung bei langen Berechnungszeiten und führt zu ständigen Abbrüchen der Verarbeitung. Die FIPA-Spezifikationen legen lediglich die grobe Struktur der Plattform und die Schnittstellen zwischen den Teilnehmern fest, machen jedoch keine ausreichenden Vorgaben für einen standardisierten internen Aufbau der Agentenplattformen. Somit besteht keine einheitliche Struktur zwischen den verschiedenen Implementierungen der FIPA-Agentenplattformen. Darüber hinaus werden in bestehenden Implementierungen unterschiedliche Programmiersprachen eingesetzt, die unterschiedliche Betriebssysteme unterstützen und selbst bei gleicher zugrundeliegender Sprache keine Programmcodeportierbarkeit gewährleisten. Neben der Umsetzung weist auch das zugrundeliegende Beschreibungsmodell *Belief, Desire, Intention (BDI)* [83, 84] einen großen Nachteil auf, da die Agenten mit Annahmen über ihre Umwelt und anderen Teilnehmern ausgestattet werden. Somit beruht die erfolgreiche Kooperation mit anderen Agenten auf impliziten Annahmen über das Verhalten anderer Agenten. Dieser Ansatz der impliziten Annahmen sollte durch explizite, sogenannte Policies, ersetzt werden, die angeben, welche Verpflichtungen oder Bedingungen ein Agent erfüllen muss. Dieser Nachteil wurde im ersten Schritt durch eine Spezifikation zur Erstellung von Use Cases für Policies kompensiert. [85, 82]

### 2.5.4 Wandelbare Produktion

Die hohe Flexibilität der Agentensysteme kann insbesondere in wandelbaren Produktionsumgebungen [4] zur Geltung kommen. Gemäß der Idee des Plug&Produce-Ansatzes [7] können aus einem Baukasten alle erforderlichen Anlagenkomponenten entnommen und mit diesen die Anlage aufgebaut bzw. erweitert werden. Unter dem Hot-Plug&Produce-Ansatz wird die Veränderung, d.h. sowohl die Erweiterung als auch Reduzierung, einer Produktionsanlage während der Laufzeit verstanden, ohne einen notwendigen Stillstand der Anlage oder Teile hiervon im Kauf nehmen zu müssen [86, 87].

In der Organisation des Produktionssystems mittels Agentenansatz können die Vorzüge der autonom agierenden Agenten, die zur Laufzeit ad hoc Entscheidung treffen, effektiv genutzt werden. In diesem hohen, wandelbaren Umfeld kann der Agent somit das Scheduling der Aufgaben gemäß der aktuellen Anlagensituation, d.h. den aktuell verfügbaren Ressourcen und ihrem Auslastungsgrad, optimieren. Jedoch fehlen hierzu bislang ausgereifte Konzepte und Technologien, die flexible Produktionssysteme und wandelbare Produktionsanlagen umfassend in der Steuerungstechnik in Kombination mit einem Agentensystem einbeziehen.

Im wandelbaren Produktionssystem müssen gleichermaßen zur Automatisierung auch der Mensch bzw. die manuellen Eingriffsmöglichkeiten einbezogen werden. Dies betrifft sowohl die Produktion, als auch die Inbetriebnahme und Umbaumaßnahmen. Motiviert aus den hohen Aufwänden zur Inbetriebnahme der Steuerungssysteme und den hohen Kosten für Leitsoftwaresysteme, die je Installation 100.000 Euro und mehr betragen können [88], besteht in diesem Bereich erhebliches Einsparungspotential. Um den Menschen in der Planung einerseits und Entscheidungsfindung zur Laufzeit andererseits maßgeblich zu entlasten, muss das System möglichst selbständig auf Störungen und Veränderungen reagieren können. Voraussetzung hierfür ist die automatisierte Ableitung der aktuellen Anlagenstruktur und die automatisierte Erkennung von Störungen. Das wandelbare Produktionssystem soll sich selbständig durch die Ableitung neuer Abläufe und Flüsse im Falle von Veränderungen anpassen. Dies beinhaltet u.a. die flexible Erweiterung der Produktion um neue Anlagenteile oder die Reduktion dieser bei Auslaufen eines Produktes. Eine weitere Form von Störungen sind geplante Abschaltungen von Anlagenteilen, z.B. zu Wartungszwecken. Für den Menschen stellen Veränderungen im Produktionsablauf, ausgelöst durch unerwartete oder selbst geplante Störungen, auf Grund der hohen Anlagenkomplexität eine zunehmend große Herausforderung dar, einen optimierten Anlagenbetrieb zu gewährleisten. Dies liegt insbesondere darin begründet, dass die zahlreichen Anlagenteilezustände für den Menschen nicht mehr zu überblicken sind. Zur Bewältigung der Problematik sind verschiedene Mechanismen erforderlich, die den Menschen bei der Inbetriebnahme der Anlage, bei Umbaumaßnahmen und auch während des Betriebs maßgeblich unterstützen. Durch die Reduktion der manuellen Konfiguration und des manuellen Eingriffs können mit diesen Automatismen auch menschliche Fehler vermieden werden.

Im Bereich der Softwaretechnik bietet die Agententechnologie die weitestgehende Unterstützung für dezentrale Systemarchitekturen. Bisherige Forschungsarbeiten konnten die Potentiale der Agententechnologie nicht vollständig ausschöpfen [89]. Diese beschäftigten sich überwiegend mit der Entwicklung von Konzepten, der Durchführung von Machbarkeitsstudien und der Umsetzung von Softwareprototypen für Planungs- und Simulationssysteme. Bisherige Arbeiten haben jedoch den Aspekt der Wandlungsfähigkeit [4] im Agentenansatz nur eingeschränkt betrachtet. Zur Gewährleistung einer

hohen Wandlungsfähigkeit ist es erforderlich, den Baukastenansatz nicht nur auf die Mechanik, sondern insbesondere auf die Steuerungstechnik, inkl. der Steuerungssoftware anzuwenden. Hierzu müssen vordefinierte, modular abgeschlossene Teilsysteme, bestehend aus einer mechanischen, elektro- und steuerungstechnischen Komponente mit zugehöriger Steuerungssoftware als Baustein in einem Baukasten zur Verfügung gestellt werden. Hierunter wird die Umsetzung einer mechatronischen Komponente bestehend aus den Fachbereichen Mechanik, Elektrotechnik und Steuerungstechnik verstanden. Jedoch wird in vielfachen Umsetzungen mechatronischer Komponenten die Steuerungstechnik vernachlässigt. Zur Gewährleistung einer durchgängigen Modularisierung ist die Differenzierung der Steuerungssoftware in abgeschlossene Funktionseinheiten mit bevorzugter Integration der Steuerungsfunktionen am Gerät erforderlich. Somit kann der einfache modulare und insbesondere aufwandsarme Aufbau nicht nur auf Seiten der Mechanik und Elektronik, sondern auch auf Seiten der Steuerungstechnik gewährleistet werden. Um einen schnellen Umbau und somit eine schnelle Anpassung der Anlage an neue Produkte bzw. an neue Anforderungen gewährleisten zu können, ist es erforderlich, dass die Komponenten sowohl auf physischer als auch steuerungstechnischer Seite über standardisierte Schnittstellen verfügen und sich automatisch in das Gesamtsystem eingliedern. Dabei müssen die Module nicht nur standardisierte Schnittstellen zur Steuerung der Geräte implementieren, sondern als Teilnehmer bzw. Agenten mit standardisiertem Verhalten, die nach außen über eine normierte Kommunikationssprache mit anderen interagieren. Das Ziel in der Anwendung besteht darin, die Bausteine aus dem Baukasten zusammenzustecken, einzuschalten und eine unmittelbar einsatzbereite Anlage zur Verfügung zu haben, die sich selbständig beim Start konfiguriert.

Mit dem Agentenmodell wird die dynamische Produktionsorganisation zwischen verschiedenen Teilnehmern gewährleistet, die sich ad hoc zur Laufzeit finden und die unmittelbare Neuorganisation der Abläufe im verteilten System bei Veränderungen bewerkstelligen müssen. Bisherige Arbeiten wurden nur an starren Produktionssystemen implementiert und umgesetzt. In der vorliegenden Arbeit werden neue Ansätze als Grundlage zur Realisierung hoch-flexibler Anlagensysteme vorgestellt, um derzeitige wirtschaftliche Probleme flexibler Produktionssysteme, verursacht durch die hohen Umbauaufwände, zu bewältigen (vgl. Kapitel 2.1).

## Kapitel 3

# Agentenbasiertes Systemmodell für wandelbare Produktionsanlagen

Um die vorgestellten Anforderungen durchgängig modularer Anlagensysteme auch auf Seiten der Steuerungssysteme zu erfüllen (vgl. Kapitel 2.5.4) und dabei die flexiblen und dynamischen Abläufe im Materialfluss zur Laufzeit steuern und regeln zu können, wird in diesem Kapitel zunächst ein Systemmodell für wandelbare Produktionsanlagen vorgestellt. Im Modell werden alle Komponenten, die maßgeblich am Produktionsablauf aus Sicht des Materialflusses beteiligt sind, erfasst und gemäß dem Prinzip der Abstraktion standardisiert. Sie bilden die Grundlage für den modularen Aufbau der Steuerungssoftware, bestehend aus funktionell abgeschlossenen Einheiten mit eigenen, autarken Steuer- und Regelkreisen. Die Kapselung der Anlagenteile in abgeschlossene, eigenständige Komponenten bietet den Vorteil von der dahinterliegenden Ausprägung, d.h. vom konkreten Transportsystem oder der Type der Bearbeitungsstation, zu abstrahieren. Bezogen auf die Funktion, können diesen standardisierte Softwareschnittstellen mit einheitlich vordefiniertem Verhalten zugewiesen werden. Somit können alle gleichberechtigten Teilnehmer im Anlagenverbund direkt miteinander ad hoc Verbindungen aufbauen, da die Schnittstellen gemäß ihrer zugrundeliegenden Type, wie Transportsystem, Bearbeitungsstation, Werkstück usw. normiert sind.

Basierend auf dieser Formalisierung wird der Agentenansatz eingeführt (vgl. Kapitel 2.4). Den am operativen Entscheidungsprozess mitwirkenden aktiven oder passiven Komponenten werden Agenten als sogenannte Stellvertreter zugeordnet, die selbstständig zur Laufzeit Informationen sammeln und mit diesen aktuellen Daten Entscheidungen treffen. Das resultierende System bietet ein sehr hohes Maß an Anpassungsfähigkeit bzw. Wandelbarkeit für Anwendungsbereiche mit hoch flexiblen Anforderungen [90].

### 3.1 Identifikation der Komponenten einer Produktionsanlage aus Sicht des Materialflusses

Zur modellhaften Erfassung von Produktionsanlagen werden im Folgenden alle am Materialfluss beteiligten Komponenten erfasst und ihre Aufgaben bzw. Funktionen beschrieben. Hierbei wird nicht der Anspruch gestellt einen vollständigen Baukasten zu entwerfen, der alle derzeit erhältlichen Maschinenmodelle beinhaltet. Es geht vielmehr darum, die wesentlichen Komponenten, deren Funktionsweise und Wirkprinzip sich auf die Mehrzahl bestehender Anwendungsfälle übertragen lassen, in ihrem Funktionsumfang zu identifizieren.

In der Produktion werden im Folgenden vier verschiedene Ressourcengruppen unterschieden, die für die Sicherstellung der Produktion wesentlich sind (Bild 3.1). Zur Veränderung bzw. Bearbeitung von Bauteilen werden sogenannte *Bearbeitungsstationen* vorgesehen, die sich in den Materialfluss ein-

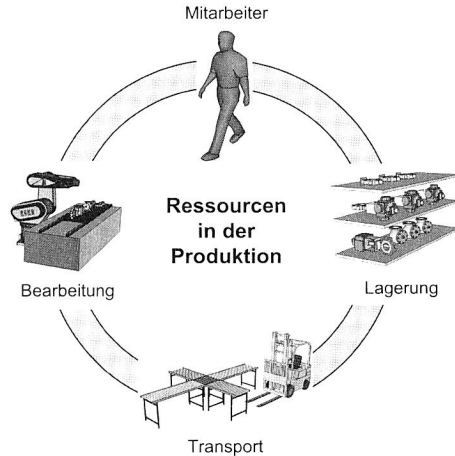


Bild 3.1: Unterteilung der Ressourcen in Gruppen aus Sicht des Materialflusses zur funktionalen Kategorisierung der Teilnehmer. Den Teilnehmern dieser Gruppen kann aus organisatorischen Aspekten einheitliches Verhalten für die Kooperation mit anderen Teilnehmern zugewiesen werden, unabhängig von ihrer konkreten Ausprägung.

gliedern und die Anlaufstellen für Werkstücke darstellen. Zur Gewährleistung einer effizienten Auslastung der Bearbeitungsstationen, müssen alle Bauteile, die zur Durchführung eines Schrittes erforderlich sind, direkt an den Bearbeitungsstationen termingerecht vorrätig sein. Hierfür sind prinzipiell *Läger* vonnöten, die Bauteile entweder selektiv oder nach einer vorgegebenen Reihenfolge, z.B. FIFO, ein- und auslagern können [91]. Zu diesen Lägern zählen sowohl Wareneingangs-, Warenausgangs- und Zwischenlager innerhalb der Produktion. Zudem werden auch die Eingangspuffer an den Bearbeitungsstationen als Läger angesehen, da sie im Allgemeinen die gleiche Aufgabe, eben die zeitliche Aufbewahrung von Gütern, übernehmen.

Die Überführung der Bauteile zwischen den Bearbeitungsstationen und Lägern übernimmt ein *Transportsystem*. Hiervon gibt es verschiedenste Systeme, die in spurgebundene und spurungebundene einerseits und aktive und passive andererseits unterteilt werden. Die genauen Definitionen und die Beschreibung dieser Systeme werden in Kapitel 3.4 näher erläutert.

Der *Mitarbeiter* steht bei allen Ressourcen immer an oberster Stelle, da er in allen Bereichen der Produktion flexibel eingesetzt werden kann - entweder zur aktiven Durchführung einer Aufgabe oder als Entscheidungsträger. Bei den hier betrachteten Bearbeitungsstationen, Transportsystemen oder Lagersystemen werden sowohl vollautomatisierte Anlagen, als auch manuelle oder hybride Systeme betrachtet. Von der Differenzierung manuell oder automatisiert wird im Folgenden abstrahiert, da die Systeme unabhängig von ihrer Ausprägung die gleiche Funktionalität zur Verfügung stellen und die gleichen Bemessungsgrößen, wie Durchsatz, Zuverlässigkeit, Wartung, Betriebskosten, Investitionskosten usw. aufweisen.



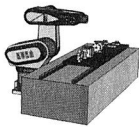
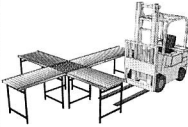
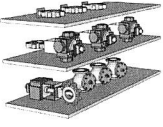

Materialflussteilnehmer			
Bearbeitungsstationen	Transportsysteme	Läger	Mitarbeiter
			
Maschinelle oder manuelle Bearbeitung von Werkstücken	Bewegung von Gütern in der Produktion	Zwischenlagerung der Werkstücke bis zur weiteren Verarbeitung oder Transport – langfristig oder kurzfristig als Puffer	Mitarbeiter als Entscheidungsträger zur Werkstückbearbeitung, Handhabung, Transport und Organisation der Abläufe
Modifikation oder Zusammenführen von Bauteilen mit Qualitätsanforderungen und -kontrollen	Termingerechte Lieferung muss sichergestellt werden	Optimaler Zugriff und schnelle Kommissionierung	Bietet höchste Flexibilität und kann bedingt durch seine Fähigkeiten in unterschiedlichen Bereichen eingesetzt werden
Ziel: Optimierte Auslastung			

Bild 3.2: Charakterisierung der Ressourcengruppen innerhalb des Materialflusses.

In Bild 3.2 sind alle Ressourcengruppen mit ihren Aufgaben und Zielen nochmals zusammengefasst. Das gemeinsame Ziel jeder Ressource ist die optimale Auslastung, d.h. ihre effektive Nutzung zur Amortisierung der Investitionskosten.

In Bild 3.3 sind die in dieser Arbeit verwendeten Symbole für die einzelnen Komponenten aufgelistet. Bei den Transportsystemen wird von der Ausprägung, d.h. Stetigfördersystemen (z.B. reibschlussbasiert) oder fahrzeuggestützten Unstetigfördersystemen, abstrahiert. Die Übergabebahnhöfe stellen die Schnittstelle zwischen dem Transportsystem einerseits und der Bearbeitungsstation andererseits dar. Prinzipiell besitzen sie keine Pufferfunktion. Bei Stetigfördersystemen ist i.d.R. kein Puffer erforderlich, da diese unmittelbar Werkstücke aufnehmen können bzw. eine integrierte Pufferfunktion besitzen. Im Falle von fahrzeuggestützten Systemen werden die Bahnhöfe mit einem Lager kombiniert. Dabei verbinden die Bahnhöfe sowohl physisch als auch logistisch die Bearbeitungsstation mit dem Transportsystem, lasten Transportaufträge ein, koordinieren die Werkstückübergabe und stellen auf softwaretechnischer Seite standardisierte Schnittstellen zur Verfügung, die unabhängig von der Art des Transportsystems immer einheitlich sind.

Die Werkstücke können, müssen aber nicht, auf Werkstückträgern transportiert werden. Die Werkstücke bzw. die Werkstückträger sind eindeutig identifizierbar und können darüber hinaus dem zugewiesenen Fertigungsauftrag zugeordnet werden. Die Identifikation ist zudem Grundlage zur Bestimmung der Produktvariante und des Bearbeitungszustands, um daraus die noch erforderlichen Bearbei-

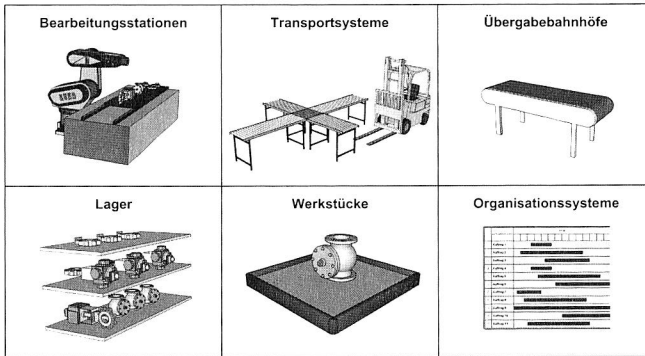


Bild 3.3: Identifizierung der funktional abgeschlossenen Komponenten in der Produktion auf einer anwendungs-unabhängigen Abstraktionsebene.

tungsschritte abzuleiten und an jeder Station die erforderlichen Prozessparameter vorzugeben.

Zu den Organisationssystemen zählen alle Ablaufsteuerungssysteme, die für die Planung und Abwicklung der Produktion erforderlich sind. In erster Linie zählt hierzu die Auftragsseinlastung, die gemäß der aktuellen Auslastung und den Optimierungsgrößen, wie die kostengünstige und termingerechte Herstellung, die Aufträge einlasten muss.

### 3.2 Modellierung von Produktionsanlagen mittels standardisierter Komponenten

Im Folgenden werden Verknüpfungsvorschriften vorgestellt, wie die in Kapitel 3.1 identifizierten Komponenten bzw. Ressourcengruppen zur Modellierung von Produktionsanlagen miteinander verbunden werden können. Hieraus werden Regeln abgeleitet, die eine vollständige strukturelle und funktionale Erfassung der Produktionsanlage mit ihren Bestandteilen ermöglicht:

**Regel 1:** Es gibt nur Eingangsläger, jedoch keine Ausgangsläger.

**Regel 2:** Für jede Übergaberichtung befindet sich ein Übergabebahnhof zwischen dem Transportsystem und einer Bearbeitungsstation.

**Regel 3:** Zwei Bearbeitungsstationen sind durch ein Transportsystem miteinander verbunden.

**Regel 4:** Ein Eingangslager existiert immer in Kombination mit einem Übergabebahnhof.

**Regel 5:** Eine Bearbeitungsstation verfügt immer über mindestens ein Push-Eingangslager für den Werkstückfluss und kann beliebig viele Pull-Eingangslager für den Bauteilfluss haben.

**Regel 6:** Transportsystemen kann an den Übergabebahnhöfen zur Werkstückannahme ein Push-Eingangslager zugeordnet werden.

**Regel 7:** Zwei Transportsysteme werden für jede Übergaberichtung über je einen Übergabebahnhof mit oder ohne Push-Eingangslager gekoppelt.

**Regel 8:** Alleinstehende Push-Läger werden über mindestens einen Übergabebahnhof an das bzw. die Transportsysteme gekoppelt.

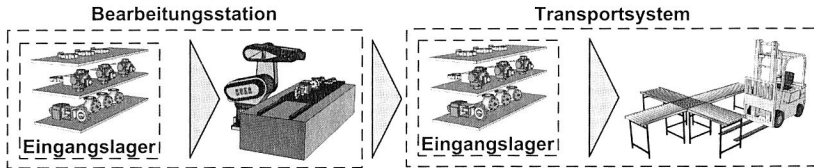


Bild 3.4: Verkettung der Bearbeitungsstationen und Transportsysteme. Das funktional zugeordnete Eingangslager der nachfolgenden Station ist zugleich Ausgangslager der vorhergehenden.

Basierend auf diesen Regeln lassen sich die in Kapitel 3.1 beschriebenen funktionalen Standardkomponenten mit einheitlichem und vordefiniertem Verhalten definieren, die in ein umfassendes Systemmodell einfließen. Diese Standardkomponenten, die quasi eine Klassifizierung der Anlagenkomponenten gemäß ihrer Funktion vornehmen, lassen sich auf konkrete Anlagenkomponenten abbilden. Somit können die geräte- und anlagentypspezifischen Charakteristika der Anlagenkomponenten für die Organisation des Materialflusses und der Produktion ausgeblendet werden. Dies unterstützt somit den hohen Grad an Wiederverwendbarkeit bestehender Komponenten, die sich vereinfacht in eine bestehende Anlage oder andere Komponenten integrieren lassen. Zudem werden auch die Anforderungen an eine schnelle Inbetriebnahme und vereinfachte Umbaumaßnahmen gewährleistet.

### Regel 1: Es gibt nur Eingangsläger, jedoch keine Ausgangsläger.

Im vorliegenden Systemmodell wird angenommen, dass jede Bearbeitungsstation und jedes Transportsystem an der Eingangsschnittstelle über einen Puffer verfügt, in dem Werkstücke und Bauteile eingelagert werden und diese, bis zur weiteren Bearbeitung bzw. zum Transport, dort verweilen können. Da es sich im Allgemeinen bei Puffern auch um Läger handelt, werden im vorliegenden Modell die Pufferstellen mittels Lägern modelliert, da beide die gleiche Funktion erfüllen und mit den gleichen Verhaltensbausteinen aufgebaut werden können. Diese Läger befinden sich immer nur am Eingang einer Bearbeitungsstation oder eines Transportsystems und stellen zugleich das Ausgangslager der vorliegenden Einheit dar. Folglich ist die Modellierung von ausgezeichneten Ausgangslägern nicht erforderlich und daher wird an dieser Stelle immer von Eingangslägern gesprochen. Ermöglicht wird dies, indem das nachfolgende Eingangslager physisch direkt an die vorangegangene Einheit gekoppelt ist (vgl. Bild 3.4). Organisatorisch ist das Eingangslager immer der nachfolgenden Ressource, d.h. Bearbeitungsstation oder Transportsystem, zugeordnet, die selbst die Entnahme der Bauteile aus dem Eingangslager bestimmt und veranlasst.

Zudem können die Läger zu unterschiedlichen Zwecken, wie Werkstückfluss oder Bauteilvorratshaltung an den Bearbeitungsstationen eingesetzt werden (siehe Kapitel 3.3). Die einzige Ausnahme in der keine Eingangspuffer vorhanden sind, stellen taktgetriebene Anlagensysteme, wie Rundtaktautomaten oder getaktete Produktionslinien, dar. In diesem Fall sollte von einer Bearbeitungsstation ausgegangen werden, in der jeder Rundtaktautomat oder jede getaktete Linie eine Bearbeitungsstation darstellt. Die einzelnen Fertigungs- und Montagemodule darin werden als Prozessmodule angesehen, die in vielen Fällen in Echtzeit koordiniert und aufeinander abgestimmt werden müssen.

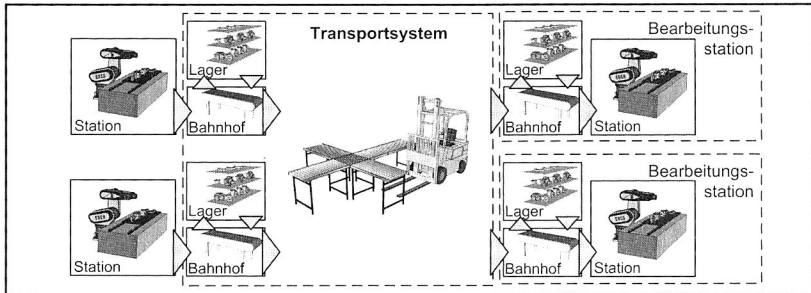


Bild 3.5: Sowohl Reibschlussfördersysteme als auch fahrzeugbasierte Transportsysteme verfügen über mindestens ein Eingangslager.

### Regel 2: Für jede Übergaberichtung befindet sich ein Übergabebahnhof zwischen dem Transportsystem und einer Bearbeitungsstation.

An den Schnittstellen zum Transportsystem können sich physische Bahnhöfe befinden - unbedingt erforderlich bei Fahrerlosen Transportfahrzeugen [92] - oder sie existieren lediglich virtuell als Softwarekomponenten - möglich bei Reibschlussfördersystemen, um in beiden Fällen Transportaufträge einzulasten und die Werkstückübergabe zu koordinieren. Für jede Richtung, d.h. Eingang oder Ausgang, existiert ein eigener Übergabebahnhof. Optional könnten mit einem Bahnhof auch bidirektionale Verbindungen modelliert werden, was jedoch in Produktionsumgebungen selten vorzufinden, aufwendig zu koordinieren und zu steuern ist.

Da ein Transportsystem verschiedene Bearbeitungsstationen miteinander verbindet, kann dieses folglich über mehrere Eingangsläger verfügen. Für jede Bearbeitungsstation, die Werkstücke an das Transportsystem abgeben muss, existiert auf Seiten des Transportsystems in der Regel ein Eingangslager, das physisch direkt an die Bearbeitungsstation gekoppelt ist und aus Sicht der Bearbeitungsstation das Ausgangslager darstellt (siehe Bild 3.5).

Im Allgemeinen verbergen die Bahnhöfe die Ausprägung des Transportsystems, u.a. aktive fahrzeugbasierte oder passive spurgeführte Systeme, und bieten nach außen einheitliche Schnittstellen mit einheitlichem Verhalten an. Damit wird ein Beitrag zum modularen Aufbau der Anlagensysteme geleistet. Dies ist begründet in der Begrenzung der Sichten außenstehender Einheiten, wie die der Bearbeitungsstationen oder Läger, die keinen Einblick in den Transport benötigen, mit welchen Transportmitteln dieser ausgeführt wird und woher die Werkstücke oder Bauteile bezogen werden.

### Regel 3: Zwei Bearbeitungsstationen sind durch ein Transportsystem miteinander verbunden.

Die Verkettung der Bearbeitungsstationen und der Transportsysteme erfolgt i.d.R. immer abwechselnd (Bild 3.6). Es können zwar zwei aufeinanderfolgende Transportsysteme angebunden werden, z.B. der Übergang von einem Reibschlussfördersystem auf ein fahrzeugbasiertes Transportsystem, jedoch dürfen keine zwei Bearbeitungsstationen direkt ohne ein Transportsystem dazwischen verbun-

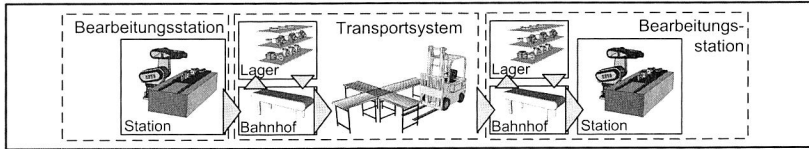


Bild 3.6: Verketten von Bearbeitungsstationen und Transportsystemen nur in abwechselnder Reihenfolge erlaubt.

den werden. Der Grund dafür ist, dass die vorgesehenen Eingangsläger keinen räumlichen Transport von einer Bearbeitungsstation zur nächsten vorsehen.

#### **Regel 4: Ein Eingangslager existiert immer in Kombination mit einem Übergabebahnhof.**

Da die Eingangsläger immer Pufferplätze für die kurzzeitige Zwischenlagerung von Werkstücken oder Bauteilen zur Verfügung stellen, treten sie immer in Verbindung mit einem Transportsystem und folglich mit einem Übergabebahnhof auf. Der Eingangs-Übergabebahnhof, der die Schnittstelle zwischen der Station und dem Transportsystem oder zwischen zwei Transportsystemen darstellt, hat die Aufgabe, Werkstücke und Bauteile entgegen zu nehmen und diese kurzzeitig im angebundenen Eingangslager zwischenzulagern.

### 3.3 Differenzierung verschiedener Lagertypen

Die Unterteilung der Lagertypen erfolgt nicht nach ihrer mechanischen Ausprägung, ihrer Größe oder der Dauer der Zwischenlagerung, sondern nach ihrem Funktions- und Aufgabenbereich im logistischen Verbund des Materialflusses. Dabei genügt es zwischen Lagertypen zu unterscheiden, die Werkstücke aus dem Werkstückfluss im Produktionssystem aufnehmen und jenen, die Bauteile zur Montage an der Bearbeitungsstation vorrätig halten.

Die zwei relevanten Typen sind Push- und Pull-Läger (siehe Bild 3.7). Im einen Fall werden die Güter von außen zugeführt und das Lager muss sich nur um die Ein- und Auslagerung kümmern (Push). Im anderen Fall übernimmt das Lager eigenständig die Aufgabe, einen Mindestbestand bestimmter Artikel sicherzustellen (Pull). Fehlen diese, generiert das Pull-Lager entweder an den Wareneingang oder eine andere Produktionsanlage eine Order. Ein dazwischengeschaltetes Transportsystem wird beauftragt die Bauteile zuzustellen. Dieser Lagertyp wird nur bei Bearbeitungsstationen eingesetzt, die Bauteile zur Montage benötigen und letztere zur Bearbeitung einer bestimmten Produktvariante vorrätig sein müssen.

#### **Regel 5: Eine Bearbeitungsstation verfügt immer über mindestens ein Push-Eingangslager für den Werkstückfluss und kann beliebig viele Pull-Eingangsläger für den Bauteilfluss haben.**

Bearbeitungsstationen verfügen über mindestens ein Eingangslager. Das sogenannte Push-Eingangslager ist immer dem Werkstückfluss zugeordnet, in dem die zu bearbeitenden Werkstücke ankommen

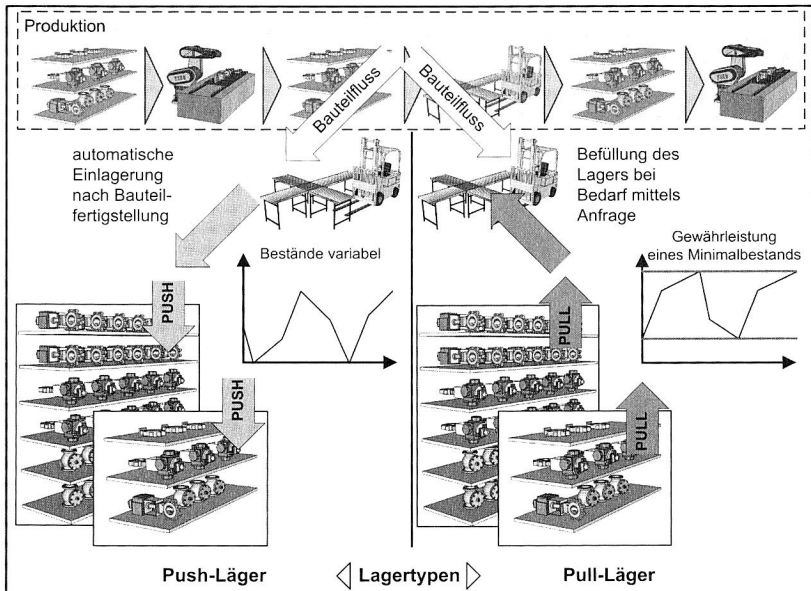


Bild 3.7: Differenzierung zwischen den passiven Push- und aktiven Pull-Lagertypen.

und zwischengespeichert werden. Die Pull-Läger sind optional an eine Bearbeitungsstation anbindbar und für den Bauteilfluss vorgesehen (Bild 3.8). Sie halten die zu verbauenden Bauteile auf Vorrat und sorgen selbständig dafür, dass ein Mindestbestand vorrätig ist.

**Regel 6: Transportsystemen kann an den Übergabebahnhöfen zur Werkstückannahme ein Push-Eingangslager zugeordnet werden.**

Analog zur Bearbeitungsstation kann ebenso das Transportsystem über Push-Eingangslager verfügen. Diese sind insbesondere bei Unstetigfördersystemen erforderlich, da die Werkstücke bis zur Ankunft der Fördereinheiten kurzzeitig zwischengelagert werden müssen. Analog zur Bearbeitungsstation koordiniert das Transportsystem die Entnahme von Werkstücken.

**Regel 7: Zwei Transportsysteme werden für jede Übergaberichtung über je einen Übergabebahnhof mit oder ohne Push-Eingangslager gekoppelt.**

Analog zur Kopplung von Bearbeitungsstationen und Transportsystemen, muss ebenso zwischen zwei Transportsystemen je ein Übergabebahnhof zu jeder Seite integriert werden (Bild 3.9). Bei einem unidirektionalen Fluss übernimmt der eine Bahnhof den Ausgang und der direkt dahinter gekoppelte Bahnhof den Eingang. Im Falle zweier uni-direktionaler Kopplungen sind insgesamt vier Bahnhöfe erforderlich, die jeweils mit dem Bahnhof der gegensätzlichen Funktion gekoppelt sind. Dabei kann

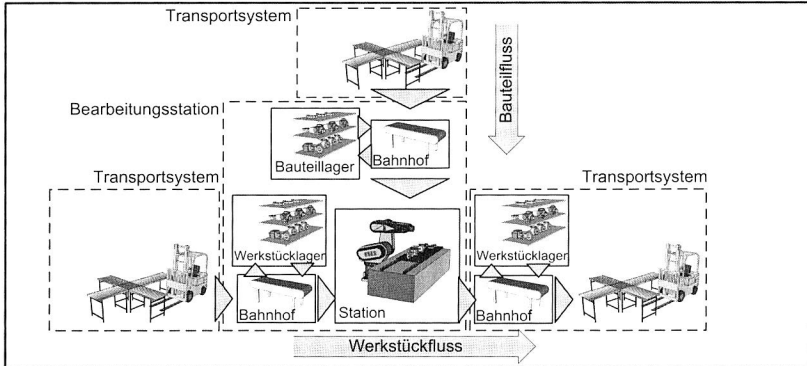


Bild 3.8: Die Zuführung von Bauteilen für die Montage an eine Bearbeitungsstation erfolgt über weitere Läger.

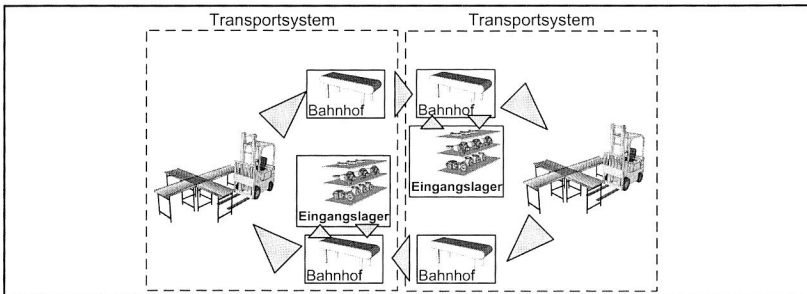


Bild 3.9: Die Kopplung zweier Transportsysteme erfolgt analog mit den Übergabebahnhöfen, wobei die Bahnhöfe auf jeder Seite die Schnittstelle zum zugehörigen Transportsystem bilden.

dem Eingansbahnhof ein Einganslager zugeordnet werden, das die Werkstücke und Bauteile für das nachfolgende Transportsystem zwischenlagern kann.

### Regel 8: Alleinstehende Push-Läger werden über mindestens einen Übergabebahnhof an das bzw. die Transportsysteme gekoppelt.

Bisher wurden Läger nur in Verbindung mit einem Transportsystem oder einer Bearbeitungsstation betrachtet. Es gibt aber auch alleinstehende Läger, die für die langfristige Aufbewahrung von Bauteilen oder Werkstücken vorgesehen sind oder zusätzliche Puffer darstellen, um Schwankungen ausgleichen zu können. Diese können wie die Bearbeitungsstationen an ein vor- und nachgelagertes Transportsystem gekoppelt werden (siehe Bild 3.10). Im Falle einer Produktionsanlage befindet sich ein Lager immer am Ende dieser, das die fertiggestellten Produkte aufnimmt und für den weiteren Transport z.B. an eine weiterführende Anlage oder für den Versand zwischenlagert.

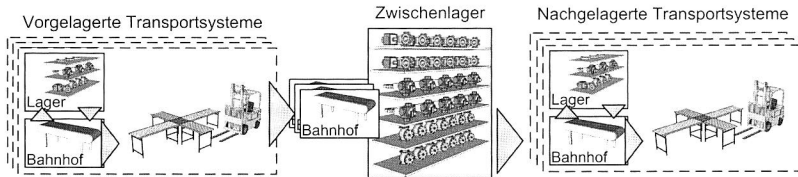


Bild 3.10: Zwischenlager dienen zur längerfristigen Aufbewahrung oder Aufnahme zahlreicher Güter.

### 3.4 Klassifikation von Transfersystemen

In der Produktion sind eine Vielzahl von Transfertechniken etabliert, die sich in der Art und im Automatisierungsgrad unterscheiden. Reibschlussfördersysteme werden in Form von gurt-, ketten- oder rollengetriebenen Fördereinrichtungen realisiert. Manuell geführte Fahrzeuge existieren in verschiedenen Größen und für verschiedene Anwendungen, wie Hubwagen, Ameisen, Gabelstapler, Wagen, usw. Aus technologischer Sicht können diese Prinzipien in die beiden Hauptkategorien *spurgebundene* und *spurungebundene* Transfersysteme eingeteilt werden [93].

*Spurgebundene Systeme* zeichnen sich durch ein Transfernetzsystem aus, das aus Transfersegmenten und Verzweigungen besteht. Sie sind aufgebaut vergleichbar mit einem Graph mit Verzweigungen und Arbeitsstationen als Knoten sowie Transfersegmenten als Kanten. Neben den Reibschlussfördersystemen zählen zu dieser Kategorie alle Formen schienen- bzw. leitliniengebundener Fahrzeuge.

*Spurungebundene Systeme* bestehen aus mobilen Einheiten, die sich frei in einer Umgebung bewegen können. Mögliche vorgegebene Wege haben hierbei eine, bezogen auf die Längsachse, horizontale Ausdehnung, in der die Einheiten autonom über ihre Bahn unter Zuhilfenahme von Umgebungsinformationen entscheiden können. Zu den spurungebundenen Systemen in der Produktion zählen sowohl der Mensch als Transporteur selbst, als auch in der Regel alle von Personen geführten Fahrzeuge.

Aus steuerungstechnischer Sicht sind aktive und passive Transfersysteme unterscheidbar [93]. Aktive Systeme verfügen über Rechen- und Kommunikationstechnik und sind in die logistischen Entscheidungsprozesse involviert. Passive Systeme verfügen dagegen weder über Aktorik und Sensorik, noch über Steuerungstechnik und sind nicht an den Entscheidungsprozessen beteiligt.

Spurungebundene Systeme sind immer aktive Systeme, da sie über keinerlei Wegeführung verfügen, die aktiv von außen die Richtung vorgibt. Spurgebundene Systeme können sowohl als aktive als auch als passive Systeme realisiert sein. Im Falle aktiver Systeme übernehmen die Transporteinheiten vollständig oder teilweise die Richtungsentscheidung, die jedoch von externen Einheiten, wie den Kreuzungen ausgeführt werden. Passive Transporteinheiten werden von einem aktiven Wegenetz zum Ziel geführt und optimale Routenentscheidungen extern, ohne ihr Mitwirken, getroffen. Da in der industriellen Anwendung i.d.R. die aktiven den spurungebundenen und die passiven Systeme den spurgebundenen zuzuordnen sind, wird im Folgenden *passiv* mit spurgebunden und *aktiv* mit spurungebunden gleichgesetzt.



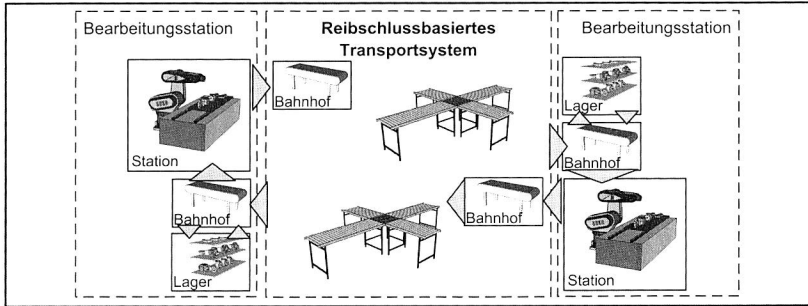


Bild 3.11: Transportsystem, das auf Reibschlussverfahren basiert, verbindet mehrere Bearbeitungsstationen über die Bahnhöfe. An den Ausgangsbahnhöfen sind keine Eingangsläger erforderlich, da das Transportsystem selbst über Pufferkapazitäten verfügt.

Im Falle der aktiven, spurungebundenen Transportsysteme ist jede Transporteinheit mit einem eigenen Agenten ausgestattet, der die Aufgaben koordiniert und mit den anderen Agenten in Verbindung tritt. Im Falle der passiven, spurgebundenen Transportsysteme kommt das Agentenverhalten nicht in der Transporteinheit zum Einsatz, sondern ist extern realisiert, z.B. in den intelligenten Kreuzungen, die autonom Routenentscheidungen treffen (Kapitel 4.3).

### 3.5 Modellierung von Reibschlussfördersystemen

Das Fördern mittels Reibschluss ist in der Produktion ein etabliertes Verfahren, das mittels unterschiedlicher Technologien, wie Gurt-, Ketten- oder Rollenantrieben, umgesetzt werden kann. Es zählt zu den spurgebundenen Stetigfördersystemen, die durch Transfernetze, bestehend aus Transfersegmenten und Weichen charakterisiert sind. Transfernetze entsprechen damit einem Graph, dessen *Knoten* Weichen oder Arbeitsstationen und dessen *Kanten* Transfersegmente repräsentieren.

Einem Reibschlussfördersystem können ein oder mehrere Eingangsläger zugeordnet werden, abhängig von der Zahl der angebotenen Bearbeitungsstationen, die Transportgüter zum Transport einlasten. In den meisten Realisierungen reibschlussbasierter Transportsysteme sind keine Eingangsläger erforderlich, da die Transfersegmente selbst u.a. die Funktion von Puffern übernehmen (Bild 3.11). In diesem Fall ist zumindest der Einsatz von Übergabebahnhöfen zur allgemeinen Abwicklung der Transportauftragserteilung und Übernahme des Werkstücks durch das Transportsystem erforderlich. Die genauen Abläufe zur Werkstückübergabe und das Verhalten der Übergabebahnhöfe im Zusammenspiel mit den Lägern werden in Kapitel 6.4 erläutert.

Wie bereits beschrieben, stellen die Eingangsläger der Bearbeitungsstationen die zugehörigen Ausgangsläger dar und sind im Gegensatz zu den Eingangslägern auf Seiten des Transportsystems in jedem Falle erforderlich.

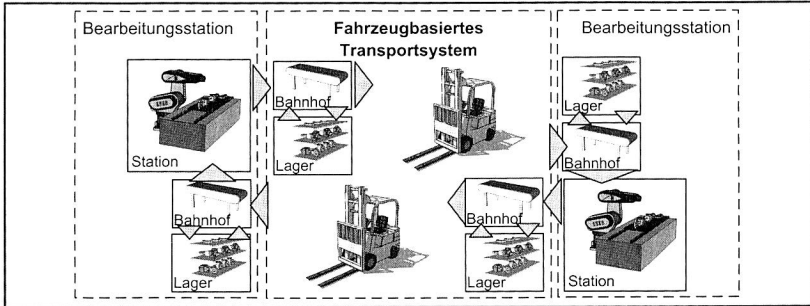


Bild 3.12: Fahrzeugbasierte Transportsysteme verbinden die Bearbeitungsstationen über die angebotenen Bahnhöfe, die auf Seiten des Transportsystems und der Bearbeitungsstationen über Eingangsläger verfügen.

### 3.6 Modellierung von fahrzeugbasierten Transportsystemen

Zu den fahrzeugbasierten Transportsystemen zählen alle Transportsysteme, die aus mobilen Einheiten bestehen und sich selbständig mit eigener Aktorik auf einer Wegeinfrastruktur bewegen können. Beispiele hierfür sind Hubwagen, Ameisen, Gabelstapler, Fahrerlose Transportsysteme usw.

Aus Sicht der Steuerung sind die fahrzeugbasierten Transportsysteme aktive Systeme, die im Falle Fahrerloser Transportsysteme über Rechen- und Kommunikationstechnik verfügen und hierüber automatisiert gesteuert werden. Dagegen besitzt im Falle manuell geführter Systeme der Mensch die Entscheidungs- und Steuerungsgewalt, aktiv den Transportprozess durchzuführen und sich ggf. am logistischen Entscheidungsprozess zu beteiligen.

Fahrzeugbasierte Transportsysteme zeichnen sich im Verbund mit der Produktionsanlage dadurch aus, dass sie neben den Eingangslägern sowohl am Ein- als auch am Ausgang physische Übergabebahnhöfe erfordern (siehe Bild 3.12). Diese stellen Übergabeeinrichtungen dar, die an das Transportmittel angepasst sind und entweder eine manuelle oder automatisierte Übergabe von Transportgütern von und zum Transportmittel erlauben. Funktional sind ihnen zunächst keinerlei Pufferfunktionen zugeordnet. Ist eine Pufferfunktion erforderlich, so wird diese in Form eines angeschlossenen Lagers realisiert. Zur Modellierung eines Übergabebahnhofs für Fahrzeuge ist am Eingang des Transportsystems ein Lager erforderlich, das sowohl physisch als auch steuerungstechnisch direkt mit dem Übergabebahnhof verbunden ist. Eine ebensolche Verbindung existiert bei den Bahnhöfen, die Ausgänge des Transportsystems zum Eingangslager der Bearbeitungsstation darstellen. Dennoch ist das Lager organisatorisch der Bearbeitungsstation zugeordnet, da diese das Auslagern von Werkstücken steuert. Im Sinne der Modularisierung handelt es sich bei dem Übergabebahnhof und dem Lager um zwei getrennte Komponenten, die über standardisierte Schnittstellen miteinander interagieren können (vgl. Kapitel 6.4).

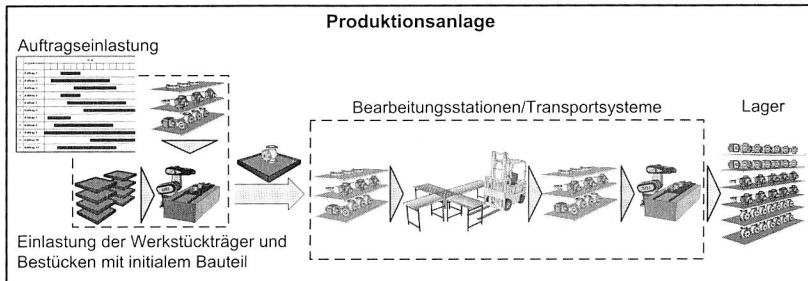


Bild 3.13: Jede Produktionsanlage verfügt über eine eigene Auftragseinlastung zur lokal optimierten Einlastung von Produktionsaufträgen.

### 3.7 Auftragseinlastung zur Organisation der Produktion innerhalb eines Produktionssystems

Jedes Produktionssystem, beispielsweise eine einzelne Fertigungslinie, die bestimmte Bauteile oder Produkte herstellen kann, verfügt über eine eigene Auftragseinlastung. Sie hat die Aufgabe, lokal, abgestimmt auf den Zustand der Anlage, die anstehenden Fertigungsaufträge einzulasten und termingerecht fertig zu stellen. Die Zahl der im System befindlichen Werkstückträger („Work-In-Progress“) ist hierbei bekannt und wird von der Auftragseinlastung gemäß dem Bedarf und der vorhandenen Kapazitäten geregelt. Hierzu ist ein direkter und ständiger Informationsfluss von den Teilkomponenten einer Anlage, wie Transportsysteme, Bearbeitungsstationen, Lager usw. zur Auftragseinlastung erforderlich.

An der Quelle der Produktionsanlage werden initiale Bauteile, optional auf Werkstückträgern, in die Anlage eingeschleust. Hierzu ist zu Beginn eine spezielle Kommissionierungsstation vorgesehen, die lediglich Handhabungsaufgaben übernimmt, um ein oder mehrere Bauteile auf dem Werkstückträger abzulegen. Jedem initialen Bauteil bzw. Werkstück wird eindeutig die zu fertigende Produktvariante mit den erforderlichen Prozessschritten zugeordnet. Ihre Aufgabe besteht darin, durch die Anlage bis zur Senke zu navigieren, um die anstehenden Bearbeitungsschritte aus dem produktspezifischen Arbeitsplan zu erledigen. Die Freigabe der Werkstücke erfolgt hierbei von der Auftragseinlastung, die zugleich die Zuordnung zum Produkt und dem Fertigungsauftrag vornehmen kann (siehe Bild 3.13). Letzterem sind wiederum Stückzahlen, Kundendaten und Fertigstellungszeitpunkt zugeordnet. Somit hält die Kommissionierungsstation ein Lager für die initialen Bauteile und für die Werkstückträger, falls diese erforderlich sind, auf Vorrat.

### 3.8 Agentenbasiertes Modell zur Organisation der Produktionsabläufe

Zur Einbindung der bisher beschriebenen Ressourcen in die Organisation der Produktion, wird der Agentenansatz [94] verfolgt. Dieser bietet sich insbesondere für den Einsatz in modularen, wandelbaren Produktionssystemen an, in denen die Vorgänge während der Produktion nur bedingt im Vorfeld planbar sind, da im Falle häufiger, kurzfristiger Umplanungen, bedingt durch Umbaumaßnahmen


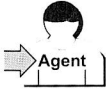

Agentenklassen		
Dienstgeber	Dienstnehmer	Broker
		
<ul style="list-style-type: none"><li>• Trifft alle Entscheidungen für zugeordnete Ressource</li><li>• Bietet Dienstleistung an</li><li>• Kann bestimmte Dienstnehmer ablehnen oder bevorzugen</li><li>• Erwartet Gegenwert abhängig von der Dienstleistung</li></ul>	<ul style="list-style-type: none"><li>• Trifft alle Entscheidungen für die zugeordnete Ressource</li><li>• Sucht die passenden Dienstgeber</li><li>• Entscheidet sich für den geeigneten Dienstgeber</li><li>• Informiert Dienstgeber über die durchzuführende Tätigkeit und ihre Parameter</li></ul>	<ul style="list-style-type: none"><li>• Tritt als Vermittler zwischen den Agenten ein</li><li>• Filtert die Ergebnisse und leitet die günstigsten weiter</li><li>• Entlastet die Dienstnehmer</li><li>• Abstraktion vom dahinterliegenden System</li><li>• Verbirgt Implementierungsdetails</li></ul>

Bild 3.14: Unterscheidung zwischen drei Agentenklassen, die spezifische Rollen im Verbund zur Organisation des Materialflusses einnehmen.

oder Turbulenzen im Materialfluss, langfristige Pläne ständig nachjustiert werden müssen. In diesem Anwendungsfall bietet sich ein Organisationssystem, basierend auf dem Prinzip der ad hoc Entscheidungen, an, das kurzfristig zur Laufzeit Entscheidungen trifft. Dies wird durch den ständigen Abgleich von Zustandsinformationen zwischen den Teilnehmern und die Einbeziehung dieser aktuellen Daten zur Laufzeit in die Entscheidungsprozesse ermöglicht. Zudem konnte in eingehenden Simulationsstudien verschiedener Routingstrategien für den Materialfluss die Effizienz der Agentenstrategien nachgewiesen werden [95].

Im Folgenden wird ein Schema zur Klassifizierung der Agenten in Agentenklassen vorgestellt, das aus den Anforderungen des Materialflusses abgeleitet ist. Die in Kapitel 3.1 beschriebenen Materialflussteilnehmer werden gemäß ihres Rollenverhaltens den Agentenklassen zugeordnet. Abschließend wird ein Beispielszenario einer agentenbasierten Produktionssteuerung aufgezeigt, in dem die Funktionsweise aller Materialflusskomponenten erläutert wird.

3.8.1 Definition der Agentenklassen

Der Agent ist seiner zugehörigen Ressource eindeutig zugeordnet, kennt ihre Fähigkeiten und ihren Zustand und verwaltet basierend auf diesen Daten ihre Aufträge [44]. In einem Agentensystem können die Teilnehmer im Materialflussverbund entsprechend ihrer Aufgabe und somit ihres grundlegenden Verhaltens in drei Kategorien eingeteilt werden (siehe Bild 3.14).

Dienstgeber-Agenten haben die Aufgabe, Dienstleistungen, wie Fertigungs-, Montagearbeiten oder den Transport von Werkstücken, anzubieten. Sie zielen auf eine möglichst hohe Auslastung ihrer zugeordneten Ressource ab, damit diese effektiv genutzt wird und sich somit sehr frühzeitig amortisiert.

Der Agent als Stellvertreter der Dienstgeber-Ressource kann selbständig über die Priorisierung und Favorisierung von Werkstücken entscheiden, inwiefern bestimmte Werkstücke günstiger für die bessere Auslastung der eigenen Ressource sind und somit diese Werkstücke gegenüber anderen bevorzugt werden. Werkstücke, deren Anforderungsprofil nicht dem der zugehörigen Bearbeitungsstationen entspricht, werden abgelehnt.

Dienstnehmer-Agenten repräsentieren die Güter oder Ressourcen, die eine Dienstleistung in Anspruch nehmen. Ihre Aufgabe besteht darin, die möglichen Dienstgeber mit den technischen Voraussetzungen zu suchen und den geeignetsten Dienstgeber mittels Verhandlung herauszufinden. Zudem ist der Dienstnehmer-Agent für die Abwicklung der Dienstleistung mitverantwortlich. Er interagiert mit dem Dienstgeber und informiert diesen über den eigenen Zustand und die durchzuführenden Schritte, die zur erfolgreichen Erbringung der Dienstleistung erforderlich sind. Nach der Bearbeitung, nimmt der Dienstnehmer-Agent die erforderlichen Informationen vom Dienstgeber-Agenten auf, die den weiteren Weg durch die Produktion bestimmen bzw. für das Produkt-Tracking und zur Qualitätsanalyse erforderlich sind.

Der Broker-Agent verfügt über keinerlei eigenständige Entscheidungsgewalt, sondern tritt als Vermittler zwischen Dienstnehmer und Dienstgeber ein. Der Broker erhält Aufträge von den Dienstnehmer-Agenten, wie die Ermittlung geeigneter Ressourcen. Somit kann einerseits der Dienstnehmer-Agent entlastet werden. Andererseits stellt der Broker eine Schnittstelle zwischen dem Dienstnehmer-Agenten und dem Dienstgeber-Agenten dar, um jeweils zu beiden Seiten die dahinterliegenden Systeme zu verbergen und von deren Ausprägung oder genauen Type zu abstrahieren. Dies ist in allen Bereichen erforderlich, in denen die Art und Ausprägung des Dienstgebers unterschiedlich sein kann und der Dienstnehmer mit all diesen Varianten gleichermaßen interagieren muss. Beispielsweise trifft dies zwischen dem Werkstückagenten und dem Transportsystem zu, bei dem der Werkstückagent sowohl z.B. mit fahrzeugbasierten Unstetigfördersystemen als auch mit Stetigfördersystemen basierend auf den gleichen Schnittstellen und dem gleichen Verhalten interagieren soll. Hierzu ist es erforderlich, dass der Broker die nach außen angebotene Funktionalität immer über standardisierte Schnittstellen zur Verfügung stellt, die neben einem einheitlichen syntaktischen auch ein einheitliches semantisches Verhalten aufweisen.

#### **3.8.2 Spezifikation der Teilnehmer im Agentenverbund wandelbarer Produktionsanlagen**

Basierend auf den in Kapitel 3.8.1 eingeführten Agentenklassen werden im Folgenden die Teilnehmer im Materialflusssystem, entsprechend der in Kapitel 3.1 eingeführten Typen, in das agentenbasierte Gesamtsystem eingeordnet. In Bild 3.15 sind die bekannten Materialflussteilnehmer den möglichen Agentenklassen zugeteilt. Hierbei handelt es sich entweder um echte Alternativen, in denen entweder zwischen zwei Agententypen unterschieden werden muss, wie im Falle der Lager, oder der zugehörige Agent kann zwei Rollen gleichzeitig übernehmen, wie bei den Organisationssystemen. Ein Lager kann entweder als eigenständiges Zwischenlager und somit als Dienstgeber agieren oder selbst nur die Rolle als Vermittler, d.h. Broker, einnehmen, der sich selbständig um die Aquisie von Bauteilen kümmert. Die Organisationssysteme zeichnen sich dadurch aus, dass sie zum einen die Ressourcen der zugehörigen Produktionsanlage in Anspruch nehmen, um die Fertigungsaufträge abzuarbeiten. Zum anderen übernehmen sie auch die Rolle eines Dienstgebers, da sie von außen Fertigungsaufträge

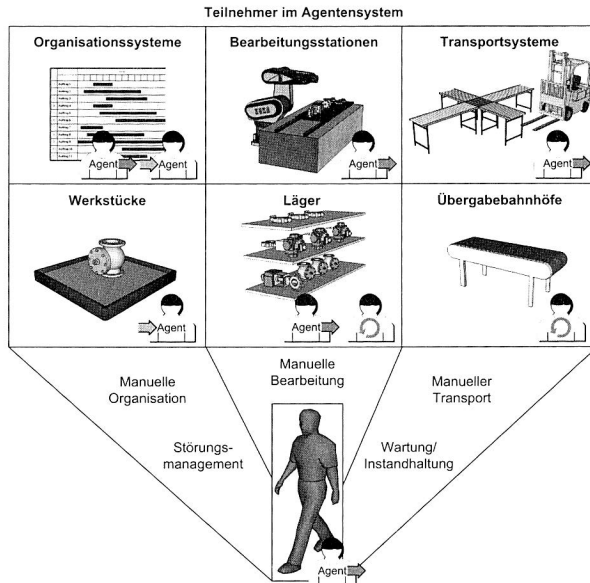


Bild 3.15: Identifikation der Agententeilnehmer mit ihren Rollen im Agentenverbund - der Mitarbeiter kann organisatorisch oder aktiv als Bearbeiter mitwirken.

entgegennehmen und diese auch zwischen mehreren Produktionsanlagen verhandelt werden können. Bearbeitungsstationen und Transportsysteme übernehmen eindeutig die Funktion von Dienstgebern, während Werkstücken die Rolle von Dienstnehmern zugeteilt ist.

Jedem Werkstück ist ein eigener Agent zugeordnet, der das zu fertigende Produkt, die durchzuführenden Bearbeitungsschritte und den Bearbeitungszustand des Werkstücks kennt. Parallel zum Werkstückfluss ist der Werkstückagent in der Lage, sich mit dem Werkstück durch die Anlage zu bewegen. Der Werkstückagent tritt dabei direkt mit den Ressourcen, wie Transportsystem und Bearbeitungsstation, über Verhandlungsmechanismen oder direkte Anweisungen in Verbindung.

Die Übergabebahnhöfe übernehmen hierbei insbesondere die Rolle von Brokern, da sie Anfragen von den Werkstückagenten erhalten und diese an die dahinterbefindlichen Systeme, wie Transportsysteme oder Bearbeitungsstationen weiterreichen und die Ergebnisse vorselektieren. Ein Lager kann jedoch alternativ zum Broker auch die Rolle des Dienstgebers übernehmen, wenn es sich um ein alleinstehendes Lager zur lang- und längerfristigen Zwischenlagerung handelt und dabei mit den Dienstnehmern, hier den Werkstückagenten, in Verbindung tritt.

Allen Systemen ist noch der Mensch übergeordnet, der verschiedene Aufgaben sowohl in der Organisation als auch in der operativen Ausführung übernimmt. Neben dem Einsatz in manuellen Ar-

beitsplätzen oder menschgeführten Transportfahrzeugen, werden alle Tätigkeiten zur Inbetriebnahme, Wartung und Instandhaltung in der Regel vom Menschen übernommen. Somit agiert er selbst auch als Dienstgeber und kann im selbstorganisierenden und automatisierten Anlagenverbund über Assistenzsysteme eingeplant werden. Anfallende Aufgaben werden vom Assistenzsystem, das als Mitarbeiter-agent agiert, mit anderen Teilnehmern verhandelt und diese dem Mitarbeiter vorgegeben. Der Mensch bietet im wandelbaren Produktionsumfeld die höchste Flexibilität für unterschiedliche Aufgaben und kann an unterschiedlichen Orten mit kurzer Umstellungszeit eingesetzt werden.

#### 3.8.3 Funktion und Aufgaben der Werkstückagenten

Das Werkstück ist im Verbund wandelbarer Produktionssysteme immer vom zugehörigen Werkstückagenten abhängig und kann nur mit dessen Hilfe bearbeitet und durch die Anlage navigiert werden. Hierzu ist es erforderlich, die technische Voraussetzung zu schaffen, um den Werkstückagenten durch die Produktionsanlage migrieren zu können. Der Vorgang des Migrierens kann auf unterschiedliche Art und Weise realisiert werden. Zunächst muss bestimmt werden, ob der zugehörige Programmcode zum Werkstückagenten ebenfalls migriert oder ob nur der innere Zustand ausgelesen und an anderer Stelle wiederhergestellt werden soll. Damit jedoch Programmcode portiert und am Zielrechner während der Laufzeit zur Ausführung gebracht werden kann, ist eine Laufzeitumgebung erforderlich, die interpretierbaren Code ausführen kann. Interpretierbare Programmiersprachen, wie Java oder Python, bieten von vornherein diese Möglichkeit. In anderen Sprachen, die Maschinencode erstellen, wie C++ oder Delphi, ist eine Laufzeitumgebung für Skriptsprachen erforderlich, die jedoch in vielen Fällen manuell erstellt werden muss. Alternativ ist ein Plug-in-Mechanismus realisierbar, der binären Programmcode dynamisch selbst zur Laufzeit an die lokale Steuerungssoftware anbindet und zur Ausführung bringt. Hierbei muss jedoch die Kompatibilität des binären Programmcodes zur Rechnerplattform gewährleistet werden.

Der Programmcode eines Werkstückagenten unterteilt sich prinzipiell in zwei Kernkomponenten:

- **Werkstückmanagement:** In dieser Komponente werden die Produktherstellungsdaten, z.B. in Form eines Arbeitsplanes oder Produktionsvorranggraphen (vgl. Kapitel 6.2.1), verarbeitet und zusammen mit den Anlagenstrukturdaten der Weg des Werkstücks durch die Produktion automatisiert abgeleitet (vgl. Kapitel 6.2). Zudem ist diese Komponente in der Lage mit den Ressourcen, z.B. den Bearbeitungsstationen, zu interagieren, um diesen u.a. Prozessspezifika, -parameter und Ablaufskripte zur Durchführung des vorgesehenen Prozessschrittes vorzugeben. Das Werkstückmanagement kann für alle Anwendungen gleichermaßen ohne Anpassungen genutzt und übernommen werden.
- **Verhandlungs- und Navigationskomponente:** Im Gegensatz zum Werkstückmanagement übernimmt die Verhandlungs- und Navigationsroutine anwendungsspezifische Aufgaben und muss für diese angepasst werden. Aus technischer Sicht kommt hierbei ein Skript zum Einsatz, das anwendungs- oder auch produktgruppenspezifisch erstellt und sich auf Aufgaben der Verhandlung und Navigation bzw. Entscheidungsfindung entsprechend anpassen lässt. Somit können unterschiedliche Verhandlungsstrategien implementiert und z.B. abhängig von der Art oder Dringlichkeit eines Auftrages unterschiedlich zum Einsatz kommen. Die Navigationskomponente ist zur Entscheidungsfindung für die Organisation des nächsten Prozessschrittes verant-

wortlich. Sie trifft auf Grund der Ergebnisse oder Qualitätsanalysen der letzten Prozessschritte und der Verhandlungsergebnisse die Entscheidung, welche nachfolgende Bearbeitungsstation den Zuschlag erhält oder falls es sich um eine Fehlproduktion handelt, sortiert sie das zugehörige Werkstück heraus.

Ausgehend von dieser Struktur ist es ausreichend, bei der Werkstückmanagement-Komponente lediglich den Zustand ohne Programmcode zu übertragen. Hingegen bei der Verhandlungs- und Navigationsroutine ist es erforderlich, sowohl Skriptcode als auch den Zustand bei der Migration des Werkstückagenten zu migrieren. Darüber hinaus gibt es wiederum zwei Möglichkeiten, wie der Zustand bzw. Programmcode übertragen werden kann:

- entweder auf einem beschreibbaren Speicher, z.B. beschreibbaren RFID-Tag, oder
- die Übermittlung aller Daten über ein Netzwerk.

Die erste Variante hat den maßgeblichen Vorteil, dass der Speicher immer physisch am Werkstück bzw. am Werkstückträger angebracht ist und neben den Agenten-Daten auch dessen Produktvariante, Bearbeitungszustand und nächste Zielstation hinterlegt sind. Wird das Werkstück aus dem System, z.B. auf Grund von Störungen oder Staus, herausgenommen und zu einem späteren Zeitpunkt wieder in die Anlage eingebracht, kann die Bearbeitung sofort weitergeführt werden, da alle erforderlichen Informationen ausgelesen werden können.

Steht jedoch keine Möglichkeit zur Verfügung, Daten auf beschreibbaren und mobilen Speichermedien abzulegen, können die Daten über das Netzwerk von einem Teilnehmer zum nächsten übermittelt werden. In diesem Fall erfolgt die Migration der Werkstückagenten zwischen den Bearbeitungsstationen, an denen die Werkstückagenten bei der Ankunft der zugehörigen physischen Werkstücke in ihrem ursprünglichen Zustand wiederhergestellt werden. Voraussetzung für die Zuordnung vom Werkstückagent zum physischen Werkstück ist ein einheitliches Identensystem, das jedes Werkstück bzw. jeden Werkstückträger eindeutig identifiziert und somit die Zuordnung zum Werkstückagenten ermöglicht. An der Bearbeitungsstation wird die ID des Werkstücks ausgelesen, wenn dieses als nächstes für die Bearbeitung ausgewählt wurde. Aus der internen Tabelle lokal wartender Werkstückagenten wird dabei der zugehörige Werkstückagent ausgewählt und initiiert (siehe Bild 3.16). Dies veranlasst den Werkstückagenten die erforderlichen Bearbeitungsschritte inkl. der Fertigungsparameter und dem Prozessablaufskript an die Station zu übermitteln und den Vorgang zu starten. Nach erfolgreicher Durchführung informiert die Station den Werkstückagenten über die Fertigstellung und übermittelt die angefallenen Prozess- u. Qualitätsdaten.

Daraufhin liegt die Handlungsfähigkeit wiederum beim Werkstückagenten, der sich um die Nachfolgestation kümmern muss. Hierzu ermittelt er aus einem Arbeitsplan bzw. Produktionsvorranggraphen (Kapitel 6.2) den nächsten Bearbeitungsschritt. Über einen verteilten Katalogdienst, der lokal an den Übergabebahnhöfen zur Verfügung steht, werden die Teilnehmer im Anlagenverbund mit ihren Fähigkeiten verzeichnet. Somit können die zur Durchführung des nächsten Bearbeitungsschrittes fähigen Bearbeitungsstationen ermittelt werden. Der Werkstückagent wendet sich über ein Verhandlungsverfahren an die Stationen, wodurch die geeignete Station ermittelt wird. Das Ziel des Verhandlungsverfahrens ist, alle Teilnehmer im Verbund möglichst gleichmäßig auszulasten, den Gesamtdurchsatz in



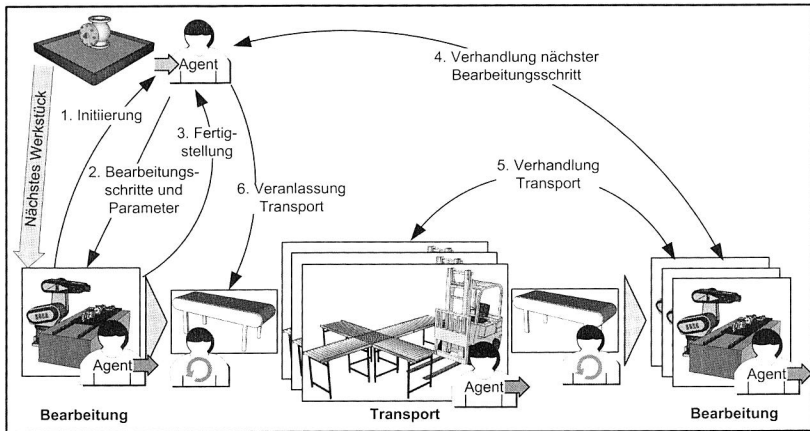


Bild 3.16: Verhandlungsprinzip zwischen Werkstückagent einerseits und den Transportsystemen und Bearbeitungsstationen andererseits - die Marktmacht, d.h. die Entscheidungsgewalt, liegt hier beim Werkstückagenten.

der Fertigung zu maximieren und dabei gleichzeitig die Termintreue zu erreichen.

Grundlegend werden zwei verschiedene Verfahren bei den Verhandlungsprozessen unterschieden, abhängig davon, wer die *Marktmacht* besitzt. Unter Marktmacht wird die Instanz verstanden, die die endgültigen Entscheidungen im Verhandlungsprozess trifft. Dies kann entweder der Werkstückagent in Vertretung für sein Werkstück vornehmen, wenn z.B. zu entscheiden ist, welche Bearbeitungsstation und welches Transportsystem den Zuschlag erhält (siehe Bild 3.17). Umgekehrt kann aber auch die Marktmacht bei den Ressourcen, d.h. bei den Stationen und den Transportsystemen, liegen (siehe Bild 3.18). In diesem Fall entscheiden die Ressourcenagenten primär selbst, welches Werkstück für die optimale Auslastung der eigenen Ressource am besten geeignet ist.

Liegt die Marktmacht beim Werkstückagenten (Bild 3.17), wendet sich dieser über Anfragen zunächst an die Transportsysteme und fragt für alle potentiellen Nachfolgestationen die geschätzte Transportzeit nach. Anschließend wendet er sich an die möglichen Nachfolgestationen und übermittelt in der Anfrage aus der ermittelten Transportzeit den frühesten Ankunftszeitpunkt. Dieser ist wichtig, damit die Bearbeitungsstationen genauer kalkulieren und u.a. in der Zukunft die Auslastung einschätzen können. Die genaue Realisierung dieses Verfahrens ist sehr anwendungsspezifisch und muss von Fall zu Fall erörtert werden. Die jeweiligen Gebote zum Transport und zur Bearbeitung werden für jede potentielle Nachfolgestation zusammengefasst. Diese dienen als Entscheidungsgrundlage für die Werkstückagenten, die ihre Zusage jeweils an die ausgewählte Bearbeitungsstation und das Transportsystem übermitteln.

Wenn die Marktmacht bei den Ressourcen liegt (Bild 3.18), unterscheidet sich das Verfahren dahingehend, dass die zur Entscheidung erforderlichen Informationen nicht am Werkstückagenten, sondern

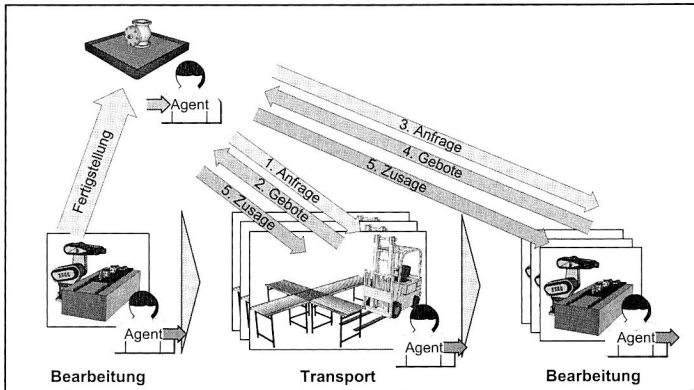


Bild 3.17: Verhandlungsablauf mit Marktmacht beim Werkstückagenten.

an den Ressourcen gesammelt werden. Hierbei werden lediglich die möglichen Nachfolgestationen informiert, dass das Werkstück für die weitere Bearbeitung verfügbar ist. Die Fertigungsspezifikationen, die in den Entscheidungsprozess an den Stationen einfließen, werden zudem mitgesandt. Damit jedoch die Stationen den frühestmöglichen Ankunftszeitpunkt in die Entscheidung einbeziehen können, wenden sie sich an das Transportsystem und fragen nach der geschätzten Transportzeit zwischen den beiden Stationen. Nachdem sich die Station für ein Werkstück entschieden hat, wird dieses in der Zusage nachricht darüber informiert. Der Werkstückagent sendet nochmals die Bereitschaft zum Transport zur neuen Nachfolgestation an das Transportsystem, das wiederum analog zum Verfahren mit den Stationen, die Zusage gibt, das Werkstück zu transportieren.

In den Abbildungen 3.17 und 3.18 sind noch keine Bahnhöfe dargestellt, um zunächst auf die wesentlichen Komponenten einzugehen. Der Bahnhof stellt hierbei die Schnittstelle zum Transportsystem dar und übernimmt alle Kommunikationsaufgaben und Interaktionen zwischen den beteiligten Einheiten auf beiden Seiten, u.a. als Vermittlerrolle bei Verhandlungen (vgl. Bild 3.19). Er ist ein wesentliches Bindeglied im modularen und wandelbaren Aufbau des Produktionssystems, da er erlaubt, verschiedene Transportsysteme anzubinden, die im folgenden Kapitel zunächst klassifiziert werden, bevor detaillierter auf die Funktionsweise des Bahnhofs (Kapitel 6.4) eingegangen wird.

### 3.8.4 Übergabebahnhof als Schnittstelle zwischen Transportsystem und Bearbeitungsstationen

Der Übergabebahnhof bildet immer die Schnittstelle zwischen der Bearbeitungsstation und dem Transportsystem und ist sowohl in die Verhandlungsvorgänge involviert, als auch für die Abwicklung und Koordinierung aller Transportaufgaben, Übergabevorgänge und kurzzeitigen Zwischenlagerung der Werkstücke zuständig. Über den Bahnhof wird von der Art und Ausprägung des dahinterliegenden Transportsystems abstrahiert, was für die Realisierung modularer, wandlungsfähiger Produktionssysteme wesentlich ist. Im Prinzip werden die in Kapitel 3.4 vorgestellten Arten von Transportsystemen - die aktiven und passiven - unterschieden. In beiden Fällen übernimmt der Bahnhof unterschiedli-

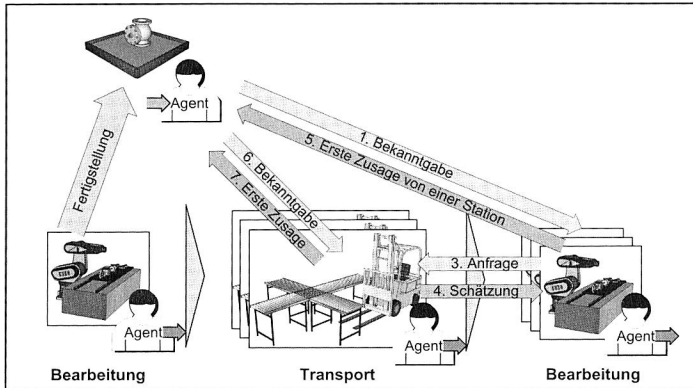


Bild 3.18: Verhandlungsablauf mit Marktmacht beim Ressourcenagenten.

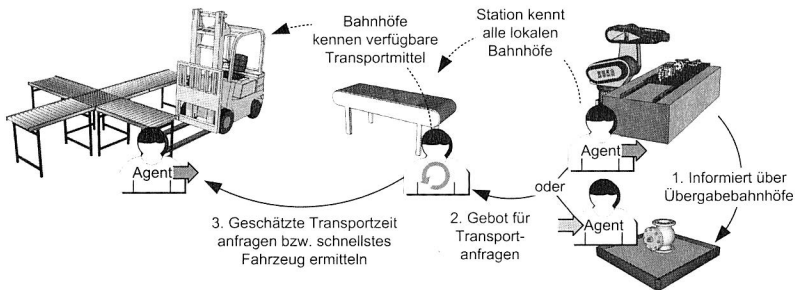


Bild 3.19: Angebotserstellung am Transportsystem zur Ermittlung des schnellsten Transportmittels.

che Aufgaben und erfordert unterschiedliches Verhalten auf Seiten des Transportsystems. Auf Seiten der Bearbeitungsstation und folglich des Werkstücks ist hingegen ein einheitliches Verhalten erforderlich. Somit sind auf informationstechnischer Seite einheitliche Schnittstellen notwendig, um das Transportsystem, unabhängig von dessen Art, d.h. fahrzeug- oder reibschlussbasiert, ansprechen zu können. Diese standardisierten Schnittstellen sind in den Übergabebahnhöfen implementiert, die von den Bearbeitungsstations- und den Werkstückagenten sowohl zur Angebotserstellung bzw. Schätzung der Transportzeit gemäß der aktuellen Auslastung, als auch zur Übergabe des Werkstücks genutzt werden. Der Bahnhof übernimmt somit eine Vermittlerrolle zwischen den Dienstnehmern, d.h. den Werkstücken einerseits und den Dienstgebern, dem dahinterliegenden Transportsystem, andererseits.

Eine Bearbeitungsstation kann an mehrere Transportsysteme angeschlossen sein und somit über mehrere lokal angebundene Übergabebahnhöfe verfügen. Hierzu muss die Station Kenntnis über das Vorhandensein der Bahnhöfe besitzen, jedoch muss die Bearbeitungsstation über keinerlei Wissen der Gesamtstruktur des hinter dem Bahnhof liegenden Transfersystems verfügen. An dieser Stelle ge-

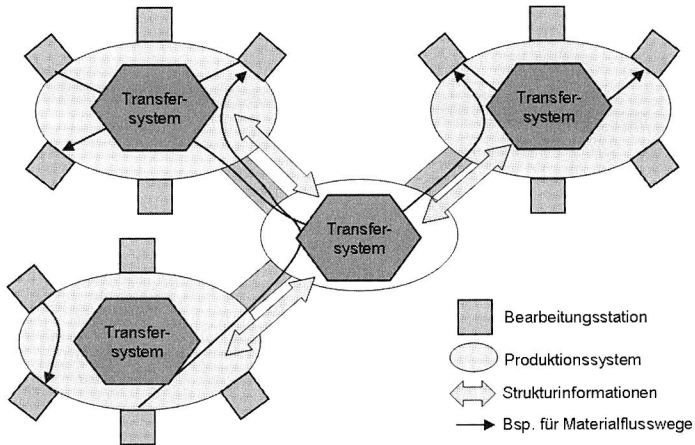


Bild 3.20: Austausch von Strukturdaten zwischen direkt gekoppelten Transfersystemen.

nügt ein lokaler Sichtbereich, den die Bahnhöfe innerhalb des angeschlossenen Transfersystems besitzen. Die einzige Ausnahme stellen zwei direkt miteinander verbundene Transportsysteme dar. Hierbei müssen zwischen diesen beiden unmittelbar Strukturinformationen ausgetauscht werden, damit in jedem Teilsystem bekannt ist, an welchen Stationen die benachbarten Transfersysteme angeschlossen sind (Bild 3.20). Diese Eigenschaft des lokalen Sichtbereichs ist wichtig im Umfeld modularer, wandelbarer Produktionssysteme, um eine stetige und konsequente Trennung der Funktionalitäten in direkt miteinander interagierende Komponenten und somit flexible Umbaumaßnahmen zu gewährleisten. Die Funktionalität jeder Komponente im Produktionssystem konzentriert sich somit einerseits auf die Sub-Komponenten, die der Komponente direkt als Unterfunktionseinheiten zugeordnet sind, und andererseits auf die direkt benachbarten, angeschlossenen Komponenten. Die Grenzen der lokalen Sichtbereiche, d.h. der Kenntnis der dahinterliegenden Komponenten, legen hierbei die Bahnhöfe als Separatoren fest, die auf Seiten des Transfersystems und auf Seiten der Bearbeitungsstation bzw. dem Werkstückagenten den Sichtbereich begrenzen. Durch diese Begrenzung der Sichtbereiche kann eine schnelle Integration der Komponenten gewährleistet werden, da diese keine anlagen- oder ortsspezifischen Einstellungen benötigen, um z.B. sich in den Anlagenverbund zu integrieren und dabei den zellenübergreifenden Materialfluss gewährleisten zu können. Diese aufwandsarme Integrationsmöglichkeit wird mit einem neuartigen Strukturerkennungsmechanismus gewährleistet, der automatisiert die Anlagenstruktur erkennt und daraus die grundlegenden Daten für Materialflussentscheidungen ableitet (siehe Kapitel 4.3).

### 3.8.5 Vermittlerrolle des Übergabebahnhofs

Neben der Schnittstellenaufgabe zur Modularisierung des Anlagenaufbaus, übernimmt der Bahnhof auch wesentliche organisatorische Aufgaben zur Steuerung und Koordination des Materialflusses. Diese unterteilen sich zum einen in die Bestimmung der nächsten potentiell verfügbaren Bearbeitungsstationen für ein Werkstück und zum anderen in der Bewerksstellung einer Infrastruktur zur

Verhandlung und Bewertung der nächsten Bearbeitungsschritte an den potentiellen Bearbeitungsstationen und der Bewertung des Transports. Mit dieser Vorgehensweise wird der Forderung nach den lokalen Sichtbereichen Genüge getan. Somit benötigen die Bearbeitungsstationen kein globales Wissen, d.h. welche anderen Bearbeitungsstationen in der Anlage existieren und wie diese erreichbar sind. Die erforderliche Informationsvorratshaltung beschränkt sich auf das bzw. die modular angeordneten Transportsysteme mit den Übergabebahnhöfen.

Zur Wahrung der Dynamik und der schnellen Reaktionsfähigkeit bei Wandlungen, sollte der Werkstückagent lediglich den Arbeitsplan des zu fertigenden Produktes und die darin enthaltenen Prozessschritte kennen. Somit muss zur Laufzeit die Verbindung der verfügbaren Bearbeitungsstationen mit ihren Fähigkeiten und dem jeweils nächsten erforderlichen Prozessschritt erfolgen (Kapitel 3.8.3). Bedingt durch die Forderung der lokalen Sichtbereiche, kennt die Bearbeitungsstation die restlichen Stationen in der Anlage nicht. Die Informationen, welche Bearbeitungsstationen verfügbar sind, welche Prozessschritte sie durchführen können und wie sie erreichbar sind, liegt lediglich an den Übergabebahnhöfen vor. Zur Ermittlung der nächsten Bearbeitungsstation bzw. Evaluierung der geplanten Belegungszeiten an den Ressourcen, initiiert der Werkstückagent einen sogenannten Verhandlungsagenten. Dieser übernimmt die Funktion eines kurzzeitigen Stellvertreters, der im Namen des Werkstückagenten Teilaufgaben übernimmt. Dieser wird analog zur Anzahl der vorhandenen Bahnhöfe mehrfach instanziiert und dient lediglich zur Bewertung der nächsten Prozessschritte. Nach erfolgreicher Bestimmung der Nachfolgestation werden die Instanzen der Verhandlungsagenten nicht mehr benötigt.

Der Werkstückagent instanziiert für jeden Bahnhof einen Verhandlungsagenten, dem er den nächsten Prozessschritt oder alternativ eine Liste der nachfolgenden Prozessschritte übergibt. Da einer Bearbeitungsstation mehrere Transportsysteme zur Verfügung stehen können, die unterschiedliche oder auch gemeinsame Stationen miteinander verbinden, ist die Einbeziehung aller angeordneten Bahnhöfe zur Evaluierung der nächsten Station vorgesehen. Sollte die erforderliche Station über das dahinterliegende Transportsystem nicht erreichbar sein, wird lediglich die Nichterreichbarkeit zurückgemeldet. Dies bringt den Vorteil mit sich, dass im Falle von Änderungen am Anlagenaufbau oder am Materialfluss des Produktionssystems keinerlei Anpassungen an den Bearbeitungsstationen oder Materialflussregeln im Transportsystem erforderlich sind und sich das System selbst zur Laufzeit an Veränderungen anpasst (vgl. Kapitel 4).

Die Übergabe einer Liste von Prozessschritten an den Verhandlungsagenten ist nur dann relevant, wenn nicht nur für den nächsten Schritt die möglichen, nachfolgenden Bearbeitungsstationen, sondern die nächsten  $n$  Schritte evaluiert werden sollen. Dies hat den Vorteil, dass über die nächste Station hinaus, die Belastung der nachfolgenden Stationen in den Bewertungsprozess einbezogen werden kann. Somit kann beispielsweise die Bewertung und effektive Entscheidung für redundante Teillinien, bestehend aus mehreren Stationen, die über einen unidirektionalen Materialfluss verbunden sind, getroffen werden. Hierdurch können z.B. Staus oder Verzögerungen, die nicht in der nächsten, aber übernächsten Station auftreten, frühzeitig erkannt und entsprechend darauf reagiert werden, indem eine andere redundante Teillinie gewählt wird.

Nachdem der Verhandlungsagent zum Übergabebahnhof migriert ist, fragt er vom Bahnhof die Liste der möglichen Bearbeitungsstationen zu einem Prozessschritt an. Diese Information liegt beim Bahn-

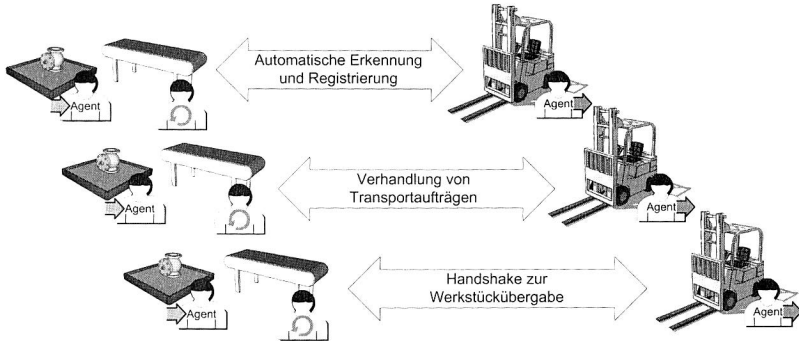


Bild 3.21: Aufgaben und erweiterte Funktionalität des Übergabebahnhofs für spurungebundene Fahrzeuge in der wandelbaren Produktionsumgebung.

hof vor, da er weiß, welche benachbarten Bahnhöfe mit den dahinterliegenden Bearbeitungsstationen das zugeordnete Transportsystem verbindet. Hierzu existiert ein eigens entwickelter Mechanismus zur Strukturerkennung, der es erlaubt, die Anlagenstruktur selbst zur Laufzeit mit einer niedrigen Reaktionszeit zu erkennen (vgl. Kapitel 4.3 und 4.5.1). Die Verbindung der Produktherstellungsdaten mit den Anlagenstrukturdaten wird detailliert in Kapitel 6.2 beschrieben.

Der Verhandlungsagent initiiert, basierend auf dieser Stationsliste, die Verhandlungen mit diesen Stationen, um zu ermitteln, welche Bearbeitungsstation die günstigste zur Durchführung des nächsten Prozessschrittes ist. Parallel dazu ermittelt er zu jeder Bearbeitungsstation über den Bahnhof die erforderliche Transportzeit (Bild 3.19). Sind alle grundlegenden Voraussetzungen, wie die zu transportierende Werkstückgröße und das Gewicht, erfüllt, ermittelt der Bahnhof die geschätzte Transportzeit. Hierbei kommt es darauf an, um welches Transportmittel, d.h. passives oder aktives System, es sich handelt. Im ersten Fall wird eine Anfrage an die nächste benachbarte, intelligente Kreuzung gesandt, die im verteilten Steuerungsnetzwerk die Zeitschätzung für den Transport durchführt (vgl. Kapitel 7.1). Im zweiten Fall der fahrzeuggesteuerten Systeme wird zwischen dem Bahnhof und allen Fahrzeugen eine Verhandlung initiiert, um das geeignete Fahrzeug zu ermitteln. Die Aufgaben, die der Bahnhof im Falle der fahrzeuggesteuerten Transportsysteme übernimmt und welche Interaktionen erforderlich sind, ist in Bild 3.21 dargestellt. Nachdem der Bahnhof das günstigste Fahrzeug ermittelt hat, leitet er sein Gebot an den Verhandlungsagenten weiter.

Liegt die Marktmacht beim Werkstück- und somit beim Verhandlungsagenten, ermittelt letzterer zunächst vorab die günstigste Bearbeitungsstation. Im Vergleich fließt dabei die Summe der frühesten Fertigstellungszeitpunkte und der Transportzeiten in die Entscheidung ein. Liegt die Marktmacht bei den Bearbeitungsstationen, informiert lediglich der Verhandlungsagent diese über das Werkstück und wartet auf den ersten eingegangenen Zuschlag einer Bearbeitungsstation, die das Werkstück bearbeiten möchte. Dieses Ergebnis wird an den Werkstückagenten zurückgemeldet. Steht jedoch an einem Bahnhof keine Bearbeitungsstationen mit der geforderten Funktionalität zur Verfügung, meldet er dieses Ergebnis an den Werkstückagenten zurück. Liegt die Marktmacht beim Werkstückagenten, so

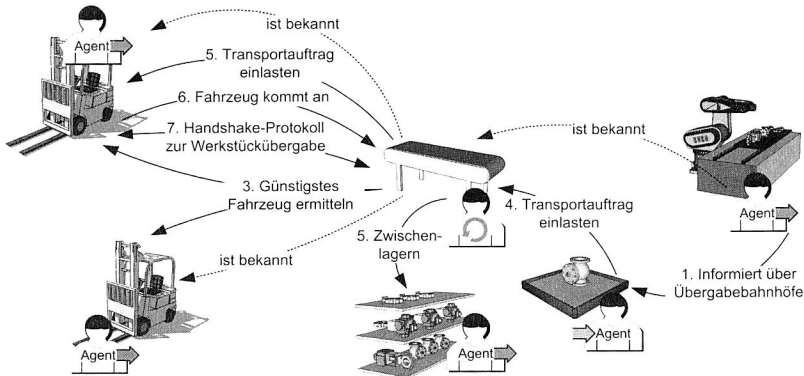


Bild 3.22: Interaktion zwischen Fahrzeugen und Bahnhöfen zur Ermittlung des günstigsten Fahrzeugs - Einlastung des Auftrags und Palettenübergabe mittels angepasstem Handshakemechanismus.

muss er sich in zweiter Instanz ggf. aus mehreren Vorschlägen seiner Verhandlungsagenten für die günstigste Vorgehensweise, d.h. Bearbeitungsstation mit Transportweg, entscheiden. Zudem ist an dieser Stelle ein Timeout erforderlich, um den Verlust von Nachrichten oder Ausfällen von Dienstgebern, die auf keine Nachrichten mehr antworten können, zu kompensieren. Dieser Timeout muss am Werkstückagenten eingebaut sein, um als letztes Glied in der rekursiven Kette mehrfacher Aufrufe den Abbruch einleiten zu können.

Sobald die Verhandlungsergebnisse beim Werkstückagenten eingetroffen sind, stößt dieser den Transportauftrag an (vgl. Kapitel 3.8.3). Die Zusage für ein Transportmittel teilt immer der Werkstückagent dem Bahnhof mit, unabhängig davon wer die Marktmacht inne hat. Selbst wenn diese bei den Bearbeitungsstationen liegt, wird der Werkstückagent darüber informiert, welche Bearbeitungsstation ihn ausgewählt hat.

In aktiven Fahrzeugsystemen muss das geeignete Fahrzeug für den Transportauftrag ermittelt, der Auftrag beim Fahrzeug eingelastet und die Übergabe des Werkstücks mittels eines Handshakemechanismus koordiniert werden (vgl. Bild 3.22). Hierbei handelt es sich um ein Kommunikationsprotokoll zwischen dem Fahrzeug und dem Bahnhof, das ausschließlich bei automatisierter Werkstückübergabe zum Einsatz kommt. Dies ist insbesondere bei Fahrerlosen Transportsystemen erforderlich, da hierbei zwischen beiden Teilnehmern, sowohl zum Aufnehmen als auch zum Abgeben von Werkstücken die Bereitschaft gegenseitig signalisiert werden muss. Eine ausführliche Beschreibung des Handshake-mechanismus für Fahrerlose Transportfahrzeuge mit autonomer, agentenbasierter Steuerung wird in Kapitel 6.4 erläutert.

### 3.8.6 Lageragent zur Verwaltung der Ein- und Auslagerungsvorgänge

Der Lageragent kann zwei verschiedene Rollen im Anlagenverbund einnehmen, abhängig vom Lagertyp (Kapitel 3.3). Push-Läger übernehmen die Dienstgeberrolle, da sie die Dienstleistung zum

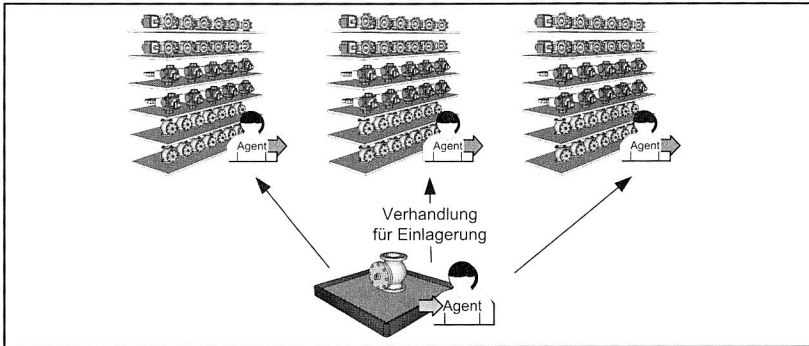


Bild 3.23: Lager als Dienstgeber, um Werkstücke für einen bestimmten Zeitraum zwischenzulagern. Der Werkstückagent tritt hierzu zur Wahl des geeigneten Lagers über einen Verhandlungsmechanismus mit den Lageragenten in Verbindung.

Einlagern von Gütern anbieten. Pull-Läger tragen die Rolle eines Brokers, da ihnen eine Vermittleraufgabe zugewiesen wird, z.B. Mindestmengenvorräte von Bauteilen sicher zu stellen und diese bei Bedarf selbständig von beliebigen, zur Laufzeit zu bestimmenden Quellen anzufordern.

In der Dienstgeberrolle verhandelt der Werkstückagent u.a. mit den möglichen Lägern, in welches das zugehörige Werkstück günstigerweise aufgenommen werden kann (siehe Bild 3.23). Diese Art von Lägern sind dedizierte Zwischenlager, die sich innerhalb der Produktion befinden können, z.B. wenn ein Produkt in einem Zwischenstadium der Fertigung längere Zeit abgelegt werden soll, um zu einem späteren Zeitpunkt weiterproduziert zu werden. Das Zwischenlager kann sich sowohl innerhalb eines Produktionssystems, als auch am Ende zur Aufnahme der fertiggestellten Produkte oder Zwischenerzeugnisse befinden. Von letzterem aus erfolgt die bedarfsgerechte Auslieferung zu weiteren Produktionssystemen oder zum Versand.

Ebenso die Werkstückläger an den Übergabebahnhöfen agieren als Dienstgeber. Bei den aktiven, fahrzeugbasierten Systemen müssen die Werkstücke bis zur Ankunft des Fahrzeugs zwischengelagert werden. Hierzu ist dem Übergabebahnhof ein Lager zugeordnet, dessen Dienste nur vom Bahnhof genutzt werden (vgl. Bild 3.22). Bei den passiven, spurgeführten Systemen, wie den Reibschlusstransfersystemen, ist i.d.R. kein Zwischenlager erforderlich. Dennoch sind für bestimmte Implementierungen, wie den fahrzeugbasierten Unstetigfördersystemen, die Lager als Option notwendig. Im Gegensatz dazu benötigen die passiven Stetigfördersystemen in der Regel keinerlei Mechanismen zur Einlastung von Transportaufgaben. Reibschlusstransfersysteme zum Beispiel, können unmittelbar ohne Voranmeldung eines Transportauftrags oder Vorausplanung Werkstücke entgegennehmen, da sie selbst auf den Transfersegmenten über Pufferkapazitäten verfügen.

Die Vermittlerrolle übernimmt ein Lager zwischen Dienstnehmer und Dienstgeber, wie im Falle des Bauteillagers an der Bearbeitungsstation (vgl. Bild 3.24). Hier besteht die Aufgabe des Lagers darin, Bauteile eines bestimmten Typs vorrätig zu halten, damit diese in der Montage an der zugeordneten



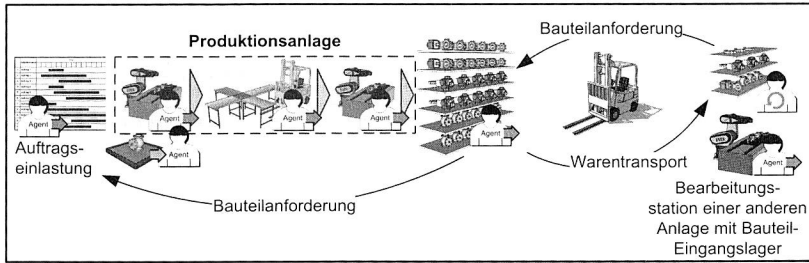


Bild 3.24: Ein Lager kann sowohl Dienstgeber, als auch Broker sein, der für eine zugeordnete Ressource bestimmte Auflagen und Aufgaben erfüllen muss, wie die Gewährleistung eines Mindestbestands bestimmter Bauteile.

Bearbeitungsstation verfügbar sind. Die Bearbeitungsstation als Dienstnehmer gibt dem Bauteillager vor Ort die Aufgabe, Bauteile nach Bedarf von anderen dienstgebenden Lägern anzufordern, sofern ein Mindestbestand unterschritten ist. Das Bauteillager ist hier in der Lage, die ihm zugewiesene Aufgabe selbständig zu erfüllen. Bevor es die Bauteilanforderung bearbeiten kann, muss es zur Laufzeit die potentiellen Quellen, d.h. Zwischenlager, ermitteln. Diese Flexibilität ist notwendig, um der schnellen Anpassungsfähigkeit nach Veränderungen an der Anlage zur Laufzeit genügen zu können.

Darüber hinaus bietet es Schnittstellen an, um prüfen zu können, ob Bauteile verfügbar sind. In einem iterativen Prozess muss die Auftragseinlastung vor der Freigabe eines Auftrags prüfen, ob alle Voraussetzungen erfüllt sind. Hierzu wird bei allen an der Fertigung eines Produktes beteiligten Stationen angefragt, ob die Bearbeitungsschritte durchgeführt werden können. Voraussetzung für die Montage ist das Vorhandensein von Bauteilen, deren Existenz am Bauteillager geprüft und die erforderliche Menge ggf. vorreserviert werden muss.

## Kapitel 4

# Verteilte Steuerungsarchitektur als Grundlage für den effektiven Einsatz in der Produktion

Zur Realisierung der in Kapitel 2 beschriebenen Anforderungen an flexible und wandelbare Produktionssysteme wird in diesem Kapitel ein Konzept aufgezeigt, das neben einem modularen mechanischen Aufbau die analoge softwaretechnische Modularisierung vorsieht. Zur durchgängigen Umsetzung wandelbarer Systeme auf allen Ebenen der Produktion wird ein Baukasten-Ansatz verfolgt, der Komponenten für den flexiblen und modularen Aufbau von Anlagensystemen vorsieht, in denen eine Komponente den Verbund des Gerätes mit der Rechen- und Steuerungstechnik darstellt. Durch diesen Ansatz werden sowohl die Skalierbarkeit von Produktionsanlagen, als auch die schnelle Inbetriebnahme bzw. Umbaumaßnahmen sichergestellt. Im Folgenden werden zu Beginn in den Kapiteln 4.1 und 4.2 die Ansätze für eine verteilte Steuerungsumgebung beschrieben, die sich aus autonom gesteuerten Einheiten zusammensetzt. Im Anschluss daran wird die auf diesen Ansätzen basierende Steuerungsarchitektur vorgestellt, die mittels Selbstkonfigurations- und Selbstorganisationsmechanismen den Entwicklungs- und Inbetriebnahmeaufwand reduziert (Kapitel 4.3 und 4.4). Die Umsetzung und Implementierung selbst gliedert sich dabei ebenfalls in mehrere Softwaremodule, die u.a. unabhängig voneinander genutzt werden können (Kapitel 4.5).

### 4.1 Dezentraler Steuerungsansatz für wandelbare Produktionsumgebungen

Der Lösungsansatz zur Bewerkstelligung einer hohen Wandelbarkeit sind Baukastensysteme, bestehend aus Anlagen-Komponenten, die einen flexiblen und schnellen Aufbau von Anlagensystemen erlauben. Dabei kann eine hohe Wandelbarkeit nur dann erreicht werden, wenn der Komponenten-Ansatz ganzheitlich in allen Fachbereichen, d.h. der Mechanik, Elektrotechnik und Steuerungstechnik, angewandt wird. Dabei müssen die Komponenten nach außen über standardisierte Schnittstellen, sowohl zur physischen Verbindung der Komponenten untereinander, zur Energiezuführung und Anbindung an das Informationsnetz als auch zur softwaretechnischen Integration der verteilten Steuerungsfunktionen verfügen. Dabei setzt der Bereich der Steuerungstechnik voraus, dass jede Komponente mit einer eigenen Recheneinheit versehen ist und alle Aufgaben zur Steuerung bzw. Regelung der zugehörigen Komponente übernimmt. Somit werden die Komponenten nicht von einer übergeordneten, entfernten, zentralen Steuereinheit gesteuert, sondern die Steuerungsfunktionalität bzw. -logik, umgesetzt in der Steuerungssoftware, befindet sich direkt am Gerät vor Ort. Daraus resultiert eine Komponente bestehend aus dem Gerät, der zugehörigen Elektronik und der Rechnerhardware mit der darauf befindlichen Software. Durch diese Vereinigung wird auch die Austauschbarkeit der Komponenten begünstigt, da alle Aspekte des Anlagenaufbaus in dieser vereint sind und somit nicht getrennt betrachtet bzw. bearbeitet werden müssen. Dieser Ansatz führt neben einer verbesserten Skalierbarkeit auch zu einer Steigerung der Autonomie der Komponenten und zu einer Verringerung der Ge-

samtausfallwahrscheinlichkeit der Produktionsanlage. Das Konzept sieht zudem die Möglichkeit vor, Rechentechnik bedarfsgerecht auf allen Ebenen der Produktion einzusetzen.

Folglich bedeutet dies, dass durch die Modularisierung und Verteilung der Software auf jede Komponente, sich ein Verbund echt verteilter Steuerungssysteme in der gesamten Produktionsanlage bildet. Dies setzt voraus, dass die verteilten Steuerungssysteme in der Lage sein müssen, miteinander zu interagieren, um im Verbund eine gemeinsame Aufgabe zu erfüllen. Dieser Ansatz steht im Gegensatz zu den bisher bekannten zentralistischen Steuerungsstrukturen, bei denen ein zentraler Rechner über ein Bussystem verteilte Ein- und Ausgangsmodule abfragt bzw. diese ansteuert [25, 96].

Um die Variabilität bewahren zu können, Komponenten schnell und mit geringem Aufwand austauschen bzw. die Anlage um neue erweitern zu können, müssen in der Entwicklung der Softwaresysteme die Grenzen der funktionalen Aufgaben einer jeden Komponente nach oben zu den Superkomponenten und zu benachbarten Komponenten eingehalten werden. Dies bedeutet, dass eine Komponente nur den Subkomponenten Befehle vorgeben darf und somit nur die eigene oder untergeordnete Komponenten, keinesfalls Komponenten auf gleicher oder übergeordneter Ebene, steuern oder regeln darf. Dies schränkt nicht die Fähigkeit ein, mit beliebigen Komponenten unabhängig von der hierarchischen Stufe auf Peer-to-Peer-Basis zu kommunizieren, sondern legt lediglich fest, wer die Koordinierungsgewalt besitzt, d.h. komponentenübergreifend Steuerungs- und Regelungsaufgaben übernimmt.

Die homogenen Interaktionsformen und hier vorgestellten Kommunikationsmittel (Kapitel 4.5) sehen die direkte Interaktion vom Informationserzeuger bis hin zum Verbraucher vor. Somit entfallen alle Zwischenschichten bzw. zwischengelagerte Rechner, die im Regelfall Aufgaben zur Informationsvermittlung, Datenfilterung oder Datenvorverarbeitung übernehmen. Dadurch kann der Entwicklungsaufwand sowohl beim Anlagenaufbau als auch bei späteren Umbauphasen reduziert werden, da nur die kommunizierenden Partner, jedoch nicht weitere Teilnehmer, konfiguriert und angepasst werden müssen.

Um effektive Wandelbarkeit zu erzielen, muss die in der Praxis übliche Heterogenität der Rechentechnik einem durchgängigen homogenen Architektur-Konzept weichen. Das bedeutet, dass nur eine Rechnerarchitektur verwendet wird, mit dem Ziel, Mehrfächanpassungen für verschiedene Plattformen bei Anlagenmodifikation zu vermeiden und so Entwicklungsaufwendungen zu reduzieren. In dieser verteilten Systemlandschaft ist die homogene informationelle Vernetzung aller Komponenten Voraussetzung für die Gewährleistung einer flexiblen und bedarfsgerechten Wandelbarkeit mit reduziertem Inbetriebnahmeaufwand. Damit jedoch Informationen von jedem beliebigen Erzeuger in der Produktionsanlage ohne weitere Zwischenspeicherung oder -verarbeitung direkt zum Verbraucher übermittelt werden können, ist folgende Funktionalität von Bedeutung:

- Ein Adressierungsschema für den anlagenübergreifenden Zugriff auf alle Modellbausteine in einer Produktionsanlage.
- Die Vergabe eindeutiger IDs an jede Komponente, um diese, unabhängig von ihrer aktuellen Position innerhalb der Anlagenstruktur, unterscheiden zu können. Die Notwendigkeit hierzu liegt in einer möglichen Positionsänderung einer Komponente, da z.B. eine Komponente von einer Teilanlage des Produktionssystems in eine andere versetzt werden kann. Zudem können

mehrere Komponenten des gleichen Typs zum Einsatz kommen, die jeweils unterscheidbar sein müssen.

- Eine durchgängige Kommunikations-Middleware, eingegliedert in einer umfassenden Steuerungsarchitektur, die von der Kommunikationstechnik und den Protokollen abstrahiert und einen selbst rechnerübergreifenden Zugriff zwischen hierarchisch organisierten Modulen erlaubt. Gleichzeitig muss sie den spezifischen Anforderungen der Wandelbarkeit genügen, um im Falle von Änderungen oder Ausfällen dennoch den Betrieb bestmöglich aufrecht zu erhalten.
- Die Festlegung der Interaktionsformen zur Beschreibung der Kommunikationssemantik bzw. des Kommunikationsverhaltens zwischen zwei oder mehreren Teilnehmern. Die in Kapitel 4.5 vorgestellte Steuerungsarchitektur bietet standardisierte Interaktionsformen an, die fallspezifisch, abhängig von den Anforderungen, eingesetzt werden können.

## **4.2 Hierarchische Modellierung und Anordnung der Anlagenkomponenten**

Der hier verfolgte Ansatz zur Modellierung der Struktur der Steuerungssysteme folgt dem modularen Gedanken auf allen Ebenen der Produktion. Der Modulgedanke drückt aus, dass abgeschlossene, funktional autarke Module zum einen selbst aus Submodulen aufgebaut sein können, zum anderen können diese wiederum in Supermodulen eingesetzt werden, um dort in Kooperation mit anderen Modulen das funktionale Ganze zu bilden [97]. Analog zum realen Aufbau der Anlagen, gemäß dem Baukastenprinzip (Kapitel 4.1), zusammengesetzt aus wiederverwendbaren Komponenten, wird der gleiche Baukastenansatz im Bereich der Steuerungssoftware nachgebildet. Hierzu kommt eine Baumstruktur zum Einsatz, die primär die strukturelle, hierarchische Beziehung der physischen Module untereinander beschreiben soll. Ein Gerät, bestehend aus verschiedenen Sub-Komponenten, diese wiederum aggregiert aus Sub-Sub-Komponenten usw. können somit identisch in Software nachgebildet werden (Bild 4.1). Den Komponenten kann ein abgeschlossener Programmcode mit eigenem Funktionsumfang zugeordnet werden, der für standardisierte Module in einem Baukastensystem vorliegt und lediglich in die Endanwendung eingebunden werden muss. Dies fördert maßgeblich die Wiederverwendbarkeit von Programmcode und erlaubt funktional abgeschlossene Module mit Standardschnittstellen zu verwenden. Neben der vereinfachten Inbetriebnahme der Softwaremodule wird zudem die frühzeitige Validierung der Einzelkomponenten ermöglicht, wodurch beim Zusammenstellen der Endanwendung funktionsfähige Softwaremodule zum Einsatz kommen.

Der Hierarchiebildung ist an dieser Stelle keine Grenze gesetzt. Die Abbildung der Steuerungsstruktur analog zur physischen Struktur kann auf allen Ebenen einer Produktion vollzogen werden (vgl. Bild 4.1). Ein Fertigungsstandort kann mehrere Produktionssysteme beherbergen, die wiederum aus Zellen und Subzellen zusammengesetzt werden. Die Zellen enthalten Produktionseinrichtungen für die Bearbeitung und Transportsysteme zur Förderung der Werkstücke innerhalb einer Zelle. Zellen wiederum können aus mehreren Zellen aufgebaut sein, die untereinander über ein Transfersystem Waren austauschen. Eine Produktionseinrichtung oder ein Transfersystem besteht aus Modulen, die selbst hierarchisch bis zum unteilbaren Basismodul zerlegbar sind. Mit diesem Ansatz erstreckt sich der hierarchische Steuerungsbaum ebenenübergreifend, indem die lokalen Bäume der Einzelkomponenten miteinander zu einem großen, globalen Steuerungsbaum verknüpft werden. Jedem Knoten

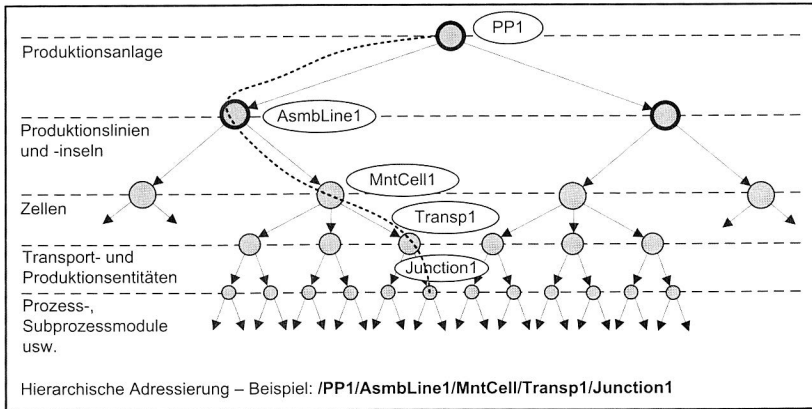


Bild 4.1: Alle Komponenten der Produktionsanlage sind in einem hierarchischen Baum angeordnet. Gemäß diesem orientiert sich das Gesamtsteuerungsmodell und das Adressierungsschema, in dem die Komponentennamen mit einem Separator konkateniert werden.

innerhalb des Steuerungsbaums können sogenannte Funktionsbausteine zugeordnet werden. Diese repräsentieren die Funktionsweise des zugeordneten Moduls auf physischer Seite und sind in der Lage, dieses zu steuern oder Schnittstellen nach außen zur Interaktion bereitzustellen. [87]

Im Baum gibt es zudem ausgezeichnete Knoten, die sogenannten Separatoren, die Grenzen abgeschlossener Module spezifizieren (Bild 4.2) [98]. Alle Knoten unterhalb eines Separators, d.h. der darunterliegende Teilbaum, sind Bestandteile des Moduls. Da ein Modul auch aus Submodulen bestehen kann, können somit auch weitere Separatoren unterschiedlicher hierarchischer Stufen im Teilbaum existieren. Jedem Separatorknoten ist eine ganzzahlige negative Zahl zugeordnet, die in absteigender Richtung im Baum abnimmt. Der globale Wurzelknoten hält die Segmentzahl *-1*. Die Separatoren stellen hierbei eine lokale Referenz zur Verfügung, um ausgehend von diesen alle untergeordneten Knoten relativ adressieren zu können.

Parallel zum hierarchischen Steuerungsbaum orientiert sich das Adressierungsschema. Jeder Knoten des Baumes verfügt über einen Vaterknoten und kann Kindknoten enthalten. Jeder Knoten verfügt über einen Namen, der diesen auf der lokalen Ebene, d.h. beim Vaterknoten, eindeutig identifiziert. Benachbarte Knoten einer Ebene dürfen folglich nicht den gleichen Namen tragen. Die global eindeutige Adresse setzt sich aus der Konkatenation der Knotennamen entlang eines Pfades, ausgehend vom Wurzelknoten, zusammen (Bild 4.1) [87]. Es gibt verschiedene Möglichkeiten einen Knoten innerhalb des Baumes zu adressieren:

- Absolute Adressierung: ausgehend von der Baumwurzel
- Zonenorientierte Adressierung: ausgehend von bestimmten Separatoren, die eine Zone spezifizieren, können alle Subknoten adressiert werden [98]. Hierzu wird die negative Ganzzahl des Separators an die erste Stelle des Adresspfades gesetzt.

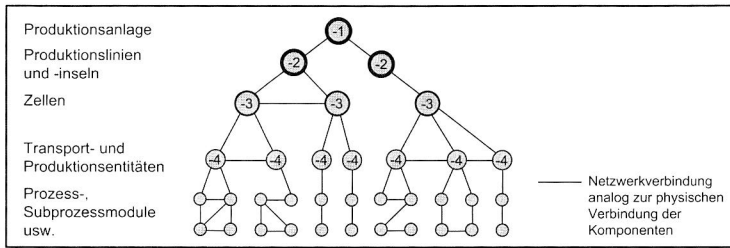


Bild 4.2: Da die Knoten auch untereinander frei verbunden sein können, gibt die Netzwerkstruktur keinen Aufschluss über die organisatorische Hierarchie. Die Separatoren sind an den jeweiligen Rechnern hinterlegt und werden mit einer negativen Zahl gekennzeichnet, wobei der höchste Knoten die -1 trägt. Dieser kann anwendungsabhängig frei definiert werden - in diesem Beispiel repräsentiert er die Produktionsanlage.

- Relative Adressierung: ausgehend vom aktuellen Knoten kann mit „...“ aufwärts bzw. auch abwärts gegangen werden.
- Lokale Adressierung: ausgehend vom Wurzelknoten eines lokalen Baumes können alle Subknoten hiervon adressiert werden.

Um Wandelbarkeit gewährleisten zu können, sollte nach Möglichkeit immer auf die absolute Adressierung verzichtet werden. Änderungen an der übergeordneten Baumstruktur eines Knotens, z.B. ausgelöst durch den Umbau eines Submoduls von einem Supermodul in ein anderes, haben die Änderung des absoluten Pfades, vom globalen Wurzelknoten ausgehend, zur Folge. Daher ist darauf zu achten, möglichst relativ und lokal bzw. wenn erforderlich, auch zonenorientiert zu adressieren.

### 4.3 Selbsterkennung und Selbstkonfiguration der Entitäten

Nach der Wandlung einer Produktionsanlage ist eine Voraussetzung für die steuerungstechnische Funktionsbereitschaft der Anlage die Kenntnis über die vorhandenen Komponenten und deren strukturelle Beziehungen. Diese Informationen stellen die Grundlage zur Ableitung weiterführender steuerungstechnisch und organisatorisch relevanter Daten dar, u.a. die automatisierte Generierung der Routentabellen für den Materialfluss (Kapitel 7.2) und die Verknüpfung der Produktherstellungsdaten mit den Anlagendaten (Kapitel 6.2). Um hierbei den manuellen Entwicklungs- und Konfigurationsaufwand zu reduzieren, wurde ein Strukturerkennungsalgorithmus entwickelt, der automatisiert aus dem aktuellen Anlagenaufbau unmittelbar und selbständig die Anlagenstruktur erkennen kann. Analog zum physischen Aufbau der Anlage aus stationären Komponenten bzw. Entitäten, wird ein isomorpher Aufbau des Kommunikationsnetzwerks und der Anlagenstruktur vorausgesetzt. Eine *Entität* repräsentiert die Komposition aus Rechnerhardware und Software und stellt eine steuerungstechnisch abgeschlossene, funktionale Einheit dar. Anhand des Beispiels der passiven, spurgeführten Transfersysteme müssen Netzwerkkabel zwischen Entitäten, in diesem Fall den Kreuzungen und Bearbeitungsstationen, verlegt werden (vgl. Bild 4.3). Zwar ist damit auf der einen Seite ein leicht erhöhter Aufwand verbunden, bietet aber auf der anderen Seite folgende maßgebliche Vorteile:

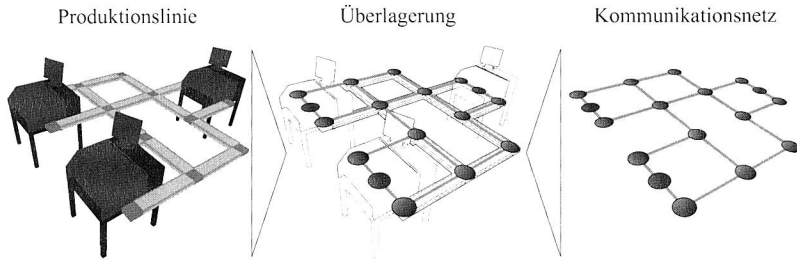


Bild 4.3: Isomorphismus zwischen Produktionslinie und kabelgebundenem Kommunikationsnetz - während die Rechner Bearbeitungsstationen oder Weichen repräsentieren, stellen die Kabelverbindungen die Streckenabschnitte dar. [86]

- Reduzierter manueller Konfigurationsaufwand der einzelnen Entitäten und der Gesamtanlage
- Vermeidung manueller Fehler bedingt durch den Einsatz geprüfter Automatismen
- Neben der Strukturerkennung gleichzeitige Nutzung der Netzkabel zur anlagenweiten Kommunikation
- Automatisierte Erkennung neuer oder entfernter Entitäten mit ihrer Type und ihrem Funktionsumfang

Bei mobilen Teilnehmer hingegen wird auf ein Funknetzwerk (WLAN) gesetzt, damit diese ortsunabhängig zu jedem Zeitpunkt mit anderen Teilnehmern interagieren können. Bei diesem kann zwar keine räumliche Struktur erkannt werden, jedoch können alle anderen Vorzüge, wie Anwesenheitserkennung und Ermittlung der Funktionalität, in gleichem Maße genutzt werden.

Die Strukturerkennung der Anlage wird unterteilt in die Anwesenheitserkennung der direkten Nachbarentitäten (Kapitel 4.3.1), die Verteilung der lokalen Sichten (Kapitel 4.3.2) und die Zusammenführung dieser lokalen Informationen, um daraus die Strukturen abzuleiten. All diese Verfahren laufen dezentral, ohne einen ausgezeichneten Master, ab. Die Ableitung der Anlagenstruktur und der Kommunikationsnetzwerkkonfiguration führt jede Entität selbständig nach einem deterministischen Verfahren durch (Kapitel 4.3.4). Mit diesen Ansätzen wird die Skalierbarkeit gewährleistet und die Gesamtausfallwahrscheinlichkeit reduziert, da es keinen Single-Point-of-Failure gibt.

### 4.3.1 Verfahren zur Anwesenheitserkennung

Das Verfahren zur Anwesenheitserkennung hat die Aufgabe die nächsten Nachbarn, die über ein Netzkabel mit der eigenen Entität verbunden sind, zu ermitteln. Hierbei sendet jede Entität mit einer vorgegebenen Frequenz Nachrichten über alle Netzwerkschnittstellen zu den benachbarten Entitäten. Die Nachricht enthält zunächst die eigene ID, die in der Gesamtanlage eindeutig ist. Hierzu können bekannte Algorithmen zur Generierung weltweit eindeutiger IDs genutzt werden [99]. Nach Empfang dieser Nachricht über eine Netzwerkschnittstelle ist der Empfänger-Entität bekannt, dass sich dort eine Nachbarentität befindet und welche ID diese trägt. Diese Informationen, zusammen mit dem

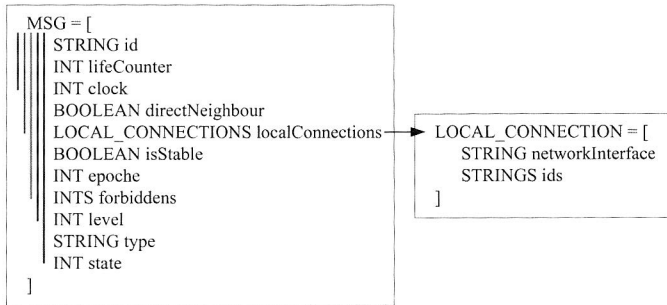


Bild 4.4: Datenstruktur des Mikronetzes.

Zeitstempel der letzten empfangenen Nachricht, werden lokal gespeichert und weiterführende Maßnahmen für die neu erkannte Entität eingeleitet (siehe Kapitel 4.3.2). Wird nach einem vorgegebenen Zeitintervall  $\Delta t$ , das größer sein muss als die Sendefrequenz, keine Nachricht über die Netzwerkschnittstelle empfangen, wird davon ausgegangen, dass die Nachbarentität nicht mehr existiert oder ausgefallen ist. Daraufhin wird wiederum der interne Speicher aktualisiert und es kann, analog zur Neuaufnahme der Entität, auch auf die weggefallene Entität reagiert werden. Das Ergebnis der Anwesenheitserkennung ist eine Liste, die zu jeder Schnittstelle die IDs der erkannten, benachbarten Entitäten enthält.

### 4.3.2 Weiterleitung der Nachbarschaftsdaten an alle Entitäten

Zur Verteilung der lokalen Informationen an alle Entitäten im Anlagenverbund, wird die Nachrichtenstruktur der Anwesenheitserkennung um die Mikronetz-Struktur erweitert. Hierzu ist primär die Liste *localConnections* erforderlich, die zu jeder Schnittstelle eine Liste von Identitäten direkt angeschlossener Entitäten speichert (Bild 4.4). Ist die Entität über eine Schnittstelle an einen unstrukturierten Bereich gekoppelt, werden die Identitäten aller Teilnehmer dieses Bereichs dort gespeichert. Ist eine Schnittstelle unbesetzt, so bleibt die Identitätenliste leer. Da die aktuellen Informationen zu dem Mikronetz zyklisch versendet werden, werden alle Entitäten zeitnah von Veränderungen in Kenntnis gesetzt. [86]

Dieses Mikronetz wird per Broadcast-Mechanismus nicht nur an die benachbarten Entitäten gesendet, sondern rekursiv von diesen wieder an die restlichen verbleibenden Nachbar-Entitäten bzw. Schnittstellen, außer über die Schnittstelle von der die Information angekommen ist, weitergeleitet. Um zu prüfen, ob eine Nachricht veraltet ist, wird ein logischer Zeitstempel [100] mitgesendet, der jedesmal beim Versand einer Nachricht vom Erzeuger der Nachricht um eins erhöht wird. Wird eine Nachricht mit einem kleineren Zeitstempel, als dem bislang höchsten, empfangen, wird sie ignoriert und auch nicht mehr weitergesendet. Somit können veraltete Nachrichten im System herausgefiltert und nur noch die neuen Informationen berücksichtigt werden. Ebenso wird das mehrmalige Eintreffen der gleichen aktuellen Nachricht, mit dem gleichen Zeitstempel erkannt, worauf diese ignoriert und nicht weitergeleitet wird. Dadurch können endlose Nachrichtenzyklen im System vermieden werden, die das Kommunikationsnetzwerk durch die Nachrichtenflut zum Erliegen bringen würden.



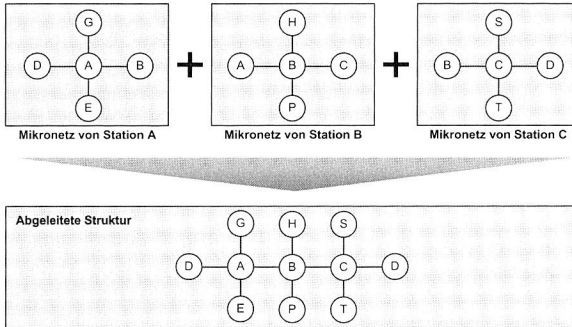


Bild 4.5: Beispiel zur dezentralen Ableitung des Gesamtnetzes durch Überlagerung der Mikronetze einer Linie bestehend aus drei sequentiell angeordneten Stationen. Die Mikronetze werden lokal an jeder Entität ermittelt und an alle Nachbarn versendet. Somit liegen an jeder Entität alle Mikronetze vor, die anhand der überlappenden Knoten verbunden werden müssen.

Der Versand der Mikronetze wird zugleich zur Anwesenheitserkennung der direkten Nachbarn verwendet (vgl. Kapitel 4.3.1). Kommt in einem bestimmten Zeitintervall keine Mikronetz-Nachricht an, wird die Entität als ausgefallen oder entfernt eingestuft. Damit die Entitäten unterscheiden können, ob die angekommene Mikronetz-Nachricht vom direkten Nachbarn generiert wurde oder ob es sich um eine weitergeleitete Nachricht handelt, wird deren Datenstruktur um ein Lokales-Nachbar-Flag (*isDirectNeighbour*) erweitert (Bild 4.4). Jede Entität versendet ihre eigenen Nachrichten stets mit gesetztem Flag - für alle weitergeleiteten Nachrichten wird das Flag wieder zurückgesetzt. Basierend auf diesem Flag kann die Empfänger-Entität ihr eigenes Mikronetz aktualisieren, sollte es sich um einen direkten Nachbarn handeln. [98]

### 4.3.3 Algorithmus zur Strukturermittlung

Nach der Ermittlung der lokalen Struktur und Verbreitung dieser Daten an jede Entität im System, besteht nun die Aufgabe darin, die Gesamtstruktur der Anlage anhand der gesammelten Daten abzuleiten. Da die Mikronetze nach deren Verteilung allen Entitäten vorliegen, kann die Strukturableitung deterministisch und eigenständig an jeder Entität vorgenommen werden. Diese Mikronetze müssen mit einem geeigneten Algorithmus überlagert werden (Bild 4.5). Ermöglicht wird dies durch den Einsatz von anlagenweit eindeutigen IDs für jeden Knoten im Mikronetz. Zwei Mikronetze überlagern sich immer an zwei Knoten und sind somit in der Ausrichtung zueinander eindeutig zuordnungsbar. Somit kann die Überlagerung der Mikronetze anhand der Knoten mit gleicher ID durchgeführt werden, woraus nach vollständiger Verarbeitung dieser die Gesamtstruktur der Anlage resultiert.

### 4.3.4 Automatisierte, verteilte Konfiguration der IP-Adressen und IP-Routen-Tabellen

Neben der automatisierten Ableitung der Anlagenstruktur können die bestehenden Verbindungen zwischen den Entitäten ebenso zur Kommunikation herangezogen werden. Somit ist kein weiteres Kom-

munikationsnetzwerk, wie Ethernet oder Feldbus, erforderlich, das z.B. im Falle von Ethernet üblicherweise mit erhöhtem Aufwand sternförmig von einem zentralen Switch oder Hub ausgehend verlegt wird. Voraussetzung hierfür ist die automatisierte Vergabe von IP-Adressen im verteilten Steuerungsumfeld und die automatisierte Generierung der IP-Routentabellen, da jede Entität als Router fungiert und somit die Datenpakete in Richtung des kürzesten Weges zum Ziel weiterleiten muss.

Die IP-Adressvergabe kann an dieser Stelle nicht mit bekannten Mechanismen, wie DHCP, vorgenommen werden, da es sich hierbei nicht um eine sternförmige Verkabelung handelt, bei der jede Entität direkten Zugang zum DHCP-Server besitzt, sondern jede Teilverbindung zwischen zwei Entitäten als eigenes Teilnetz anzusehen ist. Aus diesem Grund ist ein eigenes Verfahren erforderlich, das die automatisierte Vergabe der IP-Adressen in diesem verteilten Umfeld ermöglicht. Zudem soll auf einen Adressvergabe-Server verzichtet werden, um eine bessere Gesamtausfallsicherheit zu ermöglichen. Das hierzu entwickelte deterministische Verfahren sortiert die Liste der Entitäten nach der ID und vergibt im verfügbaren Teilnetz-Adressbereich die Adressen in aufsteigender Reihenfolge. Da alle Entitäten die gleiche globale Sicht auf die Produktionsanlage haben, kann somit jede Entität die IP-Adressen aller Entitäten im System bestimmen, wodurch die IP-Adressinformationen nicht zwingenderweise zwischen den Teilnehmern ausgetauscht werden müssen.

Die kommunikationstechnische Konfiguration der Entitäten kann nur dann erfolgreich sein, wenn alle Mikronetze (Kapitel 4.3.2) vollständig empfangen wurden. Dies kann der Strukturermittlungs-Algorithmus daran erkennen, dass zu allen vermerkten Nachbarn die zugehörigen Mikronetze existieren. Wenn jedoch eine Entität einen Nachbarknoten vermerkt hat, zu dem kein Mikronetz vorliegt, muss auf das Eintreffen der noch fehlenden Mikronetzdaten gewartet werden, bevor sich die Entität neu konfigurieren kann.

Nach der Festlegung der IP-Adressen müssen die IP-Routentabellen deklariert werden, da jede Entität im Netzwerkverbund als Router agiert und die Datenpakete in Richtung des kürzesten Weges zum Empfänger weiterleiten muss. Hierbei können keine Teilnetz-Bereiche eingetragen werden, da sich alle Rechner im gleichen IP-Teilnetz befinden und zudem mehrere Netzwerkschnittstellen mit IP-Adressen aus diesem Teilnetz konfiguriert sind. Daher ist es erforderlich nur Host-Einträge in die Routentabelle vorzunehmen, bei der u.a. die Netzwerkschnittstelle anzugeben ist, über welche die Datenpakete weitergeleitet werden müssen. Im Falle eines Entitätenausfalls reagiert die Anwesenheits- und Strukturerkennung unmittelbar und veranlasst hierbei die Neukonfiguration der Routentabellen, um fehlende Entitäten aus der Tabelle zu entfernen und alternative Kommunikationswege zu den bestehenden neu zu definieren. Die Anwendungen bekommen keine Notiz von diesen Veränderungen, da die Datenpakete transparent über die Alternativroute weitergeleitet werden. Ungesicherte Protokolle, wie UDP, haben den Nachteil, dass die erfolgreiche Übermittlung von Daten nicht geprüft wird, jedoch ist dieses Protokoll sehr schnell und effizient in der Verarbeitung. Hierbei können jedoch Pakete bis zur Erkennung der ausgefallenen Entität und während der Neukonfiguration verloren gehen. Der Entwickler muss sich bewusst sein, dass ein Paketverlust in diesen Fällen mit einer höheren Wahrscheinlichkeit auftreten kann. Entweder ist dies für die Anwendung nicht von Bedeutung oder sie kann selbständig auf verloren gegangene Pakete reagieren. Aus diesem Grund sind streamorientierte Datenkanäle, wie TCP, zu empfehlen, die eine sichere Datenübermittlung gewährleisten und dabei verlorene Pakete selbständig erneut versenden.

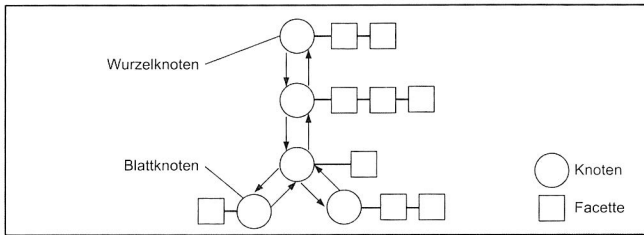


Bild 4.6: Struktur des Steuerungsbaumes, in dem jeder Knoten Referenzen auf einen Vaterknoten, seine Kindknoten und die Facetten hält. Die Facetten selbst verweisen wiederum auf den ihnen zugeordneten Knoten.

## 4.4 Wandelbare Softwarearchitektur für verteilte Steuerungssysteme

In der systemtheoretischen Analyse (Kapitel 4.2) wurde gezeigt, dass die Systemstruktur von Anlagensystemen einer Baumstruktur gleicht. Diese Struktur erstreckt sich zum einen zwischen den verschiedenen Entitäten, zum anderen ist die hierarchische Struktur der Steuerungssysteme auch innerhalb jeder Entität vorzufinden. Um diese Strukturen aufbauen zu können, werden im Folgenden Modellierungs-Elemente vorgestellt, die es erlauben, Steuerungsbäume mittels einer Spezifikation zu deklarieren und mittels standardisierter Softwarekomponenten zu implementieren. Dieses Modell unterteilt sich zum einen in die Spezifikation der Baumstruktur (Kapitel 4.4.1) und zum anderen in die Festlegung der Funktionalität (Kapitel 4.4.2).

### 4.4.1 Knoten zum strukturellen Aufbau

Ein Knoten entspricht einem Datencontainer, der verschiedene Informationen in Form einer Datenstruktur speichert und zugleich Methoden anbietet, um auf dem Baum zu navigieren. Zum einen hält er Referenzen auf seinen Vater- und seine Kindknoten - zum anderen hält er die Funktionsbausteine, die sog. Facetten (Kapitel 4.4.2), die zur Steuerung der zugeordneten physischen Komponente erforderlich sind. Alle Knoten in einer Produktionsanlage bilden miteinander durch ihre hierarchische Verknüpfung eine Baumstruktur, die das Gerüst des gesamten Softwareverbunds der Produktionsanlage darstellt. Jeder Knoten ist von allen anderen über das Adressierungsschema (Kapitel 4.2) erreichbar, wodurch die geforderte informationelle Vernetzung gegeben ist.

Der strukturelle Aufbau des Knotens ist in Bild 4.6 am Beispiel eines Steuerungsbaumes dargestellt. Zunächst benötigt jeder Knoten im Umfeld mit den Geschwisterknoten (zeichnen sich durch den gemeinsamen Vater aus) einen eindeutigen Namen (*name*). Der Locator (*locator*) enthält die weltweit eindeutige Adresse gemäß des vorgestellten Adressierungsschemas (siehe Kapitel 4.2). Die weltweit eindeutige Identität (*ID*) ermöglicht die einfachere Handhabung und Identifizierung der Knoten. Die Auflösung nach der ID erfolgt in der Regel über das Adressierungsschema und wird in der Steuerungsarchitektur u.a. zur internen Referenzierung verwendet.

Die Baumstruktur wird über Referenzen zu seinem Vater- (*parent*) und seinen Kindknoten (*children*) beschrieben. Da der Knoten Kind seines Vaters und gleichzeitig Vater seiner Kinder ist, halten sein Va-

```

FACET = [
  Attributes = [
    STRING name
    STRING locator
    STRING id
    NODE host

    LIST_FACETS facets
    LIST_CHILDREN nodes
    LIST_PUBLISHERS publishers
    Callbacks = [
    ]
  ]
  GetInfo(): FACET_INFO
]

```

Bild 4.7: Grundlegende Struktur der Basis-Facette.

ter und seine Kinder wiederum Referenzen auf ihn. Dadurch entstehen bidirektionale Referenzen, die eine auf- wie abwärts gerichtete Navigation im Baum ermöglichen. Analog halten sowohl die Knoten Referenzen auf die Facetten als auch umgekehrt diese auf ihre Knoten. Somit wird die Navigation in beide Richtungen ermöglicht, d.h. einerseits von außen Zugriff auf eine Facette über den Baum und andererseits von der Facette selbst auf den Baum und darüber ebenso auf alle anderen Facetten.

#### 4.4.2 Facette als Träger der Funktionalität

Im Systemkonzept für wandelbare Produktionsanlagen übernimmt die Facette die Rolle als Träger elementarer Funktionalität. Die Bezeichnung resultiert aus der Eigenschaft, dass mehrere Funktions-Basisbausteine in einem Container zusammengefasst sein können, so dass dieser, je nach Nutzung eine andere Seite, sprich Facette, offenbart. Diese Funktionalität ist zwar logisch dem physischen Gerät zugeordnet, stellt jedoch, motiviert aus der hohen zu erzielenden Wiederverwendbarkeit, eine Abstraktionsebene dar, die hersteller- oder gerätespezifische Ausprägungen des physischen Gerätes verbirgt. Voraussetzung hierfür ist die Einhaltung bestimmter Normschnittstellen, die für eine bestimmte Gerätefunktionalität bzw. gleiche Gerätetypen einheitliche Schnittstellen anbieten und somit die Austauschbarkeit inkl. ihrer Funktionalität gewährleisten. Zudem können Facetten auf weiteren Facettenbausteinen aufbauen, wodurch der modulare und komponentenbasierte Ansatz der Softwareentwicklung umgesetzt wird.

Der strukturelle Aufbau der Facette ist in Bild 4.7 dargestellt. Als Attribut enthält die Facette einen Namen (*name*), einen Locator (*locator*) zur eindeutigen Referenzierbarkeit und eine weltweit eindeutige Identität (*id*). Über das Attribut *host* hält sie eine Referenz auf den zugeordneten Knoten (Kapitel 4.4.1), der Navigations- und Verwaltungsfunktionen anbietet. Dazu gehören insbesondere die Methoden *GetNode(locator)* und *GetFacet(locator)*, die Referenzen auf einen Knoten oder eine Facette liefern. Jede Facette besitzt die Möglichkeit nicht nur mit lokalen, sondern auch mit anderen Facetten auf entfernten Entitäten gleichermaßen und transparent zu interagieren, d.h. ohne sich um die Netzwerkkommunikation kümmern zu müssen. Statische Referenzen, die unbedingt zur Ausführung der Facette erforderlich sind, werden beim Start der Anwendung ermittelt und in den lokalen Listen

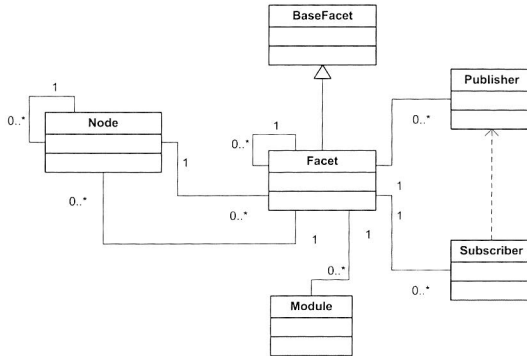


Bild 4.8: UML-Modell der grundlegenden Klassen inkl. ihrer Relationen - die Facette als Träger der Funktionalität besitzt Zugriff auf Knoten, Publisher, Subscriber und Module. Ein Subscriber ist genau einem Publisher zugeordnet, während sich mehrere Subscriber für einen Publisher registrieren können. Pro Publisher gibt es an einem Rechner nur einen Subscriber, der das eingetroffene Event an die registrierten Facetten weiterleitet.

*facets* und *nodes* der Facette gespeichert. Diese stehen somit der Facette während der Laufzeit zur Verfügung. Des weiteren besteht die Möglichkeit, dass jede Facette zur Laufzeit dynamisch Referenzen zu Knoten oder Facetten abrufen. Dies kann in unbekannten Umgebungen der Fall sein, in denen z.B. die Zahl der Entitäten mit denen die Facette interagieren muss, erst zur Laufzeit ermittelt werden bzw. sich zur Laufzeit auch verändern kann.

Eine weitere Form der Interaktion zwischen Facetten stellt der Ereignismechanismus dar. Dabei kann eine Facette Publisher anbieten, um andere Facetten über Ereignisse zu informieren. Alle Publisher werden in einer weiteren Liste (*publishers*) gespeichert, damit aus dem Facetten-Code heraus an den bereits vorinitialisierten Publishern ein Notify, im Falle des Ereigniseintritts, aufgerufen werden kann. Umgekehrt hält jede Facette Subscriber, die sich bei den Publishern der Gegenseite registrieren und sich zusammen mit der Subscriberverwaltung um die Zustellung der Ereignisse zu den Interessenten kümmern. Für jede Ereignisart muss die Facette über eine passende Callbackprozedur verfügen, die vom Subscriber bei Ereigniseintritt aufgerufen wird.

Alle Facetten sind von einer Basis-Facette abgeleitet, die verwaltungstechnische Aufgaben, wie die Beschaffung von Referenzen auf andere Facetten oder Knoten und diverse interne Funktionen zur Anbindung an den Steuerungsbaum auf die Kinder-Facetten vererbt (Bild 4.8).

Um die Funktionalität komplexer Facetten überschaubar darzustellen, kann zur Wahrung der Übersichtlichkeit und Gewährleistung der komponentenbasierten Softwareentwicklung in abgeschlossene Funktionsmodule eine Auslagerung der Implementierung in mehrere Module von Vorteil sein. Module sind rein interne Strukturen der Facette und treten gegenüber anderen Facetten oder Knoten nicht in Erscheinung. Als Schnittstelle nach außen dienen nur die Facettenschnittstellen.



lokalen Entität ab. Zudem werden alle Interaktionsformen, wie Request-Reply, Messaging und Ereignismechanismen, über das Netzwerk unterstützt.

- **Includ-O-Matic:** Um Geräte in die Steuerungsarchitektur einzubinden, wird eine Abstraktionsschicht eingeführt, die von den herstellerspezifischen Schnittstellenimplementierungen abstrahieren soll. Hierüber wird der Zugriff auf gleichartige Geräte mit standardisierten Schnittstellen vereinheitlicht, um die Modularität auch auf Geräteebene bewerkstelligen zu können.
- **Net-O-Matic:** Implementiert den in Kapitel 4.3 vorgestellten Algorithmus zur Strukturerkennung. Zudem steht eine Erweiterung zur Verfügung, die eine Entität mit allen erforderlichen Kommunikationsparametern, wie IP-Adresse und Routentabellen, konfiguriert.

#### 4.5.1 Net-O-Matic zur Nachbarschaftserkennung und Selbstkonfiguration

Mit Net-O-Matic sind die Entitäten zunächst in der Lage, die Struktur der Anlage zu erkennen (*NetStructure*) und die Konfiguration des Kommunikationsnetzwerks vorzunehmen (*NetConfig*). Dabei ist die Konfiguration in eine eigene Anwendung ausgelagert, um auch in statischen Umgebungen, in denen die Kommunikationsparameter nicht verändert werden sollen, lediglich die Strukturerkennung zur Ausführung bringen zu können. Die Anwendungen sind in Python [101, 102] realisiert, wodurch eine Portierbarkeit auf beliebige Betriebssystem- und Hardwareplattformen gewährleistet werden kann. Die derzeit verfügbare Implementierung läuft sowohl unter Linux [103] als auch unter Windows [103] gleichermaßen und wurde ebenso unter verschiedenen Windows-Versionen getestet.

In der Teilanwendung *NetStructure* ist der Algorithmus zur Nachbarschaftserkennung und zur Ableitung der globalen Anlagenstruktur aus der Summe der lokalen Sichten implementiert. Auf jeder Entität wird ein Sende-Thread ausgeführt, der mit einer vorgegebenen Frequenz Informationen zu den lokalen Nachbarn über alle Netzwerkschnittstellen verteilt. Diese Nachrichten werden von den benachbarten Entitäten über Empfänger-Threads entgegengenommen. Um die Richtung zu ermitteln, aus der die Nachricht einging, wird für jede Schnittstelle ein eigener Empfänger-Thread initiiert.

Jeder Empfänger-Thread wertet die Nachrichten aus und ermittelt, ob eine Strukturänderung vorliegt, indem der Inhalt der alten, hinterlegten Nachricht mit dem der aktuellen verglichen wird. Nur in diesem Falle wird die neue Nachricht mit den neuen Informationen über alle Netzwerkschnittstellen, außer der Empfangs-Schnittstelle, weitergeleitet. Zudem wechselt in diesem Fall *NetStructure* in den Zustand *instabile Phase*. Werden über einen freikonfigurierbaren Zeitraum keine Änderungen mehr detektiert, wechselt das Programm zurück in den Zustand *stabile Phase* [86]. Dabei wird sowohl die eigene als auch für alle Entitäten im Netz die IP-Adresse berechnet. Die neue Anlagenstruktur mit den aktuellen IP-Adressen wird der *NetConfig*-Anwendung übergeben (Bild 4.10). Die Konfiguration sollte erst dann vorgenommen werden, wenn die Kommunikationsmiddleware, repräsentiert durch den *ComManager*, heruntergefahren wurde. Hierzu wird der *ComManager* über die Änderungen an der aktuellen Anlagenstruktur informiert, terminiert vorübergehend die Kommunikationsverbindungen und signalisiert die Bereitschaft zur Umkonfiguration an *NetConfig*. Letzteres übernimmt die Werte von *NetStructure* und nimmt die Konfiguration der IP-Adressen an den lokalen Netzwerkschnittstellen und die Änderung der Routentabelle vor. Nach erfolgreicher Konfiguration wird wiederum dem *ComManager* signalisiert, dass das Kommunikationsnetzwerk bereit und wieder einsatzfähig ist, worauf dieser die Verbindungen neu aufbaut.

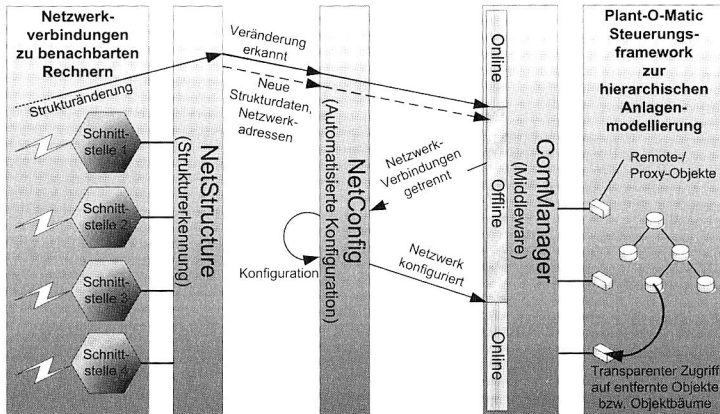


Bild 4.10: Darstellung der strukturellen Beziehungen und Abläufe während einer Umkonfigurationsphase. Net-Structure kommuniziert über die Netzwerkschnittstellen direkt mit den Nachbarn und signalisiert Veränderungen an NetConfig und den ComManager. Letzterer muss seine Kommunikationsverbindungen trennen bevor die IP-Konfiguration (IP-Adresse und IP-Routing) von NetConfig geändert werden kann.

#### 4.5.2 Plant-O-Matic - Referenzimplementierung der Softwarearchitektur

Zur Umsetzung der hierarchischen Steuerungsansätze aus Kapitel 4.2 ist eine eigenständige Softwareplattform erforderlich, die entsprechende Softwaremodule in Bibliotheken bzw. Baukästen und Softwarewerkzeuge für folgende wiederkehrende Aufgaben zur Verfügung stellt:

- Spezifikation der Facetten-Schnittstellen
- Ableitung eines Code-Grundgerüsts aus den Schnittstellenspezifikationen
- Implementierung der Facetten mit ihrem spezifischen Verhalten
- Generierung von Stellvertretern (Proxies) für den transparenten Zugriff auf alle Facetten über das Netzwerk
- Festlegung des anwendungsspezifischen Baumes mit Knoten als Verbindungselemente und Facetten zur Beschreibung der Funktionalität
- Generierung des ablauffähigen Programmes gemäß der Baumspezifikation
- Standardkomponenten bzw. -facetten in Softwarebibliotheken

Neben der Entwicklung muss die Softwarearchitektur zahlreiche Funktionen zur Verfügung stellen:

- Verknüpfung aller Entitäten zu einem globalen Steuerungsbaum



- Adressierung aller Knoten und Entitäten
- Verschiedene Interaktionsformen mit den Facetten (Request-Reply, Messaging, Ereignisse)

In all diesen Bereichen ist *Plant-O-Matic* durchgängig auf Wandelbarkeit ausgelegt, um die besonderen Herausforderungen, wie Anlagenkomponenten selbst zur Laufzeit umbauen oder neue hinzufügen zu können, zu bewerkstelligen.

### Standardfacetten

Die in den Kapiteln 4.4.1 und 4.4.2 vorgestellten Modelle stehen in dieser Architektur als eigenständige Klassen zur Verfügung. Neben der Basisimplementierung der Facetten werden zudem weiterführende Standardfacetten, z.B. für Standardgeräteschnittstellen, zur Verfügung gestellt. Eine Facette repräsentiert i.d.R. nur ein Gerät bzw. einen Ein- oder Ausgang. Sollen mehrere Geräte oder Ein- oder Ausgänge angesprochen werden, so muss dieselbe Facette mehrfach in den Steuerungsbaum eingebunden und die Facetten über die eindeutige hierarchische Adresse unterschieden werden. Hierbei ist es von Vorteil, die Namen der Facetten gemäß ihrer Funktion zu wählen, um dabei die weiterführende Entwicklung der Steuerungssoftware funktions- und aufgabenbezogen auszurichten. Die wichtigsten Vertreter der Standardfacetten sind im Folgenden aufgeführt [98]:

- *FacetDigitalIn / FacetDigitalOut* (Facetten für digitalen Ein- und Ausgang): Eine Instanz der Facette repräsentiert genau einen digitalen Ein- oder Ausgang. Neben der Möglichkeit digitale Zustände zu setzen und abzufragen, stehen ebenso Ereignisse zur Verfügung, die im Falle steigender Flanken, fallender Flanken oder Zustandswechsel informieren.
- *FacetAnalogIn / FacetAnalogOut* (Facetten für analogen Ein- und Ausgang): Eine Instanz der Facette repräsentiert genau einen analogen Ein- oder Ausgang. Die Analogwerte werden auf einen Float-Zahlenbereich von 0 bis 1 normiert und abstrahieren somit vom konkreten Wertebereich, den ein Eingang abtasten oder ein Ausgang setzen kann. Dies ist ein Beitrag zur Erhöhung der Wandelbarkeit, da vom konkreten Gerät abstrahiert und sowohl einheitenunabhängig, als auch auflösungsunabhängig Werte ausgetauscht werden.
- *FacetStringIn / FacetStringOut*: Diese Facetten repräsentieren zeichenkettenbasierte Schnittstellen, wie serielle Schnittstellen, TCP/IP- oder UDP/IP-basierte Socket-Verbindungen oder Bussysteme.
- *FacetScriptableController*: Diese Facette stellt die Möglichkeit zur Verfügung, anwendungsspezifische Skripte auszuführen. Diese umfassen Anweisungen, die den Ablauf eines Prozesses oder Teile der Produktionsanlage steuern und dabei auf den globalen Steuerungsbaum bzw. die darin integrierten Facetten zugreifen können.

### Facettenspezifikationen

Ausgangspunkt für die Entwicklung neuer Facetten ist die Erstellung von Facetten-Spezifikationen, aus denen ein allgemein gültiger und verfügbarer Generator in der Plant-O-Matic-Plattform ein Grundgerüst der Facetten in der Zielprogrammiersprache, hier Python, erzeugen kann. In diesen Spezifikationen werden folgende Elemente beschrieben (Bild 4.11):

- Superklasse (*INHERITS\_FROM*): Aus der Basisklasse, von der die Spezifikation abgeleitet ist, werden alle Konfigurationen übernommen.
- Module (*MODULES*): Eine Facette kann, wie bereits in Kapitel 4.4.2 beschrieben, aus mehreren Modulen aufgebaut sein, bzw. sich der Standardfunktionalität aus allgemeingültigen Modulen bedienen. Für die Erstellung des Code-Grundgerüsts werden alle erforderlichen Module angegeben.
- Attribute (*ATTRIBUTES*): Alle internen Attribute werden aufgeführt, um diese durch einen Generator automatisiert in einem Code-Grundgerüst vorzugeben. Die Typen der Variablen können sowohl primitive Datentypen darstellen, als auch individuelle Datenstrukturen, die in weiteren Spezifikationen deklariert werden müssen.
- Knoten (*NODES*): Besteht die Notwendigkeit, initial Zugriff auf feste und bestimmte Knoten zu haben, werden hier alle notwendigen Knoten anhand ihres Locators angegeben.
- Facetten (*FACETS*): Hier werden die Facetten mit ihrem Typ erfasst, auf die eine Referenz erforderlich ist.
- Publisher (*PUBLISHER*): Auf alle zur Verfügung gestellten Publisher, die über das Eintreffen von Ereignissen informieren, werden lokale Referenzen gehalten.
- Subscriber (*SUBSCRIPTIONS*): Interessiert sich eine Facette für ein Ereignis, angeboten durch einen Publisher einer anderen Facette, wird ein Subscriber-Objekt vorgesehen, das eine Verbindung zum Publisher-Objekt aufbaut, alle Verwaltungs- und Registrierungsaufgaben übernimmt und über den Eintritt des Ereignisses Rückmeldung an die lokale Facette gibt. Zudem lassen sich mit dieser Methodik mehrmalige Registrierungen für das gleiche Ereignis bündeln, da pro registriertes Ereignis an einem Publisher maximal ein lokaler Subscriber existiert, der auch mehrere lokale Facetten informieren kann.
- Methoden (*METHODS*): Die Methoden stellen die Schnittstelle der Facette nach außen dar. Zusätzlich wird angegeben, ob eine Methode nur in der lokalen Entität oder auch über das Netzwerk erreichbar ist. Eine Methode wird über den lokal eindeutigen Namen identifiziert und über die Aufrufparameter spezifiziert.

## Code-Generierung

Nach der Spezifikation der Facetten, Datentypen, Exceptions, Publisher und Subscriber kann mit Hilfe von Code-Generatoren aus diesen Spezifikationen automatisiert Programm-Code, außer dem Facetten-Verhalten, abgeleitet werden (Bild 4.12). Bei den Facetten wird ein Rahmengerüst erzeugt, das noch vom Entwickler mit Programmcode gefüllt werden muss. Diese Facetten stehen somit als Softwarebausteine zur Verfügung, die in beliebigen Anwendungen über den Steuerungsbaum verknüpft und modular eingesetzt werden können. Hierzu muss für die jeweilige Anwendung der Steuerungsbaum einer jeden Entität spezifiziert werden. Dabei wird zum einen die Struktur des Baumes mit den Knoten und den zugeordneten Facetten festgelegt, zum anderen wird ein Teil des Verhaltens der Steuerungsanwendung durch die logische Verknüpfung der Facetten untereinander beschrieben. Die starren Verknüpfungen, die in den Facetten-Spezifikationen hinterlegt sind, können fest in den Programmcode

```

BASE_FACET = OBJECT(util.spec_defs.SpecFacet.Object) [
  INHERITS_FROM = STRING: SPEC base_spec
  MODULES = LIST [
    STRING: SPEC module_spec
  ]
  INIT_ATTRIBUTES = DICT [
    param = STRING: SPEC datatype_spec
  ]
  ATTRIBUTES = DICT [
    attr = STRING: datatype_spec
  ]
  NODES = DICT [
    node = STRING: SPEC node_spec
  ]
  FACETS = DICT [
    facet = STRING: SPEC facet_spec
  ]
  PUBLISHER = DICT [
    publisher = STRING: SPEC publisher_spec
  ]
  SUBSCRIPTIONS = DICT [
    subscr = STRING: SPEC subscr_spec
  ]
  METHODS = DICT [
    PING = DICT [
      EXPORT = STRING: YES
      PARAMETERS = DICT [
        param = STRING: SPEC param_spec
      ]
      RETURN_TYPE = STRING: datatype_spec
      THROWS = LIST [
        excp = STRING: SPEC excp_spec
      ]
    ]
  ]
]

```

Bild 4.11: Die *BASE\_FACET* gibt die Meta-Deklaration von Facetten vor. Sie dient als Schema für den Parser, der damit die Konsistenz der Facetten-Spezifikationen prüfen kann. Die terminalen Symbole sind groß, die Variablen klein geschrieben.

aufgenommen und vorgesehene Variablen vorinitialisiert werden. Dies betrifft die Verbindungen zu anderen Facetten *FACETS*, Knoten *NODES* und die Registrierung der Subscriber *SUBSCRIPTIONS* bei den Publishern *PUBLISHER*, die mit eindeutigen Adressen, dem sog. *Locator*, identifiziert werden können. Dabei wird zudem der in der Facettenspezifikation angegebene, erforderliche Typ mit den Typen der zugeordneten Instanzen verglichen und auf Konsistenz geprüft.

## Transparente Adressierung

Aus der Spezifikation des lokalen Baumes für jede Entität ergeben sich infolgedessen lokale Adressierungsschemata, die ausgehend vom lokalen Wurzelknoten einer Entität jeden Knoten und jede Facette referenzieren können [87]. Erst zur Laufzeit werden diese lokalen Steuerungsbäume zu einem globalen Baum zusammengefasst, der sich über die gesamte Produktion erstreckt. Somit sind Knoten, Facetten und Publisher, inkl. deren Methoden über einen global eindeutigen Locator selbst ortstransparent, d.h. unabhängig vom Ausführungsort, über das Netzwerk erreichbar. Die Adressen werden für den Entwickler transparent in Referenzen auf lokale oder entfernte Objekte aufgelöst, so dass zwischen der lokalen und rechnergrenzenüberschreitenden Interaktion weitgehend kein Unterschied besteht. Somit trägt diese Funktionalität zu einer Reduktion der Entwicklungszeiten bei.

Die vereinfachte und transparente Interaktion zwischen Entitäten wird durch den *ComManager* realisiert, der auf jeder Entität ausgeführt wird. Er übernimmt alle Kommunikationsaufgaben, die über das Netzwerk mit anderen Entitäten erfolgen. Für den Entwickler werden transparente Kommunikations-

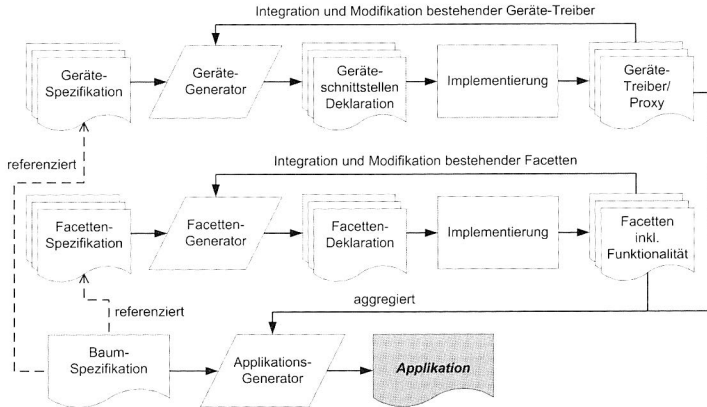


Bild 4.12: Aus den Geräte- und Facettenspezifikationen resultieren Code-Grundgerüste, die alle erforderlichen Deklarationen als Voraussetzung für die Implementierung enthalten. Die Baum-Spezifikation referenziert die Baum-Struktur mit den zugehörigen Facetten, um daraus die ausführbare Applikation abzuleiten.

verbindungen zur Verfügung gestellt, die sich u.a. nach Wandlungen an neue Situationen automatisch anpassen. Der Zugriff auf entfernte Komponenten erfolgt über das Adressierungsschema, wobei bei der Adressauflösung nicht-lokale Adressen erkannt und für die entfernten Objekte sogenannte Stellvertreter (Proxies) durch den *ComManager* zur Laufzeit erzeugt werden. Diese leiten lokal erzeugte, für den Entwickler transparente Aufrufe, zum Empfänger weiter. Der Entwickler erhält somit über die Proxies Zugriff auf entfernte Objekte, als ob es sich dabei um lokale Objekte handeln würde.

### 4.5.3 Includ-O-Matic - Integration peripherer Geräte

Das Softwarepaket *Includ-O-Matic* bietet dem Entwickler Hilfswerkzeuge an, Peripherie möglichst aufwandsarm in die Steuerungsarchitektur einzugliedern. Zudem leistet diese Anbindung über standardisierte, herstellerunabhängige Geräteschnittstellen einen maßgeblichen Beitrag zur Modularisierung der Anlagensysteme. Die Notwendigkeit Hilfswerkzeuge und Entwicklungsketten zur Gewährleistung dieser Funktionalität entwickeln zu müssen, zeigt, dass kein etablierter, einheitlicher Standard existiert. Je nach Hersteller und teilweise je nach Typ erfolgt die Anbindung über unterschiedliche Hardwareschnittstellen, wie z.B. USB, Seriellport, Parallelport, Ethernet oder Steckkarten sowie proprietäre Softwareschnittstellen und Protokolle. Trotz vielfacher Standardisierungsbemühungen, entwickeln Geräteanbieter dennoch ihre eigenen, proprietären Treiber, woraus insgesamt enorme systemintegratorische Entwicklungsaufwendungen resultieren. *Includ-O-Matic* stellt eine Abstraktionsschicht zur Verfügung, um für Entwicklungen unter *Plant-O-Matic* einen standardisierten Zugriff auf die Peripherie zu ermöglichen und somit die einfache Austauschbarkeit der Geräte auf Seiten der Steuerung zu gewährleisten. Dies wird erreicht, indem ideale Gerätemodelle und zugehörige Standardschnittstellen spezifiziert werden. Im *Includ-O-Matic*-Treiber werden diese Schnittstellen nach unten zu den bestehenden Treibern mit ihren proprietären Schnittstellen übersetzt, wodurch die Einbindung beste-

hender peripherer Geräte gewährleistet werden kann.

Der Zugriff auf alle Geräte innerhalb einer Entität wird von einer zentralen Peripherieverwaltung gewährt. Einer Schnittstelle im System, z.B. einem Digitalen Ausgang oder Eingang, wird ein eigenes Objekt zugeordnet. Die Verwaltung hat die Aufgabe, für jedes Schnittstellenobjekt den exklusiven Zugriff nur für einen Nutzer zu gewähren, um konkurrierende und dadurch kritische Mehrfachzugriffe zu unterbinden. Ein Nutzer kann ein Schnittstellenobjekt jedoch wieder freigeben, um dieses einem anderen Nutzer zur Verfügung zu stellen.

Da dennoch herstellerspezifische Treiber gekapselt werden müssen, ist eine eindeutige Adressierung der Geräte bzw. deren elektrischer Anschlüsse an der Schnittstellenkarte erforderlich. Jedes Schnittstellenobjekt ist durch ein Tripel, bestehend aus Treiber-Id, SuperDevice-Id und Device-Id, eindeutig in der Entität identifizierbar. Dieses Schema folgt einer dreistufigen Hierarchie:

1. Treiber-Id: Identifiziert den herstellerspezifischen Treiber.
2. SuperDevice-Id: Zur eindeutigen Unterscheidung mehrerer gleichartiger Geräte z.B. desselben Herstellers und derselben Type.
3. Device-Id: Identifizierung der eigentlichen Schnittstelle, z.B. des analogen Ein- oder Ausganges.

Auf Seiten von Plant-O-Matic wird jedes Schnittstellenobjekt, das eine Hardwareschnittstelle repräsentiert, in einer Schnittstellenfacette gekapselt. Somit existieren für alle physischen Schnittstellen im System die gleiche Anzahl an Facetten-Instanzen im Steuerungsbaum. Mit diesem Steuerungsmodell wird gewährleistet, dass Plant-O-Matic flexibel und aufwandsarm an unterschiedliche Hardwarekomponenten gekoppelt werden kann.

## **4.6 Graphische Entwicklungsumgebung (GIDE) zur Ablaufspezifikation**

Die vorliegende Steuerungsarchitektur besitzt einen Skript-Controller (siehe Kapitel 4.5.2), der anwendungsspezifisch Skripte z.B. zur Prozesssteuerung ausführen kann. Diese können zur Laufzeit geladen werden und besitzen analog zu den Facetten Zugriff auf den gesamten Steuerungsbaum. Die Erstellung der Skripte erfolgt textuell in einer Skriptsprache, die von Python abgeleitet ist und um diverse Befehle bzw. Funktionen erweitert wurde. Um den Entwickler in der vereinfachten und schnellen Umsetzung der Skripte zu unterstützen, wird im Folgenden eine graphische Programmiersprache vorgestellt, die von den Implementierungsdetails abstrahiert und dem Entwickler erlaubt, die Abläufe auf einer hohen Abstraktionsebene, d.h. nahe den grundlegenden Abläufen in der Anwendung, zu spezifizieren.

### **Anforderungen an die Programmierung der Ablaufsteuerung**

Die Ablaufsteuerung in der Produktion ist geprägt von koordinierenden Aufgaben zwischen verschiedenen Komponenten. Dies beinhaltet die Festlegung der sequentiellen Schrittfolge durchzuführender Vorgänge, die in eine Ordnungsrelation gebracht werden müssen. Zudem muss die Möglichkeit bestehen, diese Sequenz zur Laufzeit zu verändern, d.h. anhand von Eingangswerten den weiteren

Programmfortgang zu beeinflussen. Auf Grund der zunehmend komplexen Vorgänge ist es für den Entwickler wichtig, einen guten Überblick über die Abläufe zu erlangen, um diese mit geringem Einarbeitungsaufwand verstehen zu können. Aus dieser Motivation heraus wurde im vorliegenden Ansatz eine graphische Programmierungsumgebung mit einem erweiterten Programmierformalismus entwickelt, die auf einer hohen Abstraktionsebene die Spezifikation der Abläufe ermöglicht und dabei die Implementierungsdetails vor dem Anwender verbirgt. Diese Details umfassen insbesondere Wissen über den syntaktischen Overhead, der in jeder textuellen Programmiersprache vorliegt, Kenntnisse der Schnittstellen, die Konvertierung der Parameterwerte in ein geeignetes Format u.v.a.m.

Dem Entwickler steht ein Formalismus zur Verfügung, der die Eingabe der Abläufe möglichst analog zur Denkweise des Entwicklers im Anwendungsumfeld ermöglicht, indem u.a. reale Objekte, wie die Geräte oder die zu bearbeitenden Werkstücke, in der Entwicklungsumgebung identisch vorkommen und ebenso als solche gehandhabt werden können.

### **Formalismus zur impliziten Deklaration der Programmanweisungen**

Der Erstellungsprozess von Ablaufprogrammen für Gerätesteuerungen zeichnet sich in den meisten Fällen durch die Aneinanderreihung von Anweisungen für sequentielle oder parallelisierte Aktionen aus. Die erforderliche Menge an Anweisungen kann in den meisten Fällen sehr klein gehalten werden - die Variantenvielfalt in der Ausführung ist maßgeblich durch die Anzahl und Kardinalität der Parameter bestimmt.

Zunächst besteht die Aufgabe des Entwicklers darin, alle Zustände der Anlage, die bei Durchführung z.B. eines Prozesses unterscheidbar sind, zu identifizieren und diese zueinander in Beziehung zu bringen. Für diesen Schritt ist die Spezifikation von Programmmodulen möglich, wobei ein Modul wiederum aus Modulen aufgebaut sein kann.

In der nächsten Entwicklungsphase müssen die Module mit Programmanweisungen gefüllt werden, um die Einzelschritte zu spezifizieren. Der Informationsgehalt der in einer Anweisung festgelegt wird, kann mit folgenden Fragen erfasst werden:

1. Wer soll die Aufgabe durchführen?
2. Was muss durchgeführt werden, um das Ziel zu erreichen?
3. Welche Randbedingungen, d.h. Parameter, sind erforderlich?

Durch Beantwortung dieser Fragen in der angegebenen Reihenfolge wird eine Anweisung vollständig spezifiziert. Diese Methodik ist einheitlich in der Entwicklungsumgebung GIDE bei der Festlegung jeglicher Anweisungen vorzufinden. Grundsätzlich werden analog zu dieser Methodik drei unterschiedliche Symboltypen unterschieden (Bild 4.13):

1. Jedes Gerät, das sowohl aktiv angesteuert werden kann, als auch nur passiv genutzt wird, besitzt eine eigene Repräsentation in GIDE in Form eines stellvertretenden Symbols, den *aktiven Objekten*. Mit den aktiv steuerbaren Geräten wird spezifiziert, wer die Aktion ausführt.
2. Die Menge verfügbarer Befehle wird in Form eigener Symbole vorgegeben, um dem Entwickler die Menge der Möglichkeiten in einer Liste übersichtlich darstellen zu können. Hierbei wählt

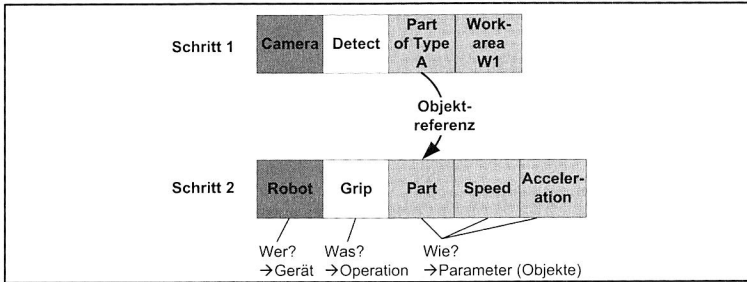


Bild 4.13: Kurzes Programmbeispiel in GIDE, in dem zunächst ein Bauteil der bestimmten Type A mit einer Kamera auf einer Arbeitsfläche W1 gefunden werden soll. Ist das Bauteil identifiziert, wird dessen Position im Bauteil-Objekt *Part* hinterlegt und liegt als Objektreferenz beim Greif-Befehl vor. Die weiteren Parameter zur Geschwindigkeit und Beschleunigung sind an dieser Stelle optional.

der Entwickler einen Befehl passend zum Gerät aus und setzt ihn mit dem Gerät zur Anweisung zusammen.

3. Zur weiteren Spezifikation der Befehlsausführung erfordern einige Befehle weitere Informationen, die mittels Parameter festgelegt werden. Die Parameter werden durch *aktive oder passive Objekte* repräsentiert, die u.a. physische Objekte in der realen Welt repräsentieren können. Zu den passiven Objekten zählen u.a. Werkstücke, Behälter oder Werkzeuge ohne Aktorik. Aktive Objekte können auch als Parameter einer Anweisung auftreten, z.B. wenn diese selbst gehandhabt werden müssen, im Falle eines aktiv steuerbaren Greifers, der über ein Greiferwechselsystem abgelegt und wiederum aufgenommen werden kann.

## Operationen

Diese Symbole müssen direkt aneinander zu einem Befehl zusammengefügt werden - idealerweise per Drag&Drop, mit dem sich der Benutzer aus der Bibliothek die geeigneten Geräte, Operatoren und Parameterobjekte auswählen kann (vgl. Bild 4.14). Die Bibliothek ist individuell erweiterbar, stellt aber auch eine Auswahl von Standardbefehlen zur Verfügung:

- Jedes Programm-Modul trägt ein *Start-Symbol*, das u.a. die übergebenen Aufrufparameter über Objekte referenzieren kann.
- *Bedingungen* zur Verzweigung des Programmablaufes während der Laufzeit, beeinflusst durch aktuelle Objektzustände.
- *Schleifen* dienen zur Mehrfachausführung gleichen Programmcodes - entweder in einer festen Anzahl an Schleifendurchläufen oder bedingt durch einen logischen Ausdruck.
- Ein spezieller *Synchronisations-Operator* ist erforderlich, um parallele Abläufe an einer Stelle wieder zusammenlaufen zu lassen und erst nach vollständiger Beendigung dieser mit dem weiteren Verlauf fortzufahren.

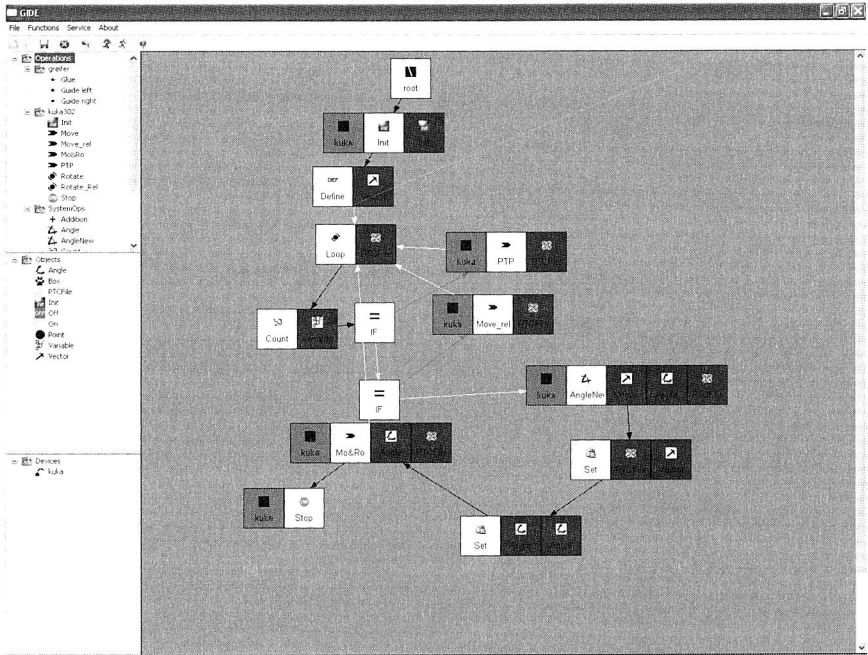


Bild 4.14: Prototypische Implementierung der graphischen Programmierungsumgebung zur Spezifikation der Ablaufsequenz. Links wird die Bibliothek mit vordefinierten Geräten, Operationen und Objekten zur Verfügung gestellt.

## Objekte

Der Ansatz der Objekte in GIDE ist vergleichbar mit der Objektorientierung hoher Programmiersprachen. Diese sieht die Aufteilung der zu beschreibenden Welt in Objekte vor, denen Eigenschaften und Operationen zu Grunde liegen. Die Objekte werden in diesem Falle über die Klassen, von denen sie instanziiert werden, typisiert und übernehmen aus diesen die Struktur und Implementierung. Mittels der Instanziiierung ist es zudem möglich von einem Typ für jedes physische Objekt eine eigene Instanz zu bilden, die dem zugeordneten physischen entspricht. Analog zu dieser Theorie, stellen die Objekte in GIDE die virtuelle Repräsentation physischer Teile dar. Von einem Objekttyp in GIDE können zur Laufzeit ebenso mehrere Instanzen gebildet werden, die Auskunft über die Art des physischen Objektes geben, den Zustand dokumentieren und ggf. Operationen anbieten.

Der Datenfluss wird in GIDE über Objekt-Referenzen gesteuert. Dies bedeutet, dass statt einer Objekt-Instanz der Entwickler eine Referenz zu einem bestehenden Objekt bilden und diese an eine andere Operation anhängen kann. Somit kann beispielsweise im vorhergehenden Befehl die Position des Bauteils ermittelt werden, während in einem der nachfolgenden Schritte diese Position zum Greifen



des Bauteils verwendet wird. Dabei muss sich der Entwickler nicht darum kümmern, in welchen Variablen, mit welchen Datentypen diese Positions- und Bauteiltypinformationen hinterlegt werden, da GIDE dies automatisiert übernimmt. GIDE hinterlegt alle Daten in den relevanten Objekten, die dem ausgeführten Befehl angehängt sind - in diesem Beispiel am Bauteilobjekt, da sich die Ortsangabe auf das Bauteil bezieht. Ebenso können die Werte wiederum automatisiert bei der Greifoperation aus dem Bauteil-Objekt herausgelesen werden. Somit kann der bisherige Entwicklungsoverhead, der bei jeder Programmiersprache besteht, maßgeblich reduziert werden und der Entwickler kann sich auf die Umsetzung der Steuerungsabläufe konzentrieren.

## Kapitel 5

### Konzepte und Algorithmen zur Realisierung autonom navigierender Fahrerloser Transportsysteme

Die zunehmende Forderung nach höherer Flexibilität an die Produktion zwingt die Unternehmen innovative Systemlösungen einzuführen. Die erforderliche Dynamik ist insbesondere im Materialflusssystem zu berücksichtigen, das den hohen Anforderungen hinsichtlich einer optimierten Auslastung der Stationen genügen und eine hohe Gesamtverfügbarkeit gewährleisten muss, um nachhaltige Lieferverzögerungen zu vermeiden. Diese Forderungen fließen bei diesem Beitrag in ein neuartiges Steuerungskonzept für Fahrerlose Transportsysteme (FTS) ein, das eine schnelle Inbetriebnahme, hohe Flexibilität und die unmittelbare Anpassung an sich ständig verändernde Produktionsbedingungen gewährleistet. Zudem wird der bereits in Kapitel 3 eingeführte agentenbasierte Ansatz zur optimierten Verteilung der Transportaufträge und impliziten Störkompensation an den autonom agierenden Fahrzeugen eingesetzt.

Unter dem neu eingeführten Begriff *autonom agierende Fahrerlose Transportfahrzeuge* werden zwei Aspekte verstanden:

- Zum einen die Erweiterung der Fahrzeuge um die Fähigkeit sich selbständig in einer teilweise bekannten Umgebung zu bewegen und in dieser selbständig Änderungen wahrnehmen und darauf reagieren zu können. Somit wird die Flexibilität der Fahrzeuge gesteigert, in wandelbaren und sich ständig verändernden Umgebungen mit geringem manuellen Konfigurationsaufwand zu navigieren zu können.
- Zum anderen die Verteilung der Steuerungs- und Koordinierungsaufgaben auf jedes einzelne Fahrzeug, wodurch die Inbetriebnahme und Umkonfigurationsaufwände reduziert und die Wandelbarkeit erhöht wird. Zudem entfällt die Notwendigkeit einer zentralen Leitsteuerung, die weitere Einrichtungsaufwände erfordert und zudem einen Single-Point-of-Failure darstellt.

Die Realisierung des ersten Ansatzes der autonomen Navigationsfähigkeit mit niedrigem Konfigurationsaufwand setzt verschiedene Algorithmen voraus, die automatisiert Daten aus den Primärquellen vorverarbeiten, ableiten und diese integrativ ohne manuellen Anpassungsaufwand an die nächste Softwarekomponente weiterleiten. Hierbei kommen verschiedene Verfahren und Hilfsmittel zum Einsatz, die zum einen erlauben, die Entwicklung und Inbetriebnahme maßgeblich zu vereinfachen und zum anderen Vorteile für den laufenden Betrieb mit sich bringen bzw. Anpassungen zur Laufzeit ohne Stilllegung der Anlage ermöglichen.

Voraussetzung für die einfache Anpassbarkeit des Fahrerlosen Transportfahrzeugs an veränderte Produktionsumgebungen stellt die schnelle Erfassung des Kartenmaterials dar. Hierzu wird ein Vermes-

sungswerkzeug vorgestellt, das integriert in das Fahrerlose Transportsystem die Koordinatendaten der Hindernisse und Bahnhöfe, selbst zur Laufzeit und ohne weiteren Anpassungsaufwand direkt an die Fahrzeuge weitergeben kann (Kapitel 5.3). Diese Daten werden in ein Kartenmodell eingetragen, das abgestimmt auf den Bahnplanungsalgorithmus gewählt werden muss (Kapitel 5.4). Aus verschiedenen bewerteten Bahnplanungsverfahren (Kapitel 5.5) wurde die Zellenmethode (Kapitel 5.6), die es ermöglicht, Fahrwege zur Laufzeit mit den aktuellen Umgebungsdaten zu berechnen, gewählt.

## **5.1 Anforderungen in wandelbaren Produktionsumgebungen**

Aus den in Kapitel 2.3.1 beschriebenen Gründen bieten Fahrerlose Transportsysteme nach dem heutigen Stand der Technik nicht die notwendige Flexibilität, um in Produktionen mit stark variierenden Bedingungen und hochfrequenten Umbaumaßnahmen wirtschaftlich eingesetzt werden zu können. Hieraus ergeben sich neuartige Anforderungen in den Bereichen der Steuerungsentwicklung für Fahrerlose Transportsysteme, der Inbetriebnahmephase, neuartiger Wahrnehmungsfähigkeiten während der Transportausführung und der Organisation während des Betriebs.

Für den Einsatz von Fahrerlosen Transportsystemen in wandelbaren Produktionsumgebungen müssen sowohl der Konfigurationsaufwand als auch die Störanfälligkeit maßgeblich reduziert werden. Zur Reduktion des Konfigurationsaufwands ist es erforderlich, geeignete und erweiterte Algorithmen zur Navigation umzusetzen, die selbständig aus vorhandenem Datenmaterial, wie den Umgebungsinformationen, die Pfade ableiten können. Wandelbare Produktionsumgebungen zeichnen sich dadurch aus, dass ad hoc selbst im Betrieb einer Produktionsanlage Umbaumaßnahmen vorgenommen werden können. Somit muss dem Problem begegnet werden, zweidimensionale Pläne der Produktion von Vermessungsfirmen erheben zu müssen, um genaue Ausgangsdaten zur manuellen Wegeerfassung zur Verfügung zu haben (vgl. Kapitel 2.3). Sind hierbei Änderungen an der Produktionsanlage erforderlich, muss wiederum das Kartenmaterial durch Experten angepasst werden. Auf Grund dieser Notwendigkeit und dem Ansatz die Fahrwege manuell spezifizieren bzw. ändern zu müssen, kann die Technologie derzeit verfügbarer Fahrerloser Transportsysteme in wandelbaren Produktionsumgebungen dem wirtschaftlichen Einsatz nicht Rechnung tragen. Zur Bewältigung dieser Herausforderungen sind neue Mechanismen erforderlich, die es erlauben die Konfiguration des Fahrerlosen Transportsystems zu vereinfachen und Aufwände maßgeblich zu verkürzen. Ein Ansatz ist, technisch gering versierten Mitarbeitern Hilfsgeräte, wie Vermessungssysteme, zur Verfügung zu stellen, die in das Fahrerlose Transportsystem vollständig integriert sind. Hiermit kann der Mitarbeiter neue Hindernisse präzise im Koordinatensystem der Fahrzeuge erfassen und das Kartenmaterial selbständig verändern.

Weiteres Potential zur Aufwandsreduktion liegt in der softwaretechnischen Anbindung der Übergabebahnhöfe in den Gesamtverbund. Die Installation neuer oder das Entfernen bestehender Bahnhöfe wird zwar manuell im Kartenmaterial vermerkt, jedoch müssen die steuerungstechnischen und logistischen Anpassungen nach derzeitigem Stand der Technik manuell durchgeführt werden. Diese manuellen Aufwände sollen durch eine automatisierte Integration neuer Bahnhöfe mit neuen Bearbeitungsstationen auf Softwareseite erfolgen. Die spezifische, hard-, software- und anwendungsspezifische Auslegung und Anpassung der Softwareschnittstellen zwischen den Bahnhöfen und dem zentralen Leitrechner soll vollständig entfallen und normierten Softwarebausteinen weichen, um eine

reibungsfreie Inbetriebnahme ohne Testphase zu ermöglichen. Hierbei sollen sich die Bahnhöfe in die bestehende wandelbare Steuerungsarchitektur (Kapitel 4) integrieren, um dabei einerseits von den Vorzügen der Strukturerkennung und Selbstkonfiguration (Kapitel 4.3) zu profitieren. Andererseits können die Bahnhöfe unmittelbar mit allen Entitäten im Produktionssystem kommunizieren und interagieren, d.h. sowohl mit den Bearbeitungsstationen als auch mit dem angebundenen Transportsystem.

Die Gesamtverfügbarkeit des fahrzeuggesteuerten Materialflusssystems kann verbessert werden, indem auf zentrale Steuerungssysteme verzichtet wird. Wie bereits in Kapitel 2.3 erläutert, stellen diese einen Single-Point-of-Failure dar, d.h. deren Ausfall würde den Totalausfall des gesamten Transportsystems bedeuten. Die Entscheidung, ob auf einen zentralen Leitrechner verzichtet werden kann oder nicht, hängt ausschließlich von der Optimierungsstrategie zur Einlastung der Transportaufträge an den Fahrzeugen ab.

Neben der flexiblen Integration der Bahnhöfe, sollen sich ebenso die Fahrzeuge an neue Umgebungen selbstständig anpassen können. Dies betrifft einerseits die flexible Interaktion mit allen Teilnehmern im Verbund des Fahrerlosen Transportsystems. Mit der bestehenden Steuerungsarchitektur wird gewährleistet, dass die Fahrzeuge direkt mit den Übergabebahnhöfen und zur Konfliktauflösung mit anderen Fahrzeugen interagieren können. Andererseits müssen die Fahrzeuge in einer ständig wandelbaren Umgebung auf Veränderungen reagieren können, um den effektiven und wirtschaftlichen Einsatz zu gewährleisten. Die Fahrzeuge sollen selbstständig unbekannte Hindernisse erkennen und diesen zur Laufzeit ausweichen können. In veränderbaren Umgebungen treten unbekannte Hindernisse u.a. dann auf, wenn sich die Anlage noch im Umbau befindet und zu verbauende physische Komponenten temporär zwischengelagert werden oder die neue Anlagenstruktur noch nicht rechtzeitig gemessen wurde. Zudem müssen sich die Fahrzeuge bei Anlagenänderungen in der Organisation der Transportaufträge anpassen. Dies betrifft zum einen Störungen im Transportsystem, wenn z.B. Fahrzeuge ausfallen oder auf Grund regelmäßiger Wartungszyklen oder Batterieaufladevorgänge nicht zur Verfügung stehen. Zum anderen müssen dynamische Veränderungen im Materialfluss erkannt und unmittelbar darauf reagiert werden. Transportaufträge zu ausgefallenen Bearbeitungsstationen, die nicht mehr beliefert werden können, müssen revidiert und die zugehörigen Werkstücke auf alternative Stationen, sofern verfügbar, umverteilt bzw. temporär an einer anderen Stelle zwischengelagert werden.

## 5.2 Neuer Steuerungsansatz für Fahrerlose Transportsysteme

Um den in Kapitel 5.1 beschriebenen Anforderungen an Fahrerlose Transportsysteme im wandelbaren Produktionsumfeld bei gleichzeitig wirtschaftlichem Einsatz genügen zu können, wird im Folgenden ein neuer Steuerungsansatz entwickelt, der dem Prinzip verteilter, autonomer Systeme folgt.

Hierzu wird eine verteilte, dezentrale Steuerungsstruktur konzipiert, die aus verschiedenen, gleichberechtigten Teilnehmern besteht (Bild 5.1). Diese kommunizieren und arbeiten, basierend auf dem Agenten-Prinzip, direkt miteinander, um die gemeinsame Gesamtaufgabe zu erfüllen (Kapitel 2.4). Um der Forderung nach mehr Dynamik in Produktionssystemen nachzukommen, sollen die Fahrerlosen Transportfahrzeuge nicht mehr lediglich als Befehlsempfänger eines zentralen Leitrechners agieren, sondern autark zur Laufzeit selbstständig handeln können. Dies setzt voraus, dass ihnen bestimmte

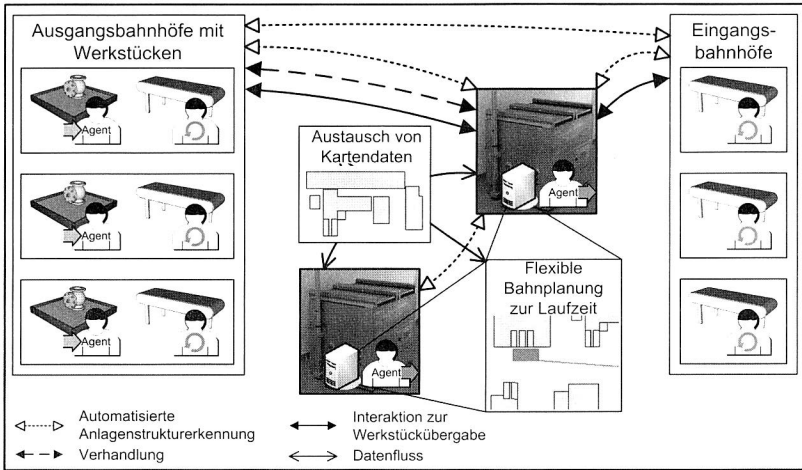


Bild 5.1: Neuer Steuerungsansatz für Fahrerlose Transportfahrzeuge mit flexibler Bahnplanung zur Laufzeit, basierend auf kontinuierlich aktualisiertem Kartenmaterial. Bahnhöfe und Fahrzeuge können automatisiert die Anlagenstruktur erkennen und direkt miteinander zur Auftragszuteilung und Werkstückübergabe interagieren.

Teilaufgaben übertragen werden, die sie in Eigenregie erfüllen müssen. Mit dieser Fähigkeit können sie jegliche Entscheidungen, die den Transport der Werkstücke zwischen zwei Bahnhöfen betreffen, selbst vornehmen. Dies umfasst u.a. die Fähigkeit, Bahnen selbständig zur Laufzeit planen und berechnen zu können. Hierbei sollen keine starren Bahnen mehr vorgegeben werden, sondern diese zur Laufzeit entsprechend dem aktuellen Hallenplan, der sich selbst unmittelbar zu jedem beliebigen Zeitpunkt verändern kann, vor jedem Transportauftrag berechnet werden. In der Berechnung können somit nicht nur bekannte, d.h. manuell vermessene Hindernisse betrachtet, sondern ebenso neue, bislang unbekannte Hindernisse, berücksichtigt werden. Hierzu ist Messsensorik erforderlich, die in Fahrtrichtung den Umgebungsbereich vor dem Fahrzeug abtastet, um dabei neue Hindernisse zu erkennen und diese automatisiert in das Kartenmaterial aufzunehmen.

Dadurch entfällt die manuelle Festlegung der Bahnen im Raum und die Validierung dieser mittels Testfahrten (vgl. Kapitel 2.3). Zum einen kann die Akzeptanz Fahrerloser Transportsysteme in der Fertigung durch eine vereinfachte Handhabung gestärkt und zum anderen der wirtschaftliche Anreiz geschaffen werden, diese selbst in hochflexiblen Fertigungen einzusetzen.

Zur optimierten Auslastung der Fahrzeuge und automatisierten Störkompensation soll der in Kapitel 3 eingeführte agentenbasierte Ansatz zur Materialflusssteuerung in verteilten Steuerungsumgebungen angewendet werden, dessen Funktionsweise in Kapitel 6 detaillierter beschrieben wird. Voraussetzung für dieses Modell ist die Fähigkeit, mit den Teilnehmern direkt interagieren zu können. Daher baut der vorliegende Ansatz auf der eigens entwickelten Steuerungsarchitektur (siehe Kapitel 4), die erweiterte Kommunikationsdienste (Middleware-Dienste) und Entwicklungswerkzeuge zur vereinfachten

Softwareentwicklung zur Verfügung stellt, auf. Die Interaktion zwischen den Bahnhöfen und Fahrerlosen Transportfahrzeugen umfasst hierbei nicht nur die Einlastung von Transportaufträgen. Mit den autonomen Fahrzeugen, die über eine integrierte Steuerung verfügen und selbständig im Produktionsumfeld agieren können, sind neue Konzepte erforderlich, die ein erhöhtes Kommunikationsaufkommen zur Lösungsfindung erfordern. Dies betrifft zunächst Einigungsverfahren, die entscheiden, in welcher Reihenfolge die wartenden Fahrzeuge den Bahnhof anfahren dürfen, um Konfliktsituationen bzw. Verklemmungen zu vermeiden. Zudem ist ein Handshake-Verfahren zur Übergabe der Werkstückträger erforderlich, das allgemein, d.h. transportsystem- und fahrzeugsystemunabhängig, gültig ist und mit den autonomen Fahrzeugen Zustandsinformationen zur sicheren Werkstückübergabe austauscht. Hierbei werden die in Kapitel 2.3 beschriebenen Sensorik-Lösungen zur Anwesenheitserkennung der Fahrzeuge durch die Kommunikation über das Netzwerk abgelöst und somit die Grundlage für weiterführende Interaktionsformen zur Erhöhung der Zuverlässigkeit und Kompensation von Problemen geschaffen.

### 5.3 Kartographierung mittels integriertem Vermessungswerkzeug

Die Kartographierung der Produktionsumgebungen ist Voraussetzung für alle Anwendungen von Fahrerlosen Transportsystemen - unabhängig von der Navigationsstrategie. Grundlegend müssen alle relevanten Objekte in der Produktionsumgebung, wie Hindernisse und Übergabebahnhöfe, erfasst und diese mit geeigneten Beschriftungen und Identifikatoren versehen werden, damit sich zum einen der Anwender möglichst schnell und einfach zurechtfindet und sich zum anderen die Komponenten, wie die Bahnhöfe, automatisiert in die Materialflusssteuerung einbinden lassen. Dieses grundlegende Datenmaterial wird bislang oftmals von externen Vermessungsunternehmen [104] erfasst oder aus den Planungsdaten der Produktionshalle abgeleitet (siehe Kapitel 2.3.1). Der erste Fall erfordert u.a. einen höheren finanziellen Aufwand, im zweiten Fall entspricht oftmals das vorhandene Kartenmaterial aus der Planungsphase jedoch nicht den realen Aufbauten.

Der vorliegende Ansatz fußt auf dem Vorhandensein eines freinavigierenden Positionserkennungssystems, das für freinavigierende Fahrzeuge eingesetzt wird. Zwei weitverbreitete Systeme zur freien Positionsbestimmung sind Lasernavigationssysteme und Rasternavigationssysteme [105]. Letztere erfordern die Installationen von Markierungen im Boden in äquidistanten Abständen. Zudem kann jede Markierung mit einer eindeutigen Kennung, z.B. einem RFID-Tag [106], versehen werden, um selbst die absolute Position nach dem Einschalten des Fahrzeugs, unabhängig von der letzten Position, bestimmen zu können.

Bei der Positionsbestimmung mittels Laserscanner sind Reflektionsmarken erforderlich, die an festen Orten, bevorzugt den Wänden oder Säulen bzw. Stahlträgern der Produktionshalle angebracht werden. Diese sollten zur Gewährleistung einer hohen Genauigkeit einmalig von einer Vermessungsfirma in einem globalen Koordinatensystem erfasst werden. Die Markierungen sind über ein bis zwei Reflexionsstreifen pro Markierung und mit unterschiedlicher Breite bzw. Abstand zueinander typisiert. Diese können über einen Laserscanner am Fahrzeug erfasst werden. Basierend auf mindestens drei erfolgreich identifizierten, bekannten Markierungen, kann der dem Laserscanner zugeordnete Rechner eindeutig die Position und Drehlage des Fahrzeugs im Raum bestimmen [107].

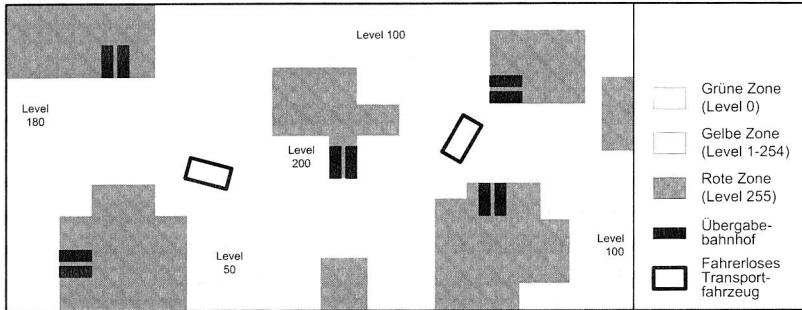


Bild 5.2: Umgebungskarte mit grünen, gelben und roten Zonen - rote Zonen stellen Hindernisse und grüne Zonen für das Fahrzeug freie Bereiche dar. Die gelben Zonen repräsentieren alternative Bereiche, die nur bei Bedarf genutzt werden sollen. Sie fließen mit unterschiedlicher Gewichtung in die Bahnplanung zur Bewertung der möglichen Alternativwege ein.

Der neue Ansatz zur Erfassung der Produktionsumgebung beruht darauf, das gleiche Messsystem, das zur Positionsbestimmung für Fahrerlose Transportfahrzeuge zum Einsatz kommt, für die manuelle Vermessung der Produktionshalle einzusetzen. In der Analysephase wurden verschiedene Möglichkeiten der Vermessung, basierend auf dem Positionsbestimmungssystem, erörtert. Diese haben ergeben, dass ein manuell geführter, auf passiven Rollen gelagerter Messwagen die Anforderungen einer schnellen und einfachen Umgebungserfassung am besten erfüllt. Dieser kann selbst von ungeschulten Mitarbeitern nach kurzer Einführung genutzt werden und stellt nur die wesentlichen Eingabefunktionen zur Verfügung, um Hindernisse und Übergabebahnhöfe im Raum zu erfassen bzw. zu ändern.

Zur datentechnischen Weiterleitung der aufgenommenen Messpunkte vom Handwagen an alle Fahrzeuge bzw. einen Kartenserver, muss dieser über eine integrierte Steuereinheit verfügen, die eine Benutzerinteraktion zur Erfassung der Messpunkte und Spezifikation des zu vermessenden Objektes gewährleistet. Über die Eingabeschnittstellen kann neben der Bearbeitung der Punkte auch dem erfassten Objekt ein eindeutiger Name vergeben werden. Als Objekt-Typen werden Zonen und Übergabebahnhöfe unterschieden. Zonen sind gewichtete Bereiche, denen ein Levelwert zwischen 0 und 255 zugewiesen wird.

Die Zonen werden wie folgt klassifiziert (Bild 5.2):

- In grünen Zonen darf sich das FTF völlig frei bewegen. Diese haben den Level-Wert 0. Standardmäßig werden initial alle freien Bereiche auf den Level-Wert 0 gesetzt.
- Gelbe Zonen sollten nur dann vom FTF befahren werden, wenn sich z.B. im grünen Bereich ein Hindernis befindet. Ihr Level liegt zwischen 1 und 254. Je niedriger der Wert, desto eher wird der Bereich von den Fahrzeugen bevorzugt.
- Rote Zonen markieren Hindernisse oder sonstige verbotene Bereiche, die das FTF niemals befahren darf. Diese halten einen Level-Wert von 255.

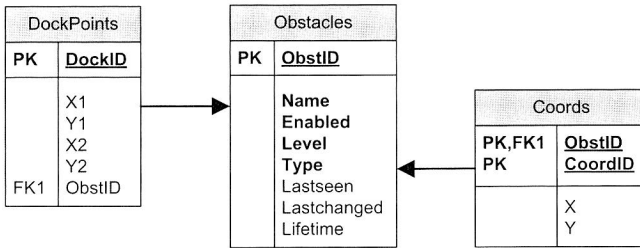


Bild 5.3: Entity-Relation-Ship-Diagramm des Kartenmaterials zur Aufnahme der Zonen und Bahnhöfe.

Der Kartenbestand wird sowohl am Kartenserver, als auch in jedem Fahrzeug in einer SQL-Datenbank gespeichert. Das zugehörige Entity-Relationship-Diagramm ist in Bild 5.3 dargestellt.

In der Tabelle *Obstacles* werden alle Zonen gespeichert, sowohl die gelben, als auch die roten Zonen. Die grünen Zonen müssen nicht explizit eingegeben werden, da alle Bereiche, die nicht erfasst sind, für die Fahrzeuge automatisch frei und somit befahrbar sind. Die zugehörigen Koordinaten der Eckpunkte einer Zone werden in der Tabelle *Coords* gespeichert und müssen gemäß dem Schlüssel *CoordId* aufsteigend sortiert sein. Neben einem lesbaren Namen besitzen die Zonen auch eine eindeutige ID. Diese wird vom Ersteller der Zone vergeben und ist in der gesamten Produktion für die Lebenszeit dieser Zone eindeutig. Da unterschiedliche Datenerfasser zur Vermessung der Umgebung zum Einsatz kommen können, ist es erforderlich, an den verteilten Erzeugern eine global eindeutige ID, z.B. mittels UUID-Verfahren [108], zu vergeben. Somit kann Kartenmaterial konsistent und ohne Vervielfältigung gleicher Dateninhalte verbreitet werden. Die Bearbeitung der Kartendaten kann sowohl von mehreren Mitarbeitern, als auch von den Fahrzeugen vorgenommen werden, die in der Lage sind, neue Hindernisse zu erkennen, in die Karte einzutragen und diese Informationen an alle Fahrzeuge weiterzuleiten. Im letzten Falle, der Hinderniserkennung durch die Fahrzeuge, handelt es sich um temporäre Hindernisse, von denen nicht bekannt ist, ob sie dauerhaft existieren. Aus diesem Grund muss nach einer bestimmten Zeit überprüft werden, ob die Hindernisse noch existieren und somit ggf. bislang blockierte Wege, die zu Umwegen führten, wieder befahrbar sind. Zu diesem Zweck wird in den temporären Hindernissen ein Zeitstempel mit dem Zeitpunkt des letzten Sichtkontakts gespeichert. Wurde seit einem festgelegten Timeout kein weiterer Sichtkontakt gemeldet, wird das Hindernis automatisch aus der Karte entfernt, damit das Fahrzeug erneut den Weg in die Bahnplanung einbezieht und dabei prüft, ob das Hindernis noch vorhanden ist. Voraussetzung hierzu ist, dass die Uhren auf den Fahrzeugen synchronisiert sind. Hierbei ist jedoch keine hohe Synchrongenauigkeit erforderlich. Da es sich jedoch nicht um zeitkritische Abläufe handelt, spielt die Genauigkeit der Uhrensynchronisation keine Rolle und darf sogar im Sekundenbereich liegen.

Neben den Hindernissen werden im Kartenmaterial u.a. die Koordinaten der Übergabebahnhöfe hinterlegt. Ein Übergabebahnhof wird definiert durch ein einzelnes Transportband. Sind zwei Bänder nebeneinander angeordnet, z.B. eines zur Aufnahme und eines zur Abgabe, dann handelt es sich in diesem Fall um zwei Übergabebahnhöfe (vgl. Bild 5.4).

Die Übergabebahnhöfe werden als Vektoren zweier Punkte in der Tabelle *Dockpoints* gespeichert,



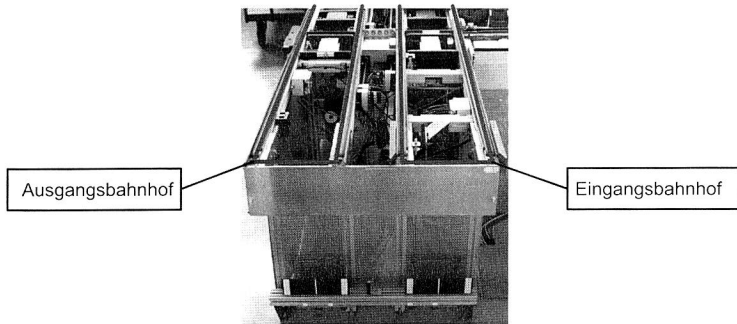


Bild 5.4: Darstellung eines Doppelbahnhofs, bei dem jedes Förderband je einen Ein- und Ausgangsbahnhof repräsentiert.

die eine bestimmte Richtung aufweisen müssen. In Richtung der Vektoren muss der Andockbereich für die Fahrzeuge auf der linken Seite liegen, d.h. in  $90^\circ$  gegen den Uhrzeigersinn. Somit kann der Bahnplaner eindeutig erkennen auf welcher Seite der Vektoren der Andockbereich liegt. Zwar ist es möglich den Andockbereich automatisch erkennen zu lassen, da sich hinter einen Bahnhof i.d.R. ein Hindernis befindet, aber es können auch virtuelle Bahnhöfe definiert werden, die keine physische Repräsentation haben. In diesem Fall handelt es sich um Haltestellen für das Fahrzeug, die als Parkplätze dienen, um die Batterien aufzuladen oder auf Transportaufträge zu warten. Es können auch solche Haltestellen als Übergabebahnhöfe definiert werden, an denen kein physischer Bahnhof existiert und Mitarbeiter die Werkstückübergabe manuell vornehmen. Hierzu ist keinerlei Mechanik erforderlich und der Ansatz passt sich sehr gut in das Konzept wandelbarer Produktionssysteme ein, das u.a. sehr von manuellen Arbeitsplätzen geprägt sein kann.

Mit dem vorliegenden Konzept zur Vermessung der Karte auf Basis des Positionsbestimmungssystems, kann neben einer hohen Genauigkeit die unmittelbare Erfassung des Kartenmaterials im gleichen Koordinatensystem der Fahrzeuge gewährleistet werden. Somit können diese ohne weitere Rotations-, Translations- oder Skalierungsschritte in das Kartenmaterial der Fahrzeuge übernommen werden. Folglich kann der Inbetriebnahme- und der Anpassungsaufwand bei Änderungen an der Produktionsanlage maßgeblich durch diese integrierten Datenverbindungen und -bereitstellungen reduziert werden.

Die Problematik zur genauen und schnellen Kartographierung der Produktion besteht selbst in bestehenden kommerziellen Lösungen Fahrerloser Transportsysteme. Das vorliegende Konzept zur Kartographierung kann somit gleichermaßen auf alle freinavigierenden Fahrerlosen Transportsysteme angewendet werden, um, basierend auf dem erfassten Kartenmaterial, nach aktuellem Stand der Technik die Fahrwege manuell zu definieren.

## 5.4 Bekannte Methoden zur Umgebungsmodellierung

Die Grundvoraussetzung für die Bahnplanung sind detaillierte Informationen sowohl über das Fahrzeug selbst, als auch über die Umgebung in der es sich bewegt. Zur Erfassung der Fahrzeugumgebung stehen eine Reihe von Modellen zur Verfügung, die jeweils ihre Rechtfertigung in verschiedenen Anwendungsfällen haben und unterschiedliche Aspekte des Kartenmaterials hervorheben bzw. speichern oder verarbeitungsgünstig ablegen können. Um sicherzustellen, dass kollisionsfreie Bahnen berechnet werden, ist es außerdem notwendig, auf ein zum Umgebungsmodell passendes Fahrzeugmodell zurückzugreifen.

Die Forderung an die Modellierung der Umgebung ist ein möglichst geringer Speicherbedarf, bei gleichzeitig möglichst hohem Detaillierungsgrad. Zudem besteht die Anforderung an die Datenmodellierung, eine möglichst schnelle und einfache Verarbeitungsmöglichkeit des Datenbestands nachgelagerter Bahnplanungsalgorithmen zu ermöglichen. Hierzu werden im Folgenden verschiedene weit verbreitete Modelle vorgestellt.

### Rastermodell

Das Rastermodell teilt die Karte gemäß einer Matrix in disjunkte Bereiche, vergleichbar mit einem Digitalbild, ein. Jeder Eintrag dieser Matrix nimmt einen diskreten Zustand an, d.h. *belegt durch ein Hindernis* oder *frei von Hindernissen* [109]. Diese Form der Datenrepräsentation hat jedoch zur Folge, dass eine hohe Genauigkeit bei der Darstellung der Umgebung zu einem sehr hohen Speicherplatzbedarf führt. Außerdem bringt das Rastermodell für die Bahnplanung den Nachteil mit sich, dass einzelne Zellen nur mit weiteren Zusatzinformationen einem Hindernis zugeordnet werden können, was wiederum einen verstärkten Speicherbedarf zur Folge hat. Diese Zuordnung ist wesentlich, wenn Hindernisse eingetragen und ebenso auf einfache Art und Weise wieder aus der Karte entfernt werden sollen.

In Hinblick auf die Bahnplanungsmethoden, eignet sich der Rasteransatz für die Bahnplanung auf Zellenbasis, die direkt auf dem Kartenmaterial arbeiten kann, ohne diese in die Matrix-Form konvertieren zu müssen [110]. Bild 5.5 stellt ein Beispiel für eine Rasterkarte dar. Bei diesem Modell muss zu jeder Zelle ein binärer Wert gespeichert werden, der aussagt, ob eine Zelle frei oder belegt ist. Hier ist gut zu erkennen, wie Detailinformationen der Hindernisse, d.h. die exakten Maße, verloren gehen.

### Quadtree-Methode

Alternativ zum Rastermodell bietet die Quadtree-Methode die Möglichkeit, Speicherplatz mittels Zellen unterschiedlicher Größe einzusparen. Hierbei werden in der nahen Umgebung der Hindernisse kleine Zellen verwendet, während große freie Bereiche mit großen Zellen abgedeckt werden (vgl. Bild 5.5).

### Topographisches Vektormodell

Das Vektormodell erlaubt die Modellierung der Umgebung mit hoher Genauigkeit und gleichzeitig niedrigem Speicherbedarf. Bei Verwendung des topographischen Vektormodells werden Objekte als Polygone abgebildet, die Begrenzungslinien zum freien Bereich definieren [111]. Das bedeutet, dass

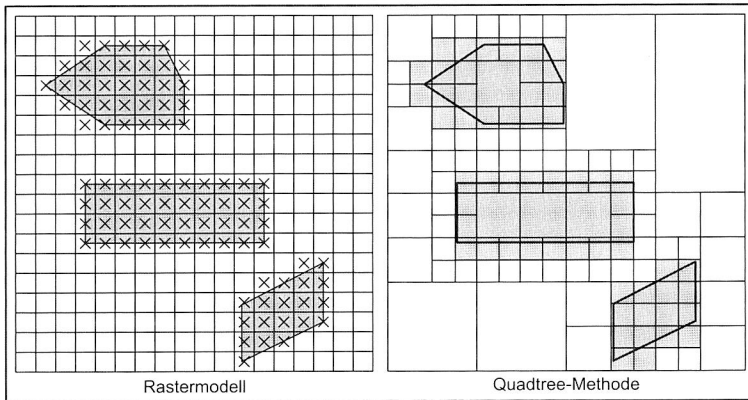


Bild 5.5: Beispiel zum Vergleich der Modellierungsmethoden mittels Rastermodell und Quadtree-Methode.

sich der Speicherbedarf auf eine überschaubare Anzahl von Polygoneckpunkten beschränkt. Muss jedoch festgestellt werden, ob ein Punkt innerhalb oder außerhalb eines Hindernisses liegt, sind hierzu rechenaufwendige Algorithmen im Gegensatz zum Rastermodell erforderlich. Der Vorteil dieser Modellierungsform liegt darin, dass jedes Hindernis einzeln gespeichert und ebenso wieder bearbeitet werden kann. In diesem Falle sind selbst überlagerte Polygone unterscheidbar. Da alle Eckpunkte der Polygone zusammengehörig gespeichert sind, wird eine einfachere Handhabung der Hindernisinformationen ermöglicht. Die Polygone können durch Hinzufügen, Entfernen bzw. Ändern von Eckpunkten modifiziert oder auch vollständig aus der Karte entfernt werden. Diese Vorzüge kommen insbesondere in sich ständig verändernden Umgebungen und folglich anzupassenden Kartenmaterials zum Tragen. In Bild 5.6 ist ein Beispiel für eine topographische Karte dargestellt. Es sind lediglich die Eckpunkte und somit die Umrisse der Hindernisse gespeichert.

### Topologisches Vektormodell

Im Gegensatz zu den topographischen Karten, stellen die topologischen Karten ein negatives Abbild der topographischen Karten dar. Sie enthalten Informationen über die erreichbaren, also hindernisfreien Bereiche des Arbeitsraumes [109]. Dazu werden Knotenpunkte gespeichert, die über einen Graphen verbunden sind. Daraus ergibt sich der sogenannte Erreichbarkeitsgraph, der alle möglichen kollisionsfreien Pfade enthält. Die Erstellung einer topologischen Karte im Vorfeld ist zwar rechenintensiv, erfordert aber einen geringen Rechenaufwand bei der Bahnplanung. Hierbei steht komprimiertes Datenmaterial über mögliche Fahrwege zur Verfügung, das zur Laufzeit eine schnelle Berechnung der kürzesten Wege erlaubt. Diese Modellierungsform eignet sich bevorzugt für statische Umgebungen. Das Abbilden dynamischer Umgebungen gestaltet sich als rechenaufwendig, da selbst für geringfügige Änderungen mit dem Basisalgorithmus die gesamte Karte neu berechnet werden muss. Um diesen Aufwand zu reduzieren, sind erweiterte Algorithmen und Mechanismen erforderlich, die zahlreiche Ausnahmefälle und Sonderbedingungen erfordern. Ein Beispiel des Graphen ist in Bild 5.6 dargestellt.

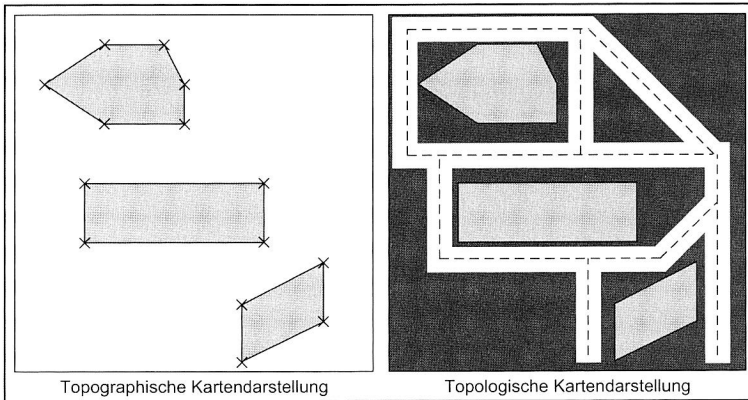


Bild 5.6: Repräsentation zwei-dimensionaler Kartendaten mittels topographischer und topologischer Darstellung im Vergleich.

### Modellierung des Fahrzeugs

Zur Modellierung des Fahrzeugs gibt es im Wesentlichen zwei Möglichkeiten: die punktförmige Modellierung oder die Modellierung als Rechteck.

Bei der punktförmigen Modellierung ist es erforderlich, die Umgebung an das Fahrzeug anzupassen, um die Kollisionssicherheit mit den Hindernissen sicherzustellen. Dies bedeutet, dass beim Rastermodell oder bei den topographischen Karten sämtliche Hindernisse mindestens um den Radius des Fahrzeug-Umkreises ausgedehnt werden müssen. Bei Verwendung von topologischen Karten muss bei Erstellung der Karte darauf geachtet werden, dass eine Teilroute nur dann zum Graphen aufgenommen wird, wenn sie einen genügend großen Abstand zu allen angrenzenden Hindernissen aufweist. Im resultierenden freien Raum kann sich das punktförmige Fahrzeug dann völlig frei bewegen.

Die Modellierung des Fahrzeugs in seiner tatsächlichen Größe erfordert keinerlei Anpassungen der Hindernisse. Da das Fahrerlose Transportfahrzeug im Gegensatz zu den Hindernissen ein bewegtes Objekt ist, muss für jeden Punkt auf dem geplanten Pfad die Kollisionsfreiheit der Hindernisse und des FTF sichergestellt werden. Dies kann zum Beispiel bewerkstelligt werden, indem zur geplanten Route auf beiden Seiten parallele temporäre Strecken angelegt werden und deren Kollisionsfreiheit mit sämtlichen Hindernissen geprüft wird. Dies führt aber bei einer großen Anzahl von Hindernissen und Routenpunkten zu einem erheblichen Rechenaufwand.

## 5.5 Übersicht der bekanntesten Bahnplanungsalgorithmen

Im neuen Navigationskonzept für autonom navigierende Fahrerlose Transportfahrzeuge berechnet jedes Fahrzeug selbständig die Bahn durch die Produktion von vorgegebenem Kartenmaterial (Kapitel 5.2). Die Karte wird, wie in Kapitel 5.3 beschrieben, manuell erfasst oder kann selbst vom FTF mittels

Sensorik aktualisiert werden. Zur Berechnung kollisionsfreier Pfade in bekannten Umgebungen, gibt es bereits etablierte Algorithmen, die im Folgenden kurz vorgestellt werden.

### 5.5.1 Bahnplanung mittels topologischer Karten

Der Graph der topologischen Karte wird durch eine Menge von Punkten definiert, die sich auf einer Verbindungsstrecke in der Mitte zwischen zwei Hindernissen oder zwischen einem Hindernis und dem Rand der Karte befinden. Zur Ermittlung der Verbindungsstrecken existieren verschiedene Verfahren zur Zerlegung der freien Bereiche der Karte. Im Folgenden werden beispielhaft einige Varianten vorgestellt [112]:

- Trapezoide Zellzerlegung: Von jedem Eckpunkt eines Hindernisses wird jeweils eine Gerade nach links, rechts, oben und unten gezogen, bis sie auf ein anderes Hindernis oder die Begrenzung der Karte trifft (Bild 5.7a).
- Konvexe polygonale Zellzerlegung: Von jedem Eckpunkt eines Hindernisses wird eine Gerade zu allen Eckpunkten anderer Hindernisse oder dem Eckpunkt der Begrenzung der Karte gezeichnet.
- Sichtbarkeitsgraph: Alle Ecken der Hindernisse sowie Start- und Zielpunkt werden mit sämtlichen anderen durch eine Gerade direkt erreichbaren Ecken verbunden.
- Voronoi-Diagramm: Ausgehend von einem Hindernis wird zu einem Voronoi-Punkt und von diesem zu einem anderen Hindernis jeweils eine Linie gelegt (Bild 5.7b). Die beiden Teilstrecken müssen gleich lang sein.

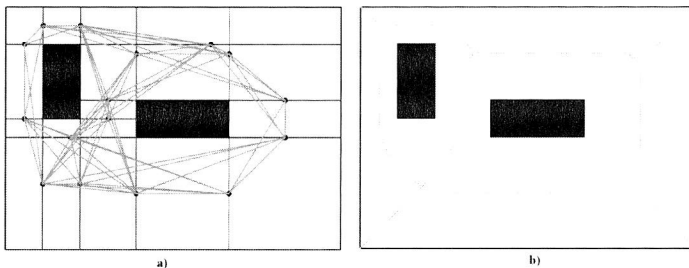


Bild 5.7: Vergleich zwischen a) Trapezoide Zellzerlegung und b) Voronoi-Diagramm. [109]

Das Voronoi-Diagramm ist als eine Punktmenge definiert, die die Eigenschaft besitzt, dass jeder Punkt von zwei unterschiedlichen Hindernissen gleich weit entfernt ist. Diese Punkte werden als Voronoi-Punkt bezeichnet [113]. Theoretisch gibt es in einer Karte unendlich viele Voronoi-Punkte, da die Voronoi-Punkte nicht nur von den Ecken, sondern auch von den Kanten der Hindernisse erstellt werden können. Ziel dieses Verfahrens ist die Gewährleistung einer ausreichenden Genauigkeit bei gleichzeitigem Einsatz einer möglichst geringen Anzahl von Voronoi-Punkten. Kommen zu wenige Punkte zum Einsatz, können direkte Verbindungen zwischen den Punkten unter Umständen sehr nahe

am Hindernis vorbeiführen. Sehr viele Punkte dagegen verursachen einen hohen Rechenaufwand bei der Erstellung des Diagrammes und auch bei der Planung einer Route. Bei einer optimalen Abwägung zwischen der Genauigkeit einerseits und dem Rechenaufwand andererseits, ergibt sich ein Netz von Punkten, die in der Mitte zwischen den Hindernissen liegen und deren Verbindungen nahezu in der Mitte zwischen den Hindernissen verläuft. In [113] wird ein Algorithmus zur schnellen und effizienten Suche einer ausreichenden Anzahl von Voronoi-Punkten für das Voronoi-Diagramm beschrieben. Die Verbindung sämtlicher Punkte ergibt einen Erreichbarkeitsgraphen, der alle möglichen Routen, die das Fahrzeug befahren darf, zusammenfasst. Basierend auf diesem, kann mit geringem Rechenaufwand die kürzeste Route zum Ziel bestimmt werden. Zum Vergleich ist in Bild 5.7 ein Beispiel zur trapezoiden Zellzerlegung und dem Voronoi-Diagramm dargestellt. Die Pfade im Trapezoiden Diagramm verlaufen sehr nahe an den Hindernissen, während diese im Voronoi-Diagramm immer möglichst weit vom Hindernis entfernt sind.

### 5.5.2 Zellenmethode

Bei der Bahnplanung mittels Zellenmethode wird die Karte mit einem Gitter fester oder variabler Zellengröße überzogen, so dass aus der kontinuierlichen Karte eine diskrete Matrix entsteht. Jeder Zelle wird der Zustand frei oder belegt durch ein Hindernis zugeordnet. Um den kürzesten Weg zwischen Start- und Zielpunkt zu bestimmen, wird im Anschluss daran ein Algorithmus zur Flutung der Karte mit Distanzinformationen angewendet, der dabei Hindernisse auslässt und umflutet. Hierbei stellt die Zielzelle einen Brunnen dar, aus dem Wasser austritt, sich kreisförmig ausbreitet und wie eine zähe Flüssigkeit die Hindernisse umströmt. Dabei wird jeder freien Zelle, ausgehend von der Zielzelle, ein Potential als Maß für den Abstand zur Zielzelle zugewiesen. In diesem Vorgang des Flutens erhält die Zielzelle das Potential 0, alle direkten Nachbarzellen das Potential 1, wiederum die daran anschließenden Zellen das Potential 2 usw. (Bild 5.8). Zur Planung des Pfades muss schließlich nur von einer beliebigen Startzelle ausgehend die Nachbarzelle mit dem kleinsten Potential gesucht und als nächster Wegpunkt des Pfades gespeichert werden. Dies wird wiederholt, bis die Zielzelle mit dem Potential 0

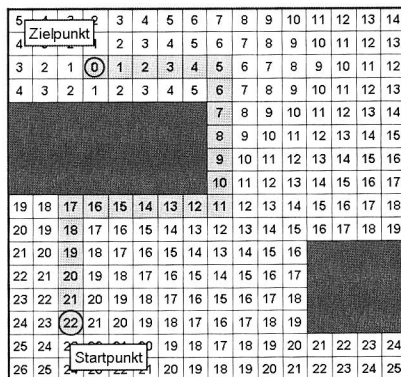


Bild 5.8: Geflutete Rasterkarte mit den Abstandswerten zum Zielpunkt. Die Navigation erfolgt entlang des Pfades mit den kleineren Zahlen.

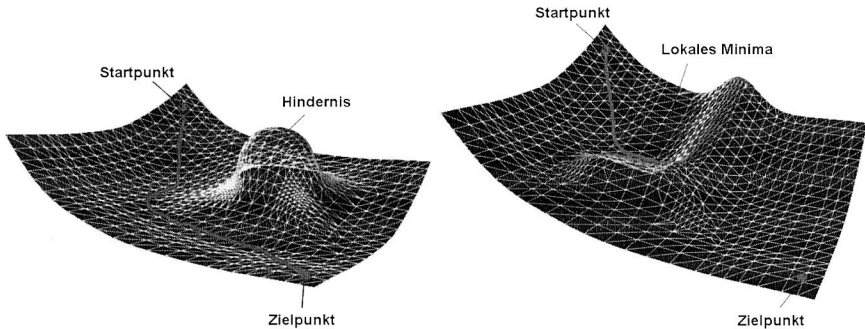


Bild 5.9: Darstellung der Bahnplanung mittels Potentialfeldmethode - Links: vom Startpunkt auf einem hohen Potential zum niedrigen Potential des Zielpunktes entlang eines Hindernisses. Rechts: Darstellung des Problems lokaler Minima, das zu Verklebungen führt. [109]

erreicht ist, wodurch sich automatisch der kürzeste Weg zwischen Start und Ziel ergibt. Zudem findet der Algorithmus immer zum Ziel, sofern es einen erreichbaren Weg gibt.

### 5.5.3 Potentialfeldmethode

Ein weiteres bekanntes Verfahren ist die Bahnplanung mittels der Potentialfeldmethode, die mit virtuellen, abstoßenden Kräften arbeitet und somit einen Zusammenstoß mit den Hindernissen verhindert. Hierbei übernimmt das fahrerlose Transportfahrzeug die Rolle eines elektrisch geladenen Teilchens, das sich in einem elektrischen Feld bewegt - abgestoßen von den gleichgeladenen Hindernissen und angezogen vom gegensätzlich gepolten Zielpunkt. Das linke Teilbild 5.9 veranschaulicht einen Pfad, der sich aufgrund der Überlagerung der Kräfte einstellt. Das Hindernis stellt hierbei einen Berg innerhalb des Potentialgefälles zwischen Start- und Zielpunkt dar. [109]

Der Algorithmus kann zur Laufzeit für jeden Punkt im Raum ein Potential berechnen. Das hat gegenüber der Zellenmethode oder der Bahnplanung mittels topologischer Karten den Vorteil, dass nicht der gesamte Raum betrachtet werden muss, um den nächsten Bewegungsschritt zu berechnen, sondern nur die nähere Umgebung. Daher handelt es sich um ein lokales Verfahren.

Der Schwachpunkt der Potentialfeldmethode liegt in der lokalen Planung, da sich bei der Überlagerung der Einzelkräfte des Zielpunktes und der Hindernisse lokale Minima des Potentials, wie im rechten Teilbild 5.9 dargestellt, einstellen können. Befindet sich das Fahrzeug in diesem Zustand, kann kein Pfad zum Ziel bestimmt werden, da es in der lokalen Umgebung keinen Punkt mit niedrigerem Potential gibt. Somit stellt sich ein Gleichgewicht zwischen abstoßender und anziehender Kraft auf das Fahrzeug ein. Um dies zu vermeiden, müssen zusätzliche Maßnahmen bzw. Ausnahmebedingungen implementiert werden. Zur Problemlösung können zum Beispiel lokale Minima künstlich aufgefüllt oder Strategien in den Algorithmus zur Erkennung und Auflösung von Verklebungen integriert werden.

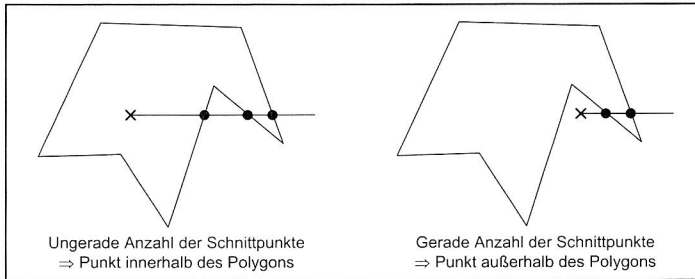


Bild 5.10: Mit der Strahlmethode wird bestimmt, ob ein Punkt innerhalb oder außerhalb eines Polygons liegt - das Verfahren ist selbst auf konkave Polygone anwendbar.

## 5.6 Flexible Bahnplanung anhand von Umgebungsinformationen

Der für die Anwendung am autonom navigierenden Fahrerlosen Transportfahrzeug eingesetzte Bahnplanungsalgorithmus basiert auf der Zellenmethode und wurde um weitere Funktionen erweitert. Zusätzlich wird eine abstoßende Wirkung der Hindernisse, ähnlich der Potentialfeldmethode, realisiert, indem die realen Hindernisse zusätzlich ausgedehnt werden. Somit kann immer ein genügend großer Abstand zum Hindernis eingehalten werden.

Voraussetzung für die Bahnplanung ist ein auf dem Rastermodell basierendes Kartenmaterial, das die Hindernisinformationen (Kapitel 5.6.1) und Übergabebahnhöfe (Kapitel 5.6.2) enthält. Hierzu werden im Folgenden geeignete Algorithmen und Berechnungsverfahren bzw. Entscheidungsregeln vorgestellt, nach denen das Andocken und Abdocken von Bahnhöfen realisiert wird. Erst danach kommt der Bahnplanungsalgorithmus basierend auf der Zellenmethode zum Einsatz (Kapitel 5.6.3), der u.a. das spezifische Fahrzeugverhalten berücksichtigen muss. Damit das Fahrzeug auch homogen der Bahn folgen kann, wird die Bahn in einem weiteren Verfahren geglättet, um nicht erforderliche Ecken zu eliminieren (Kapitel 5.6.4).

### 5.6.1 Füllen der Rasterkarte mit Zonendaten

Nach der Vermessung der Produktionshalle mit dem Handwagen (Kapitel 5.3) steht das Kartenmaterial in Form des topographischen Vektormodells (Kapitel 5.4) zur Verfügung. Ausgehend von diesem Datenbestand wird es in das Rastermodell (Kapitel 5.4) für den zellenbasierten Bahnplanungsalgorithmus (Kapitel 5.5.2) überführt. Hierbei kommt eine Rasterkarte mit uniformer Zellengröße zum Einsatz, da der Geschwindigkeitsvorteil für den schnellen Zugriff auf das Kartenmaterial wesentlicher ist, als das Einsparen von Speicherplatz bzw. von Arbeitsspeicher. Die vorliegende Implementierung im Forschungslabor mit einer Größe von ca. 50m x 15m und einer Zellengröße von 10cm benötigt lediglich 30 MB Arbeitsspeicher.

Bei der Überführung in die Rasterkarte wird ermittelt, welche Zellen durch ein Hindernis bzw. durch eine Zone belegt sind und welche nicht. Dies führt zum Punkt-im-Polygon-Problem, d.h. der Frage, ob sich ein Punkt innerhalb eines Polygons befindet oder nicht. Diese Frage ist umso komplexer zu



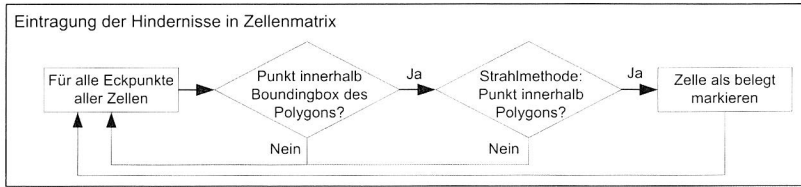


Bild 5.11: Effektive Methode zur Überführung der Hindernisse aus den Polygondaten in die Rasterkarte. Eine schnelle Vorauswahl erfolgt mittels einer Bounding-Box, die im Vorfeld für alle Hindernisse in Polygonform ermittelt wird.

beantworten, wenn konkave Polygone zum Einsatz kommen dürfen. Um selbst diese Polygonzüge berücksichtigen zu können, kommt die sogenannte Strahl-Methode zum Einsatz [111] (Bild 5.10). Dazu wird eine Halbgerade so gewählt, dass sie ausgehend von dem zu untersuchenden Punkt keinen Polygoneckpunkt schneidet. Dies kann sichergestellt werden, indem sukzessive die Koordinaten der Polygoneckpunkte in die Geradengleichung der gewählten Halbgerade eingesetzt und überprüft werden. Stellt sich dabei heraus, dass ein Eckpunkt auf der Halbgeraden liegt, so muss eine andere Halbgerade gewählt werden. Anschließend wird die Existenz eines Schnittpunktes einer jeden Teilstrecke des Polygons mit dem gewählten Strahl geprüft und dabei die Zahl der Schnittpunkte erfasst. Ergibt sich für ein Polygon eine ungerade Anzahl von Schnittpunkten, so befindet sich der untersuchte Punkt innerhalb des Polygons - bei einer geraden Anzahl an Schnittpunkten liegt der Punkt außerhalb. Zur Berechnung der Schnittpunkte werden die Geradengleichungen der Einzelstrecken mit der Halbgeraden gleichgesetzt und daraus die Koordinaten bestimmt. Da aus der Polygoneilstrecke eine Geradengleichung ohne Begrenzung abgeleitet und diese zur Berechnung des Schnittpunktes verwendet wurde, muss letztendes überprüft werden, ob der gefundene Schnittpunkt innerhalb der Strecke liegt. Um jedoch den Rechenaufwand zu minimieren, werden die relevanten Polygone mit dem Bounding-Box-Verfahren vorab ermittelt. Hierzu wird für jedes Polygon eine äußere Bounding-Box definiert, die das gesamte Polygon umfasst. Nur wenn sich der zu untersuchende Punkt innerhalb der Bounding-Box befindet, muss dieser mit der Strahlen-Methode weiter untersucht werden (vgl. Bild 5.11).

Beim Füllen der Rasterkarte wird lediglich der Level-Wert in jede Zelle aufgenommen (siehe Kapitel 5.3), der angibt, ob es sich um ein Hindernis (rote Zone), einen Alternativ-Bereich (gelbe Zone) oder einen bevorzugten Fahrbereich (grüne Zone) handelt.

Die Modellierung des Fahrzeugs als Punkt hat den Vorteil, Rechenaufwand während der Pfadplanung einzusparen. Zwar erfordert dies einen weiteren Aufwand zur Erweiterung der Hindernisse, jedoch fällt dieser Vorgang nur einmalig bei der Aufnahme eines neuen Hindernisses in die Karte an. Zudem erhalten die Hindernisse ein Abstoßungspotential, um einen ausreichenden Abstand einzuhalten. Dieser verhindert einerseits Kollisionen des FTS mit den Hindernissen und erlaubt andererseits dennoch in engen Bereichen sehr nahe an den Hindernissen vorbeifahren zu können.

Zunächst werden sämtliche Hindernisse in alle Richtungen um die halbe Fahrzeugbreite vergrößert und alle Zellen in diesem Bereich als Hindernis (rote Zone) mit dem Level-Wert 255 markiert (Bild

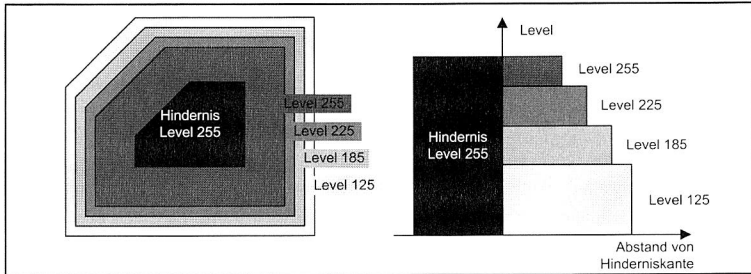


Bild 5.12: Festlegung verschieden gewichteter Abstoßungspotentiale um die Hindernisse, damit die Fahrzeuge einerseits bei verfügbarem Platz einen höheren Sicherheitsabstand zu den Hindernissen halten und andererseits bei engen Stellen dicht an den Hindernissen vorbeifahren können.

5.12). Somit wird die Kollision des Fahrzeugs, dessen Mittelpunkt zur Berechnung herangezogen wird, mit den Hindernissen verhindert. Zusätzlich werden nochmals Ringe um die Hindernisse gezogen, die abstoßende Kräfte auf das FTF ausüben, indem die zugehörigen Zellen als gelbe Zonen (Level 1 bis 254) deklariert werden. Je weiter die Zellen vom Hindernis entfernt sind, desto geringer ist der zugewiesene Level-Wert (vgl. Bild 5.12). Somit wird gewährleistet, dass Fahrzeuge selbst in engen, schmalen Gängen zwischen zwei Hindernissen genau in der Mitte hindurchfahren und dabei auf beiden Seiten den gleichen Abstand zu den Hindernissen halten.

Die Begrenzungslinien der erweiterten Hindernisse verlaufen parallel zu denen des Originals. Die davon ausgehende abstoßende Kraft, die auf die geplante Bahn wirkt, steht senkrecht zu den Begrenzungslinien. Die Steigung der abfallenden Flanken wird dabei als statischer Parameter in Form eines Arrays von Level-Werten vorgegeben (Bild 5.12). Diese stufenweise Definition ist völlig ausreichend, da die weitere Berechnung des Pfades ohnehin nur mit der Genauigkeit des Zellenrasters erfolgt. Eine kontinuierliche Definition der abstoßenden Kraft, zum Beispiel in Form einer Funktion, ist daher nicht erforderlich.

## 5.6.2 Berechnung der Andock- und Abdockpunkte

Um einen Pfad zum Bahnhof berechnen zu können, muss der Haltepunkt vor dem Bahnhof, d.h. vor dem Transportband, berechnet werden. Dieser befindet sich auf der Vorderseite des Bahnhofs und ist abhängig von der erforderlichen Stellung des Fahrzeugs. Die Fahrzeugstellung richtet sich nach der Fahrzeugtype und seinem Aufbau, abhängig an welcher Stelle die Werkstücke aufgenommen und abgegeben werden können. Darüber hinaus können Fahrerlose Transportfahrzeuge ggf. Werkstücke auf unterschiedlichen Aufnahmebändern/-rollen befördern. Hierzu ist es abhängig von der Bahnhof- und Fahrzeugauslegung erforderlich, dass das Fahrzeug sich vor dem Bahnhof unterschiedlich positioniert. Im Falle des Demonstrationsfahrzeugs besitzt dieses zwei Reihen von Aufnahmebändern und kann somit zwei unterschiedliche Stellungen vor dem Bahnhof einnehmen. Hierbei entscheidet die Job-Verwaltung an welcher Stelle und somit an welcher Rollenreihe Werkstücke aufgenommen, bzw. von welcher diese am Eingangsbahnhof wiederum abgegeben werden. Die Auftragsverwaltung gibt dem Bahnplaner einen Verschiebevektor vor, der diesen veranlasst, parallel zur Längsseite des Fahr-

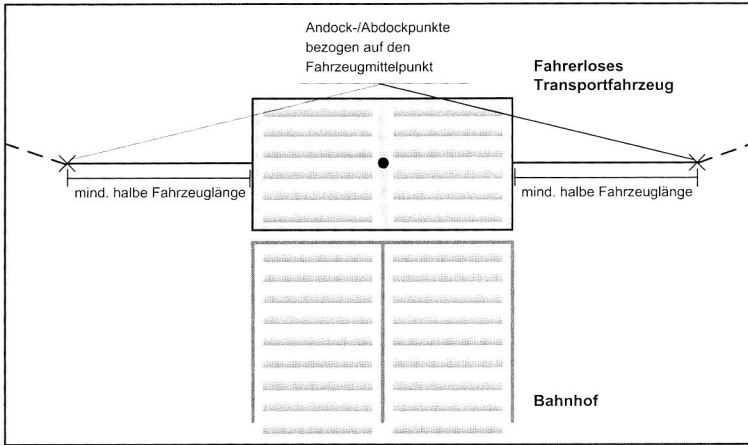


Bild 5.13: Zum Andocken in Vorwärtsfahrtrichtung muss das Fahrzeug kurz davor ausgerichtet sein, da eine Richtungskorrektur an der Bahnhofsvorderseite auf Grund des geringen Abstands nicht möglich ist.

zeugs den Mittelpunkt nach vorne oder hinten zu verschieben, um somit mit den vorderen oder hinteren Rollen des Fahrzeugs vor dem Bahnhof zum Stillstand zu kommen.

Zudem ist es erforderlich, neben dem Haltepunkt weitere Hilfspunkte zu definieren, die wesentlich für ein An- und Abdocken am Bahnhof sind (Bild 5.13). Das Fahrzeug entfernt sich vorwärts parallel zum Bahnhof bzw. dockt ebenso in der Vorwärtsfahrt an diesem an. Da der Abstand zwischen dem FTF und dem Bahnhof nur wenige Zentimeter beträgt, darf sich das Fahrzeug vor dem Bahnhof nicht mehr drehen. Daher muss es beim An- bzw. Abdocken parallel zum Bahnhof ausgerichtet sein. Um dies zu erreichen, muss ein An- und ein Abdockpunkt für die Bahnhöfe berechnet werden. Somit befinden sich die Andock- und Abdockpunkte parallel zum Bahnhof auf gleicher Höhe zum Haltepunkt (Bild 5.13). Auf welcher Seite angedockt oder abgedockt wird, hängt von der Richtung ab, aus der das Fahrerlose Transportfahrzeug den Bahnhof anfährt bzw. in welche Richtung es zum Erreichen des nächsten Bahnhofs weiterfahren muss.

Dabei muss der Bahnplaner entscheiden, über welche Seite das Fahrzeug ab- bzw. andockt. Die Entscheidung trifft dieser anhand des kürzesten Abstands der Hilfspunkte zum nächsten Ziel bzw. Bahnhof. Im Anschluss daran muss der Bahnplanungsalgorithmus die Bahn zwischen dem Hilfspunkt des aktuellen Bahnhofs, an dem sich das Fahrzeug befindet, und demjenigen des nächsten Zielbahnhofs berechnen.

### 5.6.3 Ermittlung des Pfades

Nach der Ermittlung der Hilfs- und Haltepunkte kommt der Bahnplanungsalgorithmus zum Einsatz, der basierend auf der mit den Zonen vorbelegten Karte den kürzesten Weg vom Hilfspunkt des Ausgangsbahnhofs bzw. vom aktuellen Standort zum Hilfspunkt des Zielbahnhofs bestimmt. Zunächst

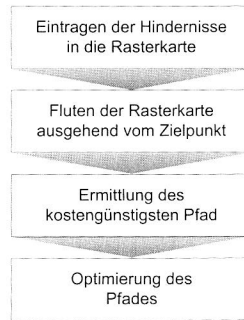


Bild 5.14: Prinzipieller Ablauf des Bahnplanungsalgorithmus, der um die Berechnung und Bewertung der Wegkosten beim Fluten und um ein Pfadoptimierungsverfahren erweitert wurde.

muss dieser erkennen, ob sich das Fahrzeug noch vor einem Bahnhof befindet. Hierbei verlässt sich der Algorithmus nicht auf persistent gespeicherte Historiendaten, die Aufschluss darüber geben, an welchem Bahnhof das Fahrzeug zuletzt angedockt ist, da es möglich ist, das Fahrzeug im ausgeschalteten Zustand per Hand an eine neue Position zu bewegen. Befindet sich im nahen Umkreis ein Bahnhof, so wird ein Hilfspunkt zum Abdocken berechnet. Ist jeglicher Bahnhof weit genug entfernt, so kann die Fahrt unmittelbar ohne ein Abdock-Manöver fortgesetzt werden.

Basierend auf der mit den Zonen gefüllten Rasterkarte und den Hilfspunkten, kann der Bahnplanungsalgorithmus den kürzesten Weg berechnen. Hierzu kommt, wie in Kapitel 5.5.2 beschrieben, die geflutete Rasterkarte mit Distanzinformationen (Bild 5.15) zum Einsatz. Um jedoch die unterschiedliche Gewichtung der Zonen zu berücksichtigen und somit die resultierenden Kosten alternativer Wege vergleichen zu können, ist die Erweiterung des Algorithmus zum Fluten der Rasterkarte erforderlich (Bild 5.14). Anschaulich ist dies mit einer Mautgebühr vergleichbar. In bestimmten Bereichen fällt die minimale Mautgebühr pro Streckeneinheit an (grüne Zone), in anderen Bereichen (gelbe Zone) schlagen unterschiedlich hohe Gebühren zu Buche. Während eines Weges summiert sich die Mautgebühr über die unterschiedlich gewichteten Streckeneinheiten auf. Daher werden in jede Zelle diese aufsummierten Kosten eingetragen, die Auskunft darüber geben, mit welchen geringsten Kosten von dieser Zelle zur Zielzelle zu kalkulieren ist.

Somit wird in jeder Zelle nicht nur der Abstand der Zelle zur Ziel-Zelle vermerkt, sondern auf dem Weg dahin alle Level-Werte der Zonen aufsummiert und diese als Distanz- bzw. Kostenwert in die Zelle eingetragen (vgl. Bild 5.16). Folglich ist in jeder Zelle vermerkt, wie hoch die sogenannten „Kosten“ zur Ziel-Zelle sind. Diese Kosten geben somit einen Aufschluss darüber, in welchem Umfang die gelbe Zone als Alternativzone bei Nutzung des Pfades befahren werden muss. Dabei kann u.a. genau kalkuliert werden, ob es sich lohnt einen kürzeren Weg durch eine gelbe Zone zu nehmen, als einen längeren Umweg, der nur durch grüne Zonen führt.

Um jedoch sicherzustellen, dass in jeder Zelle wirklich der geringste Kostenwert eingetragen ist, muss beim Fluten zudem beachtet werden, dass im Falle einer Änderung einer Zelle auch sämtliche Nach-

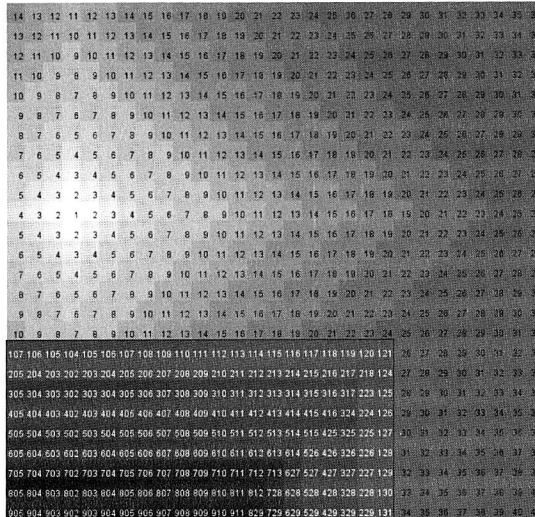


Bild 5.15: Beispiel einer gefluteten Rasterkarte mit den eingetragenen Kostenwerten in den Zellen, die den Weg zum Zielpunkt bewerten. Am unteren Bildrand sind die Kostenwerte höher, da in diesem Bereich das Abstoßungspotential eines Hindernisses wirkt.

barzellen überprüft werden müssen. Dies liegt darin begründet, dass alle Nachbarzellen von der veränderten aktuellen Zelle abhängig sind und eventuell ein niedriger Kostenwert für die benachbarten Zellen aus der Veränderung resultieren kann. Als Beispiel kann eine Zelle durch eine erste Flutung von der einen Seite und in der nächsten Flutungswelle von der anderen Seite mit Kostenwerten konfiguriert werden. Zudem kann es passieren, dass sich Zellen im Schattenbereich von Hindernissen befinden und in diese im ersten Durchlauf keine Kostenwerte eingetragen werden. Folglich sind mehrere Durchläufe erforderlich, um die Bearbeitung aller Zellen sicherzustellen. Werden die Zellen im nächsten Durchlauf bearbeitet, können diese eventuell einen geringeren Kostenwert tragen, als bereits konfigurierte Nachbarzellen aus früheren Läufen. In diesem Fall würde sich der Kostenwert der bereits gefluteten Nachbarzellen vermindern. Um sicherzustellen, dass diese Fälle berücksichtigt und solch angrenzende Zellen nochmals überprüft werden, wird ein Flag in jeder Zelle eingeführt, das angibt, ob diese nochmals zu überprüfen ist. Dieses Flag wird jedesmal dann gesetzt, wenn der Kostenwert einer Nachbarzelle verändert wurde. Dies kann u.U. wiederum weitere Flutungen auslösen. Ist jedoch der aktuelle Kostenwert einer Zelle bereits der geringste Wert im Vergleich zu den Kostenwerten der Nachbarzellen, dann wird das Flag bei den Nachbarzellen nicht mehr gesetzt und die „Flutwellen“ kommen zum Erliegen. Es sollte sichergestellt werden, dass dieser Vorgang solange wiederholt wird, bis keine Veränderung an der Matrix mehr stattfindet und es sich somit um das optimale Ergebnis handelt (Bild 5.17). Um jedoch aus Zeitgründen diesen Vorgang zu begrenzen, wird im vorliegenden Algorithmus nur eine maximale Obergrenze zur Wiederholung der Flutungsdurchläufe zugelassen.

```

repeat = 255
finished = false

// toChange-Flag aller Zellen auf TRUE setzen
setToChangeFlagOfAllCells(TRUE)

WHILE finished = false and repeat >= 0
    repeat --
    finished = true
    // Über alle Zeilen iterieren
    FOR row = 1 to MatrixHeight
        // Über alle Spalten iterieren
        FOR col = 1 TO MatrixWidth
            IF toChangeFlag(row, col) = TRUE THEN
                nbCostValue = getLowestCostValueofNeighbourCells(row, col)
                IF nbCostValue > 0 THEN
                    oldCostValue = getCostValue(row, col)
                    // Kosten berechnen sich aus den niedrigsten Kosten der Nachbarzellen
                    // zusammen mit dem Level-Wert der aktuellen Zelle plus eins
                    newCostValue = nbCostValue + getLevelValue(row, col) + 1
                    IF oldCostValue = 0 OR newCostValue < oldCostValue THEN
                        setCostValue(row, col, newCostValue)
                        setToChangeFlag(row, col, FALSE)
                        setToChangeFlagOfNeighbourCells(row, col, TRUE)
                        finished = false // Vormerkung, um ein weiteres mal zu iterieren
                    END IF
                END IF
            END IF
        END FOR
    END FOR
END WHILE

```

Bild 5.16: Erweiterter Algorithmus zum Fluten der Rasterkarte, der statt den Distanzwerten die Kostenwerte in jeder Zelle festhält. Zugleich muss der Algorithmus beim Ändern der Kostenwerte sicherstellen, dass die Nachbarzellen im nächsten Durchlauf geprüft bzw. falls erforderlich, neu berechnet und nicht untersuchte Schattenbereiche hinter Hindernissen miteinbezogen werden.

Nach dem Fluten erfolgt die Pfadermittlung. In diesem iterativen Prozess sucht der Algorithmus ausgehend von der Zielzelle in der Rasterkarte die nächste Zelle mit dem geringsten Kostenwert. Somit entscheidet sich der Algorithmus automatisch für den günstigsten Weg. Dabei wird implizit sichergestellt, dass das Fahrzeug möglichst die grünen Zonen bevorzugt und nur in Ausnahmefällen bzw. abhängig von der Länge alternativer Umwege die gelben Zonen befährt. Existiert beispielsweise ein langer Umweg in der grünen Zone, der jedoch auf Grund der großen Länge kostenaufwändiger ist, als ein direkter Weg durch eine gelbe Zone, so wird der Weg durch die gelbe Zone bevorzugt.

### 5.6.4 Optimierung des Pfades

Der aus dem Bahnplanungsalgorithmus resultierende Pfad ist eine Menge von Zellen, die den Weg vom Ausgangspunkt zum Zielpunkt weisen. Jedoch stellt diese hohe Genauigkeit auch Nachteile bei der Pfadverfolgung dar, da nach sehr kurzen zurückgelegten Strecken ständig ein neuer Teilstreckenpunkt zur Richtungsberechnung ausgewählt werden muss. Zudem bestehen im Pfad kleine, scharfe Kurven, die selbst an völlig freien Stellen im Raum auftreten können. Dies ist bedingt durch die eckige Pfadrepräsentation im Rastermodell, in dem Kurven mit beliebiger Steigung nur grob gerastert werden können. Durch diese kleinen, scharfen Kurven ist nicht auszuschließen, dass das Fahrzeug Schlingerbewegungen in der Pfadverfolgung ausführen wird. Um solche Schlingerbewegungen zu vermeiden, besteht im folgenden Schritt das Ziel, die Bahn zu glätten und dabei die Datenmenge des

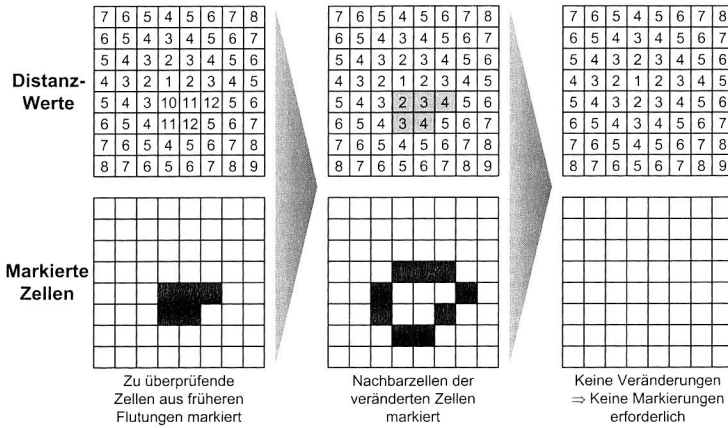


Bild 5.17: Beispielhafte Darstellung der Rasterkarte mit dem *toChange*-Flag, das zur Markierung der Zellen für die Nachkontrolle der Kostenwerte verwendet wird.

ermittelten Pfades auf die unbedingt erforderlichen Eckpunkte zu reduzieren (Bild 5.18). Somit wird eine einfachere und stabilere Verarbeitung der Pfaddaten in der Pfadverfolgung gewährleistet.

Hierzu wird ein iterativer Algorithmus angewendet, der sukzessive alle Zellenmittelpunkte des im vorherigen Schritt ermittelten Pfades durchläuft und dabei die Zahl der Eckpunkte minimiert. Bei diesem Algorithmus kommen zwei Zeiger zum Einsatz (Bild 5.19). Der erste verweist auf den Start-Punkt der zu betrachtenden Strecke (A), der zweite auf den End-Punkt (B). Zu Beginn wird der Start-Punkt auf den ersten Punkt des Pfades gesetzt und der End-Punkt auf den zweiten Punkt. Im Anschluss werden drei Strecken definiert. Die Erste verbindet die beiden Punkte, die beiden anderen verlaufen parallel zur ersten in einem definierten Abstand, d.h. in der halben Fahrzeugbreite, zu beiden Seiten angeordnet. Somit wird ein Korridor definiert, der genau der Breite des Fahrzeugs entspricht. Die Aufgabe des Algorithmus besteht darin, mögliche Kollisionen des Korridors mit Hindernissen zu untersuchen. Zur Erfüllung dieser Aufgabe werden auf den drei Strecken Hilfspunkte mit einem vorgegebenen Abstand definiert. Für jeden dieser Punkte wird lediglich geprüft, ob sie sich in einer als Hindernis, d.h. roten Zone, deklarierten Zelle befinden. Dies kann mit nicht rechenaufwendigen Vergleichsoperationen bewerkstelligt werden, wodurch selbst lange Strecken und zahlreiche Hilfspunkte keine relevanten Geschwindigkeitseinbußen zur Folge haben. Schneidet der Korridor kein Hindernis, wird der Zeiger des Streckenendpunktes auf den nächsten Punkt des Pfades gesetzt und ein neuer Korridor mit den neuen Streckenpunkten definiert und überprüft. Befindet sich jedoch ein Hilfspunkt in einem Hindernis, wird die letzte befahrbare Strecke, deren Korridor-Prüfung erfolgreich war, in den neuen Pfad aufgenommen und der End-Punkt dieser Strecke als Start-Punkt der nächsten Strecke gesetzt.

Somit steht ein effizientes Optimierungsverfahren zur Verfügung, um die zahlreichen Eckpunkte in den Pfaden zu eliminieren. Die gesamte Berechnung der Pfadplanung, d.h. sowohl die Bahnplanung

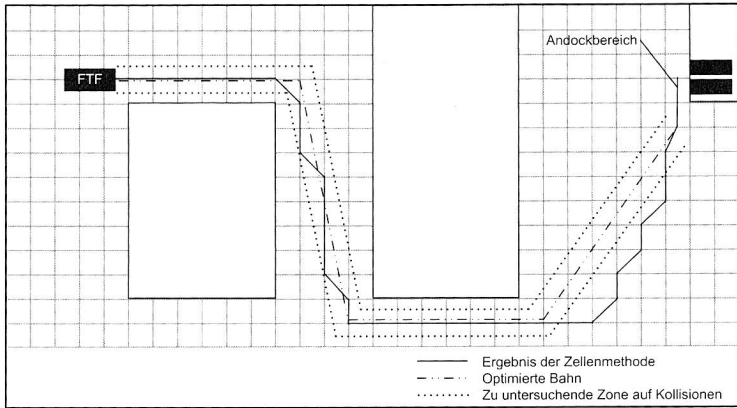


Bild 5.18: Beispielhafte Darstellung zweier Pfade als Ergebnis der Rastermethode und der Optimierung. Letztere bietet eine Reduktion der Pfadknotenpunkte, eine Glättung und in bestimmten Fällen eine Verkürzung des Pfades.

mittels Zellenmethode, als auch die anschließende Optimierung des Pfades, erfordert auf einem Pentium IV-PC bei einer Karte von 50m x 15m und einer Zellengröße von 10cm weniger als 1 Sekunde Berechnungszeit. Somit kann dieser Ansatz, selbst auf leistungsschwächeren PCs in den Fahrzeugen effizient angewendet werden, da selbst Berechnungszeiten von mehreren Sekunden vertretbar sind. Die Bahnplanung kann vorausschauend für den nächsten Auftrag, z.B. während der Werkstückaufnahme oder -abgabe am Bahnhof, die i.d.R. mindestens 8 Sekunden dauert, durchgeführt werden.

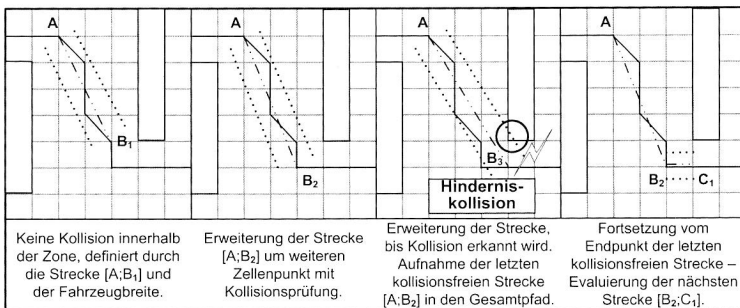


Bild 5.19: Zur Ermittlung der Streckenabschnitte werden mögliche Kollisionen mit dem Fahrzeug geprüft. Hierzu werden parallel Hilfslinien mit Hilfspunkten auf beiden Seiten gezogen, deren Punkte in keinem Hindernis enthalten sein dürfen.



## Kapitel 6

# Agentenbasierter Materialfluss zur Koordinierung der Bearbeitungs- und Transportvorgänge

Basierend auf dem agentenbasierten Systemmodell aus Kapitel 3 und der Integration autonom agierender Fahrerloser Transportsysteme in den Anlagenverbund, werden im Folgenden die Komponenten des dezentral agierenden Materialflusssystems und ihre Steuerungskomponenten eingehend erläutert. Diese genügen zum einen den Anforderungen an eine flexible Integration der Anlagenkomponenten selbst zur Laufzeit und gewährleisten somit die selbstadaptive Rekonfigurierbarkeit. Zum anderen erschließt sich mit dem agentenbasierten Ansatz der Aspekt der Selbstorganisation, bei dem die autonomen Komponenten, vertreten durch die Agenten, direkt miteinander in Verbindung treten und zur Laufzeit selbstständig Entscheidungen zur Gewährleistung und Optimierung des Materialflusses treffen.

Zur Veranschaulichung des Systemverhaltens wird dabei u.a. auf das Verhalten des Werkstückagenten im Verbund mit den Stationen- bzw. Bahnhofagenten eingegangen (Kapitel 6.1). Der Bahnhof als transparente Schnittstelle zwischen der Bearbeitungsstation und dem Transportsystem stellt das wesentliche Bindeglied zwischen Anlagenkomponenten unterschiedlicher Technologie dar, indem dieser standardisierte Schnittstellen und standardisiertes Verhalten vorweist (Kapitel 6.4). Voraussetzung für die Umsetzung des verteilten Steuerungsansatzes sind diverse Serverdienste, die Informationen bereitstellen bzw. Produkt- und Produktionsdaten sammeln (Kapitel 6.3). Die hier vorgestellten Konzepte wurden in einer Softwarearchitektur umgesetzt und zum einen in einem IT-Labor mit 23 verteilten Rechnersystemen und zum anderen im Labor mit realen Anlagensystemen angewendet und validiert (Kapitel 7.4).

## 6.1 Koordinierung der Werkstückbearbeitung durch den Werkstückagenten

Der Werkstückagent stellt ein wesentliches Bindeglied bei der Organisation von Werkstücken im Materialfluss und der Zuordnung der anfallenden Bearbeitungsschritte zu den Ressourcen dar. Im Folgenden werden das Verhalten und die Aufgaben des Werkstückagenten eingehend erläutert. Zudem werden Schnittstellen zwischen diesem und den restlichen Komponenten der Anlage beschrieben, die mittels normierter Schnittstellen und normiertem Kommunikationsverhalten ad hoc Verbindungsaufbauten ermöglichen. Somit wird der Anforderung nach der einfachen und schnellen Inbetriebnahme gemäß dem Plug&Produce-Ansatz Genüge getan.

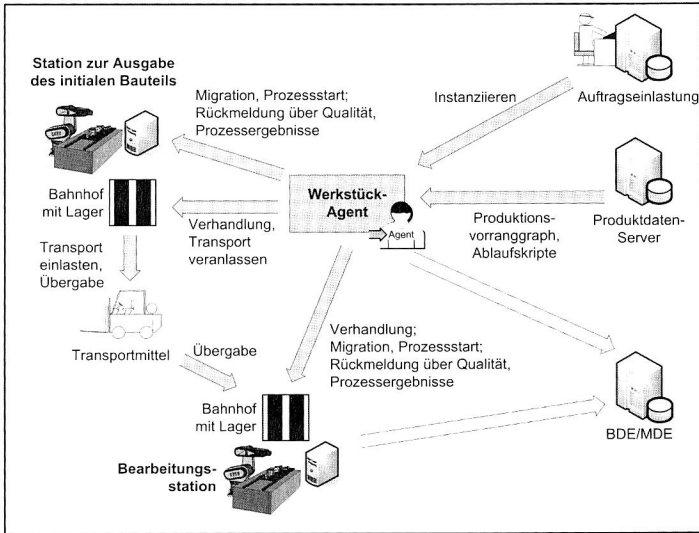


Bild 6.1: Interaktionen des Werkstückagenten mit anderen Teilnehmern im Agentenverbund. Abruf und Ablage anlagen- und produktionsspezifischer Daten an zentralen Servern.

### 6.1.1 Aufgaben des Werkstückagenten

Der Werkstückagent repräsentiert ein einzelnes Werkstück oder eine Sammlung von Baugruppen die sich gemeinsam auf einem Werkstückträger befinden. Der Werkstückagent begleitet das Werkstück bzw. die Baugruppen solange durch die Produktion bis dieses mit einem anderen, z.B. in der Montage, verbunden wird. In diesem Fall übernimmt das Werkstück die Rolle eines zu einer Produktionslinie zugeführten Bauteils, das mit dem auf der Produktionslinie gefertigten Hauptbauteil physisch verbunden wird. Parallel zum Montageprozess findet eine Aggregation der Zustandsdaten zweier oder mehrerer Werkstückagenten-Instanzen statt, wodurch lediglich der Werkstückagent des Hauptbauteils der Fertigungslinie weiter besteht. Zum Vergleich hierzu wird auf Kapitel 3.8.6 verwiesen, das die beschriebenen Produktionsstrukturen, als Kombination von Bauteilen an Bearbeitungsstationen mit angeordneten Bauteillägern vorsieht.

Die Aufgabe des Werkstückagenten besteht darin, alle Schritte, die das Werkstück in der Produktion durchlaufen muss, zu begleiten und zugleich zu initiieren. Im Gegensatz zu bisherigen, industriell verfügbaren Systemen, initiiert hierbei nicht die Fertigungszelle oder ein Leitrechner die Prozesse an den Bearbeitungsstationen, sondern der Werkstückagent. Er überwacht und koordiniert zudem den weiteren Verlauf des Werkstücks durch die Produktion.

Zur Modellierung verteilter, autonomer Komponenten müssen globale Ziele und das globale Verhalten der Gesamtanlage mittels Dekomposition in kleine, abgeschlossene Teilaufgaben mit spezifischem Verhalten zerlegt werden. Diese werden in den Agenten implementiert, die an die Komponenten an-

gebunden sind. Sie leisten für die Komponenten repräsentative Aufgaben und müssen zur Erfüllung der Gesamtaufgabenstellung zusammenarbeiten. Dabei stehen sie ständig in Kontakt und interagieren direkt miteinander. Die Identifizierung der Komponenten wurde bereits in Kapitel 3 vorgenommen. In Bild 6.1 sind die erforderlichen Interaktionen des Werkstückagenten mit den anderen Teilnehmern im Überblick dargestellt. Zunächst wird der Werkstückagent von der Auftragseinlastung instanziiert, mit den erforderlichen Produktdaten vorbelegt und zum selbständigen Handeln aktiviert. Auf zentrale Dienste hat der Werkstückagent jederzeit Zugriff - unabhängig auf welchem Rechner bzw. welcher Bearbeitungsstation er sich befindet.

Vom Produktdaten-Server erhält der Werkstückagent nach Durchführung eines Bearbeitungsschrittes den nächsten Schritt inklusiv aller Prozessparameter und ggf. Detailanweisungen. Zudem werden auf diesem alle in der Produktion erforderlichen Ablaufskripte gespeichert, damit diese bei Bedarf heruntergeladen werden können. Dies bietet sich insbesondere in Umgebungen mit zahlreichen redundanten Arbeitsstationen an, um den manuellen Konfigurationsaufwand an jeder einzelnen Station zu vermeiden bzw. zu reduzieren. Mit den Prozessspezifikationen wendet sich der Werkstückagent an den lokalen Katalogdienst, der an den angebotenen Bahnhöfen vorliegt und eine lokale Sicht der Anlagenstruktur, d.h. zu den am dahinterliegenden Transportsystem angebotenen Stationen, enthält. Die Adressen und Fähigkeiten einer jeden Entität bzw. Ressource erhält der lokale Katalogdienst aus der Strukturerkennung auch über die Grenzen der Übergabebahnhöfe (vgl. Kapitel 3.8.4) bzw. Separatoren (vgl. Kapitel 4.2) hinaus. Der Katalogdienst liefert eine Liste der Bearbeitungsstationen, die von der aktuellen Station erreichbar sind und eine bestimmte Tätigkeit durchführen können. Mit diesen Daten ist der Werkstückagent bzw. sein Stellvertreter, der Verhandlungsagent, in der Lage, die nächsten Schritte, wie Verhandlungen und Veranlassung des Transports, durchzuführen. Während der gesamten Produktionsdauer des Werkstücks bekommt der Werkstückagent Rückmeldung über den Erfolg bzw. die erzielte Qualität. Diese Daten können von ihm sowohl zur Planung des weiteren Fortgangs, als auch zur Protokollierung der Produktherstellungsprozesse herangezogen werden. Sie werden zudem an einen zentralen BDE-Server weitergeleitet, der mittels eindeutiger Serien-Nr. die Daten produktspezifisch ablegen kann.

Der Werkstückagent ist zudem in der Lage zu migrieren. Im Idealfall besteht die Möglichkeit, diesen zu serialisieren und auf einem internen Speicher, z.B. einem RFID-Tag, am Werkstück oder Werkstückträger abzulegen, damit er bei Bedarf wieder heruntergeladen und zur Ausführung gebracht werden kann. Steht kein beschreibbarer Speicher zur Verfügung, besteht die Möglichkeit diesen über das bestehende Kommunikationsnetzwerk migrieren zu lassen. Hierbei ist lediglich die eindeutige Markierung des Werkstücks mit einer eindeutigen ID, z.B. in Form eines Strichcodes oder eines lesbaren RFID-Tags, erforderlich. Über diese ID kann die Zuordnung des Werkstücks zum zugehörigen Agenten erfolgen.

Zu Beginn der Herstellung einer Baugruppe besteht die Aufgabe der Auftragseinlastung an der ersten Bearbeitungsstation darin, ein initiales Bauteil, ggf. auf einem Werkstückträger, in die Produktion einzuschleusen und diesem den instanziierten Werkstückagenten zuzuweisen. Zusammenfassend übernimmt dieser dabei an jeder der darauffolgenden Bearbeitungsstationen folgende Aufgaben:

1. Prozessinitialisierung: Der Werkstückagent kennt alle durchzuführenden Tätigkeitsschritte bis zur Fertigstellung des Bauteils. Hierbei stößt der Werkstückagent die Bearbeitung mit allen zugehörigen Prozessparametern an und erhält zugleich Rückmeldung zum Erfolg der Verarbeitung

und zur Qualitätsanalyse.

2. Der Abschluss des Prozesses wird dem Werkstückagenten mitgeteilt, worauf dieser zunächst den nächsten Bearbeitungsschritt ermittelt.
3. Anhand des Bearbeitungsschrittes kann er alle relevanten Bearbeitungsstationen, die die erforderliche Tätigkeiten ausführen können, ermitteln.
4. Mittels Verhandlung werden die nächste Station ermittelt bzw. die Vorabplanungen nochmals geprüft. Die Verhandlung läuft rekursiv über den bzw. die Bahnhöfe der lokalen Station. Dabei werden alle möglichen Kombinationen des Transports mit den relevanten Bearbeitungsstationen berücksichtigt.
5. Zusage an das günstigste Tupel bestehend aus Bearbeitungsstation und Transportsystem. Die Zusage des Transportsystems erfolgt über den zugehörigen Bahnhof des ausgewählten Transportmittels. Da der Bahnhof die Schnittstelle zum Transportsystem darstellt und aus Gründen der Modularität die Art des Transports gegenüber dem Werkstück verbirgt, stellt er sowohl auf Seiten der physischen Werkstückübergabe als auch auf Seiten der Organisation des Transports das zentrale Bindeglied dar.
6. Die Freigabe des Werkstücks an der Bearbeitungsstation und die Veranlassung zur Zuführung zum ausgewählten Bahnhof erfolgt ebenfalls durch den Werkstückagenten.
7. Abschließend veranlasst er die direkte oder indirekte Migration zur nächsten Bearbeitungsstation. Im ersten Fall erfolgt dies über das Kommunikationsnetzwerk, im zweiten Fall über beschreibbare Speicherträger am Werkstück, z.B. RFID-Tags.
8. Sobald das physische Werkstück an der Zielstation angekommen ist, wird der zugehörige Werkstückagent informiert. Ebenso wenn das Werkstück als nächstes bearbeitet wird, damit dieses die Prozessschritte und deren Parameter festlegt.

### 6.1.2 Verhandlungsaufgaben

In der Navigation des Werkstücks wird spezifiziert, an welchen konkreten Bearbeitungsstationen es bearbeitet wird und wie der Transport zu diesen erfolgt. Es gibt zwei grundlegende Möglichkeiten den Werkstückträger mit zugehörigem Agenten durch die Produktion zu navigieren. Entweder wird der Weg im Vorfeld geplant und somit dem Werkstück vorgegeben oder das Werkstück bewegt sich ohne Planung durch die Anlage und entscheidet zur Laufzeit nach jedem abgeschlossenen Schritt die weitere Vorgehensweise. Zudem sind verschiedene Varianten zwischen diesen beiden grundlegenden Verfahren als Zwischenlösungen, z.B. mit Teilplanungen der nächsten  $n$  Bearbeitungsschritte, möglich.

Im ersten Fall prüfen Verhandlungsverfahren zur Laufzeit zwischen dem Werkstückträger und den Ressourcen lediglich den vorab geplanten Weg durch die Anlage und gleichen die geplanten Schritte bzw. Teilziele mit den Ist-Zuständen, d.h. den Werten des oder der unmittelbar bevorstehenden Schritte, ab. Sollte es hierbei zu maßgeblichen Abweichungen kommen, können weitere Maßnahmen entweder von zentraler Stelle eingeleitet werden oder der Werkstückagent ist mit intelligenten

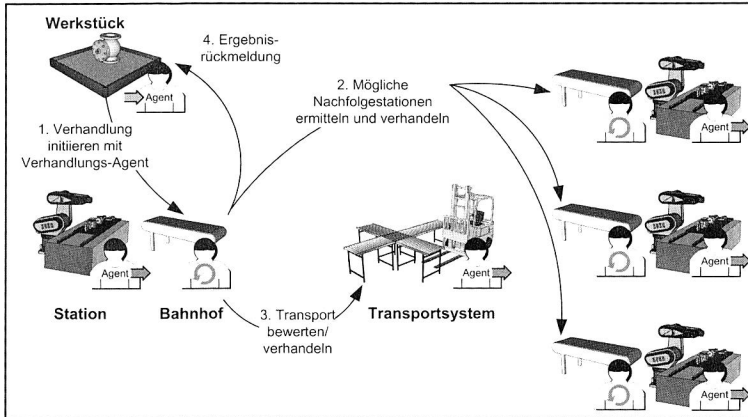


Bild 6.2: Der Übergabebahnhof bildet die Schnittstelle zwischen dem Transportsystem und der Bearbeitungsstation und übernimmt eine Vermittlerrolle zur Verhandlung und Bestimmung des Transportsystems und der Nachfolgestation. Gemäß dem Prinzip der Kapselung kennt nur er die erreichbaren Nachfolgestationen und gibt diese Daten dem migrierenden Verhandlungsagenten weiter.

Mechanismen zur Störkompensation ausgestattet. Hierbei kann er sich z.B. selbständig alternative Ressourcen suchen und diese für die Bearbeitung auswählen.

Alternativ zur Planung der Belegzeiten an den einzelnen Ressourcen, besteht die Möglichkeit, insbesondere in sehr flexiblen und dynamischen Umgebungen das Werkstück ohne Vorgabe eines geplanten Weges durch die Produktionsanlage laufen zu lassen. Der zugehörige Agent hat nach jedem Verarbeitungsschritt somit die Aufgabe, den nächsten Bearbeitungsschritt und passend zu diesem die technologisch geeigneten Bearbeitungsstationen zu ermitteln. Aus dieser Menge möglicher Stationen muss als nächstes die günstigste Station ermittelt werden. Hierzu initiiert der Werkstückagent eine Gebotsanfrage über die Übergabebahnhöfe der aktuellen Bearbeitungsstation, indem er einen Verhandlungsagenten beauftragt (Bild 6.2). Der Verhandlungsagent kennt die durchzuführende Tätigkeit und initiiert basierend auf der Liste der erreichbaren Stationen die geeigneten Stationen aus. Diese Liste liegt nur an den jeweiligen Bahnhöfen vor, um gemäß der Kapselung die einfache Handhabbarkeit der Daten zu gewährleisten. Neben der Verhandlung mit den potentiellen Stationen, wird die Bewertung des Weges durch gezielte Anfragen zu den möglichen Zielstationen initiiert. Die Schätzung des Transportaufwands kann in Kosten und Zeit vorgenommen werden. Die Station selbst gibt ein Gebot ab, wann der Bearbeitungsschritt für das Werkstück nach dem aktuellen Stand der Auslastung abgeschlossen sein kann. Aufsummiert zu den Kosten und der Zeit des Transports, kann die günstigste Station ausgewählt werden. Ist die Verhandlung nicht nur des nächsten, sondern der nächsten  $n$  Schritte gewünscht, um im Vorfeld bessere Aussagen über die nahe Zukunft treffen zu können, kann der Verhandlungsagent rekursiv zur ausgewählten Zielstation migrieren (Bild 6.3). Am Ziel der rekursiven Kette angekommen, wandert dieser wiederum auf demselben Weg zur Anfrage zurück. Dem Werkstückagent steht u.a. eine Menge von Geboten zur Verfügung, die bereits von den Verhandlungsagenten vorgefiltert wurden. Aus mehreren Verhandlungsagenten wird die günstigste

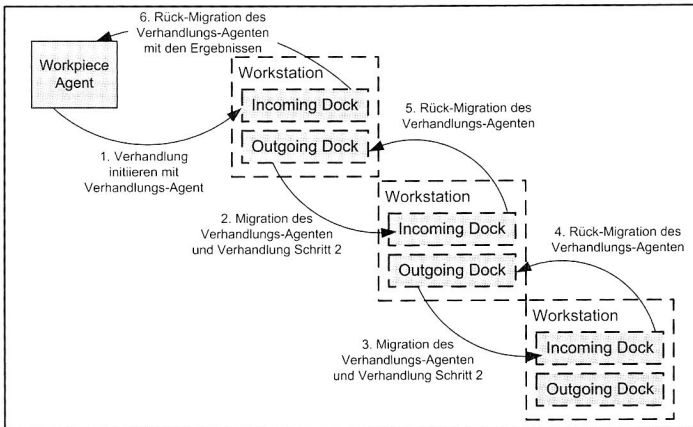


Bild 6.3: Der Werkstückagent initiiert den Verhandlungsagenten mit dem Ziel eine Vorauswahl potentiell geeigneter Wege zu ermitteln. Dieser kann zwischen den Bahnhöfen migrieren, um eine höhere Verhandlungstiefe zu gewährleisten.

oder schnellste Bearbeitungsstation ausgewählt. Da die Verhandlungsmethoden und -algorithmen in großem Maße von den Anforderungen der Anwendung abhängen, wird an dieser Stelle nicht weiter auf die Umsetzung des Verhandlungsverfahrens eingegangen. Die vorliegende Architektur bietet jedoch die Möglichkeit unterschiedlichste Verhandlungsstrategien zu implementieren.

### 6.1.3 Interaktion zwischen Werkstückagenten und Bearbeitungsstation

An jeder Bearbeitungsstation verwaltet der Werkstück-Manager alle Werkstücke, die an der lokalen Station eingetroffen sind. Er nimmt die Werkstückagenten auf und referenziert ihre Instanzen über eine interne Liste. Entweder werden die Werkstückagenten von einem beschreibbaren Speicher am Werkstück ausgelesen und instanziiert oder sie migrieren über das Kommunikationsnetzwerk zwischen den Stationen. Letzterer Fall ist nur dann erforderlich, wenn kein beschreibbares Speichermedium am Werkstück mit ausreichender Speicherkapazität vorhanden ist. In diesem Fall ist zur Erhöhung der Datensicherheit vor Ausfällen die Replizierung an einen zentralen Werkstück-Server empfehlenswert, der selbst im Falle eines Stationsausfalls die aktuellen Zustandsdaten des Werkstücks bzw. Werkstückagenten enthält und diese bei Bedarf abgerufen werden können. Zur Migration sowohl über einen beschreibbaren Speicher als auch über das Kommunikationsnetzwerk, bietet der Werkstück-Manager Verwaltungsaufgaben, wie die Serialisierung und Deserialisierung der Werkstückagentenzustände, an.

Von den Eingangs-Bahnhöfen wird der Werkstück-Manager darüber hinaus über das Eintreffen der physischen Werkstücke informiert (vgl. Kapitel 6.4.2) und leitet diese Informationen an den zugehörigen Werkstückagenten weiter (vgl. Bild 6.4). Wird ein Werkstück als nächstes für die Bearbeitung herangezogen, informiert zunächst die Maschinen-Steuerung den Werkstück-Manager, der wiederum die Benachrichtigung an den zugehörigen Werkstückagenten weiterleitet, da er die Zuordnung der

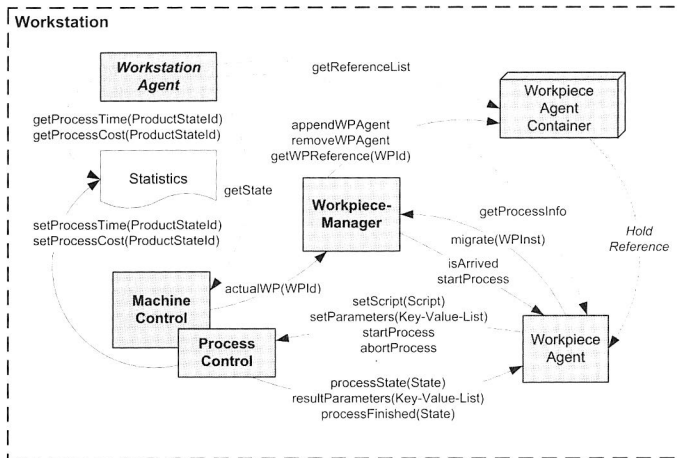


Bild 6.4: Infrastruktur in der Bearbeitungsstation zur Aufnahme und Ausführung der Werkstückagenten - der Werkstückmanager verwaltet die Werkstückagenten und unterhält gleichzeitig Verbindungen mit der Maschinen- und Prozesssteuerung.

Werkstück-ID zum Werkstückagenten vornehmen kann. Der Werkstückagent besitzt sowohl Zugriff auf die Maschinen-Steuerung (*MachineControl*), als übergeordnete, koordinierende Instanz der Bearbeitungsstation, als auch auf die Prozesssteuerung zur Initiierung der Prozessschritte. Hierbei ist der Werkstückagent in der Lage, alle zur Durchführung des Prozesses erforderlichen Parameter der Prozesssteuerung vorzugeben. Diese kann der Werkstückagent automatisiert aus dem für eine spezifische Produktvariante konfigurierten Produktionsvorranggraphen (Kapitel 6.2.1) entnehmen. Analog zu den Produktdaten kann ebenso das Ablaufskript zur Prozesssteuerung vom Werkstückagenten mitgebracht und zur Laufzeit an die Prozesssteuerung übergeben werden. Alternativ besteht bei einer zentralen Serverlösung eines Produktdaten-Servers (Bild 6.1) die Möglichkeit, dass sich die Prozesssteuerung direkt von diesem das Skript anhand der Technologie-URL und der Maschinen-Type abrufen.

Der Vorteil das Skript bei Bedarf der Station vorzugeben, liegt in der vereinfachten Handhabbarkeit und Entwicklungsarbeit bei der Modifikation bestehender Prozesse oder der Neueinführung eines neuen Produktes bzw. einer neuen Variante. Hierbei kann ein Skript an einer Bearbeitungsstation oder in einer Simulationsumgebung, repräsentativ für alle baugleichen Stationen, getestet werden. Die redundanten Stationen erhalten somit das Skript bei Bedarf zur Laufzeit, ohne den Betrieb durch manuelles Einspielen eines neuen Softwarestandes einstellen zu müssen. Somit werden an dieser Stelle manuelle Entwicklungsaufwände auf ein Minimum reduziert und erforderliche Stilllegungen vermieden.

Sollte aus technischen Gründen, z.B. auf Grund von Speicherplatzmangel, das Hinterlegen der Produktionsvorranggraph- oder Skriptdaten im Werkstückagenten nicht möglich sein, kann der Werkstückagent diese bei Bedarf von einem zentralen Dienst, dem Produktdaten-Server (Kapitel 6.1.1), beziehen. Um die Kommunikationslast über das Netzwerk zu reduzieren, kommt an jeder Bearbei-

tungsstation ein Cachespeicher zum Einsatz, der die Ergebnisse jeder Anfrage vor Ort für eine bestimmte Zeit zwischenspeichern kann. Somit können Anfragen nachfolgender Werkstückagenten für die gleichen Produktvarianten aus dem lokalen Speicher bedient werden. Liegt jedoch eine Änderung der Daten am zentralen Produktionsvorranggraph-Dienst vor, informiert dieser über einen Ereignis-Mechanismus die Bearbeitungsstationen, die infolgedessen die lokalen Cache-Speicher leeren. Somit werden diese bei Bedarf mit neuen Daten gefüllt.

Nach Abschluss des Prozesses, wird der Werkstückagent informiert. Dabei wird ihm immer ein koordinierter Zustandswert übermittelt, der Auskunft über Erfolg oder Misserfolg der Bearbeitung gibt. Alternativ besteht die Möglichkeit, Prozessparameter an den Werkstückagenten weiterzuleiten, die zum einen als Grundlage für die weiteren Entscheidungsprozesse, zum anderen zur Protokollierung in der Historie des Werkstücks dienen können. Für letzteren Verwendungszweck gibt es zwei Alternativen die Prozessparameter zu hinterlegen. Steht dem Werkstückagenten lokal genügend Speicherplatz zur Aufnahme der Produktionsdaten zur Verfügung, ist es von Vorteil, diese dort direkt zu hinterlegen und erst am Ende der Produktion vollständig auszulesen. Ist diese technische Möglichkeit nicht gegeben, kann der Werkstückagent ebenso an jeder Bearbeitungsstation die Daten direkt an den Betriebsdatenerfassungs-Server weiterleiten.

Die Daten der Prozessabläufe werden an jeder Bearbeitungsstation von einer Statistikkomponente erfasst und lokal gespeichert. Hierzu zählt u.a. die Dauer der Prozessausführung unter den vorgegebenen Bedingungen, wie Fertigungsparameter, Einschwingphase, etc. Diese Statistikwerte können zum einen für Wartungszwecke herangezogen werden, zum anderen sind sie wesentlich für die Verhandlungen mit den Werkstückagenten, bei denen eine Abschätzung der erforderlichen Prozesszeiten und -kosten erforderlich ist.

## 6.2 Verarbeitung von Produkt- und Anlageninformationen aus Primärquellen

Voraussetzung für die Abwicklung und Koordination des Materialflusses sind Produktherstellungsdaten einerseits und Anlagendaten andererseits. Hierbei unterliegt es dem Werkstückagenten, die Daten automatisiert zu verarbeiten und miteinander zu verknüpfen, um daraus die potentiellen Bearbeitungsstationen und Transportsysteme für die weiteren Schritte zu ermitteln. Um Entwicklungszeiten sowohl beim Anlagenumbau, als auch bei Änderungen in der Logistik einzusparen, wurde bereits in Kapitel 4.3 ein Verfahren vorgestellt und umgesetzt, das sowohl die Anlagenstruktur, als auch die Fähigkeiten der einzelnen Ressourcen automatisch ermittelt. Zudem besteht das Ziel, dass Änderungen am Produkt oder die Einführung einer neuen Variante keine manuellen Konfigurationen im Materialfluss nach sich ziehen. Motiviert hieraus wird im Folgenden ein allgemeingültiger Produktionsvorranggraph vorgestellt, der sowohl in der Montage, als auch in der Fertigung gleichermaßen eingesetzt werden kann (Kapitel 6.2.1). Voraussetzung für eine automatisierte Verarbeitung der Produktherstellungsdaten und folglich für eine automatisierte Ableitung des Materialflusses, ist eine standardisierte Datenstruktur, die zudem die Spezifikation der Produktvarianten gemäß dem Kundenauftrag bei der Auftragseinlastung vorsieht (Kapitel 6.2.4).



### 6.2.1 Spezifikation der Produktherstellungsschritte mittels Produktionsvorranggraph

Der Produktionsvorranggraph beschreibt mittels einer Baumstruktur die Herstellung eines Produktes. Die Elemente des Vorranggraphen sind hierbei Knoten, die einzelne Tätigkeiten repräsentieren, und Kanten, die Knoten zu einer hierarchischen Struktur verbinden. Jeder innere Knoten im Baum besitzt einen Vaterknoten und einen oder mehrere Kindknoten. Lediglich die Blattknoten am Ende des Baumes besitzen keine Kindknoten. Somit können für jeden Schritt im Produktherstellungsprozess die Vorgänger- und Nachfolgeschritte ermittelt werden.

Der Produktionsvorranggraph stellt einen Binärbaum dar, dessen Tätigkeitsknoten maximal zwei Kindknoten besitzen dürfen. Somit ist die Verbindung mehrerer Baugruppen in einem Schritt nicht erlaubt und müssen auf mehrere aufeinanderfolgende Einzelschritte zerlegt werden. Die Motivation in dieser Struktur liegt insbesondere in der späteren automatisierten Verarbeitung der Produktherstellungsdaten begründet, wodurch die Durchführung jedes einzelnen Montageschrittes in einem eigenen Knoten beschrieben werden kann. Folglich vereinfacht sich der Formalismus und Sonderfälle entfallen, die oftmals applikationsspezifisch vorgesehen werden müssen.

Der gesamte Produktionsvorranggraph beschreibt ein Produkt, dessen Endzustand der Wurzelknoten repräsentiert. Die inneren Knoten des Baumes stellen die Zwischenzustände des Produktes dar. Ein Knoten ist entweder eine Kombination oder eine Modifikation. Während die Kombination zwei Elemente durch die Montage mittels einer Prozesstechnologie vereint, beschreibt die Modifikation eine Zustandsänderung ebenfalls über eine Prozesstechnologie, z.B. mittels einer Bohrung oder eines formgebenden Prozesses. Ein Element stellt ein Bauteil oder eine Baugruppe, entstanden aus einer Kombination oder Modifikation, dar. Ein Bauteil ist ein Produkt, das nur an den Blattknoten des Baumes auftreten darf und an dieser Stelle als abgeschlossene Einheit betrachtet wird. Ein Blattknoten kann u.U. auf weitere Produktionsvorranggraphen verweisen, die das dortige Produkt spezifizieren. Produktvarianten unterscheiden sich durch verschiedene Teilbäume, die an einer oder mehreren ausgezeichneten Stellen im Baum ausgetauscht werden können. Bei der Festlegung einer Produktvariante werden für alle am Produkt beteiligten Produktionsvorranggraphen die Teilgraphen ausgewählt.

Zusammenfassend werden die Knoten im Baum in folgende Typen unterteilt (vgl. Bild 6.5):

- *Kombinations (Assembly)-Knoten* spezifizieren die Verbindung zweier Baugruppen (Kindknoten) zu einer neuen Baugruppe. Dieser Knoten enthält genau zwei Kindknoten.
- *Modifikations (Modification)-Knoten* beschreiben eine Zustandsänderung an einer Baugruppe (Kindknoten), die folglich als neue Baugruppe beschrieben wird. Dieser Knoten darf nur einen Kindknoten besitzen.
- *Bauteil (Part)-Knoten* sind die Blattknoten innerhalb des Baumes und verweisen auf ein Bauteil. Ein Bauteil stellt eine abgeschlossene Einheit eines Erzeugnisses dar, das selbst wiederum in einem anderen Produktionsvorranggraphen beschrieben werden kann, worauf lediglich verwiesen wird.
- *Auswahl (Choice)-Knoten* bieten die Möglichkeit Produktvarianten zu definieren. Sie ermöglichen mittels Konfiguration Teilbäume auszutauschen. Somit können mehrere Teilbäume für unterschiedliche Varianten spezifiziert und über einen gemeinsamen Auswahl-Knoten an einer

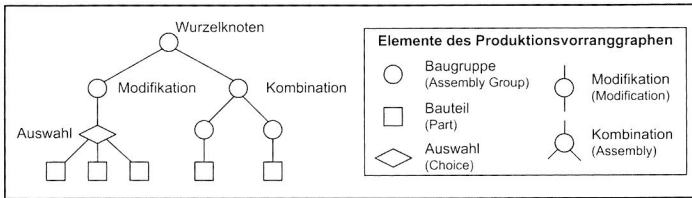


Bild 6.5: Komponenten des Produktionsvorranggraphen, basierend auf einem Binärbaum.

Stelle an den Baum angehängt werden. Ein Auswahl-Knoten ist lediglich ein Strukturelement und enthält somit keine Prozess- oder Produktinformationen.

Um die Varianten eindeutig definieren zu können, wird jeder Ast eines Auswahl-Knotens mit einer im gesamten Baum eindeutigen ID versehen. Somit muss zur Spezifikation der Variante lediglich eine Liste von Auswahl-IDs zu einem Produktionsvorranggraphen angegeben werden, die für alle Auswahl-Knoten im Baum eindeutig den resultierenden Baum für die Produktvariante festlegen.

Die Angabe des Bearbeitungszustands eines Produktes wird über die Knoten innerhalb des Vorranggraphen angegeben. Hierzu ist es erforderlich, die Knoten eindeutig zu identifizieren. Diesen wird bei Neuanlage eine innerhalb des Graphen eindeutige ID zugeordnet. Die Knoten-IDs werden nur einmalig einem Knoten zugeordnet. Wird ein Knoten gelöscht, so wird dessen ID nicht erneut an einen anderen neuen Knoten vergeben.

Die Produktionsvorranggraphen werden in XML-Dateien abgelegt, deren Struktur durch eine XSD-Datei vorgegeben wird. Hierdurch kann der syntaktische Aufbau der XML-Datei bei der Datenerfassung validiert werden. Zu jedem Produkt, d.h. Produktionsvorranggraphen, werden verschiedene Meta-Informationen hinterlegt, wie die Type und Bezeichnung des Produktes und die Artikel-Nr., die für die Referenzierung zum ERP-System [114] verwendet wird. Darüber hinaus besteht die Möglichkeit, Regeln für die Varianten-Verzweigungen zu hinterlegen. Somit darf es z.B. nicht erlaubt sein, Gehäuseteile unterschiedlicher Farbe miteinander zu kombinieren. Hierzu können Regeln als Formeln mit logischen Operationen im Vorranggraphen hinterlegt werden.

## 6.2.2 Spezifikation der Technologien

Die Technologie legt die Methode fest, wie ein Bearbeitungsschritt, deklariert über die Modifikation oder die Kombination im Produktionsvorranggraphen, ausgeführt wird (Bild 6.6). Die Technologie ist hierbei unabhängig vom eigentlichen Produkt und kann somit in unterschiedlichen Produkten oder innerhalb des Herstellungsprozesses eines Produktes mehrfach verwendet werden. Somit entfallen Mehraufwände in der Spezifikation der Produkte durch die Wiederverwendung gleicher Technologien.

Zudem abstrahiert die Technologie von der konkreten Implementierung nach außen. Eine Technologie kann z.B. bohren, fräsen oder die Montage zweier spezifischer Bauteile sein. Somit besteht für jeden spezifischen Bearbeitungsschritt, der lediglich Unterschiede in der Bearbeitung aufweist, eine

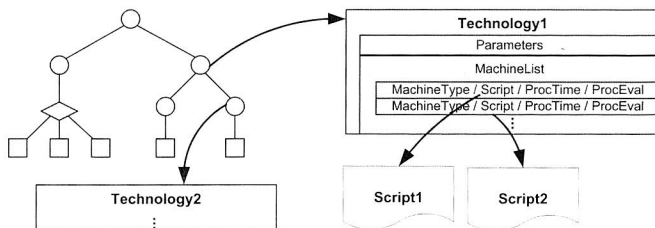


Bild 6.6: Spezifikation der Technologie für jeden Knoten im Graphen - für jeden Maschinentyp wird auf eigene Ablaufskripte verwiesen.

eigene Technologie-Spezifikation. Abhängig von der Technologie-Gruppe können die Technologie-Spezifikationen hierarchisch in Unterverzeichnissen organisiert werden, woraus sich eine hierarchisch strukturierte URL ergibt.

Die Technologie ist ebenso wie der Produktionsvorranggraph in einer eigenen XML-Datei beschrieben, auf die mittels einer eindeutigen URL verwiesen werden kann. Somit kann durch mehrfache Referenzierung diese Spezifikation wiederverwendet werden. Eine Technologie kann Fertigungsparameter enthalten, die für die Durchführung des Prozessschrittes herangezogen werden können. Darüber hinaus hält sie eine Liste von Maschinentypen, die die spezifizierte Technologie anbieten. Die Maschinentypen sind durch einen global eindeutigen String festgelegt. Jedem Eintrag in der Maschinentypen-Tabelle ist zudem eine Referenz auf ein Ablaufskript und Vorgabezeiten (*ProcTime*) bzw. Kostenwerte (*ProcEval*) für die Durchführung eines Prozesses zugeordnet. Das Ablaufskript ist spezifisch für den Aufbau einer Bearbeitungsstation ausgelegt und kann von diesem direkt zur Durchführung des Prozesses herangezogen und ausgeführt werden. Es berücksichtigt dabei alle relevanten Subkomponenten innerhalb der Bearbeitungsstation, die zur Ausführung des Prozesses koordiniert und gesteuert werden müssen. Darüber hinaus berücksichtigt das Skript alle zur Bearbeitung relevanten Parameter, die eine spezielle Konfiguration des Skriptes durch die Technologie, in der Spezifikation der Produktvariante (siehe Kapitel 6.2.3) oder auch zur Laufzeit im Werkstückagenten erlaubt. Hierbei werden aus diesen genannten Parameterquellen die Parametersätze aggregiert und gebündelt an die Prozesssteuerung in der Bearbeitungsstation (Kapitel 6.1.3) zur Ausführung des Skriptes übergeben.

Die Werte zur Dauer bzw. zu den Kosten/Aufwänden eines Prozessschrittes werden zur Abschätzung der Bearbeitung bei Verhandlungsroutinen, insbesondere in der Einführung eines neuen Produktes, herangezogen. Im Falle einer Gebotsanfrage kann somit die Bearbeitungsstation eine Aussage über die mögliche Dauer bzw. die Kosten/Aufwände eines Prozesses abschätzen, ohne diesen zuvor ausgeführt zu haben. Während der Verarbeitung kann die Prozesssteuerung die Zeiten und Kosten selbst ermitteln und somit die vorgegebenen Werte durch aktuelle Durchschnittswerte austauschen.

### 6.2.3 Spezifikation einer Produktvariante

Um der Produktvielfalt und den individuellen Kundenwünschen in der Abwicklung und Koordination der Produktion effizient gerecht werden zu können, ist eine integrierte Datenverarbeitung erforder-

lich, die ausgehend vom Auftragseingang die Daten zur Konfiguration der bestellten Produktvariante direkt in die Produktion einbringt. Hierbei muss wiederum ein allgemeingültiges Datenmodell entworfen werden, das auf die Spezifikation des Produktionsvorranggraphen verweist und zugleich die Möglichkeit bietet, den Bearbeitungszustand einer Produktvariante eindeutig angeben zu können. Da ein Produkt auch aus mehreren Baugruppen, beschrieben in unterschiedlichen Produktionsvorranggraphen aufgebaut sein kann, muss sich die Spezifikation der Produktvariante über alle Produktionsvorranggraphen erstrecken. Hierbei besteht die Aufgabe dieser zusätzlichen Datenstruktur darin, zum einen die Teilgraphen, die mittels der Auswahl-Knoten spezifiziert werden können, für die Produktvariante zu spezifizieren. Zum anderen muss die Verknüpfung der verschiedenen Vorranggraphen über die Blattknoten untereinander in beide Richtungen sichergestellt werden. In den Vorranggraphen verweist ein Blattknoten des übergeordneten Vorranggraphen auf einen untergeordneten Vorranggraphen, jedoch nicht umgekehrt. Die Bearbeitungsfolge zur Herstellung erfordert jedoch, dass der Baum von unten nach oben durchlaufen wird - auch über die Grenzen der Vorranggraphen hinweg. Daher muss auch die Zuordnungsrichtung vom Wurzelknoten des Vorranggraphen zum Blattknoten der übergeordneten Graphen für eine spezifizierte Produktvariante sichergestellt werden.

Die Datenstruktur zur Spezifikation einer Produktvariante bezieht sich auf einen oder mehrere miteinander verknüpfte Produktionsvorranggraphen. Diese können selbst wiederum mehrere Varianten unterschiedlicher Baugruppen repräsentieren (Bild 6.7). Somit existiert zunächst für jeden Produktionsvorranggraphen (*AssemblyGroup*) eine zugeordnete Datenstruktur *AssemblyGroupConfig*, die über die eindeutige *URL* auf diese verweist. Getrennt für jeden Vorranggraphen wird die Liste der Auswahl-IDs (*ChoiceIdList*) angegeben, die innerhalb des Vorranggraphen die produktvariantenspezifischen Teilbäume auswählen bzw. festlegen. Jeder *AssemblyGroupConfig* wird eine eindeutige *AssemblyGroupConfigID* vergeben, die zur Verkettung der *AssemblyGroupConfigs* untereinander verwendet wird. So verweist die einem untergeordneten Vorranggraphen zugeordnete *AssemblyGroupConfig* im Feld *Parent-AG* auf die übergeordnete, um die bottom-up-Verkettung der Vorranggraphen untereinander zu gewährleisten. Um jedoch zu wissen an welcher Stelle, d.h. an welchem Blattknoten, der untergeordnete Vorranggraph am übergeordneten hängt, muss neben der *Parent-AG* auch die ID des Blattknotens hinterlegt werden.

Die angegebenen Parameter in jeder Konfigurationsstruktur beziehen sich auf die Herstellung der zugeordneten Baugruppe und werden zusammen mit den Parametern in der Technologie-Spezifikation an die Prozesssteuerung der Bearbeitungsstation übergeben. Hierbei handelt es sich insbesondere um Prozessparameter, die nicht den Ablauf des Materialflusses oder die Koordination der Prozessschritte beeinflussen, sondern nur für die Konfiguration eines oder mehrerer Prozesse vorgesehen sind.

Um den Zustand eines Produktes während der Bearbeitung anhand des Bearbeitungsschrittes im Vorranggraphen spezifizieren zu können, ist eine Struktur mit verschiedenen Identifikatoren (*Product-StateID*) erforderlich, die es erlaubt, auf einen Knoten eindeutig verweisen zu können. Diese enthält zunächst die ID der Produktvariante, die innerhalb aller Produktvarianten auf eine spezifische Variante verweist. Innerhalb dieser wird auf die Konfiguration der Baugruppe (*AssemblyGroupConfig*) über die *AssemblyGroupConfigID* verwiesen, die zudem auf den zugehörigen Vorranggraphen referenziert. Zuletzt muss der Knoten innerhalb dieses Vorranggraphen angegeben werden, der dem Bearbeitungsschritt entspricht.

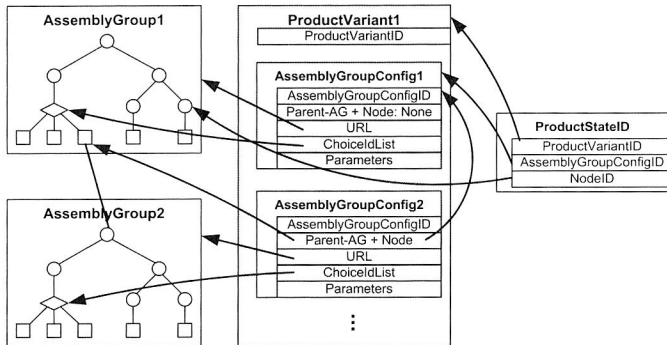


Bild 6.7: Spezifikation einer Produktvariante auch über mehrere Produktionsvorrangraphen hinweg möglich. Der Verweis auf den aktuellen Bearbeitungsstatus des Werkstücks erfolgt über die Datenstruktur *ProductStateID*.

## 6.2.4 Ableitung von Materialflussentscheidungen

Das Ziel der autonomen Materialflusssteuerung ist neben der direkten Interaktion der Bausteine miteinander auch die automatisierte Ableitung von Materialflussregeln. Hierzu sind die Bausteine in der Lage, direkt auf die Primärquellen zuzugreifen und die Daten hieraus direkt zu verarbeiten. Zur Koordination des Materialflusses muss der Werkstückagent die Produktdaten mit den Produktionsanlagendaten verknüpfen (vgl. Kapitel 6.1.1). In den Produktdaten, d.h. hier den Produktionsvorrangraphen, ist verzeichnet, welche Tätigkeiten in welcher Reihenfolge ausgeführt werden müssen. In den Daten der Produktionsanlage ist enthalten, welche Ressourcen im System existieren, welche Rolle sie übernehmen und welchem Maschinen-Typ sie zuzuordnen sind. Darüber hinaus sind die Fähigkeiten der Maschinen bekannt, die formal beschrieben werden. Aus den Fähigkeiten einerseits und den Anforderungen des Produktes an die Prozessschritte andererseits, kann eine Zuordnung der nächsten anstehenden Aufgabe für das zu bearbeitende Werkstück und den möglichen Bearbeitungsstationen erfolgen. Analog erfolgt diese Zuordnung auch zwischen dem Werkstück und den Transportsystemen, da diese i.d.R. nur Werkstücke bestimmter Größe aufnehmen können.

Die Auftragseinlastung, die einem Produktionssystem zugeordnet ist, erhält bei Auftragseingang eine *ProductStateID* (siehe Bild 6.7), die auf die zu fertigende Produktvariante bzw. Baugruppe verweist. Ist ein Produktionssystem für die Herstellung eines Zwischenerzeugnisses zuständig, erhält die zugehörige Auftragseinlastung lediglich den Auftrag, eine Baugruppe herzustellen. Diese Baugruppe wird durch den Verweis auf einen inneren Knoten innerhalb des Verbundes von ggf. mehreren Produktionsvorrangraphen spezifiziert. Dieser Knoten bestimmt im Gesamtgraphen einen Unterbaum, der alle Schritte zur Herstellung der zugehörigen Baugruppe festlegt. Hierbei muss die Auftragseinlastung die Durchführung aller im Unterbaum angegebenen Schritte gewährleisten, um das geforderte Bauteil herzustellen.

Der gesamte zusammengesetzte Produktionsvorrangraph beschreibt die Herstellung eines Produktes selbst anlagenübergreifend. Um die Arbeitsschritte automatisiert auf die verschiedenen Produktions-

anlagen zuordnen zu können, muss im Vorfeld ein Abgleich zwischen der Produktionseinrichtung einerseits und dem zusammengesetzten Produktionsvorranggraphen andererseits erfolgen. Zum einen können somit die Verfügbarkeit der erforderlichen Bearbeitungsstationen und Transportverbindungen ermittelt werden. Zum anderen werden gleichzeitig die Vorräte der notwendigen Bauteile an den einzelnen Bearbeitungsstationen, die zur Herstellung des Produktes erforderlich sind, geprüft. Diese Prüfungsvorgänge werden mittels virtuellen Werkstückagenten durchgeführt, die ohne physische Repräsentation durch die Anlage geschleust werden und die gleichen Wege einschlagen, wie das physische Werkstück. Der Vorteil dieser Methode liegt darin, die bestehende softwaretechnische Infrastruktur, die für Werkstückagenten mit physischer Repräsentation existiert, für die Vorausplanungen mittels virtueller Werkstückagenten wiederverwenden zu können. Dabei sind nur geringfügige Änderungen zur Erweiterung der bestehenden Infrastruktur erforderlich. Lediglich die virtuellen Agenten müssen mit einem anderen Verhalten bzw. Algorithmen versehen werden. Da jedoch die virtuellen Agenten an zentraler Stelle instanziiert und in das System eingeschleust werden, sind somit keine Änderungen an den verteilt gesteuerten Entitäten erforderlich.

Die virtuellen Agenten werden von den Auftragseinlastungsinstanzen der jeweiligen Produktionsanlagen initiiert. Der Anstoß der einzelnen zu fertigenden Baugruppen in den verschiedenen Produktionsanlagen erfolgt rekursiv ausgehend von der Produktionsanlage, die für die Herstellung des Endproduktes zuständig ist. Dabei werden die für das Endprodukt erforderlichen Bauteile an den benachbarten Produktionsanlagen angefordert. Wie in Kapitel 3 beschrieben, erfolgt die Kopplung zwischen den Produktionsanlagen immer über Bauteilläger, d.h. die Produkte werden von den Lägern am Ende der Produktionsanlage an die Bauteilläger an den Bearbeitungsstationen geliefert und dort zur weiteren Verarbeitung vorrätig gehalten. Mittels Verhandlung zwischen dem Bauteillager und den Lagern der benachbarten Produktionslinien wird geklärt, von welchem Lager die Bauteile bezogen werden können. Fehlen die Bauteile, so können sie in der lokalen Auftragsverwaltung eingelastet werden. Im Falle der Vorabprüfung mittels virtueller Werkstückagenten werden an dieser Stelle ebenso virtuelle Agenten zur Prüfung der Verfügbarkeiten und ggf. Zielvorgaben eingeschleust. Die verschiedenen virtuellen Werkstückagenten-Instanzen laufen bei der Montage in das Endprodukt wiederum zusammen, wobei in jedem Aggregationsschritt die ermittelten Daten zusammengefasst werden. Somit kann die Auftragseinlastung der Produktionsanlage zur Herstellung des Endproduktes über die Verfügbarkeit aller erforderlichen Ressourcen informiert werden, die als Grundlage zur Einlastung des Auftrages herangezogen wird.

Während die Auftragseinlastung die Vorabplanung übernimmt, bestimmt während der Laufzeit, d.h. nach Einlastung des Werkstücks, der Werkstückagent den Materialfluss durch das Produktionssystem. Er übernimmt die Zuordnung des Werkstücks zur Ressource, um alle erforderlichen Bearbeitungsschritte durchzuführen. Hierbei besitzt er zur Laufzeit Zugriff auf beide Informationsquellen, die Ressourcen und die Produktherstellungsdaten. Die Zuordnung, welche Ressource welchen Bearbeitungsschritt übernimmt, kann er entweder vor der Einlastung des Werkstücks treffen oder auch während der Laufzeit, d.h. nach Abschluss des vorhergehenden Schrittes. Dies ist abhängig von der Optimierungsstrategie der Auftragseinlastung. Dabei werden zwei mögliche Szenarien betrachtet:

1. Globale Optimierung: Um eine möglichst hohe Auslastung des Produktionssystems zu erzielen, eignet sich die globale Optimierung, die zentral an einer Koordinierungsinstanz, der Auftrags-einlastung, vorgenommen wird. Hierzu ist es erforderlich, dass die Auftragseinlastung ständig über alle Zustände im Produktionssystem und den Auslastungsgrad der Ressourcen informiert

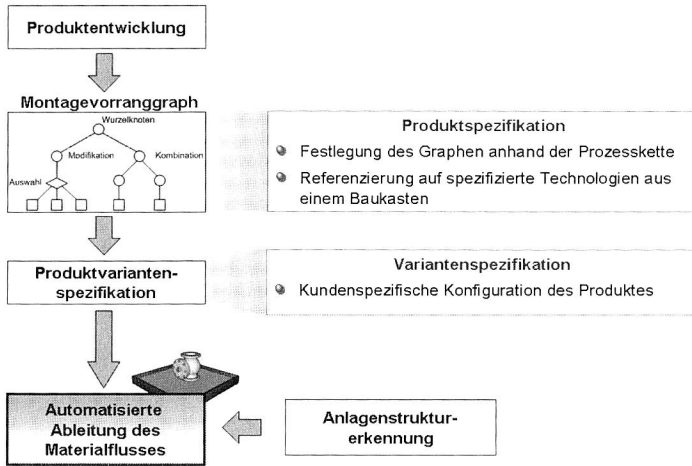


Bild 6.8: Verarbeitung grundlegender Produkt- und Anlagenstrukturdaten direkt aus den Primärquellen mit einem automatisierten Kombinationsverfahren zur Ableitung der Materialflussregeln.

wird. Ein Nachteil ist der Single-Point-of-Failure, den die Auftragseinlastung an dieser Stelle darstellt und somit bei Ausfall die gesamte Produktion im Produktionssystem zum Stillstand kommt. Darüber hinaus eignet sich diese Lösung nur bedingt für wandelbare Produktionsumgebungen, in denen sich ständig die Anforderungen ändern. Die globale Optimierung plant die anstehenden Aufträge in die Zukunft voraus, die jedoch für den Einsatz in wandelbaren Produktionsumgebungen nicht geeignet ist, da sich die Pläne bei häufigen Anpassungen am Produktionssystem ständig ändern.

2. **Lokale Optimierung:** Im Falle von wandelbaren Produktionsbedingungen eignet sich die lokale Optimierung, in der nach jedem Bearbeitungsschritt der Werkstückagent die nächsten Schritte koordiniert und nach lokalen Optimierungsregeln die geeignete Entscheidung trifft. Hierbei wird in die Entscheidung der aktuelle Auslastungsgrad der Ressourcen miteinbezogen und somit, basierend auf realen Zustandsgrößen, Entscheidungen für den Materialfluss getroffen.

Der Werkstückagent hat zu jedem Zeitpunkt und an jedem Ort Zugriff auf die Produktinformationen einerseits und die Anlageninformationen andererseits (vgl. Kapitel 6.1.1). Technisch können die Produktdaten, wie der erforderliche Teilbaum des Produktionsvorranggraphen, entweder auf einem beschreibbaren Speicher des Werkstücks hinterlegt werden oder über einen zentralen Server verfügbar sein. Die Strukturdaten und weitergehenden Informationen über die Ressourcen im Produktionssystem stehen an jeder Ressource über den lokalen Katalogdienst zur Verfügung.

Basierend auf den in Kapitel 6.2.3 vorgestellten Produktvarianten und der automatisierten Ableitung der erforderlichen Maschinentypen aus der Technologie-Spezifikation, kann der Werkstückagent in der Umgebung nach Bearbeitungsstationen mit der geforderten Maschinentype suchen und deren Verfügbarkeit prüfen (Bild 6.8). Ist keine geeignete Bearbeitungsstation verfügbar, wird eine Rückmel-

dung an die Betriebsdatenerfassung gesendet und automatisch das Ausschleusen bzw. Zwischenlagern des Werkstücks veranlasst. Im Zwischenlager kann der Werkstückagent kontinuierlich die Verfügbarkeit der erforderlichen Stationen prüfen und selbst wieder aktiv werden, wenn alle Voraussetzung zur erfolgreichen Bearbeitung erfüllt sind.

Trifft der Werkstückagent an einer Bearbeitungsstation ein, interagiert er zur Initiierung der Prozessschritte direkt mit der lokalen Prozesssteuerung, wie bereits in Kapitel 6.1.3 beschrieben. Hierzu gibt er die Durchführung des Prozesses in Auftrag, indem er die Ausführung des maschinenbezogenen Ablaufskriptes anstößt und diesem vorab die Fertigungsparameter übermittelt. Die Parameter werden aus der Technologie-Spezifikation, aus der Produkt-Varianten-Spezifikation und aus den individuellen Parametern des Werkstücks aggregiert. Letztere haben die höchste Priorität, wodurch ein Werkstück Parameterwerte aus der Produkt-Varianten-Spezifikation oder aus der Technologie-Spezifikation überschreiben kann. Ebenso können Parameterwerte aus der Technologie durch jene aus der Produkt-Varianten-Spezifikation ersetzt werden.

## 6.3 Maschinen- und Betriebsdatenerfassung

Die Aufgabe der Maschinen- und Betriebsdatenerfassung im wandelbaren Produktionsumfeld hat die Aufgabe, beliebige und selbst zur Entwicklungszeit unbekannte Größen aufnehmen und erfassen zu können [115]. Hierzu ist die Auslegung einer herstellungs- und produktunabhängigen Datenbankstruktur erforderlich, die mit standardisierten Schnittstellen zur Ablage einerseits, als auch zur Abfrage von Produktionsdaten andererseits den Einsatz in beliebigen Produktionsumgebungen ermöglicht. Die Ablage der Daten unterteilt sich in drei Bereiche:

- **Produktfertigungsspezifikation:** Um die Nachvollziehbarkeit der Produktherstellung gewährleisten zu können, müssen die produktspezifischen Herstellungsdaten hinterlegt werden. Hierzu stehen Tabellen zur Verfügung, die den Produktionsvorranggraphen aufnehmen können und die Referenzierung jedes Bauteilzustands erlauben.
- **Anlagendaten:** Zum Zeitpunkt der Produktherstellung kann der Anlagenaufbau von Interesse sein, um ggf. Zusammenhänge bei Fehlproduktionen besser nachvollziehen zu können. Zudem müssen maschinenabhängige Daten, wie Maschinenfehler und Abnutzungserscheinungen, erfasst werden.
- **Herstellungsdaten:** Geeignete Datenstrukturen ermöglichen die Aufnahme beliebiger Produktionsherstellungsdaten. Dabei müssen die erfassten Daten den Produktzuständen in der Produktfertigungsspezifikation zugeordnet werden.

Zur Erfassung dieser Datenbestände, werden im Folgenden die zugehörigen Entity-Relationship-Diagramme [116] vorgestellt, die eine umfassende Datenbankstruktur beschreiben.



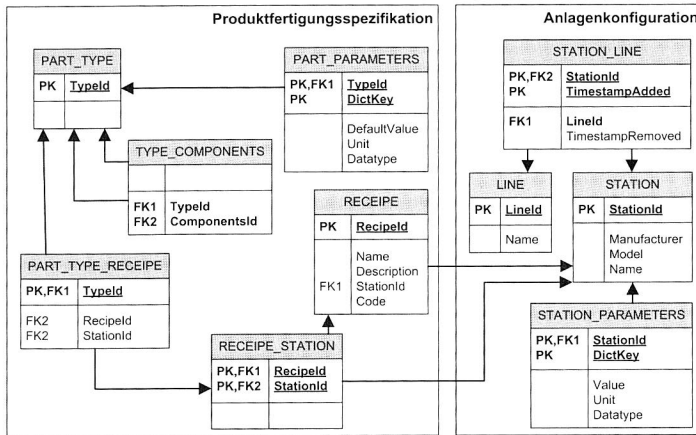


Bild 6.9: Entity-Relationship-Diagramm mit applikationsunabhängiger Datenstruktur zur Erfassung aller Versionen der Produktspezifikation und der Struktur des Produktionssystems.

## Produktfertigungsspezifikation

Die Tabelle *PART\_TYPE* ist das zentrale Bindeglied zur Beschreibung des Baumes aus dem Produktionsvorranggraphen (Bild 6.9). Hierüber wird jeder Knoten des Baumes mit einer global eindeutigen *Typeld* versehen. Die hierarchische Struktur wird über die Tabelle *TYPE\_COMPONENTS* bewerkstelligt, die zwei Fremdschlüssel auf die *PART\_TYPE*-Tabelle enthält: Zum einen auf den Vaterknoten, zum anderen auf den Kindknoten. Für  $n$  Kindknoten im Baum existieren somit  $n$  Tabelleneinträge für den Vaterknoten. Unter *PART\_PARAMETERS* sind für einen Knoten im Produktionsvorranggraphen alle Fertigungsparameter als Schlüssel-Werte-Paare gespeichert. Die Tabelle *RECEIPE* hält alle Ablaufskripte zur Steuerung der Prozesse, die über *RECEIPE\_STATION* einer oder mehreren Stationen zugeordnet werden können. Die Verbindung zum Knoten des Produktionsvorranggraphen erfolgt über *PART\_TYPE\_RECEIPE*.

Somit kann der vollständige Produktionsvorranggraph inkl. der zugehörigen Herstellungsparameter und Prozessablaufskripte hinterlegt werden. Diese Daten sind Grundlage für die vollständige Nachvollziehbarkeit des Herstellungsprozesses und ermöglichen z.B. qualitative Analysen in der Produktion zur Steigerung der Qualität.

## Anlagendaten

Die Anlagendaten umfassen die Maschinenspezifikationen und die laufenden Maschinendaten, die Auskunft über den Nutzungsgrad und den Zustand der Maschine geben. Zudem werden zur Nachvollziehbarkeit der Produktherstellung die gesamte Anlagenstruktur zum Zeitpunkt der Produktherstellung erfasst. Die Zuordnung der Bearbeitungsstationen zu den Produktionsanlagen erfolgt in der Tabelle *STATION\_LINE*. Hierbei dient neben der Stations-Id der Zeitstempel, zu dem die Station einer Anlage hinzugefügt wurde, als Schlüsselwert. Grundlegend werden die Spezifikationen jeder

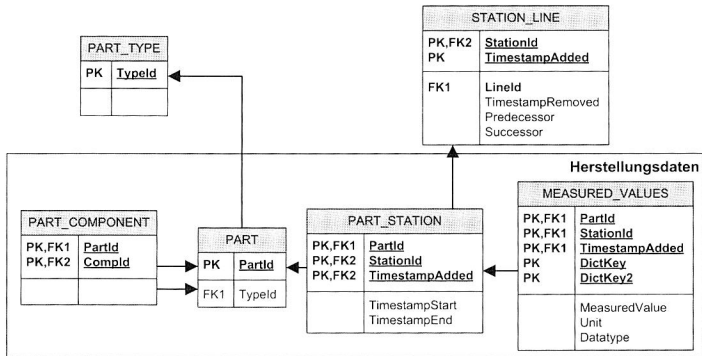


Bild 6.10: Entity-Relationship-Diagramm zur Ablage der Produktherstellungsdaten in Datenbank-Tabellen. Die Einträge zur Nachverfolgung der Herstellungsprozesse sind mit den Anlagenstrukturdaten zum Herstellungszeitpunkt verknüpft.

einzelnen Bearbeitungsstation in der Tabelle *STATION* hinterlegt (Bild 6.9). Somit ist der Aufbau einer Produktionsanlage zu jedem Zeitpunkt erfassbar. Änderungen an der Anlage, z.B. dem Hinzufügen oder Entfernen von Bearbeitungsstationen, werden unmittelbar in der Tabelle *STATION\_LINE* erfasst. Weitere Parameter und Spezifikationen können in der *STATION\_PARAMETERS*-Tabelle als Schlüssel-Werte-Paare erfasst werden, die genaueren Aufschluss über den Aufbau der Station geben. In dieser können ebenso Dokumente oder Konstruktionszeichnungen hinterlegt werden, um somit in einem Agglomerat alle die Bearbeitungsstation betreffenden Informationen vorrätig zu halten. Werden Veränderungen an einer Bearbeitungsstation vorgenommen, so erfolgt ein neuer Eintrag in der Tabelle *STATION* mit einer neuen Stations-Id, um Modifikationen an einer Bearbeitungsstation nachvollziehen zu können.

## Herstellungsdaten

Die Herstellungsdaten umfassen die spezifischen Prozessdaten, die bei der Durchführung des Prozesses erfasst werden. Zentrales verbindendes Element ist die Tabelle *PART*, die für jeden Bauteilzustand eines Bauteils einen Eintrag vorsieht (Bild 6.10). Dabei erhält jedes Bauteil im neuen Zustand, d.h. nach jeder Bearbeitung an einer Station, eine eigene eindeutige *Partld*. Über die Tabelle *PART\_COMPONENT* kann nachvollzogen werden, welche Bauteile miteinander in der Montage verbunden wurden, bzw. welche *Partld* nach einer Modifikation in die nächste *Partld* überführt wurde. Über *PART\_STATION* erfolgt die Zuordnung des Bearbeitungsschrittes zur Station. Hierbei wird auf die *STATION\_LINE*-Tabelle verwiesen, um zu erkennen, in welchem Produktionssystem die Station zum Zeitpunkt der Bearbeitung eingebunden war. In *MEASURED\_VALUES* werden alle Prozessparameter und Qualitätsprüfergebnisse für jeden Bearbeitungsschritt bzw. Bauteilzustand erfasst. Zur Erhaltung der Flexibilität werden ebenfalls an dieser Stelle Schlüssel-Werte-Paare abgelegt.

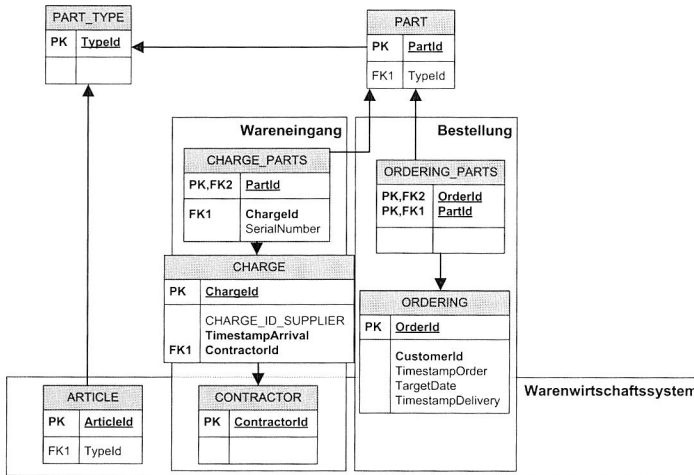


Bild 6.11: Entity-Relationship-Diagramm mit Anbindung an das Bestellwesen und den Wareneingang.

### Anbindung an das ERP-System

Die Betriebsdatenerfassung ermöglicht ebenso die Anbindung an herkömmliche Enterprise Resource Management (ERP)-Systeme. Zunächst erfolgt hierbei die Zuordnung der End-Bauteiltypen zu den Artikeln im Artikel-Stamm des Warenwirtschaftssystems (Bild 6.11). Dies ist erforderlich, um in späteren Abfragen z.B. alle Historiedaten oder die Parametervorgaben für einzelne Bearbeitungsschritte eines bestimmten Artikels abrufen zu können. Dies wird u.a. dadurch ermöglicht, dass alle Bauteile (*PART*-Einträge) auf die Bauteiltypen (*PART\_TYPE*-Tabelle) verweisen. Zudem müssen die Bauteile aus der *PART*-Tabelle, die aus dem Eingangswarenlager entnommen werden, ebenfalls aus der Warenbestandsliste abgebucht werden. Um darüber hinaus die Nachvollziehbarkeit selbst der eingegangenen Bauteile, die eindeutig identifizierbar sind, zu gewährleisten, muss die Zuordnung der initialen Bauteile und der Chargen aus dem Wareneingang gewährleistet werden (*CHARGE\_PARTS*). Hierbei besteht eine Charge (*CHARGE*) entweder aus einem einzigen Bauteil, das eindeutig identifizierbar ist oder aus einer Gruppe von Bauteilen, die keine Identifikation der Einzelteile zulässt. Im letzten Fall erhält das Einzelteil einer Charge erst seine Identifikation, wenn dieses dem Produktionssystem als Bauteil hinzugefügt wird. Dennoch ermöglicht diese Zuordnung die Bestimmung des Lieferanten (*CONTRACTOR*), der die Charge geliefert hat. Neben der Zuordnung zum Wareneingang kann eine Referenz vom produzierten Produkt zum Kundenauftrag im Bestellwesen gesetzt werden. Hierbei erfolgt in der Tabelle *ORDERING\_PARTS* der Verweis von der Auftrags-tabelle (*ORDERING*) zur Baugruppentabelle *PART* des fertig hergestellten Bauteils. Somit kann zu einem eindeutig identifizierbaren Produkt eines Kundenauftrags die Produktionshistorie abgerufen werden.

Mit der vorliegenden Datenbankstruktur können die wesentlichen Produktionsdaten für die umfassende Nachvollziehbarkeit der Herstellungshistorie einzelner Produkte erfasst werden. Wesentliches Merkmal der vorgestellten Datenstruktur ist die geeignete Verknüpfung der Tabellen untereinander,

um alle erforderlichen Anfragetypen, ausgehend vom Produkttyp, dem Kundenauftrag, dem Wareneingang und dem Anlagenaufbau, bewerkstelligen zu können. Dabei bietet die Struktur die Möglichkeit, nur den erforderlichen Datenbestand nach Bedarf zu erfassen - nicht benötigte Teile der Tabellen oder Tabellengruppen können leer belassen werden.

## 6.4 Übergabebahnhof als Schnittstelle zwischen Stationen und Transportsystemen

Der Übergabebahnhof stellt die Schnittstelle zwischen dem Transportsystem und der Umgebung dar (Kapitel 3.1). Hierbei müssen die Bahnhöfe unabhängig von der Art des Transportsystems nach außen den externen Komponenten, wie Bearbeitungsstation oder Werkstück, einheitliche Softwareschnittstellen zur Verfügung stellen, um die Integration dieser Komponenten untereinander zu vereinfachen bzw. selbst ad hoc Verbindungen zur Laufzeit ohne manuellen Konfigurations- oder Anpassungsaufwand zu ermöglichen. Basierend auf dieser Anforderung wird im Folgenden sowohl der interne Aufbau der Bahnhofssteuerung (Kapitel 6.4.1), als auch die Schnittstellen und das Verhalten des Bahnhofs mit den externen Bausteinen, d.h. den Bearbeitungsstationen (Kapitel 6.4.2 und 6.4.3) und dem Transportsystem (Kapitel 6.4.2) eingehend erläutert.

### 6.4.1 Konzeption eines allgemeinen Bahnhofmodells

Der Ausgangsbahnhof hat die Aufgabe, Werkstücke von der Bearbeitungsstation entgegenzunehmen und an das Transportsystem weiterzuleiten. Der Eingangsbahnhof nimmt eingehende Werkstücke vom Transportsystem auf, kann diese ebenso zwischenlagern und auf Anfrage der Bearbeitungsstation an diese weiterleiten. Die Pufferfunktion beim Ausgangsbahnhof kann bei Bedarf, abhängig von der Art des Transportsystems, aktiviert werden. Der Bahnhof selbst ist modular aufgebaut und besteht aus verschiedenen standardisierten Softwarebausteinen, die primär unterschiedliche Aufgaben übernehmen. Die Softwaremodule werden über die verteilte, hierarchische Steuerungsmiddleware als Facetten implementiert, die sich als Funktionskomponenten in einen Steuerungsbaum einbinden lassen (Kapitel 4). Hierbei wird der Bahnhof selbst durch einen Teilbaum repräsentiert.

Die organisatorischen Aufgaben werden von der Job-Facette (Bild 6.12) übernommen. Diese stellt das Bindeglied für alle organisatorischen bzw. planerischen Aufgaben zum Transportsystem dar. Hierzu zählt insbesondere die Einlastung eines Transportauftrags beim Transportsystem, sofern dieses erforderlich ist. Stetigfördersysteme erfordern in den meisten Ausprägungen keine Anmeldung eines Auftrages, wohingegen Unstetigfördersysteme nur bei Bedarf, d.h. bei Auftragserteilung, den Bahnhof anfahren. Die Job-Facette interagiert hauptsächlich mit dem Werkstückagenten, der die weitere Organisation der nächsten Schritte prüft und die Einlastung von Transportaufträgen an der Job-Facette veranlasst. Die Dock-Facette übernimmt Koordinationsaufgaben am Bahnhof. Hierzu zählt insbesondere im Falle von Fahrzeugen die Erteilung von Berechtigungen, welches Fahrzeug andocken darf, um somit zum einen Kollisionen zu vermeiden und zum anderen die Reihenfolge der Werkstückaufnahme bzw. -abgabe steuern zu können. Die Handover-Facette implementiert die Interaktion zwischen dem Bahnhof und dem Transportsystem zur Übergabe des Werkstücks. Die Feeder-Facette repräsentiert das Fördersegment, das die physische Übergabe des Werkstücks durchführt. Die Feeder-Facette interagiert mit der Lager-Facette (*Stock*), die das Lager bzw. die Pufferplätze zur Zwischenlagerung der Werkstücke repräsentiert. Das Lager ist somit dem Fördersegment des Bahnhofs untergeordnet und

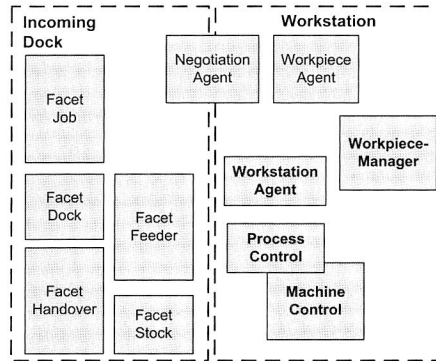


Bild 6.12: Komponenten des Bahnhofs und der Bearbeitungsstation, die von außen erreichbar sind.

führt nur Ein- bzw. Auslagerungen auf Anweisung durch. Die Entscheidung, ob ein Bauteil zwischenlagert wird, trifft die Feederfacette, abhängig davon, ob bereits eine Anfrage zur Weiterleitung des Werkstücks eingetroffen ist bevor das physische Werkstück ankam.

Auf Seiten der Bearbeitungsstation übernimmt die Maschinensteuerung (*MachineControl*) die übergreifende Koordination aller Vorgänge in der Station. Sie kennt den genauen Aufbau der Bearbeitungsstation und welche Übergabebahnhöfe lokal angebunden sind. Zudem koordiniert sie die Vorgänge durch Interaktion mit den angegliederten Bahnhöfen zur Aufnahme neuer Werkstücke aus dem Puffer/Lager der Eingangsbahnhöfe und steuert die Positionierung und Fixierung des Werkstücks in der Station als Voraussetzung zu dessen Bearbeitung. Analog zum physischen Werkstück befindet sich immer der zugehörige Werkstückagent in der Bearbeitungsstation. Diese werden vom Werkstück-Manager verwaltet (vgl. Kapitel 6.1.3). Der Verhandlungsagent (*Negotiation-Agent*) übernimmt Verhandlungsaufgaben im Auftrag des Werkstückagenten. Dieser wird vom Werkstückagenten vor jeder Verhandlungsaufgabe mit den Zielparametern instanziiert und über die Bahnhöfe auf die Reise geschickt, um an jeder erforderlichen Ressource für die nächsten Bearbeitungsschritte Informationen zu sammeln und ggf. weitere Entscheidungen mittels Verhandlungsverfahren zu treffen.

#### 6.4.2 Anbindung des Eingangs-Bahnhofs an die Bearbeitungsstation

Der Eingangsbahnhof dient zum einen zur Annahme physischer Werkstücke, zum anderen nimmt er Anfragen vom Transportsystem entgegen und leitet diese an die Bearbeitungsstation weiter. Letztere treffen in Form der Verhandlungsagenten (*Negotiation*) bei der Job-Facette ein, die serialisiert über das Kommunikationsnetzwerk zwischen den Bahnhöfen migrieren. Nach der Initialisierung ist der Verhandlungsagent in der Lage, mit der lokalen Bearbeitungsstation zu verhandeln bzw. Planvorgaben zu prüfen. Über die Job-Facette des Ausgangsbahnhof kann der Verhandlungsagent wiederum zum nächsten Eingangsbahnhof migrieren.

Der Eingang des physischen Werkstücks am Eingangsbahnhof wird über die Handover-Facette koordiniert und die Übergabe über das Fördersegment, repräsentiert durch die Feeder-Facette, durchgeführt.

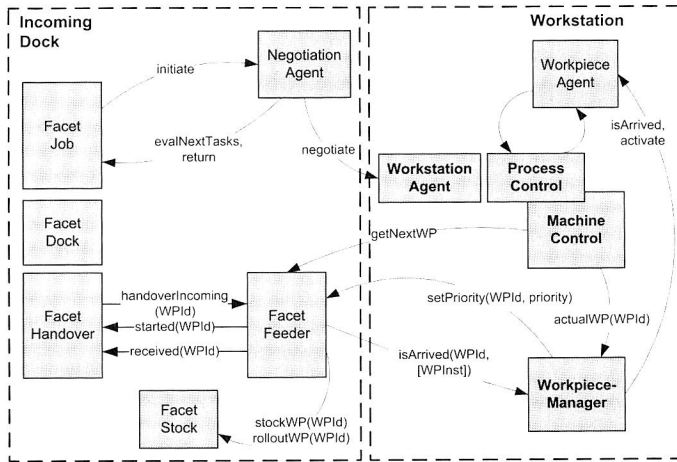


Bild 6.13: Eingangsbahnhof zur Aufnahme eingehender Werkstücke zur Weiterleitung an die Bearbeitungsstation. Zudem treffen eingehende Verhandlungsagenten zur Prüfung der Ziele bzw. Bestimmung der nächsten Schritte im Materialfluss ein.

Hierbei signalisiert die Handover-Facette mit *handoverIncoming* (Bild 6.13) die Aufforderung zur Aufnahme des Werkstücks an die Feeder-Facette. Diese antwortet mit *started*, wenn die Aufnahme bereit und gestartet ist und mit *received*, wenn das Werkstück aufgenommen wurde. Sofern das Werkstück angekommen ist, wird das Einlagern mit *stockWP* veranlasst und die Ankunft dem Werkstück-Manager mittels *isArrived* mitgeteilt. Dabei wird ggf. die Instanz des Werkstückagenten übergeben, die vom mitgeführten Speicher ausgelesen wurde.

Der Werkstückagenten-Instanz wird die Information zur Ankunft des Werkstücks weitergeleitet. Im Gegenzug legt der Werkstück-Manager abhängig von Prioritäts- bzw. Fälligkeitswerten die Priorisierung des Werkstücks zur Bearbeitung innerhalb der Station beim Feeder fest. Dieser hält somit die Bearbeitungsreihenfolge der Werkstücke in einer Liste und leitet die Werkstücke gemäß dieser Sortierung an die Station weiter, sofern ein wahlfreier Zugriff auf den Puffer bzw. das Lager besteht.

Die Maschinen-Steuerung signalisiert bei Bedarf der Feeder-Facette lediglich den Bedarf eines neuen Werkstücks und die Feeder-Facette veranlasst das Auslagern aus dem Puffer mit *rolloutWP*. Nach erfolgreichem Einspannen des Werkstücks in der Bearbeitungsstation signalisiert die Maschinen-Steuerung dem Werkstück-Manager, welches Werkstück eingespannt ist, worauf rekursiv mittels *activate* der Werkstückagent zum Anstoß der Verarbeitung aufgefordert wird. Die Initiierung der Bearbeitung und Rückmeldungen über den Fortschritt bzw. Erfolg der Bearbeitung, tauscht der Werkstückagent mit der Prozess-Steuerung (*Process-Control*) direkt aus.

### 6.4.3 Anbindung des Ausgangs-Bahnhofs an die Bearbeitungsstation

Nach erfolgreicher Bearbeitung des Werkstücks und Rückmeldung von der Prozess-Steuerung an den Werkstückagenten, koordiniert dieser die nachfolgenden Bearbeitungsschritte. Hierzu müssen im ersten Schritt die nächste bzw. nächsten Bearbeitungsstationen selektiert oder die vorgegebenen Planvorgaben mit dem Verhandlungsagenten überprüft werden. Der Werkstückagent erhält über *getOutDockLocators* die Locators der Ausgangs-Bahnhöfe, die an der Station angebunden sind (Bild 6.14). Für jeden Ausgangsbahnhof wendet sich der parametrisierte Verhandlungsagent an die Job-Facette (*evalNextTasks*) und sendet sich als Referenz dabei mit. Hierbei prüft der Verhandlungsagent die nächsten Schritte (vgl. Kapitel 6.5). Nach erfolgreicher Ermittlung der nächsten erforderlichen Ressourcen und Evaluierung des günstigsten Pfads im Materialfluss, veranlasst der Werkstückagent über *addOutgoing* das Hinzufügen des Werkstücks zum ausgewählten Bahnhof und übergibt dabei den Locator der nächsten Bearbeitungsstation (*DestLoc*). Die Job-Facette kann zum einen die Einlastung des Transportauftrags im dahinterliegenden Transportsystem planen und koordinieren, zum anderen fordert sie die Maschinen-Steuerung auf, das Werkstück zu sich zu befördern. Dieser Vorgang wird erst dann gestartet, wenn der Werkstückagent mit *isFinished* dem Werkstück-Manager signalisiert hat, dass die Bearbeitung erledigt ist und infolgedessen der Werkstück-Manager der Maschinen-Steuerung die Freigabe des Werkstücks mittels *releaseWP* erlaubt. Dabei wird zudem die Instanz des Werkstückagenten übergeben, um diesen, sofern vorhanden, auf den beschreibbaren Speicher am Werkstück zu hinterlegen. Alternativ übernimmt an dieser Stelle der Werkstück-Manager auch die Migration des Werkstückagenten über das Netzwerk, sofern kein beschreibbarer, mobiler Speicher am Werkstück existiert.

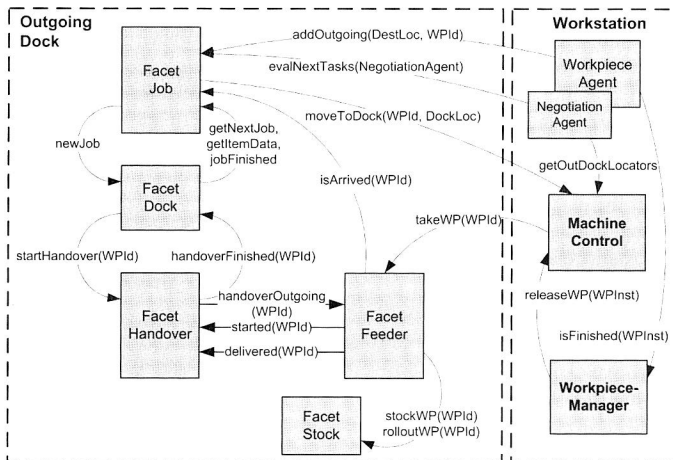


Bild 6.14: Ausgangsbahnhof mit Anbindung an die Bearbeitungsstation - standardisierte Facetten bieten Standardfunktionalität für den Werkstückagenten und die Bearbeitungsstation zur transparenten Interaktion mit dem Transportsystem.

Die Maschinen-Steuerung signalisiert der Feeder-Facette des Bahnhofs die Übergabe des Werkstücks, welche wiederum diese Aufforderung zur Zwischenpufferung des Werkstücks an das optional angebundene Lager, sofern von der Art des Transportsystems erforderlich, weiterleitet. Die Dock- und die Handover-Facetten übernehmen dabei die Koordination der Werkstückübergabe an das Transportsystem. Die Job-Facette informiert die Dock-Facette im Falle neuer Transportaufträge, um diese bei Stetigfördersystemen sofort zu bearbeiten. Ist das Transportsystem ein unstetiges Fördersystem, so wird diese Mitteilung ignoriert, da die Fahrzeuge die Job-IDs kennen und diese beim Andocken an die Dock-Facette übermitteln, um den geplanten Transportauftrag anzukündigen. Ist die Voraussetzung für den physischen Werkstückübergang, z.B. das Andocken des Fahrzeugs, erfüllt, veranlasst die Dock-Facette mittels *startHandover* die Werkstückübergabe. Während auf der einen Seite die Handover-Facette mit der Handover-Facette des Transportsystems interagiert und die Übergabe koordiniert, veranlasst sie auf interner Seite des Bahnhofs an der Feeder-Facette die Abgabe mittels *handoverOutgoing* und erhält Rückmeldung über den Zustand der Übergabe (*started* und *delivered*). Umgekehrt übermittelt die Handover-Facette der Dock-Facette die Übergabe mittels *handoverFinished*, woraufhin die Dock-Facette die Fertigstellung der Job-Facette weiterleitet (*jobFinished*) und gleichzeitig prüft, ob noch ein Transportauftrag eines weiteren Werkstücks z.B. für das angedockte Fahrzeug ansteht (siehe *getNextJob* und *getItemData*). In diesem Fall würde die Handover-Facette solange den Auftrag zur Werkstückübergabe mittels *startHandover* erhalten, bis kein weiterer relevanter Auftrag ansteht.

#### 6.4.4 Interaktionen zwischen Bahnhof und Transportsystem

In den bisherigen Kapiteln wurde ausschließlich die Anbindung der Bahnhöfe zu den Bearbeitungsstationen betrachtet und erläutert. Im Folgenden wird auf die andere Seite des Bahnhofs eingegangen, d.h. die Anbindung dessen an das Transportsystem. Beim Transportsystem werden zunächst zwei Grundtypen unterschieden (Kapitel 3.4), die maßgeblichen Einfluss auf die Interaktion zwischen Bahnhof und dem Transportsystem haben. Die eine Gruppe der aktiven Fördersysteme, d.h. Unstetig-Fördersysteme, wie Fahrerlose Transportfahrzeuge, Gabelstapler oder Schienenfahrzeuge, erfordern spezifische Mechanismen, die das Andocken, Abdocken und die korrekte Werkstückübergabe koordinieren. In der zweiten Gruppe der passiven Fördersysteme, d.h. der Stetig-Fördersysteme, wie gurtgetriebene Fördersysteme, in denen die Transporteinheiten durch eine externe Aktorik angetrieben werden, kann die Übergabe der Werkstücke sofort durchgeführt werden, da das Transportsystem immer angebunden und somit physisch jederzeit verfügbar ist. Zudem besitzen die Transportstrecken im passiven Fördersystem eine Pufferfunktion, deren Auslastung aber vor der Übergabe geprüft werden muss.

Im Falle der aktiven Fördersysteme übernimmt die Dock-Facette des Bahnhofs (Bild 6.12) die Koordination der Fahrzeuge. Dabei muss geprüft werden, welches von den wartenden Fahrzeugen zuerst an den Bahnhof andocken darf. Ebenso muss im Falle eines Fahrzeugausfalls und nach Ablauf eines Timeouts eine Umsortierung der Aufträge durchgeführt werden, damit wartende Fahrzeuge für nachfolgende Transportaufgaben nicht zu lange blockiert werden. Dies ist jedoch nur dann erforderlich, wenn der Puffer des Ausgangsbahnhofs keinen selektiven Zugriff auf die auszuliefernden Werkstücke ermöglicht. Damit es jedoch vor den Bahnhöfen zu keinen Kollisionen oder Behinderungen kommt, müssen die Bahnhöfe mit einem Zugangsbereich ausgestattet sein. Dieser Bereich befindet sich direkt vor jedem Bahnhof und darf nur von einem Fahrzeug gleichzeitig befahren werden.



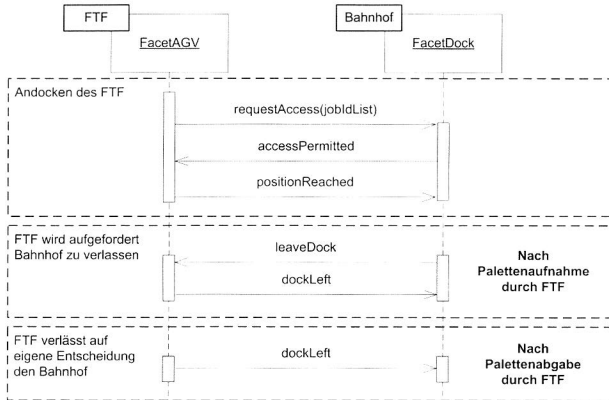


Bild 6.15: Interaktion zwischen dem autonom agierenden Fahrerlosen Transportfahrzeug und dem Bahnhof zur Koordinierung der Prioritäten, in welcher Reihenfolge die Fahrzeuge andocken dürfen.

Alle Fahrzeuge, die am Bahnhof andocken möchten, müssen von diesem die Erlaubnis einholen, um den Zugangsbereich befahren zu dürfen. Bis zur Zusageerteilung müssen diese vor dem Bereich warten. Hierzu fragen die Fahrzeuge aktiv beim Bahnhof bzw. bei der Dock-Facette mit *requestAccess* nach (vgl. Bild 6.15) und melden sich an. Sobald der Zugang gewährt wird, erhalten sie über *accessPermitted* die Erlaubnis. Das erfolgreiche Andocken am Bahnhof signalisiert das Fahrzeug mittels *positionReached*, um dabei die weiteren Schritte, wie die Übergabe der Werkstückträger anzustoßen. Nach der Werkstückübergabe kann das Fahrzeug entweder den Bahnhof auf eigene Entscheidung verlassen oder der Bahnhof fordert das Fahrzeug auf, sich zu entfernen. Ersterer Fall tritt insbesondere bei den Eingangsbahnhöfen auf, da dort das Fahrzeug die abzugebenden Werkstücke kennt. Der zweite Fall kommt ausschließlich bei den Ausgangsbahnhöfen zum Tragen, da dort der Bahnhof entscheidet, welche Werkstücke er an das Fahrzeug übergibt.

Die Werkstückübergabe sowohl bei den aktiven als auch passiven Fördersystemen wird mittels der Handover-Facette koordiniert. Die Implementierung dieser ist auf beiden Seiten, d.h. der Bahnhofs- und Transportsystemseite, identisch und somit lassen sich die Interaktionen beim Eingangs- und Ausgangsbahnhof spiegelverkehrt darstellen (vgl. Bild 6.16). Den initialen Anstoß gibt immer diejenige Handover-Facette mit *readyForHandover*, die ein Werkstück abzugeben hat. Dabei übermittelt sie die Job-Id des Transportauftrags, der für das zu übergebende Werkstück eingelastet wurde. Die Gegenseite antwortet mit *startHandover*, wenn diese für die Aufnahme bereit ist. Nach erfolgreicher Aufnahme sendet der Empfänger ein *itemReceived*, um die ordnungsgemäße Übergabe zu signalisieren. Die abgebende Instanz sendet ein *handoverFinished*, um das Ende der Werkstückübergabe bekannt zu geben. Zur Übergabe des nächsten Werkstücks wird wiederum ein *readyForHandover* gesandt. Dieser Vorgang wiederholt sich solange, bis alle erforderlichen Werkstücke übergeben wurden.

Mit diesen Bahnhof-Steuerungsmodulen steht sowohl auf Seiten der Bearbeitungsstation (Kapitel

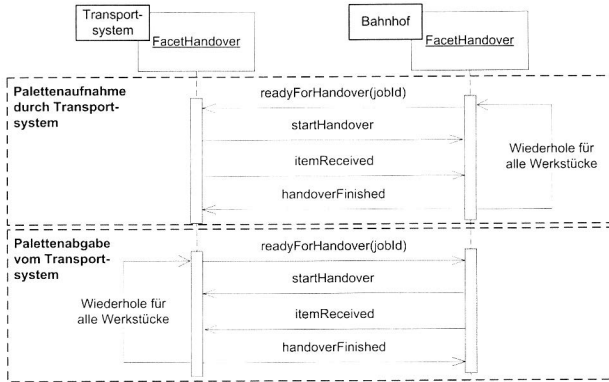


Bild 6.16: Interaktion zwischen dem Transportsystem und dem Bahnhof zur Koordinierung der Werkstückübergabe.

6.4.3), als auch auf Seiten des Transportsystems ein einheitliches Verhalten mit standardisierten Schnittstellen zur Verfügung. Somit kann gegenüber diesen beiden Seiten die konkrete Ausprägung des dahinterliegenden Systems verborgen werden, wodurch die effektive Umsetzung der Modularisierung der mechanischen und elektrotechnischen Anlagenkomponenten als mechatronisches Modul unterstützt werden kann.

## 6.5 Verhandlungsmechanismus zwischen Werkstück und den Ressourcen

In der vorliegenden Architektur werden die Grundlagen für einen durchgängigen Verhandlungsmechanismus zur Verfügung gestellt, der es erlaubt, dass Werkstücke autonom Entscheidungen für den weiteren Verlauf innerhalb des Materialflusses treffen können. Hierzu kann der Werkstückagent bei Bedarf einen Verhandlungsagenten (*Negotiation-Agent*) in Umlauf bringen, der im Auftrag des Werkstücks Verhandlungen vollzieht bzw. im Vorfeld vorgenommene Planungen auf die Durchführbarkeit validiert (Bild 6.17). Der Verhandlungsagent wird dabei für eine spezielle Aufgabe instanziiert und sowohl mit den erforderlichen Parametern, als auch mit einem Skript-Code versehen. Die vorgesehenen Aufgaben werden in Form eines Algorithmus in einem Skript hinterlegt. Dieses Skript koordiniert die Aktionen, die der Verhandlungsagent durchführen muss. Dabei stehen dem Skript Standardfunktionen zur Verfügung. Dazu zählen die Fähigkeit zu migrieren, die Liste aller Ressourcen nach bestimmten Filterkriterien einzuholen oder spezifische Informationen über die Ressourcen abzurufen. Das Skript muss immer zwei spezifische Funktionen enthalten, die zu zwei unterschiedlichen Zeitpunkten aufgerufen werden:

- Funktion *evalMigration*: Diese Funktion wird aufgerufen, bevor eine Migration stattfindet und erhält als Parameter die ID der Bearbeitungsstation, wohin der Verhandlungsagent als nächstes migrieren möchte. Hierbei hat der Verhandlungsagent aus der Liste der erforderlichen Prozesse und der Fähigkeiten der Bearbeitungsstationen eine Vorauswahl getroffen, um somit die

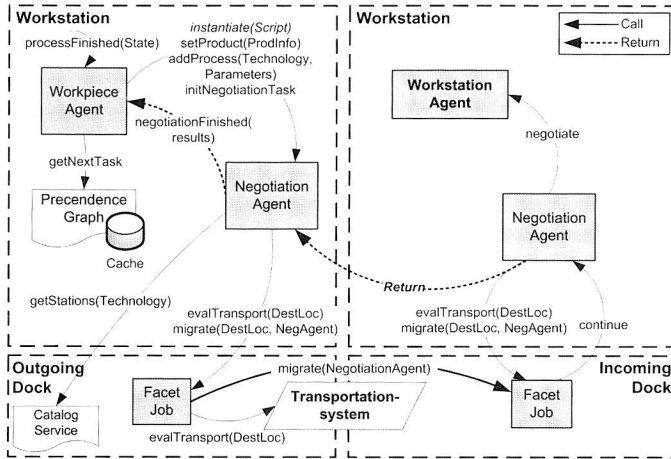


Bild 6.17: Framework zur Umsetzung von Verhandlungsaufgaben zwischen den Teilnehmern des Materialflusses. Der Verhandlungsagent (Negotiation-Agent) verfügt über Migrationsfähigkeiten und übernimmt stellvertretend für das Werkstück Aufgaben zur Evaluierung der optimalen Route in der Produktion.

Grundvoraussetzungen zu erfüllen. Über den Rückgabeparameter der *evalMigration*-Funktion kann aus dem Skript Einfluss auf diesen Vorentscheid genommen werden, zu welchen weiteren Bearbeitungsstationen migriert wird. Im Falle einer Ablehnung einer Station, überliefert die Funktion den Rückgabewert der Konstanten *AbortMigration*.

- Funktion *negotiate*: Diese Funktion implementiert das Verhandlungs- bzw. Evaluierungsverfahren, das an jeder Bearbeitungsstation ausgeführt wird. Hierbei sind alle Vorgänge enthalten, die lokale Interaktionen mit dem Stationenagenten vorsehen, um mögliche Fertigstellungszeitpunkte zu ermitteln und die fristgerechte Durchführbarkeit zu prüfen.
- Funktion *evalResult*: Zum Abschluss der Verhandlungen bewertet diese Funktion das Gesamtergebnis der Verhandlungen. Hierbei kann die Funktion verschiedene Aspekte, wie Kosten oder Zeit, individuell gewichten und dies zu einem skalaren Wert zusammenfassen, der dem Werkstückagenten zur vereinfachten Selektion des günstigsten Weges als Grundlage dient.

Die weiteren Parameter, die dem Verhandlungsagenten übergeben werden, sind die Produktinformationen, die festlegen, welches Zielprodukt bzw. Zwischenerzeugnis am aktuellen Produktionssystem hergestellt werden soll (*setProduct*). Darüber hinaus erhält der Verhandlungsagent die zu untersuchenden nächsten Prozessschritte (*addProcess*), die ursprünglich aus dem Produktionsvorranggraphen stammen. Hierbei verknüpft der Verhandlungsagent die Prozessschritte mit den Strukturdaten der Produktionsanlage (Kapitel 6.2) und erhält im Falle von Redundanzen verschiedene Kombinationen zur Verhandlung der *n* nachfolgenden Schritte. Der Verhandlungsagent folgt dabei immer den primär geeigneten, d.h. günstigsten oder schnellsten, Ressourcen. Sollen alle Alternativen gleichzeitig untersucht werden, so kann sich der Verhandlungsagent an den entsprechenden Stellen duplizieren. Aus

den mehrfachen Instanzen kann sich am Ende der Verhandlungen der Werkstückagent für den geeigneten Weg entscheiden.

Durch Aufruf der *initNegotiationTask*-Funktion wird der Verhandlungsagent beauftragt, mit allen spezifizierten Bearbeitungsstationen für die nächsten  $n$  Bearbeitungsschritte zu verhandeln. Der Verhandlungsagent wendet sich daraufhin an alle Ausgangsbahnhöfe der aktuellen Station. Um zu ermitteln, ob das dahinterliegende Transportsystem eine Station erreichen kann und zu welchen Konditionen (Kosten/Zeit), fragt er über die *evalTransport*-Funktion nach. Bedingt durch die Tatsache, dass für jedes Transportsystem ein eigener Bahnhof existiert und der Verhandlungsagent alle Bahnhöfe berücksichtigt, können auch unterschiedliche Transportsysteme verglichen werden, die in der Lage sind, die nächste Bearbeitungsstationen zu erreichen. Der Verhandlungsagent kann sich für das günstigste Transportsystem entscheiden oder alle Transportwege in Betracht ziehen. Von der Job-Facette des Ausgangsbahnhofs erhält er die geschätzten Transportzeiten und -kosten übermittelt und migriert daraufhin zur Job-Facette des Eingangsbahnhofs der nächsten Ziel-Station. Dort wird er wieder mit *continue* zur Ausführung gebracht. Dabei nimmt er mit dem lokalen Stationsagenten die Verhandlung auf und ermittelt bzw. verhandelt gemäß dem im Skript implementierten Algorithmus den voraussichtlichen Fertigstellungstermin. Im Anschluss daran migriert der Verhandlungsagent über den lokalen Ausgangsbahnhof zur nächsten Bearbeitungsstation, bis alle eingetragenen Stationen besucht wurden. An der letzten Station migriert er direkt zur Ausgangsstation der Verhandlung zurück und teilt dem Werkstückagenten das Gesamtergebnis, d.h. den Weg der Bearbeitung und einen Zahlenwert zur Bewertung dieses Weges mit. Der Werkstückagent wartet lediglich eine bestimmte Zeitspanne auf die Rückkehr aller Verhandlungsagenten, da diese, z.B. im Falle eines Ressourcenausfalls, verloren gehen können. Somit stehen dem Werkstückagenten u.U. die Bewertungen verschiedener Materialflusswege an unterschiedlichen Bearbeitungsstationen zur Verfügung.

## Kapitel 7

# Prototypische Implementierung und Validierung

Im Folgenden wird die prototypische Implementierung des verteilten Steuerungsansatzes vorgestellt. Dieser wurde in verschiedenen Stufen mit unterschiedlicher Ausstattung validiert. Zunächst wird die Implementierung und Validierung der in Kapitel 4 vorgestellten verteilten Steuerungsarchitektur im IT-Labor an einem Teststand mit 23 Rechnern erläutert (Kapitel 7.1 und 7.2). Diese verteilten Steuerungsansätze sind zudem Grundlage für die Realisierung des neuen Steuerungsansatzes für autonom navigierende Transportfahrzeuge (Kapitel 5). Für eingehende Tests dieser Konzepte wurde ein Demonstrationsfahrzeug entwickelt, das hohe Bewegungsfreiheit und umfassende Umgebungserfassung mittels geeigneter Sensorik ermöglicht (Kapitel 7.3). Basierend auf der verteilten Steuerungsarchitektur, in Kombination mit dem autonom handelnden und navigierenden Transportfahrzeug wurde der agentenbasierte Materialfluss im Labor umgesetzt und die Grundmechanismen validiert (Kapitel 7.4). Bei der Implementierung wurde ein hoher Wert auf die Entwicklung allgemeingültiger und allgemein einsetzbarer Softwaremodule unabhängig von der Anwendung gelegt.

### 7.1 Aufbau des Labors für verteilte Steuerungssysteme

Um verteilte Softwaresysteme und Algorithmen insbesondere im Bereich der Wandelbarkeit effektiv testen und validieren zu können, ist ein Testszenario mit mehreren Rechereinheiten erforderlich, in denen die Inbetriebnahme, Umbaumaßnahmen und der Entwicklungsprozess nachgestellt werden können. Da sich die neuartigen Steuerungskonzepte auf die Softwaresysteme konzentrieren, wäre die Evaluierung der Algorithmen in erster Instanz an einer realen Produktionsanlage nicht zielführend gewesen. Mechanische Umbauten zur Nachstellung von Wandlungsszenarien sind zu zeitintensiv, um systematisch eventuelle Fehler in Algorithmen und Implementierungen zu ermitteln. Aus diesem Grund wurde ein Labor für verteilte Steuerungssysteme zur Validierung der Implementierungen eingerichtet.

Das Labor ist mit Steckwänden zur beliebigen Anordnung von Modul-PCs ausgestattet (Bild 7.1). Über eine integrierte Steckverbindung in der Wand werden die Modul-PCs mit Energie versorgt und für Verwaltungszwecke, z.B. für ein Softwareupdate oder für Debugging-Zwecke, an ein Service-Netzwerk angekoppelt. Die Modul-PCs können somit schnell an unterschiedlichen Stellen in der Wand angesteckt werden, wodurch ein flexibler Umbau von Produktionsanlagen nachempfunden werden kann. Die PCs verfügen über vier Netzwerkschnittstellen, die an den vier Seitenwänden angeordnet sind. Über diese können die PCs untereinander mit Crosslink-Netzwerkkabeln verbunden werden, wodurch die Anlagenstruktur gemäß den vorgestellten Konzepten abgebildet werden kann, in denen jede Entität mit einer eigenen Rechereinheit ausgestattet ist. Somit kann der in Kapitel 4.3 vorgestellte Isomorphismus angewendet werden. Mit diesem System können Produktionssysteme unterschiedlicher hierarchischer Ebenen untersucht und nachgebildet werden. Neben der Wandelbarkeit, d.h. der Möglichkeit selbst zur Laufzeit die Produktionsanlage umzubauen, Elemente hinzuzufügen

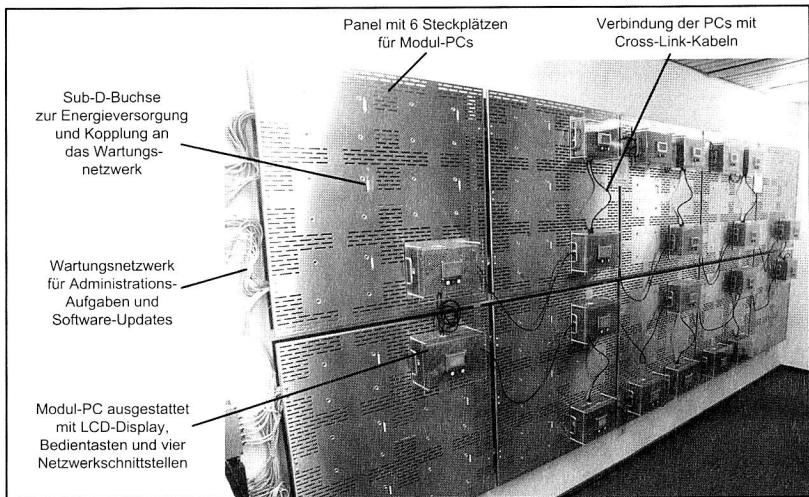


Bild 7.1: IT-Testlabor ausgestattet mit 60 Steckplätzen für 23 Modul-PCs zum Zweck, die Steuerungstechnik für beliebige Anlagensysteme aufbauen und die verteilten Algorithmen evaluieren zu können.

oder wieder zu entfernen, können weitere Szenarien, wie der Ausfall von Entitäten, untersucht werden. Zudem müssen Variationen untersucht werden, in denen Netzwerke bzw. Produktionssysteme partitioniert und ebenso zusammengefügt werden können. Hierbei erfährt das System massive Veränderungen in der Gesamtsteuerungsstruktur, die Auswirkungen auf Materialflussregeln, Konfigurationen und Applikationsverbindungen aller Entitäten im System zur Folge haben. Das zugrundeliegende Steuerungssystem muss auch diesen speziellen Anforderungen genügen und dennoch ausreichende Funktionssicherheiten bieten.

Auf den Modul-PCs kommt eine eigene Linux-Distribution FINUX (FAPS-Linux) zum Einsatz, die mit möglichst wenig Speicherplatz auskommt und durch den angepassten Funktionsumfang Rechenleistung einspart. Somit können die Festplatten durch Compact-Flash-Karten ersetzt und das Betriebssystem zu Updatezwecken über das Netzwerk in doppelter Ausführung - eines als aktives Betriebssystem und eines als Backup-System - darauf gespeichert werden. Da alle Updatevorgänge sowohl des Betriebssystems als auch der Anwendungen über das Netzwerk vorgenommen werden können, lässt sich die nicht-nutzbare Belegzeit maßgeblich reduzieren und die Hardware effektiver nutzen. Hierzu existieren verschiedene Softwarewerkzeuge, mit *binSpector*, dem wichtigsten Vertreter zur Fernadministration der Modul-PCs. Von einem zentralen PC aus können über eine graphische Oberfläche (Bild 7.2) auf alle Modul-PCs gleichzeitig Softwareupdates geladen werden. Ebenso können an allen Modul-PCs die Softwareanwendungen heruntergefahren, neu gestartet, beliebige Protokolldateien eingesehen oder Konfigurationsdateien geändert werden.

Als HMI-Schnittstelle verfügen die PCs über ein graphisches LCD-Display zur Ausgabe von Zuständen und Variablenwerten. Über drei Tasten an der Vorderseite und kippbare Schalter mit drei

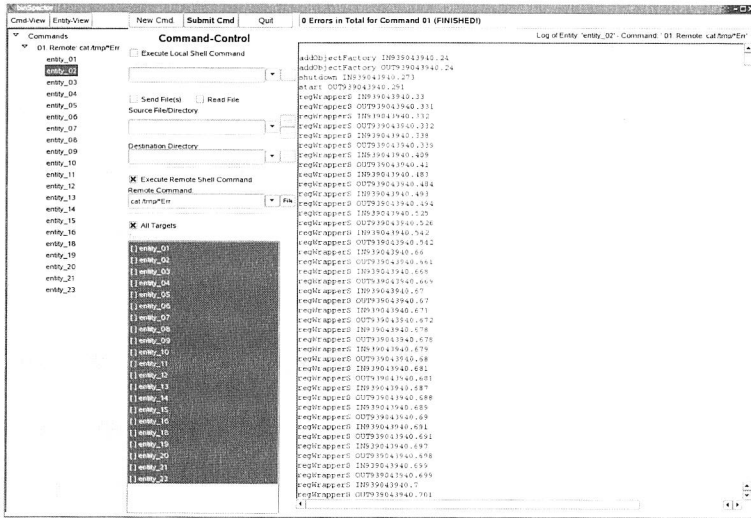


Bild 7.2: *binSpector* ein Remote-Verwaltungs- und Administrationswerkzeug zum automatisierten Verteilen von Programmen und Konfigurationen. Die lokalen Anwendungen können gleichzeitig aus der Ferne auf mehreren Rechnern gestartet, beendet und deren Ausführung überwacht werden.

Stellungen zu jeder der vier Seiten kann mit der Entität interagiert werden. Während die seitlichen Kippschalter primär zur Konfiguration der Flussrichtung des Materialflussesegments vorgesehen sind, stehen die drei Tasten vorne für den beliebigen Einsatz und die beliebige Verschaltung dieser mit dem Ablaufprogramm bzw. der Interaktion mit dem LCD-Display zur Verfügung.

## 7.2 Evaluierung der Steuerungsarchitektur im IT-Labor

Ein Modul-PC im Labor kann in der aktuellen Demonstration eine von zwei möglichen Rollen übernehmen: entweder als Kreuzung (kleine Quadrate), die im reibschlussbasierten Transfersystem das Routen der Werkstückträger übernimmt oder als Bearbeitungsstation (Plattformmodule). Im vorliegenden Beispiel ist ein ringförmiger Haupttransfer mit drei Bearbeitungsstationen im Nebenschluss aufgebaut. Die Netzkabel zwischen den Entitäten stellen dabei die Transfersegmente zwischen den Kreuzungen und Bearbeitungsstationen dar. Die Projektion des Anlagenmodells in 3D wird von einem Visualisierungs-Rechner übernommen, der dieses in Echtzeit anpasst (Bild 7.3). Dieser Rechner ist nur zur Visualisierung vorgesehen und übernimmt keine steuerungstechnischen Funktionen oder Konfigurationsaufgaben.

Der Materialfluss läuft auf jeder Entität völlig dezentral. Hierzu wird auf jeder Entität für den Materialfluss eine Routentabelle zur Laufzeit generiert, die sich an Änderungen in der Anlagenstruktur unmittelbar anpasst [93]. Diese Entitäten sind in der Lage, zur Demonstration virtuelle Werkstückträger zu transportieren. In der ersten Ausbaustufe erhält jeder Werkstückträger einen Arbeitsplan mit allen

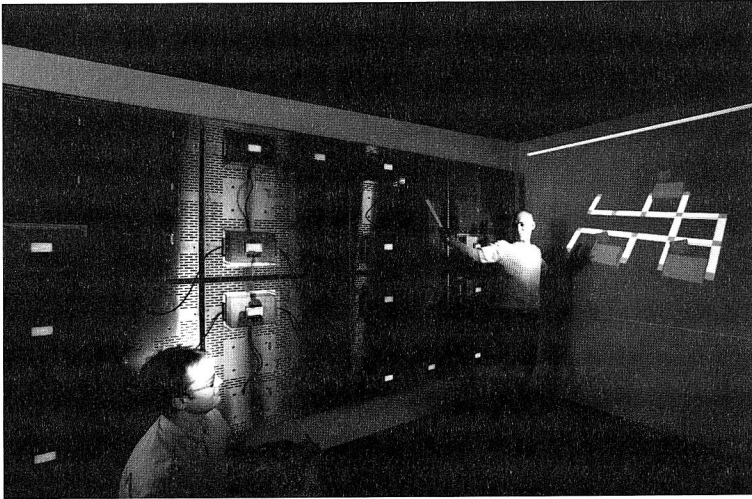


Bild 7.3: Testlabor im Betrieb mit projizierter Anlagenstruktur - das 3D-Anlagenmodell wird zur Laufzeit automatisiert aus der aktuellen Struktur der Modul-PCs generiert. Strukturelle Veränderungen an der IT-Wand sind unmittelbar im 3D-Modell sichtbar.

Bearbeitungsstationen, die durchlaufen werden müssen. In der zweiten Ausbaustufe übernehmen die Agenten die dynamische Bestimmung der nächsten Bearbeitungsstation. Die Kreuzungen entnehmen dem Werkstück die ID der nächsten Bearbeitungsstation und routen den Werkstückträger gemäß der automatisiert erstellten Routentabelle, die nach einem vergleichbaren Verfahren zur IP-Routentabelle (Kapitel 4.3.4) generiert wird.

Das echt verteilte Steuerungssystem mit der damit verbundenen Fähigkeit, dass jede Entität zur Laufzeit selbständig Routenentscheidungen gemäß dem aktuellen Anlagenzustand treffen kann, ermöglicht schnelle Reaktionen auf Anlagenveränderungen oder Störungen. Die veränderte Anlagenstruktur wird vom Nachbarschafts- und Strukturerkennungsalgorithmus in 0,2 Sekunden erkannt und dabei alle Routentabellen angepasst. Somit kann eine sehr hohe Reaktionsfähigkeit gewährleistet werden, in der z.B. ausgefallene Wege, Kreuzungen oder Bearbeitungsstationen erkannt und hierzu Alternativen herangezogen werden. In Bild 7.4 ist ein solches beispielhaftes Szenario dargestellt, in dem zunächst die Produktionsanlage um Bearbeitungsstationen im Nebenschluss erweitert und danach wieder reduziert wird. Somit können Anlagen sehr einfach gemäß dem Produktlebenszyklus, der lediglich während der mittleren Hochphase maximale Absatzzahlen und somit maximale Anlagengröße erfordert, angepasst werden. Zu Beginn kann die Anlage parallel zum steigenden Bedarf erweitert und ebenso zum Ende bei Auslaufen des Produktes wieder sukzessive abgebaut werden. Dieses Steuerungssystem wurde in einem Forschungsvorhaben umgesetzt, bei dem die Bearbeitungsstationen aus Tischmodulen bestehen, die selbst zur Laufzeit flexibel umgestellt werden können (siehe [117]).



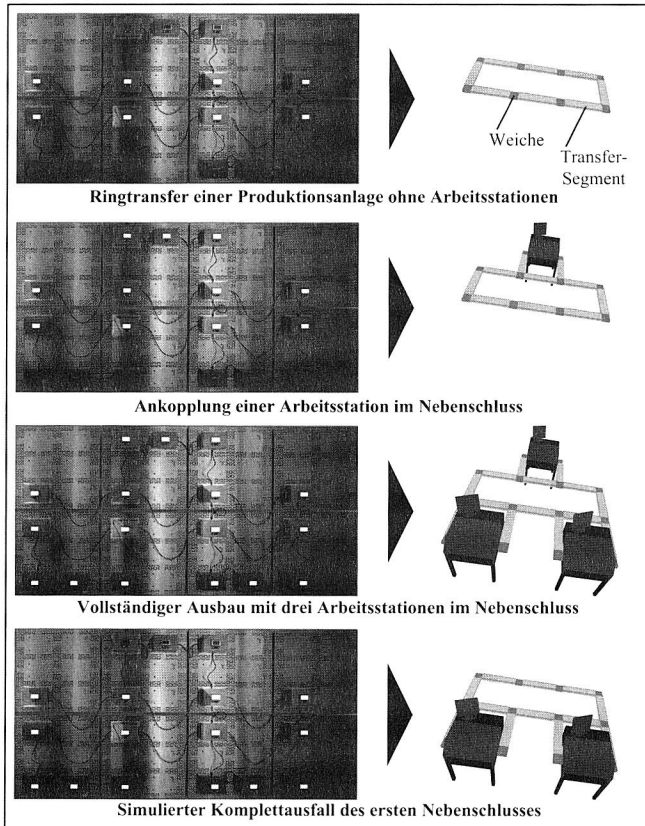


Bild 7.4: Wandlungssequenz einer Produktionslinie - die Modul-PCs emulieren Kreuzungen und Arbeitsstationen, Netzkabel dagegen Transfersegmente [98]. Zum Testen von Anlagenumbauten können die PCs bzw. Kabel zur Laufzeit beliebig umgebaut werden.

## 7.3 Mechanischer Aufbau des Demonstrationsfahrzeugs

Zur Validierung der beschriebenen Verfahren für Fahrerlose Transportsysteme und neuartigen Navigationsalgorithmen im Einsatzfeld der Produktion (Kapitel 5), wurde ein Fahrzeug mit hoher Bewegungsfreiheit konstruiert und am Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik aufgebaut. Die hierbei gewählten mechanischen, elektrischen, sensorischen und steuerungstechnischen Komponenten sind auf eine hohe Modularisierung des Fahrzeugs ausgelegt.

### 7.3.1 Konstruktion eines flexiblen Fahrzeugs mit hoher Bewegungsfreiheit

Das vorliegende Fahrerlose Transportfahrzeug ist für verschiedene Anwendungsfälle ausgelegt. Dies wird durch einen modularen Aufbau gewährleistet, der dieses zunächst in das Grundfahrzeug mit allen fahrzeugrelevanten Komponenten und dem aufsetzbaren Funktionsmodul unterteilt. Somit können verschiedene Funktionsmodule zum Einsatz kommen, die Transportaufgaben für Werkstücke unterschiedlicher Größe wahrnehmen. Zudem ist ein Funktionsmodul z.B. mit integriertem Gerät zur Bearbeitung oder Handhabung von Werkstücken denkbar, wodurch das Fahrzeug in eine mobile Bearbeitungsstation umfunktioniert werden kann. In seiner derzeitigen Funktion besteht die Aufgabe des FTF im Transport von Werkstückträgern zwischen Bearbeitungsstationen (Bild 7.5).

Unter Bild 7.6 ist der Grundriss des Grundfahrzeugs dargestellt. Am Aluminiumrahmen sind alle Komponenten des Fahrzeugs fixiert. Hierzu zählen neben den Antriebs- und Stützrädern auch alle Sensorelemente, die Batterien und die Elektronik mit den Steuereinheiten. Zudem ist ebenso am

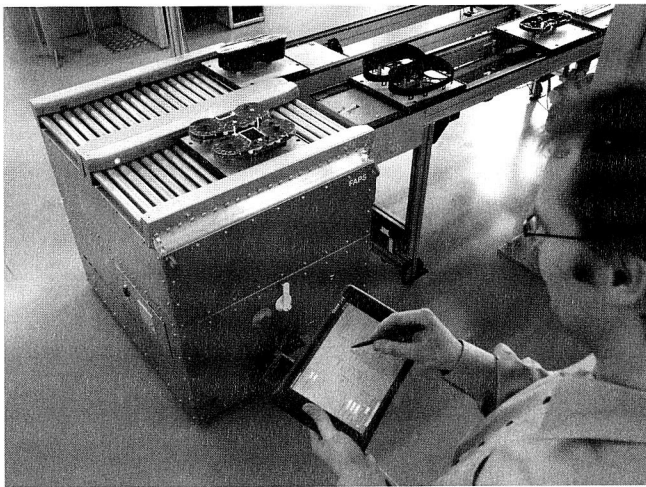


Bild 7.5: Abbildung des autonom navigierenden fahrerlosen Demonstrationsfahrzeugs mit einem modularen Aufsatz zum Werkstücktransport. Es erfüllt alle Voraussetzungen von Seiten der Sensorik, Positionsbestimmung, Steuerungs- und Kommunikationstechnik zur Umsetzung der integrierten Navigations- und eigenständigen Steuerungsfunktionen.

Grundfahrzeug eine Stange befestigt, die auf 2,20m Höhe einen Laserscanner zur Positionsbestimmung trägt. Das Fahrzeug verfügt über zwei gelenkte Antriebsmotoren, die ein hohes Maß an Bewegungsfreiheit gewährleisten. Es kann sowohl seitlich fahren, als auch auf der Stelle drehen und seine ausgesprochen gute Wendigkeit eignet sich insbesondere für enge Gänge und den begrenzten Bewegungsraum im Labor. Somit enthält das Grundfahrzeug alle für die Fahrfunktion relevanten Komponenten. Das Funktionsmodul kann über Zentrierstifte auf das Grundfahrzeug aufgesetzt werden und bildet eine funktional abgeschlossene Einheit, die lediglich am Strom- und Kommunikationsnetzwerk angebunden ist.

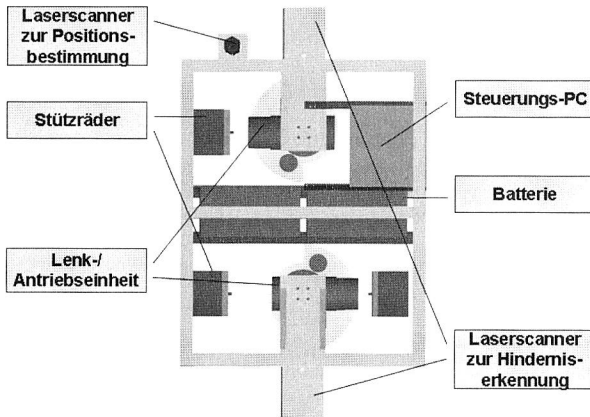


Bild 7.6: Grundriss des mechanischen Aufbaus des Demonstrationsfahrzeugs - mit seinen lenkbaren Antriebsmotoren bietet es eine sehr hohe Bewegungsfreiheit. Dies umfasst ein gleichförmiges Fahren in beide Richtungen, Drehen auf der Stelle und Seitwärtsfahren.

### 7.3.2 Sensorik zur Positionsbestimmung und Erfassung der Umgebung

Damit das Fahrzeug frei in der Umgebung ohne jegliche Bodeninstallationen navigieren kann, verfügt es zur Positionsbestimmung über einen Laserscanner, der mit Hilfe von bekannten Reflektormarkierungen an den Wänden im Raum die Position und Drehlage des Fahrzeugs ermitteln kann (Bild 7.7). Hierzu befindet sich in der Produktionshalle ein Koordinatensystem mit einem festen Ursprungspunkt, in das mit absoluten Positionsangaben die Reflektormarkierungen einvermessen sind. Es werden fünf verschiedene Reflektormarkierungstypen unterschieden, die in beliebiger Reihenfolge angebracht sind, um Mehrdeutigkeiten zu vermeiden.

Zur Erkennung von Hindernissen verfügt das Fahrzeug über Laserscanner und Infrarot-Sensoren zur Abtastung der Umgebung (Bild 7.8). Da das Fahrzeug gleichförmig sowohl nach vorne, als auch nach hinten fahren kann, sind in beiden Richtungen Laserscanner angebracht, die in der Lage sind, Hindernisse bis zu einer Distanz von 15 m zu erkennen und frühzeitig darauf reagieren zu können.

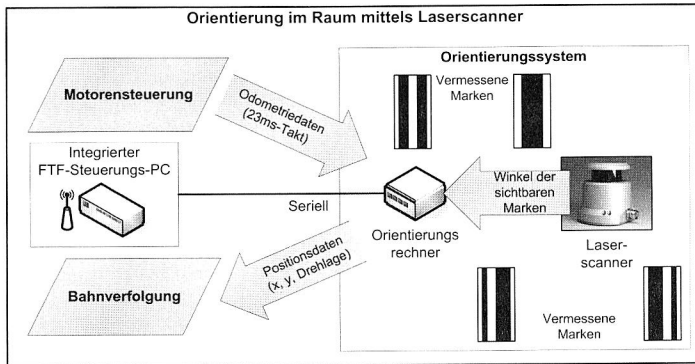


Bild 7.7: Die Orientierung des Fahrerlosen Transportfahrzeugs erfolgt mittels Laserscanner, der über identifizierbare, im Raum vermessene Reflexionsmarken die Position und Drehlage des Fahrzeugs bestimmen kann. Voraussetzung hierzu sind Odometriedaten, die in Echtzeit übermittelt werden müssen. Die Bahnverfolgung übernimmt anhand der Soll-/Ist-Positionen die Lenkregelung.

Die Infrarot-Sensoren an den Seiten gewährleisten die Erkennung von Hindernissen beim Seitwärtsfahren. Sie verfügen nur über eine geringe Reichweite von 30 cm und sind im Abstand von 5 cm angebracht. Ihre Aufgabe besteht lediglich darin, größere Hindernisse oder Personen beim Seitwärtsfahren im Weg zu erkennen und einen sofortigen Stopp einzuleiten.

### 7.3.3 Verteilte Steuerungshardware zur aufgabenbezogenen Steuerung und Regelung

Die interne Steuerung des Fahrzeugs ist modular aufgebaut. Jeder Aktor verfügt über einen eigenen Mikrocontroller, der für die Ansteuerung der zugehörigen Aktorik und Auswertung der daran angebrachten Sensorik zuständig ist, um Regelungs- oder Steuerungsaufgaben wahrnehmen zu können.

Die Mikrocontroller sind über Cross-Link-Ethernet-Kabel mit dem integrierten FTF-Steuers-PC verbunden und kommunizieren mit diesem über TCP/IP. Über ein standardisiertes und geräteunabhängiges Kommunikationsprotokoll wird der modulare Grundansatz unterstützt und die Austauschbarkeit der Geräte ermöglicht. Folgende steuerungstechnischen Module werden unterschieden:

- **FTF-Steuers-PC**  
Übernimmt alle organisatorischen und rechenintensiven Aufgaben. Als übergeordnete Koordinierungseinheit interpretiert der PC die eingelesenen Sensordaten des Positionsbestimmungssystems und der Sensoren zur Umgebungsabtastung und steuert alle Aktoren an. Auf diesen laufen, wie in Kapitel 7.4.2 beschrieben, die verschiedenen Softwaremodule, u.a. zur Bahnverfolgung, Bahnplanung, Auftragsverwaltung und Transportauftragssteuerung.
- **Ein Mikrocontroller pro Lenk-/Antriebsmotor**  
Jeder der zwei im Fahrzeug enthaltenen Lenk- und Antriebsmotoren wird durch jeweils einen

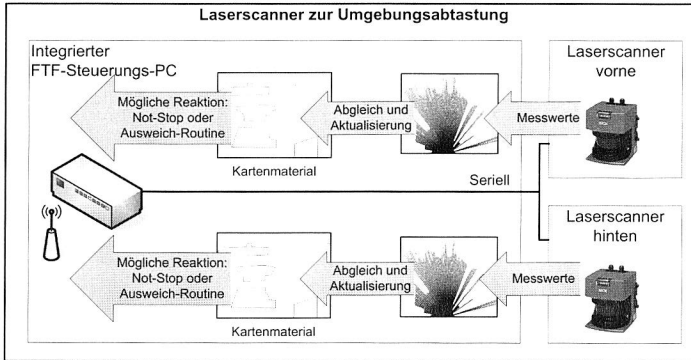


Bild 7.8: Die Laserscanner zur Umgebungsabstastung liefern nach jedem Scan 180 oder 360 verschiedene Distanzwerte, aus denen durch Abgleich mit dem bestehenden Kartenmaterial neue Hindernisse erkannt werden können. Diese haben eine Neuplanung der Route oder selbst einen Not-Stopp zur Folge, sollte eine Kollision mit einem Hindernis drohen.

Mikrocontroller gesteuert und geregelt (Bild 7.9). Hierbei übernimmt der Mikrocontroller sowohl die Lenk- als auch Geschwindigkeitsregelung. Aufgabe der Antriebsregelung ist die Gewährleistung konstanter und gleicher Geschwindigkeiten zwischen den beiden Antriebsmotoren. Dies ist wichtig, da beide Motoren beim Vorwärtsfahren aktiv sind und ungleiche Geschwindigkeiten einen Schlupf beim langsameren Motor zur Folge haben. Darüber hinaus befindet sich auf den Mikrocontrollern ebenso die Lenkregelung, die ständig die Lenkposition prüft und mit  $0,1^\circ$  Genauigkeit nachregelt.

Jeder Mikrocontroller muss zudem die Odometriedaten, d.h. die aktuelle Lenkstellung und die zurückgelegte Strecke in einem hochfrequenten Datenstrom (derzeit ca. 5ms-Intervall) an den PC übermitteln, der die Odometriedaten zur Ermittlung der Fahrzeugposition im 23ms-Takt weiterverarbeitet.

- **Ein Mikrocontroller für die Infrarot-Sensoren**

Die seitlichen Infrarot-Sensoren zur Hinderniserkennung beim Seitwärtsfahren werden von einem Mikrocontroller verarbeitet, der mit einer bestimmten Frequenz jeden Infrarot-Sensor abfragt. Durch die enge Anordnung der Sensoren und die mögliche gegenseitige Beeinflussung, können nicht alle Sensoren gleichzeitig aktiviert sein. Hierzu müssen die Sensoren in zwei Gruppen unterteilt werden, d.h. der ersten Gruppe gehören die Sensoren an den geraden Stellen an, während der zweiten Gruppe die Sensoren der ungeraden Stellen angehören. Während in der ersten Phase die Sensoren aus der ersten Gruppe aktiviert sind und Messwerte liefern, sind die Sensoren der zweiten Gruppe deaktiviert. Umgekehrt werden die Sensoren der zweiten Sensorengruppe gleichermaßen ausgelesen. Die Messwerte werden in einer Callback-Routine dem PC übermittelt, der sich vorab für die kontinuierliche Übermittlung der Messwerte registriert hat. Somit kann der Mikrocontroller bei Erkennen eines Hindernisses den PC unmittelbar informieren und die PC-Software muss nicht kontinuierlich die Werte abfragen und damit Rechenzeit belegen.

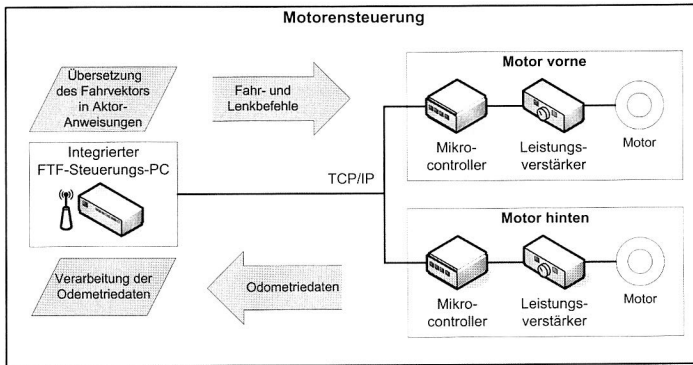


Bild 7.9: Jede Antriebseinheit besteht aus einem lenkbaren Antriebsmotor. Diese wird jeweils durch einen Mikrocontroller geregelt, der zudem die Odometriedaten erfasst und an den Steuerungs-PC weiterleitet.

- **Ein Mikrocontroller für das Aufsatz-Modul zum Werkstückträgertransport**

Die Aufsatzmodule sollen sowohl mechanisch, elektrisch als auch steuerungstechnisch an das Grundfahrzeug über Standardschnittstellen angekoppelt werden. Bei der Steuerungstechnik besteht die Aufgabe in einer strikten Trennung der Aufgaben und Funktionen zwischen dem Grundfahrzeug und den möglichen Aufsätzen. Jedoch kann dies nur dann gewährleistet werden, wenn die Modulsteuerung mit Steuerungsmodulen außerhalb des Fahrzeugs, wie z.B. dem Übergabebahnhof, kommunizieren kann. Da im vorliegenden Anwendungsfall die ressourcenübergreifende Kommunikation über eine Middleware erfolgt und hierbei auf Grund von finanziellen Aspekten kein eigenständiger PC für die Modulsteuerung vorgesehen wurde, fiel die Entscheidung auf die Entwicklung einer Plug&Play-Technologie, die beim Hochfahren des Systems erkennt, welches Modul aufgesetzt ist und dabei die zugehörigen Softwaremodule z.B. zur Interaktion mit den Bahnhöfen am Steuerungs-PC nachlädt (Bild 7.10).

- **Orientierungsrechner zur Bestimmung der Position**

Der Orientierungsrechner ist eine eigene Recheneinheit, die dem Laserscanner zur Positionsbestimmung zugeordnet ist. Seine Aufgabe besteht darin, im 23ms-Takt aus den erkannten Reflektormarkierungen und den gespeicherten Hallendaten die Position des Fahrzeugs im einvermessenen Koordinatensystem zu ermitteln. Die Taktung des Orientierungsrechner wird vom FTF-Steuers-PC vorgegeben, indem dieser alle 23ms ein Paket mit den aggregierten und umformatierten Odometriedaten [118] beider Motoren übermittelt.

- **Mikrocontroller für die Laserscanner**

Die Laserscanner vorne und hinten am Fahrzeug wurden aus Gründen der einfachen Umsetzbarkeit direkt an den PC über eine RS-485-Schnittstelle mit hoher Transferrate (1 MBit) angeschlossen. Hier kann ebenso auf den Einsatz eines Mikrocontrollers zurückgegriffen werden, um dem modularen Ansatz zu folgen. Heutige Mikrocontrollersysteme verfügen bereits über ausreichende Performance und die Schnittstellen zur Anbindung der Laserscanner. Sie können eigenständig Aufgaben übernehmen, wie das Filtern der Messdaten, die Vorverarbeitung, d.h.

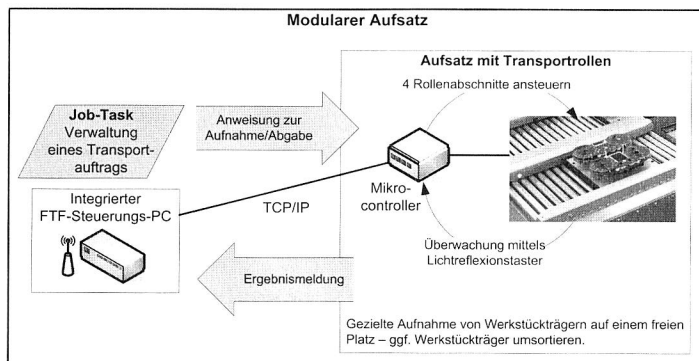


Bild 7.10: Der modulare Aufsatz verfügt über eine eigene, integrierte Mikrocontrollersteuereinheit, die über TCP/IP mit dem Steuerungs-PC verbunden ist.

das Umrechnen in Koordinaten oder das direkte Einleiten von Not-Stopps im Falle von möglichen Kollisionen naher Hindernisse. Im letzten Fall kann die Sensorik direkt mit den Mikrocontrollern der Motoren interagieren, um alle kritischen Maßnahmen unmittelbar einzuleiten. Aus Kostengründen wurde hierbei auf den Einsatz weiterer Controller verzichtet.

Der FTF-Steuerungs-PC ist sowohl für die Steuerung des Fahrzeugs als auch für die Interaktion mit den Übergabebahnhöfen verantwortlich (Bild 7.11). Seine Aufgaben lassen sich wie folgt zusammenfassen:

- **Anbindung an die Zellen über die verteilte Middleware-Steuerung**

Die in Kapitel 4 vorgestellte Middleware-Lösung zur ortstransparenten Interaktion verteilter Steuerungssysteme über das Netzwerk kommt neben den Steuerungseinheiten für die Bearbeitungsstationen und die Kreuzungsmodule ebenso auf dem PC des Fahrerlosen Transportfahrzeugs zum Einsatz. Hierüber kann sich das Fahrzeug mit beliebigen Übergabebahnhöfen zur Koordination der Werkstückübergabe und zum Job-Management verbinden. Die Schnittstellen auf beiden Seiten müssen normiert sein und somit die gerätespezifischen Implementierungen verbergen.

- **Verhandlung der Transportaufträge gemäß dem Agentenansatz**

Über die Steuerungsmiddleware werden Transportaufträge zwischen den Übergabebahnhöfen mit den zu transportierenden Werkstücken und den Fahrzeugen gemäß dem in Kapitel 3 beschriebenen Agentenansatz verhandelt. Hierbei übernimmt das Job-Management des Fahrzeugs Aufgaben zu Zeit- und Kostenabschätzungen für Gebotsanfragen und lastet die angenommenen Transportaufträge in die lokale Auftragsverwaltung ein.

- **Organisation aller Transportaufträge**

Jedes Fahrzeug organisiert selbständig alle angenommenen Transportaufträge in einer vorsortierten Warteschlange mit priorisierten Aufträgen und entnimmt diese dem FIFO-Prinzip. Mit weiterführenden Optimierungsstrategien können mehrere Aufträge zusammengefasst werden,

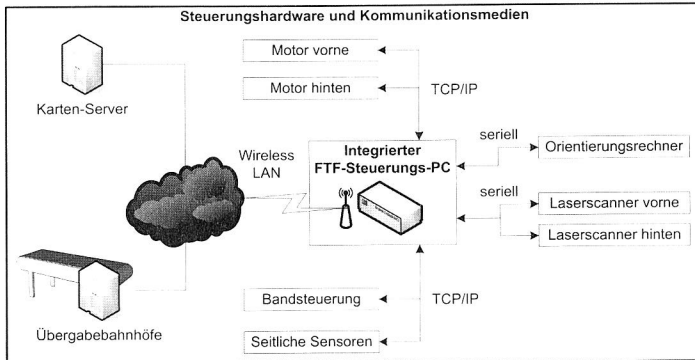


Bild 7.11: Die Steuerungshardware des Fahrerlosen Transportfahrzeugs besteht aus einem zentralen Steuerungs-PC, der an die Sensoren und Aktoren über TCP/IP oder die serielle Schnittstelle angebunden ist.

die auf einer Strecke liegen, wodurch eine Zeitersparnis und ein höherer Durchsatz erzielt werden kann. Der Job-Manager ist für die Aktivierung der Aufträge zuständig. Jeder Auftrag übernimmt dabei selbständig die Durchführung und somit die Koordinierung aller Teilschritte eines Transportauftrags. Somit ist dieser für die vollständige Ausführung verantwortlich. Unauflösbare Fehler führen zum Abbruch des Transportauftrags und ggf. zum Einleiten einer Sonderbehandlung, in der z.B. ein bereits aufgenommenes Werkstück an einem definierten Lager abgegeben wird.

- **Interaktion mit den Übergabebahnhöfen zur Werkstückübergabe**

Damit ein autonom navigierendes Fahrzeug sich vor den Bahnhof stellen darf, muss es bei diesem anfragen, ob er frei ist. Somit können Kollisionen mit bereits davorstehenden Fahrzeugen vermieden und die Belegung des Übergabebahnhofs kann priorisiert werden, falls mehrere Fahrzeuge zum gleichen Zeitpunkt eintreffen sollten.

Wenn sich das Fahrzeug vor dem Übergabebahnhof positioniert hat, muss die Übergabe in einem Handshakeverfahren koordiniert werden. Voraussetzung hierfür ist die Bereitschaft des Fahrzeugs und der Übergabestation, die gegenseitig signalisiert werden muss. Im Anschluss daran kann das Werkstück übergeben werden. Zur Kontrolle bestätigen sich beide Teilnehmer gegenseitig, das Werkstück erfolgreich aufgenommen bzw. abgegeben zu haben. Nur in diesem Fall kann das Fahrzeug abkoppeln und die weitere Fahrt fortsetzen (vgl. Kapitel 6.4.4).

- **Verwaltung der Kartendaten**

Jedes Fahrzeug verfügt über Kartenmaterial, anhand dessen die Bahn in der Halle geplant wird. Das Kartenmaterial wird in Form von Polygonen in einer Datenbank gespeichert und kontinuierlich angepasst. Hierbei werden sowohl neue Hindernisse über die eigene Sensorik erkannt und eingepflegt, als auch Änderungen am Kartenmaterial durch den Mitarbeiter bzw. von anderen Fahrzeugen über das WLAN-Netzwerk übernommen.

- **Planung der Route für jeden Transportauftrag**

Zur Durchführung eines Transportauftrags muss die erforderliche Route im Pfadplaner, beste-



hend aus zwei Teilstrecken, ermittelt werden, um zunächst zum ersten Übergabebahnhof zu gelangen und anschließend den Transportauftrag auszuführen. Neben den zweidimensionalen Positionangaben müssen weitere Informationen in der Route hinterlegt werden, um verschiedene Bewegungsmodi wie das Drosseln der Geschwindigkeit nahe dem Ziel oder das Seitwärtsfahren zu ermöglichen.

- **Trajektorienverfolgung mit Soll-/Ist-Positionsregelung**

Wenn die abzufahrende Route errechnet wurde, muss sich das Fahrzeug entlang dieses Pfades bewegen. Dabei soll es möglichst gleichförmig um die Kurven fahren und nicht darauf angewiesen sein, auf der Stelle drehen zu müssen. Da dies nicht alle Lenk- und Antriebskonzepte zulassen, muss der geplante Pfad in eine fahrbare Trajektorie umgewandelt werden. Während der Fahrt greift die Positionsregelung der FTF-Steuerung, die in kurzen Zeitintervallen (kleiner 1 Sekunde) einen Soll-/Ist-Positionsabgleich vornimmt und die Lenkmotoren des Fahrzeugs ansteuert, um möglichst genau der geplanten Bahn zu folgen und Kollisionen mit den Hindernissen zu vermeiden.

- **Erfassung der Odometriedaten beider Motoren und Weiterleitung dieser an den Orientierungsrechner**

Der FTF-Steuerungs-PC erhält von den Mikrocontrollern beider Motoren die aktuellen Odometriedaten, d.h. die zurückgelegte Wegstrecke und die aktuelle Lenkstellung, und sendet diese im Zyklus von 23 ms an den Orientierungsrechner. Dieser ist auf die Odometriedaten angewiesen um in der sehr kurzen Zykluszeit die exakte Position im Raum zu finden. Über die Odometriedaten wird ein Suchfenster definiert, innerhalb dessen mit dem Laserscanner die genaue Position ermittelt wird. Im Ausnahmefall, wenn keine gültigen Marken erkannt werden, kann der Orientierungsrechner auch nur mit den Odometriedaten die Position berechnen.

- **Koordinierung der detaillierten Schritte eines Transportauftrags**

Am FTF-Steuerungs-PC laufen alle Informationskanäle zusammen. Die verteilten Mikrocontroller übernehmen die lokale Steuerung und Regelung, sind jedoch auf Anweisungen des PCs angewiesen bzw. liefern diesem alle erforderlichen Werte aus den Sensordatenauswertungen. Hierbei stellt der PC im komplexen Gebilde eines autonom agierenden Fahrerlosen Transportfahrzeugs den Leitreechner dar, der, angefangen von der Abtastung der Umgebung, über die Steuerung der Aktorik, wie Motoren, Transportrollen etc., bis hin zur logistischen Koordinierung und Durchführung der Aufträge vielfältige und sehr rechenintensive Aufgaben wahrnimmt.

## 7.4 Referenzimplementierung im Labor zur Evaluierung der agentenbasierten Materialflusssteuerung

Die Konzepte eines durchgängigen, dezentralen Materialflusssteuerungssystems mit autonom agierenden Einheiten wurde im Labor des Lehrstuhls für Fertigungsautomatisierung und Produktionssystematik implementiert [119]. Dabei wurden fünf verschiedene Bearbeitungsstationen und das aus Kapitel

7.3 beschriebene Fahrerlose Transportfahrzeug angebunden. Das Fahrzeug bewerkstelligt den internen Materialfluss zwischen den Stationen und kann darüber hinaus im Sinne des Agentenansatzes autonome Entscheidungen treffen und sich frei und unabhängig von einer zentralen Koordinationsinstanz im Raum bewegen. Das Fahrzeug ist somit in der Lage, selbständig mit den Übergabebahnhöfen der Bearbeitungsstationen in Interaktion zu treten, um zum einen Aufträge zu verhandeln und anzunehmen und zum anderen den Werkstück austausch, unterstützt durch ein Kommunikationsprotokoll, zu koordinieren.

### 7.4.1 Vermessung der Halle

Voraussetzung für die autonome Navigation ist die manuelle Erfassung und Vermessung der Umgebung, insbesondere der statischen Hindernisse und der Übergabebahnhöfe. Gemäß dem Konzept aus Kapitel 5.3 erfolgt die Vermessung mittels eines Handwagens, der mit dem gleichen Messsystem zur Positionsbestimmung der Fahrzeuge ausgestattet ist. Somit müssen die Werte nicht zwischen unterschiedlichen Koordinatensystemen transformiert und folglich können Fehler bei der Abbildung der realen Umwelt auf die virtuelle Karte vermieden werden.

Da in der vorliegenden prototypischen Implementierung des Demonstrationsfahrzeugs (vgl. Kapitel 7.3) ein Lasermesssystem zur Positionsbestimmung zum Einsatz kommt, wurde hier ein Handwagen ebenfalls mit dem gleichen Laserscanner entwickelt. Das Konzept lässt sich jedoch auf beliebige Positionserfassungssysteme, u.a. Rasternavigation [105] oder GPS-Navigation [120], anwenden.

Der Laserscanner wird in einer möglichst großen Höhe angebracht, um die meisten Hindernisse, u.a. den Menschen, zu überragen und dabei möglichst freie Sicht auf die Reflektionsmarken zu erzielen. Daher wird, analog zum Demonstrationsfahrzeug, der Laserscanner auch hier auf einer Stange angebracht. Damit der Orientierungsrechner, der die Recheneinheit zum Laserscanner darstellt, die Position im Raum während der Bewegung verfolgen und eindeutig bestimmen kann, benötigt dieser Odometriedaten, d.h. die Lenkungsstellungen und den zurückgelegten Weg der Räder. Aus diesem Grund wurde ein Handwagen mit passiven Rollen konstruiert, der in der Lage ist, die Odometriedaten aufzunehmen und an den Orientierungsrechner weiterzugeben. Somit entspricht der Handwagen einem Fahrerlosen Transportfahrzeug mit einem Differentialantrieb, da es lediglich genügt an den mittleren zwei Rädern links und rechts die Wegstrecke zu erfassen. Die Erfassung der Wegdaten erfolgt über Inkrementalgeber an diesen zwei festgestellten Rollen (Bild 7.12). Zur Aufnahme von Messpunkten stehen dem Anwender drei verschiedene Stellen am Handwagen zur Verfügung, die er nach Bedarf zur Erhöhung der Bewegungsflexibilität auswählen kann. Hierzu wird dem Orientierungsrechner der entsprechende Verschiebevektor des Messpunktes relativ zur Position des Laserscanners vorgegeben.

Die Erfassung, Weiterleitung und Verarbeitung aller im Handwagen angefallenen Daten übernimmt ein Mikrocontroller (Bild 7.13). Auf steuerungstechnischer Seite bildet der Mikrocontroller das Bindeglied zwischen Sensorik, Benutzerschnittstelle und dem Karten-Server. Zunächst müssen vom Mikrocontroller die Odometriedaten in Echtzeit erfasst und an den Orientierungsrechner weitergeleitet werden, damit dieser die Position aus der Koppelnavigation mit jener aus dem Laserscanner interpolieren und somit die Ermittlung der Position im 23ms-Takt gewährleisten kann [107]. Nach jedem Versand der Odometriedaten im 23ms Zyklus erhält der Mikrocontroller umgekehrt vom Orientie-

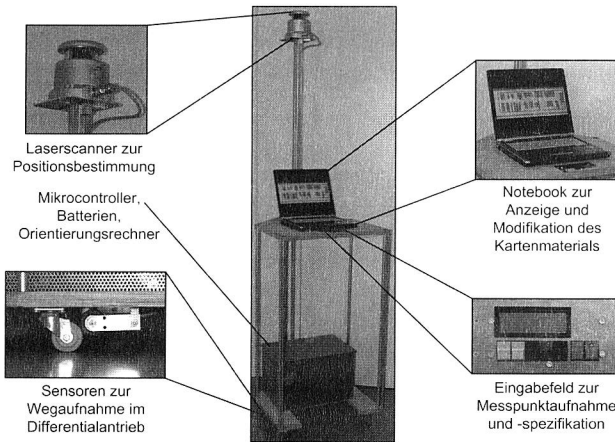


Bild 7.12: Handwagen zur Vermessung der Produktionshalle und Erfassung der stationären Hindernisse und Bahnhöfe. Die Steuerung übernimmt ein Mikrocontroller, der direkt an das Wireless-LAN gekoppelt ist und neue Hindernisdaten direkt übermitteln kann. Der Kartenzugriff mittels Notebook kann optional in Kombination oder auch unabhängig vom Handwagen genutzt werden.

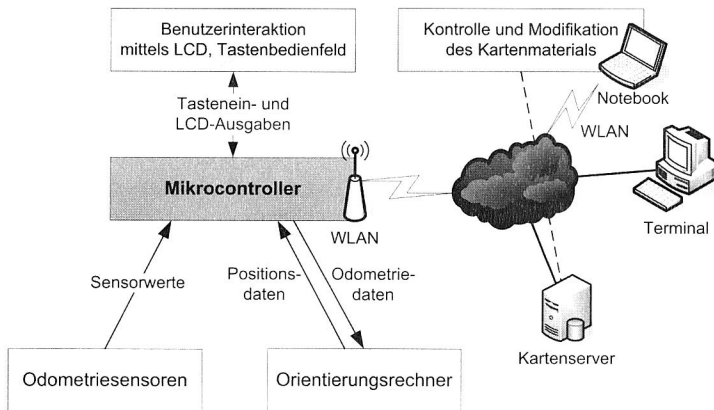


Bild 7.13: Steuerungsstruktur des Handwagens mit Vernetzung zum Kartenserver. Die Anzeige und Modifikation erfolgt über PCs, die per LAN oder WLAN gekoppelt sind und alle Änderungen unmittelbar anzeigen können.

rungsrechner die aktuellen Positionsdaten, die zur Aufnahme der Messpunkte herangezogen werden.

Ausgestattet mit einem WLAN-Adapter kann der Mikrocontroller das aktuelle Kartenmaterial unmittelbar an den Kartenserver und gleichzeitig an alle Fahrzeuge übermitteln. Somit wird der Gedanke aus Kapitel 5.1 der integrativen Lösung mit geringem Inbetriebnahmeaufwand ganzheitlich verfolgt. Der Kartenserver stellt die zentrale Datenbasis für das Kartenmaterial dar, an der sich jedes Fahrzeug beim Hochfahren bzw. auch der Handwagen die aktuellen Daten herunterladen kann. Der Kartenserver ist nicht unbedingt erforderlich, da die Daten auch von anderen Fahrerlosen Transportfahrzeugen abgefragt werden können. Somit kann auf diesen Single-Point-of-Failure verzichtet werden, bietet jedoch die Möglichkeit einfachere Algorithmen zur Verteilung eines konsistenten Datenbestands auf die Fahrzeuge zu realisieren.

Der Handwagen bietet über die Interaktionen mit dem Bediendisplay die Möglichkeit, ein Notebook anzuschließen, auf dem das gesamte Kartenmaterial graphisch angezeigt wird. Sowohl die direkte Interaktion mit dem Kartenserver zum Herunterladen und manuellen Anpassen des aktuellen Kartenmaterials als auch die Interaktion mit dem Handwagen zur Vermessung neuer Hindernisse ist gewährleistet. Dem Anwender stehen somit zwei alternative Eingabemöglichkeiten zur Verfügung, die ihm erlauben, abhängig vom Änderungsumfang, ein Notebook zur besseren Visualisierung heranzuziehen.

Nach erfolgreicher Erfassung des Kartenmaterials können die Fahrzeuge unmittelbar mit den neuen Daten die Routen berechnen. Bei der Umsetzung wurde ein großer Wert auf die Plug&Produce-Fähigkeit des Gesamtsystems gelegt, um selbst zur Laufzeit Änderungen am Kartenmaterial vornehmen zu können und diese Daten während des Betriebs an allen Fahrzeugen zu aktualisieren. Somit muss weder das Transportsystem noch sonstige Teilsysteme zur Einspielung der aktuellen Daten heruntergefahren werden. Gemessen an den Anforderungen für wandelbare Produktionsumgebungen, eignet sich dieser Ansatz maßgeblich für diese Einsatzgebiete.

### 7.4.2 Autonome Fahrzeugsteuerung

Die Umsetzung der in Kapitel 5 beschriebenen Ansätze für autonom navigierende Fahrerlose Transportfahrzeuge, resultiert in einem modularen Ansatz integrierter Softwaremodule, die auf dem Steuerungs-PC des Fahrzeugs ausgeführt werden (Bild 7.14). Die Fahrzeugsteuerung übernimmt die Aufgaben der Peripherieansteuerung, Bahnplanung, Bahnverfolgung, Transportauftragssteuerung und Koordination der Transportaufträge. Zudem muss die Interaktion mit den Übergabebahnhöfen gewährleistet werden, um Transportaufträge zu verhandeln und die Werkstückübergabe zu koordinieren.

Der zentrale Kern der Fahrzeugsteuerung übernimmt die Anbindung aller Peripheriegeräte und kapselt diese über standardisierte Schnittstellen gegenüber der restlichen Steuerung. Die angebundenen Sensoren umfassen zum einen die Laserscanner zur Hinderniserkennung, als auch den Laserscanner zur Positionsbestimmung. Letzterem ist ein Orientierungsrechner zugeordnet, der gemäß den erfassten Marken und den Sensorwerten des Laserscanners, die Position im Raum bestimmt. Dieser muss in weicher Echtzeit über eine serielle Schnittstelle [121] mit Odometriedaten der Motoren versorgt werden, um daraus ein Suchfenster zur Positionsbestimmung ableiten zu können [122]. Hierzu kommt ein interruptgesteuerter Prozess zum Einsatz, der mittels eines Timerinterrupts zur Ausführung einer Interrupthandleroutine in konstanten Zeitabschnitten unterbrochen wird. Diese Routine erfasst

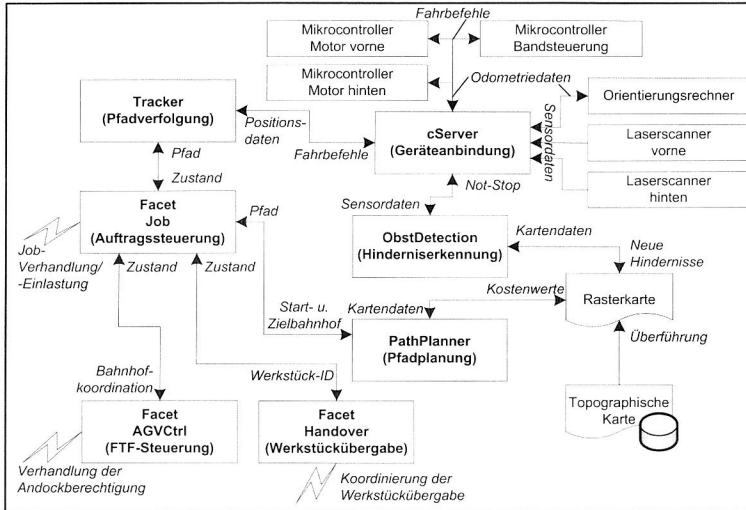


Bild 7.14: Steuerungssystem des autonom navigierenden Fahrerlosen Transportfahrzeugs gegliedert in funktional getrennte Programmmodule für interne und externe Aufgaben.

die aktuellen Odometriedaten, d.h. die seit dem letzten Auslesen erfassten Inkremente des zurückgelegten Weges und die aktuelle Lenkstellung, und leitet diese in einem Telegramm an den Orientierungsrechner des Laserscanners weiter. Der Orientierungsrechner antwortet daraufhin unmittelbar mit einem Telegramm, das die aktuelle Position enthält. Die beiden Laserscanner zur Hinderniserkennung bzw. die seitlichen Infrarot-Sensoren, angesteuert durch einen Mikrocontroller über TCP/IP, werden jeweils mittels Polling-Verfahren ausgelesen und die Messwerte in je zwei Puffern hinterlegt. Ein Puffer enthält somit immer eine vollständige Messung, während der andere Puffer für den aktuellen Einlesevorgang herangezogen wird.

Die Mikrocontroller der gelenkten und angetriebenen Motoren sind über TCP/IP an den Steuerungskern angeschlossen, der wiederum die Schnittstelle zu den weiteren Softwaremodulen, wie der Bahnverfolgung, bildet, um transparent auf diese zuzugreifen. Zudem informieren die Mikrocontroller der Motoren den Steuerungskern immer über die Odometriedaten, d.h. den Inkrementen des zurückgelegten Weges und den Lenkwinkel, sobald Änderungen vorliegen. Der Steuerungskern nimmt die Absolut-Werte entgegen und hinterlegt die Delta-Werte seit der letzten Übertragung an den Orientierungs-Rechner.

Der Mikrocontroller des aufgesetzten Moduls - in diesem Fall die Transportrollen des Demonstrationsfahrzeugs (Kapitel 7.3) - ist direkt mit der Transportauftragssteuerung verbunden. Gemäß dem modularen Ansatz werden die Transportrollen autark von einem integrierten Mikrocontroller gesteuert, der vom Steuerungs-PC Befehle zur Aufnahme oder Abgabe von Werkstücken erhält und direkt vor Ort die Aktoren steuert und Sensoren abfragt. Nach Abschluss des Übergabeauftrags oder im Feh-

lerfall meldet sich der Mikrocontroller bei der Transportauftragssteuerung zurück.

Der Steuerungs-Kern stellt intern die gerätespezifischen Schnittstellen über einen *Shared-Memory* zur Verfügung, der eine Interprozesskommunikation zwischen verschiedenen Prozessen ermöglicht [123, 124]. Dabei ist der Shared-Memory in Bereiche unterteilt, deren Größe abhängig von der abzulegenden Datenmenge für die einzelnen Objekte ausgelegt ist. Diese Funktionalität wird von einem Shared-Memory-Manager zur Verfügung gestellt, der zudem Signalmechanismen zur Interaktion zwischen den verschiedenen Prozessen bereitstellt. Diese werden intern über Semaphoren realisiert. Dem Entwickler stehen Werkzeuge und Bibliotheken zur Verfügung, um Ereignisse aufwandsarm zu deklarieren und zu verwenden. Die benachbarten Prozesse registrieren sich in einem dedizierten Server-Thread für ein oder eine Menge von Ereignissen und legen sich bis zum Signalempfang schlafen. Hierüber werden u.a. Änderungen am Shared-Memory-Speicher mitgeteilt und hierüber indirekt Funktionen benachbarter Prozesse aufgerufen bzw. Rückgabeparameter übergeben. Über diesen Mechanismus können verschiedene Prozesse mit dem Steuerungs-Kern kommunizieren, um somit zeitnah Befehle zu übermitteln bzw. über neue Sensorwerte informiert zu werden. Zu diesen Daten zählen insbesondere die Messwerte zur Hinderniserkennung, als auch die aktuelle Position, die unmittelbar über diesen ereignisbasierten Signalmechanismus an die Interessenten, wie Bahnverfolgung oder Hinderniserkennung, weitergeleitet werden.

Direkt an den Steuerungskern sind der Pfadverfolger (*Tracker*), die Hinderniserkennung und der Autonomous Guided Vehicle (AGV)-Manager angebunden. Der Pfadverfolger erhält die vom Pfadplaner (*PathPlanner*) ermittelten Eckpunkte des zu verfolgenden Pfades und steuert die Motoren über den Steuerungs-Kern. Die Gesamtkoordination zur Abwicklung eines Transportauftrags übernimmt die AGV-Manager-Facette, die von der AGV-Control-Facette beauftragt wird. Die AGV-Control-Facette ihrerseits erhält die Transportaufträge von der Job-Facette, die mittels Verhandlungsmechanismen die Einlastung der Transportaufträge vornimmt.

Die AGV-Manager-Facette gibt aus den vorliegenden Auftrags-Daten dem Pfadplaner die anzufahrenden Bahnhöfe vor, steuert ebenso den Datenfluss zwischen dem Pfadplaner und Pfadverfolger und stößt die Werkstückübergabe über die AGVCtrl-, Handover- und Feeder-Facetten an (vgl. Kapitel 6.4). Der Pfadplaner benötigt als Eingangsgröße die Kartendaten der Produktionsumgebung. Diese liegen primär in Form von Polygonen vor, die in einem ersten Schritt in eine Zellenmatrix als Basis für den zellenbasierten Pfadplaner überführt werden müssen (vgl. Kapitel 5.6). Diese zellenbasierte Karte ist zudem Grundlage für das Hinderniserkennungsmodul, das mit einem schnellen Zugriff in der Zellenkarte prüfen kann, ob ein Hindernis bereits bekannt ist. Hierbei liest das Hinderniserkennungsmodul die Messdaten der Laserscanner aus und prüft, ob die Endpunkte der Laserstrahlen auf eine Zelle in der Zellenkarte zeigen, die bereits belegt ist. Wird jedoch ein Hindernis erkannt, das sich zudem auf dem in der Zellenkarte markierten, zuletzt geplanten Pfad befindet, wird der AGV-Manager informiert, der am Pfadplaner anhand der modifizierten Kartendaten eine Neuplanung veranlasst. Daraufhin wird der Pfadverfolger vom Job-Manager unterbrochen und der neue Pfad eingelastet. Somit besitzt immer der AGV-Manager die Kontrolle über das Fahrzeug und die Abwicklung des aktuellen Transportauftrags, während die Job-, AGV-Control- und Handover-Facetten das Fahrzeug nach außen vertreten und mit den Bahnhöfen direkt interagieren.

### 7.4.3 Anbindung der Bahnhöfe an die bestehende Zellensteuerung

Zur Einbindung der Bearbeitungsstationen mit ihren bestehenden Steuerungssystemen wurde ein Kommunikationsprotokoll zwischen der Stationssteuerung und dem Bahnhof entwickelt. Dieses kann sowohl über das Netzwerk, z.B. über UDP, als auch über digitale Kommunikationsleitungen umgesetzt werden. Abhängig von den technischen Möglichkeiten zur Anbindung der bestehenden PC- und SPS-basierten Steuerungssysteme, kann für jede Station aus einer Menge verschiedener technischer Möglichkeiten der Interaktionen eine ausgewählt werden.

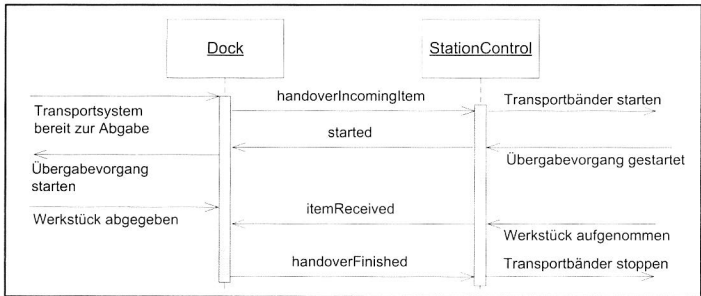


Bild 7.15: Kommunikationssequenz mit der Zellensteuerung zur Anbindung des Eingangsbahnhofs.

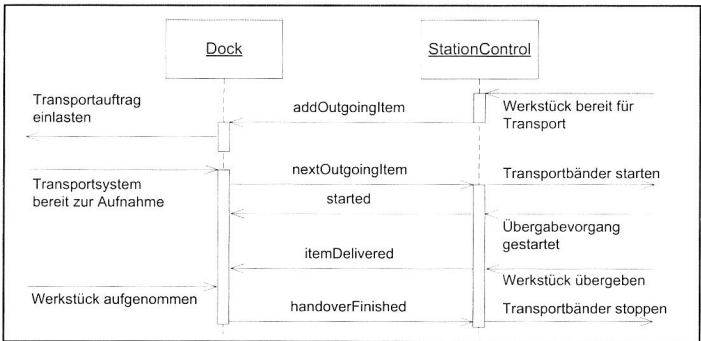


Bild 7.16: Kommunikationssequenz mit der Zellensteuerung zur Anbindung des Ausgangsbahnhofs.

In Bild 7.15 ist die Kommunikation zwischen dem Eingangsbahnhof und der Steuerung der Station beschrieben. Ist ein neues Werkstück zur Aufnahme verfügbar, fordert der Bahnhof über die Nachricht *handoverIncomingItem* die Stationssteuerung zur Aufnahme auf. Sobald diese bereit und die Transportbänder angelaufen sind, meldet sie mittels *started* die Bereitschaft. Die über Sensorik erfolgreich erkannte Aufnahme des Werkstücks wird über *itemReceived* dem Bahnhof mitgeteilt. Zur Übergabe des Werkstücks wird erneut vom Bahnhof ein *handoverIncomingItem* gesendet und die Kommunikation beginnt von vorne. Zum Schluss, z.B. wenn alle Werkstücke eines Fahrzeugs abgegeben wurden,

sendet der Bahnhof ein *handoverFinished* an die Stationssteuerung, damit diese informiert ist, dass der Aufnahmevorgang beendet und ggf. die Transportbänder angehalten werden können.

Die Kommunikation des Ausgangsbahnhof (Bild 7.16) läuft auf die gleiche Art und Weise ab, wie beim Eingangsbahnhof, lediglich mit dem Unterschied, dass die Nachrichten den Ausgang des Werkstücks anstoßen bzw. bestätigen. Zur Ausgabe des Werkstücks sendet der Ausgangsbahnhof *nextOutgoingItem* und fordert die Station auf, das nächste in der FIFO-Warteschlange befindliche Werkstück zu übergeben. Da die erfolgreiche Übergabe mittels Sensorik erkannt werden kann, bestätigt die Station die erfolgreiche Übergabe mittels *itemDelivered*. Die restlichen Nachrichten übernehmen die gleiche Funktion wie beim Eingangsbahnhof.



## Kapitel 8

### Zusammenfassung

Einer der wesentlichen Schlüsselfaktoren für die Wettbewerbsfähigkeit produzierender Unternehmen stellt Wandelbarkeit dar, die in Zukunft immer mehr an Bedeutung gewinnen wird. Dabei sehen sich die Produzenten neben den Anforderungen einer flexiblen Organisation der Produktionssysteme zunehmend mehr mit den Herausforderungen dynamischer und struktureller Veränderungen des Anlagenaufbaus konfrontiert. Individuelle Kundenwünsche, hohe Produkt- und Variantenvielfalt und geringe Losgrößen sind die wesentlichen Treiber, die eine steigende Wiederverwendung von Anlagenkomponenten zur Gewährleistung der Wirtschaftlichkeit erfordern.

Die Unterstützung der strukturellen Wandelbarkeit muss dabei u.a. in der Automatisierung gewährleistet werden, indem der Entwicklungsaufwand reduziert und die Komplexität der Systeme mittels Modularisierung beherrschbar gemacht werden. Die durchgängige Modularisierung ist dabei in allen Fachbereichen der Produktionstechnik erforderlich, um effektiv die Vorzüge des modularen Anlagenaufbaus nutzen zu können. Die modulare Gestaltung von Produktionsanlagen von mechanischer und energetischer Seite ist in Teilen bereits Stand der Technik. Dennoch fehlen an dieser Stelle durchgängige Ansätze über alle Ebenen. Die steuerungs- und softwaretechnischen Potentiale zur Selbstkonfiguration und Selbstorganisation wurden bislang unzureichend behandelt. Motiviert aus diesem Handlungsbedarf widmet sich diese Arbeit der Entwicklung eines modular ausgelegten Steuerungsmodells für Produktionsanlagen im Hinblick auf die materialflusstechnische Verbindung und Organisation mittels autonomer und dezentral agierender Systemeinheiten (Agenten).

Voraussetzung hierfür ist die modellhafte Erfassung aller Anlagenkomponenten, die an der Materialflussorganisation beteiligt sind. Grundlegend wurde hierzu ein Anlagenmodell entworfen, das von den konkreten Implementierungen der einzelnen Systemkomponenten abstrahiert und auf beliebige Anwendungen in der Produktion angewendet werden kann. Im ersten Schritt wurden diese Komponenten identifiziert und ihr Verhalten bzw. ihre Rolle im Verbund mit den anderen Anlagenkomponenten nach außen spezifiziert. Die Abstraktion erlaubt nach funktionalen Gesichtspunkten diese zu klassifizieren und ihr Verhalten gemäß ihrer Funktion zu vereinheitlichen. Dabei werden nicht nur physische Teile der Anlage, sondern auch organisatorische Einheiten, wie die Auftragseinlastung, als relevante Teilnehmer des Materialflusses angesehen. Wie die Funktion erfüllt wird, z.B. ob eine manuelle Tätigkeit damit verbunden ist oder voll automatisierte Stationen oder Transporteinheiten zur Verfügung stehen, ist an dieser Stelle irrelevant. Neben der Abstraktion wurde das Prinzip der Kapselung angewandt, in dem nicht nur die Implementierungsdetails einer Anlagenkomponente und die Umsetzung der geforderten Funktionalität, sondern auch Anlagenstrukturen gegenüber benachbarten Teilnehmern verborgen werden. Dies schafft die Voraussetzung für die modulare Integrität der Teilnehmer, die nur den lokalen Sichtbereich benötigen, der zur Erfüllung ihrer Aufgabenstellung erforderlich ist. Somit werden die Abhängigkeiten zwischen den Teilnehmern auf ein Mindestmaß reduziert. Bedingt durch standardisierte Softwareschnittstellen, die sich nicht nach der Ausprägung der Teilnehmer, sondern

nur an ihrem funktionalen Verhalten orientieren, bleibt die Menge dieser Koppelpunkte überschaubar und somit auch die dynamische und aufwandsarme Integrierbarkeit neuer Teilnehmer gegeben.

Nach der Identifizierung und Klassifizierung der Teilnehmer im Produktionssystem wurde das Modell um den Agentenansatz erweitert, der jeden Teilnehmer als eigenständige Einheit ansieht. Der Fokus lag hierbei insbesondere in der Modellierung eines überschaubaren und auf die Produktion ausgerichteten Agentensystems, das zudem mit einem hohen Maß an Wandelbarkeit ausgestattet ist. Mit diesem Ansatz kann eine Wandelbarkeit auf der organisatorischen Seite erzielt werden, indem bislang zentral hinterlegte und ausgeführte Regeln zur Materialflusskoordination in kleine Regeleinheiten zerlegt und lokal an den Agententeilnehmern zur Anwendung gebracht werden. Diese Methodik sieht vor, jedem Teilnehmer im Anlagenverbund einen Agenten zur Seite zu stellen, der im Auftrag die Handlungen des Teilnehmers koordiniert und Entscheidungen zur weiteren Vorgehensweise trifft. Somit kann jeder Agent selbständig Entscheidungen treffen und im Verbund mit anderen Teilnehmern interagieren, um eine gemeinsame Aufgabenstellung zu erfüllen. Bei der Spezifikation des Agentenmodells wurde ein hoher Wert auf die Festlegung des spezifischen Rollenverhaltens der verschiedenen Agentenklassen im Anlagenverbund gelegt. Dies ist maßgeblich für die flexible und dynamische Interaktion beliebiger Teilnehmer, die selbst zur Entwicklungszeit noch nicht feststehen, sondern erst zur Laufzeit Verbindungen eingehen.

Voraussetzung für die aufwandsarme Integrierbarkeit ist ein Steuerungssystem, das den Automatisierer bei der Umsetzung der agentenbasierten Teilnehmer unterstützt. Hierzu wurde eine Steuerungsarchitektur vorgestellt, die sich zum Ziel setzt, eine durchgängige Lösung zur Selbstkonfiguration und Selbstorganisation von Anlagensystemen zu entwickeln. Die Selbstkonfiguration vereinfacht den Aufbau von und Änderungen an Anlagensystemen durch Reduktion der manuellen Aufwände, unterstützt durch automatisierte Verfahren und Mechanismen zur Informationsgenerierung. Die Selbstorganisation unterstützt maßgeblich die Laufzeitsteuerung und reagiert unmittelbar auf Veränderungen bzw. Störungen im Produktionssystem. Mit der Steuerungsarchitektur werden dem Entwickler zunächst verschiedene Entwicklungswerkzeuge und vorgefertigte Bibliotheken zur Seite gestellt, die eine modulare Strukturierung von Anlagenkomponenten in Form eines hierarchischen Baumes erlauben und diese zugleich mit Standardschnittstellen, generiert aus funktional orientierten Schnittstellenspezifikationen, versehen werden können. Der selbst Rechengrenzen übergreifende Komponentenbaum, bietet administrative Dienstleistungen, wie Adressierung und Navigation, an und erlaubt einen ortstransparenten Zugriff auf alle Komponenten in der Anlage, unabhängig davon, auf welchem Rechner sie laufen. Diese Grundfunktionen werden zudem durch weiterführende Mechanismen erweitert, die es dem Entwickler erlauben, aufwandsarm neue Geräte, basierend auf bestehenden herstellerspezifischen Treibern, einzubinden und ihre Schnittstellen gemäß der Gerätefunktion zu normieren. Zudem steht ein neuartiger Strukturerkennungsalgorithmus zur Verfügung, der die eigentliche Wandelbarkeit zur Laufzeit unterstützt, indem dieser während der Laufzeit die genaue Anlagenstruktur und daraus weiterführende Daten, wie IP- und Materialfluss-Routentabellen, ableiten kann. Mit integrierten Mechanismen wird ebenso die automatisierte Konfiguration aller Teilnehmer mit Netzwerkparametern vorgesehen, wodurch Entwicklungsaufwände reduziert und die grundlegende Voraussetzung der unmittelbaren Funktions- und Kommunikationsbereitschaft nach Umbaumaßnahmen gewährleistet werden kann.

Materialflusssysteme bieten nur dann die erforderliche Flexibilität, wenn nicht nur die Mechanik

---

und Elektrotechnik darauf ausgelegt sind, sondern auch die Steuerungstechnik geeignete Verfahren zur Verfügung stellt, diese aufwandsarm umbauen und umorganisieren zu können. Defizite in der mangelnden Flexibilität im Materialfluss wurden sowohl bei den passiven, als auch aktiven Transportsystemen identifiziert. Motiviert aus den Defiziten bestehender Materialflusssteuerungstechniken, wurden in dieser Arbeit zwei Lösungen zur dynamischen Organisation und dezentralen Steuerung vorgestellt. Zum einen für reibschlussbasierte Fördersysteme, die mittels intelligenter Kreuzungen agieren, zum anderen für Fahrerlose Transportfahrzeuge, die unabhängig von zentralen Steuereinheiten agieren und direkt mit den anderen, verteilten Agententeilnehmern interagieren. Hierzu wurde der Ansatz für autonom navigierende Fahrerlose Transportfahrzeuge entwickelt, der einerseits den Aspekt der Eigenständigkeit in der agentenbasierten Organisation der Transportaufträge vorsieht und andererseits die flexible und dynamische Bahnplanung zur Laufzeit zum Ziel hat. Im Gegensatz zu bisherigen Lösungen Fahrerloser Transportsysteme, wird die Bahn dem Fahrzeug nicht vorgegeben, sondern das Fahrzeug plant sich die Bahn selbständig zur Laufzeit, gemäß der ihm vorgegebenen Umgebungskarte und der gesammelten Informationen zur Umgebung. Somit ist auch hier der Inbetriebnahmeaufwand der Fahrzeuge auf ein Minimum reduziert. Bei Einbringen eines neuen Fahrzeugs in eine ihm noch unbekannte Umgebung, steht es unmittelbar nach dem Einschalten funktionsbereit zur Verfügung und kann sofort Transportaufträge ausführen. Es kennt sich und seine Fähigkeiten und kann die erforderlichen Umgebungsinformationen per WLAN von anderen Fahrzeugen oder einem Kartenserver beziehen. Auch zur aufwandsarmen Kartographierung wurde eine Messlösung, basierend auf dem Positionsbestimmungssystem der Fahrzeuge, vorgestellt, die direkt Datenaktualisierungen an die Fahrzeuge übermitteln kann.

Aufbauend auf den vorgestellten Anlagenmodellen und den zwei ausgewählten Lösungen zur Umsetzung wandelbarer Materialflusssteuerungssysteme, wurden sowohl die Softwarekonzepte und Mechanismen zur Realisierung dieser als auch deren Implementierungen vorgestellt. Bei den Softwarekonzepten wurden Fragestellungen beantwortet, in welche Softwaremodule die Agententeilnehmer im Anlagenverbund intern unterteilt werden können und wie die detaillierten Interaktionen zwischen den Teilnehmern aussehen. Zudem wurde ein Verfahren aufgezeigt, wie die Produktherstellungsdaten strukturiert erfasst und abgelegt und mit den Anlagendaten aus der Strukturerkennung automatisiert zu den Arbeitsplänen vereint werden können. Somit können weitere manuelle Konfigurations- und Datenerfassungsschritte eingespart werden. Die Implementierung erfolgte in zwei Schritten in zwei unterschiedlichen Laboren. Im IT-Labor wurde zunächst die Funktionsfähigkeit der Steuerungsarchitektur und das Materialflussrouting für passive Transfersysteme an 23 verteilten Rechnern validiert. Im zweiten Labor, ausgestattet mit realen Anlagensystemen, wurde die agentenbasierte Steuerungsarchitektur an verschiedenen Produktionszellen umgesetzt und validiert. Zudem wurde der neuartige Steuerungsansatz für autonom navigierende Fahrerlose Transportfahrzeuge an einem eigens entwickelten und aufgebauten Demonstrationsfahrzeug implementiert und getestet. Das Fahrzeug integriert sich ebenfalls in den agentenbasierten Verbund autonom und dezentral agierender Teilnehmer. Es ist in der Lage selbständig die Bahnplanung zur Laufzeit basierend auf aktuellen Kartendaten vorzunehmen. Die Planung und Durchführung der Transportaufträge erfolgt zudem ohne zentralen Koordinierungsrechner, da das Fahrzeug direkt mit den Bahnhöfen um Transportaufträge verhandelt und mit ihnen die Werkstückübergabe koordiniert.

Mit der Vereinigung neuartiger steuerungstechnischer Mechanismen zur strukturellen Wandelbarkeit einerseits und dem Agentenansatz zur flexiblen und dynamischen Materialflussorganisation anderer-

seits, können die Vorzüge beider Technologien effektiv genutzt werden. Dabei bedingen sich die Selbstkonfiguration und die Selbstorganisation, integriert in der vorgestellten Steuerungsarchitektur, gegenseitig. Die Selbstkonfiguration bietet die Basis, aufwandsarm zur Laufzeit strukturelle Veränderungen vornehmen zu können. Diese funktionale Fähigkeit wurde gestützt von einem flexiblen Organisationssystem mit schneller Reaktionsfähigkeit, damit strukturelle Veränderungen im Produktionssystem unmittelbar von einer flexiblen Organisation aufgefangen und keine bzw. nur geringfügige Anpassungsaufwände in den Produktionsabläufen erforderlich sind. Umgekehrt bietet die Selbstorganisation mit den verteilt und autonom agierenden Agenten, die eigenständig Entscheidungen treffen, einen flexiblen Ansatz für die Produktionssteuerung, der mit heutigen Steuerungssystemen im vollen Funktions- und Leistungsumfang nicht ausschöpfbar ist. Klassische SPS-Steuerungssysteme erlauben nach heutigem Stand der Technik nur starre Verbindungen zwischen den Teilnehmern, die zur Entwicklungszeit bekannt sein müssen. Ausgehend von diesen Gründen, wurden im Rahmen dieser Arbeit die Mechanismen der Selbstkonfiguration und Selbstorganisation in gegenseitiger Abstimmung betrachtet.

## Summary

Changeableness is one of the significant key factors for producing companies in order to be competitive, which will gain more importance in future. Producers are more and more confronted with requirements for flexible organization of production systems and with challenges of dynamical and structural changes on the plant. Individual customer wishes, a high flexibility in products and variants and small lot sizes are the significant drivers, which claim an increasing reuse of plant components for economical reasons.

The support of structural changeableness must be guaranteed in the automation by reducing development efforts. The complexity of systems must be made controllable with the concept of modularization. The continuous modularization is needed in all fields of production engineering for the effective benefit of modular plant build-up. In the mechanical and energetic field modularization of production plants is state of the art. Nevertheless, at this point continuous approaches over all levels are missing. Potentials in the field of control systems and software technique for self-configuration and self-organization were regarded insufficiently up to now. Out of this motivation this present work focuses on the development of a modular control system for production plants with regard to the interconnection and organization of the material flow, based on the concept of autonomous and decentral operating system units (agents).

Precondition for this is the modelbased registration of all plant components, which are involved in the material flow organization. As a basis a plant model has been evolved, which abstracts from the concrete implementation of the single system components and which can be adopted to arbitrary applications in production. In the first step these components were identified and their behaviour and their role in collaboration with other plant components were specified. The abstraction allows to classify them based on functional issues and to unify them upon their function. Not only physical parts of a plant but also organizational units, like order dispatching, are regarded as members of the material flow. How the function will be provided, either manually or fully automated, is not important at this point. Beside of the abstraction the principle of encapsulation has applied, in which not only the implementation details of a plant component and the realization of the requested functionality, but also the plant structures are hidden towards neighbouring participants. This provides the precondition for the modular integrity of the participants, who only need the local view to fulfill their tasks. As a consequence the dependencies between the participants are reduced to a minimum. With standardized interfaces, which are not geared to the characteristics of the participants, but to their functional behaviour, the amount of coupling points remains manageable and thus the dynamical integrity of new participants is ensured with minimum effort.

After identification and classification of all members in the production system, the model was extended by the agent based approach, which regards every participant as an independent unit. Here the modeling of an easily understandable and production oriented agent based system was focused. This system has been equipped with a high degree of changeableness. With this approach change-

ableness can be obtained on the organizational side, while central deposited and executed rules for the coordination of the material flow are broken down in smaller rule units, being applied by the agent members. This method intends to attach an agent to every participant in the plant community. The agent coordinates the participant under his authority and makes decisions regarding further steps. As a consequence every agent can make decisions on his own and can interact with other agents to fulfill a common task. In the specification of the agent model, a high value was set on the determination of specific role behavior for different agent classes in the plant network. This is important for the flexible and dynamical interaction between arbitrary participants, that are still unknown during development time, but will be able to connect at run time.

Precondition for an integration with reduced effort is a control system that supports the automation engineer in the realization of agent-based participants. Therefore a control system architecture was presented with the goal to provide a continuous solution for self-configuration and self-organization of plant systems. The self-configuration offers an easier build-up of and even changes on plant systems, supported by automated mechanisms for automated information generation. The self-organization significantly supports the control at run-time and reacts immediately on changes and disturbances in production systems. With the control system architecture the developer is equipped with different development tools and predefined libraries. These allow to structure the plant components in a hierarchical tree and in parallel offer standardized interfaces, which can be generated out of functional interface specifications. The component tree offers administrative services, like addressing and navigation, and allows a transparent access to all components of a plant, independently of the computer they are running on. These base functions were extended by additional mechanisms, which allow developers to integrate new devices with reduced effort based on existing manufacturer specific drivers and to norm their interfaces. Beyond a new structure recognition algorithm is available, which supports the origin changeableness at run-time. During run-time the exact plant structure can be identified and based on this further data, like routing tables for IP protocol and material flow can be derived. Integrated mechanisms offer an automated configuration of all participants with network parameters, while development efforts can be reduced and the general precondition for the immediate functional and communicational readiness after reconstructions can be guaranteed.

Material flow systems only offer the required flexibility, if beside mechanic and electronic flexibility also the control technique provides the needed flexibility. In both passive and active transportation systems, their deficits in their flexibility were identified. Motivated out of the deficits of existing material flow techniques in the present work, two solutions for dynamical organization and decentral control are offered: on the one side for continuous conveying systems, which act with intelligent junctions - on the other side for autonomous guided vehicles, which act directly with other distributed agents independent of central control systems. Therefore the approach for autonomous navigating vehicles has been developed, which can react autonomously within the agent based organization and which provides a flexible and dynamic path planning during run time. In contradiction to existing solutions of autonomous guided vehicles no predetermined path to the vehicle exists, but the vehicle itself can determine the path at run-time. This is done based on an existing map and based on collected information about the environment. Therefore the effort for commissioning of the vehicles can be reduced to a minimum. When introducing a new vehicle into an unknown environment, immediately after switching on, it is operable and can execute transportation tasks. The vehicle knows itself and its abilities and can retrieve the needed environment information via WLAN from other vehicles or from

---

a map server. For reducing efforts at cartography tasks a solution for measuring, based on existing position determination systems for vehicles, has been presented. Further it offers data integrity with the vehicles and therefore can transmit changes on the map directly to them.

Based on the presented plant models and the two selected solutions for the realization of changeable material flow systems, the appropriate software concepts and their implementations were presented. Within the software concepts questions were answered, as in which software modules the agent members can be divided and how the detailed interactions between the members look. Beyond a mechanism was presented how the product manufacturing data can be stored in a structured way and can be combined with the plant data, automatically retrieved from the structure recognition, in order to derive the working plans. Thus additional manual configuration and data acquisition steps can be saved. The implementation took place in two steps in two different laboratories. At first the functionality of the control architecture and of the material flow routing for passive transfer systems was validated in the IT-Laboratory by using 23 distributed computers. In the second laboratory equipped with real plant systems, the agent based control architecture was realized and validated on different production cells. In addition the new control approach for autonomous navigating vehicles was applied and validated on a vehicle constructed for demonstration purposes. The vehicle integrates itself in the agent-based network of autonomous and decentralized acting participants. It has the ability to calculate the path at run-time autonomously based on actual map data. The planning and execution of the transportation tasks can be accomplished without any central coordination computer, as the vehicle directly negotiates with the stations for transportation tasks and even coordinates with them the handover of the workpieces.

By the unification of new control techniques for the structural changeableness on the one side and the agent approach for the flexible and dynamic organization of material flows on the other side, the benefits of both can be used effectively. In this way self-configuration and self-organization, ingrained in the presented control architecture, require one another. The self-configuration offers the basis to make structural changes at run-time. This functional feature was supported by a flexible organization system with fast reaction time, so that structural changes in the production system can immediately be compensated by a flexible organization or only need minimum manual adoptions. In return the self-organization with the distributed and autonomous acting agents, a flexible approach for the production control is being offered, which potentials cannot be used effectively with the full functionality and yield. Based on the present state of the art, classic PLC control systems only allow fixed interconnection between the participants, which need to be known at run-time. Motivated out of these reasons, the mechanisms for mutual self-configuration and self-organization were regarded in the present work.

## Literaturverzeichnis

- [1] SCHULZ-WOLFGGRAMM, C.: *Neues Denken und Handeln für Innovation und Restrukturierung*. In: REINHART, G. und H. HOFFMANN (Herausgeber): *Nur der Wandel bleibt. Wege jenseits der Flexibilität*, Seiten 41–58, München, 2000. Münchener Kolloquium.
- [2] DÜRRSCHMIDT, S.: *Planung und Betrieb wandlungsfähiger Logistiksysteme in der variantenreichen Serienproduktion*. Herbert Utz Verlag, München, 2001.
- [3] KIRCHNER, S., R. WINKLER und E. WESTKÄMPER: *Unternehmensstudie zur Wandlungsfähigkeit von Unternehmen*. In: *wt Werkstatttechnik online*, Jahrgang 93 2003.
- [4] MORALES, R. H.: *Systematik der Wandlungsfähigkeit in der Fabrikplanung*. VDI Verlag, Düsseldorf, 2003.
- [5] SCHENK, M. und S. WIRTH: *Fabrikplanung und Fabrikbetrieb*. Springer Verlag, 2004.
- [6] WIENDAHL, H.-P., D. NOFEN, J. H. KLUSMANN und F. BREITENBACH: *Planung modularer Fabriken*. Carl Hanser Verlag, 2005.
- [7] HILDEBRAND, T.: *Theoretische Grundlagen der bausteinbasierten, technischen Gestaltung wandlungsfähiger Fabrikstrukturen nach dem PLUG+PRODUCE Prinzip*. IBF, Chemnitz, 2005.
- [8] BÄR, M. und S. WIRTH: *Flexible Fertigungssysteme – Gestaltung und Anwendung in der Teilefertigung*. Hüthig Buch Verlag, Heidelberg, 1989.
- [9] DANGLMAIER, W.: *Fertigungsplanung. Planung von Aufbau und Ablauf der Fertigung - Grundlagen, Algorithmen und Beispiele*. Springer Verlag, 1999.
- [10] GOTTSCHALK, E. (Herausgeber): *Rechnergestützte Produktionsplanung und -steuerung*. Technik Verlag, 1989.
- [11] HACKSTEIN, R.: *Produktionsplanung und -steuerung (PPS) - Ein Handbuch für die Betriebspraxis*. Springer Verlag, 1989.
- [12] VEREIN DEUTSCHER WERKZEUGMASCHINENFABRIKEN (VDW): *Werkzeugmaschinenproduktion 2003*. VDW, Frankfurt, 2004.
- [13] VEREIN DEUTSCHER WERKZEUGMASCHINENFABRIKEN (VDW): *Deutsche Werkzeugmaschinenindustrie, Daten und Fakten 2003*. VDW, Frankfurt, 2004.
- [14] VEREIN DEUTSCHER INGENIEURE (VDI): *Flexible Fertigung (FFS). Leitfaden des VDI-Gemeinschaftsausschusses CIM*. VDI-Verlag, Düsseldorf, 1990.



- 
- [15] DANGELMAIER, W.: *Fertigungsplanung. Planung von Aufbau und Ablauf der Fertigung*. Springer, Berlin, 2001.
- [16] WECK, M.: *Werkzeugmaschinen. Band 1*. Springer, Berlin, 1997.
- [17] Fastems. <http://www.fastems.com>, Stand: 01.03.2008.
- [18] BRENNAN, R.W. und D.H. NORRIE: *From FMS to HMS*. In: DEEN, S.M. (Herausgeber): *Agent-Based Manufacturing. Advances in the Holonic Approach*, Seiten 31–52, Berlin, 2003. Springer.
- [19] HOFMANN, G.: *Flexible Fertigungssysteme von heute*. In: *Werkstatt und Betrieb 133*, Seiten 46–49, München, 2000. Hanser.
- [20] LAY, G.: *Sackgasse Hochautomatisierung? Mitteilungen aus der Produktionsinnovationserhebung*. Fraunhofer Institut Systemtechnik und Innovationsforschung, Karlsruhe, 2001.
- [21] MEHRABI, M.G., A.G. ULISOY und Y. KOREN: *Reconfigurable Manufacturing Systems: Key to Future Manufacturing*. In: *International Journal of Intelligent Manufacturing*, Vol. 11, No. 4, Seiten 403–419, August 2000.
- [22] STAVE, H.: *Flexible Produktionstechnik für die Großserie - heute und morgen*. In: KLOCKE, F. (Herausgeber): *Autonome Produktion*, Seiten 567–578, Berlin, 2003. Springer.
- [23] ZEICHEN, G., B. FAVRE-BULLE und W. STÖCHER: *Europäische Produktion am Wendepunkt*. wt Werkstatttechnik online, 96, 2006.
- [24] BRINZER, B., J. PRIESE, P. KLEMM und G. PRITSCHOW: *Neue Ansätze zur dezentralen Produktionsregelung*. In: wt Werkstatttechnik online, Jahrgang 95 2005.
- [25] PRITSCHOW, G., G. SPUR, M. WECK und A. PAULS: *Leit- und Steuerungstechnik in flexiblen Produktionsanlagen*. Carl Hanser Verlag, 1991.
- [26] JÜNEMANN, R. und A. BEYER: *Steuerung von Materialfluss- und Logistiksystemen*. Springer, Berlin, 1998.
- [27] FELDMANN ET. AL. (HRSG.): *Montage strategisch ausrichten - Praxisbeispiele marktorientierter Prozesse und Strukturen*. Springer, Berlin, 2004.
- [28] SCHMITT, M.: *Ein Leitstand zur Einsatzplanung und Überwachung mobiler Roboter*. In: G. SCHMIDT, ET AL. (Herausgeber): *Autonome Mobile Systeme 1999*, Berlin, 1999. Springer.
- [29] *Open Transportation Control System (OpenTCS)*. <http://www.opentcs.de>, Stand: 12.09.2005.
- [30] WALLACE, A.: *Sequential Resource Allocation Utilizing Agents*. In: *International Journal of Production Research*, Band 47, Seiten 2481–2499, 2003.
- [31] VEREIN DEUTSCHER INGENIEURE (VDI): *VDE-VDI-Richtlinien. Kompatibilität von Fahrerlosen Transportsystemen (FTS); Blatt 6: Sensorik für Navigation und Steuerung*. VDI, Düsseldorf, 2003.

- [32] SCHRECKER, A.: *Planung und Steuerung fahrerloser Transportsysteme : Ansätze zur Unterstützung der Systemgestaltung*. Doktorarbeit, Dt. Univ.-Verl., Wiesbaden, 2000.
- [33] WOOLDRIDGE, M. und N. JENNINGS: *Intelligent Agents*. Springer-Verlag, Amsterdam, 1994.
- [34] RITTER, A. und C. SCHAEFFER: *Agenten zur Steuerung flexibler Fertigungssysteme*. Seiten 333–337. *wt Werkstatttechnik*, 2001.
- [35] BUSSMANN, S., N. JENNINGS und M. WOOLDRIDGE: *Multiagent Systems for Manufacturing Control*. Springer, Berlin, 2004.
- [36] WAGNER, T., P. GÖHNER und P.G. URBANO: *Softwareagenten - Einführung und Überblick über eine alternative Art der Softwareentwicklung. Teil I*. In: *atp - Automatisierungstechnische Praxis* 45, Heft 10, 2003.
- [37] AL., LUCK M. ET: *Agent Technology Roadmap: A Roadmap for Agent Based Computing*. In: *Agentlink III*, <http://www.agentlink.org/roadmap/al3rm.pdf>, Oktober 2006.
- [38] CSÁJI, B.CS., L. MONOSTORI und B. KÁDÁR: *Learning and Cooperation in a Distributed Market-Based Production Control System*. In: *Proc. 5th International Workshop on Emergent Synthesis, IWES 2004*, Seiten 109–117, Budapest, Hungary, May 25-25 2004.
- [39] CSÁJI, B.CS., L. MONOSTORI und B. KÁDÁR: *Learning and Cooperation in a Distributed Market-Based Production Control System*. In: *Proc. 5th International Workshop on Emergent Synthesis, IWES 2004*, Seiten 109–117, Budapest, Hungary, May 24-25 2004.
- [40] *The Foundation for Intelligent Physical Agents*. <http://www.fipa.org>, Stand: 12.09.2005.
- [41] RITTER, A.: *Ein Multi-Agenten-System für mobile Einrichtungen in Produktionssystemen*. Jost Jetter Verlag, Heimsheim, 2003.
- [42] FERBER, J.: *Multiagentensysteme. Eine Einführung in die künstliche Intelligenz*. Addison-Wesley, München, 2001.
- [43] JENNINGS, N.R.: *An Agent-Based Approach for Building Complex Software Systems*. In: *Communications of the ACM. Volume 44*, Seiten 35–41, New York, USA, April 2001. ACM.
- [44] WOOLDRIDGE, M.: *An Introduction to Multiagent Systems*. Addison-Wesley, Reading, MA, 2000.
- [45] BORDINI, R.H., M. DASTANI, J. DIX, A. EL FALLAH SEGHRUCHINI und (EDS.): *Multi-Agent Programming: Languages, Platforms and Applications*. Springer, Boston, MA, 2005.
- [46] MICHEL, F., H. VAN DYKE PARUNAK und D. WEYNS: *Environments for Multi-Agent Systems*. Springer, Berlin, 2005.
- [47] MÜLLER, J.: *Verteilte künstliche Intelligenz*. Wissenschafts-Verlag, Mannheim, 1993.
- [48] PRIESE, L.: *Petri-Netze*. Springer, Berlin, 2008.

- [49] POSSEL-DÖLKEN, F.: *Hybride verteilte Steuerungs- und Leitsysteme*. In: *Seminar Herausforderung Fertigungssteuerung*, Aachen, 2003. ADITEC gGmbH.
- [50] BONGAERTS, L.: *Integration of Scheduling and Control in Holonic Manufacturing Systems*. Diss., Katholieke University Leuven, Belgium, 1998.
- [51] DENNETT, D.C.: *The Intentional Stance*. MIT Press, Cambridge (Massachusetts), 1987.
- [52] GRIFFEL, F.: *Componentware – Konzepte und Techniken eines Softwareparadigmas*. dpunkt Verlag, Heidelberg, 1998.
- [53] SZYBERSKI, C.: *Component Software. Beyond Object-Oriented Programming*. ACM Press, New York, 2002.
- [54] COUNCILL, W. T. und G. T. HEINEMAN: *Component-Based Software Engineering*. Addison-Wesley, New York, 2001.
- [55] MCILROY, M. D.: *Mass Produced Software Components*. In: *Software Engineering, Report on a conference sponsored by the NATO Science Committee*, 1968.
- [56] BRITTON, C. und P. BYE: *IT architectures and middleware : strategies for building large, integrated systems*. Addison-Wesley, München, 2007.
- [57] BERNSTEIN, P. A.: *Middleware - An Architecture for Distributed System Services*. Technischer Bericht CRL 93/6, Digital Equipment Corporation, 1993.
- [58] AMARA, H., P. DÉPINCÉ und J-Y. HASCOET: *Interactive CAPP System in a Distributed Environment*. In: *Proc. of the Integrated Design and Manufacturing, Mechanical Engineering (IDMME)*, Clermont-Ferrand, France, Mai 2002.
- [59] CUTKOSKY, M., R. ENGLEMORE, R. FIKES, M.R. GENESERETH, T.R. GRUBER, W.S. MARK, J.M. TENENBAUM und J.C. WEBER: *PACT: An Experiment in Integrating Concurrent Engineering Systems*. In: *IEEE Computer*, Band 26, Seiten 28–37, Januar 1993.
- [60] KAMBHAMPATI, S., M.R. CUTKOSKY, J.M. TENENBAUM und S.H. LEE: *Integrating General Purpose Planners and Specialized Reasoners: Case Study of a Hybrid Planning Architecture*. In: *IEEE Trans. on Systems, Man and Cybernetics*, Band 23, Seiten 1503–1518, 1993.
- [61] JIN, Y. und S.C.-Y. LU: *Agent-Based Negotiation for Collaborative Design Decision Making*. In: *Annals of the CIRP*, Band 53, Seiten 121–124, 2004.
- [62] BRUCCOLERI, M., G. LO NIGRO, G. PERRONE, P. RENNA und S. NOTO LA DIEGA: *Production Planning in Reconfigurable Enterprises and Reconfigurable Production Systems*. In: *Annals of the CIRP*, Band 54, Seiten 433–436, 2005.
- [63] MAROPOULOS, P.G., K.R. MCKAY und D.G. BRAMALL: *Resource-Aware Aggregate Planning for the Distributed Manufacturing Enterprise*. In: *Annals of the CIRP*, Band 51, Seiten 363–366, 2002.

- [64] LEE, Y., S. KUMARA und K. CHATTERJEE: *Multiagent-based Dynamic Resource Scheduling for Distributed Multiple Projects Using a Market Mechanism*. In: *Journal of Intelligent Manufacturing*, Band 14, Seiten 471–484, 2003.
- [65] BUSSMANN, S. und K. SCHILD: *Self-Organizing Manufacturing Control: An Industrial Application of Agent Technology*. In: DURFEE, E. (Herausgeber): *Proceedings of the 4th International Conference on Multi-Agent Systems*, Amsterdam, 2002.
- [66] SCHOOP, R., R. NEUBERT und A.W. COLOMBO: *A Multiagent-based Distributed Control Platform for Industrial Flexible Production Systems*. In: *Proceedings of IECON'01. IEEE Electronics Society*, Band 1, Seiten 279–284, Denver, 2001.
- [67] ANUSSORNNITISAN, P., J. PERALTA und S. NOF: *Time-out Protocol for Task Allocation in Multiagent Systems*. In: *Journal of Intelligent Manufacturing*, Band 13, Seiten 511–522, 2002.
- [68] TIETZE, E. und R. SCHOOP: *Fertigungs-Broker*. In: *IEE*, Band 51, Seiten 38–42. Hüthig GmbH Verlag, 2000.
- [69] COLOMBO, A.W. und R. SCHOOP: *A Solution to Holonic Control Systems*. In: *Proceedings of the IEEE-ETFA*, Antibes, 2001. IEEE Electronics Society.
- [70] LANGER, G.: *A Methodology for Holonic Multi-Cell Control Systems*. Technical University of Denmark, 1999.
- [71] KLEMM, E. und A. LÜDER: *Agentenbasierte Flexibilisierung der Produktion bei Verwendung von vorhandenen Steuerungssystemen*. In: *atp - Automatisierungstechnische Praxis* 45, Heft 10, Seiten 66–75, 2003.
- [72] KONSORTIUM, PABADIS: *Definition of Overall PABADIS PROMISE Control System Architecture*. Pabadis Konsortium, Magdeburg, 2006.
- [73] JACKE, B., O. SAUER und G. SUTSCHET: *ProVis.Agent: An agent-based production monitoring and control system*. In: *Software-Agents in Information Systems and Industrial Applications*, SAISIA, Seiten 75–82, Aachen, 2006. Fraunhofer IRB Verlag.
- [74] SAUER, O. und G. SUTSCHET: *Provis.Agent. Ein agentenorientiertes Produktionsleitsystem - erste Erfahrungen im industriellen Einsatz*. In: *VDE-Kongress*, Seiten 297–302, Aachen, 2006.
- [75] HOMPEL, M. TEN: *Revolution im Materialfluss*. In: *mundo*, Seiten 54–59, Dortmund, 2004. Universität Dortmund.
- [76] BULLINGER, H.-J. und M. TEN HOMPEL: *Multiagentensysteme im Internet der Dinge - Konzepte und Realisierung*. Springer, Berlin, 2007.
- [77] BAUM, W., A. BREDENFELD, M. HANS, J. HERTZBERG und A. RITTER ET. AL.: *Integrating Heterogeneous Robot and Software Components by Agent Technology*. In: *Robotik 2002: Leistungsstand - Anwendungen - Visionen - Trends*, Düsseldorf, 2003. VDI Verlag.
- [78] VOTH, D.: *Holonics in Manufacturing: Bringing Intelligence Closer to the Machine*. In: *IEEE Intelligent Systems*, Band 19, Seiten 4–7, 2004.

- 
- [79] KAWAMURA, K.: *Holonic Manufacturing Systems: An Overview and Key Technical Issues*. In: *4th IFAC Workshop on Intelligent Manufacturing Systems: IMS'97*, Seiten 33–36, Seoul, Korea, 1997.
- [80] *FIPA Agent Software Integration Specification (FIPA00079)*. <http://www.fipa.org/specs/fipa00079/>, Stand: 2005.
- [81] MASAK, D.: *Legacysoftware*. Springer, Berlin, 2006.
- [82] CHARLTON, P., R. CATTONI, A. POTRICH und E. MAMDANI: *Evaluating the FIPA Standards and Its Role in Achieving Cooperation in Multiagent Systems*. In: *Proceedings of the 33rd Hawaii International Conference on System Sciences, IEEE*, Seiten 8034–8043, Island of Maui, Januar 2000. IEEE Computer Society.
- [83] DASTANI, M., J. HULSTIJN und L. VAN DER TORRE: *How to Decide What to Do?* In: *European Journal of Operational Research*, Band 160, Seiten 762–784, Amsterdam, 2005.
- [84] RAO, A.S. und M. GEORGEFF: *BDI Agents : From Theory to Practice*. In: *Proceedings of First International Conference on Multi-Agent Systems (ICMAS)*, Band 1, Seiten 312–319, San Francisco, California, 1995.
- [85] *FIPA Agent Software Integration Specification (FIPA00089)*. <http://www.fipa.org/specs/fipa00089/>, Stand: 2005.
- [86] FELDMANN, K., M. WEBER und W. WOLF: *Decentralized Structure Recognition and Automated Network Configuration for Plug&Produce-able Modular Assembly Systems*. *Production Engineering*, XIII(2), 2006.
- [87] FELDMANN, K., W. WOLF und M. WEBER: *Development of an Open, Event-Based and Platform Independent Architecture for Distributed and Intelligent Control Systems*. *IEEE International Conference on Industrial Informatics (INDIN)*, 2004.
- [88] FASTEMS: *Flexibles Palettenmagazin*. Helvar Mercar Group, Tampere, 2005.
- [89] SESSELER, R.: *Eine modulare Architektur für dienstbasierte Interaktion zwischen Agenten*. TU Berlin, 2002.
- [90] FELDMANN, K., W. WOLF und M. WEBER: *Design of a Formal Model for the Specification of Agent Platforms Based on Plug&Produce-able Production Systems*. *Production Engineering*, 1.
- [91] DAGANZO, C. F.: *Logistics Systems Analysis*. Springer, Berlin, 2005.
- [92] GERBER, C., J. SIEKMANN und G. VIERKE: *Fahrerlose Transportsysteme (FTS)*. In: *VDI-Handbuch Materialfluß und Fördertechnik, Band 2, VDI 2510*, Düsseldorf, 2005. VDI.
- [93] FELDMANN, K., W. WOLF und M. WEBER: *Self Organizing Material Flow for Plug&Produce-able Modular Assembly Systems*. *Production Engineering*, XIII(2), 2006.

- [94] WEISS, G.: *Multiagent systems: a modern approach to distributed artificial intelligence*. The MIT Press, Cambridge, Massachusetts, 2000.
- [95] FELDMANN, K., M. WEBER und W. WOLF: *Performance Analysis of Distributed Strategies for Material Flow Control of Modular Production Systems Based on Discrete Event Simulation*. IEEE International Conference on Industrial Informatics (INDIN), 2004.
- [96] MAHALIK, N. P.: *Fieldbus Technology*. Springer, Berlin, 2007.
- [97] LUHMANN, N.: *Einführung in die Systemtheorie*. Carl-Auer-Systeme-Verlag, Heidelberg, 2008.
- [98] WEBER, M.: *Unterstützung der Wandlungsfähigkeit von Produktionsanlagen durch innovative Softwaresysteme*. Meisenbach Verlag, Bamberg, 2007.
- [99] SCHULTE, H.: *RFC's und Internetstandards*. Interest Verlag, Augsburg, 2004.
- [100] TANNENBAUM, A. S.: *Computerarchitektur. Strukturen Konzepte Grundlagen*. Pearson Studium, München, 2004.
- [101] PYTHON SOFTWARE FOUNDATION. <http://www.python.org>, Stand: 18.05.2007.
- [102] MARTELLI, A. und D. ASCHER: *Python Cookbook*. O'Reilly, Beijing, 2005.
- [103] PLÖTNER, J. und S. WENZEL: *Linux - das distributionsunabhängige Handbuch*. Galileo Press, Bonn, 2006.
- [104] PETRAHN, G.: *Grundlagen der Vermessungstechnik*. Cornelsen, Berlin, 2007.
- [105] VEREIN DEUTSCHER INGENIEURE (VDI): *Kompatibilität von Fahrerlosen Transportsystemen (FTS) - Sensorik für Navigation und Steuerung*. VDI, Düsseldorf, 2003.
- [106] KERN, C.: *Anwendung von RFID-Systemen*. Springer, Berlin, 2006.
- [107] DIERKS, F.: *Freie Navigation autonomer Fahrzeuge*. Fortschritt-Berichte / VDI : Reihe 12, Verkehrstechnik, Fahrzeugtechnik ; Nr. 291, Düsseldorf, 1996.
- [108] INTERNATIONAL TELECOMMUNICATION UNION: *ITU-T Recommendation X.667*. <http://www.itu.int/ITU-T/studygroups/com17/oid/X.667-E.pdf>, Stand: 18.05.2007.
- [109] LATOMBE, J-C.: *Robot Motion Planning*. Kluwer Academic Publishers, Massachusetts, 2004.
- [110] KRASCHL, G.: *Routenplanung für mobile Roboter*. Satellitengeodäsie der Technischen Universität Graz, Graz, 2005.
- [111] JUNGnickel, D.: *Graphen, Netzwerke und Algorithmen*. Wissenschaftsverlag, Mannheim, 1987.
- [112] BÖHM, R.: *Bahnplanung in der Robotik*. <http://www.inf.hs-zigr.de/boehm/seite26.htm>, Stand: 15.01.2007.

- [113] ZELL, A. und A. MOJAEV: *Aufbau topologischer Karten und schnelle globale Bahnplanung für mobile Roboter*. Technischer Bericht, <http://www-ra.informatik.uni-tuebingen.de/mitarb/mojaev/documents/ams01tp.ps.gz>, 2001.
- [114] HESSELER, M. und M. GÖRTZ: *Basiswissen ERP-Systeme : Auswahl, Einführung und Einsatz betriebswirtschaftlicher Standardsoftware*. W3L-Verlag, Witten, 2007.
- [115] MARIHART, O. und H. PEIRLBERGER: *Betriebsdatenerfassung: ein Instrument für mehr Transparenz im Unternehmen*. WIFI der Bundeskammer der Gewerblichen Wirtschaft, Wien, 1990.
- [116] STAUD, J. L.: *Datenmodellierung und Datenbankentwurf: ein Vergleich aktueller Methoden*. Springer, Berlin, 2005.
- [117] ZIMMERMANN, M., M. WEBER, W. WOLF, S. JUNKER und A. MENSCHIG: *Kleinst-Produktionslinie zur Fertigung hochpräziser faseroptischer Mikroteile*. In: M. GEIGER, S. POLSTER (Herausgeber): *Tagungsband Laser in der Elektronikproduktion und Feinwerktechnik - LEF 2006*, Bamberg, 2006. Meisenbach-Verlag.
- [118] HOMBURG, D. und E.-C. REIFF: *Weg- und Winkelmessung : absolute Messverfahren*. In: *Mosaik der Automatisierung*, Band 1, Stutensee, 2003. PKS Presse Kontakte Stutensee Homburg Verlag.
- [119] WOLF, W.: *Flexible, adaptierbare Produktionssysteme durch agentenbasierten Materialfluss und kooperierende Industrieroboter*. In: *Internationales Forum Mechatronik*, Winterthur, Schweiz, September 2007.
- [120] EL-RABBANY, A.: *Introduction to GPS*. Artech House, Norwood, Mass., USA, 2006.
- [121] *Orientation Computer with Serial Communication - Specification of the Interface Protokoll*. Fraunhofer Institut Materialfluss und Logistik, Dortmund, 2004.
- [122] *Serviceunterlagen Laser-Orientierungsrechner*. Fraunhofer Institut Materialfluss und Logistik, Dortmund, 2006.
- [123] MATTHEW, N. und R. STONES: *Linux-Programmierung*. mitp-Verlag, Bonn, 2005.
- [124] DÜDING, D.: *Ein Beitrag zum Einsatz von echtzeitfähigen Linux-Varianten in der Automatisierungstechnik*. Universität Dortmund, Dortmund, 2003.





## Lebenslauf

## Persönliche Daten

Name: Wolfgang Robert Wolf  
Geburtsdatum: 26.09.1976  
Geburtsort: Nürnberg  
Staatsangehörigkeit: deutsch  
Familienstand: verheiratet

## Ausbildung

02/2002 - 06/2002	Auslandsaufenthalt an der University of Bath in England
10/1996 - 10/2002	Informatik-Studium an der Friedrich-Alexander-Universität Erlangen-Nürnberg Abschluss: Dipl.-Inf. (Univ.)
09/1987 - 05/1996	Gymnasium Stein Abschluss: Abitur

## Berufspraxis

Seit 06/2008	<b>Teilprojektleiter</b> Teilprojekt Digitale Fabrik im Projekt Automation Designer mit dem Schwerpunkt der Anbindung der SIEMENS PLM Produkte Teamcenter und eM-Server TechConnect GmbH, München bei SIEMENS AG, Fürth
01/2003 - 04/2008	<b>Wissenschaftlicher Mitarbeiter</b> Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik der Friedrich-Alexander Universität Erlangen-Nürnberg Leiter: Prof. Dr.-Ing. K. Feldmann
09/1992 - 12/2002	<b>Softwareentwickler und Netzwerk-Administrator</b> Software Systeme Wolf GmbH, Roßtal



# Reihe Fertigungstechnik - Erlangen

[www.mb.uni-erlangen.de/diss](http://www.mb.uni-erlangen.de/diss)

**Band 1 - 52**  
**Carl Hanser Verlag, München**

**ab Band 53**  
**Meisenbach Verlag, Bamberg**  
**45,-- Euro**

Band 1: Andreas Hemberger  
**Innovationspotentiale in der rechnerintegrierten Produktion  
durch wissensbasierte Systeme**  
208 Seiten, 107 Bilder. 1988.

Band 2: Detlef Classe  
**Beitrag zur Steigerung der Flexibilität automatisierter Montagesysteme  
durch Sensorintegration und erweiterte Steuerungskonzepte**  
194 Seiten, 70 Bilder. 1988.

Band 3: Friedrich-Wilhelm Nolting  
**Projektion von Montagesystemen**  
201 Seiten, 107 Bilder, 1 Tabelle. 1989.

Band 4: Karsten Schlüter  
**Nutzungsgradsteigerung von Montagesystemen  
durch den Einsatz der Simulationstechnik**  
177 Seiten, 97 Bilder. 1989.

Band 5: Shir-Kuan Lin  
**Aufbau von Modellen zur Lageregelung von Industrierobotern**  
168 Seiten, 46 Bilder. 1989.

Band 6: Rudolf Nuss  
**Untersuchungen zur Bearbeitungsqualität  
im Fertigungssystem Laserstrahlschneiden**  
206 Seiten, 115 Bilder, 6 Tabellen. 1989.

Band 7: Wolfgang Scholz  
**Modell zur datenbankgestützten Planung  
automatisierter Montageanlagen**  
194 Seiten, 89 Bilder. 1989.

Band 8: Hans-Jürgen Wißmeier  
**Beitrag zur Beurteilung des Bruchverhaltens  
von Hartmetall-Fließpreßmatrizen**  
179 Seiten, 99 Bilder, 9 Tabellen. 1989.

Band 9: Rainer Eisele  
**Konzeption und Wirtschaftlichkeit  
von Planungssystemen in der Produktion**  
183 Seiten, 86 Bilder. 1990.

Band 10: Rolf Pfeiffer  
**Technologisch orientierte Montageplanung  
am Beispiel der Schraubtechnik**  
216 Seiten, 102 Bilder, 16 Tabellen. 1990.

Band 11: Herbert Fischer  
**Verteilte Planungssysteme zur Flexibilitätssteigerung  
der rechnerintegrierten Teilefertigung**  
201 Seiten, 82 Bilder. 1990.

- Band 12: Gerhard Kleineidam  
**CAD/CAP: Rechnergestützte Montagefeinplanung**  
 203 Seiten, 107 Bilder. 1990.
- Band 13: Frank Vollertsen  
**Pulvermetallurgische Verarbeitung eines übereutektoiden verschleißfesten Stahls**  
 XIII u. 217 Seiten, 67 Bilder, 34 Tabellen. 1990.
- Band 14: Stephan Biermann  
**Untersuchungen zur Anlagen- und Prozeßdiagnostik für das Schneiden mit CO<sub>2</sub>-Hochleistungslasern**  
 VIII u. 170 Seiten, 93 Bilder, 4 Tabellen. 1991.
- Band 15: Uwe Geißler  
**Material- und Datenfluß in einer flexiblen Blechbearbeitungszelle**  
 124 Seiten, 41 Bilder, 7 Tabellen. 1991.
- Band 16: Frank Oswald Hake  
**Entwicklung eines rechnergestützten Diagnosesystems für automatisierte Montagezellen**  
 XIV u. 166 Seiten, 77 Bilder. 1991.
- Band 17: Herbert Reichel  
**Optimierung der Werkzeugbereitstellung durch rechnergestützte Arbeitsfolgenbestimmung**  
 198 Seiten, 73 Bilder, 2 Tabellen. 1991.
- Band 18: Josef Scheller  
**Modellierung und Einsatz von Softwaresystemen für rechnergeführte Montagezellen**  
 198 Seiten, 65 Bilder. 1991.
- Band 19: Arnold vom Ende  
**Untersuchungen zum Biegeumformen mit elastischer Matrice**  
 166 Seiten, 55 Bilder, 13 Tabellen. 1991.
- Band 20: Joachim Schmid  
**Beitrag zum automatisierten Bearbeiten von Keramikguß mit Industrierobotern**  
 XIV u. 176 Seiten, 111 Bilder, 6 Tabellen. 1991.
- Band 21: Egon Sommer  
**Multiprozessorsteuerung für kooperierende Industrieroboter in Montagezellen**  
 188 Seiten, 102 Bilder. 1991.
- Band 22: Georg Geyer  
**Entwicklung problemspezifischer Verfahrensketten in der Montage**  
 192 Seiten, 112 Bilder. 1991.
- Band 23: Rainer Flohr  
**Beitrag zur optimalen Verbindungstechnik in der Oberflächenmontage (SMT)**  
 186 Seiten, 79 Bilder. 1991.
- Band 24: Alfons Rief  
**Untersuchungen zur Verfahrensfolge Laserstrahlschneiden und -schweißen in der Rohkarosseriefertigung**  
 VI u. 145 Seiten, 58 Bilder, 5 Tabellen. 1991.
- Band 25: Christoph Thim  
**Rechnerunterstützte Optimierung von Materialflußstrukturen in der Elektronikmontage durch Simulation**  
 188 Seiten, 74 Bilder. 1992.
- Band 26: Roland Müller  
**CO<sub>2</sub>-Laserstrahlschneiden von kurzglasverstärkten Verbundwerkstoffen**  
 141 Seiten, 107 Bilder, 4 Tabellen. 1992.
- Band 27: Günther Schäfer  
**Integrierte Informationsverarbeitung bei der Montageplanung**  
 195 Seiten, 76 Bilder. 1992.

Band 28: Martin Hoffmann  
**Entwicklung einer CAD/CAM-Prozeßkette  
für die Herstellung von Blechbiegeteilen**  
149 Seiten, 89 Bilder. 1992.

Band 29: Peter Hoffmann  
**Verfahrensfolge Laserstrahlschneiden und –schweißen :  
Prozeßführung und Systemtechnik in der 3D–Laserstrahlbearbeitung  
von Blechformteilen**  
186 Seiten, 92 Bilder, 10 Tabellen. 1992.

Band 30: Olaf Schrödel  
**Flexible Werkstattsteuerung mit objektorientierten Softwarestrukturen**  
180 Seiten, 84 Bilder. 1992.

Band 31: Hubert Reinisch  
**Planungs- und Steuerungswerkzeuge  
zur impliziten Geräteprogrammierung in Roboterzellen**  
XI u. 212 Seiten, 112 Bilder. 1992.

Band 32: Brigitte Bärnreuther  
**Ein Beitrag zur Bewertung des Kommunikationsverhaltens  
von Automatisierungsgeräten in flexiblen Produktionszellen**  
XI u. 179 Seiten, 71 Bilder. 1992.

Band 33: Joachim Hutfless  
**Laserstrahlregelung und Optikdiagnostik  
in der Strahlführung einer CO<sub>2</sub>-Hochleistungslaseranlage**  
175 Seiten, 70 Bilder, 17 Tabellen. 1993.

Band 34: Uwe Günzel  
**Entwicklung und Einsatz eines Simulationsverfahrens für operative  
und strategische Probleme der Produktionsplanung und –steuerung**  
XIV u. 170 Seiten, 66 Bilder, 5 Tabellen. 1993.

Band 35: Bertram Ehmann  
**Operatives Fertigungscontrolling durch Optimierung  
auftragsbezogener Bearbeitungsabläufe in der Elektronikfertigung**  
XV u. 167 Seiten, 114 Bilder. 1993.

Band 36: Harald Kolléra  
**Entwicklung eines benutzerorientierten Werkstattprogrammiersystems  
für das Laserstrahlschneiden**  
129 Seiten, 66 Bilder, 1 Tabelle. 1993.

Band 37: Stephanie Abels  
**Modellierung und Optimierung von Montageanlagen  
in einem integrierten Simulationssystem**  
188 Seiten, 88 Bilder. 1993.

Band 38: Robert Schmidt–Hebbel  
**Laserstrahlbohren durchflußbestimmender Durchgangslöcher**  
145 Seiten, 63 Bilder, 11 Tabellen. 1993.

Band 39: Norbert Lutz  
**Oberflächenfeinbearbeitung keramischer Werkstoffe  
mit XeCl–Excimerlaserstrahlung**  
187 Seiten, 98 Bilder, 29 Tabellen. 1994.

Band 40: Konrad Grampp  
**Rechnerunterstützung bei Test und Schulung  
an Steuerungssoftware von SMD–Bestücklinien**  
178 Seiten, 88 Bilder. 1995.

Band 41: Martin Koch  
**Wissensbasierte Unterstützung der Angebotsbearbeitung  
in der Investitionsgüterindustrie**  
169 Seiten, 68 Bilder. 1995.

Band 42: Armin Gropp  
**Anlagen- und Prozeßdiagnostik  
beim Schneiden mit einem gepulsten Nd:YAG–Laser**  
160 Seiten, 88 Bilder, 7 Tabellen. 1995.

Band 43: Werner Heckel  
**Optische 3D-Konturerfassung und on-line Biegewinkelmessung  
mit dem Lichtschnittverfahren**  
149 Seiten, 43 Bilder, 11 Tabellen. 1995.

Band 44: Armin Rothhaupt  
**Modulares Planungssystem  
zur Optimierung der Elektronikfertigung**  
180 Seiten, 101 Bilder. 1995.

Band 45: Bernd Zöllner  
**Adaptive Diagnose in der Elektronikproduktion**  
195 Seiten, 74 Bilder, 3 Tabellen. 1995.

Band 46: Bodo Vormann  
**Beitrag zur automatisierten Handhabungsplanung  
komplexer Blechbiegeteile**  
126 Seiten, 89 Bilder, 3 Tabellen. 1995.

Band 47: Peter Schnepf  
**Zielkostenorientierte Montageplanung**  
144 Seiten, 75 Bilder. 1995.

Band 48: Rainer Klotzbücher  
**Konzept zur rechnerintegrierten Materialversorgung  
in flexiblen Fertigungssystemen**  
156 Seiten, 62 Bilder. 1995.

Band 49: Wolfgang Greska  
**Wissensbasierte Analyse und Klassifizierung von Blechteilen**  
144 Seiten, 96 Bilder. 1995.

Band 50: Jörg Franke  
**Integrierte Entwicklung neuer Produkt- und Produktionstechnologien  
für räumliche spritzgegossene Schaltungsverträge (3-D MID)**  
196 Seiten, 86 Bilder, 4 Tabellen. 1995.

Band 51: Franz-Josef Zeller  
**Sensorplanung und schnelle Sensorregelung für Industrieroboter**  
190 Seiten, 102 Bilder, 9 Tabellen. 1995.

Band 52: Michael Solvie  
**Zeitbehandlung und Multimedia-Unterstützung  
in Feldkommunikationssystemen**  
200 Seiten, 87 Bilder, 35 Tabellen. 1996.

Band 53: Robert Hopperdietzel  
**Reengineering in der Elektro- und Elektronikindustrie**  
180 Seiten, 109 Bilder, 1 Tabelle. 1996.  
ISBN 3-87525-070-2

Band 54: Thomas Rebhan  
**Beitrag zur Mikromaterialbearbeitung mit Excimerlasern –  
Systemkomponenten und Verfahrensoptimierungen**  
148 Seiten, 61 Bilder, 10 Tabellen. 1996.  
ISBN 3-87525-075-3

Band 55: Henning Hanebuth  
**Laserstrahlhartlöten mit Zweistrahltechnik**  
157 Seiten, 58 Bilder, 11 Tabellen. 1996.  
ISBN 3-87525-074-5

Band 56: Uwe Schönherr  
**Steuerung und Sensordatenintegration für flexible Fertigungszellen  
mit kooperierenden Robotern**  
188 Seiten, 116 Bilder, 3 Tabellen. 1996.  
ISBN 3-87525-076-1

Band 57: Stefan Holzer  
**Berührungslose Formgebung mit Laserstrahlung**  
162 Seiten, 69 Bilder, 11 Tabellen. 1996.  
ISBN 3-87525-079-6

Band 58: Markus Schultz  
**Fertigungsqualität beim 3D–Laserstrahlschweißen  
von Blechformteilen**  
165 Seiten, 88 Bilder, 9 Tabellen. 1997.  
ISBN 3-87525-080-X

Band 59: Thomas Krebs  
**Integration elektromechanischer CA–Anwendungen  
über einem STEP–Produktmodell**  
198 Seiten, 58 Bilder, 8 Tabellen. 1997.  
ISBN 3-87525-081-8

Band 60: Jürgen Sturm  
**Prozeßintegrierte Qualitätssicherung  
in der Elektronikproduktion**  
167 Seiten, 112 Bilder, 5 Tabellen. 1997.  
ISBN 3-87525-082-6

Band 61: Andreas Brand  
**Prozesse und Systeme zur Bestückung  
räumlicher elektronischer Baugruppen (3D-MID)**  
182 Seiten, 100 Bilder. 1997.  
ISBN 3-87525-087-7

Band 62: Michael Kauf  
**Regelung der Laserstrahlleistung und der Fokusparameter  
einer CO<sub>2</sub>-Hochleistungslaseranlage**  
140 Seiten, 70 Bilder, 5 Tabellen. 1997.  
ISBN 3-87525-083-4

Band 63: Peter Steinwasser  
**Modulares Informationsmanagement  
in der integrierten Produkt– und Prozeßplanung**  
190 Seiten, 87 Bilder. 1997.  
ISBN 3-87525-084-2

Band 64: Georg Liedl  
**Integriertes Automatisierungskonzept  
für den flexiblen Materialfluß in der Elektronikproduktion**  
196 Seiten, 96 Bilder, 3 Tabellen. 1997.  
ISBN 3-87525-086-9

Band 65: Andreas Otto  
**Transiente Prozesse beim Laserstrahlschweißen**  
132 Seiten, 62 Bilder, 1 Tabelle. 1997.  
ISBN 3-87525-089-3

Band 66: Wolfgang Blöchl  
**Erweiterte Informationsbereitstellung an offenen CNC–Steuerungen  
zur Prozeß– und Programmoptimierung**  
168 Seiten, 96 Bilder. 1997.  
ISBN 3-87525-091-5

Band 67: Klaus–Uwe Wolf  
**Verbesserte Prozeßführung und Prozeßplanung  
zur Leistungs– und Qualitätssteigerung beim Spulenwickeln**  
186 Seiten, 125 Bilder. 1997.  
ISBN 3-87525-092-3

Band 68: Frank Backes  
**Technologieorientierte Bahnplanung für die 3D–Laserstrahlbearbeitung**  
138 Seiten, 71 Bilder, 2 Tabellen. 1997.  
ISBN 3-87525-093-1

Band 69: Jürgen Kraus  
**Laserstrahlumformen von Profilen**  
137 Seiten, 72 Bilder, 8 Tabellen. 1997.  
ISBN 3-87525-094-X

Band 70: Norbert Neubauer  
**Adaptive Strahlführungen für CO<sub>2</sub>-Laseranlagen**  
120 Seiten, 50 Bilder, 3 Tabellen. 1997.  
ISBN 3-87525-095-8

Band 71: Michael Steber  
**Prozeßoptimierter Betrieb flexibler Schraubstationen  
in der automatisierten Montage**  
168 Seiten, 78 Bilder, 3 Tabellen. 1997.  
ISBN 3-87525-096-6

Band 72: Markus Pfestorf  
**Funktionale 3D-Oberflächenkenngrößen in der Umformtechnik**  
162 Seiten, 84 Bilder, 15 Tabellen. 1997.  
ISBN 3-87525-097-4

Band 73: Volker Franke  
**Integrierte Planung und Konstruktion  
von Werkzeugen für die Biegebearbeitung**  
143 Seiten, 81 Bilder. 1998.  
ISBN 3-87525-098-2

Band 74: Herbert Scheller  
**Automatisierte Demontagesysteme und recyclinggerechte  
Produktgestaltung elektronischer Baugruppen**  
184 Seiten, 104 Bilder, 17 Tabellen. 1998.  
ISBN 3-87525-099-0

Band 75: Arthur Meßner  
**Kaltmassivumformung metallischer Kleinstteile  
– Werkstoffverhalten, Wirkflächenreibung, Prozeßauslegung**  
164 Seiten, 92 Bilder, 14 Tabellen. 1998.  
ISBN 3-87525-100-8

Band 76: Mathias Glasmacher  
**Prozeß- und Systemtechnik zum Laserstrahl-Mikroschweißen**  
184 Seiten, 104 Bilder, 12 Tabellen. 1998.  
ISBN 3-87525-101-6

Band 77: Michael Schwind  
**Zerstörungsfreie Ermittlung mechanischer Eigenschaften  
von Feinblechen mit dem Wirbelstromverfahren**  
124 Seiten, 68 Bilder, 8 Tabellen. 1998.  
ISBN 3-87525-102-4

Band 78: Manfred Gerhard  
**Qualitätssteigerung in der Elektronikproduktion durch Optimierung  
der Prozeßführung beim Lötén komplexer Baugruppen**  
179 Seiten, 113 Bilder, 7 Tabellen. 1998.  
ISBN 3-87525-103-2

Band 79: Elke Rauh  
**Methodische Einbindung der Simulation  
in die betrieblichen Planungs- und Entscheidungsabläufe**  
192 Seiten, 114 Bilder, 4 Tabellen. 1998.  
ISBN 3-87525-104-0

Band 80: Sorin Niederkorn  
**Meßeinrichtung zur Untersuchung der Wirkflächenreibung  
bei umformtechnischen Prozessen**  
99 Seiten, 46 Bilder, 6 Tabellen. 1998.  
ISBN 3-87525-105-9

Band 81: Stefan Schuberth  
**Regelung der Fokusslage beim Schweißen mit CO<sub>2</sub>-Hochleistungslasern  
unter Einsatz von adaptiven Optiken**  
140 Seiten, 64 Bilder, 3 Tabellen. 1998.  
ISBN 3-87525-106-7

Band 82: Armando Walter Colombo  
**Development and Implementation of Hierarchical Control Structures  
of Flexible Production Systems Using High Level Petri Nets**  
216 Seiten, 86 Bilder. 1998.  
ISBN 3-87525-109-1

Band 83: Otto Meedt  
**Effizienzsteigerung bei Demontage und Recycling  
durch flexible Demontagetechnologien und optimierte Produktgestaltung**  
186 Seiten, 103 Bilder. 1998.  
ISBN 3-87525-108-3



Band 84: Knuth Götz  
**Modelle und effiziente Modellbildung  
zur Qualitätssicherung in der Elektronikproduktion**  
212 Seiten, 129 Bilder, 24 Tabellen. 1998.  
ISBN 3-87525-112-1

Band 85: Ralf Luchs  
**Einsatzmöglichkeiten leitender Klebstoffe  
zur zuverlässigen Kontaktierung elektronischer Bauelemente in der SMT**  
176 Seiten, 126 Bilder, 30 Tabellen. 1998.  
ISBN 3-87525-113-7

Band 86: Frank Pöhlau  
**Entscheidungsgrundlagen zur Einführung  
räumlicher spritzgegossener Schaltungsträger (3-D MID)**  
144 Seiten, 99 Bilder. 1999.  
ISBN 3-87525-114-8

Band 87: Roland T. A. Kals  
**Fundamentals on the miniaturization of sheet metal working processes**  
128 Seiten, 58 Bilder, 11 Tabellen. 1999.  
ISBN 3-87525-115-6

Band 88: Gerhard Luhn  
**Implizites Wissen und technisches Handeln  
am Beispiel der Elektronikproduktion**  
252 Seiten, 61 Bilder, 1 Tabelle. 1999.  
ISBN 3-87525-116-4

Band 89: Axel Sprenger  
**Adaptives Streckbiegen von Aluminium-Strangpreßprofilen**  
114 Seiten, 63 Bilder, 4 Tabellen. 1999.  
ISBN 3-87525-117-2

Band 90: Hans-Jörg Pucher  
**Untersuchungen zur Prozeßfolge Umformen, Bestücken  
und Laserstrahllöten von Mikrokontakten**  
158 Seiten, 69 Bilder, 9 Tabellen. 1999.  
ISBN 3-87525-119-9

Band 91: Horst Arnet  
**Profilbiegen mit kinematischer Gestalterzeugung**  
128 Seiten, 67 Bilder, 7 Tabellen. 1999.  
ISBN 3-87525-120-2

Band 92: Doris Schubart  
**Prozeßmodellierung und Technologieentwicklung  
beim Abtragen mit CO<sub>2</sub>-Laserstrahlung**  
133 Seiten, 57 Bilder, 13 Tabellen. 1999.  
ISBN 3-87525-122-9

Band 93: Adrianus L. P. Coremans  
**Laserstrahlsintern von Metallpulver – Prozeßmodellierung,  
Systemtechnik, Eigenschaften laserstrahlgesinterter Metallkörper**  
184 Seiten, 108 Bilder, 12 Tabellen. 1999.  
ISBN 3-87525-124-5

Band 94: Hans-Martin Biehler  
**Optimierungskonzepte für Qualitätsdatenverarbeitung  
und Informationsbereitstellung in der Elektronikfertigung**  
194 Seiten, 105 Bilder. 1999.  
ISBN 3-87525-126-1

Band 95: Wolfgang Becker  
**Oberflächenausbildung und tribologische Eigenschaften  
excimerlaserstrahlbearbeiteter Hochleistungskeramiken**  
175 Seiten, 71 Bilder, 3 Tabellen. 1999.  
ISBN 3-87525-127-X

Band 96: Philipp Hein  
**Innenhochdruck-Umformen von Blechpaaren:  
Modellierung, Prozeßauslegung und Prozeßführung**  
129 Seiten, 57 Bilder, 7 Tabellen. 1999.  
ISBN 3-87525-128-8

Band 97: Gunter Beitinger  
**Herstellungs- und Prüfverfahren  
für thermoplastische Schaltungsträger**  
169 Seiten, 92 Bilder, 20 Tabellen. 1999.  
ISBN 3-87525-129-6

Band 98: Jürgen Knobloch  
**Beitrag zur rechnerunterstützten verursachungsgerechten Angebotskalkulation von  
Blechteilen mit Hilfe wissensbasierter Methoden**  
155 Seiten, 53 Bilder, 26 Tabellen. 1999.  
ISBN 3-87525-130-X

Band 99: Frank Breitenbach  
**Bildverarbeitungssystem zur Erfassung der Anschlußgeometrie  
elektronischer SMT-Bauelemente**  
147 Seiten, 92 Bilder, 12 Tabellen. 2000.  
ISBN 3-87525-131-8

Band 100: Bernd Falk  
**Simulationsbasierte Lebensdauervorhersage  
für Werkzeuge der Kaltmassivumformung**  
134 Seiten, 44 Bilder, 15 Tabellen. 2000.  
ISBN 3-87525-136-9

Band 101: Wolfgang Schlögl  
**Integriertes Simulationsdaten-Management  
für Maschinenentwicklung und Anlagenplanung**  
169 Seiten, 101 Bilder, 20 Tabellen. 2000.  
ISBN 3-87525-137-7

Band 102: Christian Hinsel  
**Ermüdungsbruchversagen hartstoffbeschichteter  
Werkzeugstähle in der Kaltmassivumformung**  
130 Seiten, 80 Bilder, 14 Tabellen. 2000.  
ISBN 3-87525-138-5

Band 103: Stefan Bobbert  
**Simulationsgestützte Prozessauslegung  
für das Innenhochdruck-Umformen von Blechpaaren**  
123 Seiten, 77 Bilder. 2000.  
ISBN 3-87525-145-8

Band 104: Harald Rottbauer  
**Modulares Planungswerkzeug  
zum Produktionsmanagement in der Elektronikproduktion**  
166 Seiten, 106 Bilder. 2001.  
ISBN 3-87525-139-3

Band 105: Thomas Hennige  
**Flexible Formgebung von Blechen durch Laserstrahlumformen**  
119 Seiten, 50 Bilder. 2001.  
ISBN 3-87525-140-7

Band 106: Thomas Menzel  
**Wissensbasierte Methoden für die rechnergestützte Charakterisierung  
und Bewertung innovativer Fertigungsprozesse**  
152 Seiten, 71 Bilder. 2001.  
ISBN 3-87525-142-3

Band 107: Thomas Stöckel  
**Kommunikationstechnische Integration der Prozeßebene  
in Produktionssysteme durch Middleware-Frameworks**  
147 Seiten, 65 Bilder, 5 Tabellen. 2001.  
ISBN 3-87525-143-1

Band 108: Frank Pitter  
**Verfügbarkeitssteigerung von Werkzeugmaschinen  
durch Einsatz mechatronischer Sensorlösungen**  
158 Seiten, 131 Bilder, 8 Tabellen. 2001.  
ISBN 3-87525-144-X

Band 109: Markus Korneli  
**Integration lokaler CAP-Systeme  
in einen globalen Fertigungsdatenverbund**  
121 Seiten, 53 Bilder, 11 Tabellen. 2001.  
ISBN 3-87525-146-6

Band 110: Burkhard Müller  
**Laserstrahljustieren mit Excimer-Lasern – Prozessparameter und Modelle zur Aktorkonstruktion**  
128 Seiten, 36 Bilder, 9 Tabellen. 2001  
ISBN 3-87525-159-8

Band 111: Jürgen Göhringer  
**Integrierte Telediagnose via Internet zum effizienten Service von Produktionssystemen**  
178 Seiten, 98 Bilder, 5 Tabellen. 2001.  
ISBN 3-87525-147-4

Band 112: Robert Feuerstein  
**Qualitäts- und kosteneffiziente Integration neuer Bauelementetechnologien in die Flachbaugruppenfertigung**  
161 Seiten, 99 Bilder, 10 Tabellen. 2001.  
ISBN 3-87525-151-2

Band 113: Marcus Reichenberger  
**Eigenschaften und Einsatzmöglichkeiten alternativer Elektroniklote in der Oberflächenmontage (SMT)**  
165 Seiten, 97 Bilder, 18 Tabellen. 2001.  
ISBN 3-87525-152-0

Band 114: Alexander Huber  
**Justieren vormontierter Systeme mit dem Nd:YAG-Laser unter Einsatz von Aktoren**  
122 Seiten, 58 Bilder, 5 Tabellen. 2001.  
ISBN 3-87525-153-9

Band 115: Sami Krimi  
**Analyse und Optimierung von Montagesystemen in der Elektronikproduktion**  
155 Seiten, 88 Bilder, 3 Tabellen. 2001.  
ISBN 3-87525-157-1

Band 116: Marion Merklein  
**Laserstrahlumformen von Aluminiumwerkstoffen - Beeinflussung der Mikrostruktur und der mechanischen Eigenschaften**  
122 Seiten, 65 Bilder, 15 Tabellen. 2001.  
ISBN 3-87525-156-3

Band 117: Thomas Collisi  
**Ein informationslogistisches Architekturkonzept zur Akquisition simulationsrelevanter Daten**  
181 Seiten, 105 Bilder, 7 Tabellen. 2002.  
ISBN 3-87525-164-4

Band 118: Markus Koch  
**Rationalisierung und ergonomische Optimierung im Innenausbau durch den Einsatz moderner Automatisierungstechnik**  
176 Seiten, 98 Bilder, 9 Tabellen. 2002.  
ISBN 3-87525-165-2

Band 119: Michael Schmidt  
**Prozessregelung für das Laserstrahl-Punktschweißen in der Elektronikproduktion**  
152 Seiten, 71 Bilder, 3 Tabellen. 2002.  
ISBN 3-87525-166-0

Band 120: Nicolas Tiesler  
**Grundlegende Untersuchungen zum Fließpressen metallischer Kleinstteile**  
126 Seiten, 78 Bilder, 12 Tabellen. 2002.  
ISBN 3-87525-175-X

Band 121: Lars Pursche  
**Methoden zur technologieorientierten Programmierung für die 3D-Lasermikrobearbeitung**  
111 Seiten, 39 Bilder, 0 Tabellen. 2002.  
ISBN 3-87525-183-0

Band 122: Jan-Oliver Brassel  
**Prozesskontrolle beim Laserstrahl-Mikroschweißen**  
148 Seiten, 72 Bilder, 12 Tabellen. 2002.  
ISBN 3-87525-181-4

Band 123: Mark Geisel

**Prozeßkontrolle und -steuerung beim Laserstrahlschweißen  
mit den Methoden der nichtlinearen Dynamik**

135 Seiten, 46 Bilder, 2 Tabellen. 2002.  
ISBN 3-87525-180-6

Band 124: Gerd Eßer

**Laserstrahlunterstützte Erzeugung metallischer Leiterstrukturen auf  
Thermoplastsubstraten für die MID-Technik**

148 Seiten, 60 Bilder, 6 Tabellen. 2002.  
ISBN 3-87525-171-7

Band 125: Marc Fleckenstein

**Qualität laserstrahl-gefügter Mikroverbindungen  
elektronischer Kontakte**

159 Seiten, 77 Bilder, 7 Tabellen. 2002.  
ISBN 3-87525-170-9

Band 126: Stefan Kaufmann

**Grundlegende Untersuchungen zum Nd:YAG- Laserstrahlfügen  
von Silizium für Komponenten der Optoelektronik**

159 Seiten, 100 Bilder, 6 Tabellen. 2002.  
ISBN 3-87525-172-5

Band 127: Thomas Fröhlich

**Simultanes Lötten von Anschlußkontakten elektronischer Bauelemente  
mit Diodenlaserstrahlung**

143 Seiten, 75 Bilder, 6 Tabellen. 2002.  
ISBN 3-87525-186-5

Band 128: Achim Hofmann

**Erweiterung der Formgebungsgrenzen beim Umformen von  
Aluminiumwerkstoffen durch den Einsatz prozessangepasster Platinen**

113 Seiten, 58 Bilder, 4 Tabellen  
ISBN 3-87525-182-2

Band 129: Ingo Kriebitzsch

**3 - D MID Technologie in der Automobilelektronik**

129 Seiten, 102 Bilder, 10 Tabellen. 2002.  
ISBN 3-87525-169-5

Band 130: Thomas Pohl

**Fertigungsqualität und Umformbarkeit laserstrahlgeschweißter  
Formplatinen aus Aluminiumlegierungen**

133 Seiten, 93 Bilder, 12 Tabellen. 2002  
ISBN 3-87525-173-3

Band 131: Matthias Wenk

**Entwicklung eines konfigurierbaren Steuerungssystems für die  
flexible Sensorführung von Industrierobotern**

167 Seiten, 85 Bilder, 1 Tabelle. 2002.  
ISBN 3-87525-174-1

Band 132: Matthias Negenandank

**Neue Sensorik und Aktorik für Bearbeitungsköpfe  
zum Laserstrahlschweißen**

116 Seiten, 60 Bilder, 14 Tabellen  
ISBN 3-87525-184-9

Band 133: Oliver Kreis

**Integrierte Fertigung –  
Verfahrensintegration durch Innenhochdruck-Umformen, Trennen und  
Laserstrahlschweißen in einem Werkzeug sowie ihre tele- und multimediale Präsentation**

167 Seiten, 90 Bilder, 43 Tabellen  
ISBN 3-87525-176-8

Band 134: Stefan Trautner

**Technische Umsetzung produktbezogener Instrumente der  
Umweltpolitik bei Elektro- und Elektronikgeräten**

179 Seiten, 92 Bilder, 11 Tabellen. 2002.  
ISBN 3-87525-177-6

Band 135: Roland Meier

**Strategien für einen produktorientierten Einsatz räumlicher  
spritzgegossener Schaltungsträger (3-D MID)**

155 Seiten, 88 Bilder, 14 Tabellen. 2002.

ISBN 3-87525-178-4

Band 136: Jürgen Wunderlich

**Kostensimulation – Simulationsbasierte Wirtschaftlichkeitsregelung  
komplexer Produktionssysteme**

202 Seiten, 119 Bilder, 17 Tabellen. 2002.

ISBN 3-87525-179-2

Band 137: Stefan Novotny

**Innenhochdruck-Umformen von Blechen aus Aluminium- und  
Magnesiumlegierungen bei erhöhter Temperatur**

132 Seiten, 82 Bilder, 6 Tabellen. 2002.

ISBN 3-87525-185-7

Band 138: Andreas Licha

**Flexible Montageautomatisierung zur Komplettmontage flächenhafter  
Produktstrukturen durch kooperierende Industrieroboter**

158 Seiten, 87 Bilder, 8 Tabellen. 2003.

ISBN 3-87525-189-X

Band 139: Michael Eisenbarth

**Beitrag zur Optimierung der Aufbau- und Verbindungstechnik  
für mechatronische Baugruppen**

207 Seiten, 141 Bilder, 9 Tabellen. 2003.

ISBN 3-87525-190-3

Band 140: Frank Christoph

**Durchgängige simulationsgestützte Planung von  
Fertigungseinrichtungen der Elektronikproduktion**

187 Seiten, 107 Bilder, 9 Tabellen. 2003.

ISBN 3-87525-191-1

Band 141: Hinnerk Hagenah

**Simulationsbasierte Bestimmung der zu erwartenden  
Maßhaltigkeit für das Blechbiegen**

131 Seiten, 36 Bilder, 26 Tabellen. 2003.

ISBN 3-87525-192-X

Band 142: Ralf Eckstein

**Scherschneiden und Biegen metallischer Kleinstteile –  
Materialeinfluss und Materialverhalten**

148 Seiten, 71 Bilder, 19 Tabellen. 2003.

ISBN 3-87525-193-8

Band 143: Frank H. Meyer-Pittroff

**Excimerlaserstrahlbiegen dünner metallischer Folien  
mit homogener Lichtlinie**

138 Seiten, 60 Bilder, 16 Tabellen. 2003.

ISBN 3-87525-196-2

Band 144: Andreas Kach

**Rechnergestützte Anpassung von Laserstrahl Schneidbahnen  
an Bauteilabweichungen**

139 Seiten, 69 Bilder, 11 Tabellen. 2004.

ISBN 3-87525-197-0

Band 145: Stefan Hierl

**System- und Prozeßtechnik für das simultane Löten mit  
Diodenlaserstrahlung von elektronischen Bauelementen**

124 Seiten, 66 Bilder, 4 Tabellen. 2004.

ISBN 3-87525-198-9

Band 146: Thomas Neudecker

**Tribologische Eigenschaften keramischer Blechumformwerkzeuge-  
Einfluss einer Oberflächenendbearbeitung mittels Excimerlaserstrahlung**

166 Seiten, 75 Bilder, 26 Tabellen. 2004.

ISBN 3-87525-200-4

Band 147: Ulrich Wenger

**Prozessoptimierung in der Wickeltechnik durch innovative maschinenbauliche und regelungstechnische Ansätze**

132 Seiten, 88 Bilder, 0 Tabellen. 2004.  
ISBN 3-87525-203-9

Band 148: Stefan Slama

**Effizienzsteigerung in der Montage durch marktorientierte Montagestructuren und erweiterte Mitarbeiterkompetenz**

188 Seiten, 125 Bilder, 0 Tabellen. 2004.  
ISBN 3-87525-204-7

Band 149: Thomas Wurm

**Laserstrahljustieren mittels Aktoren – Entwicklung von Konzepten und Methoden für die rechnerunterstützte Modellierung und Optimierung von komplexen Aktorsystemen in der Mikrotechnik**

122 Seiten, 51 Bilder, 9 Tabellen. 2004.  
ISBN 3-87525-206-3

Band 150: Martino Celeghini

**Wirkmedienbasierte Blechumformung: Grundlagenuntersuchungen zum Einfluss von Werkstoff und Bauteilgeometrie**

146 Seiten, 77 Bilder, 6 Tabellen. 2004.  
ISBN 3-87525-207-1

Band 151: Ralph Hohenstein

**Entwurf hochdynamischer Sensor- und Regelsysteme für die adaptive Laserbearbeitung**

282 Seiten, 63 Bilder, 16 Tabellen. 2004.  
ISBN 3-87525-210-1

Band 152: Angelika Hutterer

**Entwicklung prozessüberwachender Regelkreise für flexible Formgebungsprozesse**

149 Seiten, 57 Bilder, 2 Tabellen. 2005.  
ISBN 3-87525-212-8

Band 153: Emil Egerer

**Massivumformen metallischer Kleinstteile bei erhöhter Prozesstemperatur**

158 Seiten, 87 Bilder, 10 Tabellen. 2005.  
ISBN 3-87525-213-6

Band 154: Rüdiger Holzmann

**Strategien zur nachhaltigen Optimierung von Qualität und Zuverlässigkeit in der Fertigung hochintegrierter Flachbaugruppen**

186 Seiten, 99 Bilder, 19 Tabellen. 2005.  
ISBN 3-87525-217-9

Band 155: Marco Nock

**Biegeumformen mit Elastomerwerkzeugen**

Modellierung, Prozessauslegung und Abgrenzung des Verfahrens am Beispiel des Rohrbiegens  
164 Seiten, 85 Bilder, 13 Tabellen. 2005.  
ISBN 3-87525-218-7

Band 156: Frank Niebling

**Qualifizierung einer Prozesskette zum Laserstrahlintern metallischer Bauteile**

148 Seiten, 89 Bilder, 3 Tabellen. 2005.  
ISBN 3-87525-219-5

Band 157: Markus Meiler

**Großserientauglichkeit trockenschmierstoffbeschichteter Aluminiumbleche im Presswerk**

Grundlegende Untersuchungen zur Tribologie, zum Umformverhalten und Bauteilversuche  
104 Seiten, 57 Bilder, 21 Tabellen. 2005.  
ISBN 3-87525-221-7

Band 158: Agus Sutanto

**Solution Approaches for Planning of Assembly Systems  
in Three-Dimensional Virtual Environments**

169 Seiten, 98 Bilder, 3 Tabellen. 2005.

ISBN 3-87525-220-9

Band 159: Matthias Boiger

**Hochleistungssysteme für die Fertigung elektronischer Baugruppen  
auf der Basis flexibler Schaltungsträger**

175 Seiten, 111 Bilder, 8 Tabellen. 2005.

ISBN 3-87525-222-5

Band 160: Matthias Pitz

**Laserunterstütztes Biegen höchstfester Mehrphasenstähle**

120 Seiten, 73 Bilder, 11 Tabellen. 2005.

ISBN 3-87525-223-3

Band 161: Meik Vahl

**Beitrag zur gezielten Beeinflussung des Werkstoffflusses  
beim Innenhochdruck-Umformen von Blechen**

165 Seiten, 94 Bilder, 15 Tabellen. 2005.

ISBN 3-87525-224-1

Band 162: Peter K. Kraus

**Plattformstrategien – Realisierung einer varianz- und  
kostenoptimierten Wertschöpfung**

181 Seiten, 95 Bilder, 0 Tabellen. 2005.

ISBN 3-87525-226-8

Band 163: Adrienn Cser

**Laserstrahlschmelzabtrag – Prozessanalyse und -modellierung**

146 Seiten, 79 Bilder, 3 Tabellen. 2005.

ISBN 3-87525-227-6

Band 164: Markus C. Hahn

**Grundlegende Untersuchungen zur Herstellung von  
Leichtbauverbundstrukturen mit Aluminiumschaumkern**

143 Seiten, 60 Bilder, 16 Tabellen. 2005.

ISBN 3-87525-228-4

Band 165: Gordana Michos

**Mechatronische Ansätze zur Optimierung von Vorschubachsen**

146 Seiten, 87 Bilder, 17 Tabellen. 2005.

ISBN 3-87525-230-6

Band 166: Markus Stark

**Auslegung und Fertigung hochpräziser Faser-Kollimator-Arrays**

158 Seiten, 115 Bilder, 11 Tabellen. 2005.

ISBN 3-87525-231-4

Band 167: Yurong Zhou

**Kollaboratives Engineering Management in der integrierten virtuellen  
Entwicklung der Anlagen für die Elektronikproduktion**

156 Seiten, 84 Bilder, 6 Tabellen. 2005.

ISBN 3-87525-232-2

Band 168: Werner Enser

**Neue Formen permanenter und lösbarer elektrischer  
Kontaktierungen für mechatronische Baugruppen**

190 Seiten, 112 Bilder, 5 Tabellen. 2005.

ISBN 3-87525-233-0

Band 169: Katrin Melzer

**Integrierte Produktpolitik bei elektrischen und elektronischen  
Geräten zur Optimierung des Product-Life-Cycle**

155 Seiten, 91 Bilder, 17 Tabellen. 2005.

ISBN 3-87525-234-9

Band 170: Alexander Putz

**Grundlegende Untersuchungen zur Erfassung der realen Vorspannung von  
armierten Kaltfließpresswerkzeugen mittels Ultraschall**

137 Seiten, 71 Bilder, 15 Tabellen. 2006.

ISBN 3-87525-237-3

Band 171: Martin Prechtl

**Automatisiertes Schichtverfahren für metallische Folien –  
System- und Prozesstechnik**

154 Seiten, 45 Bilder, 7 Tabellen. 2006.  
ISBN 3-87525-238-1

Band 172: Markus Meidert

**Beitrag zur deterministischen Lebensdauerabschätzung  
von Werkzeugen der Kaltmassivumformung**

131 Seiten, 78 Bilder, 9 Tabellen. 2006.  
ISBN 3-87525-239-X

Band 173: Bernd Müller

**Robuste, automatisierte Montagesysteme durch adaptive Prozessführung  
und montageübergreifende Fehlerprävention am Beispiel flächiger Leichtbauteile**

147 Seiten, 18 Bilder, 0 Tabellen. 2006.  
ISBN 3-87525-240-3

Band 174: Alexander Hofmann

**Hybrides Laserdurchstrahlsschweißen von Kunststoffen**

136 Seiten, 72 Bilder, 4 Tabellen. 2006.  
ISBN 978-3-87525-243-9  
ISBN 3-87525-243-8

Band 175: Peter Wölflick

**Innovative Substrate und Prozesse mit feinsten Strukturen  
für bleifreie Mechatronik-Anwendungen**

177 Seiten, 148 Bilder, 24 Tabellen. 2006.  
ISBN 978-3-87525-246-0  
ISBN 3-87525-246-2

Band 176: Attila Komlódi

**Detection and Prevention of Hot Cracks during Laser Welding  
of Aluminium Alloys Using Advanced Simulation Methods**

155 Seiten, 89 Bilder, 14 Tabellen. 2006.  
ISBN 978-3-87525-248-4  
ISBN 3-87525-248-9

Band 177: Uwe Popp

**Grundlegende Untersuchungen zum Laserstrahlstrukturieren  
von Kaltmassivumformwerkzeugen**

140 Seiten, 67 Bilder, 16 Tabellen. 2006.  
ISBN 978-3-87525-249-1  
ISBN 3-87525-249-7

Band 178: Veit Rückel

**Rechnergestützte Ablaufplanung und Bahngenerierung  
Für kooperierende Industrieroboter**

148 Seiten, 75 Bilder, 7 Tabellen. 2006.  
ISBN 978-3-87525-250-7  
ISBN 3-87525-250-0

Band 179: Manfred Dirscherl

**Nicht-thermische Mikrojustiertechnik mittels ultrakurzer Laserpulse**

154 Seiten, 69 Bilder, 10 Tabellen. 2007.  
ISBN 978-3-87525-251-4  
ISBN 3-87525-251-9

Band 180: Yong Zhuo

**Entwurf eines rechnergestützten integrierten Systems für Konstruktion und  
Fertigungsplanung räumlicher spritzgegossener Schaltungsträger (3D-MID)**

181 Seiten, 95 Bilder, 5 Tabellen. 2007.  
ISBN 978-3-87525-253-8

Band 181: Stefan Lang

**Durchgängige Mitarbeiterinformation zur Steigerung von Effizienz und  
Prozesssicherheit in der Produktion**

172 Seiten, 93 Bilder. 2007.  
ISBN 978-3-87525-257-6

Band 182: Hans-Joachim Krauß

**Laserstrahlinduzierte Pyrolyse präkeramischer Polymere**

171 Seiten, 100 Bilder. 2007.  
ISBN 978-3-87525-258-3



Band 183: Stefan Junker  
**Technologien und Systemlösungen für die flexibel automatisierte Bestückung  
permanent erregter Läufer mit oberflächenmontierten Dauermagneten**  
173 Seiten, 75 Bilder. 2007.  
ISBN 978-3-87525-259-0

Band 184: Rainer Kohlbauer  
**Wissensbasierte Methoden für die simulationsgestützte Auslegung  
wirkmedienbasierter Blechumformprozesse**  
135 Seiten, 50 Bilder. 2007.  
ISBN 978-3-87525-260-6

Band 185: Klaus Lamprecht  
**Wirkmedienbasierte Umformung tiefgezogener Vorformen unter besonderer  
Berücksichtigung maßgeschneiderter Halbzeuge**  
137 Seiten, 81 Bilder. 2007.  
ISBN 978-3-87525-265-1

Band 186: Bernd Zolleiß  
**Optimierte Prozesse und Systeme für die Bestückung  
mechatronischer Baugruppen**  
180 Seiten, 117 Bilder. 2007.  
ISBN 978-3-87525-266-8

Band 187: Michael Kerausch  
**Simulationsgestützte Prozessauslegung für das Umformen lokal  
wärmebehandelter Aluminiumplatten**  
146 Seiten, 76 Bilder, 7 Tabellen. 2007.  
ISBN 978-3-87525-267-5

Band 188: Matthias Weber  
**Unterstützung der Wandlungsfähigkeit von Produktionsanlagen  
durch innovative Softwaresysteme**  
183 Seiten, 122 Bilder, 3 Tabellen. 2007.  
ISBN 978-3-87525-269-9

Band 189: Thomas Frick  
**Untersuchung der prozessbestimmenden Strahl-Stoff-Wechselwirkungen  
beim Laserstrahlschweißen von Kunststoffen**  
104 Seiten, 62 Bilder, 8 Tabellen. 2007.  
ISBN 978-3-87525-268-2

Band 190: Joachim Hecht  
**Werkstoffcharakterisierung und Prozessauslegung für die wirkmedienbasierte  
Doppelblech-Umformung von Magnesiumlegierungen**  
107 Seiten, 91 Bilder, 2 Tabellen. 2007.  
ISBN 978-3-87525-270-5

Band 191: Ralf Völkl  
**Stochastische Simulation zur Werkzeuglebensdaueroptimierung und  
Präzisionsfertigung in der Kaltmassivumformung**  
178 Seiten, 75 Bilder, 12 Tabellen. 2008.  
ISBN 978-3-87525-272-9

Band 192: Massimo Tolazzi  
**Innenhochdruck-Umformen verstärkter Blech-Rahmenstrukturen**  
164 Seiten, 85 Bilder, 7 Tabellen. 2008.  
ISBN 978-3-87525-273-6

Band 193: Cornelia Hoff  
**Untersuchung der Prozesseinflussgrößen beim  
Presshärten des höchstfesten Vergütungsstahls 22MnB5**  
133 Seiten, 92 Bilder, 5 Tabellen. 2008.  
ISBN 978-3-87525-275-0

Band 194: Christian Alvarez  
**Simulationsgestützte Methoden zur effizienten Gestaltung von  
Lötprozessen in der Elektronikproduktion**  
149 Seiten, 86 Bilder, 8 Tabellen. 2008.  
ISBN 978-3-87525-277-4

Band 195: Andreas Kunze  
**Automatisierte Montage von makromechatronischen  
Modulen zur flexiblen Integration in hybride Pkw-Bordnetzsysteme**  
160 Seiten, 90 Bilder, 14 Tabellen. 2008.  
ISBN 978-3-87525-278-1

Band 196: Wolfgang Hußnätter  
**Grundlegende Untersuchungen zur experimentellen Ermittlung und zur Modellierung von Fließortkurven bei erhöhten Temperaturen**  
152 Seiten, 73 Bilder, 21 Tabellen 2008.  
ISBN 978-3-87525-279-8

Band 197: Thomas Bigl  
**Entwicklung, angepasste Herstellungsverfahren und erweiterte Qualitätssicherung von einsatzgerechten elektronischen Baugruppen**  
175 Seiten, 107 Bilder, 14 Tabellen 2008.  
ISBN 978-3-87525-280-4

Band 198: Stephan Roth  
**Grundlegende Untersuchungen zum Excimerlaserstrahl-Abtragen unter Flüssigkeitsfilmen**  
113 Seiten, 47 Bilder, 14 Tabellen 2008.  
ISBN 978-3-87525-281-1

Band 199: Artur Giera  
**Prozesstechnische Untersuchungen zum Rührreibschweißen metallischer Werkstoffe**  
179 Seiten, 104 Bilder, 36 Tabellen 2008.  
ISBN 978-3-87525-282-8

Band 200: Jürgen Lechler  
**Beschreibung und Modellierung des Werkstoffverhaltens von presshärtbaren Bor-Manganstählen**  
154 Seiten, 75 Bilder, 12 Tabellen 2009.  
ISBN 978-3-87525-286-6

Band 201: Andreas Blankl  
**Untersuchungen zur Erhöhung der Prozessrobustheit bei der Innenhochdruck-Umformung von flächigen Halbzeugen mit vor- bzw. nachgeschalteten Laserstrahlfügeoperationen**  
120 Seiten, 68 Bilder, 9 Tabellen 2009.  
ISBN 978-3-87525-287-3

Band 202: Andreas Schaller  
**Modellierung eines nachfrageorientierten Produktionskonzeptes für mobile Telekommunikationsgeräte**  
120 Seiten, 79 Bilder, 0 Tabellen 2009.  
ISBN 978-3-87525-289-7

Band 203: Claudius Schimpf  
**Optimierung von Zuverlässigkeitsuntersuchungen, Prüfabläufen und Nacharbeitsprozessen in der Elektronikproduktion**  
162 Seiten, 90 Bilder, 14 Tabellen 2009.  
ISBN 978-3-87525-290-3

Band 204: Simon Dietrich  
**Sensoriken zur Schwerpunktslagebestimmung der optischen Prozessemissionen beim Laserstrahl-tiefschweißen**  
138 Seiten, 70 Bilder, 5 Tabellen 2009.  
ISBN 978-3-87525-292-7

Band 205: Wolfgang Wolf  
**Entwicklung eines agentenbasierten Steuerungssystems zur Materialflussorganisation im wandelbaren Produktionsumfeld**  
167 Seiten, 98 Bilder, 2009.  
ISBN 978-3-87525-293-4