

Konfigurierbares Leitsystem für modulare Montagezellen am Beispiel von Festplatten

Von der Fakultät für Maschinenbau
der Universität Stuttgart zur Erlangung der Würde eines
Doktor-Ingenieurs (Dr.-Ing.) genehmigte Abhandlung

Vorgelegt von
Dipl.-Inform. Ralf Muckenhirn
aus Neuenburg

Hauptberichter: Univ.-Prof. Dr.-Ing. Prof. E.h. Dr.-Ing. E.h. Dr. h. c.
mult. E. Westkämper

Mitberichter: Prof. Dr. rer. nat. H. Kück

Tag der Einreichung: 30.06.2004

Tag der mündlichen Prüfung: 07.06.2005

Institut für Industrielle Fertigung und Fabrikbetrieb (IFF)

2005

IPA-IAO Forschung und Praxis

Berichte aus dem
Fraunhofer-Institut für Produktionstechnik und
Automatisierung (IPA), Stuttgart,
Fraunhofer-Institut für Arbeitswirtschaft und
Organisation (IAO), Stuttgart,
Institut für Industrielle Fertigung und
Fabrikbetrieb (IFF), Universität Stuttgart
und Institut für Arbeitswissenschaft und
Technologiemanagement (IAT), Universität Stuttgart

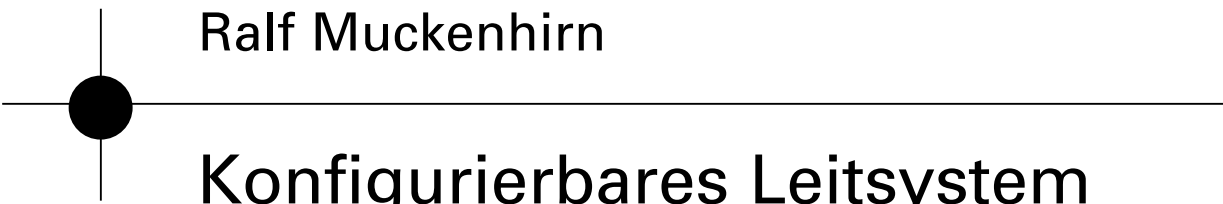
Herausgeber:

Univ.-Prof. Dr.-Ing. Prof. e.h. Dr.-Ing. e.h. Dr. h.c. mult. Engelbert Westkämper
und

Univ.-Prof. Dr.-Ing. habil. Prof. e.h. Dr. h.c. Hans-Jörg Bullinger
und

Univ.-Prof. Dr.-Ing. Dieter Spath

Ralf Muckenhirn



Konfigurierbares Leitsystem für modulare Montagezellen am Beispiel von Festplatten

Nr. 420

Dr.-Ing. Dipl.-Inform. Ralf Muckenhirn

Fraunhofer-Institut für Produktionstechnik und Automatisierung (IPA), Stuttgart

Univ.-Prof. Dr.-Ing. Prof. e.h. Dr.-Ing. e.h. Dr. h.c. mult. Engelbert Westkämper

ord. Professor an der Universität Stuttgart

Fraunhofer-Institut für Produktionstechnik und Automatisierung (IPA), Stuttgart

Univ.-Prof. Dr.-Ing. habil. Prof. e.h. Dr. h.c. Hans-Jörg Bullinger

ord. Professor an der Universität Stuttgart

Präsident der Fraunhofer-Gesellschaft, München

Univ.-Prof. Dr.-Ing. Dieter Spath

ord. Professor an der Universität Stuttgart

Fraunhofer-Institut für Arbeitswirtschaft und Organisation (IAO), Stuttgart

D 93

ISBN 3-936947-63-5 Jost Jetter Verlag, Heimsheim

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der jeweils gültigen Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

© Jost Jetter Verlag, Heimsheim 2005.

Printed in Germany.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Sollte in diesem Werk direkt oder indirekt auf Gesetze, Vorschriften oder Richtlinien (z. B. DIN, VDI, VDE) Bezug genommen oder aus ihnen zitiert worden sein, so kann der Verlag keine Gewähr für die Richtigkeit, Vollständigkeit oder Aktualität übernehmen. Es empfiehlt sich, gegebenenfalls für die eigenen Arbeiten die vollständigen Vorschriften oder Richtlinien in der jeweils gültigen Fassung hinzuzuziehen.

Druck: printsystem GmbH, Heimsheim

Geleitwort der Herausgeber

Über den Erfolg und das Bestehen von Unternehmen in einer marktwirtschaftlichen Ordnung entscheidet letztendlich der Absatzmarkt. Das bedeutet, möglichst frühzeitig absatzmarktorientierte Anforderungen sowie deren Veränderungen zu erkennen und darauf zu reagieren.

Neue Technologien und Werkstoffe ermöglichen neue Produkte und eröffnen neue Märkte. Die neuen Produktions- und Informationstechnologien verwandeln signifikant und nachhaltig unsere industrielle Arbeitswelt. Politische und gesellschaftliche Veränderungen signalisieren und begleiten dabei einen Wertewandel, der auch in unseren Industriebetrieben deutlichen Niederschlag findet.

Die Aufgaben des Produktionsmanagements sind vielfältiger und anspruchsvoller geworden. Die Integration des europäischen Marktes, die Globalisierung vieler Industrien, die zunehmende Innovationsgeschwindigkeit, die Entwicklung zur Freizeitgesellschaft und die übergreifenden ökologischen und sozialen Probleme, zu deren Lösung die Wirtschaft ihren Beitrag leisten muss, erfordern von den Führungskräften erweiterte Perspektiven und Antworten, die über den Fokus traditionellen Produktionsmanagements deutlich hinausgehen.

Neue Formen der Arbeitsorganisation im indirekten und direkten Bereich sind heute schon feste Bestandteile innovativer Unternehmen. Die Entkopplung der Arbeitszeit von der Betriebszeit, integrierte Planungsansätze sowie der Aufbau dezentraler Strukturen sind nur einige der Konzepte, welche die aktuellen Entwicklungsrichtungen kennzeichnen. Erfreulich ist der Trend, immer mehr den Menschen in den Mittelpunkt der Arbeitsgestaltung zu stellen - die traditionell eher technokratisch akzentuierten Ansätze weichen einer stärkeren Human- und Organisationsorientierung. Qualifizierungsprogramme, Training und andere Formen der Mitarbeiterentwicklung gewinnen als Differenzierungsmerkmal und als Zukunftsinvestition in *Human Resources* an strategischer Bedeutung.

Von wissenschaftlicher Seite muss dieses Bemühen durch die Entwicklung von Methoden und Vorgehensweisen zur systematischen Analyse und Verbesserung des Systems Produktionsbetrieb einschließlich der erforderlichen Dienstleistungsfunktionen unterstützt werden. Die Ingenieure sind hier gefordert, in enger Zusammenarbeit mit anderen Disziplinen, z. B. der Informatik, der Wirtschaftswissenschaften und der Arbeitswissenschaft, Lösungen zu erarbeiten, die den veränderten Randbedingungen Rechnung tragen.

Die von den Herausgebern langjährig geleiteten Institute, das

- Fraunhofer-Institut für Produktionstechnik und Automatisierung (IPA),
- Fraunhofer-Institut für Arbeitswirtschaft und Organisation (IAO),
- Institut für Industrielle Fertigung und Fabrikbetrieb (IFF), Universität Stuttgart,
- Institut für Arbeitswissenschaft und Technologiemanagement (IAT), Universität Stuttgart

arbeiten in grundlegender und angewandter Forschung intensiv an den oben aufgezeigten Entwicklungen mit. Die Ausstattung der Labors und die Qualifikation der Mitarbeiter haben bereits in der Vergangenheit zu Forschungsergebnissen geführt, die für die Praxis von großem Wert waren. Zur Umsetzung gewonnener Erkenntnisse wird die Schriftenreihe „IPA-IAO - Forschung und Praxis“ herausgegeben. Der vorliegende Band setzt diese Reihe fort. Eine Übersicht über bisher erschienene Titel wird am Schluss dieses Buches gegeben.

Dem Verfasser sei für die geleistete Arbeit gedankt, dem Jost Jetter Verlag für die Aufnahme dieser Schriftenreihe in seine Angebotspalette und der Druckerei für saubere und zügige Ausführung. Möge das Buch von der Fachwelt gut aufgenommen werden.

Engelbert Westkämper Hans-Jörg Bullinger Dieter Spath

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Fraunhofer Institut für Produktionstechnik und Automatisierung (IPA) in Stuttgart.

Herrn Professor Dr.-Ing. Prof. E.h. Dr.-Ing. E.h. Dr. h.c. mult. E. Westkämper danke ich für seine wohlwollende Unterstützung und Förderung der Arbeit.

Herrn Professor Dr. rer. nat. H. Kück danke ich für die gewissenhafte Durchsicht meiner Arbeit und für die Übernahme des Mitberichts.

Ich danke allen Kolleginnen und Kollegen der Abteilung Reinst- und Mikroproduktion für die vielen Anregungen und die Unterstützung während der Entstehung der Arbeit.

Einen besonderen Dank möchte ich Herrn Akhauri Prakash Kumar, Herrn Oliver Röhrsheim und Herrn Frank Lautenbach für die Zusammenarbeit und die vielen hilfreichen Anregungen aussprechen.

Für die Unterstützung und die Durchsicht der Arbeit danke ich Herrn Frank Frauenhoffer, Frau Sandra Kritzler, Herrn Jochen Schließer, Herrn Ralf Schütz, Frau Heike Schweikart und meiner Schwester Frau Silvia Muckenhirn.

Für ihre große Unterstützung vor und während der Entstehung dieser Arbeit danke ich ganz herzlich meinen Eltern und widme ihnen dieses Buch.

Stuttgart, Juni 2005

Ralf Muckenhirn

Inhaltsverzeichnis

Abkürzungen und Formelzeichen	13
Kapitel 1 Einleitung	15
1.1 Problemstellung	15
1.2 Zielsetzung und Vorgehensweise	17
Kapitel 2 Ausgangssituation bei der Festplattenmontage	19
2.1 Aufbau einer Festplatte	19
2.2 Entwicklung und Trends bei Festplatten	21
2.3 Aktuelle Montagevarianten von Festplatten	22
2.4 Grundlegende Anforderungen an eine neue Festplattenmontage	24
Kapitel 3 Analyse des Montageablaufs und Konzeption einer Montagezelle	25
3.1 Grundlagen der Ablaufplanung	25
3.2 Montagefunktionen bei der Festplattenmontage	28
3.3 Ablaufplanung bei der Festplattenmontage	30
3.3.1 Erweiterte Produktstruktur	30
3.3.2 Ableitung und Konfiguration der Montageablaufstruktur	31
3.4 Varianten und Versionen	33
3.4.1 Varianten von Produkten	33
3.4.2 Varianten von Montagefunktionslisten	34
3.4.3 Versionen und Verwaltung der erweiterten Produktstruktur	34
3.5 Konzeption einer modularen Montagezelle	35
3.5.1 Anforderungen an eine modulare Montagezelle und ihr Leitsystem	35
3.5.2 Aufbau einer Montagezelle	37
3.5.3 Bildung einer Montageanlage	39
3.5.4 Transport der Produkte auf einem Werkstückträger	41

3.5.5	Verknüpfung von Montageanlagen zu einem Montageanlagenverbund	43
3.6	Abgrenzung der Arbeit und Zusammenfassung der Anforderungen an das Leitsystem	44
Kapitel 4	Stand der Technik	47
4.1	Systemarchitekturen	47
4.1.1	Zentrale Systemarchitektur	47
4.1.2	Hierarchische Systemarchitektur	48
4.1.3	Vollständig verteilte Systemarchitektur	49
4.2	Softwaresysteme	49
4.2.1	Programmierung	49
4.2.2	Konfiguration	50
4.2.3	Modellierungsansätze	51
4.2.4	Discrete Event System	51
4.3	Zusammenfassung	52
Kapitel 5	Entwicklung des konfigurierbaren Leitsystems für Montagezellen	53
5.1	Komponentenbildung	53
5.1.1	Definition einer physikalischen Komponente	53
5.1.2	Klassifizierung von Komponenten	54
5.1.3	Anbindung der Komponenten an das Leitsystem	55
5.2	Standardisierte Schnittstelle für Komponenten	56
5.2.1	Schicht 1: Physikalisches Protokoll	57
5.2.2	Schicht 2: Datentransport	57
5.2.3	Schicht 3: Komponenten-Kommunikations-Protokoll	57
5.2.4	Schicht 4: Komponenten-Kommunikations-Schnittstelle	58
5.2.5	Wrapper zur standardisierten Schnittstelle für Komponenten	60
5.2.6	Vorteile der standardisierten Schnittstelle für Komponenten	61
5.3	Logischer Repräsentant	61
5.3.1	Logische Komponente	61
5.3.2	Zustandsmaschine einer Aufgabe	62
5.3.3	Grundlegender Aufbau des logischen Repräsentanten	62
5.4	Logische Position	64

5.4.1	Definition der logischen Position	64
5.4.2	Logische Positionen bei Werkstückträgern	65
5.4.3	Logische Positionen bei der Be-/Entladekomponente	66
5.4.4	Logische Positionen bei Komponenten allgemein	67
5.4.5	Abbildung der logischen Positionen auf Weltkoordinaten	68
5.4.6	Vorteile der logischen Positionen	69
5.5	Produktverfolgung	69
5.5.1	Definition des logischen Produktes	70
5.5.2	Logische Produktpositionen	70
5.5.3	Zustandsmaschine für eine logische Produktposition	70
5.5.4	Produkttransport zwischen Komponenten	72
5.5.5	Abbildung der logischen Produktposition auf ihre Zustandsmaschine	73
5.5.6	Erweiterung des logischen Repräsentanten durch die Produktverfolgung	74
5.6	Mathematisches Modell der Aufgaben	75
5.6.1	Abbildung einer Komponentenaufgabe durch eine Transformationsmatrix	76
5.6.2	Detaillierung der Transformationsmatrix	77
5.6.3	Transformation bei Fügeaufgaben	79
5.6.4	Gleichsetzung aller Aufgaben	81
5.7	Komponentenauftrag	81
5.8	Zustandsnachrichten	83
5.8.1	Definition einer Zustandsklasse	83
5.8.2	Definition und Erzeugung von Zustandsnachrichten	84
5.8.3	Einfluss der Zustandsnachrichten auf den Komponentenauftrag	85
5.8.4	Rücksetzen der Zustandsnachrichtenanforderungen in einem Komponentenauftrag	87
5.8.5	Abbildung der Zustandsnachrichten auf die logischen Positionen eines Komponentenauftrags	89
5.8.6	Endausbau des logischen Repräsentanten	90
5.8.7	Einschränkung der Zustandsnachrichten	92
5.8.8	Vergleich von Zustandsnachrichten	93
5.9	Konfiguration des logischen Repräsentanten	94

5.9.1	Konfiguration der Komponentenaufträge	95
5.9.2	Konfiguration der Zustandsnachrichtenanforderungen in den Komponentenaufträgen	96
5.10	Zusammenarbeit der logischen Repräsentanten	98
5.10.1	Einführung von Agenten	99
5.10.2	Funktionsweise des Multi-Agenten-Systems	101
5.11	Konfiguration von Gruppen	103
5.11.1	Nicht unterbrechbare Sequenzen von Aufgaben	103
5.11.2	Einführung von Gruppen-Identifizierung	104
5.12	Zusammenfassung	105
Kapitel 6	Realisierung des konfigurierbaren Leitsystems	107
6.1	Aufbau des Prototyps	107
6.1.1	Montagezelle	107
6.1.2	Montageanlage	110
6.2	Konfigurierbares Leitsystem	111
6.2.1	Auswahl des Produktes	112
6.2.2	Konfiguration der Komponentenaufträge	113
6.2.3	Konfiguration der Zustandsnachrichtenanforderungen	115
Kapitel 7	Erprobung und Bewertung des konfigurierbaren Leitsystems	119
7.1	Erweiterter Produktbaum eines Actuators	119
7.2	Konfiguration der Montagezellen	120
7.2.1	Konfiguration der Montagezelle Flex-Fold	122
7.2.2	Konfiguration der Montagezelle Pivot-Assembly	124
7.2.3	Konfiguration der Montagezelle Actuator-Quasi	126
Kapitel 8	Zusammenfassung und Ausblick	127
Kapitel 9	Summary	131
Kapitel 10	Literaturverzeichnis	137

Abkürzungen und Formelzeichen

GByte	Giga Byte
i, j	Indizes
I	Einheitsmatrix
IDE	Integrated Device Equipment
IP	Internet Protocol
h	Anzahl Aufgaben einer Komponente
k	Anzahl der Eingangsprodukte bei Fügeaufgaben
MByte	Mega Byte
MFL	Montagefunktionsliste
OMG	Object Management Group
P	Vektor eines Eingangsproduktes
PC	Personal Computer
SCSI	Small Computer System Interface
SPS	Speicherprogrammierbare Steuerung
Q	Vektor eines Ausgangsproduktes
R, S, T	Transformationsmatrizen
SATA	Serial Advanced Technology Attachment
t_{ACK}	Zeit für das Warten auf Bestätigung
TCP	Transmission Control Protocol
t_{RES}	Zeit für das Warten auf Antwortdaten
UML	Unified Modeling Language
ZK	Zustandsklasse
ZN	Zustandsnachricht
ZNA	Zustandsnachrichtenanforderung

Kapitel 1

Einleitung

1.1 Problemstellung

Für Unternehmen, die sich mit der Herstellung von Festplatten beschäftigen, ist es zunehmend wichtiger, neue Produkte in immer kürzeren Intervallen zu erstellen, um sich längerfristig damit am Markt behaupten zu können. Hierbei spielt vor allem auch eine große Rolle, eine vielfältige Palette von Festplatten zu haben, um die Anforderungen heutiger Computeranlagen in allen Bereichen erfüllen zu können [IBM02a, Seagate03]. Die Produktpalette hat sich in den letzten Jahren, zusätzlich zu den klassischen Einsatzgebieten für Festplatten (Desktop Computer, Server und Mobilgerät), um Festplatten für den so genannten Unterhaltungselektronikmarkt (engl.: Consumer Electronics), zu dem Spielkonsolen, Navigationssysteme, Digitalkameras und vieles mehr gehören, erweitert [Donovan03].

Obwohl sich inzwischen andere Techniken für die Speicherung von Daten, wie zum Beispiel Flash-Speicherkarten, verbreiten, steigen die Verkaufszahlen für Festplatten in den klassischen Bereichen Desktop Computer, Server und Mobilgerät stetig an (siehe Abb. 1.1). Die Prognosen für die Verkaufszahlen im Unterhaltungselektronikmarkt sind noch deutlich höher [Massengill03]. Die hohen Erwartungen hängen vor allem damit zusammen, dass die Kapazität der Festplatten immer größer wird und einhergehend ihre Baugröße sich ständig verkleinert [Grochowski03, Kozierok01c].

Die Montage erfolgt heute weitestgehend manuell bzw. manuell mit automatischer Zuführung der Einzelteile [Bohn99, IBM02b]. Dies führt zu Fehlern bei der Montage selbst als auch zu partikulären und elektrostatischen Kontaminationen, wodurch die Festplatte sofort oder nach kurzer Betriebsdauer unbrauchbar wird. Vorhandene ganzheitliche Automatisierungsansätze sind inzwischen zwar in der Lage, die Montage einer Festplatte vollautomatisch durchzuführen und sogar den Wechsel auf andere Produkte zu ermöglichen [Seagate01, Seagate02]. Dennoch dauert der Wechsel auf ein

anderes Produkt mehrere Stunden und die gleichzeitige Montage von verschiedenen Produkten ist nur schwer möglich.

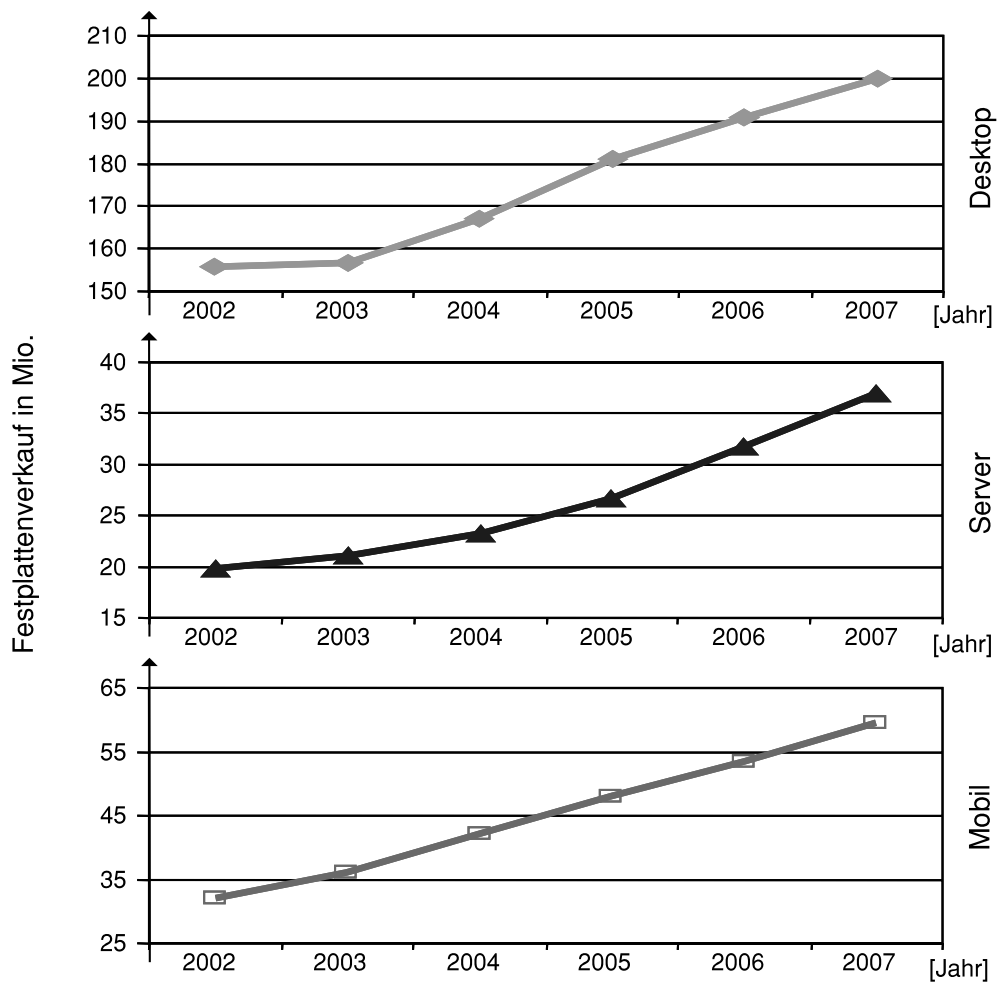


Abb. 1.1: Überblick der Entwicklung des Festplattenverkaufs getrennt nach Desktop, Server und Mobilgeräten [Massengill03]

Um die Ansprüche, welche zum einen durch die Miniaturisierung und Kapazitätserhöhung und zum andern durch die steigende Produktvielfalt und die steigenden Verkaufszahlen an die Montage von Festplatten gestellt werden, erfüllen zu können, bedarf es einer neuen vollautomatischen Montagezelle, welche flexibel an neue Produkte, an kürzere Montagezeiten und an steigende Produktionszahlen angepasst werden kann. Um eine Montagezelle vollautomatisch und flexibel steuern zu können, bedarf es eines konfigurierbaren Leitsystems.

1.2 Zielsetzung und Vorgehensweise

In dieser Arbeit wird ein solches konfigurierbares Leitsystem für eine Montagezelle entwickelt. Ziel ist es, dass das Leitsystem in der Lage ist, die Montage verschiedener Produkte zur gleichen Zeit durchzuführen und auch einfach für neue Produkte konfiguriert werden kann. Darüber hinaus soll es möglich sein, das Leitsystem durch Konfiguration auch für neue Montagefunktionalitäten zu benutzen.

Um die grundlegenden Anforderungen, welche an eine neue Montagezelle gestellt werden, abzuleiten, wird auf die Ausgangssituation bei der Montage von Festplatten eingegangen.

Nachdem die Montage und speziell der Montageablauf analysiert sind, wird eine modulare Montagezelle konzipiert, für die das konfigurierbare Leitsystem später entwickelt wird. Sowohl aus dem Montageablauf als auch aus dem Konzept der modularen Montagezelle ergeben sich die Anforderungen, die ein konfigurierbares Leitsystem für eine solche Montagezelle zu erfüllen hat.

Danach wird der Stand der Technik von verschiedenen Leitsystemen betrachtet. Dabei wird nicht nur die Festplattenmontage herangezogen, sondern auch allgemeine Techniken für die Steuerung von Montagezellen.

Die Entwicklung des konfigurierbaren Leitsystems für die konzipierte modulare Montagezelle wird sukzessive hergeleitet.

Eine prototypische Realisierung einer Montagezelle sowie die Erprobung und Bewertung des konfigurierbaren Leitsystems für diese Montagezellen zeigt die Möglichkeiten des entwickelten Leitsystems.

Abschließend folgt eine Zusammenfassung und ein Ausblick über weitere Entwicklungsmöglichkeiten im Bereich des konfigurierbaren Leitsystems für modulare Montagezellen.

Kapitel 2

Ausgangssituation bei der Festplattenmontage

Festplatten dienen heute nicht nur in Computern als Massenspeicher, sondern vermehrt auch in Geräten der Unterhaltungselektronik (vgl. Abschnitt 1.1). Um den Anforderungen dieser Märkte gerecht zu werden, bedarf es einer stetigen Entwicklung immer neuer Produkte. Diese Entwicklungen führen aber auch zu einem höheren Anspruch bei der Montage. Um dies zu verdeutlichen, wird zuerst der Aufbau einer Festplatte beschrieben, bevor auf die Entwicklung und die Trends bei Festplatten eingegangen wird.

2.1 Aufbau einer Festplatte

Eine heutige Festplatte besteht aus mehreren flachen, runden Scheiben, welche *Platten* genannt werden. Die Platten bestehen aus Glas oder Metall und sind magnetisch beschichtet [Karbo01b, Kilian03]. Sie besitzen in ihrer Mitte ein Loch, über das sie auf einer *Spindel* gestapelt werden. Die Spindel wird über den *Spindelmotor* angetrieben und somit auch der Plattenstapel [Kozierok01b]. Um Informationen auf die Platten zu schreiben oder von ihnen zu lesen, werden Schreib-/Leseköpfe benutzt. Diese sind auf einem *Slider* montiert. Der Slider ist so gebaut, dass durch die unter ihm schnell rotierenden Platten ein Luftpolster von einigen Nanometern zwischen Schreib-/Lesekopf und Platte entsteht [Kilian03, Kursawe98, Schuster04]. Mehrere dieser Slider sind auf einen *Actuator Arm* montiert. In den Actuator Arm ist ein Drehzapfen, das *Pivot*, eingelassen. Auf der gegenüberliegenden Seite der montierten Slider befindet sich eine Spule, welche sich in einem Magnetfeld befindet, so dass durch eine gezielte Ansteuerung der Actuator Arm, und somit die Schreib-/Leseköpfe, aufgrund der Drehachse am Pivot auf den Platten positioniert werden kann [Kozierok01b, Schuster04]. Diese Technik wird *Voice Coil* genannt. Im Gegensatz zu den früher eingesetzten Schrittmotoren zur Positionierung des Actuator Arms, mit denen nur

diskrete Positionen angefahren werden konnten, lassen sich mit der Voice Coil Technik beliebige Positionen anfahren [Kozierok01e, Kozierok01h]. Der Actuator Arm mit Pivot und Spule wird *Actuator* genannt. Die elektrische Ansteuerung der Köpfe auf dem Actuator geschieht über ein flexibles elektrisches Band, welches direkt am Actuator angebracht ist [Kozierok01a, Schuster04]. Alle Teile, Actuator, Spindel mit Motor und Platten, werden in das Basisgehäuse eingebaut und mit dem Deckel in einer reinen Umgebung verschlossen [Bohn99, Kozierok01b, Kursawe98, Seagate01]. Zusätzlich wird eine elektronische Schaltung an dem Gehäuse angebracht, welche den Spindelmotor antreibt, die Positionierung des Actuators auf den Platten vornimmt und die Schreib-/Leseköpfe ansteuert [IBM02c, Karbo01a, Kozierok01a, Kozierok01b, Kursawe98, Schuster04]. Die elektronische Schaltung bietet darüber hinaus auch eine standardisierte Schnittstelle (heute übliche Schnittstellen sind IDE, SATA und SCSI) der Ansteuerelektronik nach außen an [Bär03, Karbo01b]. Abbildung 2.1 zeigt eine schematische Darstellung des Aufbaus einer modernen Festplatte.

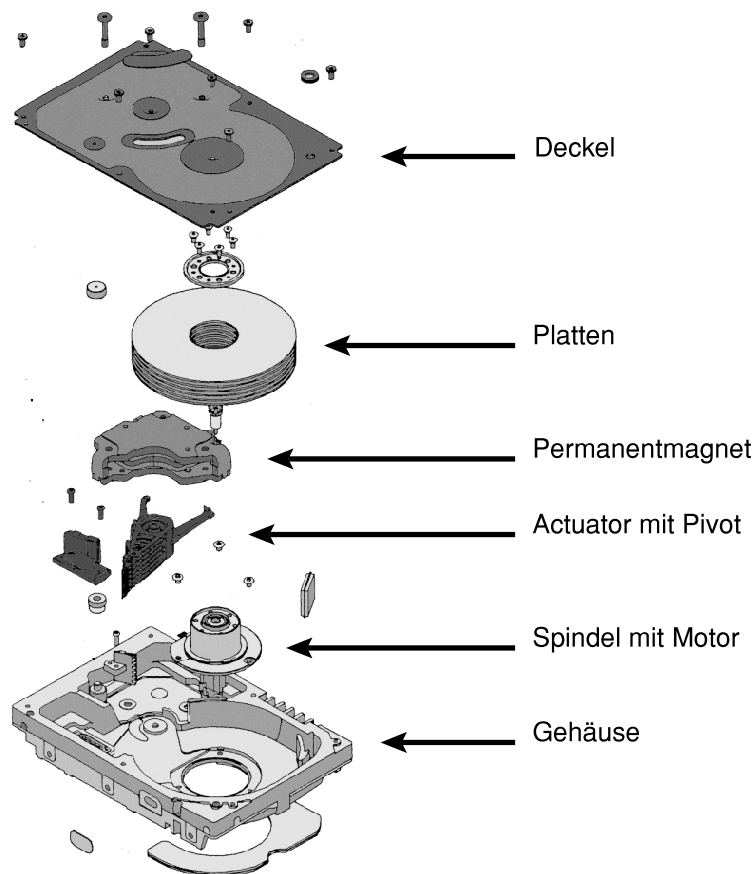


Abb. 2.1: Aufbau einer Festplatte (© IBM Corporation)

2.2 Entwicklung und Trends bei Festplatten

Obwohl die erste Festplatte schon 1956 von IBM entwickelt wurde [Idema01, Kozierok01c, Schuster04], hat ihre große Verbreitung erst mit der Einführung des ersten Personal Computers von IBM, dem IBM XT, im Jahre 1983 begonnen [Karbo01b, Kozierok01d]. Seit den Anfängen wird die Festplatte stetig weiterentwickelt. Hatte sie anfangs noch eine Baugröße von 24" und eine Kapazität von ungefähr 5 MByte [Kilian03, Kursawe98, Rostky98, Schuster04, Wattrodt98], so haben heutige Festplatten, welche üblicherweise in Desktop Computern und Servern zum Einsatz kommen, eine Baugröße von 3,5" [EETimes03, Kozierok01d, Kursawe98] und Kapazitäten mit bis zu 250 GByte [Seagate04, WesternDigital04]. Darüber hinaus sind, vornehmlich für den mobilen Bereich, 2,5" und 1,8" Festplatten mit bis zu 80 GByte erhältlich. Auch 1" Festplatten, welche für den Unterhaltungselektronikmarkt entwickelt wurden, haben inzwischen eine Kapazität von bis zu vier GByte (siehe Abb. 2.2). Sowohl die Miniaturisierung als auch die Verbreitung der Festplatte führen bei der Montage zu neuen Anforderungen. Waren es in den Anfängen meist Einzelanfertigungen oder Kleinstserien, welche aufgrund ihrer Größe noch einfach manuell zu montieren waren, so ist die Festplatte inzwischen zu einem Massenprodukt geworden, deren Verkaufszahl jährlich steigt (vgl. Abschnitt 1.1).

Jahr	Entwicklungsschritt
1956	Erste Festplatte von IBM (24" und etwa 5 MByte)
1980	Erste 5,25" Festplatte
1983	Erster IBM XT Personal Computer
1983	Erste 3,5" Festplatte
1986	Erste 3,5" Festplatte mit Voice Coil Technik
1986	Erste 3,5" Festplatte mit einer Bauhöhe von 1"
1988	Erste 2,5" Festplatte
1991	Erste 1,8" Festplatte
1992	Erste 1,3" Festplatte
1998	Erste 1" Festplatte von IBM (Microdrive) mit 340 MByte
2004	1" Festplatten mit bis zu 4 GByte 3,5" Festplatten mit bis zu 250 GByte

Abb. 2.2: Einige Meilensteine bei der Entwicklung der Festplatte

Einhergehend mit der Miniaturisierung der Festplatten hat sich auch stetig ihre Kapazität erhöht. Dies wird durch immer höhere Speicherdichten auf den Platten selbst erreicht (siehe Abb. 2.3). Inzwischen steigt die Speicherdichte jährlich um 100% [Kilian03]. Eine Erhöhung der Speicherdichte zieht eine Verkleinerung der Schreib-/Leseköpfe und eine Verringerung der Flughöhen nach sich, was wiederum eine präzisere Montage des Actuators erfordert.

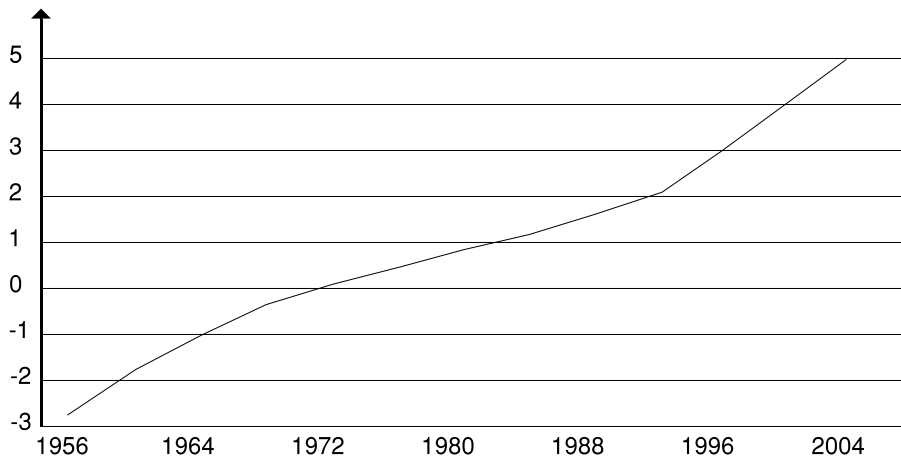


Abb. 2.3: Entwicklung der Speicherdichte in 10^x Megabit pro Quadratzoll bei Festplatten

Neben der kleineren Baugröße und der höheren Kapazität hat sich auch die Leistung der Festplatte in den letzten Jahren stark erhöht. Eine moderne 3,5" Festplatte bietet eine mittlere Zugriffszeit kleiner als 5 ms und eine maximale Datenübertragungsrate von bis zu 75 MByte in der Sekunde [Seagate04]. Dies wird vor allem durch die hohen Rotationsgeschwindigkeiten der Platten (bis zu 15.000 Umdrehungen in der Minute) erreicht. Dies hat wiederum höhere Ansprüche sowohl an die Bauteile als auch an deren Montage zur Folge.

Der Trend bei den Festplatten geht hin zu höheren Kapazitäten und Leistungen, auch für die 2,5", die 1,8" und die 1" Größen. Dadurch kann die 2,5" Festplatte zum Standard im Desktop Bereich werden und somit die 3,5" Festplatte verdrängen. Im mobilen Bereich wird die 1,8" Festplatte die 2,5" Festplatte als Standard ablösen [Donovan03]. Dies bedeutet natürlich auch, dass die Genauigkeit und die Geschwindigkeit bei der Montage der Festplatten gesteigert werden müssen.

2.3 Aktuelle Montagevarianten von Festplatten

Die Herstellung von Festplatten fordert eine hohe Genauigkeit von zuvor gefertigten Einzelteilen, welche zu einer Festplatte montiert werden. Es sind Montagefunktionen vorhanden, welche eine Positioniergenauigkeit der zu montierenden Einzelteile von 1-2 μm erfordern [IBM03a]. Darüber hinaus muss die Endmontage der Platten und der Schreib-/Leseköpfe in einer reinen Umgebung stattfinden, welche der Luftreinheitsklasse 3 [VDI2083_1] entspricht, da die Schreib-/Leseköpfe während des Betriebes der Festplatte in einem Abstand von nur wenigen Nanometern über den Platten schweben. Schon ein menschliches Haar, welches eine Größe von 60-100 μm hat, kann

die Schreib-/Leseköpfe im Betrieb zerstören [Bohn99, Karbo01a, Kilian03, Kozierok01b, Schuster04].

- **Manuelle Montage**

Bei dieser Art der Montage werden viele Montageschritte manuell, das heißt von einem Menschen, durchgeführt [Bohn99, IBM02b]. Um die benötigte Positioniergenauigkeit von 1-2 µm zu erzielen, stehen sowohl Einspannhilfen als auch optische und optoelektronische Hilfsmittel zur Verfügung. Die Montage selbst findet innerhalb eines Reinraumes statt, so dass die Personen, welche die Montage durchführen, reinraumtaugliche Schutzkleidung tragen müssen [VDI2083_6]. Darüber hinaus muss das montagedurchführende Personal einen Potentialausgleich mit den benutzten Hilfsmitteln und Werkzeugen und den zu montierenden Einzelteilen durchführen, um speziell den sehr empfindlichen Schreib-/Lesekopf nicht zu beschädigen. Der Transport der Einzelteile und der montierten Komponenten findet mit speziellen Transportbehältnissen durch Personen statt. Sowohl die Kontamination als auch die elektrostatischen Beschädigungen an den Schreib-/Leseköpfen durch das Personal führen zu einem Ausschuss von etwa 15-20% [IBM02d].

- **Halbautomatische Montage**

Die halbautomatische Montage baut auf der manuellen Montage auf. Hierbei werden die Transporte der Einzelteile und der montierten Komponenten nicht mehr durch Personen, sondern von Transportbändern zwischen den Montagestationen durchgeführt. Durch die Transportbänder wird nicht nur der Transport zwischen den Montagestationen automatisiert, sondern auch ein Takt für die Montage an den einzelnen Stationen vorgegeben. Dadurch konnte der Durchsatz im Gegensatz zur manuellen Montage um etwa 5-10% gesteigert werden [IBM02b, IBM02d]. Der Ausschuss hat sich bei dieser Art der Montage allerdings nicht verändert.

- **Vollautomatische Montage**

Vollautomatische Montagesysteme kommen nur sehr selten zum Einsatz, da automatisierte Systeme schwieriger an neue Festplattenmodelle angepasst werden können [Bohn99]. Es gibt jedoch inzwischen auch vollautomatische Anlagen, bei denen der Mensch nicht mehr die Montage selbst durchführt, sondern darauf achtet, dass die Montageanlage ununterbrochen und fehlerfrei funktioniert [Seagate01]. Eine solche Anlage ist in der Lage, beliebige

Festplatten zu montieren. Der Wechsel zwischen zwei verschiedenen Produkten dauert auf einer solchen Anlage einige Stunden [Seagate02]. Dennoch ist eine solche vollautomatische Montageanlage in der Lage, jeden Festplattentyp schneller und mit einer höheren Qualität herzustellen [Seagate03a]. In der Literatur finden sich nur wenige Installationen von solchen vollautomatischen Anlagen für die Festplattenmontage.

2.4 Grundlegende Anforderungen an eine neue Festplattenmontage

Wird der komplexe Aufbau einer Festplatte im Zusammenhang mit der Entwicklung und den Trends bei Festplatten betrachtet, so zeigt sich, dass die aktuelle Montage von Festplatten verbessert werden muss [IBM02b]. Für eine neue Montage von Festplatten ist die Verbesserung von folgenden Kriterien gefordert [IBM02d, Khan03, Seagate03b]:

- **Kosten**
 - Personal
 - Reinraum
- **Durchsatz**
- **Zuverlässigkeit und Robustheit**
- **Produktvielfalt und Erweiterbarkeit**
- **„Time to market“**
- **„Total Quality Assurance“**

Diese Kriterien können durch eine neue vollautomatische Montagezelle erzielt werden, welche in der Lage ist, ohne den Eingriff von menschlichem Personal eine Festplatte zu montieren.

Kapitel 3

Analyse des Montageablaufs und Konzeption einer Montagezelle

Um zu einer vollautomatischen Montagezelle zu gelangen, sind fundierte Kenntnisse über die Montage von Festplatten sowie deren Ablauf von entscheidender Bedeutung. Aus dieser Analyse kann das Konzept einer Montagezelle sowie die Anforderungen an das steuernde Leitsystem dieser Montagezelle abgeleitet werden.

3.1 Grundlagen der Ablaufplanung

Die Ablaufplanung besteht aus der Strukturierung des Produktes und der Strukturierung des Montageablaufs [Bullinger86].

Produktstruktur

Jedem Produkt liegt eine Struktur ihres Aufbaus zugrunde [DIN6789]. Dieser Strukturaufbau wird Produktstruktur genannt. Hierbei werden in mehreren Ebenen Einzelteile zu Baugruppen und diese schließlich zum endgültigen Produkt montiert. Die Anzahl der Ebenen einer solchen Produktstruktur kann dabei beliebig groß sein. Ein Produkt oder eine Baugruppe ist dabei nur aus Baugruppen oder Einzelteilen der direkt darunter liegenden Ebene zusammengesetzt. Jede Baugruppe und jedes Einzelteil hat genau einen Nachfolger in der ihm übergeordneten Ebene. Wird eine Baugruppe oder ein Einzelteil in mehreren Ebenen benötigt, so wird es mehrfach dargestellt (siehe Abb. 3.1).

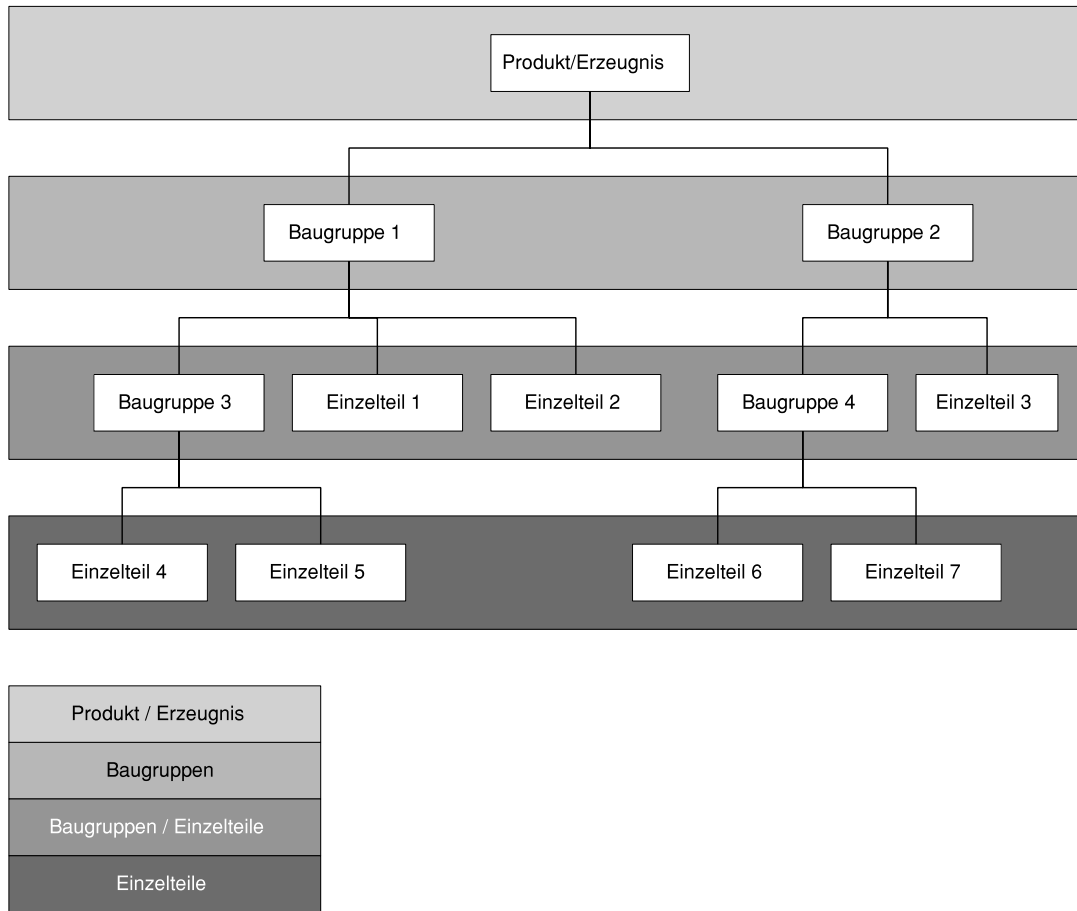


Abb. 3.1: Aufbau eines Produktes durch eine Produktstruktur

Auch Baugruppen können als Produkte und ihr Strukturaufbau als Produktstruktur bezeichnet werden. Einzelteile werden im weiteren Verlauf auch als Produkte bezeichnet. Im Gegensatz zu den Baugruppen und zu dem endgültigen Produkt besitzen diese allerdings keine Produktstruktur. Um die Abhängigkeit eines Produktes von seinen untergeordneten Produkten zu verdeutlichen, werden die untergeordneten Produkte auch als Teilprodukte bezeichnet.

Montageablaufstruktur

Die Montageablaufstruktur eines zu montierenden Produktes zeigt die logische und zeitliche Aufeinanderfolge von Teilfunktionen der Montage, welche schließlich zum endgültigen Produkt führen. Die Montageablaufstruktur beschreibt die eigentliche Montagearbeit [Angerer03]. Obwohl es in der Literatur mehrere mögliche Darstellungsmöglichkeiten gibt, ist der Montagevorranggraph eine der übersichtlichsten und am weitesten verbreitete Darstellungsform [Bullinger86, Delchambre92]. Der Montagevorranggraph besteht aus Knoten und Kanten (siehe Abb. 3.2). Die

Knoten repräsentieren hierbei Teilfunktionen, welche Teilverrichtungen genannt werden. Die Kanten zwischen den Knoten bezeichnen die Abhängigkeitsbeziehungen, und zwar die logischen und zeitlichen, zwischen diesen Knoten [Bullinger86].

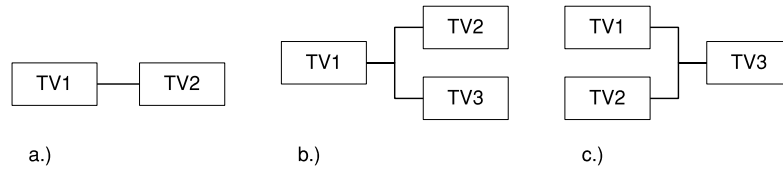


Abb. 3.2: a.) einfache Struktur b.) Struktur mit Verzweigung c.) parallele Struktur mit Zusammenführung

Es gibt in einem Montagevorranggraphen drei wesentliche Strukturmerkmale. Die einfache Struktur (siehe Abb. 3.2 a) bedeutet, dass die Teilverrichtung 2 nach der Teilverrichtung 1 auszuführen ist. Die Struktur mit Verzweigung (siehe Abb. 3.2 b) bedeutet, dass die Teilverrichtungen 2 und 3 nach der Teilverrichtung 1 auszuführen sind. In dieser Form können beliebig viele Teilverrichtungen parallel nach Teilverrichtung 1 dargestellt werden. Diese Struktur ist nicht auf zwei Teilverrichtungen begrenzt. Bei der parallelen Struktur mit Zusammenführung (siehe Abb. 3.2 c) werden die Teilverrichtungen 1 und 2 parallel ausgeführt und anschließend zur Teilverrichtung 3 zusammengeführt. Auch hier ist die parallele Struktur nicht auf zwei Teilverrichtungen begrenzt. Bei allen parallelen Strukturen können die Teilverrichtungen parallel ausgeführt werden, insofern keine Abhängigkeiten von Produkten zwischen den Teilverrichtungen existieren. Wenn solche Abhängigkeiten existieren, so bedeutet die parallele Struktur die Wahlfreiheit bei der Ausführungsreihenfolge [Bullinger86].

Mit Hilfe dieser drei Strukturmerkmale lassen sich komplexe Montageabläufe in einer übersichtlichen Struktur darstellen (siehe Abb. 3.3).

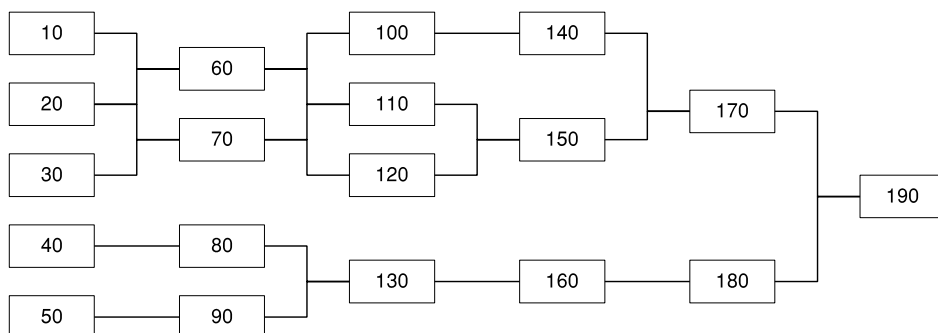


Abb. 3.3: Montagevorranggraph mit den Teilverrichtungen 10 bis 190

3.2 Montagefunktionen bei der Festplattenmontage

Die Teilverrichtungen eines Montagevorranggraphen sind Teilfunktionen, welche bei der Montage eines Produktes an dessen Teilprodukten vorgenommen werden. Diese Teilfunktionen werden Montagefunktionen genannt [Bullinger86]. Die Analyse der Montage von Festplatten führt dabei zur Einteilung in folgende Montagefunktionen [IBM02e, Schraft03]:

- *Fügen:* Beim Fügen entsteht aus zwei oder mehreren Teilprodukten genau ein neues Produkt. Es gibt verschiedene Fügearten bei der Montage von Festplatten, wie zum Beispiel Schrauben, Löten oder Klemmen. In der Produktstruktur besteht das neue durch Fügen entstehende Produkt aus zwei oder mehr Teilprodukten der direkt untergeordneten Ebene.
- *Sonderfunktionen:* Sonderfunktionen nehmen Veränderungen an einem Teilprodukt vor, so dass dieses zu einem neuen Produkt wird. In der Produktstruktur besteht das neue Produkt aus genau einem Teilprodukt der direkt untergeordneten Ebene.
- *Reinigen:* Reinigen ist eine Spezialisierung einer Sonderfunktion. Obwohl beim Reinigen keine direkten Veränderungen an einem Teilprodukt stattfinden, wird ein gereinigtes Teilprodukt als ein neues Produkt angesehen. In der Produktstruktur besteht das gereinigte Produkt aus genau diesem einen Teilprodukt der direkt untergeordneten Ebene.
- *Optisches Messen:* Das optische Messen ist eine Montagefunktion, welche durch eine Messung mit einem Kamerasystem aus einem Teilprodukt einer untergeordneten Ebene der Produktstruktur ein neues Produkt erzeugt. Das neue Produkt entsteht dabei aus genau einem Teilprodukt in der Produktstruktur.
- *Elektrisches Messen:* Beim elektrischen Messen wird durch eine elektronische Schaltung ein Teilprodukt vermessen und somit zu einem neuen Produkt. Das neue Produkt besteht aus genau diesem einen Teilprodukt aus der direkt untergeordneten Ebene der Produktstruktur.
- *Justieren:* Durch Einstellungen oder Abstimmungen kann ein Teilprodukt in ein neues Produkt überführt werden, wobei das justierte Produkt aus genau einem Teilprodukt der direkt untergeordneten Ebene in der Produktstruktur besteht.

- *Handhabung*: Handhabungen sind Transportvorgänge von Produkten oder Teilprodukten. Handhabungen sind nicht in der Produktstruktur abgebildet. Sie sind ausschließlich in einem Montagevorranggraphen abbildbar.

Die Analyse der eingesetzten Montagefunktionen bei der Festplattenmontage führt zur Einteilung in zwei Grundtypen, der Primärmontage und der Sekundärmontage. Primärmontage sind Vorgänge, welche die Wertschöpfung eines Produktes erhöhen, während Sekundärmontage die Wertschöpfung eines Produktes nicht erhöht [Lotter82].

Darüber hinaus müssen die bei der Festplattenmontage zum Einsatz kommenden Montagefunktionen bezüglich ihrer Ausführungszeit analysiert werden. Bei dieser Ausführungszeitanalyse ist nicht nur die Betrachtung der Ausführungszeiten für unterschiedliche Produkte wichtig, sondern auch die Varianz der Ausführungszeiten für gleiche Produkte. Abbildung 3.4 zeigt die Einteilung der Montagefunktionen in die Grundtypen sowie die Ausführungszeiten für unterschiedliche und die Ausführungszeitvarianz für gleiche Produkte.

Montagefunktion	Grundtyp	Ausführungszeit (unterschiedliche Prod.)	Ausführungszeitvarianz gleicher Produkte
Fügen	Primär	10 - 60 s	0 - 2 s
Sonderfunktionen	Primär	30 - 180 s	0 - 60 s
Reinigen	Primär	30 - 300 s	0 - 2 s
Optisches Messen	Primär	0 - 20 s	0 - 5 s
Elektrisches Messen	Primär	5 - 60 s	0 - 55 s
Justieren	Primär	30 - 60 s	0 - 30 s
Handhabung	Sekundär	1 - 10 s	0 - 1 s

Abb. 3.4: Analyse der Typeneinteilung und Ausführungszeiten der Montagefunktionen

Die Ablaufsteuerung bei der Montage von Festplatten wird von einem Leitsystem durchgeführt. Die hierbei zum Einsatz kommenden Montagefunktionen können durch einen Montagevorranggraphen dargestellt werden. Die Analyse der Ausführungszeiten und insbesondere der Varianzen bei den Ausführungszeiten zeigt jedoch, dass das eingesetzte Leitsystem, um eine effiziente Steuerung der Montage zu erzielen, flexibel auf das Ende einer Montagefunktion reagieren muss. Das Leitsystem kann nicht von einer konstanten Ausführungszeit für die einzelnen Montagefunktionen ausgehen, noch nicht einmal bei den gleichen Produkten.

3.3 Ablaufplanung bei der Festplattenmontage

3.3.1 Erweiterte Produktstruktur

Zwischen der Produktstruktur und der Montageablaufstruktur in Form des Montagevorranggraphen besteht keine Verbindung. Die Verbindung der einzelnen Teilverrichtungen des Montagevorranggraphen zu den Produkten beziehungsweise Teilprodukten der Produktstruktur kann durch Tätigkeitsknoten erzielt werden [Bullinger86]. Die Verbindung muss allerdings manuell durchgeführt werden. Eine solche manuelle Zuordnung der Produkte zu den einzelnen Teilverrichtungen ist jedoch fehleranfällig. Eine Analyse der Zuordnung von (Teil-) Produkten der Produktstruktur zu den Teilverrichtungen des Montagevorranggraphen hat ergeben, dass für einen automatisierten Lösungsansatz die Montagefunktionen, aus denen später die Teilverrichtungen des Montagevorranggraphen gebildet werden, den einzelnen (Teil-) Produkten zugeordnet werden sollen [IBM02d].

Die Produktstruktur wird deshalb in jedem (Teil-) Produkt durch eine Montagefunktionsliste (MFL) erweitert. Diese Montagefunktionen sind durchzuführen, um das (Teil-) Produkt zu montieren, an welches die Liste der Montagefunktionen angehängt ist [Kobusch03]. Bei der erweiterten Produktstruktur, welche als Baum dargestellt ist, in dem das endgültige Produkt die Wurzel und die Einzelteile die Blätter darstellen, ist jedem (Teil-) Produkt, mit Ausnahme der Blätter, eine Liste von Montagefunktionen angehängt (siehe Abb. 3.5).

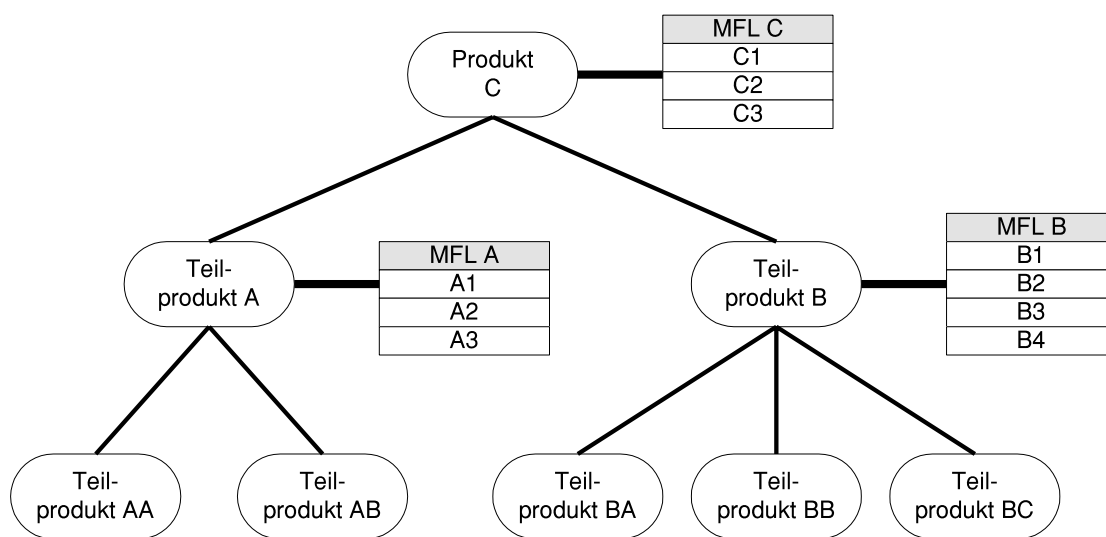


Abb. 3.5: Erweiterte Produktstruktur mit Liste von Montagefunktionen an den (Teil-) Produkten

Die Analyse der Festplattenmontage hat ergeben, dass aufgrund der Komplexität einer Festplatte (vgl. Abschnitt 2.1) und der sich daraus ergebenden Anzahl von etwa 200 Primärmontagefunktionen eine Trennung in Produktentwicklung und Prozessentwicklung stattfinden muss [IBM02c].

Bei der Produktentwicklung wird die erweiterte Produktstruktur von einem oder mehreren Produktentwicklern für ein Produkt erstellt. Diese haben ausschließlich Kenntnisse über den Aufbau des Produktes und über grundlegende Primärmontagefunktionen, welche nötig sind, um das Produkt zu montieren. Die Produktentwickler haben keine Kenntnisse über die Montagezelle und somit auch nicht über alle durchzuführenden Montagefunktionen. Insbesondere kennen sie die Sekundärmontagefunktionen nicht. Da weder die Sekundärmontagefunktionen noch alle Primärmontagefunktionen den Produktentwicklern bekannt sind, ist auch die Reihenfolge der von ihnen vorgegebenen Montagefunktionen in der erweiterten Produktstruktur nicht vorgegeben.

Bei der Prozessentwicklung werden die Listen der Montagefunktionen in der erweiterten Produktstruktur durch alle Primärmontagefunktionen und Sekundärmontagefunktionen erweitert, welche für die automatisierte Montage in einer Montagezelle nötig sind.

3.3.2 Ableitung und Konfiguration der Montageablaufstruktur

Die Erweiterung der Produktstruktur durch Listen von Montagefunktionen führt noch nicht zu einem Montagevorranggraphen, welcher von einer Montagezelle umgesetzt werden kann. Die erweiterte Produktstruktur führt zu einem Montagevorranggraphen, bei dem jede Teilverrichtung einem (Teil-) Produkt entspricht und aus der Liste der Montagefunktionen dieses (Teil-) Produktes besteht (siehe Abb. 3.6).

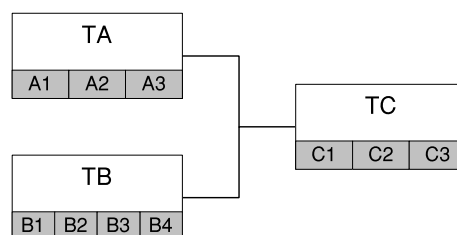


Abb. 3.6: Aus einer erweiterten Produktstruktur abgeleiteter Montagevorranggraph mit den Listen der Montagefunktionen in den Teilverrichtungen TA, TB und TC

Um einen vollständigen Montagevorranggraphen zu erhalten, welcher für eine automatisierte Montagezelle benutzt werden kann, müssen durch die Prozessentwicklung nicht nur die Listen der erweiterten Produktstruktur mit allen Primär- montagefunktionen und Sekundär- montagefunktionen, welche für die Montage des entsprechenden (Teil-) Produktes benötigt werden, erweitert werden, sondern es muss für jede aus der erweiterten Produktstruktur abgeleiteten Teilverrichtung ein Montage- vorranggraph für die darin enthaltenen Montagefunktionen erstellt werden (siehe Abb. 3.7). Daraus folgt, dass das Leitsystem, welches die automatisierte Montagezelle steuert, konfigurierbar sein muss. Die Konfiguration erfolgt durch einen Prozess- entwickler, welcher detaillierte Kenntnisse über die Montagezelle hat und somit alle für die Montage eines (Teil-) Produktes benötigten Montagefunktionen und deren Montagevorranggraphen kennt. Die so erstellte Konfiguration wird als Erweiterung an die Montagefunktionsliste des konfigurierten (Teil-) Produktes angehängt. Die Erstellung der erweiterten Produktstruktur durch die Produktentwickler wird nicht in einem konfigurierbaren Leitsystem durchgeführt, sondern in einem System unabhängig von der Montagezelle [IBM02c].

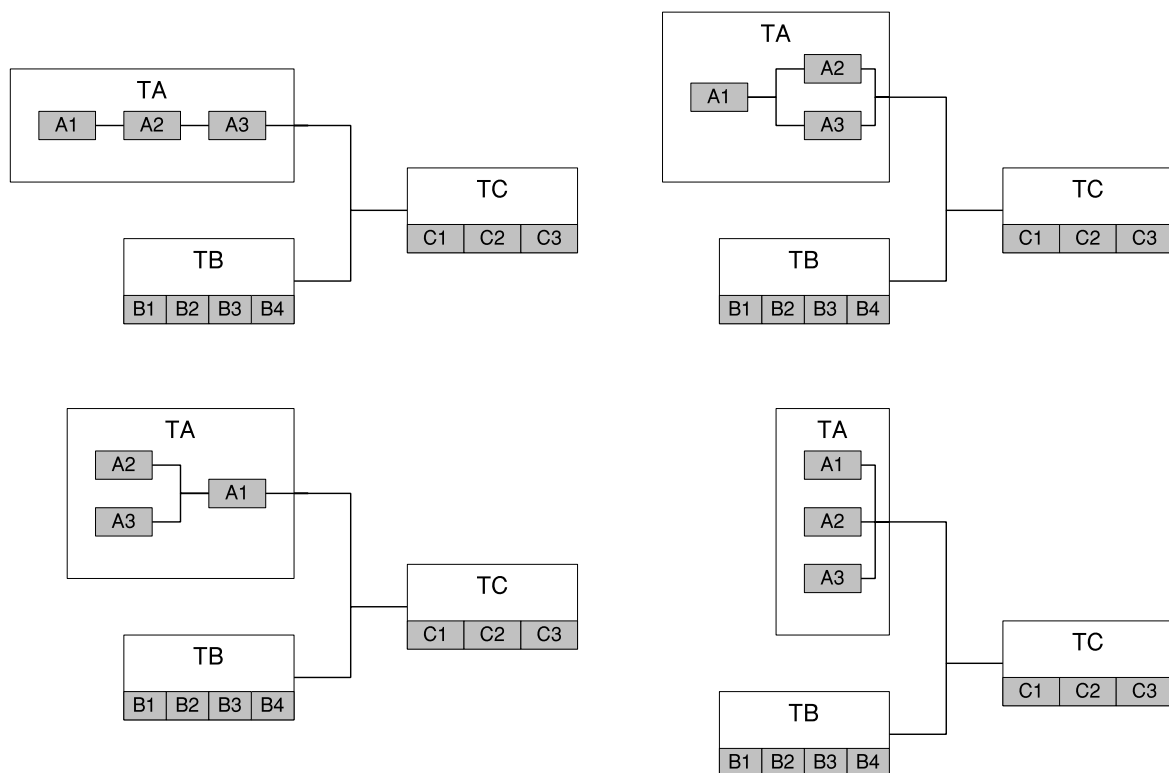


Abb. 3.7: Mögliche Konfigurationen von Montagevorranggraphen für die Teilverrichtung TA mit ihren Montagefunktionen A1, A2 und A3 ohne deren Permutationen

3.4 Varianten und Versionen

Die erweiterte Produktstruktur und der daraus abgeleitete, beziehungsweise konfigurierte, Montagevorranggraph ist nur für ein Produkt und für eine Montagezelle geeignet. Da aber die Produktpalette und auch die Stückzahlen der einzelnen Produkte wächst (vgl. Abschnitt 1.1 und Abschnitt 2.2), müssen Varianten des Produktes verfügbar sein [IBM02a].

3.4.1 Varianten von Produkten

Um Varianten von Produkten, das heißt unterschiedliche Festplatten, montieren zu können, wird in der erweiterten Produktstruktur jedes (Teil-) Produkt durch Varianten erweitert [Kobusch03]. Jede dieser Varianten besitzt wiederum eine Liste von Montagefunktionen (siehe Abb. 3.8).

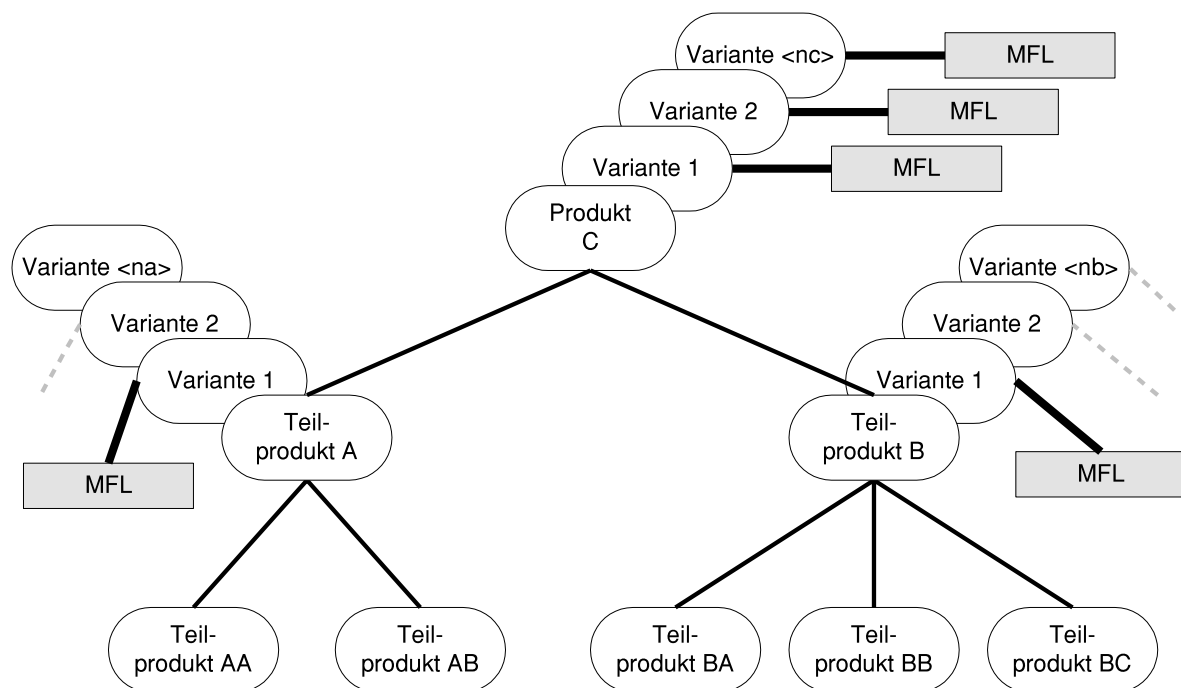


Abb. 3.8: Erweiterte Produktstruktur mit Varianten für jedes (Teil-) Produkt

Um die erweiterte Produktstruktur für eine konkrete Variante eines Produktes, einer Festplatte, zu erhalten und sie somit auf einer automatisierten Montagezelle montieren zu können, wird für jedes (Teil-) Produkt eine Variante ausgewählt. Die so ausgewählten Varianten bilden eine erweiterte Produktstruktur für ein konkretes Produkt. Durch die Variantenbildung für jedes (Teil-) Produkt wird eine beliebig große Anzahl von verschiedenen Produkten in einer Struktur vereinigt, womit auf

unterschiedliche konkrete Produktvarianten ein einfacher Zugriff besteht. Die Anzahl der damit abbildbaren Produkte wird nur durch die Kapazität des eingesetzten Rechnersystems beschränkt.

3.4.2 Varianten von Montagefunktionslisten

Ähnlich der Varianten für die (Teil-) Produkte in der erweiterten Produktstruktur lassen sich Varianten auch für die Montagefunktionslisten benutzen. Statt nur eine Liste von Montagefunktionen für jede (Teil-) Produktvariante zu haben, werden auch hier Varianten von Listen benutzt [Kobusch03]. Dies bedeutet, dass für eine konkrete Produktvariante wiederum beliebig viele Varianten von Montagefunktionslisten zur Verfügung stehen (siehe Abb. 3.9). Die Benutzung von Varianten für die Montagefunktionslisten erlaubt die Montage auf unterschiedlichen Montagezellen [IBM03b, Kobusch03].

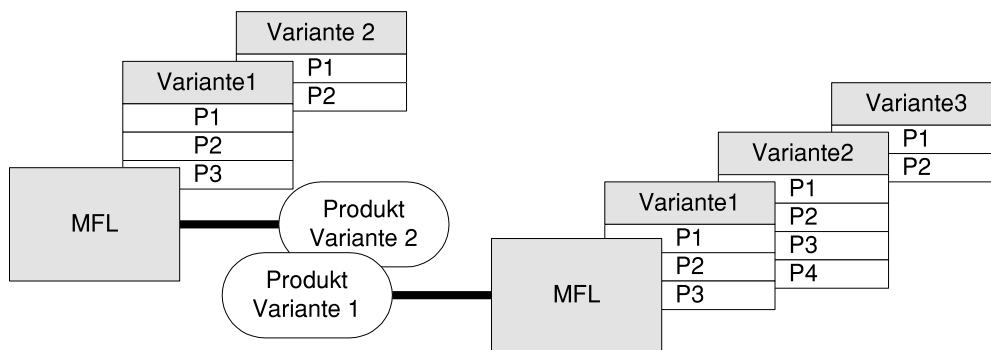


Abb. 3.9: Variantenbildung bei den Montagefunktionslisten

3.4.3 Versionen und Verwaltung der erweiterten Produktstruktur

Festplatten unterliegen oft Weiterentwicklungen, welche nicht als neue Varianten angesehen werden, sondern nur Änderungen vorhandener Varianten darstellen [IBM02c]. Um diese Änderungen in der erweiterten Produktstruktur abzubilden, werden Versionen sowohl für die Varianten der (Teil-) Produkte als auch für die Varianten der Montagefunktionslisten verwendet. Hierbei wird bei einer Änderung einer Variante eine neue Versionsnummer vergeben. Diese neue Version durchläuft einen Freigabezyklus. Erst wenn der Zyklus durchlaufen ist, das heißt die neue Version freigegeben ist, kann diese zum Einsatz kommen. Ist eine neue Version freigegeben, so werden die Vorgängerversionen inaktiv gesetzt, so dass immer nur eine Version einer Variante aktiv ist [Kobusch03].

Varianten und Versionen von Varianten machen die erweiterte Produktstruktur sehr groß und mächtig. Für die Montage einer konkreten Festplatte muss von jeder Variante eine bestimmte Version genommen werden. Für eine automatisierte Montagezelle wird deshalb ein Rezept Management System (RMS) eingesetzt [Kobusch03]. Das RMS erlaubt zum einen die Eingabe und Verwaltung der erweiterten Produktstruktur mit allen Varianten und Versionen und zum andern bietet sie dem Leitsystem der automatisierten Montagezelle den Zugriff auf die Versionen einer konkreten Variante des Produktes. Hierdurch wird gewährleistet, dass dem Leitsystem immer die richtige Konfiguration, das heißt der konfigurierte Montagevorranggraph für die Montage eines (Teil-) Produktes zur Verfügung steht.

Dadurch ergibt sich die Anforderung an das Leitsystem, dass es in der Lage sein muss, ein Rezept, nämlich den konfigurierten Montagevorranggraphen, in die erweiterte Produktstruktur des RMS einzutragen beziehungsweise vom RMS zu erhalten. Das Eintragen im RMS wird von einem Prozessentwickler vorgenommen, nachdem er das Leitsystem einer gegebenen Montagezelle für ein konkretes Produkt konfiguriert hat. Während der automatischen Montage erhält das Leitsystem einer Montagezelle den konfigurierten Montagevorranggraphen für eine Version eines (Teil-) Produktes. Das Leitsystem kann somit diese Version des (Teil-) Produktes montieren.

3.5 Konzeption einer modularen Montagezelle

Aus der Analyse des Montageablaufs und der sich daraus ergebenden erweiterten Produktstruktur wird ein Konzept einer modularen Montagezelle entwickelt. Das zu entwickelnde konfigurierbare Leitsystem kommt in einer modularen Montagezelle zum Einsatz, welche die eigentliche Montage von (Teil-) Produkten durchführt.

3.5.1 Anforderungen an eine modulare Montagezelle und ihr Leitsystem

Die Anforderungen an die Montagezelle und somit an das Leitsystem, welches in der Montagezelle zum Einsatz kommt, bauen auf den Anforderungen aus Abschnitt 2.4 auf. Diese werden aufgrund der Erkenntnisse aus der Analyse des Montageablaufs erweitert und detailliert und führen schließlich zu folgenden Anforderungen:

- *Vollautomatisierung*: Die bisherige Montage von Festplatten wurde häufig manuell durchgeführt (vgl. Abschnitt 2.3). Hierdurch sind nicht nur Fehler bei der Montage selbst entstanden [IBM02d], sondern auch Kontaminationen durch den Menschen [VDI2083_6], was den Ausschuss von fehlerhaften Festplatten erhöht hat. Die Montage muss also vollautomatisiert, das heißt ohne den Eingriff

des Menschen, vor sich gehen. Dazu muss das Leitsystem, welches in der Montagezelle zum Einsatz kommt, auf Änderungen selbständig reagieren können.

- *Flexibilität:* Eine Vollautomatisierung ist meist daran gescheitert, dass solche Montageanlagen nur schwer an neue Produkte angepasst werden können [Bohn99]. Eine vollautomatisierte Montagezelle sowie deren Leitsystem müssen also in hohem Grade flexibel sein. Flexibilität wird dabei in verschiedenen Ausprägungen gefordert [Bullinger86, IBM02a] (siehe Abb. 3.10).

Ausprägung der Flexibilität	Beschreibung der Ausprägung
Einsatzflexibilität	Fähigkeit der Montagezelle, verschiedene Montageaufgaben (unterschiedliche Produkte) ohne großen Umrüstaufwand ausführen zu können.
Anpassungsflexibilität	Eigenschaft der Montagezelle, sich an neue Anforderungen verschiedener Montageaufgaben (Montage neuer Produkte) anpassen zu können.
Durchlaufreizügigkeit	Unabhängigkeit bei der Wahl von Bearbeitungspfaden für verschiedene Montageaufgaben gleicher und unterschiedlicher Produkte.
Fertigungsredundanz	Vorhandensein von Alternativen für die Durchführung von Montagefunktionen im Störfall.
Quantitative Kapazität	Erweiterungsfähigkeit der Montagezelle hinsichtlich der quantitativen Kapazität.
Qualitative Kapazität	Erweiterungsfähigkeit der Montagezelle hinsichtlich der qualitativen Kapazität.
Ausführungszeitflexibilität	Möglichkeit zum Reagieren auf schwankende Ausführungszeiten von direkt aufeinanderfolgenden Montagefunktionen.

Abb. 3.10: Verschiedene Ausprägungen der Flexibilität, wie sie von dem Leitsystem einer Montagezelle gefordert werden

- *Produktverfolgung:* Im Zuge eines „Total Quality Assurance“ ist die Kenntnis über die Montage eines jeden (Teil-) Produktes der erweiterten Produktstruktur nötig. Es muss also verfolgt und dokumentiert werden, in welchen Teilen einer Montagezelle die (Teil-) Produkte montiert wurden, das heißt, wo wurden die einzelnen Montagefunktionen in der Montagezelle durchgeführt und wie waren die Ergebnisse einer jeden Montagefunktion. Das Leitsystem muss jederzeit in der Lage sein, die Position eines jeden (Teil-) Produktes in der Montagezelle genau zu kennen und sie zu dokumentieren.

- *Reinheit*: Aufgrund der niedrigen Flughöhen, die ein Schreib-/Lesekopf in einer Festplatte hat, ist eine Montage der Festplatte in einer reinen Umgebung, welche nach Luftreinheitsklassen spezifizierbar ist [VDI2083_1], zwingend notwendig, um Kontaminationen innerhalb der Festplatten, speziell auf den Platten, zu vermeiden. Deshalb ist eine Luftreinheitsklasse 3 für die Montageumgebung gefordert [IBM02d].

3.5.2 Aufbau einer Montagezelle

Da die Anforderung zum einen aus der Reinheit der Montageumgebung besteht und zum anderen aus der Vollautomatisierung, das heißt der Montage ohne Eingriff des Menschen, ist es sinnvoll, die komplette Montagezelle hermetisch abzuriegeln und eine reine Umgebung darin zu schaffen. Da die Montagezelle ein sehr begrenzter Raum ist und die Hauptkontaminationsquelle, nämlich der Mensch, nicht vorhanden ist, lässt sich diese reine Umgebung leicht herstellen und auch erhalten [Schließer98, VDI2083_6]. Hierzu wird auf die Montagezelle eine Filter-Ventilatoren-Komponente gesetzt, welche in der Lage ist, einen reinen Luftstrom in der Montagezelle zu erzeugen, so dass die gesamte Montagezelle die geforderte Luftreinheitsklasse erfüllt.

Um eine Basis für die geforderte Flexibilität an das Leitsystem zu erhalten, wird die gesamte Montagezelle durch Komponenten modular aufgebaut. Eine Komponente ist eine abgeschlossene Einheit, welche eine bestimmte Montagefunktion oder eine feste Menge von Montagefunktionen durchführen kann [Muckenhirn04]. Darüber hinaus sind Komponenten auch Teile der Montagezelle, welche nicht direkt an der Montage eines (Teil-) Produktes beteiligt sind, diese aber dennoch beeinflussen können. Komponenten lassen sich deshalb in zwei Gruppen einteilen, montagerelevante Komponenten und Umgebungskomponenten.

a.) Montagerelevante Komponenten

- *Be-/Entladekomponente*: Da die Montagezelle eine hermetisch abgeschlossene Einheit bildet, müssen die (Teil-) Produkte zur Montage über eine Schleuse in die Montagezelle gelangen. Hierzu verfügt die Montagezelle über eine so genannte Be-/Entladekomponente.
- *Montagekomponenten*: Diese Komponenten führen eine oder mehrere Montagefunktionen der erweiterten Produktstruktur durch. Hierbei wird im Allgemeinen eine Montagefunktion von genau einer Komponente durchgeführt. Zu den Montagekomponenten zählen ausschließlich primäre Montagefunktionen.

- *Zentrale Handhabungskomponente:* Es gibt in einer Montagezelle mindestens eine zentrale Handhabungskomponente. Sie dient zum Transport der (Teil-) Produkte von der Be-/Entladekomponente zu den Montagekomponenten und zurück. Auch zum Transport der (Teil-) Produkte zwischen den Montagekomponenten kann die zentrale Handhabungskomponente dienen. Die zentrale Handhabungskomponente entspricht einer sekundären Montagefunktion.

b.) Umgebungskomponenten

- *Schleuse:* Die Schleuse schließt die Montagezelle nach außen hermetisch ab und erlaubt gleichermaßen das Bringen oder Holen von (Teil-) Produkten zur Be-/Entladekomponente. Sie ist somit nicht montagerelevant, da sie im Gegensatz zur Be-/Entladekomponente nicht direkt an der Montage beteiligt ist.
- *Filter-Ventilatoren-Komponente:* Nachdem die Montagezelle durch die Schleuse hermetisch abgeriegelt ist, dient die Filter-Ventilatoren-Komponente zur Erzeugung und Aufrechterhaltung des Reinraumes innerhalb der Montagezelle.
- *Steuerungen:* Diese Komponenten sind die physikalischen Steuerungen für alle anderen Komponenten. Obwohl hierunter auch die physikalischen Steuerungen für die montagerelevanten Komponenten fallen, handelt es sich um Umgebungskomponenten, da sie nicht physikalisch auf das Produkt Einfluss nehmen.

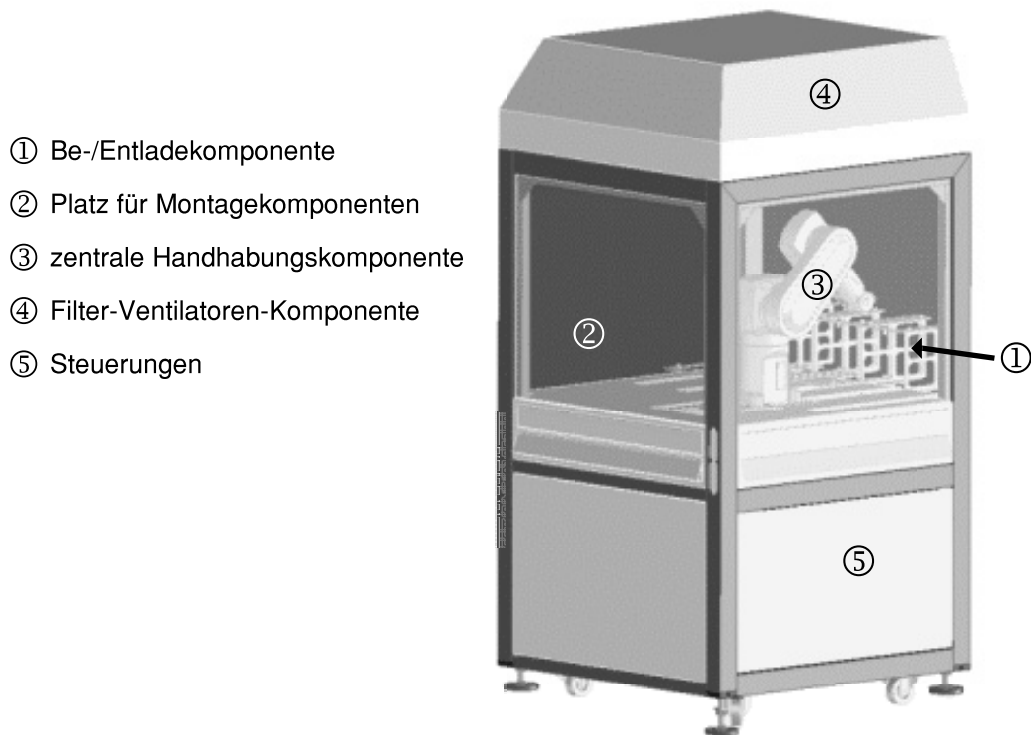


Abb. 3.11: Schematischer Aufbau einer Montagezelle (© IBM Corporation)

Abbildung 3.11 zeigt eine Montagezelle mit den oben aufgeführten Komponenten. Es ist hierbei gut zu sehen, dass in dieser Montagezelle noch keine Montagekomponenten eingebaut sind, sondern nur Platz dafür vorgehalten ist. Dies erlaubt die Zusammenstellung von beliebigen Montagekomponenten innerhalb des Raumangebotes einer Montagezelle.

Das zu entwickelnde Leitsystem muss in der Lage sein, den Ablauf aller aufgeführten Komponenten zu steuern. Hierbei sind die Umgebungskomponenten für alle Montagezellen gleich. Eine Konfiguration der Umgebungskomponenten ist deshalb nicht notwendig. Die montagerelevanten Komponenten entsprechen den primären und sekundären Montagefunktionen und müssen zu den entsprechenden Montagevorranggraphen konfiguriert werden.

3.5.3 Bildung einer Montageanlage

Aufgrund der Komplexität einer Festplatte und der damit verbundenen Anzahl von Montageschritten bei der Herstellung (vgl. Abschnitt 2.1) ist eine große Anzahl unterschiedlicher Montagezellen nötig. Nach Schätzungen werden bis zu 40 verschiedene Montagezellen benötigt [IBM02a, IBM03b].

Mehrere Montagezellen werden zu einer Montageanlage gruppiert. Es können unterschiedliche Montagezellen oder auch gleiche Montagezellen zur Steigerung der quantitativen Kapazität oder zur Bildung von Fertigungsredundanz benutzt werden. Durch die Verbindung zu einer Montageanlage wird der Transport der (Teil-) Produkte zwischen den Montagezellen automatisiert [Kobusch03, Muckenhirn03].

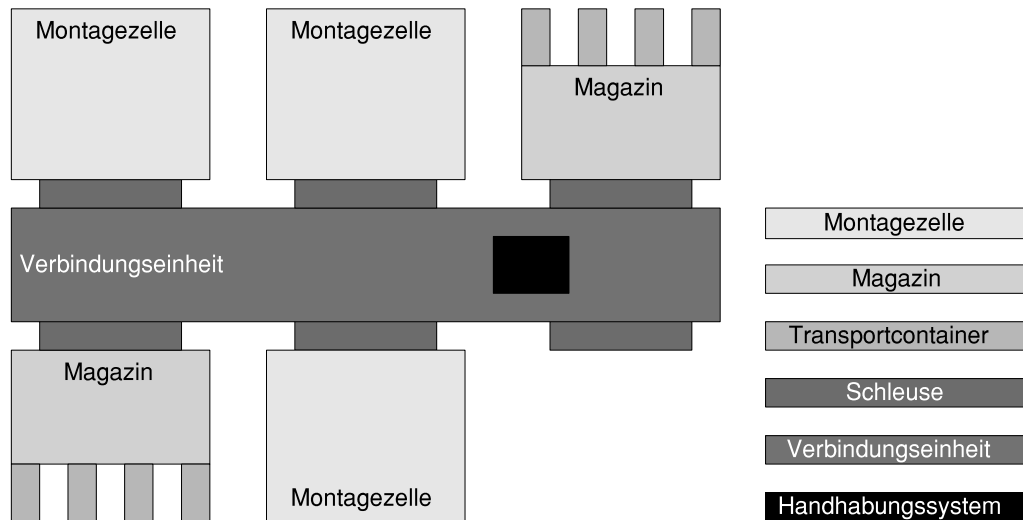


Abb. 3.12: Bildung einer Montageanlage aus Montagezellen und Magazinen

Die komplette Montageanlage, wie sie in Abbildung 3.12 zu sehen ist, besteht aus folgenden Teilen:

- *Montagezelle*: Die Montagezelle wurde in Abschnitt 3.5.2 bereits vorgestellt. Sie wird durch ihre Schleuse an der Verbindungseinheit (siehe unten) angeschlossen. Wichtig hierbei ist, dass die Schleusen aller Montagezellen in ihren Abmessungen und Positionen normiert sind. Durch diese Normierung kann jede Montagezelle an jede beliebige Stelle der Verbindungseinheit angeschlossen werden.
- *Magazin*: Nachdem die Montagezellen an die Verbindungseinheit angeschlossen sind, können (Teil-) Produkte nur noch über die Verbindungseinheit zu den Montagezellen gelangen, da diese über ihre Schleuse, und somit über die einzige Zugangsmöglichkeit, mit der Verbindungseinheit verbunden sind. Es müssen also (Teil-) Produkte von außen in die Verbindungseinheit gebracht werden. Hierfür sind Magazine zuständig. Ein Magazin besitzt wie eine Montagezelle eine Schleuse, über die sie mit der Verbindungseinheit verbunden ist. Durch eine Filter-Ventilatoren-Komponente existiert auch innerhalb des Magazins ein Reinraum. Zu einem Magazin können (Teil-) Produkte über einen Transportcontainer (siehe unten) gebracht werden. Das Magazin dient somit als

Eingabe- und Ausgabestelle für die Montageanlage.

- *Transportcontainer:* Der Transportcontainer wird zum Transport von (Teil-) Produkten zur Montageanlage hin und von der Montageanlage weg benutzt. Transportcontainer sind deshalb wichtig, um die empfindlichen (Teil-) Produkte vor einer Kontamination außerhalb der Montageanlage, und somit außerhalb der reinen Umgebung, zu schützen.
- *Schleuse:* Die Schleuse ist eine normierte Anschlussstelle an der Verbindungseinheit für Montagezellen und Magazine. Sie bildet das Gegenstück für die Schleuse an den Montagezellen und den Magazinen. Zusätzlich bietet sie einen Kontaminationsschutz der Montageanlage, falls an einer Schleuse weder eine Montagezelle noch ein Magazin angeschlossen ist.
- *Verbindungseinheit:* Die Verbindungseinheit verbindet eine maximale Anzahl von Montagezellen und den Magazinen miteinander. Dafür verfügt sie über eine feste Anzahl von Schleusen, an denen beliebig sowohl Montagezellen als auch Magazine angeschlossen sein können. Aufgrund der Schleusen können Montagezellen oder Magazine sogar während des Betriebes der Montageanlage angeschlossen oder auch entfernt werden. Um die geforderte Reinheit auch innerhalb der Verbindungseinheit zu gewährleisten, ist auch diese mit einer Filter-Ventilatoren-Komponente ausgerüstet. Dadurch erhält man innerhalb der gesamten Montageanlage den geforderten Reinraum.
- *Handhabungssystem:* (Teil-) Produkte müssen zwischen den verschiedenen Montagezellen und Magazinen transportiert werden. Dieser Transport der (Teil-) Produkte ist die Aufgabe des Handhabungssystems.

3.5.4 Transport der Produkte auf einem Werkstückträger

Durch die Verbindung der Montagezellen und der Magazine zu einer Montageanlage können die (Teil-) Produkte leicht zwischen diesen transportiert werden. Die (Teil-) Produkte, welche innerhalb der Verbindungseinheit durch das Handhabungssystem transportiert werden, sind in ihrer Form und Größe sehr unterschiedlich. Da die Montageanlage auch in der Lage sein soll, neue (Teil-) Produkte leicht zu integrieren, muss eine sehr universelles Handhabungssystem vorhanden sein [IBM02f, Muckenhirn04].

Das Handhabungssystem ist deshalb so ausgelegt, dass es einen normierten Werkstückträger transportieren kann. Hierdurch müssen für unterschiedliche (Teil-) Produkte nicht unterschiedliche Handhabungssysteme zum Einsatz kommen.

Der Werkstückträger ist von seinen äußeren Abmessungen normiert und kann somit ohne Probleme von dem Handhabungssystem transportiert werden. Um nun unterschiedliche (Teil-) Produkte mit dem Werkstückträger transportieren zu können, werden für jedes (Teil-) Produkt verschiedene Einlagen in den Werkstückträger eingesetzt (siehe Abb. 3.13). Auf einer Einlage können wiederum beliebig viele (Teil-) Produkte des gleichen Produkttyps oder eines ähnlichen Produkttyps liegen [IBM02g].

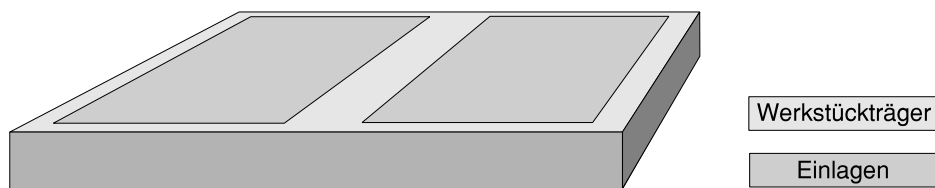


Abb. 3.13: Beispiel eines Werkstückträgers mit zwei Einlagen

Ein *Produkttyp* ist die abstrakte Beschreibung eines konkreten (Teil-) Produktes. Zwei (Teil-) Produkte sind vom selben Produkttyp, wenn sie aus den gleichen Teilprodukten bestehen und die gleichen Montagefunktionen durchlaufen haben. Sie sind also, bis auf eine eindeutige Identifizierung, identisch.

Zwei Produkttypen sind ähnlich, wenn sie in ihrer äußeren Form und Größe gleich sind. Bei zwei ähnlichen Produkttypen kann der eine durch zusätzliche Montagefunktionen in den anderen überführt werden. Häufig bestehen ähnliche Produkttypen aus den gleichen Teilprodukten, haben aber nicht die gleichen Montagefunktionen durchlaufen.

Nachdem das Handhabungssystem aufgrund der Normierung des Werkstückträgers vereinheitlicht werden konnte, ist dies auch bei den Be-/Entladekomponenten der Montagezellen und Magazine möglich. Innerhalb einer Montagezelle können an der Verbindung zur Montageanlage mehrere Be-/Entladekomponenten angebracht sein, von denen jede jeweils einen normierten Werkstückträger aufnehmen kann. Durch die Einführung des Werkstückträgers können unterschiedlichste (Teil-) Produkte auf eine einfache Weise zwischen allen Montagezellen und Magazinen transportiert werden.

3.5.5 Verknüpfung von Montageanlagen zu einem Montageanlagenverbund

Die konzipierte Montageanlage bietet eine Flexibilität bei der Zusammenstellung der Anlage aus einzelnen Montagezellen und Magazinen. Sie erlaubt auch den Umbau der Anlage während des Betriebes. Sie ist aber auch begrenzt. Aufgrund der technisch begrenzten Länge des Handhabungssystems können nur eine maximale Anzahl von Montagezellen oder Magazinen miteinander zu einer Montageanlage verbunden werden. Danach muss eine neue Montageanlage gebildet werden. Bei den Transporten zwischen Montageanlagen handelt es sich um lange Wege, das heißt Wege, welche durch menschliches Bedienpersonal oder durch automatische externe Transportsysteme überbrückt werden müssen. Um auch die Wege zwischen einer größeren Anzahl von Montagezellen oder Magazinen kurz zu halten, müssen mehrere Montageanlagen miteinander verbunden werden können. Dies geschieht über ein Verbindungsstück an den Kopfenden der Verbindungseinheiten (siehe Abb. 3.14). Das Verbindungsstück dient dabei als Zwischenspeicher für die Werkstückträger, auf den das Handhabungssystem der angeschlossenen Montageanlagen zugreifen können.

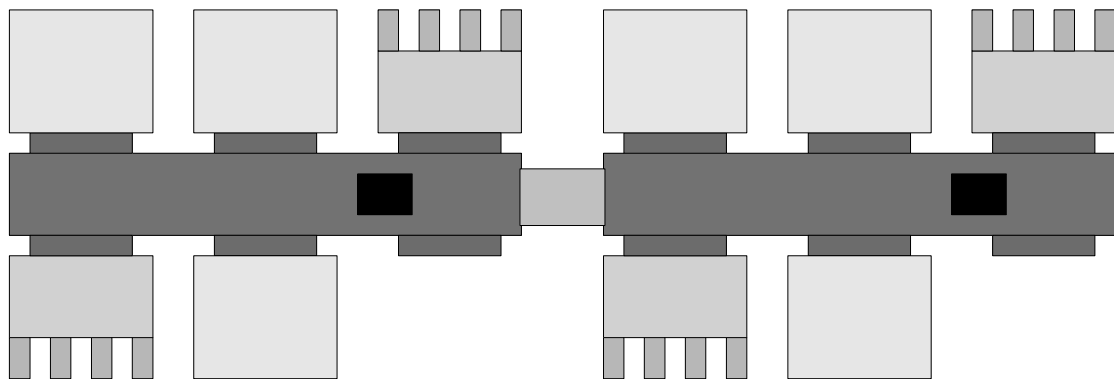


Abb. 3.14: Verknüpfung von mehreren Montageanlagen zu einem Montageanlagenverbund

Durch die Möglichkeit der Verkettung von Montageanlagen zu einem Montageanlagenverbund bieten sich mehr Möglichkeiten, eine Anlage zusammenzustellen, welche aus beliebigen Montagezellen und Magazinen besteht. Durch die Verkettung von mehr als zwei Montageanlagen lässt sich so ein Montageanlagenverbund für eine komplette Festplattenmontage zusammenstellen.

3.6 Abgrenzung der Arbeit und Zusammenfassung der Anforderungen an das Leitsystem

Kernstück der Montageanlage zur Montage von Festplatten bildet die Montagezelle. Die Montagezelle ist modular aus verschiedenen Komponenten aufgebaut, welche zum einen die primären und sekundären Montagefunktionen durchführen und zum andern für eine montagegerechte Umgebung sorgen, wie sie für die Montage von Festplatten gefordert ist. Diese Umgebungskomponenten sind für alle Montagezellen gleich. Im Gegensatz dazu können in einer Montagezelle beliebige montagerelevante Komponenten zum Einsatz kommen (siehe Abb. 3.15).

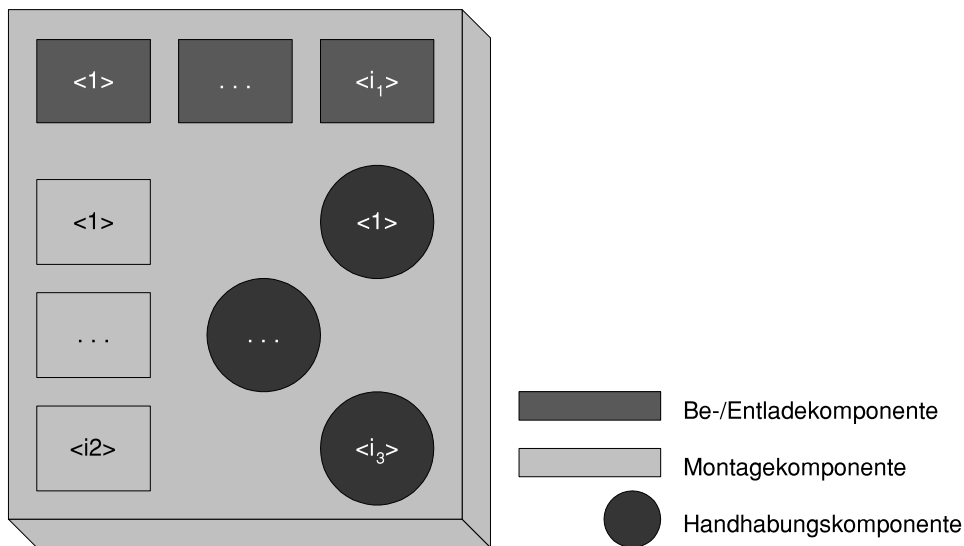


Abb. 3.15: Schematischer Aufbau der modularen Montagezelle

Aufgabe dieser Arbeit ist es, ein Leitsystem zu entwickeln, welches neben der Steuerung der Umgebungskomponenten insbesondere an die unterschiedlichsten Zusammenstellungen von montagerelevanten Komponenten innerhalb einer Montagezelle angepasst wird und damit diese steuern kann.

Sowohl aus der Analyse des Montageablaufs als auch aus der Konzeption der modularen Montagezelle und der Möglichkeit der Verknüpfung von mehreren Montagezellen zu einer Montageanlage ergeben sich folgende Anforderungen.

a.) Vollautomatisierung

Das Leitsystem muss während der Produktion selbständig, das heißt ohne den Eingriff eines Menschen, funktionieren und die Montagezelle in der Art steuern, dass die zuvor gewählten Produkte in ihrer definierten Form hergestellt werden.

Ein Wechsel auf andere Produkte muss von dem Leitsystem auch ohne den Eingriff eines Menschen unterstützt werden (insofern die physikalischen Komponenten dazu in der Lage sind).

b.) Konfiguration

Das Leitsystem muss für verschiedene Produkte konfigurierbar sein. Es muss der Ablauf zwischen den montagerelevanten Komponenten konfiguriert werden können. Das heißt, aus den primären und sekundären Montagefunktionen, denen die montagerelevanten Komponenten entsprechen, muss ein Montagevorranggraph für den Montageablauf innerhalb der Montagezelle konfiguriert werden. Obwohl die Be-/Entladekomponente keiner Montagefunktion entspricht, zählt sie dennoch zu den montagerelevanten Komponenten, da sie die (Teil-) Produkte zur Verfügung stellt und somit direkt auf die Montage Einfluss nimmt. Da über die Be-/Entladekomponente unterschiedlichste (Teil-) Produkte in verschiedenen Werkstückträgern für die Montage zur Verfügung gestellt werden, muss das Leitsystem darauf konfiguriert und während der Produktion dies steuern können.

c.) Flexibilität

Das Leitsystem muss flexibel sein. Flexibilität gibt es dabei in verschiedenen Ausprägungen:

- *Einsatzflexibilität*: Das Leitsystem muss verschiedene Montageaufgaben, das heißt die Montage unterschiedlicher (Teil-) Produkte, durchführen können.
- *Anpassungsflexibilität*: Das Leitsystem muss an neue Montageaufgaben angepasst werden können.
- *Durchlauffreizügigkeit*: Das Leitsystem muss unabhängig sein bei der Wahl von Bearbeitungspfaden für verschiedene Montageaufgaben gleicher und unterschiedlicher (Teil-) Produkte.
- *Fertigungsredundanz*: Das Leitsystem muss Alternativen für die Durchführung von Montagefunktionen im Störfall nutzen, insofern diese vorhanden sind.
- *Quantitative Kapazität*: Hinzufügen von montagerelevanten Komponenten, welche Montagefunktionen, die Engpässe darstellen, durchführen können, muss zu einer Erhöhung des Durchsatzes führen.

- *Qualitative Kapazität*: Austausch oder Hinzufügen von Komponenten, welche qualitativ besser arbeiten, müssen durch das Leitsystem konfiguriert und gesteuert werden können.
- *Ausführungszeitflexibilität*: Das Leitsystem muss auf schwankende Ausführungszeiten von direkt aufeinander folgenden Montagefunktionen reagieren können.

d.) Produktverfolgung

Die Verfolgung der (Teil-) Produkte durch die gesamte Montagezelle muss möglich sein. Der Weg der (Teil-) Produkte, das heißt, sowohl die Montagefunktionen als auch die Komponenten auf denen die Montagefunktionen durchgeführt werden, müssen für jedes (Teil-) Produkt protokolliert und archiviert werden, so dass eine Rückverfolgung der einzelnen (Teil-) Produkte aus denen eine Festplatte besteht und die Montagefunktionen, die an ihr durchgeführt wurden, später möglich ist.

Kapitel 4

Stand der Technik

Um eine modulare Montagezelle steuern zu können, soll ein konfigurierbares Leitsystem entwickelt werden. In der Festplattenmontage sind solche Leitsysteme nicht vorhanden [IBM02b]. In diesem Kapitel wird der Stand der Technik von sowohl hardwaretechnischen als auch softwaretechnischen Systemen aufgezeigt, welche für die Steuerung von modularen aber auch nicht modularen Montagezellen benutzt werden können.

4.1 Systemarchitekturen

Systemarchitekturen bilden die physikalische Grundlage für die Steuerung einer modularen Montagezelle. Sie sagen nichts über das softwaretechnische System aus, welches auf ihnen zum Einsatz kommt. Je nach Komplexität der modularen Montagezellen und damit deren Leistungsbedarf stehen drei verschiedene Ansätze zur Verfügung.

4.1.1 Zentrale Systemarchitektur

Bei der zentralen Systemarchitektur (zentrales Prozessrechensystem) gibt es genau ein Prozessrechensystem [DIN19233]. Dieses Prozessrechensystem greift in der Regel über einen Feldbus direkt auf die Sensoren und Aktoren zu (siehe Abb. 4.1). Üblicherweise handelt es sich bei diesen zentralen Prozessrechensystemen um speicherprogrammierbare Steuerungen (SPS), da auf der Ebene der Sensorik und Aktorik Echtzeitfähigkeit gefordert ist [Bartels99, Paschen99]. Aufgrund der Leistungsfähigkeit der Personal Computer (PC) sind Echtzeitanwendungen inzwischen auch dort möglich, so dass solche Systeme inzwischen auch häufiger rein PC-basiert existieren [Bartels99].

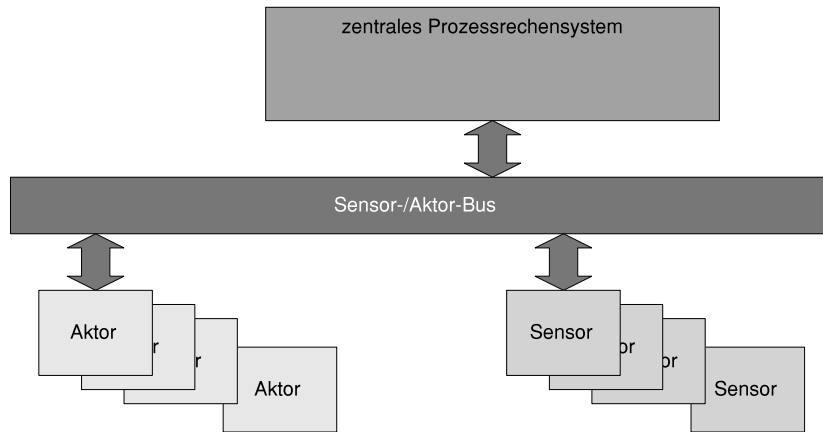


Abb. 4.1: Zentrale Systemarchitektur

Das Prozessrechensystem übernimmt hierbei nicht nur die Automatisierung der Montagezelle, sondern es bildet gleichzeitig auch das Leitsystem ab. Die bei der Konzeption eingeführten physikalischen Komponenten werden bei einem zentralen Prozessrechensystem nicht berücksichtigt. Es werden die Sensoren und Aktoren aller Komponenten von dem zentralen System gesteuert. Ein solches System wird dann eingesetzt, wenn die Anzahl der Aktoren und Sensoren aller Komponenten so gering ist, dass sie von einem Prozessrechensystem abgebildet werden können.

4.1.2 Hierarchische Systemarchitektur

Bei einer hierarchischen Systemarchitektur (hierarchisches Prozessrechensystem) wird üblicherweise jede physikalische Komponente der Montagezelle durch ein eigenes Prozessrechensystem abgebildet. Im Gegensatz zur zentralen Systemarchitektur werden hier aber noch zusätzlich übergeordnete Prozessrechensysteme eingeführt [DIN19233]. Diese haben die Aufgabe, die dezentralen Prozessrechensysteme der einzelnen Komponenten zu koordinieren und den Ablauf zwischen den einzelnen Komponenten zu übernehmen. Sie bilden somit ein Leitsystem für die dezentralen Prozessrechensysteme (siehe Abb. 4.2).

Um den Datenaustausch zwischen dem Leitsystem und den dezentralen Prozessrechensystemen zu standardisieren, wurde 1996 die OPC-Spezifikation verabschiedet und bietet in ihrer heutigen Spezifikation den herstellerübergreifenden Zugriff auf Daten von beliebigen dezentralen Prozessrechensystemen [Iwanitz99, OPC04]. Hierzu wurde zwischen einer Datenquelle (OPC-Server) und einer Datensenke (OPC-Client) eine einheitliche COM/DCOM-Schnittstelle definiert [Licha03]. Da diese Technologie seitens eines Microsoft Windows® PC von vielen Applikationen und Programmiersprachen unterstützt wird, stehen ganz neue Möglichkeiten für das

Leitsystem zur Verfügung [Munoz00].

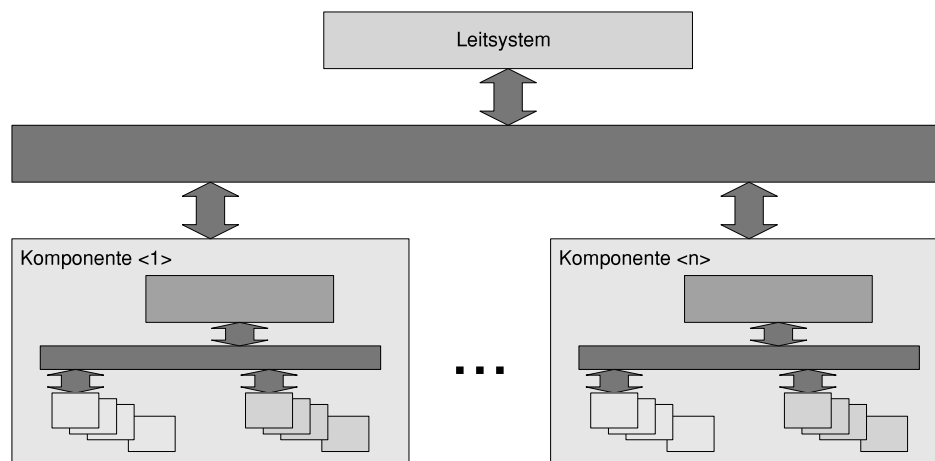


Abb. 4.2: Hierarchische Systemarchitektur

4.1.3 Vollständig verteilte Systemarchitektur

Bei einer vollständig verteilten Systemarchitektur (verteilttes Prozessrechensystem) gibt es im Gegensatz zu der hierarchischen Systemarchitektur kein zentrales Prozessrechensystem. Vielmehr übernehmen die dezentralen Prozessrechensysteme mehr Verantwortung. Sie werden autonomer und intelligenter [DIN19233]. Gleichzeitig müssen die dezentralen Prozessrechensysteme miteinander kooperieren beziehungsweise miteinander kommunizieren. Ein solches System kann auch durch redundante Prozessrechensysteme erweitert werden, so dass auch nach dem Ausfall eines Prozessrechensystems der Prozess weiter betrieben werden kann [DIN19233].

Die wichtigsten Vertreter, die diese Anforderungen erfüllen, sind die technischen Multi-Agenten-Systeme [Colombo02, Lüth98, Ritter01, Schoop00] und verteilte holonische Systeme, welche bei ihrer Realisierung meist auf ein technisches Multi-Agenten-System aufbauen [Dutzler02].

4.2 Softwaresysteme

4.2.1 Programmierung

Eine feste Programmierung ist sehr effizient, da sie speziell an ein gestelltes Problem, nämlich der Montage von einigen Festplattentypen, angepasst werden kann. Die Programmierung erfolgt üblicherweise in einer der Norm IEC61131-3 [DIN61131] konformen Programmiersprache und wird normalerweise von einem Software- oder

Steuerungsentwickler durchgeführt [Bartels99, Bengel03]. Um das Komponentenkonzept aus Abschnitt 3.5.2 umzusetzen, können die einzelnen Komponenten als Funktionsbausteine [DIN61131] und somit als Softwarekomponenten ausgebildet werden. Je nach Einsatz von wieder verwendbaren Softwarekomponenten wird die Zeit für die Neuentwicklung einer Montagezelle reduziert. Dennoch werden für die Realisierung einer Montagezelle, beziehungsweise für die physikalische Umkonfiguration einer Montagezelle, Softwareentwickler oder Steuerungsentwickler benötigt, was die Änderungen sehr kostspielig und zeitintensiv macht [Pritschow91].

4.2.2 Konfiguration

Um den Aufwand für Neuentwicklungen zu reduzieren, wird die Programmierung durch eine Konfiguration ersetzt. Hierzu werden wieder verwendbare Softwarekomponenten für Sensoren, Aktoren, Funktionseinheiten und Funktionsgruppen in Form eines Baukastensystems zur Verfügung gestellt und über ein Case Tool oder eine föderale Informations-Architektur konfiguriert und parametrisiert [Föederal04, MOWIMA97].

Durch die Bildung von Softwaremodulen, welche einer physikalischen Komponente zugeordnet sind (vgl. Abschnitt 3.5.2) und über eine standardisierte Software-schnittstelle verfügen [SEMI30], können diese herstellerübergreifend kombiniert, konfiguriert und somit gesteuert werden [Bader99a, Munoz00, Westkämper01].

Für die Konfiguration wieder verwendbarer Softwarekomponenten können auch Ablaufsprachen [DIN61131], Automatengraphen, Flussdiagramme sowie höhere Programmiersprachen (z.B. C oder Visual Basic) benutzt werden, welche von einem Prozessingenieur einfach zu erlernen sind [Frey01]. Sind Softwaremodule mit einer standardisierten Schnittstelle vorhanden, so kommen Skriptsprachen [IBM03c] oder auch Konfigurationen graphischer Ablauffolgen zum Einsatz.

Ein weiterer Schritt ist der Übergang von einer statisch konfigurierten Ablauffolge hin zu einer dynamischen Ablauffolge. Hierzu werden für jedes Softwaremodul zusätzliche Zeitinformationen (durchschnittliche Ausführungszeit, maximale Ausführungszeit, spätester Startzeitpunkt, usw.) bei der Konfiguration der Ablauffolgen angegeben. Die Reihenfolge der einzelnen Ablaufschritte wird nicht mehr fest vorgegeben, sondern es werden nur Beziehungen zwischen den Ablaufschritten konfiguriert. Mit Hilfe dieser Zeitinformationen und der Beziehungen zwischen den Ablaufschritten wird aus der konfigurierten Ablauffolge eine optimierte Ablauffolge (Schedule) erstellt [BLE01, Watts00]. Abbildung 4.3 zeigt einen solchen Schedule, der für eine Montagezelle mit vier Montagekomponenten und einer Handhabungs-

komponente erstellt wurde. Die Handhabungskomponente ist für den Transport aller Produkte zwischen den Montagekomponenten zuständig. Dadurch wird sie zum Flaschenhals, weshalb eine Berechnung einer optimierten Ablauffolge anhand der bekannten Ausführungszeiten sinnvoll ist.

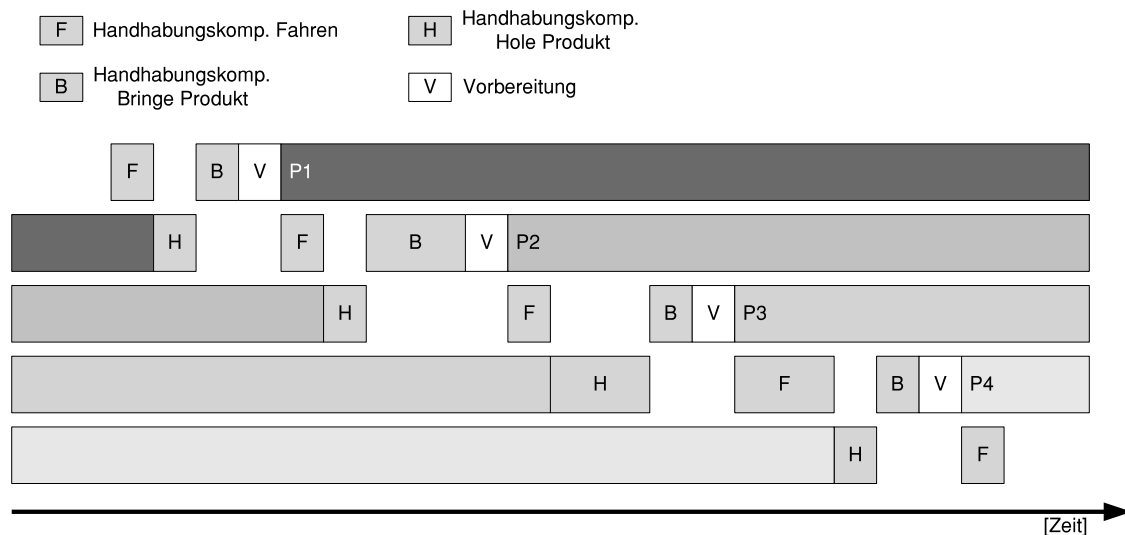


Abb. 4.3: Schedule für vier Montagekomponenten und eine Handhabungskomponente

4.2.3 Modellierungsansätze

Bei Modellierungsansätzen wird mit Hilfe einer Modellierungssprache die komplette Montagezelle beschrieben. Es gibt hierbei Ansätze bei denen für diese Modellierung die Unified Modeling Language (UML) [Oestereich04, OMG04] beziehungsweise eine Erweiterung der UML benutzt wird [Bengel03, Braatz00, Vogel-Heuser00, Westkämper01]. Andere Ansätze nutzen Erweiterungen der Petri-Netze [Aspern93, Colombo98, Desrochers95, Ezpeleta97, Frey01].

Aus dem Modell, welches die Abläufe innerhalb der Montagezelle beschreibt, kann schließlich Programmcode für die eingesetzten Prozessrechnerysteme automatisch generiert werden [Bengel02, Frey01, Prehn02, Vogel-Heuser00].

4.2.4 Discrete Event System

Eine Einheit des *Discrete Event Systems* (DES) besteht aus einem endlichen Zustandsraum und einer dazugehörenden Zustands-Übergangsstruktur. Gesteuert werden die Zustandsübergänge durch so genannte *Events*, welche üblicherweise asynchron sind, aber auch von einem Zeitgeber ausgelöst werden können. Ein *DES* ist somit diskret in seinem Zustandsraum als auch diskret bezüglich der Zeit. Da

Zustandsübergänge auch durch interne Gegebenheiten der Zustände beeinflusst werden können, ist ein *DES* auch als nicht deterministisch anzusehen [Wonham03].

DES kommen sowohl bei der Konfiguration von Softwaresystemen als auch bei der Modellierung der Softwaresysteme zum Einsatz [Freund00, Prehn02]. Um die Komplexität des *DES* zu reduzieren, kommen sie häufig bei hierarchischen Systemarchitekturen oder bei einer vollständig verteilten Systemarchitektur zum Einsatz [Brandin96, Colombo98].

4.3 Zusammenfassung

Die Steuerung einer modularen Montagezelle für die Montage von Festplatten stellt hohe Anforderungen an das einzusetzende Leitsystem. Da in der Festplattenmontage keine solchen Leitsysteme vorhanden sind, wurden Lösungsansätze aus anderen Bereichen, in denen Steuerungstechnik und Leitsysteme eingesetzt werden, untersucht. Bei dieser Untersuchung hat jedoch keines der betrachteten Systeme alle Anforderungen zur Zufriedenheit erfüllt, welche an ein Leitsystem zur Steuerung einer modularen Montagezelle in der Festplattenmontage gestellt werden [IBM02b, IBM02d].

Im nächsten Kapitel wird ein neues Leitsystem entwickelt, welches für die Steuerung von modularen Montagezellen benutzt werden kann. Grundlage für diese Entwicklung bilden Anforderungen (vgl. Abschnitt 3.6), die speziell von der Festplattenmontage an ein solches Leitsystem gestellt werden.

Kapitel 5

Entwicklung des konfigurierbaren Leitsystems für Montagezellen

In diesem Kapitel wird ein Leitsystem entwickelt, welches in der Lage ist, eine beliebige modulare Montagezelle für die Montage von Festplatten zu steuern, gleichzeitig aber so einfach ist, dass sowohl Änderungen der Komponenten in der Montagezelle als auch Änderungen der zu montierenden Produkte von einem Prozessentwickler konfiguriert werden können. Hierbei liegen der gesamten Entwicklung die Anforderungen aus Kapitel 3 zugrunde.

5.1 Komponentenbildung

Bei der Konzeption einer Montagezelle in Kapitel 3 wurde der physikalische Aufbau einer modularen Montagezelle vorgestellt. Hierbei wurde die Montagezelle in separate mechanische Komponenten unterteilt. Auch beim Leitsystem wird eine Unterteilung in Komponenten vorgenommen.

5.1.1 Definition einer physikalischen Komponente

Auf Seiten des Leitsystems wird nicht, wie im mechanischen Aufbau, zwischen montagerelevanten Komponenten und Umgebungskomponenten unterschieden. Alle *physikalischen Komponenten* sind gleichwertig und werden durch folgende Eigenschaften definiert:

- Abgeschlossenes System
- Autarke logische Steuerung
- Kann selbständig h Aufgaben übernehmen, $h \in \mathbb{N}$

Hierbei kann eine *logische Steuerung* auf einer *physikalischen Steuerung* (Prozessrechen-system [DIN19233]) zum Einsatz kommen, aber auch mehrere *logische Steuerungen* auf einer *physikalischen Steuerung* sind möglich (siehe Abb. 5.1). Eine *physikalische Steuerung* kann eine speicherprogrammierbare Steuerung (SPS), ein Personal Computer (PC), ein Mikrocontroller oder eine ähnliche Hardware sein, welche zu Steuerungszwecken sonst noch eingesetzt werden kann.

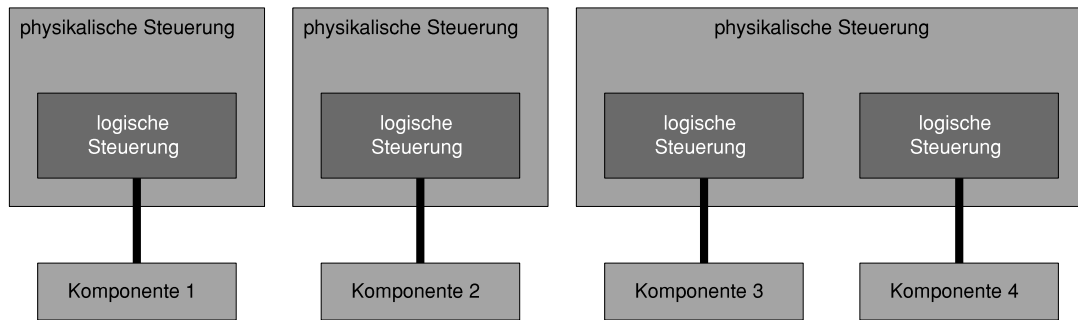


Abb. 5.1: Zuordnung der logischen Steuerung zur physikalischen Steuerung

Jede Aufgabe kann beliebig viele *Eingangsprodukte* und maximal ein *Ausgangsprodukt* haben. Die Eingangsprodukte sind hierbei Teilprodukte, welche durch die Aufgabe zu einem Produkt, dem Ausgangsprodukt, montiert werden (vgl. erweiterte Produktstruktur in Abschnitt 3.3.1).

5.1.2 Klassifizierung von Komponenten

Die physikalischen Komponenten können in drei Komponentenklassen eingeteilt werden:

- *passive Komponenten:* Keine der h Aufgaben hat eine Interaktion mit einer anderen Komponenten. Hierunter fallen im Allgemeinen Montagekomponenten, die Be-/Entladekomponente aber auch Umgebungskomponenten.
- *aktive Komponenten:* Mindestens eine Aufgabe hat eine Interaktion mit einer anderen Komponente und keine andere aktive Komponente hat mit dieser eine Interaktion. Handhabungskomponenten gehören fast immer in diese Komponentenklasse.
- *gemischte Komponenten:* Hierunter fallen alle Komponenten, die weder reine passive noch reine aktive Komponenten sind. Dies bedeutet, dass es mindestens eine Aufgabe gibt, welche eine Interaktion mit einer anderen Komponente und mindestens eine andere Komponente existiert, welche eine Interaktion mit dieser

Komponente hat. Hierunter fallen Komponenten, denen Montageteile durch eine andere Komponente zugeführt werden und gleichzeitig als Zuführung von Montageteilen zu anderen Komponenten dienen.

5.1.3 Anbindung der Komponenten an das Leitsystem

Die oben definierten Komponenten werden über Schnittstellen mit dem Leitsystem verbunden. Abbildung 5.2 zeigt ein typisches Anwendungsbeispiel eines solchen Systems mit einer Handhabungskomponente, einer Be-/Entladekomponente und einigen Montagekomponenten. Die Komponenten sind jeweils über separate Schnittstellen mit dem Leitsystem verbunden. Dies erlaubt die Konfiguration des Montageablaufs zentral im Leitsystem.

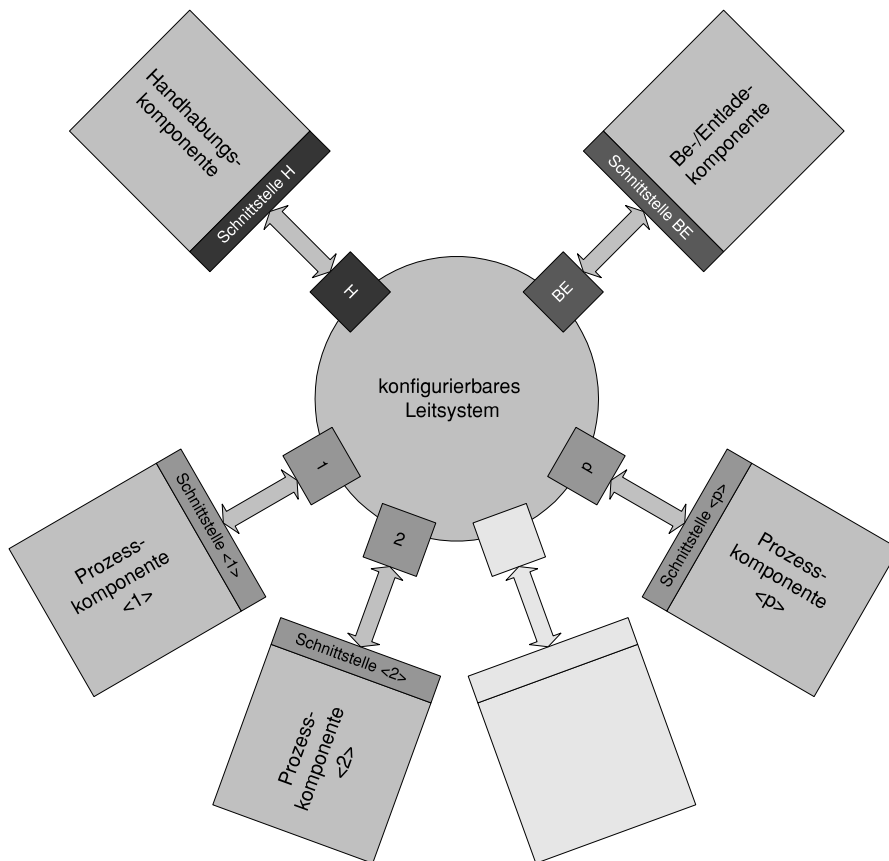


Abb. 5.2: Anbindung der Komponenten an das Leitsystem

5.2 Standardisierte Schnittstelle für Komponenten

Um die in Abschnitt 5.1.3 vorgestellte Anbindung der Komponenten an das Leitsystem zu vereinfachen, wird eine standardisierte Schnittstelle eingeführt. Dies führt dazu, dass das Leitsystem nicht mehr für jede angeschlossene Komponente eine eigene Schnittstelle benötigt (siehe Abb. 5.2), sondern eine standardisierte Schnittstelle nur einmal realisiert werden muss.

Da nicht nur PC-basierte Steuerungen zum Einsatz kommen, sondern sehr häufig auch eine SPS oder ein Mikrocontroller, muss die Schnittstelle einfach beschaffen sein, so dass sie von all diesen Steuerungen realisierbar ist. Trotz ihrer Einfachheit muss sie dennoch allen Anforderungen genügen, die durch die verschiedenen Arten von Komponenten an sie gestellt sind. Zusätzlich zu ihrer Einfachheit muss die Schnittstelle noch über andere Eigenschaften verfügen:

- Sie muss schnell und leistungsfähig sein. Dies bedeutet, dass kein unnötiger Ballast an Daten vorhanden sein soll.
- Robustheit, das heißt unanfällig gegenüber äußeren Einflüssen.
- Die Datenübertragung muss insofern sichergestellt sein, dass die von einem Schnittstellenpartner gesendeten Daten auch sicher beim Empfänger ankommen, oder aber der Absender der Daten feststellt, dass seine gesendeten Daten beim Empfänger nicht angekommen sind.
- Ausfallsicherheit der Kommunikation, wie sie bei firmeninternen Netzwerken (Intranet) oder bei firmenübergreifenden Netzwerken (Internet) häufig realisiert ist, lässt sich meist nur durch Redundanz der Kommunikationspfade bewerkstelligen. Dies ist aber bei Steuerungen, wie sie hier zum Einsatz kommen, nur selten möglich. Deshalb sollte zumindest der Ausfall der Kommunikation festgestellt werden.

Um dies alles zu erreichen, wird die gesamte Schnittstelle in vier Schichten unterteilt (siehe Abb. 5.3).



Abb. 5.3: Vier Schichten der standardisierten Schnittstelle

5.2.1 Schicht 1: Physikalisches Protokoll

Als physikalisches Protokoll wird Ethernet benutzt. Hierbei handelt es sich zwar um eine Techni

k, deren Einsatzgebiet hauptsächlich bei PC basierten Systemen liegt, die inzwischen jedoch von fast allen Steuerungen, seien es SPS oder Mikrocontroller, angeboten wird.

5.2.2 Schicht 2: Datentransport

Für den Datentransport wird TCP/IP benutzt. Aufbauend auf TCP/IP wird in der Datentransportschicht nach dem Erhalt von Daten eine Bestätigung an den Absender gesendet, so dass dieser zuverlässig über die Ankunft der Daten beim Empfänger informiert ist. Somit ist die Datenübertragung vom Sender zum Empfänger gewährleistet. Um auch noch die Ausfallsicherheit zu gewährleisten, das heißt den Ausfall der Kommunikation festzustellen, wird eine Zeitspanne t_{ACK} definiert, womit ein Ausfall der Kommunikation angezeigt wird, wenn nach dem Absenden der Daten die Bestätigung beim Absender nach Ablauf der Zeitspanne t_{ACK} nicht eingetroffen ist. Die Datentransportschicht kann somit den Ausfall der Kommunikation feststellen und diesen an seine übergeordneten Schichten weitermelden.

5.2.3 Schicht 3: Komponenten-Kommunikations-Protokoll

Schicht 1 und Schicht 2 haben sich mit dem Transport der Daten beschäftigt, ohne die Daten selbst zu spezifizieren. Das Komponenten-Kommunikations-Protokoll beschreibt, was für Daten transportiert werden. Hierzu müssen erst einmal Überlegungen über die Anforderungen an ein solches Protokoll angestellt werden. Prinzipiell gibt es zwei Arten von Kommunikation in dieser Schicht:

- *synchrone Kommunikation*: Hierbei wartet der Absender auf Antwortdaten zu den zuvor zum Empfänger versandten Daten. Der Empfänger verarbeitet die vom Sender eingetroffenen Daten und sendet danach Antwortdaten an diesen zurück.

Der Sender ist in dieser Zeit blockiert. Diese Art der Kommunikation wird aus der Sicht des Senders als Befehl bezeichnet.

- *asynchrone Kommunikation:* Bei dieser Kommunikationsart sendet der Absender seine Daten an den Empfänger, erwartet jedoch keine Antwortdaten. Der Empfänger verarbeitet die Antworten asynchron zum Sender weiter. Diese Kommunikationsart wird als Nachricht bezeichnet.

Neben der Kommunikationsart (Befehl oder Nachricht) sind im Komponenten-Kommunikations-Protokoll noch Parameter definiert. Hiermit lassen sich beliebig viele Parameter an einen Befehl oder eine Nachricht binden. Es wird hier ausschließlich die Syntax der Parameter definiert. Die Semantik der einzelnen Parameter wird erst im Komponenten-Kommunikations-Protokoll definiert und ist abhängig vom jeweiligen Befehl oder der jeweiligen Nachricht.

Auch auf dieser Schicht ist es wichtig, die Ausfallsicherheit zu gewährleisten. Der Ausfall kann jedoch nur bei der synchronen Kommunikation festgestellt werden. Ähnlich wie bei der Datentransportschicht wird hier eine Zeitspanne t_{RES} definiert, nach deren Zeit die Antwortdaten beim Sender angekommen sein müssen.

Alle drei Komponentenarten (passiv, aktiv und gemischt) haben Aufgaben, die von einem Leitsystem an sie gestellt werden (vgl. Abschnitt 5.1). Hierbei bedient sich das Leitsystem immer der synchronen Kommunikation, das heißt es sendet Befehle an die angeschlossenen Komponenten. Die Komponenten ihrerseits bedienen sich nur der asynchronen Kommunikation und senden nur Nachrichten zum Leitsystem. Eine direkte Kommunikation zwischen zwei Komponenten ist nicht erwünscht, da dies eine Konfiguration oder sogar Programmierung auf einer der beiden Komponenten bedingen würde.

5.2.4 Schicht 4: Komponenten-Kommunikations-Schnittstelle

Diese Schicht bildet die eigentliche Schnittstelle zum Leitsystem. Hier werden die Befehle vom Leitsystem zu einer Komponente und die Nachrichten der Komponenten zum Leitsystem definiert. Diese Schnittstelle kann mit vielen Befehlen und Nachrichten definiert werden, welche sowohl die Konfiguration als auch die Lösungssuche bei auftretenden Problemen erleichtern. Hier werden jedoch nur die Befehle und Nachrichten explizit definiert, welche im weiteren Verlauf benötigt werden.

a.) Befehle

- *Start*: Eine der h Aufgaben einer Komponente wird gestartet. Dieser Befehl hat als Parameter die Aufgabennummer, die Positionen der Eingangsprodukte und die Position des Ausgangsproduktes. Dieser Befehl dient lediglich zur Initiierung einer Aufgabe. Der Absender wird nicht blockiert bis die Aufgabe abgearbeitet ist, sondern nur bis die Initiierung der Aufgabe abgeschlossen ist. Die Antwort ist die Bestätigung über die erfolgreiche oder nicht erfolgreiche Initiierung der Aufgabe.
- *Pause*: Eine laufende Aufgabe wird angehalten. Hierbei wird gewartet bis die Aufgabe angehalten ist, bevor die Antwort gesendet wird. Bei der Antwort handelt es sich um eine leere Antwort, die nur zur Bestätigung über die Abarbeitung des Befehls benutzt wird.
- *Fortfahren*: Hiermit wird eine zuvor angehaltene Aufgabe fortgesetzt. Ähnlich wie bei dem Befehl *Start*, wird nicht auf die Vollendung der Aufgabe gewartet, sondern nur das Fortsetzen initiiert. Es wird eine leere Antwort zur Bestätigung gesendet.
- *Abbruch*: Dieser Befehl erlaubt das vollständige Abbrechen einer zuvor gestarteten Aufgabe. Es können auch angehaltene Aufgaben abgebrochen werden. Auch hier dient eine leere Antwort als Bestätigung.
- *RezeptRunterladen*: Ein Rezept wird vom Sender zum Empfänger gesendet. Unter einem Rezept versteht man eine Menge von Prozessparametern, Ablauffolgen oder auch ganzen Programmteilen oder sogar Programmen, die von der Steuerung der Komponente ausgeführt werden können. Dies führt zu einer Flexibilität, die nicht nur eine Änderung einzelner Aufgaben erlaubt, sondern es auch ermöglicht, neue Aufgaben hinzuzufügen oder nicht mehr benötigte Aufgaben zu löschen. Die Antwort dient als Bestätigung für ein erfolgreiches oder nicht erfolgreiches Runterladen des Rezeptes.
- *RezeptHochladen*: Entsprechend dem Befehl *RezeptRunterladen* dient dieser Befehl dazu, um das aktuelle Rezept einer Komponente zum Absender hochzuladen. Hierbei beinhaltet die Antwort das Rezept der Komponente und der Befehl selbst besitzt keine weiteren Parameter.

b.) Nachrichten

- *AufgabeBeendet*: Mit dieser Nachricht wird dem Leitsystem der Vollzug über eine zuvor auf der Komponente gestarteten Aufgabe mitgeteilt. Als zusätzlicher Parameter wird hier die Aufgabennummer mitgegeben sowie eine Liste von Positionspaaren, welche die Positionsänderungen von Produkten beinhaltet. Hierbei gibt das erste Element des Paares die ursprüngliche Position (q) an, während das zweite Element die neue Position (z) des Produktes angibt. Pro Aufgabe darf sich jedoch immer nur genau ein Produkt verändern. Es dürfen mehrere Produkte zu einem neuen Produkt zusammengefügt werden. Es dürfen aber auch mehrere Produkte ihre Position verändern, ohne jedoch das Produkt zu verändern. Haben keine Positionsänderungen stattgefunden, so ist die Liste leer.

Gegeben sind die Positionspaare $(q_1, z_1), \dots, (q_x, z_x) \quad x \in \mathbb{N}$

$$\forall a,b:(q_a \neq q_b) \Rightarrow (z_a \neq z_b) \vee \forall a,b:(q_a \neq q_b) \Rightarrow (z_a = z_b) \quad 1 \leq a,b \leq x, a \neq b$$

- *FertigZumBeladen*: Hiermit teilt die Komponente dem Leitsystem mit, dass eine physikalische Position bereit ist, ein neues Eingangsprodukt aufzunehmen. Die Position wird als Parameter der Nachricht mitgegeben.
- *FertigZumEntladen*: Diese Nachricht zeigt, im Gegensatz zur letzten Nachricht, dem Leitsystem an, dass eine physikalische Position der Komponente bereit ist, ein Ausgangsprodukt abzugeben. Auch hier wird die Position als Parameter der Nachricht mitgegeben.
- *Alarm*: Wenn in der Komponente etwas Unvorhergesehenes oder ein Zwischenfall geschieht, wird dem Leitsystem mit dieser Nachricht die Information über diesen Zwischenfall mitgeteilt. Die Art des Zwischenfalls wird der Nachricht als Parameter mitgegeben.

5.2.5 Wrapper zur standardisierten Schnittstelle für Komponenten

Obwohl die vorgestellte standardisierte Schnittstelle für Komponenten sehr einfach gehalten ist, kann es Komponenten geben, die diese Schnittstelle nicht erfüllen. Um solche Komponenten dennoch an das Leitsystem anzuschließen, wird nicht eine eigene Schnittstelle zwischen dieser Komponente und dem Leitsystem definiert, sondern es wird ein Wrapper (dt. Umschlag) auf Seiten des Leitsystems eingeführt. Die Aufgabe des Wrappers ist es, die proprietäre Schnittstelle der Komponente auf die

standardisierte Schnittstelle für Komponenten umzusetzen. Hierdurch stellt sich auch eine Komponente mit einer proprietären Schnittstelle dem Leitsystem dar, als ob sie über die standardisierte Schnittstelle für Komponenten verfügen würde.

5.2.6 Vorteile der standardisierten Schnittstelle für Komponenten

Durch die Forderung an die Komponenten nach Einhaltung der standardisierten Schnittstelle für Komponenten wird die Anbindung verschiedenster Komponenten an das Leitsystem sehr einfach. Auch wenn eine Komponente nicht in der Lage ist, diese Schnittstelle zu erfüllen, so wird es dennoch, durch die Einführung des Wrappers, dem Leitsystem so dargestellt, als verfüge sie über diese standardisierte Schnittstelle für Komponenten. Dies erlaubt es, verschiedenste Komponenten an das Leitsystem anzudocken, ohne Anpassungen im Leitsystem vorzunehmen. Dadurch wird die Basis geschaffen, um einen zentralen Punkt der im Kapitel 3 aufgestellten Anforderungen zu erfüllen, nämlich die Möglichkeit, das System auf neue beziehungsweise andere Produkte umzustellen. Dies wird insbesondere auch dadurch gefördert, dass die Komponenten in der Lage sind, Rezepte verarbeiten zu können.

5.3 Logischer Repräsentant

Nachdem in den letzten Abschnitten Komponenten und die standardisierte Schnittstelle für Komponenten, mit deren Hilfe die Komponenten an das Leitsystem angebunden werden, eingeführt wurden, wird hier auf die Einbindung der Komponenten im Leitsystem selbst eingegangen. Hierzu wird für jede Komponente, die an das Leitsystem angeschlossen wird, ein logischer Repräsentant im Leitsystem erzeugt.

5.3.1 Logische Komponente

Auf Seiten des Leitsystems wird zu einer physikalischen Komponente eine logische Komponente eingeführt. Diese stellt das Abbild der physikalischen Komponente im Leitsystem dar. Eine physikalische Komponente ist über die standardisierte Schnittstelle nur mit ihrem Abbild, der logischen Komponente, verbunden.

Die logische Komponente stellt sich somit im Leitsystem als ein abgeschlossenes System dar, welches selbständig h Aufgaben erfüllen kann.

5.3.2 Zustandsmaschine einer Aufgabe

Um über den Zustand der h Aufgaben einer Komponente Informationen zu haben, existiert für jede Aufgabe eine eigene Zustandsmaschine (siehe Abb. 5.4).

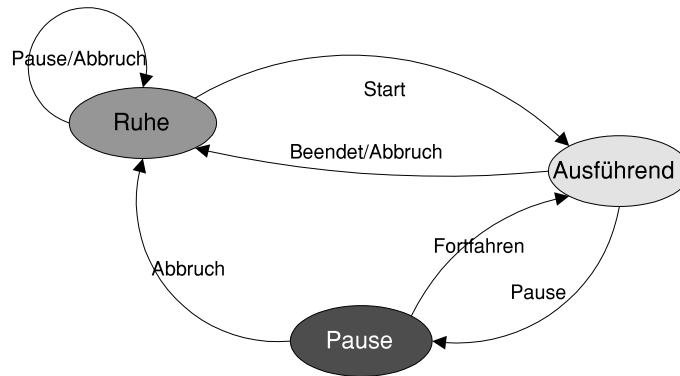


Abb. 5.4: Zustandsmaschine einer Aufgabe

Die Zustandsübergänge entsprechen den Befehlen und Nachrichten, über die eine logische Komponente angesprochen wird. Diese werden von ihr über die standardisierte Schnittstelle für Komponenten an ihre entsprechende physikalische Komponente weitergeleitet. Somit entspricht jede Zustandsmaschine genau einer Aufgabe in der entsprechenden physikalischen Komponente.

5.3.3 Grundlegender Aufbau des logischen Repräsentanten

Der grundlegende Aufbau des logischen Repräsentanten besteht aus der oben beschriebenen logischen Komponente, die über die standardisierte Schnittstelle für Komponenten mit der physikalischen Komponente verbunden ist, und den Zustandsmaschinen für jede der h Aufgaben, die eine Komponente erfüllen kann (siehe Abb. 5.5). Die Zustandsmaschinen werden entsprechend des Zustandes der jeweiligen Aufgabe auf der physikalischen Komponente von der logischen Komponente innerhalb des logischen Repräsentanten gesetzt.

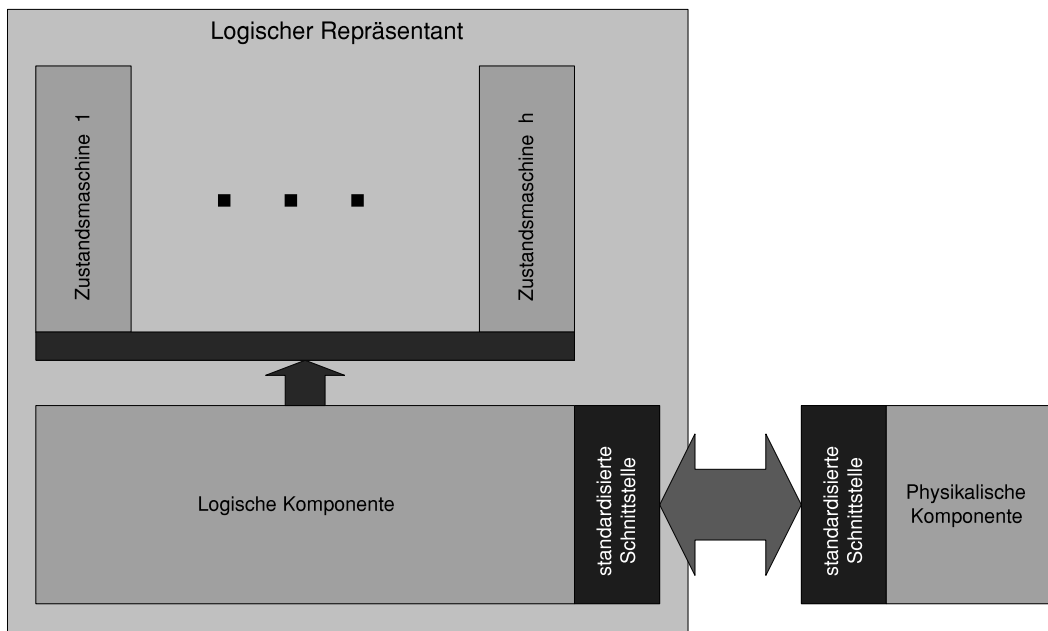


Abb.5.5: *Logischer Repräsentant mit standardisierter Schnittstelle für Komponenten*

Verfügt eine physikalische Komponente nicht über die standardisierte Schnittstelle für Komponenten, so wird in den entsprechenden logischen Repräsentanten noch ein Wrapper zur proprietären Schnittstelle der physikalischen Komponente eingefügt (siehe Abb. 5.6).

Somit existiert im Leitsystem für jede physikalische Komponente genau ein logischer Repräsentant. Bei dem logischen Repräsentanten, wie er hier vorgestellt wurde, handelt es sich allerdings nur um die Basisbeschreibung. Im Laufe der folgenden Abschnitte wird dieser schrittweise erweitert werden.

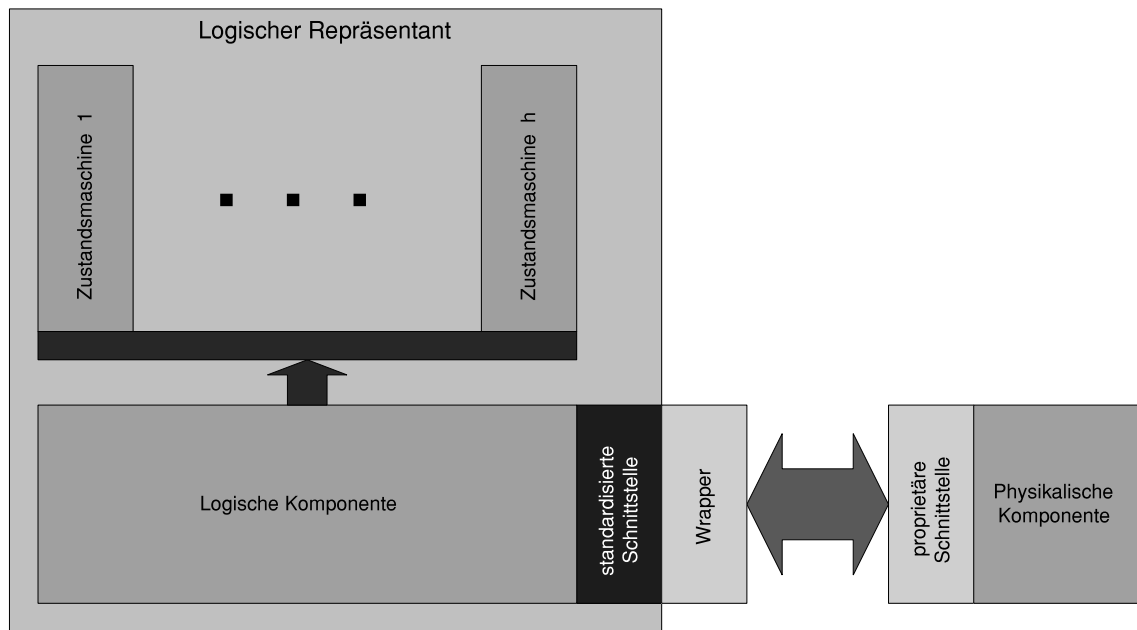


Abb. 5.6: *Logischer Repräsentant mit Wrapper zu proprietärer Komponentenschnittstelle*

5.4 Logische Position

Im Kapitel *Analyse des Montageablaufs und Konzeption einer Montagezelle* wurde ein Werkstückträger eingeführt, der dazu dient, Produkte über die Be-/Entladekomponente in die Montagezelle zu bringen oder sie aus der Montagezelle zu holen. Der Werkstückträger dient als Basis für die Definition der logischen Position eines Produktes.

5.4.1 Definition der logischen Position

Die logische Position gibt an, an welcher Stelle sich ein Produkt in einer Komponente befindet. Es werden jedoch nur Stellen mit logischen Positionen beschrieben, an denen Produkte in eine Komponente hinein beziehungsweise heraus gelangen können. Dabei werden Produkte, die sich äquidistant voneinander auf einer Geraden befinden, durch eine Dimension der logischen Position bestimmt. Jede Stelle auf dieser Geraden, an der sich ein Produkt befinden kann, wird somit über eine natürliche Zahl in einer Dimension der logischen Position bestimmt. Wenn eine logische Position durch mehrere Dimensionen bestimmt wird, so ist es nicht zwingend nötig, dass die verschiedenen Dimensionen senkrecht zueinander stehen. Entscheidend ist nur, dass die Stellen, die ein Produkt innerhalb einer Dimension einnehmen kann, äquidistant

entlang einer Geraden liegen (siehe Abb. 5.7).

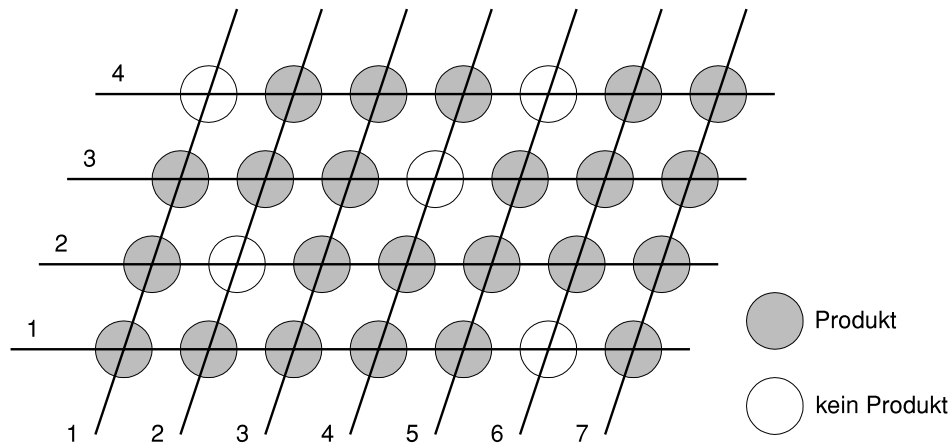


Abb. 5.7: Logische Positionen am Beispiel der Dimension 2

5.4.2 Logische Positionen bei Werkstückträgern

Werkstückträger dienen dazu, Produkte über die Be-/Entladekomponente in die Montagezelle zu bringen und von ihr zu holen. Wie im Kapitel *Analyse des Montageablaufs und Konzeption einer Montagezelle* beschrieben, sind die äußeren Abmessungen eines Werkstückträgers normiert, so dass ungeachtet der Produkte, die ein Werkstückträger fasst, er von allen Be-/Entladekomponenten einer jeden Montagezelle aufgenommen werden kann.

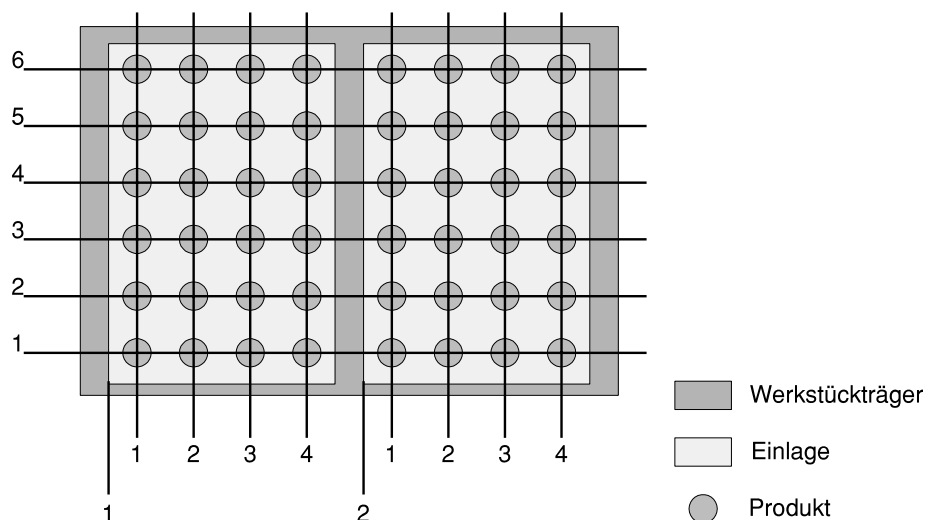


Abb. 5.8: Werkstückträger mit 3-dimensionalen logischen Positionen

Um dennoch unterschiedliche Produkte zu fassen, wurden Einlagen, genauer gesagt zwei Einlagen, die auf jedes Produkt individuell angepasst sind, geschaffen (vgl. Abschnitt 3.5.4). Dies führt jetzt jedoch zu dem Problem, dass, selbst wenn die

Produkte in jeder Einlage in beiden Dimensionen äquidistant platziert sind, die Äquidistanz über die Grenzen der Einlagen hinweg im Allgemeinen nicht vorhanden ist. Deshalb ist hierfür eine dritte Dimension vorzusehen, welche dazu benutzt wird, die Einlage zu bezeichnen (siehe Abb. 5.8).

Innerhalb einer Einlage können Produkte auch in mehreren Lagen aufeinander geschichtet sein, was dazu führt, immer vorausgesetzt, dass die Lagen äquidistant sind, dass hierfür eine dritte Dimension innerhalb einer Einlage nötig ist. Dies bedeutet, dass für die logischen Positionen auf einem Werkstückträger schon vier Dimensionen nötig sind.

Ist eine Äquidistanz zwischen zwei Lagen, Spalten oder Reihen einmal nicht gegeben, so ist auch hier eine zusätzliche Dimension für die logischen Positionen einzuführen. Dies kann dazu führen, dass für einen Werkstückträger i ($i \in \mathbb{N}$) Dimensionen notwendig sind (siehe Abb. 5.9).

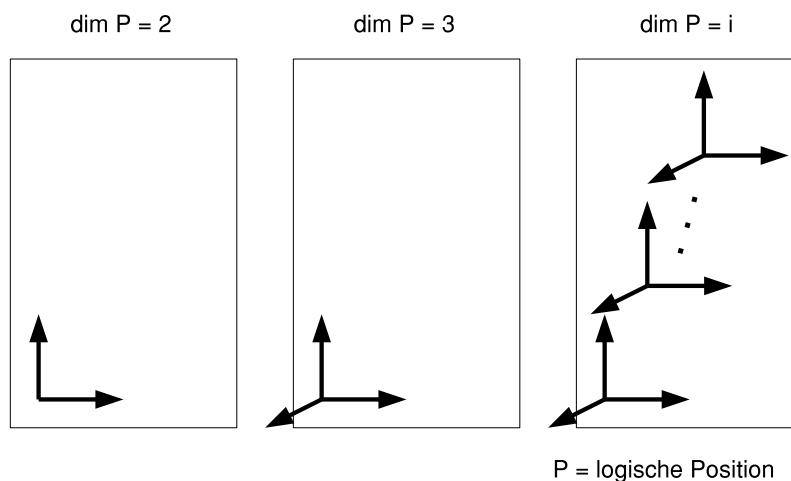


Abb. 5.9: Mögliche Dimensionen für die logischen Positionen bei einer Einlage eines Werkstückträgers

5.4.3 Logische Positionen bei der Be-/Entladekomponente

Die Be-/Entladekomponente dient zur Aufnahme der Werkstückträger, welche von außen in die Montagezelle kommen (vgl. Abschnitt 3.5.2). Innerhalb der Montagezelle werden je nach Handhabungskomponente einzelne Produkte oder mehrere Produkte transportiert.

Da die Be-/Entladekomponente im Allgemeinen mehr als einen Werkstückträger aufnehmen kann, sind auch für sie zusätzliche Dimensionen bei den logischen Positionen vorzusehen (siehe Abb. 5.10). Auch hier ist wichtig, dass Werkstückträger die entlang einer Geraden äquidistant sind, eine Dimension der logischen Positionen

bilden. Dies bedeutet, dass durch die Be-/Entladekomponente noch einmal j ($j \in \mathbb{N}$) Dimensionen zu den logischen Positionen hinzukommen. Die logischen Positionen von Produkten bei der Be-/Entladekomponente können somit durch einen Tupel oder einen Vektor P der Dimension n , wobei $n = i + j$ ($i, j, n \in \mathbb{N}$), beschrieben werden.

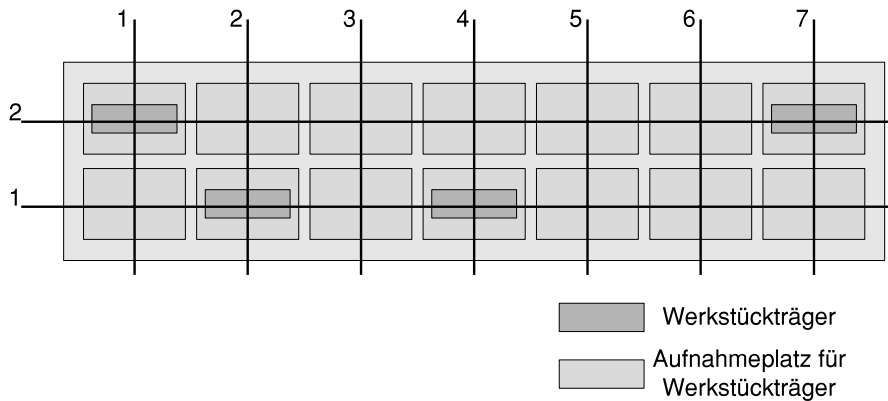


Abb. 5.10: Positionen der Werkstückträger in der Be-/Entladekomponente am Beispiel der Dimension 2

5.4.4 Logische Positionen bei Komponenten allgemein

In Abschnitt 5.4.3 wurde beschrieben, wie man anhand der logischen Position eindeutig ein Produkt in der Be-/Entladekomponente bestimmen kann. Um ein einheitliches Bild bei allen Komponenten zu haben, wird hier gezeigt, wie sich die logischen Positionen auch auf Montagekomponenten und Handhabungskomponenten abbilden lassen.

a.) Montagekomponenten

In Montagekomponenten gehen einzelne oder mehrere Produkte hinein und ein Produkt wieder heraus. Dies bedeutet, dass auch eine Montagekomponente über mehrere physikalische Positionen zur Produktaufnahme oder Produktabgabe verfügen kann. Daraus ergibt sich, dass sie als eigenständige kleine Montagezelle angesehen werden kann. Die Summe aller physikalischen Positionen zur Produktübergabe kann als Be-/Entladekomponente dieser kleinen Montagezelle angesehen werden. Dies bedeutet, dass auch bei Montagekomponenten die logischen Positionen als ein Vektor P der Dimension n angesehen werden können.

b.) Handhabungskomponenten

Diese unterscheiden sich zur Be-/Entladekomponente und zu den Montagekomponenten insofern, dass sie im Allgemeinen nicht stationär sind, sondern sich bewegen und dabei mit der Be-/Entladekomponente und den Montagekomponenten interagieren. Aber auch eine Handhabungskomponente hat ein oder mehrere physikalische Positionen um Produkte aufzunehmen. Der Unterschied liegt nur darin, dass ihr die Produkte nicht gebracht werden, sondern sie diese selbst holt. Aus ihrer Sicht, wenn man die Bewegung hin zum Produkt einmal außer Acht lässt, sind ihre physikalischen Positionen nichts anderes, als eine Be-/Entladekomponente. Dies bedeutet, dass auch sie als eigenständige kleine Montagezelle mit eigener Be-/Entladekomponente angesehen werden kann und somit die entsprechenden logischen Positionen als Vektor P der Dimension n zu sehen sind.

5.4.5 Abbildung der logischen Positionen auf Weltkoordinaten

Bisher wurde nur von logischen Positionen gesprochen. Mit logischen Positionen lässt sich aber in der realen Umgebung der Montagezelle nicht arbeiten. Hierzu werden Weltkoordinaten benötigt. Weltkoordinaten beschreiben die Position und die Orientierung im Raum [Dillmann91]. Um eine logische Position eines Produktes zu beschreiben, sind deshalb drei translatorische und drei rotatorische Koordinaten nötig. Aufgrund der Anforderungen an die Genauigkeit bei der Montage von Festplatten sind die einzelnen Koordinaten im μm -Bereich anzugeben.

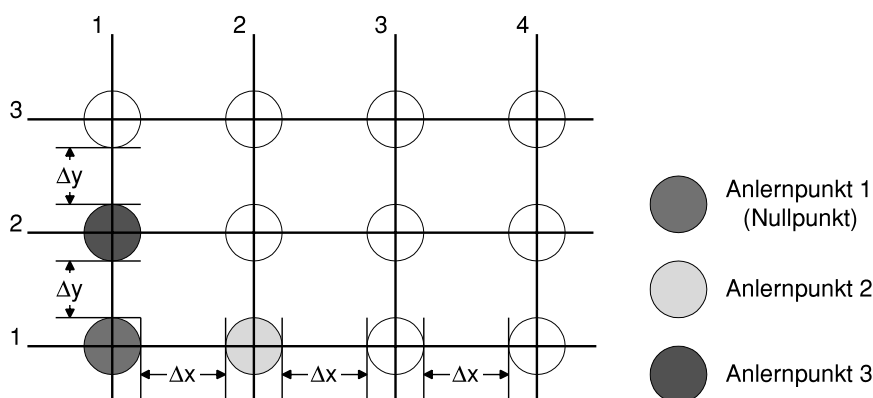


Abb. 5.11: Anlernpunkte von logischen Positionen am Beispiel der Dimension 2

Weltkoordinaten müssen den Komponenten bekannt sein. Dies bedeutet, dass zu jeder logischen Position Weltkoordinaten existieren müssen. Dies wird dadurch erreicht, dass die Weltkoordinaten in den Komponenten angelernt werden. Unter Anlernen

versteht man hierbei, dass die Weltkoordinaten zu jeder einzelnen logischen Position bestimmt werden und diese in der logischen Steuerung der physikalischen Komponente abgelegt werden. Wenn man die Anzahl der Dimensionen und die jeweils möglichen Werte in jeder Dimension einer logischen Position betrachtet, kann dies einen erheblichen Aufwand bedeuten. Um diesen Aufwand zu reduzieren, hilft die Forderung an eine Dimension, dass die Produkte entlang einer Geraden äquidistant sein müssen (siehe Abb. 5.11). Hierdurch lässt sich das Anlernen auf eine Weltkoordinate für jede Dimension und einen Weltkoordinaten-Nullpunkt, der als Referenzpunkt für jede Dimension gilt, reduzieren.

$$n \text{ Dimensionen} \quad \Rightarrow \quad (n+1) \text{ Anlern-Weltkoordinaten}$$

5.4.6 Vorteile der logischen Positionen

Um die Position eines Produktes in Weltkoordinaten anzugeben, werden sechs Koordinaten benötigt. Bei der Definition durch logische Positionen werden n Dimensionen benötigt. Speziell bei der Be-/Entladekomponente kann es vorkommen, dass n auch einmal größer sein kann (vgl. Abschnitt 5.4.3 bis Abschnitt 5.4.5). In der Realität hat sich jedoch gezeigt, dass selbst bei der Be-/Entladekomponente die Dimension immer nur zwischen drei und sechs liegt. Wenn man zusätzlich noch in Betracht zieht, dass es sich bei Weltkoordinaten um reelle Zahlen handelt (oder aber um sehr große ganze Zahlen, wenn als Einheit μm zugrunde gelegt wird) und bei den logischen Positionen um kleine natürliche Zahlen, so ist das Datenvolumen zum Übertragen von logischen Positionen zu dem von Weltkoordinaten deutlich kleiner.

Weltkoordinaten können im Rezept einer Komponente hinterlegt werden und durch die standardisierte Schnittstelle für Komponenten zu diesen herunter geladen werden (vgl. Abschnitt 5.2). Dies kann während des Einrichtens auf ein neues oder anderes Produkt geschehen. Somit hat man durch die Einführung von logischen Positionen die Komplexität beziehungsweise das Datenvolumen während der eigentlichen Montage verringert, indem dies in den Bereich des Einrichtens verschoben wurde.

5.5 Produktverfolgung

Ein wichtiger Punkt bei der Montage von Festplatten ist zu wissen, aus welchen Einzelteilen oder Teilprodukten das Endprodukt montiert wurde. Nur so lassen sich eventuell auftretende Fehlproduktionen von Teilprodukten schnell herausfinden. Deshalb ist die Verfolgung der Einzelteile und der Teilprodukte, das heißt in welchen

Komponenten sie montiert werden, eine wichtige Anforderung an das Leitsystem (vgl. Abschnitt 3.6.2).

5.5.1 Definition des logischen Produktes

Das logische Produkt ist die Abbildung des physikalischen Produktes innerhalb des Leitsystems. Wie schon früher definiert, werden hierbei alle Einzelteile, Teilprodukte und auch das Endprodukt als Produkt bezeichnet. Das logische Produkt enthält alle Informationen über eine Teilverrichtung, die am physikalischen Produkt vorgenommen wurde. Sind zwei oder mehr Produkte zusammengefügt worden, so werden die entsprechenden logischen Produkte vereint, das heißt, alle bis dahin gesammelten Information über das Produkt gehen in das neue zusammengefügte Produkt über. Das Endprodukt, beziehungsweise dessen logisches Produkt, enthält also Informationen über alle Einzelteile und Teilprodukte und über alle Teilverrichtungen, die an oder mit ihnen vorgenommen wurden, so dass, ausgehend vom Endprodukt, jederzeit eine genaue Produktverfolgung stattfinden kann.

5.5.2 Logische Produktpositionen

Um logische Produkte im logischen Repräsentanten einer physikalischen Komponente aufnehmen zu können, werden logische Produktpositionen definiert. Logische Produktpositionen sind eine Abbildung aller Beladestellen und Entladestellen einer physikalischen Komponente innerhalb des entsprechenden logischen Repräsentanten. Somit sind sie auch eine Abbildung der logischen Positionen des Repräsentanten.

Wird ein physikalisches Produkt zu einer Beladestelle einer physikalischen Komponente gebracht (oder geholt, falls es sich um eine aktive Komponente handelt), so wird auch, mit der Ablage des physikalischen Produktes, das logische Produkt an der, der logischen Position entsprechenden, logischen Produktposition abgelegt. So wie das physikalische Produkt zu den verschiedenen physikalischen Komponenten zur Montage wandert, wandert auch das logische Produkt in den entsprechenden logischen Repräsentanten mit.

5.5.3 Zustandsmaschine für eine logische Produktposition

Zusätzlich zur logischen Produktposition selbst ist der Zustand der logischen Produktposition wichtig. Durch den Zustand wird angezeigt, ob ein logisches Produkt sich auf einer logischen Produktposition befindet und somit auch das entsprechende physikalische Produkt auf der entsprechenden physikalischen Position.

Dies führt erst einmal zu den beiden Zuständen *Leer* und *Besetzt*. In Abschnitt 5.2 wurde erklärt, wie die Nachricht *AufgabeBeendet* eine Liste von Positionspaaren mit sendet. Bei den Positionen in den Positionspaaren handelt es sich wiederum um logische Positionen. Für jedes Positionspaar aus der Liste wird wie folgt verfahren:

- (q, z) ist ein Positionspaar
- q, z sind logische Positionen
- Verschiebe logisches Produkt von logischer Produktposition q nach z
- Ändere den Zustand der logischen Produktposition q auf *Leer*
- Ändere den Zustand der logischen Produktposition z auf *Besetzt*

War vor dem Verschieben des logischen Produktes auf der logischen Produktposition z schon ein logisches Produkt vorhanden, so handelte es sich um einen Fügevorgang. Dann wird das logische Produkt von der logischen Produktposition q mit dem logischen Produkt der logischen Produktposition z vereinigt, das heißt, ein neues zusammengefügtes logisches Produkt ist entstanden. Die Vereinigung selbst definiert aber noch kein neues Produkt. Sie führt nur alle Informationen aus den logischen Teilprodukten in einem neuen logischen Produkt zusammen.

War die logische Produktposition z vor dem Verschieben auf *Leer*, so handelt es sich nicht um einen Fügeprozess, sondern zum Beispiel um einen Inspektionsprozess oder einen Reinigungsprozess. Das heißt, es handelt sich um einen Prozess, der auch zu einer Produktänderung führt. Selbst ein reiner Verschiebevorgang, von einer logischen Produktposition auf eine andere innerhalb einer Komponente, führt zu einem neuen logischen Produkt. Die Namensgebung des neuen logischen Produktes geschieht durch Konfiguration.

Neben der Nachricht *AufgabeBeendet* werden die beiden Nachrichten *FertigZumBeladen* und *FertigZumEntladen* von der Komponente an das Leitsystem gesendet (vgl. Abschnitt 5.2). Diese erweitern die Aussage, die man über eine logische Produktposition treffen kann. Dies führt dazu, dass die Zustandsmaschine, welche über die Zustände der logischen Produktpositionen Auskunft gibt, aus den vier Zuständen *Leer*, *Besetzt*, *FertigZumBeladen* und *FertigZumEntladen* besteht (siehe Abb. 5.12). Dabei kann die Nachricht *AufgabeBeendet*, neben dem Verschieben der logischen Produkte, Zustandsänderungen von *Leer* nach *Besetzt* und umgekehrt nach sich ziehen und die beiden Nachrichten *FertigZumBeladen* und *FertigZumEntladen* entsprechend Änderungen in die Zustände *FertigZumBeladen* respektive *FertigZumEntladen*.

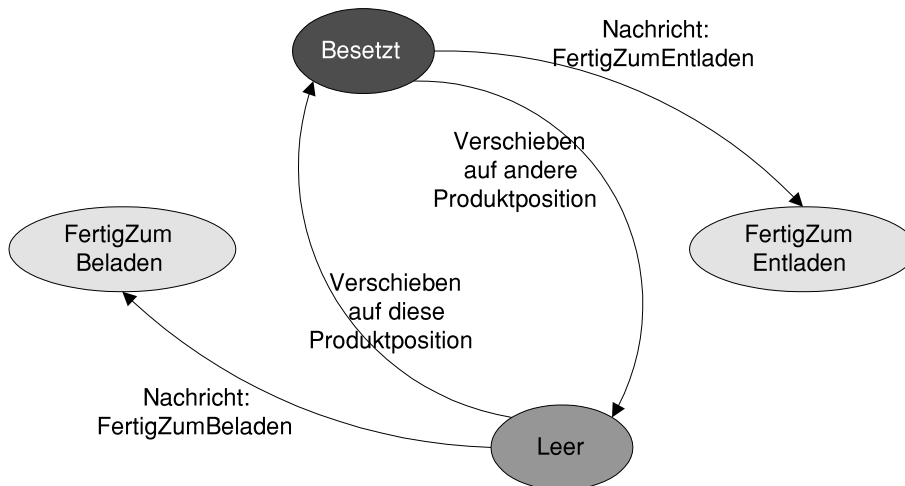


Abb. 5.12: Zustandsmaschine für logische Produktpositionen

5.5.4 Produkttransport zwischen Komponenten

Bisher wurden logische Produkte, und damit auch die physikalischen Produkte, nur innerhalb eines logischen Repräsentanten von einer Position auf eine andere Position verschoben. In Abschnitt 5.1 wurde aber schon von aktiven Komponenten gesprochen, welche Aufgaben besitzen, die eine Interaktion mit anderen Komponenten haben. Dies sind hauptsächlich Handhabungskomponenten, deren Aufgabe es ist, Produkte von einer Komponente zu einer anderen zu bringen.

Da mit dem physikalischen Produkt auch das logische Produkt mitwandern soll, muss ein Mechanismus eingeführt werden, der es erlaubt, logische Produkte von einem logischen Repräsentanten zu einem anderen zu transferieren. Bei einem Transport eines Produktes gibt es immer einen Repräsentanten, der das Produkt holt, beziehungsweise bringt, also der aktive Repräsentant ist. Gleichzeitig gibt es einen passiven Repräsentanten, bei dem das Produkt geholt, beziehungsweise gebracht wird. Deshalb werden folgende Funktionalitäten im logischen Repräsentanten eingeführt:

- *HoleProdukt*: Mit diesem Befehl, der nur im logischen Repräsentanten und nicht auf der physikalischen Komponente existiert, wird ein logisches Produkt vom aufgerufenen logischen Repräsentanten auf den aufrufenden logischen Repräsentanten übertragen.
- *BringeProdukt*: Dieser Befehl überträgt das logische Produkt vom aufrufenden logischen Repräsentanten zum aufgerufenen logischen Repräsentanten.

Bei beiden Befehlen wird jeweils die logische Position für Quelle und Ziel des logischen Produktes auf den logischen Produktpositionen der entsprechenden logischen Repräsentanten angegeben.

Durch diese Erweiterung wird auch die Zustandsmaschine für die logischen Produktpositionen erweitert (siehe Abb. 5.13). Ist eine passive Komponente in dem Zustand *FertigZumBeladen*, so kann eine aktive Komponente ein physikalisches Produkt auf die entsprechende Position bringen. Gleichzeitig ruft der logische Repräsentant der aktiven Komponente den Befehl *BringeProdukt* der passiven Komponente auf. Dies führt zur Übermittlung des logischen Produktes von der aktiven zur passiven Komponente. Entsprechend wird verfahren, wenn die passive Komponente im Zustand *FertigZumEntladen* ist.

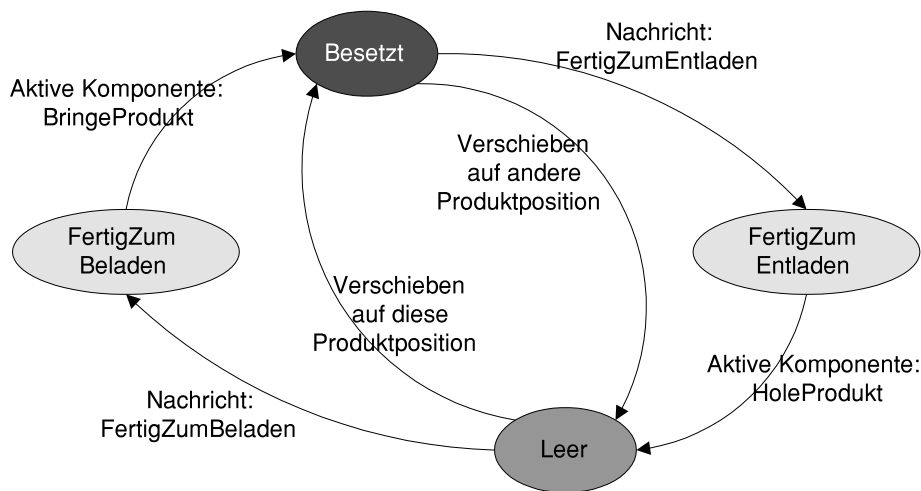


Abb. 5.13: Zustandsmaschine für logische Produktpositionen einer passiven Komponente bei der Interaktion durch eine aktive Komponente

5.5.5 Abbildung der logischen Produktposition auf ihre Zustandsmaschine

In den letzten Abschnitten wurde das logische Produkt erklärt, wie das logische Produkt auf einer logischen Produktposition abgelegt wird und wie dadurch sich der Zustand der logischen Produktposition ändert. Dies hat schließlich zu einer Zustandsmaschine für die logische Produktposition geführt.

Da logische Produktpositionen eine Abbildung der physikalischen Positionen sind und somit auch eine Abbildung der logischen Positionen, handelt es sich bei einer logischen Position auch um einen Vektor der Dimension n . Dies bedeutet, dass alle logischen Produktpositionen zusammen ein n -dimensionales Feld bilden. Die Ausdehnung einer Dimension des Feldes entspricht dem maximalen Wert, den die entsprechende Dimension des logischen Produktpositionsvektors einnehmen kann.

Abbildung 5.14 zeigt dies anhand der Dimension 2 mit der Ausdehnung des Feldes auf die Werte i und j .

Da jede logische Produktposition in diesem Feld ein logisches Produkt aufnehmen kann, folgt daraus, dass es zu jeder logischen Produktposition auch eine Zustandsmaschine gibt. Das heißt, es gibt ein Feld von Zustandsmaschinen, welches in der Dimension und in der Ausdehnung der einzelnen Dimensionen dem Feld der logischen Produktpositionen entspricht. Jeder Vektor aus diesem Feld ist eineindeutig einem Vektor aus dem Feld der logischen Produktpositionen zugeordnet (siehe Abb. 5.14).

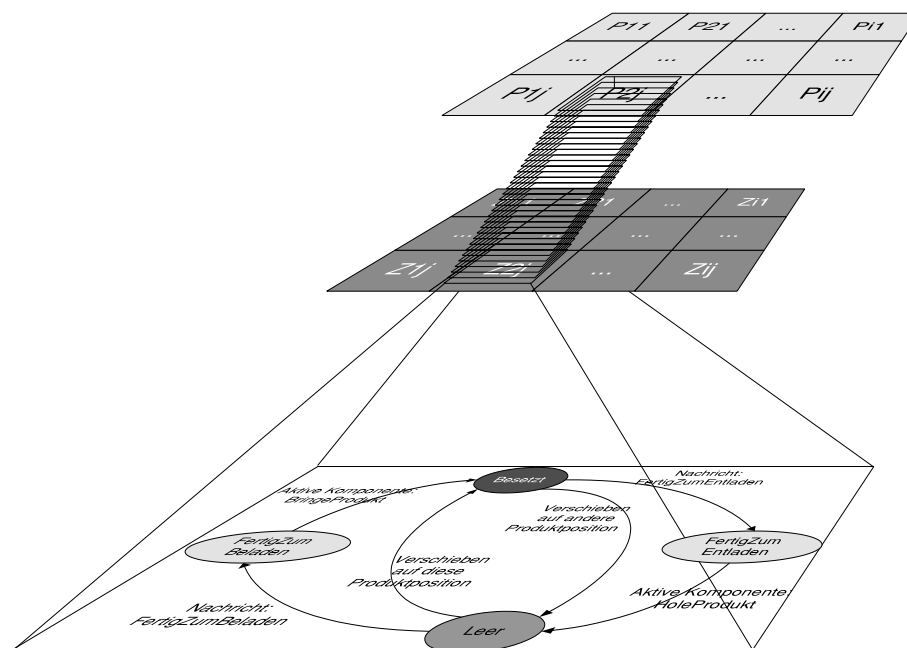


Abb. 5.14: Abbildung der logischen Positionen auf die Zustandsmaschinen am Beispiel der Dimension 2 mit Maximalausdehnung i und j

5.5.6 Erweiterung des logischen Repräsentanten durch die Produktverfolgung

Der in Abschnitt 5.3 dargestellte grundlegende Aufbau des logischen Repräsentanten wird durch zwei n -dimensionale Felder erweitert. Das erste Feld beinhaltet die Produktinformationen selbst, während das zweite Feld den Zustand der jeweils entsprechenden logischen Produktposition anzeigt. Zusätzlich wurde die Möglichkeit geschaffen, auf diese Produktinformationen von außen Einfluss zu nehmen und somit auch den Zustand der Produktpositionen zu beeinflussen (siehe Abb. 5.15).

Eine weitere wichtige Erweiterung ist die Einführung der logischen Positionen für die Abbildung einer jeden physikalischen Produktposition. Dies geschieht sowohl innerhalb des logischen Repräsentanten als auch bei der Kommunikation zwischen der logischen Komponente des logischen Repräsentanten und der physikalischen Komponente, welche über die standardisierte Schnittstelle für Komponenten geht.

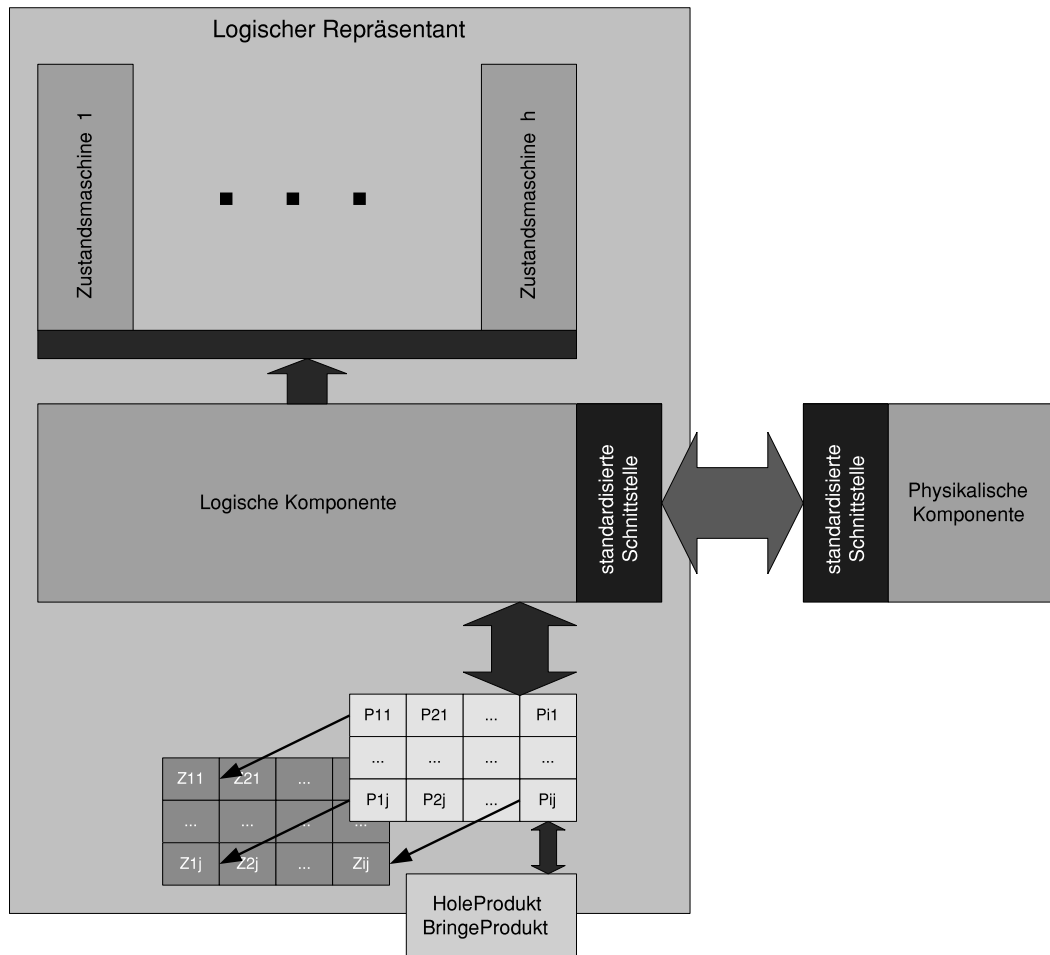


Abb. 5.15: Logischer Repräsentant nach Erweiterung mit Produktverfolgung am Beispiel der Dimension 2

5.6 Mathematisches Modell der Aufgaben

Eine Aufgabe ist bisher als eine Verschiebung von logischen Produkten auf logischen Produktpositionen innerhalb oder zwischen logischen Repräsentanten betrachtet worden. Selbst Fügevorgänge sind in dieser Art betrachtet worden (vgl. Abschnitt 5.5.3). In den folgenden Abschnitten wird ein mathematisches Modell vorgestellt, welches die Verschiebungen der logischen Produkte abbildet.

5.6.1 Abbildung einer Komponentenaufgabe durch eine Transformationsmatrix

Um die Abbildung besser zu veranschaulichen, wird zuerst eine Aufgabe einer aktiven Komponente betrachtet, welche eine Interaktion mit einer anderen Komponente hat. Genauer gesagt, wird eine Handhabungskomponente betrachtet, die ein Produkt bei einer anderen Komponente aufnimmt oder dort ablegt. Erst danach wird auf eine passive Komponente eingegangen, um am Schluss eine allgemeingültige Abbildung zu finden.

a.) Transformationsmatrix bei einer aktiven Komponente

Angenommen, eine Aufgabe einer aktiven Komponente besteht darin, ein Produkt von einer passiven Komponente aufzunehmen. Dies bedeutet, dass der logische Repräsentant der aktiven Komponente dafür verantwortlich ist, das logische Produkt von dem logischen Repräsentanten der passiven Komponente auf den logischen Repräsentanten der aktiven Komponente zu verschieben. Auf beiden Seiten wird die logische Produktposition durch eine logische Position abgebildet. Das heißt, auf der Seite der passiven Komponente existiert ein Vektor $P=(p_1 p_2 \dots p_n)$ der Dimension n und auf der Seite der aktiven Komponente ein Vektor $Q=(q_1 q_2 \dots q_m)$ der Dimension m . Das Verschieben des logischen Produktes besteht somit aus einer Transformation des Vektors P auf den Vektor Q und sieht wie folgt aus:

$$\begin{pmatrix} p_1 & p_2 & \dots & p_n \end{pmatrix} \cdot \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1m} \\ r_{21} & r_{22} & \dots & r_{2m} \\ \dots & \dots & \dots & \dots \\ r_{n1} & r_{n2} & \dots & r_{nm} \end{pmatrix} = \begin{pmatrix} q_1 & q_2 & \dots & q_m \end{pmatrix}$$

oder vereinfacht notiert

$$P^{1 \times n} \cdot R^{n \times m} = Q^{1 \times m}$$

Wird eine Aufgabe der aktiven Komponente betrachtet, welche ein Produkt bei der passiven Komponente ablegt, so entspricht das Verschieben des logischen Produktes einer Transformation des Vektors Q auf den Vektor P :

$$Q^{1 \times m} \cdot S^{m \times n} = P^{1 \times n}$$

Die beiden Transformationsmatrizen R (Verschieben des logischen Produktes von der logischen Produktposition P auf die logische Produktposition Q) und S (Verschieben des logischen Produktes von Q nach P) können folgendermaßen in

Relation zueinander gebracht werden:

$$\begin{aligned}
 & P^{l \times n} \cdot R^{n \times m} = Q^{l \times m} \\
 \Rightarrow & P^{l \times n} \cdot R^{n \times m} \cdot S^{m \times n} = Q^{l \times m} \cdot S^{m \times n} \\
 \Rightarrow & R^{n \times m} \cdot S^{m \times n} = I_n \quad (\text{Einheitsmatrix der Dim. } n) \\
 \Rightarrow & S^{m \times n} = (R^{n \times m})^{-1}
 \end{aligned}$$

b.) Transformationsmatrix bei einer passiven Komponente

Eine Aufgabe einer passiven Komponente hat, im Gegensatz zu einer Aufgabe einer aktiven Komponente, keine Interaktion mit einer anderen Komponente. Eine Abbildung findet also nur innerhalb der Komponente statt. Aber auch hier wird ein logisches Produkt von einer logischen Produktposition auf eine andere logische Produktposition verschoben (Fügebauaufgaben werden später gesondert betrachtet). Da dies allerdings nur innerhalb dieser passiven Komponente stattfindet, handelt es sich um eine Transformation von einem Vektor P der Dimension n auf einen Vektor Q auch der Dimension n . Bei der Transformationsmatrix handelt es sich somit um eine quadratische Matrix der Dimension n .

$$P^{l \times n} \cdot T^{n \times n} = Q^{l \times n}$$

Betrachtet man nun noch Aufgaben, bei denen das logische Produkt nicht von seiner logischen Produktposition wegbewegt wird, so handelt es sich um eine Transformation von dem Vektor P der Dimension n auf denselben Vektor P .

$$P^{l \times n} \cdot I^{n \times n} = P^{l \times n} \quad (I \text{ ist die Einheitsmatrix der Dimension } n)$$

Die Betrachtung der Transformationsmatrizen zuerst für aktive Komponenten und anschließend für passive Komponenten hat gezeigt, dass die Abbildung bei passiven Komponenten als Spezialisierung der aktiven Komponenten angesehen werden kann. Insbesondere ist dies bei Aufgaben zu sehen, die kein Verschieben eines Produktes nach sich ziehen. Bei diesen handelt es sich um die Identitätsabbildung.

5.6.2 Detaillierung der Transformationsmatrix

Bevor die Abbildung für Fügebauaufgaben erklärt werden kann, ist es nötig, eine Fügebauaufgabe genauer zu betrachten. Es wurde gezeigt, dass alle Aufgaben, seien es Aufgaben von aktiven oder passiven Komponenten, durch folgende Abbildung dargestellt werden können:

$$P^{1 \times n} \cdot R^{n \times m} = Q^{1 \times m}$$

wobei für passive Komponenten gilt:

$$n = m$$

Eine Aufgabe die durch eine Transformationsmatrix R abgebildet werden kann, besteht intern aus i ($i \in \mathbb{N}$) verschiedenen Aufgaben, die nach außen allerdings nicht sichtbar sind (siehe Abb. 5.16).

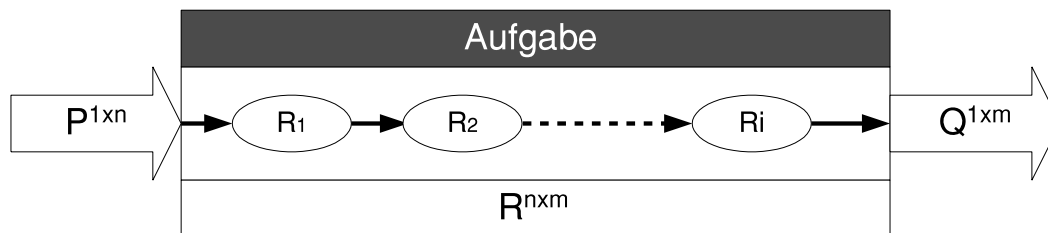


Abb. 5.16: Interne Ansicht einer Aufgabe

Die Abbildungen $R_1 \dots R_i$ sind kleine Abbildungsschritte, welche innerhalb der Aufgabe ausgeführt werden und schließlich zur Abbildung R der Gesamtaufgabe führen. $R_1 \dots R_i$ und R stehen folgendermaßen in Beziehung:

$$R^{n \times m} = R_1^{l_1 \times l_2} \cdot R_2^{l_2 \times l_3} \cdot R_3^{l_3 \times l_4} \cdot \dots \cdot R_i^{l_i \times l_{i+1}}$$

$$l_1 = n, \quad l_{i+1} = m$$

R besteht also aus der Matrixmultiplikation von $R_1 \dots R_i$. Ein Produkt kann somit, im Verlaufe einer Aufgabe, innerhalb einer Komponente verschiedene Positionen einnehmen, bis es schließlich am Ende der Aufgabe auf der Ausgangsposition zum Liegen kommt. Diese Positionen müssen nicht zwingend definierte logische Produktpositionen sein. Wie schon früher gesehen, bedeutet dies für ein Produkt, welches sich im Laufe der Aufgabe nicht von seiner ursprünglichen Position wegbewegt, dass $i=1$ ist und es sich somit bei R um die Einheitsmatrix handelt.

Wichtig ist hier noch einmal der Hinweis, dass es sich bei den ganzen Abbildungen nur darum handelt, wie ein logisches Produkt von einer logischen Produktposition zu einer anderen gelangt, und somit auch das physikalische Produkt, und nicht um Prozesse die am oder mit dem Produkt geschehen. Ganz deutlich ist dies bei Aufgaben von Handhabungskomponenten zu sehen, die ein Produkt von einem Ort zu einem anderen Ort bewegen, aber das Produkt immer in demselben Greifer und somit auf derselben logischen Produktposition behalten. Solche Aufgaben haben als Abbildung die Identitätsabbildung und damit als Transformationsmatrix die Einheitsmatrix.

5.6.3 Transformation bei Fügeaufgaben

In den bisherigen Abschnitten wurden keine Fügeaufgaben betrachtet. Eine Fügeaufgabe besteht darin, mehrere Eingangsprodukte zu einem Ausgangsprodukt zusammenzufügen. Dies bedeutet, dass es k ($k \in \mathbb{N}$) logische Produktpositionen gibt, die durch die Vektoren $P_1 \dots P_k$ mit den Dimensionen $n_1 \dots n_k$ abgebildet werden. Auf diesen k logischen Produktpositionen liegen die logischen Eingangsprodukte. Darüber hinaus gibt es eine logische Produktposition, die durch den Vektor Q der Dimension m abgebildet wird, auf der das logische Ausgangsprodukt zum Liegen kommen soll (siehe Abb. 5.17).

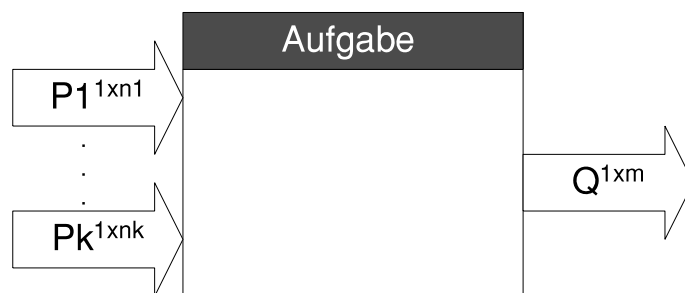


Abb. 5.17: Eingangsprodukte und Ausgangsprodukte einer Fügeaufgabe

Da es k Eingangsprodukte gibt und für alle eine Transformation durchgeführt werden muss, gibt es auch k Transformationsmatrizen $R_1 \dots R_k$. Nachdem alle logischen Produktpositionen der Eingangsprodukte auf die eine logische Produktposition des Ausgangsproduktes abgebildet werden, lassen sich daraus folgende Abbildungsgleichungen herleiten:

$$P_1^{1 \times n_1} \cdot R_1^{n_1 \times m} = Q^{1 \times m}$$

$$P_2^{1 \times n_2} \cdot R_2^{n_2 \times m} = Q^{1 \times m}$$

...

$$P_k^{1 \times n_k} \cdot R_k^{n_k \times m} = Q^{1 \times m}$$

Betrachtet man eine Fügeaufgabe etwas genauer, so stellt man fest, dass die einzelnen Produkte während der Aufgabe irgendwann zusammengefügt werden. Wie im letzten Abschnitt erklärt, finden in einer Aufgabe mehrere kleine Abbildungsschritte statt. Zwei oder mehr Produkte werden also innerhalb derselben Aufgabe nach mehreren unabhängigen kleinen Abbildungsschritten zusammengefügt und durchlaufen fortan dieselben Abbildungsschritte bis zum Ende der Aufgabe. Bis dahin können auch wieder andere Eingangsprodukte oder auch schon zusammengefügte Produkte bei

einem dieser Abbildungsschritte hinzukommen (siehe Abb. 5.18).

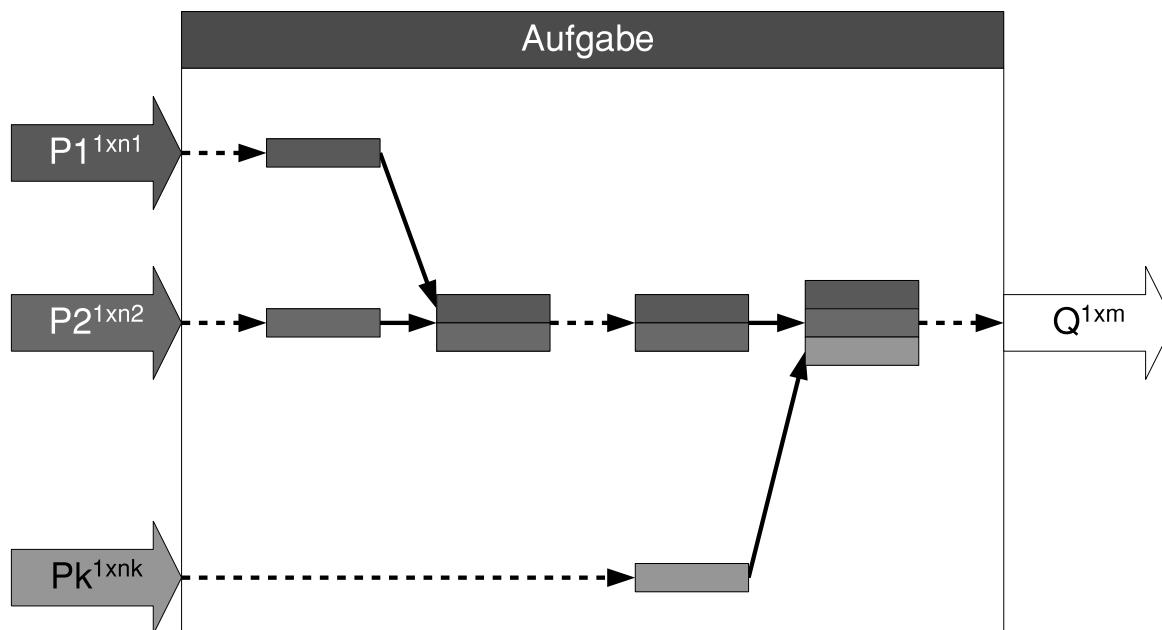


Abb. 5.18: Beispiel der Abbildungsschritte einer Fügeaufgabe mit k Eingangsprodukten

Abbildung 5.18 zeigt beispielhaft, wie die Abbildungsschritte innerhalb einer Fügeaufgabe aussehen. Um dies mathematisch zu zeigen, ist es nötig, eine Vereinfachung einzuführen. Diese Vereinfachung ist keine Einschränkung, sondern wird nur durchgeführt, um die folgenden Formeln einfach zu halten.

$$n_1 = n_2 = \dots = n_k = n \quad \Rightarrow$$

$$P_1^{1 \times n} \cdot R_1^{n \times m} = Q^{1 \times m}$$

...

$$P_k^{1 \times n} \cdot R_k^{n \times m} = Q^{1 \times m}$$

Wie in einem früheren Abschnitt gezeigt wurde, kann eine Abbildung einer Aufgabe, das heißt die Transformationsmatrix R , durch die Abbildungsschritte $R_1 \dots R_i$ innerhalb der Aufgabe beschrieben werden. Wendet man dies auf die oben aufgeführten Abbildungsgleichungen an, so erhält man:

$$R_1^{n \times m} = R_{i_1}^{l_1 \times l_2} \dots R_{i_k}^{l_{i_k} \times l_{i_k+1}} ; l_1 = n , l_{i_k+1} = m$$

$$R_k^{n \times m} = R_{i_1}^{l_{k_1} \times l_{k_2}} \dots R_{i_k}^{l_{i_k} \times l_{i_k+1}} ; l_{k_1} = n , l_{i_k+1} = m$$

Wenn man die in Abbildung 5.18 beispielhaft erklärten Abbildungsschritte auf die allgemeinen Transformationsmatrizen $R1 \dots Rk$ anwendet, so gilt für beliebige zwei Eingangsprodukte:

$$a \in [1..k], b \in [1..k], a \neq b$$

$$\Rightarrow Pa^{1 \times n} \cdot Ra^{n \times m} = Q^{1 \times m}, Pb^{1 \times n} \cdot Rb^{n \times m} = Q^{1 \times m}$$

$$\Rightarrow \exists (u_a \leq i_a, v_b \leq i_b) \forall (u \in [u_a..i_a], v \in [v_b..i_b]): (Ra_u^{i_a \times i_a} = Rb_v^{i_b \times i_b})$$

5.6.4 Gleichsetzung aller Aufgaben

In den letzten Abschnitten wurde gezeigt, dass alle Aufgaben von Komponenten, seien es aktive oder passive Komponenten, durch Abbildungen mit Transformationsmatrizen beschrieben werden können. Genauer gesagt, wurde die Bewegung der logischen Produkte zwischen den logischen Produktpositionen durch Transformationsmatrizen beschrieben. Entscheidend ist hierbei, dass alle Aufgaben (Handhabungsaufgaben, einfache Prozessaufgaben und Fügeaufgaben) durch Transformationsmatrizen abzubilden sind. Dies bedeutet, dass alle Aufgaben im Weiteren gleich behandelt werden können. Es ist nicht nötig, einen Unterschied zwischen den Aufgaben zu machen.

5.7 Komponentenauftrag

Die Aufgaben von Komponenten sind bisher als gegeben angenommen worden. Um nun aber eine Aufgabe einer Komponente zu starten, bedarf es eines Komponentenauftrages. Dieser ist dafür verantwortlich, dass die ihm zugeteilte Aufgabe unter noch zu definierenden Bedingungen gestartet wird.

Einem Komponentenauftrag ist genau eine Aufgabe seiner Komponente zugeordnet. Darüber hinaus besitzt er eine Logik, durch die der Start der ihm zugeordneten Aufgabe ausgelöst wird. Komponentenaufträge können sich allein durch diese Logik unterscheiden. Das heißt, es kann mehrere Komponentenaufträge geben, denen dieselbe Aufgabe zugeordnet ist (siehe Abb. 5.19). Die Umkehrung, nämlich dass mehrere Aufgaben einem Komponentenauftrag zugeordnet sind, ist allerdings nicht gültig.

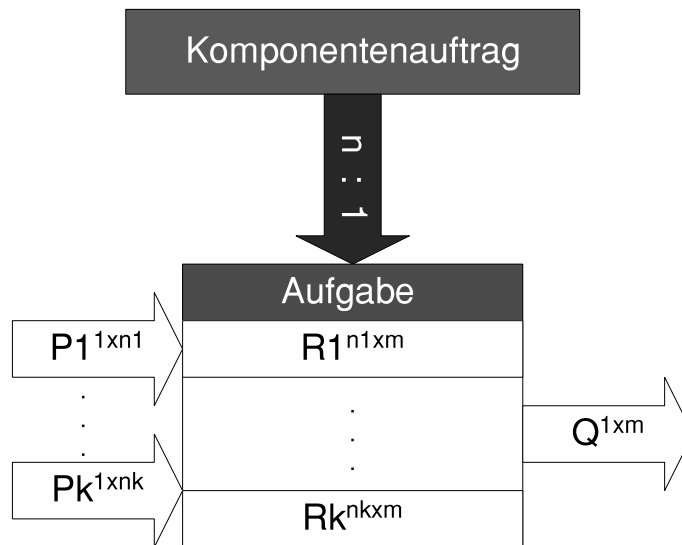


Abb. 5.19: Zuordnung der Aufgabe zum Komponentenauftrag

Nachdem eine Aufgabe aus k Transformationsmatrizen $R1 \dots Rk$ besteht, welche die k Eingangsprodukte auf ein Ausgangsprodukt abbilden, folgt daraus, dass einem Komponentenauftrag k Eingangsprodukte und genau ein Ausgangsprodukt zugeordnet sind (siehe Abb. 5.19).

Nachdem es für jede Aufgabe einer Komponente mehrere Komponentenaufträge geben kann und jede Komponente über viele Aufgaben (vgl. Abschnitt 5.1) verfügen kann, ergibt sich, dass in jedem logischen Repräsentanten beliebig viele Komponentenaufträge existieren können. Um diese Fülle an Komponentenaufträgen zu verwalten, wird im logischen Repräsentanten eine Management Komponente für Komponentenaufträge eingeführt. Diese hat, neben der Verwaltung der Aufträge, auch die Aufgabe, die Verbindung zu den Zustandsmaschinen der Komponentenaufgaben und zur logischen Komponente, und damit auch zur physikalischen Komponente, herzustellen (siehe Abb. 5.20; Die Verbindung der logischen Komponente zur physikalischen Komponente über die standardisierte Schnittstelle für Komponenten wurde in der Abbildung weggelassen).

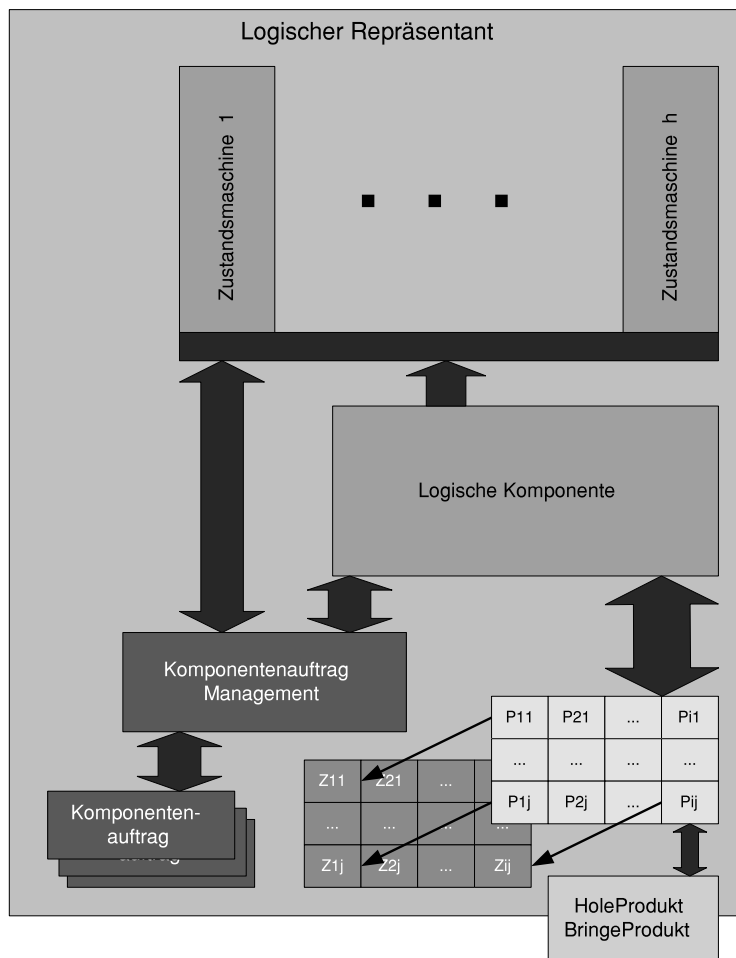


Abb. 5.20: Erweiterter logischer Repräsentant mit Komponentenauftrags-Management

5.8 Zustandsnachrichten

Im letzten Abschnitt wurde der Komponentenauftrag eingeführt und beschrieben, dass die ihm zugeteilte Aufgabe unter definierten Bedingungen gestartet wird. Diese definierten Bedingungen werden durch Zustandsnachrichten erreicht, die im Folgenden beschrieben werden.

5.8.1 Definition einer Zustandsklasse

Bevor Zustandsnachrichten definiert werden und deren Erzeugung gezeigt werden kann, ist es nötig, Zustandsklassen zu definieren. In den letzten Abschnitten wurden verschiedene Zustandsmaschinen eingeführt. Da war zum einen die Zustandsmaschine für eine Aufgabe einer logischen und damit auch einer physikalischen Komponente

(vgl. Abschnitt 5.3.2). Jeder Aufgabe ist hierbei eine Zustandsmaschine zugeordnet. Zum anderen gab es noch die Zustandsmaschine für die logischen Produktpositionen (vgl. Abschnitt 5.5.3), welche auch den physikalischen Produktpositionen entsprechen. Neben diesen explizit eingeführten Zustandsmaschinen gibt es auch noch andere Zustandsmaschinen innerhalb des logischen Repräsentanten. In Abschnitt 5.5 wurde das logische Produkt eingeführt. Das logische Produkt enthält alle Informationen über das physikalische Produkt. Es enthält also auch die Informationen, ob das Produkt gut oder schlecht ist, oder ob es noch einmal einer Bearbeitung des Produktes bedarf. Diese Informationen werden hier auch als verschiedene Zustände einer Zustandsmaschine betrachtet, auch wenn diese Zustandsmaschine nicht explizit eingeführt wurde. Gleiches gilt auch für die Zustände der logischen und damit auch der physikalischen Komponente. Wenn eine physikalische Komponente einen Zwischenfall hat (vgl. Abschnitt 5.2), wird dies als Nachricht der logischen Komponente mitgeteilt. Hiermit wird die logische Komponente in den Zustand *Alarm* versetzt. Das heißt, auch in der logischen Komponente gibt es verschiedene Zustände und somit eine Zustandsmaschine.

Eine Zustandsklasse (ZK) ist eine Menge von i ($i \in \mathbb{N}$) verschiedenen Zustandsmaschinen (z) eines logischen Repräsentanten.

$$ZK(z_1, \dots, z_i) \quad i \in \mathbb{N}$$

Je nach Bedarf können verschiedene Zustandsklassen definiert werden. Innerhalb eines Leitsystems sollten jedoch alle dem Leitsystem zugrunde liegenden logischen Repräsentanten die gleichen Zustandsklassen definiert haben.

5.8.2 Definition und Erzeugung von Zustandsnachrichten

Nachdem die Zustandsklassen definiert sind, können die Zustandsnachrichten daraus abgeleitet werden. Eine Zustandsnachricht (ZN) ist eine Änderung der Zustände in einer Zustandsklasse:

$$ZN = \Delta ZK(z_1, \dots, z_i) \quad i \in \mathbb{N}$$

Ein logischer Repräsentant erzeugt somit die Zustandsnachrichten $ZN_1 \dots ZN_j$ ($j \in \mathbb{N}$). Die Anzahl der Zustandsnachrichten j ist die Summe aller Zustandsübergänge der Zustandsmaschinen $z_1 \dots z_i$ der entsprechenden Zustandsklasse. Abbildung 5.21 zeigt den logischen Repräsentanten mit den verschiedenen Zustandsmaschinen, die Einfluss auf die erzeugten Zustandsnachrichten haben.

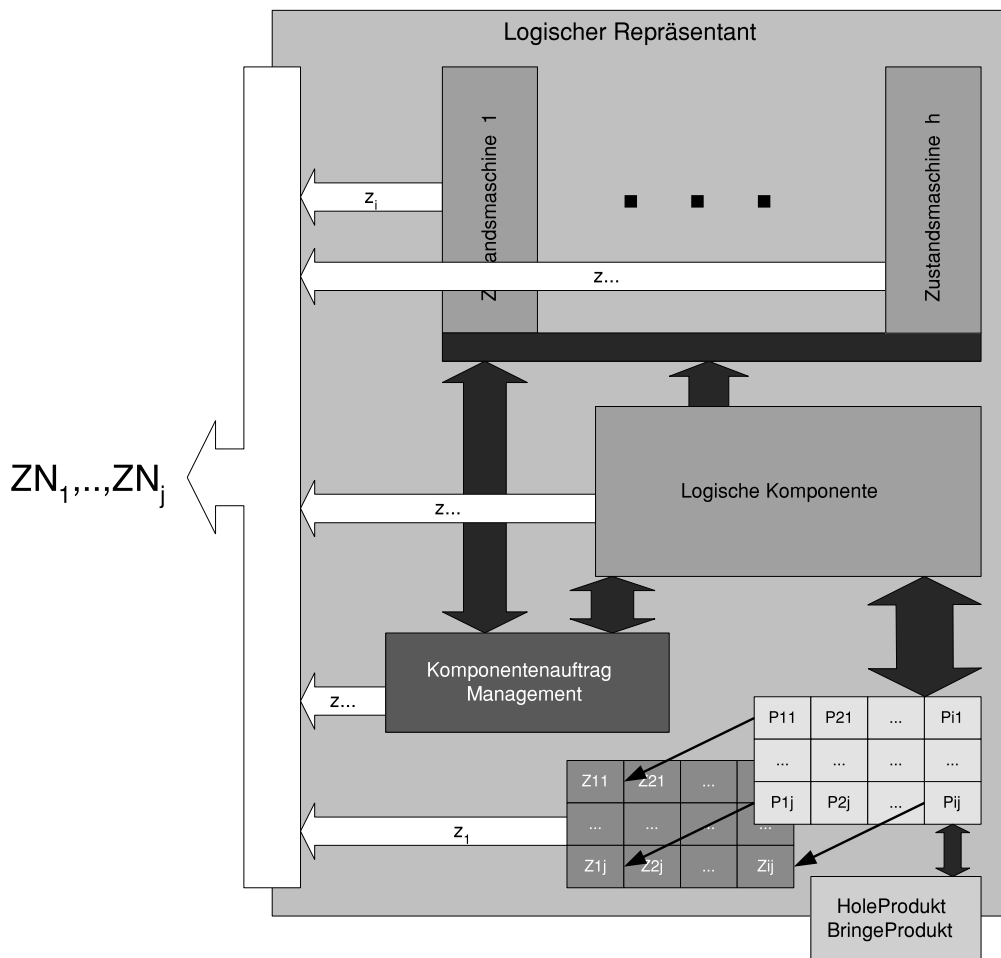


Abb. 5.21: Erzeugung der Zustandsnachrichten eines logischen Repräsentanten

Über die eigentliche Änderung der Zustände in der Zustandsklasse hinaus beinhaltet eine Zustandsnachricht noch zusätzliche Informationen, die mit der aufgetretenen Änderung in Zusammenhang stehen. Dies sind zum Beispiel bei Zustandsänderungen einer logischen Produktposition die logische Produktposition selbst, der Produktzustand und Informationen über das logische Produkt.

5.8.3 Einfluss der Zustandsnachrichten auf den Komponentenauftrag

Wie schon erwähnt, wird der Start der dem Komponentenauftrag zugeordneten Aufgabe durch das Eintreffen definierter Zustandsnachrichten erreicht. Um dies genauer zu erläutern, wird erst einmal die Zustandsnachrichtenanforderung (ZNA) eingeführt. Die ZNA sieht wie folgt aus:

$$ZNA = (A, ZN_i) \quad i \in [1..j]$$

A = Absender der Zustandsnachricht

ZN_i = Zustandsnachricht i des Absenders A

Die ZNA ist also ein Paar aus Absender (logischer Repräsentant) und einer Zustandsnachricht des Absenders. Die Zustandsnachricht darf hierbei als Schablone verstanden werden. Sie besteht nur aus der Änderung eines Zustandes der entsprechenden Zustandsklasse selbst. Ihr sind keine, wie im letzten Abschnitt beschriebenen, zusätzlichen Informationen angehängt. Darüber hinaus kann jede ZNA die beiden Zustände *erfüllt* und *nicht erfüllt* annehmen, welche durch die Bezeichnung „Erfüllung der ZNA “ auch mit den aussagenlogischen Begriffen *wahr* und *falsch* ausgedrückt werden können. Im weiteren Verlauf wird für die „Erfüllung der ZNA “ nur noch der Begriff ZNA verwandt.

Jeder Komponentenauftrag besitzt eine Menge von solchen ZNA . Zu Beginn sind alle ZNA *falsch*. Erreicht eine Zustandsnachricht eines logischen Repräsentanten diesen Komponentenauftrag, so wird überprüft, ob diese Zustandsnachricht einer ZNA aus der Menge der im Komponentenauftrag definierten ZNA entspricht, das heißt Absender und Zustandsnachricht (ohne Zusatzinformationen) müssen übereinstimmen. Ist dies der Fall, so wird die entsprechende ZNA im Komponentenauftrag auf *wahr* gesetzt. Sind alle ZNA erfüllt, das heißt, alle ZNA sind *wahr*, so wird die entsprechende Aufgabe des Komponentenauftrages gestartet (siehe Abb. 5.22).

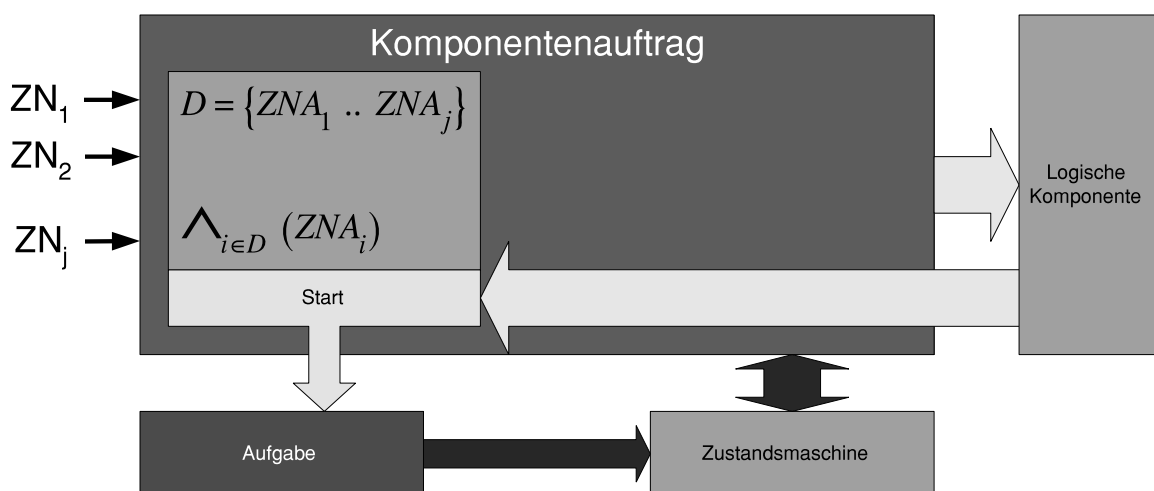


Abb. 5.22: Start einer Aufgabe in einem Komponentenauftrag

Jeder Komponentenauftrag wird also durch eine Menge von Zustandsnachrichten und deren Absender konfiguriert. Erst wenn alle diese Zustandsnachrichten eingetroffen sind, wird die Aufgabe des Komponentenauftrages gestartet.

5.8.4 Rücksetzen der Zustandsnachrichtenanforderungen in einem Komponentenauftrag

In den letzten beiden Abschnitten wurde gezeigt, wie Zustandsnachrichten erzeugt werden und welchen Einfluss sie auf die Ausführung der Aufgabe eines Komponentenauftrages haben. Sind alle *ZNA wahr*, so wird die Aufgabe gestartet. Obwohl während der Ausführung der Aufgabe Nachrichten von der physikalischen Komponente zur logischen Komponente gelangen und somit der logische Repräsentant neue Zustandsnachrichten erzeugt, ist erst nach Beendigung der Aufgabe ein stabiler Zustand erreicht. Das bedeutet, dass erst nach Abschluss der Aufgabe keine weiteren Zustandsnachrichten zu erwarten sind, abgesehen von Alarmnachrichten, welche einen Komponentenauftrag beeinflussen. Das wiederum hat zur Folge, dass erst nach der Beendigung einer Aufgabe ein anderer Komponentenauftrag seine Aufgabe starten darf. Wird dies nicht beachtet, könnte eine *ZNA*, während der Ausführung der Aufgabe, durch eine Zustandsnachricht eines anderen logischen Repräsentanten plötzlich nicht mehr erfüllt sein. Wenn man davon ausgeht, dass die Zustandsnachrichten meist einen physikalischen Bezug in der Komponente, von deren logischen Repräsentanten sie abstammen, haben, so kann dies zu erheblichen Schäden an den physikalischen Komponenten führen.

Um nun aber eine parallele Ausführung von Aufgaben, sei es auf derselben Komponente oder einer anderen, zu erreichen und dennoch Schäden von der Komponente fernzuhalten, bedarf es einer Erweiterung der *ZNA*.

Bevor jedoch die Erweiterung erklärt wird, ist es nötig, noch einmal auf die Erfüllung der *ZNA* einzugehen, wenn eine Zustandsnachricht ankommt (vgl. Abschnitt 5.8.3). Zu jeder dieser Zustandsnachrichten existiert eine negierte Zustandsnachricht. So wie die Zustandsnachricht die entsprechende *ZNA* auf *wahr* setzt, so setzt die negierte Zustandsnachricht die *ZNA* auf *falsch*. Somit kann eine schon erfüllte *ZNA* durch die entsprechende negierte Zustandsnachricht wieder nicht erfüllt sein. Zustandsnachrichten sind die einzigen Informationen, die zwischen verschiedenen logischen Repräsentanten verschickt werden. Auch wenn in einer *ZNA* der Absender explizit angegeben ist, so beinhaltet eine Zustandsnachricht immer ihren ursprünglichen Absender und somit Erzeuger der Zustandsnachricht.

Bisher bestand die *ZNA* aus dem Absender und der Zustandsnachricht des Absenders. Nun wird sie noch erweitert um ein Attribut *Rücksetzen bei Ausführung*. Sie sieht somit folgendermaßen aus:

$$ZNA = (A, ZN_i, R) \quad i \in [1..j] \quad R \in \{wahr, falsch\}$$

A = Absender der Zustandsnachricht

ZN_i = Zustandsnachricht i des Absenders A

R = Rücksetzen bei Ausführung

Ist das Attribut R auf *wahr* gesetzt, so wird, zusammen mit dem Start der Aufgabe, die der ZNA entsprechende negierte Zustandsnachricht abgesendet. Dies hat zur Folge, dass alle ZNA , die dieser Zustandsnachricht entsprechen, auf *falsch* gesetzt werden. Dadurch kann ein zweiter Komponentenauftrag seine Aufgabe starten, wenn der erste Komponentenauftrag alle negierten Zustandsnachrichten abgeschickt hat und bei ihm trotzdem noch alle ZNA *wahr* sind (vgl. hierzu auch Abschnitt 5.9). Dieser Mechanismus erlaubt also das Parallelisieren von Aufgaben auf derselben oder unterschiedlichen Komponenten.

Abbildung 5.23 zeigt ein Beispiel eines Komponentenauftrages mit 4 ZNA . Alle 4 ZNA sind in diesem Beispiel erfüllt, so dass die entsprechende Aufgabe gestartet werden kann. Zuvor werden jedoch, abhängig vom Attribut R , die negierten Zustandsnachrichten (in diesem Fall $ZN1$ und $ZN4$) verschickt.

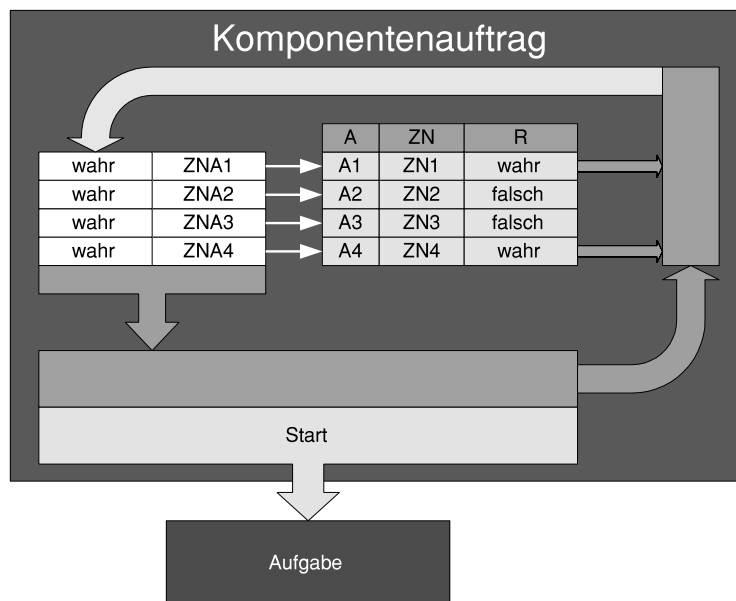


Abb. 5.23: Absenden der negierten Zustandsnachrichten aufgrund des Attributes R am Beispiel von 4 ZNA in einem Komponentenauftrag

Abbildung 5.24 zeigt denselben Komponentenauftrag wie Abbildung 5.23 nach dem Empfang der negierten Zustandsnachrichten $\neg ZN1$ und $\neg ZN4$. Das Beispiel zeigt, dass nach dem Start der Aufgabe die Zustandsnachrichtenanforderungen $ZNA1$ und $ZNA4$

nicht mehr erfüllt sind, *ZNA2* und *ZNA3* aber weiterhin erfüllt bleiben. Um die Aufgabe dieses Komponentenauftrags somit noch einmal zu starten, bedarf es lediglich der Zustandsnachrichten *ZN1* und *ZN4* (insofern keine weiteren negierten Zustandsnachrichten eintreffen).

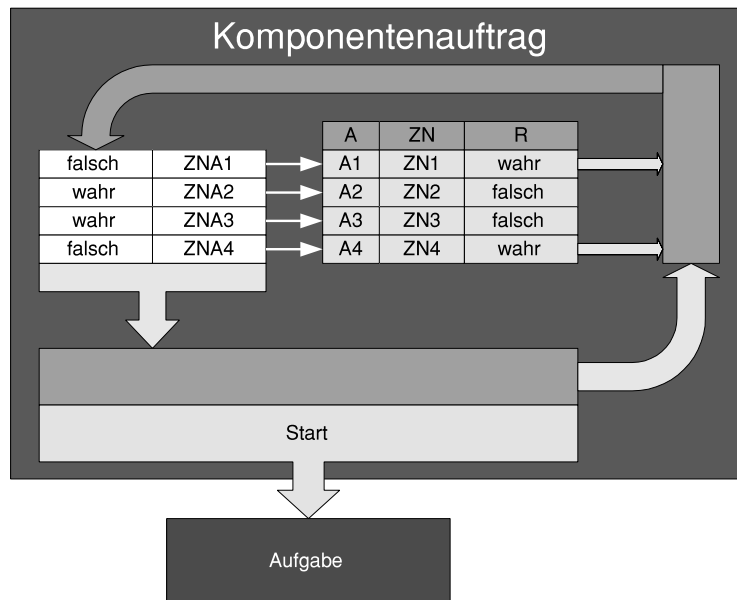


Abb. 5.24: Komponentenauftrag nach Empfang der negierten Zustandsnachrichten

5.8.5 Abbildung der Zustandsnachrichten auf die logischen Positionen eines Komponentenauftrags

Bei der Definition der Zustandsnachrichten wurde beschrieben, dass eine Zustandsnachricht, welche die Änderung des Zustandes der logischen Produktposition anzeigt, auch die logische Produktposition des Produktes selbst beinhaltet (vgl. Abschnitt 5.8.2). Die logischen Produktpositionen dieser Zustandsnachrichten können also dazu benutzt werden, die logischen Produktpositionen für die Eingangs- und Ausgangsprodukte zu konfigurieren (siehe Abb. 5.25). Es werden die Zustandsnachrichten, welche in den *ZNA* konfiguriert sind, dazu benutzt, alle Eingangsprodukte und das Ausgangsprodukt zu konfigurieren. Da es k Eingangsprodukte und ein Ausgangsprodukt bei einem Komponentenauftrag gibt, sind demnach mindestens $k+1$ *ZNA* in dem Komponentenauftrag zu konfigurieren.

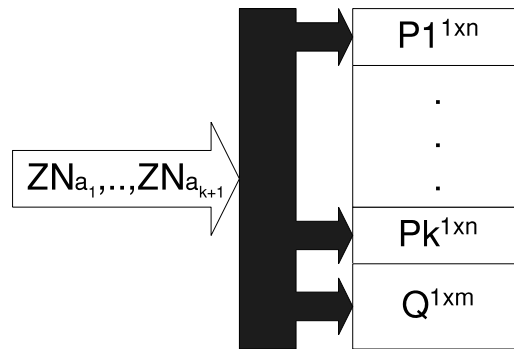


Abb. 5.25: Konfiguration der logischen Produktpositionen für Eingangs- und Ausgangsprodukte durch $k+1$ Zustandsnachrichten

5.8.6 Endausbau des logischen Repräsentanten

Aus Abschnitt 5.7 und den Abschnitten 5.8.1 bis 5.8.5 kann der endgültige Ausbau des logischen Repräsentanten abgeleitet werden. Im Folgenden werden die einzelnen Teile (oder Einheiten) beschrieben, aus denen sich der logische Repräsentant zusammensetzt (siehe Abb. 5.26).

- *Logische Komponente:* Sie ist die Abbildung der physikalischen Komponente und ist mit dieser über die standardisierte Schnittstelle für Komponenten verbunden.
- *Zustandsmaschinen:* Dies sind die Zustandsmaschinen für jede Aufgabe, die von der physikalischen Komponente ausgeführt werden kann. Jede Zustandsmaschine bildet den aktuellen Zustand der entsprechenden Aufgabe in der physikalischen Komponente ab.
- *Produktinformationen:* Dieser Teil beinhaltet alle Informationen bezüglich des Produktes auf einer der Produktpositionen. Angefangen bei dem logischen Produkt, über den Zustand der logischen Produktpositionen bis hin zu den Zuständen der logischen und damit auch der physikalischen Produkte.
- *ZN Generator:* Der Zustandsnachrichten-Generator ist für die Erzeugung und das Abschicken der Zustandsnachrichten des ganzen logischen Repräsentanten zuständig. Er bekommt deshalb von allen anderen Einheiten Informationen, um die nötigen Zustandsnachrichten zu erzeugen.

- *Komponentenauftrag Management*: Hier sind alle Komponentenaufträge des logischen Repräsentanten hinterlegt. Die gesamte Verwaltung der Aufträge geschieht hier. Alle einkommenden Zustandsnachrichten werden an die Komponentenaufträge verteilt. Ist ein Komponentenauftrag bereit, seine Aufgabe zu starten, wird dies von dieser Einheit übernommen. Ebenso wird das Abschicken der negierten Zustandsnachrichten von dieser Einheit übernommen. Die Komponentenaufträge mit ihren ZNA werden in dieser Einheit konfiguriert.
- *Zustandsnachrichten Bus*: Er wird zur Übermittlung der Zustandsnachrichten benutzt. Hierbei ist es gleichgültig, ob die Zustandsnachrichten zu einem anderen logischen Repräsentanten gelangen müssen oder ob sie nur innerhalb des Absendenden logischen Repräsentanten benötigt werden. Alle Zustandsnachrichten, auch die negierten, werden immer über diesen Bus versendet.

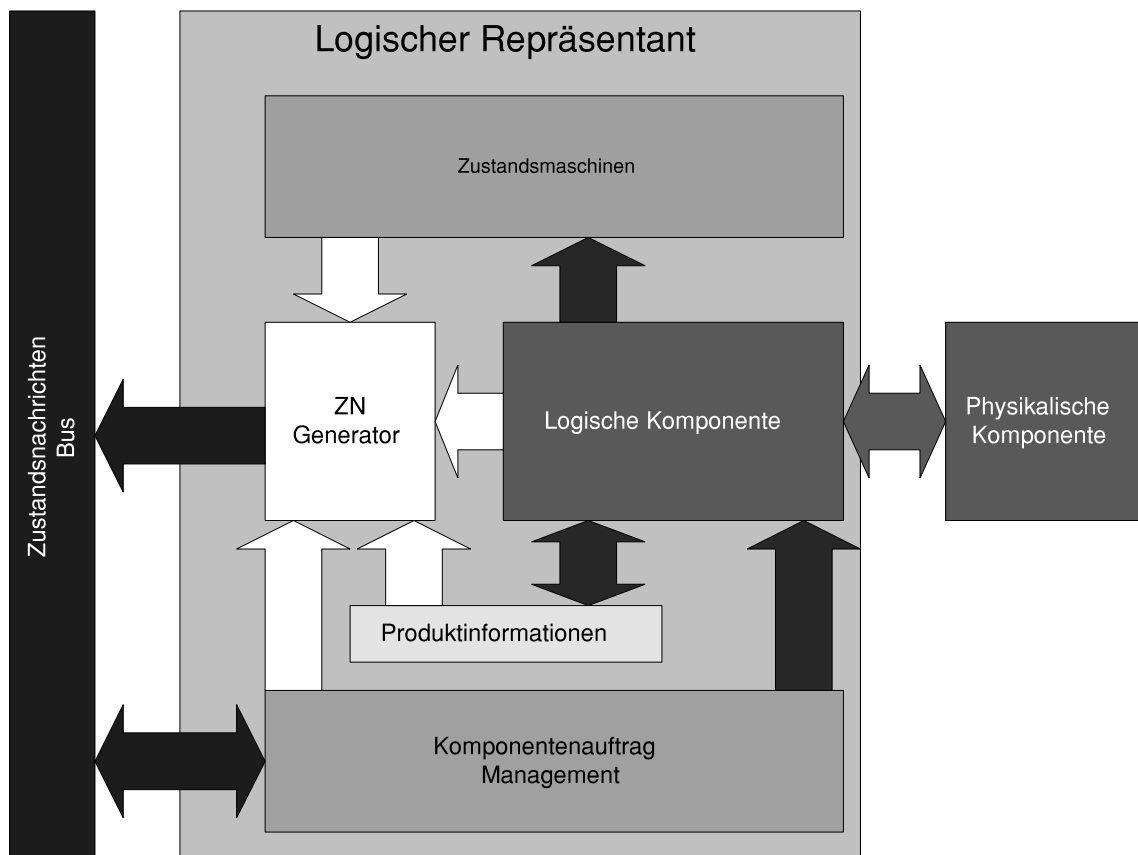


Abb. 5.26: Schematischer Endausbau des logischen Repräsentanten

5.8.7 Einschränkung der Zustandsnachrichten

Bisher wurden die Zustandsnachrichten allgemein eingeführt. Sie waren definiert durch eine Änderung eines Zustandes in einer Zustandsklasse. Um die Komplexität eines Systems mit endlich vielen Zustandsklassen und Zustandsnachrichten zu reduzieren, wird hier eine Einschränkung auf vier Zustandsnachrichten gemacht.

a.) Zustandsnachricht 1: *FertigZumBeladen*

Diese Zustandsnachricht wird erzeugt und gesendet, wenn sich der Zustand bei der Zustandsmaschine einer logischen Produktposition ändert, und zwar auf den Zustand *FertigZumBeladen* (siehe Abb. 5.12 und Abb. 5.13). Neben der logischen Produktposition selbst beinhaltet diese Zustandsnachricht auch noch die Produktinformation (das logische Produkt) und den Zustand des Produktes (gut, schlecht, wiederbearbeiten).

b.) Zustandsnachricht 2: *FertigZumEntladen*

Ähnlich wie die Zustandsnachricht 1 wird diese Zustandsnachricht erzeugt und gesendet bei einer Zustandsänderung der Zustandsmaschine einer logischen Produktposition. In diesem Fall handelt es sich um die Änderung in den Zustand *FertigZumEntladen*. Auch hier beinhaltet die Zustandsnachricht die logische Produktposition, die Produktinformation und den Zustand des Produktes.

c.) Zustandsnachricht 3: *KomponentenauftragBeendet*

Diese Zustandsnachricht hängt von einer Zustandsänderung in der Einheit Komponentenauftrag-Management und der Zustandsmaschine seiner entsprechenden Aufgabe ab. Sobald ein Komponentenauftrag beendet ist, das heißt, seine entsprechende Aufgabe auf der physikalischen Komponente beendet ist, wird diese Zustandsnachricht erzeugt und abgesendet. Diese Zustandsnachricht beinhaltet zusätzlich noch die eindeutige Identifizierung des Komponentenauftrags, über die Grenzen des logischen Repräsentanten hinweg.

d.) Zustandsnachricht 4: *Alarm*

Tritt ein Alarm in der physikalischen Komponente auf, so wird dieser über die logische Komponente dem logischen Repräsentanten bekannt gemacht. Der logische Repräsentant erzeugt daraus eine Zustandsnachricht, welche eine Alarmidentifizierung beinhaltet, und sendet diese ab.

5.8.8 Vergleich von Zustandsnachrichten

In den Komponentenauftrag eingehende Zustandsnachrichten, seien es reguläre oder negierte Zustandsnachrichten, müssen mit den Zustandsnachrichten der ZNA im Komponentenauftrag verglichen werden, um entsprechend die ZNA auf *wahr* oder *falsch* zu setzen. Dieser Vergleich muss für jede mögliche Zustandsnachricht definiert werden. Im Folgenden werden die Vergleiche der eingeschränkten vier Zustandsnachrichten beschrieben.

a.) Zustandsnachricht 1: *FertigZumBeladen*

Bei der Angabe dieser Zustandsnachricht in einer ZNA ist weder ein logisches Produkt noch eine logische Produktposition vorhanden, da es sich lediglich um eine Schablone (vgl. Definition von ZNA) handelt. Gegeben ist in dieser Schablone also nur, neben der Zustandsnachricht selbst, ein Produkttyp und der Zustand, der von dem eingehenden logischen Produkt erwartet wird (z. B. gut, schlecht, usw.). Beim Produkttyp handelt es sich wiederum nur um eine Schablone eines logischen Produktes, welches zum Vergleich mit dem logischen Produkt der einkommenden Zustandsnachricht benutzt wird. Somit ist eine einkommende Zustandsnachricht (ZN1) gleich der in der ZNA gegebenen Zustandsnachricht (ZN2), wenn die in Abbildung 5.27 aufgelisteten Werte gleich sind.

ZN1		ZN2
Nachricht (FertigZumBeladen)	=	Nachricht (FertigZumBeladen)
Absender	=	Absender (def. in ZNA)
logisches Produkt	=	Produkttyp
Zustand des Produktes	=	Vorgegebener Zustand

Abb. 5.27: Vergleichsmethode einer einkommenden Zustandsnachricht und einer gegebenen Zustandsnachricht einer ZNA am Beispiel der Nachricht *FertigZumBeladen*

b.) Zustandsnachricht 2: *FertigZumEntladen*

Der Vergleich der einkommenden Zustandsnachricht mit der Zustandsnachricht der entsprechenden ZNA funktioniert analog zu dem Vergleich der Zustandsnachricht *FertigZumBeladen*. Bei der Nachricht handelt es sich um

FertigZumEntladen.

c.) Zustandsnachricht 3: *KomponentenauftragBeendet*

Bei dieser Zustandsnachricht werden nur noch die Nachrichten verglichen und die Komponentenauftragsidentifizierung der einkommenden Zustandsnachricht mit der vorgegebenen Komponentenauftragsidentifizierung.

d.) Zustandsnachricht 4: *Alarm*

Hier wird ähnlich dem Vergleich bei Zustandsnachricht 3 verfahren. Allerdings wird hier, statt dem Vergleich der Komponentenauftragsidentifizierung, die Alarmidentifizierung verglichen.

5.9 Konfiguration des logischen Repräsentanten

Im Folgenden wird gezeigt, wie eine Konfiguration der Komponentenaufträge und ihrer ZNA in logischen Repräsentanten vor sich geht. Hierzu wird die erweiterte Produktstruktur, wie sie im Kapitel *Analyse des Montageablaufs und Konzeption einer Montagezelle* eingeführt wurde, noch einmal betrachtet. Abbildung 5.28 zeigt einen Ausschnitt aus dieser erweiterten Produktstruktur.

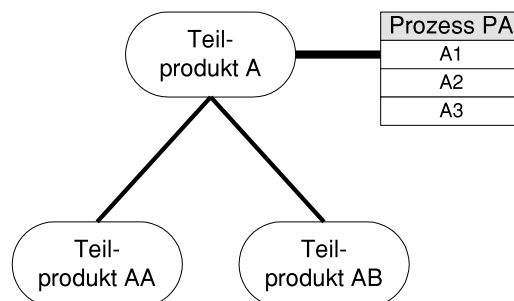


Abb. 5.28: Ausschnitt aus einer erweiterten Produktstruktur

Um Teilprodukt A aus den Teilprodukten AA und AB zu montieren, bedarf es des Prozesses PA, der aus den Subprozessen A1, A2 und A3 besteht. Ausgehend von diesen drei Subprozessen, werden im Folgenden zwei verschiedene Vorranggraphen konfiguriert. In beiden Beispielen werden die drei Subprozesse sowohl als eigenständige Komponenten als auch als Aufgaben dieser Komponenten betrachtet. Bei dem ersten Vorranggraphen wird von einer sequenziellen Ausführung der drei Subprozesse ausgegangen (siehe Abb. 5.29).

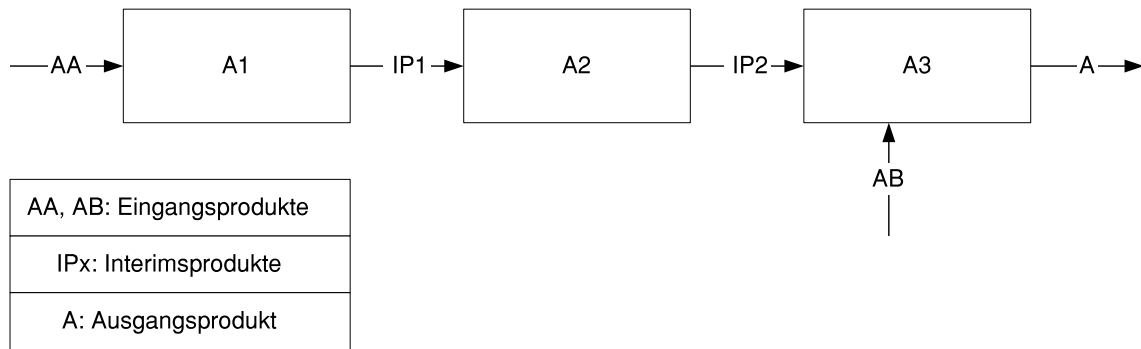


Abb. 5.29: Sequenzieller Vorranggraph der Subprozesse A1, A2 und A3

Für den zweiten Vorranggraphen wird angenommen, dass die beiden Subprozesse A1 und A2 parallel ausführbar sind und der Subprozess A3 nach Abschluss beider Subprozesse gestartet wird (siehe Abb. 5.30).

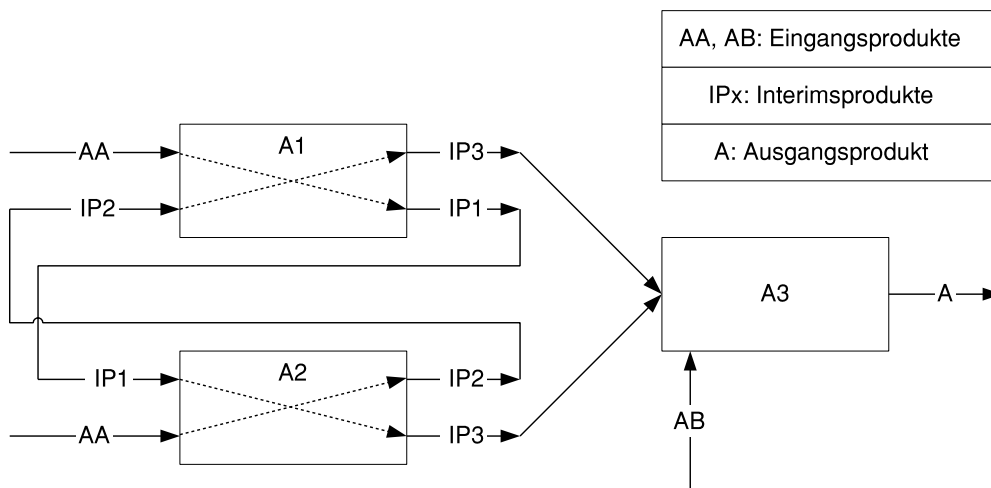


Abb. 5.30: Vorranggraph mit zwei parallelen Subprozessen A1, A2 und darauf folgendem Subprozess A3

Anhand dieser beiden Beispiele von Vorranggraphen wird die Konfiguration verdeutlicht. Um die beiden Vorranggraphen voneinander zu unterscheiden, wird im weiteren Verlauf der Vorranggraph aus Abbildung 5.29 als *Sequenzieller Vorranggraph* und jener aus Abbildung 5.30 als *Paralleler Vorranggraph* bezeichnet.

5.9.1 Konfiguration der Komponentenaufträge

Für beide Vorranggraphen werden zuerst einmal Komponentenaufträge definiert. Hierbei ist es unerheblich, ob alle Komponentenaufträge auf einem logischen Repräsentanten zum Einsatz kommen, oder ob es sich um unterschiedliche logische Repräsentanten handelt. Jedem Komponentenauftrag wird hierbei genau eine Aufgabe (Subprozess) zugeordnet und genau ein Ausgangsprodukt definiert. Wie jedoch schon

bei der Definition des Komponentenauftrags erklärt, darf es durchaus mehrere Komponentenaufträge geben, denen die gleiche Aufgabe zugeordnet ist.

a.) Sequenzieller Vorranggraph

Bei dem *Sequenziellen Vorranggraphen* werden die drei Komponentenaufträge *SK1*, *SK2* und *SK3* konfiguriert (SK = Serieller Komponentenauftrag). Abbildung 5.31 zeigt die genaue Konfiguration dieser drei Komponentenaufträge.

Komponentenauftrag	Aufgabe (Subprozess)	Ausgangsprodukt
SK1	A1	IP1
SK2	A2	IP2
SK3	A3	A

Abb. 5.31: Komponentenaufträge für Sequenziellen Vorranggraphen

b.) Paralleler Vorranggraph

Für den *Parallelen Vorranggraphen* sind sechs Komponentenaufträge *PK1*, *PK2*, *PK3*, *PK4*, *PK5* und *PK6* für die Konfiguration nötig (PK = Paralleler Komponentenauftrag). Abbildung 5.32 zeigt die genaue Konfiguration dieser fünf Komponentenaufträge.

Komponentenauftrag	Aufgabe (Subprozess)	Ausgangsprodukt
PK1	A1	IP1
PK2	A1	IP3
PK3	A2	IP2
PK4	A2	IP3
PK5	A3	A
PK6	A3	A

Abb. 5.32: Komponentenaufträge für Parallelen Vorranggraphen

5.9.2 Konfiguration der Zustandsnachrichtenanforderungen in den Komponentenaufträgen

Nachdem im letzten Abschnitt die einzelnen Komponentenaufträge konfiguriert wurden, muss nun für jeden dieser Komponentenaufträge die Liste der ZNA konfiguriert werden. In beiden Beispielen werden nur die Zustandsnachrichten *FertigZumBeladen* und *FertigZumEntladen* benutzt. Darüber hinaus wird davon ausgegangen, dass die Aufgabe (Subprozess) sich ihre Produkte immer beim

Vorgänger abholen kann, das heißt, dass keine Handhabungskomponente zwischen den Aufgaben fungiert.

a.) Sequenzieller Vorranggraph

Abbildung 5.33 zeigt die Konfiguration der ZNA für die Komponentenaufträge, um den Sequenziellen Vorranggraphen (siehe Abb. 5.29) zu realisieren. Hierbei wird davon ausgegangen, dass die Eingangsprodukte AA und AB von einer vierten Komponente XX geholt werden können. Auch die Aufgabe des ersten Komponentenauftrags SK1 holt sich ihr benötigtes Produkt AA somit von einem Vorgänger ab.

Auftrag	Absender	Nachricht	Produkttyp	Zustand	Attribut R
SK1	XX	FertigZumEntlad.	AA	gut	wahr
	A1	FertigZumBelad.	AA	gut	wahr
SK2	A1	FertigZumEntlad.	IP1	gut	wahr
	A2	FertigZumBelad.	IP1	gut	wahr
SK3	XX	FertigZumEntlad.	AB	gut	wahr
	A2	FertigZumEntlad.	IP2	gut	wahr
	A3	FertigZumBelad.	AB	gut	wahr
	A3	FertigZumBelad.	IP2	gut	wahr

Abb. 5.33: Konfiguration der ZNA für die Komponentenaufträge des Sequenziellen Vorranggraphen

b.) Paralleler Vorranggraph

Auch bei der Konfiguration der ZNA für die Komponentenaufträge des Parallelen Vorranggraphen wird von einer vierten Komponente XX ausgegangen (siehe Abb. 5.34). In diesem Beispiel ist gut zu sehen, dass Komponentenaufträge, die sich sowohl in ihrer Aufgabe als auch in ihrem Ausgangsprodukt gleichen, sich nur durch ihre ZNA unterscheiden können.

Auftrag	Absender	Nachricht	Produkttyp	Zustand	Attribut R
PK1	XX	FertigZumEntlad.	AA	gut	wahr
	A1	FertigZumBelad.	AA	gut	wahr
PK2	A2	FertigZumEntlad.	IP2	gut	wahr
	A1	FertigZumBelad.	IP2	gut	wahr
PK3	XX	FertigZumEntlad.	AA	gut	wahr
	A2	FertigZumBelad.	AA	gut	wahr
PK4	A1	FertigZumEntlad.	IP1	gut	wahr
	A2	FertigZumBelad.	IP1	gut	wahr
PK5	XX	FertigZumEntlad.	AB	gut	wahr
	A1	FertigZumEntlad.	IP3	gut	wahr
	A3	FertigZumBelad.	AB	gut	wahr
	A3	FertigZumBelad.	IP3	gut	wahr
PK6	XX	FertigZumEntlad.	AB	gut	wahr
	A2	FertigZumEntlad.	IP3	gut	wahr
	A3	FertigZumBelad.	AB	gut	wahr
	A3	FertigZumBelad.	IP3	gut	wahr

Abb. 5.34: Konfiguration der ZNA für die Komponentenaufträge des Parallelen Vorranggraphen

Um die Übersichtlichkeit sowohl der Konfiguration des *Sequenziellen Vorranggraphen* (siehe Abb. 5.33) als auch der des *Parallelen Vorranggraphen* (siehe Abb. 5.34) zu erhalten, wurde darauf verzichtet, die Eingangs- und Ausgangsparameter für die Aufgaben der konfigurierten Komponentenaufträge explizit anzugeben.

Für jeden Komponentenauftrag (*SK1...SK3, PK1...PK6*) kann eine ZNA definiert werden (zwei ZNA für die Aufträge, welche die Komponente *A3* betreffen), um dessen entsprechende Zustandsnachricht als Eingangsparameter für die Aufgabe des jeweiligen Komponentenauftrags zu benutzen (vgl. Abschnitt 5.8.5).

Ein Ausgangsparameter kann nicht definiert werden, da bei keinem Komponentenauftrag eine ZNA konfiguriert ist, welche das Ausgangsprodukt betrifft. Dies bedeutet, dass die Komponenten das Ausgangsprodukt auf der logischen Produktposition des Eingangsproduktes belassen.

5.10 Zusammenarbeit der logischen Repräsentanten

Im letzten Abschnitt wurde festgestellt, dass eine neue Aufgabe eines Komponentenauftrags erst nach der Beendigung einer vorhergehenden Aufgabe gestartet werden darf. Um dennoch Parallelität zu erreichen, wurden die negierten Zustandsnachrichten

eingeführt, welche es erlauben, eine neue Aufgabe schon nach dem Verteilen der negierten Zustandsnachrichten zu starten. Trotz dieser Einführung der negierten Zustandsnachrichten, dürfen nicht gleichzeitig zwei Aufgaben gestartet werden.

Da die Aufgaben in unterschiedlichen Komponenten sein können, und somit auch die Komponentenaufträge in unterschiedlichen logischen Repräsentanten, wird ein Mechanismus benötigt, der es immer nur einem Komponentenauftrag erlaubt, seine Aufgabe zu starten. Erst wenn dieser Komponentenauftrag alle negierten Zustandsnachrichten verschickt hat (und sie somit auch bei allen angekommen sind; siehe standardisierte Schnittstelle für Komponenten), dürfen andere Komponentenaufträge (insofern alle ZNA erfüllt sind) ihre Aufgabe starten. Dieser Mechanismus muss auch in der Lage sein, einem Komponentenauftrag den Vorrang zu geben, falls einmal zwei Komponentenaufträge zur selben Zeit alle ihre ZNA erfüllt haben.

5.10.1 Einführung von Agenten

Aus softwaretechnischer Sicht gibt es mehrere Verfahren, mit denen sich ein solcher Mechanismus realisieren lässt. Viele dieser Ansätze sind jedoch von zentralistischer Natur. Eine zentrale Instanz übernimmt hierbei die Entscheidung, ob oder welche Aufgabe gestartet werden darf. Diese zentrale Instanz weiß also darüber Bescheid, ob im System gerade eine Aufgabe gestartet wird, aber noch nicht alle negierten Zustandsnachrichten von ihr abgesendet sind, und somit der Start einer anderen Aufgabe untersagt werden muss. Zusätzlich muss sie auch noch darüber entscheiden, welche von zwei Aufgaben zuerst gestartet wird, falls zwei Aufträge zur selben Zeit alle ZNA erfüllt haben.

Eine andere Möglichkeit ist der dezentrale Ansatz. Hierbei haben sich in den letzten Jahren vor allem Agenten [Ritter01] als sehr brauchbar herausgestellt.

Da die logischen Repräsentanten eigenständige abgeschlossene Einheiten sind, bietet sich dieser Ansatz hier an. Jeder logische Repräsentant kann als eigenständiger technischer Agent angesehen werden. Somit bildet das gesamte Leitsystem mit seinen logischen Repräsentanten ein technisches Multi-Agenten-System [Lüth98].

Durch den Einsatz von Agenten findet noch keine Festlegung auf die Systemarchitektur statt (vgl. Abschnitt 4.1). Es muss nicht zwangsläufig eine vollständig verteilte Systemarchitektur zum Einsatz kommen. Auch der Einsatz einer hierarchischen Systemarchitektur ist durch die Benutzung von Softwareagenten auf dem Leitsystem möglich.

Der technische Agent wird in einen Agentenkopf und einen Agentenkörper unterteilt. Der logische Repräsentant kann hierbei als Agentenkörper angesehen werden. Dieser hat eine Verbindung zu seinem Agentenkopf. Sind in dem logischen Repräsentanten bei einem Komponentenauftrag alle ZNA erfüllt, und somit die entsprechende Aufgabe zur Ausführung bereit, so wendet sich der Agentenkörper, genauer gesagt das Komponentenauftrag-Management des logischen Repräsentanten, an seinen Agentenkopf. Dieser verhandelt mit den anderen Agentenköpfen über die Freigabe des Komponentenauftrags.

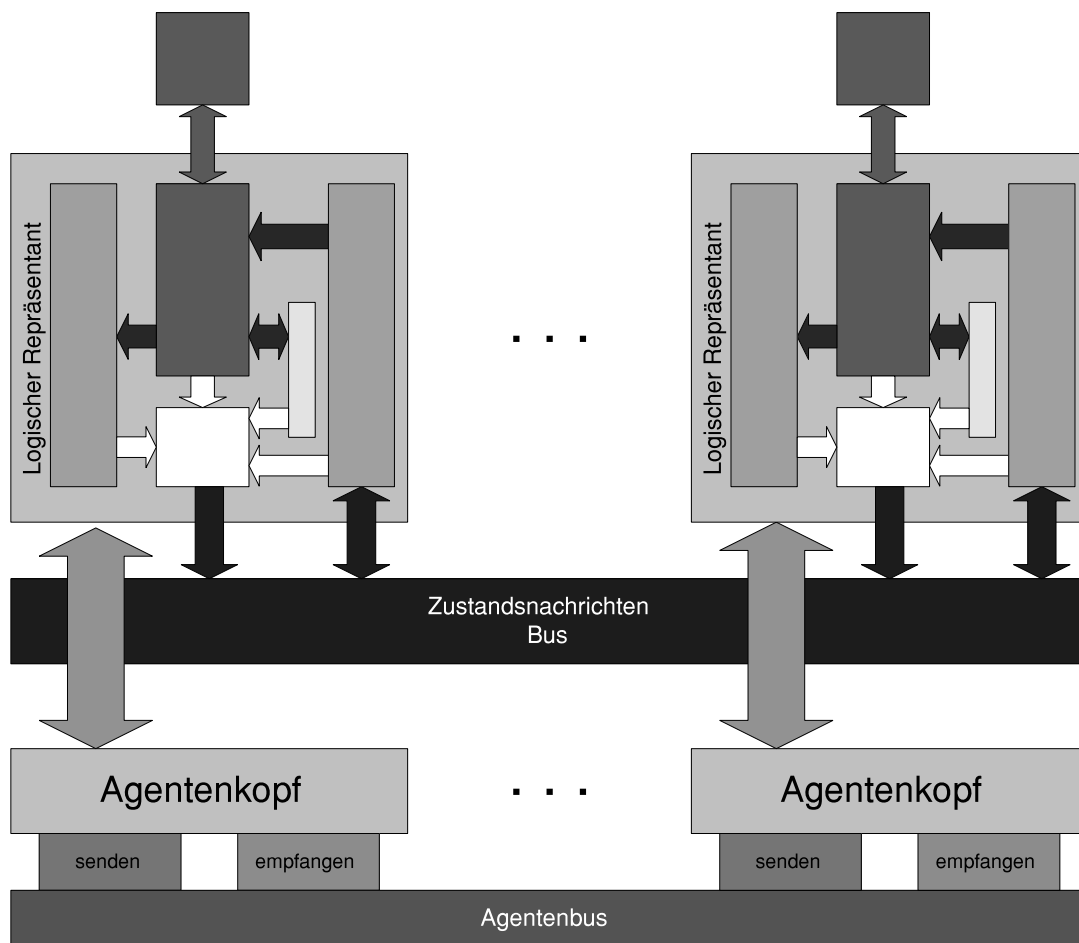


Abb. 5.35: Zusammenarbeit der logischen Repräsentanten als Multi-Agenten-System

Um zwischen den Agentenköpfen Verhandlungen, das heißt eine Kommunikation, zu erreichen, werden alle Agentenköpfe über den Agentenbus verbunden. Die logischen Repräsentanten, und damit die Agentenkörper, bleiben aber weiterhin über den Zustandsnachrichten-Bus direkt miteinander verbunden (siehe Abb. 5.35).

5.10.2 Funktionsweise des Multi-Agenten-Systems

Anhand eines Beispiels mit zwei Komponentenaufträgen, bei denen alle ZNA erfüllt sind, wird die Zusammenarbeit der logischen Repräsentanten, das heißt die Funktionsweise des Multi-Agenten-Systems, erklärt. Beide logische Repräsentanten wollen somit zur selben Zeit eine Aufgabe starten. Das Beispiel zeigt die Vorgehensweise der Agentenköpfe, um das gleichzeitige Starten, welches zu erheblichen Schäden an den physikalischen Komponenten führen kann, zu verhindern.

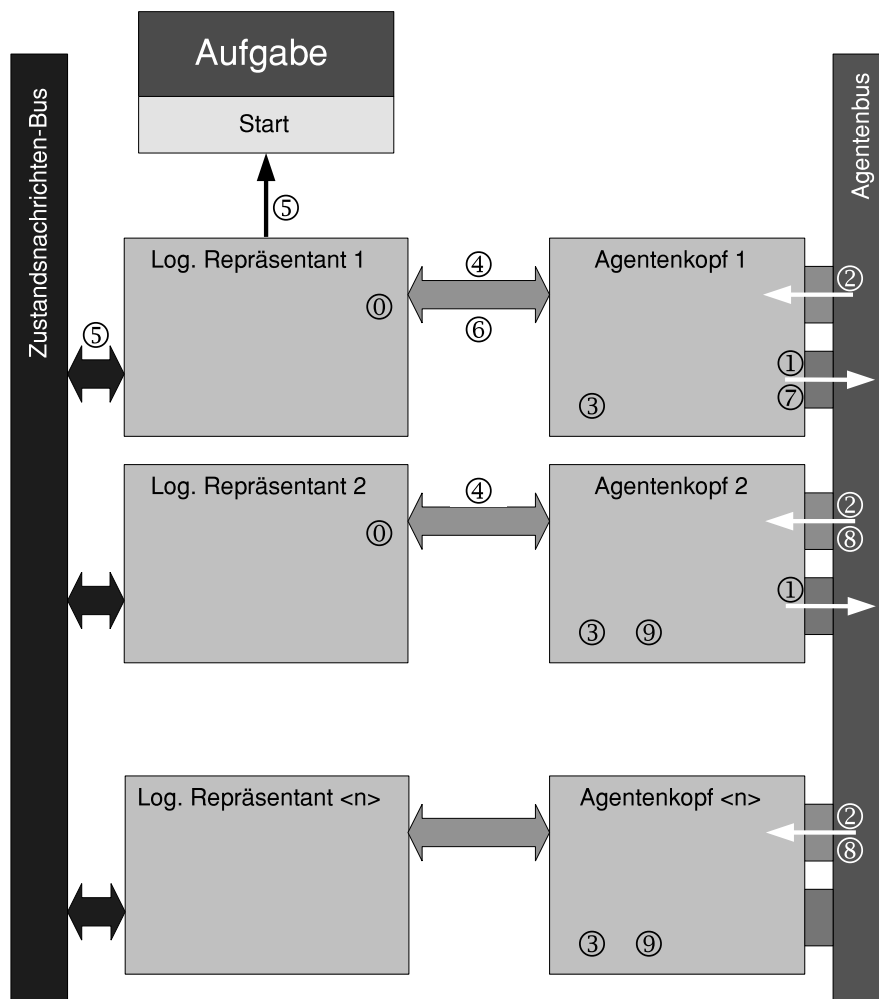


Abb. 5.36: Funktionsweise des Multi-Agenten-Systems an einem Beispiel mit zwei gleichzeitig erfüllten Komponentenaufträgen

Abbildung 5.36 zeigt dieses Beispiel. Der Ablauf der einzelnen Schritte in diesem Beispiel ist wie folgt:

- ① Je ein Komponentenauftrag im logischen Repräsentanten 1 und ein Komponentenauftrag im logischen Repräsentanten 2 sind erfüllt (alle ZNA sind erfüllt). Beide teilen dies ihrem Agentenkopf mit.
- ① Agentenköpfe von den logischen Repräsentanten 1 und logischen Repräsentanten 2 senden eine Nachricht an alle Agentenköpfe, dass ein Komponentenauftrag erfüllt ist und somit gestartet werden möchte.
- ② Agentenkopf empfängt Nachrichten von anderen Agentenköpfen über den Ausführungswunsch von deren Komponentenaufträgen.
- ③ Ist die Priorität des eigenen Komponentenauftrags höher als die eines anderen Komponentenauftrags, wird die Ausführung im Agentenkopf genehmigt (Agentenkopf 1). Ist die Priorität niedriger, wird eine Ausführungssperre gesetzt (Agentenkopf 2). Ist kein Komponentenauftrag des eigenen logischen Repräsentanten erfüllt, wird auch die Ausführungssperre gesetzt.
- ④ Benachrichtigung des zugehörigen logischen Repräsentanten über Ausführungsgenehmigung (logischer Repräsentant 1) oder über Ausführungssperre (logischer Repräsentant 2).
- ⑤ Absenden der negierten Zustandsnachrichten über den Zustandsnachrichten-Bus und Starten der entsprechenden Aufgabe des Komponentenauftrags.
- ⑥ Nachricht an Agentenkopf, dass alle negierten Zustandsnachrichten gesendet (und damit auch von allen anderen empfangen) wurden und dass die Aufgabe gestartet wurde.
- ⑦ Nachricht an alle anderen Agentenköpfe absenden, dass alle negierten Zustandsnachrichten gesendet wurden und somit die Ausführungssperre zurückgesetzt werden kann.
- ⑧ Empfang der Nachricht, dass die Ausführungssperre zurückgesetzt werden kann.
- ⑨ Rücksetzen der Ausführungssperre.

Liegt nach dem Rücksetzen der Ausführungssperre bei einem logischen Repräsentanten ein erfüllter Komponentenauftrag vor, so kann er dies sofort seinem entsprechenden Agentenkopf mitteilen und obiges Szenario beginnt für diesen Agentenkopf von vorne.

5.11 Konfiguration von Gruppen

Mit Hilfe der vier eingeschränkten Zustandsnachrichten ist es möglich, Sequenzen von Aufgaben zu konfigurieren. Das Multi-Agenten-System lässt in der bisherigen Form zu, dass die Aufgaben in ihrer konfigurierten Form sequenziell abgearbeitet werden, aber es ist nicht garantiert, dass eine andere Aufgabe zwischen zwei Aufgaben der Sequenz abgearbeitet wird. Es gibt jedoch Sequenzen von Aufgaben, die dies zwingend erfordern.

5.11.1 Nicht unterbrechbare Sequenzen von Aufgaben

In Abbildung 5.37 wird gezeigt, dass es Sequenzen von Aufgaben gibt, welche nicht durch andere Aufgaben unterbrochen werden dürfen, da dies sonst zu Schäden an den Komponenten oder an den Produkten führen kann.

Bei der Komponente 1 handelt es sich um eine Handhabungskomponente mit einem Greifer. Sie kann ein Produkt zu einer anderen Komponente bringen, ohne den Greifer zu öffnen, das heißt, ohne das Produkt los zulassen (*BringeProdukt*). Komponente 1 kann darüber hinaus den Greifer öffnen und sich von der aktuellen Position entfernen (*ÖffneGreifer_Entfernen*). Bei Komponente 2 handelt es sich um eine Montagekomponente, welche eine Position hat ein Produkt aufzunehmen, es dort zu bearbeiten und es dann wieder abzugeben. Die Besonderheit an Komponente 2 liegt in der Aufnahme des Produktes. Wenn das Produkt gebracht wird, muss es auf der Eingangsposition zuerst fixiert werden, bevor die Handhabungskomponente (Komponente 1) den Greifer öffnen und sich entfernen kann. Komponente 2 kann also das Produkt auf der Eingangsposition fixieren (*ProduktFixieren*) und das Eingangsprodukt prozessieren (*ProduktProzessieren*).

Komponente	Komponentenauftrag	Aufgabe
1	K1	BringeProdukt
2	K2	ProduktFixieren
1	K3	ÖffneGreifer_Entfernen
2	K4	ProduktProzessieren

Abb. 5.37: Beispiel einer nicht unterbrechbaren Sequenz von Aufgaben in der Reihenfolge ihrer Abarbeitung

Die Komponentenaufträge *K1*, *K2*, *K3* dürfen bei dieser Sequenz nicht unterbrochen werden. Die Aufgabe *BringeProdukt* verändert nicht den Zustand der Komponente 1. Sie hat sich zwar zu einer anderen Position bewegt, hat aber immer noch ein Produkt in ihrem Greifer. Würde es jetzt noch einen Komponentenauftrag *K5* in der Komponente 1 geben, welcher auch die Aufgabe *BringeProdukt* hat, allerdings nicht zu Komponente 2, sondern zu einer anderen Komponente, so würde dies, wenn er zwischen *K2* und *K3* zur Ausführung kommt, zu erheblichen Schäden an den Komponenten und am Produkt führen.

5.11.2 Einführung von Gruppen-Identifizierung

Um nicht unterbrechbare Sequenzen von Aufgaben zu erhalten, wird eine Gruppen-Identifizierung eingeführt. Die Gruppen-Identifizierung erlaubt es, Komponentenaufträge in Gruppen über die Grenze des logischen Repräsentanten hinaus zusammenzufassen, das heißt ihnen dieselbe Gruppen-Identifizierung zu geben. Eine Gruppen-Identifizierung muss einem Komponentenauftrag jedoch nicht zwingend gegeben werden. Soll er nicht mit anderen Aufträgen in einer Gruppe zusammengefasst sein, so wird ihm keine Gruppen-Identifizierung gegeben.

Sind mehrere Komponentenaufträge in einer Gruppe zusammengefasst, so müssen die Agentenköpfe darauf achten, dass jeder Komponentenauftrag, der dieser Gruppe angehört, mindestens einmal zur Ausführung kommt. Sobald also ein Komponentenauftrag, der einer Gruppe *G* angehört, zur Ausführung kommt, sind erst einmal alle Komponentenaufträge, die nicht dieser Gruppe *G* angehören, für die Ausführung gesperrt. Nur Aufträge, die auch Mitglied der Gruppe *G* sind, können somit ausgeführt werden. Wurde jeder Komponentenauftrag, welcher Mitglied der Gruppe *G* ist mindestens einmal ausgeführt, wird die Sperre aller Komponentenaufträge wieder aufgehoben.

In dem obigen Beispiel würde dies bedeuten, dass die Komponentenaufträge *K1*, *K2* und *K3* dieselbe Gruppen-Identifizierung bekommen. Damit sind sie Mitglieder derselben Gruppe und können somit nicht durch eine andere Aufgabe getrennt werden. Erst wenn alle drei Aufträge ausgeführt wurden, darf wieder ein anderer Komponentenauftrag zur Ausführung kommen. Komponentenauftrag *K4* braucht nicht in diese Gruppe aufgenommen zu werden. Da das Produkt auf der Komponente 2 platziert und fixiert ist, kann kein anderes Produkt mehr auf diese logische Produktposition gebracht werden, da die Komponente 2 keine Nachricht *FertigZumBeladen* für diese Position absenden wird. Die Nachricht *FertigZumEntladen* wird die Komponente 2 auch erst dann absenden, wenn sie den Komponentenauftrag *K4* abgeschlossen hat. Der Komponentenauftrag *K4* kann somit irgendwann zur Ausführung kommen und muss nicht zwingend, ohne dass eine andere Aufgabe zwischenzeitlich gestartet wurde, untrennbar mit den Komponentenaufträgen *K1*, *K2* und *K3* verbunden sein.

Das Zusammenschließen von mehreren Komponentenaufträgen in einer Gruppe besagt nicht, dass jeder Komponentenauftrag nur einmal zur Ausführung kommt. Es kann durchaus sein, dass ein Komponentenauftrag mehrere Male zur Ausführung kommt. Es wird dadurch nur garantiert, dass ein Komponentenauftrag der nicht Mitglied derselben Gruppe ist, auch nicht zur Ausführung kommt. Mit Hilfe der Zustandsnachricht *KomponentenauftragBeendet* lassen sich somit leicht nicht unterbrechbare Sequenzen von Komponentenaufträgen konfigurieren. Es erlaubt jedoch auch situationsabhängige Aufgabenfolgen (durch die Benutzung der Zustandsnachricht *FertigZumBeladen* und *FertigZumEntladen*), welche durch bestimmte Ereignisse (z.B. durch das Auftreten einer Zustandsnachricht *Alarm*) angestoßen werden.

5.12 Zusammenfassung

Es wurde hier ein konfigurierbares Leitsystem vorgestellt, welches die Anforderungen aus Abschnitt 3.6 erfüllen. Durch die Einführung von logischen Komponenten, welche den physikalischen Komponenten entsprechen, ist die Abbildung der Montagezelle auf der Steuerungsseite leicht nachvollziehbar. Da man für jede dieser Komponenten ihre Aufträge konfiguriert, ergibt sich für die gesamte Montagezelle ein Mechanismus, den man mit dem entliehenen Begriff *Plug & Play* bezeichnen könnte. Wird eine schon konfigurierte Komponente in das Leitsystem der Montagezelle eingebunden, so kann sie automatisch an der gestellten Montageaufgabe teilnehmen. Durch das Hinzufügen einer Komponente desselben Typs lässt sich so der Durchsatz auf einfache Weise erhöhen. Auch die Bedingungen für die Ausführung eines beliebigen Auftrages einer

Komponente lassen sich durch die Einführung von Zustandsnachrichten, die aus Änderungen in den Zustandsklassen resultieren, leicht konfigurieren. Auch die lückenlose Produktverfolgung, durch den konsequenten Einsatz der logischen Positionen und damit verbunden des logischen Produktes, ist in diesem konfigurierbaren Leitsystem für modulare Montagezellen vorhanden.

Kapitel 6

Realisierung des konfigurierbaren Leitsystems

Basierend auf dem Konzept einer flexiblen Montageanlage wurde der Prototyp „Manufacturing of the Future (MoF)“ realisiert [IBM03d]. In dieser prototypischen Anlage wurden eine Verbindungseinheit sowie verschiedene Montagezellen umgesetzt.

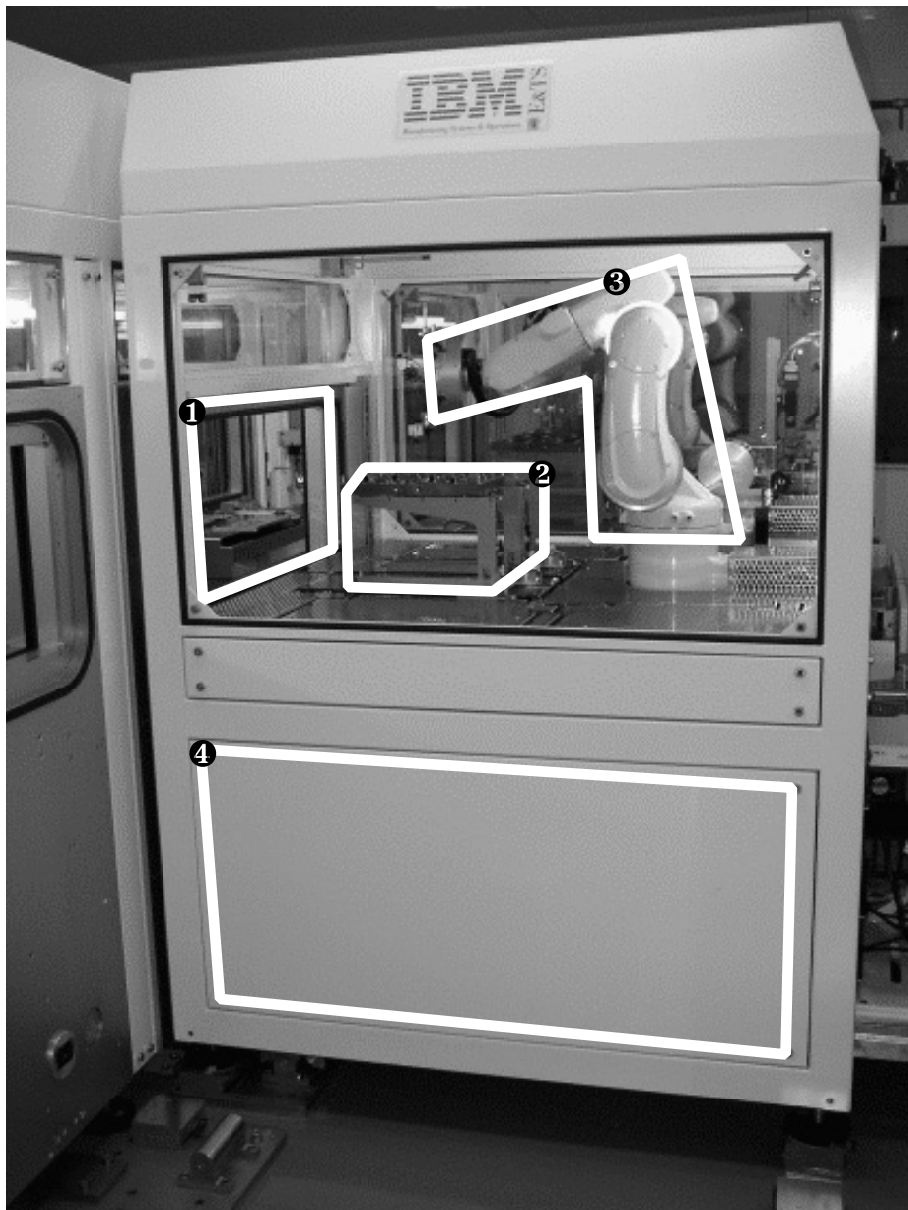
6.1 Aufbau des Prototyps

6.1.1 Montagezelle

Die Montagezelle wurde in einer Basisfassung realisiert. Diese bestand aus einem normierten Gestell, in welches zwischen vier und sechs Be-/Entladekomponenten und ein 6-Achs Roboter (die Handhabungskomponente) eingesetzt wurden (siehe Abb. 6.1). Die Positionen der Be-/Entladekomponenten waren hierbei fest vorgegeben, während die Position der Handhabungskomponente je nach Aufgabe, die von der Montagezelle zu erfüllen war, variierte. Neben diesen beiden Typen von Komponenten sind in unterschiedlichen Montagezellen noch unterschiedlichen Montagekomponenten hinzugekommen (siehe Abb. 6.2).

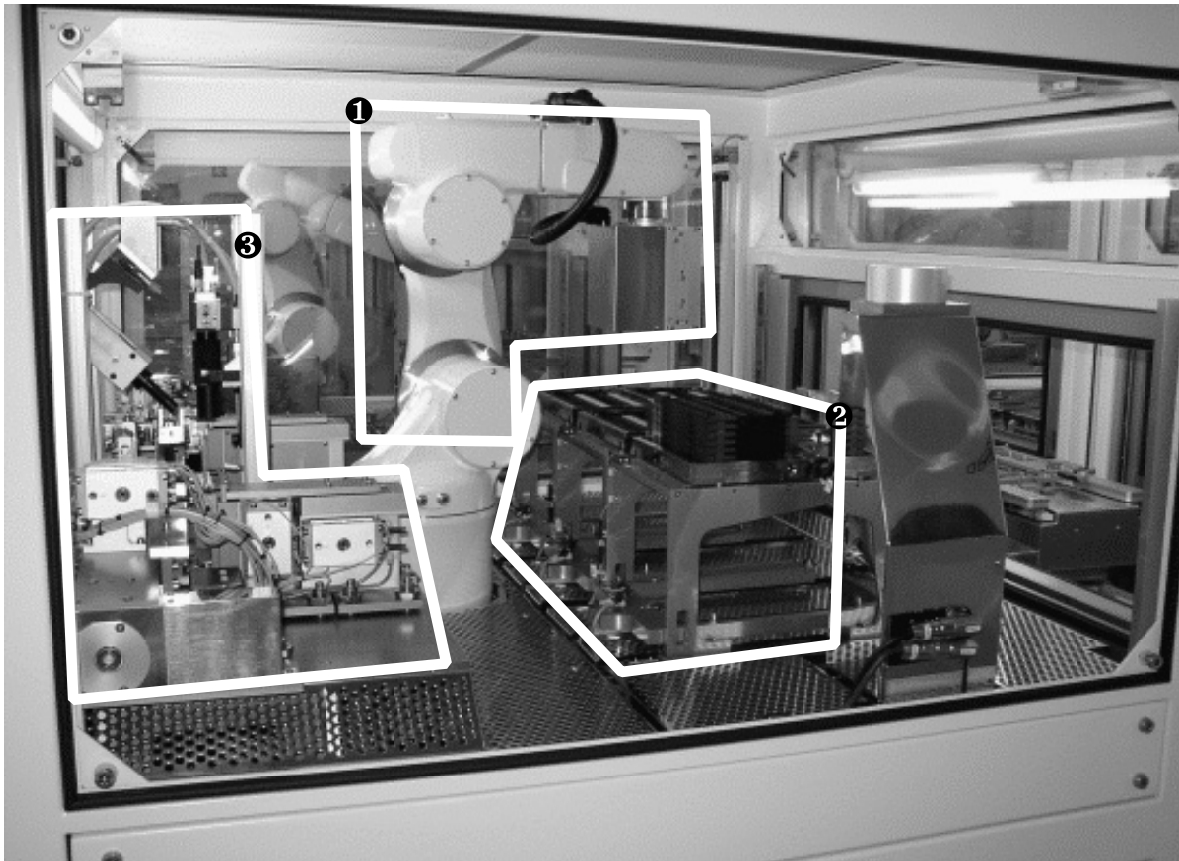
Die Steuerungen für die komplette Montagezelle, das heißt für alle darin befindlichen Komponenten, befinden sich unterhalb der eingebauten Komponenten. Die Montagezelle besitzt keinen Bildschirm. Da sie autark, das heißt völlig ohne den Eingriff eines Menschen, produzieren soll, ist ein Bildschirm nicht notwendig. Ein Bildschirm ist nur dann notwendig, wenn das Leitsystem konfiguriert werden muss. Eine Konfiguration ist nötig, wenn die Komponenten in der Montage sich geändert haben oder wenn ein neues Produkt in einer bestehenden Montagezelle zu montieren ist. Hierzu gibt es die Möglichkeit direkt an der Montagezelle einen Bildschirm sowie Tastatur und Maus anzuschließen. Wenn die Montagezelle an eine Montageanlage

angeschlossen ist, kann die Konfiguration des Leitsystems der Montagezelle über Netzwerk stattfinden.



- | | |
|-------------------------|-------------------------|
| ❶ Schleuse | ❸ Handhabungskomponente |
| ❷ Be-/Entladekomponente | ❹ Steuerungen |

Abb. 6.1: Außenansicht einer Montagezelle (© IBM Corporation)



- ❶ Handhabungskomponente
- ❷ Be-/Entladekomponenten
- ❸ Prozesskomponente

Abb. 6.2: Detailansicht einer Montagezelle mit Handhabungskomponente, Be-/Entladekomponente und Montagekomponente (© IBM Corporation)

Die Detailansicht in Abbildung 6.3 zeigt, wie der Greifer der Handhabungskomponente ein Produkt aus dem Werkstückträger auf einer Be-/Entladekomponente entnimmt. Bei dem Werkstückträger im Vordergrund sind die Einlagen gut zu erkennen. Der Werkstückträger im Hintergrund beinhaltet andere Produkte als derjenige im Vordergrund. Bei dem Greifer handelt es sich um einen Mehrfachgreifer, welcher in der Lage ist, verschiedene Produkte gleichzeitig zu greifen. Auf der rechten Seite des Bildes ist die geöffnete Schleuse zur Montageanlage zu sehen.

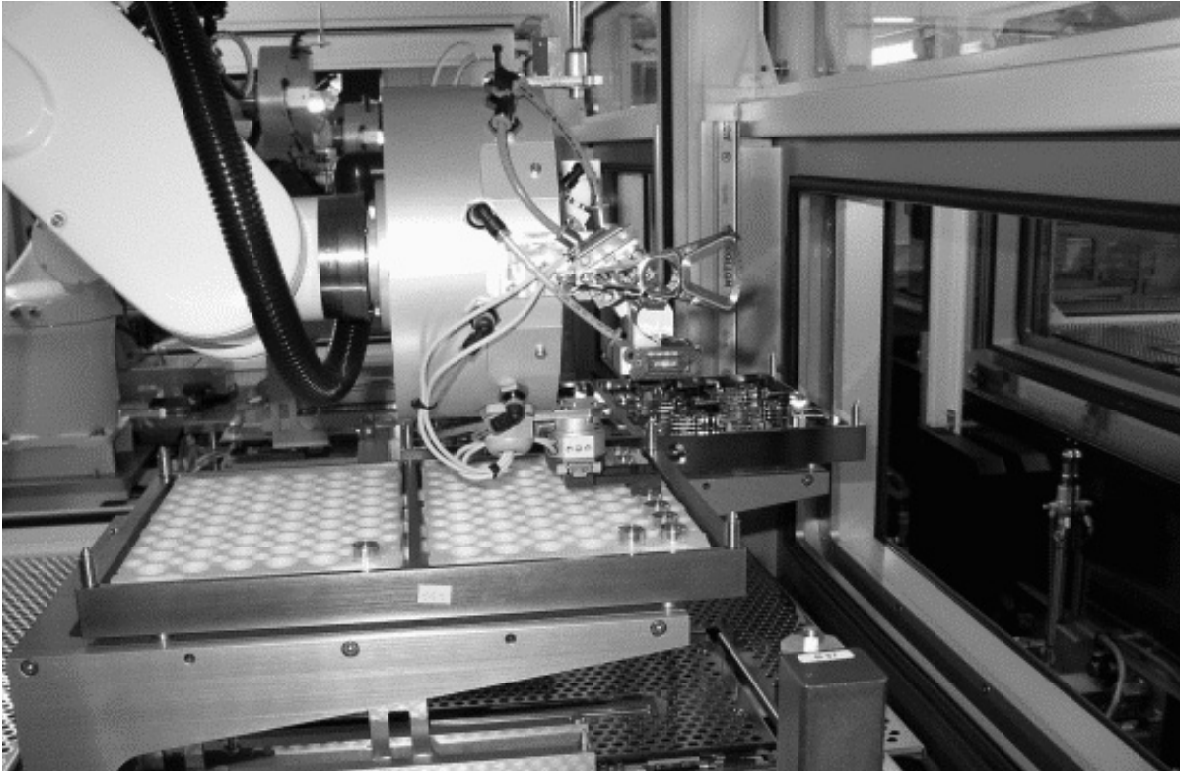


Abb. 6.3: Detailansicht einer Montagezelle mit dem Greifer der Handhabungskomponente, Be-/Entladekomponente und Werkstückträger (© IBM Corporation)

6.1.2 Montageanlage

Eine komplette Montageanlage wird aus mindestens einer Verbindungseinheit und den an sie angeschlossenen Montagezellen und Magazinen gebildet. Darüber hinaus besitzt die Montageanlage einen Leitstand mit Bildschirm, Tastatur und Maus (siehe Abb. 6.4). Dieser Leitstand dient zur Auswahl des Auftrages, welcher zur Montage eines Produktes auf dieser Montageanlage ausgeführt werden soll. Zusätzlich dient der Leitstand zur Überwachung der gesamten Montageanlage, das heißt der Anzeige von aktuell auf der Montageanlage laufenden Aktivitäten.



Abb. 6.4: *Komplette Montageanlage mit drei Montagezellen und Leitstand der Montageanlage (© IBM Corporation)*

6.2 Konfigurierbares Leitsystem

Die Realisierung des konfigurierbaren Leitsystems fand für eine Montagezelle, wie sie in Abschnitt 6.1.1 gezeigt ist, statt. Die Benutzerschnittstelle des Leitsystems wurde in englischer Sprache realisiert. Abbildung 6.5 zeigt ein solches Leitsystem, welches *Module Controller* genannt wird. Der abgebildete *Module Controller (MC)* ist das Leitsystem für eine Montagezelle mit dem Namen *Gramload*. Diese Montagezelle besteht aus drei Komponenten. Die drei Komponenten finden sich im *Module Controller* wieder. Es handelt sich um die Komponenten *Robot* (zentrale Handhabungskomponente), *IO* (Be-/Entladekomponente) und *Gramload* (eine Montagekomponente). Der *Module Controller* zeigt ständig den Zustand der einzelnen Komponenten an. Darüber hinaus besteht die Möglichkeit, über ein Menü an Funktionalitäten (Eigenschaften, RezeptRunterladen, RezeptHochladen, usw.) der einzelnen Komponenten zu gelangen. Da eine Montagezelle nicht über einen Bildschirm verfügt, ist die Benutzerschnittstelle üblicherweise nur bei der Konfiguration oder im Probetrieb der Montagezelle sichtbar.

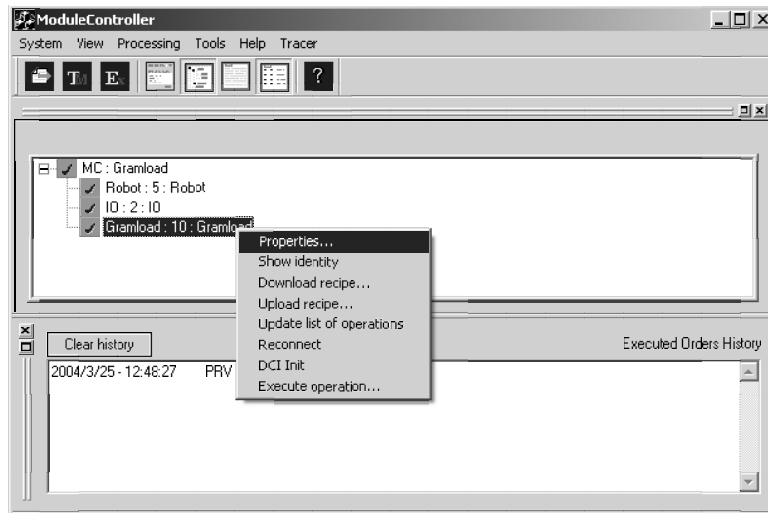


Abb. 6.5: Leitsystem einer Montagezelle (© IBM Corporation)

6.2.1 Auswahl des Produktes

Bevor das Leitsystem konfiguriert wird, muss das Produkt ausgewählt werden, für welches es konfiguriert werden soll. Hierzu existiert auf der linken Seite eine Liste aller möglichen Produkte (siehe Abb. 6.6). Wählt man ein Produkt aus, welches konfiguriert werden soll, so ist auf der rechten Seite immer die aktuellste Version aller Varianten dieses Produktes zu sehen (vgl. Abschnitt 3.4). Die Auswahl einer dieser Varianten führt zur Konfiguration dieser speziellen Variante des entsprechenden Produktes.

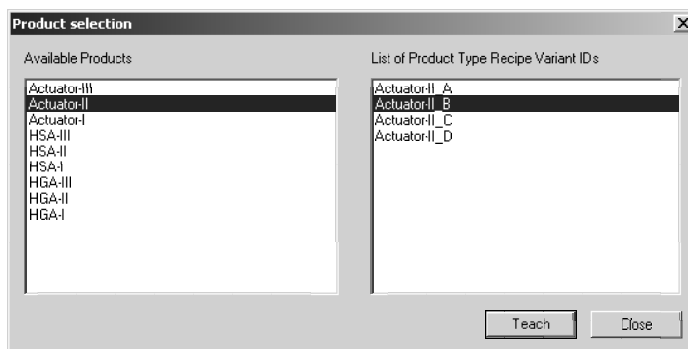


Abb. 6.6: Auswahl der zu konfigurierenden Variante eines Produktes (© IBM Corporation)

6.2.2 Konfiguration der Komponentenaufträge

Um den *Module Controller* zu konfigurieren, gibt es eine Benutzerschnittstelle, welche *Teach* genannt wird (siehe Abb. 6.7). Für jeden Prozessschritt aus der erweiterten Produktstruktur (vgl. Abschnitt 3.3.1) ist hierbei ein Tabellenblatt vorhanden (in diesem Fall nur das Tabellenblatt *Actuator*). Jedes der Tabellenblätter ist in drei horizontale Bereiche eingeteilt.

Der obere Bereich beinhaltet die *Devices* (Komponenten), der mittlere Bereich beinhaltet die *Device Orders* (Komponentenaufträge) und der untere Bereich beinhaltet die *Preconditions* (Zustandsnachrichtenanforderungen).

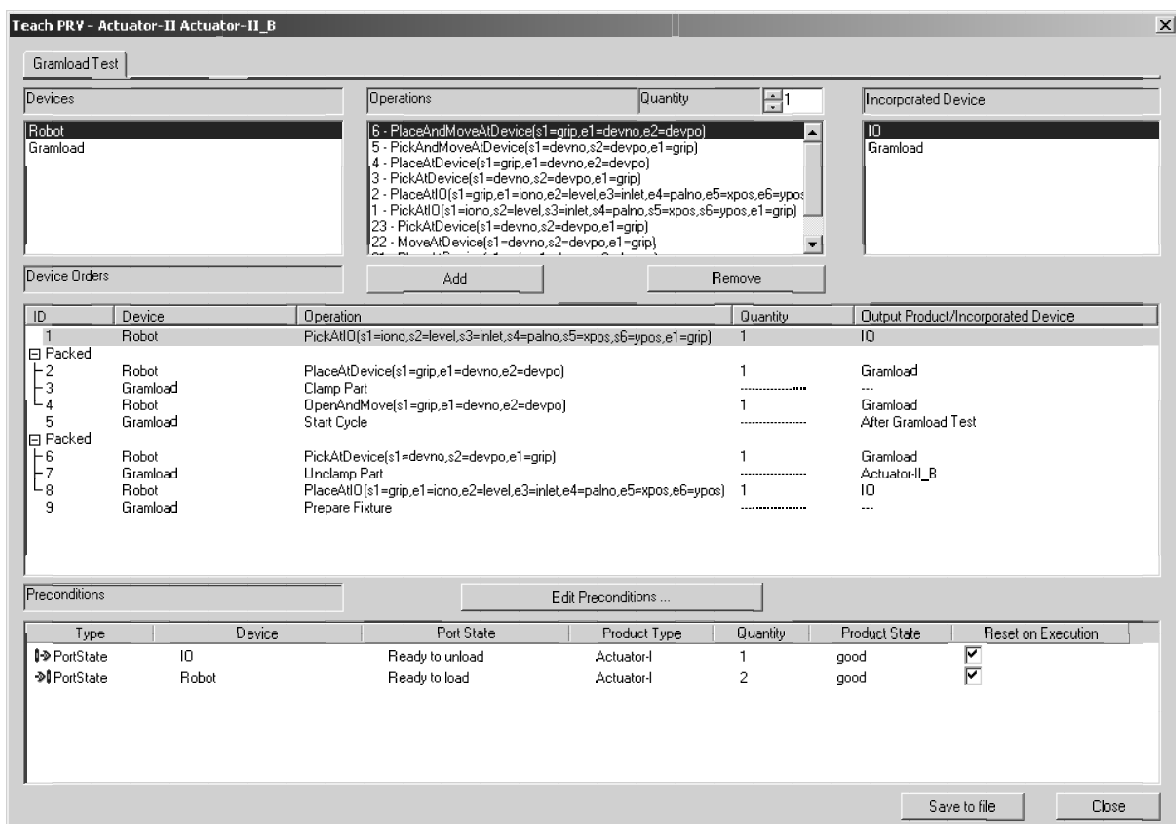


Abb. 6.7: Konfiguration des Leitsystems (© IBM Corporation)

a.) Bereich *Devices*

- *Devices*: Hier sind alle Handhabungskomponenten und alle Montagekomponenten aufgeführt, welche in der entsprechenden Montagezelle vorhanden sind. Die Be-/Entladekomponenten werden nicht mit angezeigt, da für diese der Werkstückträger normiert ist, welcher für den Austausch mit der Verbindungseinheit benutzt wird, und somit keine Aufträge für sie erstellt werden müssen.

- *Operations*: Dies ist die Liste der Aufgaben des ausgewählten *Devices* (Komponente). Sie zeigt alle Aufgaben mit deren jeweiliger Identifikation an, die von der ausgewählten Komponente ausgeführt werden können.
- *Quantity*: Dies ist nur sichtbar, wenn eine Handhabungskomponente in der Liste der *Devices* ausgewählt ist. Hiermit kann ein Komponentenauftrag erstellt werden, der mehrere Produkte gleichzeitig betrifft. Eine Handhabungskomponente kann somit die gegebene Anzahl Produkte gleichzeitig transportieren (insofern sie physikalisch dazu in der Lage ist). Ein solcher Komponentenauftrag wird intern automatisch in mehrere untrennbare Komponentenaufträge für jeweils ein Produkt aufgeteilt.
- *Incorporated Device*: Auch dies ist nur sichtbar, wenn eine Handhabungskomponente in der Liste der *Devices* ausgewählt ist. Bei der Realisierung wurde die Annahme getroffen, dass Handhabungskomponenten nur Transportvorgänge wahrnehmen und damit keine Veränderungen am Produkt vornehmen. Für die Transportvorgänge muss immer eine Komponente angegeben werden, von der oder zu der ein Transportvorgang unternommen wird, je nachdem ob die angegebene Komponente die Quelle oder das Ziel des Transportvorganges ist. Diese anzugebende Komponente ist das *Incorporated Device*.
- *Output Product* (siehe Abb. 6.8): Diese Auswahlmöglichkeit ist nur sichtbar, wenn keine Handhabungskomponenten angewählt sind, sondern Montagekomponenten, Nur Montagekomponenten nehmen Veränderungen an den Produkten vor. Somit muss für Komponentenaufträge der Montagekomponenten das Ausgangsprodukt angegeben werden. Das Ausgangsprodukt wird aus einer Liste von Ausgangsprodukten ausgewählt. Darüber hinaus ist es möglich, neue Ausgangsprodukte, speziell Interimsprodukte, zur Liste hinzuzufügen.

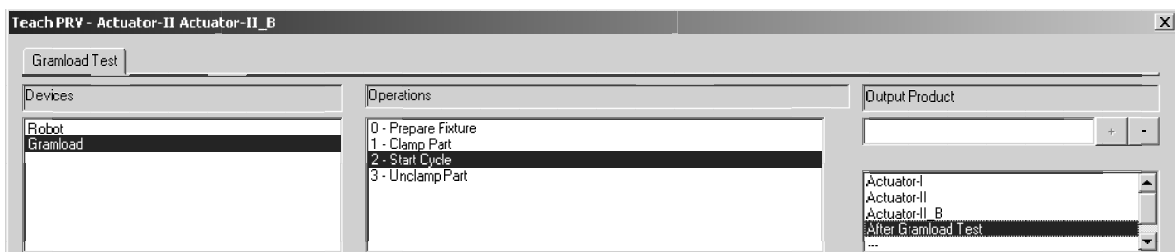


Abb. 6.8: Ausschnitt der Konfiguration der Komponentenaufträge bei einer Montagekomponente (© IBM Corporation)

b.) Bereich *Device Orders*

Hier sind alle konfigurierten Komponentenaufträge des Leitsystems aufgeführt. Es werden alle Komponentenaufträge angezeigt, die für den ausgewählten Montageschritt der erweiterten Produktstruktur benötigt werden. Die Zuordnung zu den einzelnen Komponenten findet für den Benutzer nicht sichtbar intern statt. Durch die Knöpfe *Add* und *Remove* können neue Komponentenaufträge hinzugefügt oder entfernt werden. Für das Hinzufügen von Komponentenaufträgen ist die Auswahl aus dem Bereich *Devices* ausschlaggebend. Jedem hinzugefügten Komponentenauftrag wird hierbei eine eindeutige Identifizierung (*ID*) vergeben. Diese wird in der ersten Spalte der Liste aller Aufträge angezeigt. Mehrere Komponentenaufträge können zu einer Gruppe zusammengefasst werden und mit einer Gruppen-Identifizierung versehen werden (vgl. Abschnitt 5.11). Dies wird in der Spalte *ID* durch das Wort *Packed* angezeigt und mit der Baumdarstellung werden die Gruppenmitglieder verdeutlicht.

c.) Bereich *Preconditions*

Dieser Bereich zeigt alle Zustandsnachrichtenanforderungen (*Preconditions*) für einen Komponentenauftrag an. Es ist die Liste der ZNA für den im Bereich *Device Orders* ausgewählten Komponentenauftrag. Durch den Knopf *Edit Preconditions* gelangt man zur Konfiguration der ZNA für den ausgewählten Komponentenauftrag.

6.2.3 Konfiguration der Zustandsnachrichtenanforderungen

Abbildung 6.9 zeigt die Benutzerschnittstelle, welche die Konfiguration der ZNA für einen zuvor ausgewählten Komponentenauftrag erlaubt. Im Folgenden werden die einzelnen Bereiche der Benutzerschnittstelle beschrieben.

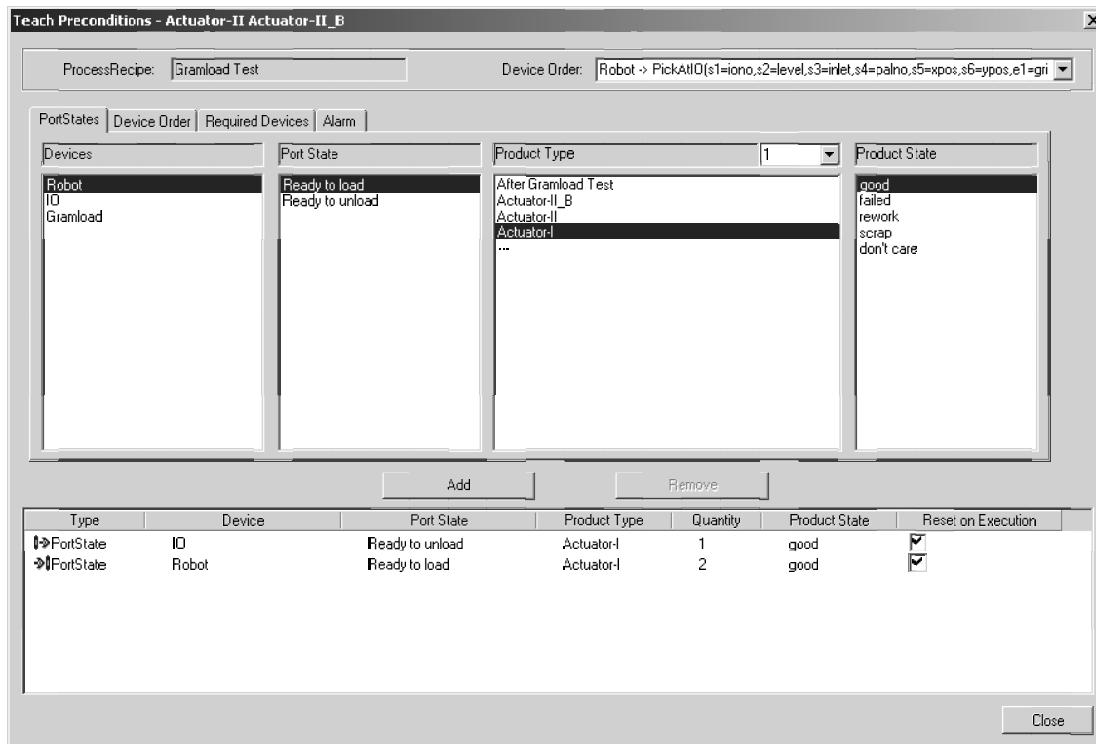


Abb. 6.9: Konfiguration der Zustandsnachrichtenanforderungen (© IBM Corporation)

- *Process Recipe*: Gibt den Montageschritt an, der aktuell durch Auswahl des Tabellenblattes bei der Konfiguration der Komponentenaufträge ausgewählt ist.
- *Device Order*: Der ausgewählte Komponentenauftrag, für den die ZNA konfiguriert werden.
- *Type*: Gibt an, um welche Zustandsnachricht es sich handelt. Es gibt den Typ *Port States*, welcher aus den Zustandsnachrichten *FertigZumBeladen* und *FertigZumEntladen* besteht, den Typ *Device Order*, welcher aus der Zustandsnachricht *KomponentenauftragBeendet* besteht und schließlich den Typ *Alarm*, welcher aus der Zustandsnachricht *Alarm* besteht. Für all diese Typen existieren einzelne Tabellenblätter. Darüber hinaus gibt es noch ein Tabellenblatt *Required Devices*, welches als ZNA dazu dient, einer Komponente den direkten Zugriff auf eine andere Komponente zu gestatten. In der Praxis hat sich gezeigt, dass diese Art von ZNA nicht benötigt und deshalb hier auch nicht genauer darauf eingegangen wird.

- *Devices*: Enthält die Liste aller Komponenten beziehungsweise die ausgewählte Komponente einer ZNA. Im Gegensatz zu der Konfiguration der Komponentenaufträge sind in dieser Liste die Be-/Entladekomponenten vorhanden.
- *Port State*: Die Auswahl, welche der beiden Zustandsnachrichten *ReadyToLoad* (*FertigZumBeladen*) oder *ReadyToUnload* (*FertigZumEntladen*) für die ZNA ausgewählt ist. Dies ist nur sichtbar bei dem Typ *Port States*.
- *Product Type*: Gibt den Produkttyp für die Zustandsnachrichten *FertigZumBeladen* und *FertigZumEntladen* an (vgl. Abschnitt 5.8.8). Dies ist nur sichtbar bei dem Typ *Port States*.
- *Quantity*: Die Anzahl des geforderten Produkttyps bei den Zustandsnachrichten *FertigZumBeladen* und *FertigZumEntladen*. Dies ist nur sichtbar bei dem Typ *Port States*.
- *Product State*: Gibt den Zustand des ausgewählten Produkttypen an. Es gibt die Produktzustände *good* (gut), *failed* (schlecht), *rework* (nachbearbeiten), *scrap* (Ausschuss) und *don't care* (irrelevant). Der Vergleich eines beliebigen Produktzustandes mit dem Produktzustand *don't care* ist immer erfüllt, das heißt sie sind immer gleich (vgl. Abschnitt 5.8.8).
- *Reset on Execution*: Dies ist das Attribut *R* einer ZNA. Das Attribut *R* wird auf *wahr* gesetzt, sobald ein Häkchen gesetzt ist. Ansonsten ist das Attribut *R falsch* (vgl. Abschnitt 5.8.4).

Nachdem die einzelnen Punkte angewählt sind, kann eine ZNA über den Knopf *Add* zu der Liste der ZNA hinzugefügt werden. Entsprechend kann eine ausgewählte ZNA mit dem Knopf *Remove* von der Liste entfernt werden.

In der Liste der ZNA für den angewählten Komponentenauftrag können noch die ZNA angegeben werden, welche die logische Position für die Eingangsprodukte und für das Ausgangsprodukt definieren. Bei der Realisierung müssen die Eingangsprodukte immer von demselben Produkttyp sein. Deshalb wird nur eine ZNA für die Eingangsprodukte ausgewählt. Durch die *Quantity* sind somit mehrere Eingangsprodukte definiert. Die ZNA, welche die logischen Positionen für die Eingangsprodukte liefert, wird mit $\mid \rightarrow$ bezeichnet. Entsprechend wird die ZNA, welche die logische Position für das Ausgangsprodukt liefert, mit $\rightarrow \mid$ bezeichnet.

Kapitel 7

Erprobung und Bewertung des konfigurierbaren Leitsystems

Die Erprobung und Bewertung des entwickelten und realisierten konfigurierbaren Leitsystems wird anhand der Montage eines Actuators vorgenommen (siehe Abb. 7.1). Nach der Vorstellung des erweiterten Produktbaumes wird auf die Konfiguration einiger Montagezellen eingegangen. Hierbei werden nicht alle Montagezellen für die Herstellung eines Actuators gezeigt, sondern nur ein Auszug an Montagezellen, welche bestimmte Problemstellungen aufweisen.

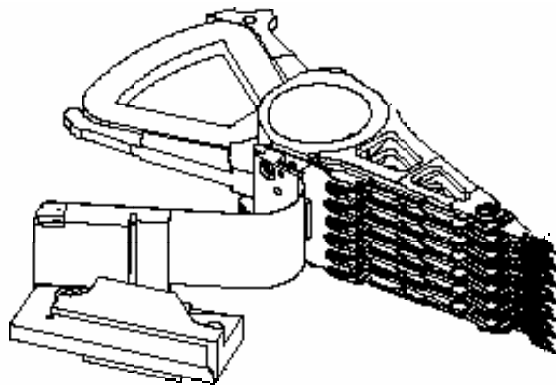


Abb. 7.1: Schematisches Bild eines Actuators (© IBM Corporation)

7.1 Erweiterter Produktbaum eines Actuators

Um die Konfigurationen für die Montagezellen erstellen zu können, muss für das Produkt *Actuator* zuerst einmal die erweiterte Produktstruktur vorhanden sein. Die erweiterte Produktstruktur wird von einem Produktentwickler erzeugt (vgl. Abschnitt 3.3.1). Abbildung 7.2 zeigt eine konkrete Variante sowohl der Produkte als auch der Prozesse einer solchen erweiterten Produktstruktur für einen Actuator (vgl. Abschnitt 3.4). Das endgültige Produkt ist nach dieser erweiterten Produktstruktur der *Actuator-III*. Obwohl bei den Übergängen von dem Produkt *Actuator-I* zu dem

Produkt *Actuator-II* und dann weiter zu dem Endprodukt *Actuator-III* nur noch Messschritte (Gramload Test und Actuator Quasi) vorgenommen werden, sind sie als eigenständige Produkte in der erweiterten Produktstruktur vorhanden. Dies hängt damit zusammen, dass Messschritte zur Primärmontage gehören und somit immer ein neues Produkte erzeugen (vgl. Abschnitt 3.1).

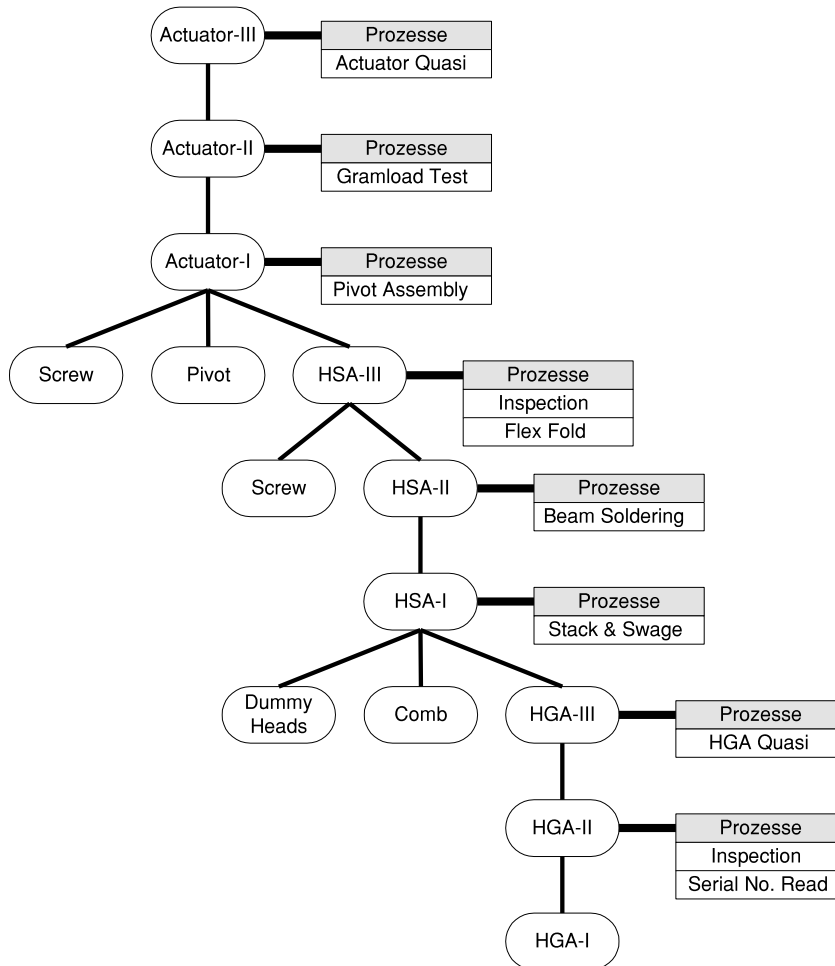


Abb. 7.2: Erweiterte Produktstruktur für einen Actuator

7.2 Konfiguration der Montagezellen

Im weiteren Verlauf werden drei Montagezellen vorgestellt. Jede der Montagezellen erzeugt hierbei ein spezielles Produkt. Es handelt sich hierbei um folgende Montagezellen:

- Flex-Fold
- Pivot-Assembly
- Actuator-Quasi

Jeder dieser drei Montagezellen liegt ein Teil der erweiterten Produktstruktur zugrunde (siehe Abb. 7.3).

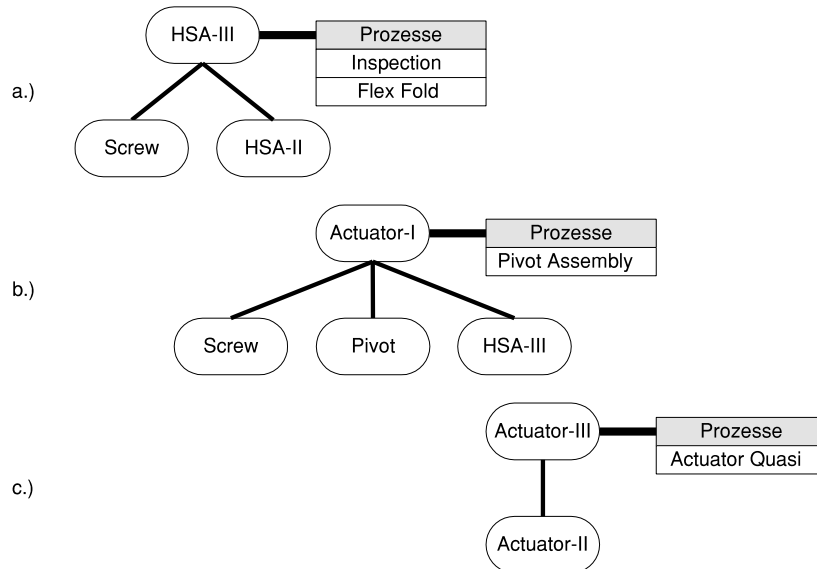


Abb. 7.3: Erweiterte Produktstrukturen für die Montagezellen Flex-Fold (a.), Pivot-Assembly (b.) und Actuator-Quasi (c.)

Alle drei Montagezellen verfügen über eine Be-/Entladekomponente (*IO*) und eine zentrale Handhabungskomponente (*Robot*). Die Komponente *Robot* stellt hierbei folgende Aufgaben zur Verfügung:

- *PickAtIO*: Diese Aufgabe greift ein Produkt von einem Werkstückträger der Komponente *IO* und kehrt zurück in eine sichere Position. Eine sichere Position ist hierbei eine Weltkoordinate, die außerhalb des Wirkungsbereichs einer anderen Komponente in der Montagezelle liegt. Die logische Position des Eingangsproduktes ist die logische Position des zu greifenden Produktes und die logische Position für das Ausgangsprodukt, welches gleich dem Eingangsprodukt ist, da es sich hier um einen reinen Transportvorgang handelt, ist der freie Greifer der Komponente *Robot*.

- *PlaceAtIO*: Ein in einem Greifer der Komponente *Robot* befindliches Produkt wird von einer sicheren Position auf einen Werkstückträger der Komponente *IO* gelegt und anschließend wieder in eine sichere Position zurückgekehrt. Logische Positionen für das Eingangs- und Ausgangsprodukt sind entsprechend der Greifer und der freie Platz auf dem Werkstückträger der Komponente *IO*.
- *PickAndMoveAtDevice*: Diese Aufgabe funktioniert wie die Aufgabe *PickAtIO*, jedoch greift die Komponente *Robot* ein Produkt von einer Montagekomponente.
- *PlaceAndMoveAtDevice*: Entspricht der Aufgabe *PlaceAtIO* mit einer Montagekomponente statt mit der Komponente *IO*.
- *PickAtDevice*: Bei dieser Aufgabe wird das Produkt auf der angegebenen Montagekomponente gegriffen, aber nicht auf eine sichere Position zurückgekehrt, sondern nach dem Greifen in der aktuellen Stellung verharrt.
- *PlaceAtDevice*: Ähnlich der Aufgabe *PickAtDevice* wird hier das Produkt bei der angegebenen Montagekomponente abgelegt, jedoch danach in der Stellung verharrt und der Greifer wird nicht geöffnet.
- *MoveAtDevice*: Die Komponente *Robot* kehrt von der angegebenen Montagekomponente zu einer sicheren Position zurück.
- *OpenAndMoveAtDevice*: Wie bei der Aufgabe *MoveAtDevice* kehrt die Komponente *Robot* auf eine sichere Position zurück, jedoch wird zuvor der Greifer geöffnet.

7.2.1 Konfiguration der Montagezelle Flex-Fold

Die Montagezelle Flex-Fold besteht neben den Komponenten *IO* und *Robot* aus den beiden Montagekomponenten *Inspection* und *FlexFold*. Die Montagezelle muss nun so konfiguriert werden, dass aus dem Produkt *HSA-II*, durch sequenzielles Ausführen des Prozessschrittes *Inspection* gefolgt von dem Prozessschritt *Flex Fold*, das Produkt *HSA-III* entsteht (siehe Abb. 7.3a). Hierbei entsteht nach dem Prozessschritt *Inspection* das Interimprodukt *HSA-II inspected*. Dieses Interimprodukt wird mit dem Prozessschritt *Flex Fold* zum Endprodukt *HSA-III* dieser Montagezelle. Die Schraube (Screw), welche für den Prozessschritt *Flex Fold* benötigt wird, steht der Montagekomponente *FlexFold* als Verbrauchsmaterial zur Verfügung. Sie muss deshalb nicht gesondert bei der Konfiguration berücksichtigt werden.

Die Komponente *Robot* besitzt zwei Greifer, von denen beide in der Lage sind, die Produkte *HSA-II* und *HSA-III* sowie das Interimprodukt *HSA-II inspected* zu greifen. Dadurch ist die Komponente *Robot* in der Lage, bei einer der beiden Montagekomponenten das fertige Produkt mit einem freien Greifer zu fassen, woraufhin ein in dem zweiten Greifer befindliches Produkt für diese Montagekomponente abgelegt werden kann. Auf diese Weise können die Transportwege und somit auch die Stillstandszeiten der Montagekomponenten minimiert werden.

ID (Gr)	Device	Operation	Quantity	OutputProduct/Incorporated Device	
1	Robot	PickAtIO	1	IO	
	P - Type	Device/ID	Port State	Product Type	
	S - PortState	IO	ReadyToUnload	HSA-II	
	E - PortState	Robot	ReadyToLoad	HSA-II	
			Quantity	Product State	Reset
			1	good	x
			2	don't care	x
			1	good	
2	Robot	PickAtIO	1	IO	
	P - Type	Device/ID	Port State	Product Type	
	S - PortState	IO	ReadyToUnload	HSA-II	
	E - PortState	Robot	ReadyToLoad	HSA-II	
			Quantity	Product State	Reset
			1	good	x
			1	don't care	x
			1	good	
3	Robot	PlaceAndMoveAtDevice	1	Inspection	
	P - Type	Device/ID	Port State	Product Type	
	S - PortState	Robot	ReadyToUnload	HSA-II	
	E - PortState	Inspection	ReadyToLoad	HSA-II	
			Quantity	Product State	Reset
			1	good	x
			1	good	x
			1	good	
4	Inspection	Inspect	1	HSA-II inspected	
	P - Type	Device/ID	Port State	Product Type	
	S - PortState	Robot	ReadyToUnload	HSA-II	
	E - PortState	Inspection	ReadyToLoad	HSA-II	
			Quantity	Product State	Reset
			---	---	x
			---	---	
			---	---	
5	Robot	PickAndMoveAtDevice	1	Inspection	
	P - Type	Device/ID	Port State	Product Type	
	S - PortState	Inspection	ReadyToUnload	HSA-II inspected	
	E - PortState	Robot	ReadyToLoad	HSA-II inspected	
			Quantity	Product State	Reset
			1	good	x
			1	don't care	x
			1	good	
6	Robot	PlaceAndMoveAtDevice	1	FlexFold	
	P - Type	Device/ID	Port State	Product Type	
	S - PortState	Robot	ReadyToUnload	HSA-II inspected	
	E - PortState	FlexFold	ReadyToLoad	HSA-II inspected	
			Quantity	Product State	Reset
			1	good	x
			1	good	x
			1	don't care	
7	Robot	PlaceAndMoveAtDevice	1	FlexFold	
	P - Type	Device/ID	Port State	Product Type	
	S - PortState	Robot	ReadyToUnload	HSA-II inspected	
	E - PortState	FlexFold	ReadyToLoad	HSA-II inspected	
			Quantity	Product State	Reset
			1	good	x
			1	good	x
			1	don't care	
8	FlexFold	Fold	1	HSA-III	
	P - Type	Device/ID	Port State	Product Type	
	S - PortState	Robot	ReadyToUnload	HSA-III	
	E - PortState	FlexFold	ReadyToLoad	HSA-III	
			Quantity	Product State	Reset
			---	---	x
			---	---	
			---	---	
9	FlexFold	Fold	1	HSA-III	
	P - Type	Device/ID	Port State	Product Type	
	S - PortState	Robot	ReadyToUnload	HSA-III	
	E - PortState	FlexFold	ReadyToLoad	HSA-III	
			Quantity	Product State	Reset
			---	---	x
			---	---	
			---	---	
10	Robot	PickAndMoveAtDevice	1	FlexFold	
	P - Type	Device/ID	Port State	Product Type	
	S - PortState	FlexFold	ReadyToUnload	HSA-III	
	E - PortState	Robot	ReadyToLoad	HSA-III	
			Quantity	Product State	Reset
			1	good	x
			1	don't care	x
			1	good	
11	Robot	PlaceAtIO	1	IO	
	P - Type	Device/ID	Port State	Product Type	
	S - PortState	Robot	ReadyToUnload	HSA-III	
	E - PortState	IO	ReadyToLoad	HSA-III	
			Quantity	Product State	Reset
			1	good	x
			1	good	x
			1	don't care	

Abb. 7.4: Konfiguration der Montagezelle Flex-Fold

Die Konfiguration der Montagezelle Flex-Fold besteht aus 11 Komponentenaufträgen (siehe Abb. 7.4). Es handelt sich hierbei ausschließlich um die Standardkonfiguration, welche davon ausgeht, dass nur Gutteile entstehen und dass kein Alarm während der Montage auftritt.

Bei den hier aufgeführten ZNA werden, vor dem Typ der ZNA, ihre Rolle als Parameter für die logische Eingangsposition und Ausgangsposition mit den Buchstaben *S* ($\lvert\rightarrow$) und *E* ($\rightarrow\lvert$) bezeichnet.

Die Komponentenaufträge 1 und 2 dienen zur Aufnahme des Produktes *HSA-II* in die Greifer der Komponente *Robot*. Während der Auftrag 1 dafür sorgt, dass nur ein Produkt gegriffen wird, kann mit dem zweiten Auftrag auch ein zweites Produkt unter der Voraussetzung gegriffen werden, dass die Komponente *Inspection* dieses Produkt sofort aufnehmen kann. Diese Voraussetzung in Form der dritten ZNA beim Komponentenauftrag 2 wird nach der Ausführung allerdings nicht ungültig, da dieser Auftrag nichts an dem Zustand der Komponente *Inspection* ändert. Würde man die Komponente *Robot* immer zwei Produkte *HSA-II* aufnehmen lassen, so würde dies zu einer Verklemmung des Systems führen. Das gleiche Vorgehen findet zwischen den Komponenten *Inspection* und *FlexFold* statt (siehe Komponentenaufträge 6 und 7 in Abb. 7.4). Die dritte ZNA des Komponentenauftrages 11 dient zur Wegoptimierung der Komponente *Robot*. Dadurch wird verhindert, dass ein Produkt *HSA-III* bei der Komponente *IO* abgelegt wird, wenn sich gleichzeitig im zweiten Greifer noch ein Produkt *HSA-II inspected* befindet, welches zuvor bei der Komponente *FlexFold* abgelegt werden kann.

7.2.2 Konfiguration der Montagezelle Pivot-Assembly

Die Besonderheit der Montagezelle Pivot-Assembly ist die Durchführung eines Fügeprozesses. Hierzu besitzt diese Montagezelle eine Montagekomponente *PivotAss*. Die benötigten Schrauben (Screw) stehen auch hier der Montagekomponente als Verbrauchsmaterial zur Verfügung. Die Konfiguration besteht nun darin, dass zuerst das Produkt *HSA-III* zur Montagekomponente gebracht wird und anschließend das Produkt *Pivot*. Diese werden in der Montagekomponente verschraubt und ergeben das neue Produkt *Actuator-I*. Eine zusätzliche Herausforderung ist die Tatsache, dass sowohl das Produkt *HSA-III* als auch das Produkt *Pivot* nicht einfach bei der Montagekomponente abgelegt werden können, sondern sie zuerst von der Montagekomponente fixiert (*clamp*) werden müssen, bevor die Komponente *Robot* den Greifer öffnen kann. Entsprechend muss vor dem Abholen des Produktes die Fixierung aufgehoben werden (*unclamp*). Während dieses Vorganges muss sichergestellt sein,

das die Komponente *Robot* nicht eine andere Aufgabe wahrnehmen kann, da dies unweigerlich zu einem Schaden führt. Aus diesem Grund sind die Aufträge, welche das Ablegen, das Fixieren und das anschließende Entfernen zu einer sicheren Position durchführen, gruppiert.

ID (Gr)	Device	Operation	Quantity	OutputProduct/Incorporated Device
1	Robot	PickAtIO	1	IO
	P - Type	Device/ID	Port State	Product Type
	S - PortState	IO	ReadyToUnload	HSA-III
	E - PortState	Robot	ReadyToLoad	HSA-III
			Quantity	Product State
			1	good
			2	don't care
				Reset
				x
				x
ID (Gr)	Device	Operation	Quantity	OutputProduct/Incorporated Device
2	Robot	PickAtIO	1	IO
	P - Type	Device/ID	Port State	Product Type
	S - PortState	IO	ReadyToUnload	Pivot
	E - PortState	Robot	ReadyToLoad	Pivot
			Quantity	Product State
			1	good
			1	don't care
				Reset
				x
				x
ID (Gr)	Device	Operation	Quantity	OutputProduct/Incorporated Device
3 (1)	Robot	PlaceAtDevice	1	PivotAss
	P - Type	Device/ID	Port State	Product Type
	S - PortState	Robot	ReadyToUnload	HSA-III
	E - PortState	PivotAss	ReadyToLoad	HSA-III
			Quantity	Product State
			1	good
			1	good
				Reset
				x
				x
ID (Gr)	Device	Operation	Quantity	OutputProduct/Incorporated Device
4 (1)	PivotAss	Clamp	1	---
	P - Type	Device/ID	Port State	Product Type
	- DeviceOrder	3	---	---
			Quantity	Product State
			---	---
				Reset
				x
ID (Gr)	Device	Operation	Quantity	OutputProduct/Incorporated Device
5 (1)	Robot	OpenAndMoveAtDevice	1	PivotAss
	P - Type	Device/ID	Port State	Product Type
	- DeviceOrder	4	---	---
			Quantity	Product State
			---	---
				Reset
				x
ID (Gr)	Device	Operation	Quantity	OutputProduct/Incorporated Device
6 (2)	Robot	PlaceAtDevice	1	PivotAss
	P - Type	Device/ID	Port State	Product Type
	S - PortState	Robot	ReadyToUnload	Pivot
	E - PortState	PivotAss	ReadyToLoad	Pivot
			Quantity	Product State
			1	good
			1	good
				Reset
				x
				x
ID (Gr)	Device	Operation	Quantity	OutputProduct/Incorporated Device
7 (2)	PivotAss	Clamp	1	---
	P - Type	Device/ID	Port State	Product Type
	- DeviceOrder	6	---	---
			Quantity	Product State
			---	---
				Reset
				x
ID (Gr)	Device	Operation	Quantity	OutputProduct/Incorporated Device
8 (2)	Robot	OpenAndMoveAtDevice	1	PivotAss
	P - Type	Device/ID	Port State	Product Type
	- DeviceOrder	7	---	---
			Quantity	Product State
			---	---
				Reset
				x
ID (Gr)	Device	Operation	Quantity	OutputProduct/Incorporated Device
9	PivotAss	Assemble	1	Actuator-I
	P - Type	Device/ID	Port State	Product Type
	- DeviceOrder	8	---	---
			Quantity	Product State
			---	---
				Reset
				x
ID (Gr)	Device	Operation	Quantity	OutputProduct/Incorporated Device
10 (3)	Robot	PickAtDevice	1	PivotAss
	P - Type	Device/ID	Port State	Product Type
	S - PortState	Robot	ReadyToUnload	Actuator-I
	E - PortState	PivotAss	ReadyToLoad	Actuator-I
			Quantity	Product State
			1	good
			1	good
				Reset
				x
				x
ID (Gr)	Device	Operation	Quantity	OutputProduct/Incorporated Device
11 (3)	PivotAss	Unclamp	1	---
	P - Type	Device/ID	Port State	Product Type
	- DeviceOrder	10	---	---
			Quantity	Product State
			---	---
				Reset
				x
ID (Gr)	Device	Operation	Quantity	OutputProduct/Incorporated Device
12 (3)	Robot	MoveAtDevice	1	PivotAss
	P - Type	Device/ID	Port State	Product Type
	- DeviceOrder	11	---	---
			Quantity	Product State
			---	---
				Reset
				x
ID (Gr)	Device	Operation	Quantity	OutputProduct/Incorporated Device
13	Robot	PlaceAtIO	1	IO
	P - Type	Device/ID	Port State	Product Type
	S - PortState	Robot	ReadyToUnload	Actuator-I
	E - PortState	IO	ReadyToLoad	Actuator-I
	- PortState	Robot	ReadyToLoad	HSA-III
			Quantity	Product State
			1	good
			1	good
			1	don't care
				Reset
				x
				x

Abb. 7.5: Konfiguration der Montagezelle Pivot-Assembly

Die Komponente *Robot* besitzt zwei Greifer, von denen beide in der Lage sind, die Produkte *HSA-III* und *Actuator-I* zu fassen sowie einen Greifer, welcher das Produkt *Pivot* fassen kann. Abbildung 7.5 zeigt eine Konfiguration für die Montagezelle *Pivot-Assembly*.

7.2.3 Konfiguration der Montagezelle Actuator-Quasi

Bei der Montagezelle *Actuator-Quasi* besteht die Besonderheit darin, dass die eine Montagekomponente, die in dieser Montagezelle vorhanden ist, nicht nur eine physikalische Position zur Aufnahme und somit zur Bearbeitung eines Produktes hat, sondern zwei. Während an einem Produkt *Actuator-II* der Prozess durchgeführt wird, so dass es zu dem Produkt *Actuator-III* wird, kann die zweite physikalische Position mit einem anderen Produkt *Actuator-II* befüllt werden. Trotz dieser Möglichkeit der Montagekomponente *ActQuasi* gestaltet sich die Konfiguration sehr einfach, da die ZNA, die der Montagekomponente die logische Position für die Produkte liefern, diese mit sich führen (siehe Abb. 7.6).

Die Komponente *Robot* dieser Montagezelle besitzt zwei Greifer, die beide in der Lage sind, die Produkte *Actuator-II* und *Actuator-III* zu fassen. Dadurch ist es auch hier möglich, eine Wegoptimierung durchzuführen, indem ein Produkt *Actuator-III* bei der Montagekomponente abgeholt und sogleich ein Produkt *Actuator-II* abgelegt wird.

ID (Gr)	Device	Operation		Quantity	OutputProduct/Incorporated Device		
1	Robot	PickAtIO		1	IO		
	P - Type	Device/ID	Port State	Product Type	Quantity	Product State	Reset
	S - PortState	IO	ReadyToUnload	Actuator-II	1	good	x
	E - PortState	Robot	ReadyToLoad	Actuator-II	2	don't care	x
ID (Gr)	Device	Operation		Quantity	OutputProduct/Incorporated Device		
2	Robot	PickAtIO		1	IO		
	P - Type	Device/ID	Port State	Product Type	Quantity	Product State	Reset
	S - PortState	IO	ReadyToUnload	Actuator-II	1	good	x
	E - PortState	Robot	ReadyToLoad	Actuator-II	1	don't care	x
	- PortState	ActQuasi	ReadyToLoad	Actuator-II	1	good	
ID (Gr)	Device	Operation		Quantity	OutputProduct/Incorporated Device		
3	Robot	PlaceAndMoveAtDevice		1	ActQuasi		
	P - Type	Device/ID	Port State	Product Type	Quantity	Product State	Reset
	S - PortState	Robot	ReadyToUnload	Actuator-II	1	good	x
	E - PortState	ActQuasi	ReadyToLoad	Actuator-II	1	good	x
ID (Gr)	Device	Operation		Quantity	OutputProduct/Incorporated Device		
4	ActQuasi	Measure		1	Actuator-III		
	P - Type	Device/ID	Port State	Product Type	Quantity	Product State	Reset
	- DeviceOrder	3	---	---	---	---	x
ID (Gr)	Device	Operation		Quantity	OutputProduct/Incorporated Device		
5	Robot	PickAndMoveAtDevice		1	ActQuasi		
	P - Type	Device/ID	Port State	Product Type	Quantity	Product State	Reset
	S - PortState	ActQuasi	ReadyToUnload	Actuator-III	1	good	x
	E - PortState	Robot	ReadyToLoad	Actuator-III	1	don't care	x
ID (Gr)	Device	Operation		Quantity	OutputProduct/Incorporated Device		
6	Robot	PlaceAtIO		1	IO		
	P - Type	Device/ID	Port State	Product Type	Quantity	Product State	Reset
	S - PortState	Robot	ReadyToUnload	Actuator-III	1	good	x
	E - PortState	IO	ReadyToLoad	Actuator-III	1	good	x
	- PortState	Robot	ReadyToLoad	Actuator-II	1	don't care	

Abb. 7.6: Konfiguration der Montagezelle Actuator-Quasi

Kapitel 8

Zusammenfassung und Ausblick

Die Herstellung von Festplatten muss aufgrund der großen Produktpalette und der sehr kurzen Innovationszeit von neuen Produkten immer flexibler werden. Ein zentraler Punkt bei der Herstellung von Festplatten bildet hierbei die Montage der Einzelteile zu einem Endprodukt. Da die Festplatten immer kleiner und leistungsfähiger werden und somit auch die Anforderungen an die Genauigkeit bei der Montage und die Anforderungen an die Reinheit bei der Montage steigen, muss der Weg hin zu einer Vollautomatisierung folgen. Manuelles Eingreifen in die Montage oder gar manuelle Montage einzelner Baugruppen erfüllt nicht mehr die Anforderungen, die bezüglich Genauigkeit und Reinheit erforderlich sind. Automatisierungsansätze, denen eine solche Vollautomatisierung zugrunde liegt, sind jedoch meist nicht flexibel genug, mehrere Produkte gleichzeitig zu montieren oder den Wechsel zwischen Produkten in wenigen Minuten zu erzielen.

Die Ausgangssituation zeigte, dass die Entwicklung und die Trends bei den Festplatten hin zu kleineren Baugrößen bei immer größeren Kapazitäten geht. Da eine Festplatte aus vielen Einzelteilen besteht, führte dies zu neuen und höheren Ansprüchen an die Montage. Um diesen Ansprüchen gerecht zu werden, wurde eine neue vollautomatische Montagezelle gefordert.

Bei der Analyse des Montageablaufs wurde durch die Einführung der erweiterten Produktstruktur mit den Montagefunktionslisten und den Möglichkeiten von Varianten und Versionen eine Basis geschaffen, aus der die Komponentenbildung innerhalb der Montagezelle und die Anforderungen an ein konfigurierbares Leitsystem abgeleitet wurden. Hierdurch kann die Montagezelle flexibel an neue Gegebenheiten angepasst werden. Aufgrund der großen Anzahl von Montageschritten, welche bei der Montage von Festplatten nötig sind, wurde eine Montageanlage entwickelt, welche mehrere Montagezellen verbindet. Um den Transport der Produkte zwischen verschiedenen Montagezellen zu erreichen, wurde ein normierter Werkstückträger zur Aufnahme unterschiedlichster Produkte benutzt. Da die Montagezellen modular und damit

flexibel für den Umbau auf neue Produkte konzipiert wurden, war die Forderung nach einem konfigurierbaren Leitsystem für all diese modularen Montagezellen da, welches auf einfache Weise an die gegebenen Komponenten und das zu montierende Produkt angepasst werden kann. Die Grundlage für die Konfiguration einer Montagezelle bildet die erweiterte Produktstruktur mit ihren Montagefunktionslisten.

Im Stand der Technik wurden hardwaretechnische als auch softwaretechnische Systeme aufgezeigt, welche für die Steuerung von modularen aber auch nicht modularen Montagezellen benutzt werden können. Da bei der Festplattenmontage keine solchen Leitsysteme vorhanden sind, wurden Lösungsansätze aus anderen Bereichen, in denen Steuerungstechnik und Leitsysteme eingesetzt werden, untersucht. Bei dieser Untersuchung hat jedoch keines der betrachteten Systeme alle Anforderungen, die an ein Leitsystem zur Steuerung einer modularen Montagezelle in der Festplattenmontage gestellt werden, zur Zufriedenheit erfüllt.

Das konfigurierbare Leitsystem wurde wie die Montagezelle selbst in Komponenten aufgeteilt. Jede dieser Komponenten entspricht einer physikalischen Komponente und kann entsprechend ein oder mehrere Aufgaben ausführen. Jede Komponente ist zusätzlich in der Lage, Prozessrezepte auszutauschen und somit neue Funktionalitäten zu erreichen. In eine Montagezelle können auch mehrere Komponenten gleichen Typs eingesetzt werden, um den Durchsatz für die durch diesen Komponententyp realisierte Funktionalität zu erhöhen. Mit Hilfe des Leitsystems können beliebige Montagezellen, die aus unterschiedlichen Komponenten bestehen, für unterschiedliche Produkte konfiguriert werden. Hierzu können Aufträge für jede Komponente angelegt werden. Jeder Auftrag führt eine definierte Aufgabe der Komponente aus. Nach der Ausführung dieser Aufträge verändern sich mehrere Zustände in dem logischen Repräsentanten der Komponente. Diese Änderung führt zu einer Änderung einer Zustandsklasse, in der beliebig viele Zustände zusammengefasst sind, was wiederum eine Zustandsnachricht nach sich zieht, welche zu allen anderen Komponenten gesendet wird. Der Ausführungsstart eines Auftrages hängt von einer Menge beliebiger Zustandsnachrichten ab. Diese werden für jeden Auftrag in Form von Zustandsnachrichtenanforderungen konfiguriert. Somit hängt der Start einer Aufgabe in einer bestimmten Komponente nicht nur von einem bestimmten Zustand oder von einer bestimmten Nachricht ab, sondern von einer Summe von Zuständen, verteilt über beliebig viele Komponenten. Um die Komplexität zu verringern, wurden die Zustandsnachrichten eingeschränkt. Das Leitsystem wird konfiguriert, indem die erstellten Komponentenaufträge mit den Zustandsnachrichtenanforderungen für die Zustandsnachrichten *FertigZumBeladen* und *FertigZumEntladen* (jeweils für ein bestimmtes Produkt), *KomponentenauftragBeendet* und *Alarm* von bestimmten

Komponenten versehen werden. Da die Zustandsnachrichten zum einen Zustände der Produkte beinhalten und zum anderen auch Informationen über die Produkte selbst, ist so auch eine Produktverfolgung möglich.

Die Umsetzung dieses konfigurierbaren Leitsystems wurde schließlich mit Hilfe einer prototypischen Realisierung gezeigt. Die Realisierung bestand aus einer Montageanlage mit mehreren unterschiedlichen Montagezellen. Für drei Montagezellen wurde beispielhaft eine Konfiguration gezeigt. Mit jeder dieser drei Montagezellen wurde dabei eine Besonderheit bei der Konfiguration von modularen Montagezellen herausgestellt. Es hatte sich hierbei gezeigt, dass das hier entwickelte konfigurierbare Leitsystem, selbst mit der Einschränkung auf vier Zustandsnachrichten, den Herausforderungen der Festplattenmontage gewachsen war.

Entwicklungsbedarf besteht bei der Entwicklung eines übergeordneten Planungssystems. Bei der Erprobung hat sich gezeigt, dass es sinnvoll ist, einen Komponentenauftrag nicht sofort zur Ausführung freizugeben, obwohl all seine Zustandsnachrichtenanforderungen erfüllt sind. Das Warten auf einen anderen Komponentenauftrag kann einen besseren Gesamtablauf bieten. Durch eine geschickte Konfiguration der Zustandsnachrichtenanforderungen in den betreffenden Komponentenaufträgen ließ sich dies zwar erzielen, dennoch würde ein übergeordnetes System, welches in der Lage ist, vorausschauend zu planen, die Konfiguration erheblich vereinfachen und somit weniger fehleranfällig machen.

Weiterer Entwicklungsbedarf besteht auch bei der Benutzereingabe der Konfiguration. Obwohl die Art der Konfiguration selbst sehr einfach zu erlernen und auch zu beherrschen ist, so ist die Eingabe durch textuelle Listen, sowohl der Komponentenaufträge als auch deren Zustandsnachrichtenanforderungen, nicht sehr übersichtlich und kann leicht zu fehlerhaften Konfigurationen führen. Deshalb wäre eine graphische Konfigurationseingabe, eventuell basierend auf schon vorhandenen graphischen Notationsstandards, sehr hilfreich. Durch eine übersichtliche graphische Konfiguration ließen sich Fehler in der Konfiguration schon bei deren Erstellung und nicht erst bei deren Erprobung feststellen.

Aufbauend auf der graphischen Konfigurationseingabe wäre die Entwicklung eines Systems sehr hilfreich, welches Verklemmungen in der Konfiguration feststellen und Hinweise geben kann, wie diese Verklemmungen aufgehoben beziehungsweise verhindert werden können. Darüber hinaus kann ein solches System darauf achten, dass keine wichtigen Konfigurationen vergessen werden, insbesondere Konfigurationen die das Fehlermanagement und das Alarmmanagement betreffen.

Kapitel 9

Summary

The rapidly changing technological characteristic of hard discs (i.e. their compactness and storage capacity) enforces hard disc manufacturing companies to produce new products not only in short intervals but also having low prices to establish themselves in the market. Further for the hard disc manufacturers it is becoming important to have a large product manifold of hard discs to satisfy the requirements of current computer systems [IBM02a, Seagate03].

The current hard discs product pallet has to provide products not only for classical desktop computers, servers and mobile equipments but also for consumer electronics such as play consoles, navigation systems, digital cameras and many other upcoming products [Donovan03].

Although other data storage techniques such as flash memories are gaining market acceptance the demand of hard discs in the classical desk top computing, servers and mobile equipments is still growing. The prognosis in the consumer electronics market show that the demand of hard discs is growing more than in the classical areas [Massengill03]. These high expectations are due to the compactness (getting smaller) and storage capacity (growing continuously) of hard discs [Grochowski03, Kozierok01c].

The hard discs to the greatest possible extent are assembled manually or manually with automated transport of individual parts [Bohn99, IBM02b]. This leads to faults during assembly process and also to particulate and electrostatic contamination resulting in the damage of hard discs after a very short time of use.

Currently existing total automation approaches are in the position to assemble the hard discs fully automated but they are restricted to produce one product at a time only. Although these approaches can assemble different products, the time required to change the setup for a new product takes several hours and it is also not possible to assemble different products simultaneously [Seagate01, Seagate02].

To satisfy the requirements such as, miniaturisation (compactness) and continuously growing storage capacity of hard discs, the product manifold and the sales numbers, an assembly cell is required. The assembly cell should be adaptable to the continuously changing requirements described above. To achieve this, a configurable control system for this assembly cell is required. The focus of this work is to design, develop, implement, test and deploy the control system for a fully automated assembly cell.

The introduction of extended product structure, the assembly function lists and the possibilities of variants and versions during the analysis of assembly process forms a basis to build different components in an assembly cell. Further the analysis lead to derive the requirements for a configurable control system. Through this the assembly cell can be flexibly adapted to the new circumstances and conditions.

Based on very large number of assembly steps, which are required to assemble a hard disc, an assembly cluster was developed. This cluster consists of many assembly cells connected with each other. To achieve the transportation of products among the different assembly cells a standardized tray holder to carry different products was used. The conception of modular assembly cells to achieve the flexible dynamical modifications of new products has resulted to several requirements. One of the most important requirements is a configurable control system for all these assembly cells, which can be very easily adapted to the given components and to the products to be assembled. The extended product structure and the assembly function lists build the basis for the configuration of an assembly cell.

The state of the art shows hardware and software systems, which can be used for controlling the modular as well as non modular assembly cells. As for the hard disc assembly such control systems are not available, therefore the solutions in different areas were investigated, where such control systems have been deployed. The investigations lead to the conclusion that none of the examined control systems satisfied all the requirements to control a modular assembly cell, which can be used for hard discs assembly.

The configurable control system was divided into different components as the assembly cell itself. Every one of these logical components corresponds to the physical component and can execute one or more jobs accordingly. In addition every component is in the position to exchange process recipes so that it can change existing functionalities as well as achieve new functionalities. The current state of a job is indicated by a corresponding state machine. The logical component together with the state machines for all jobs is grouped together within a so called logical representative (see Fig. 9.1).

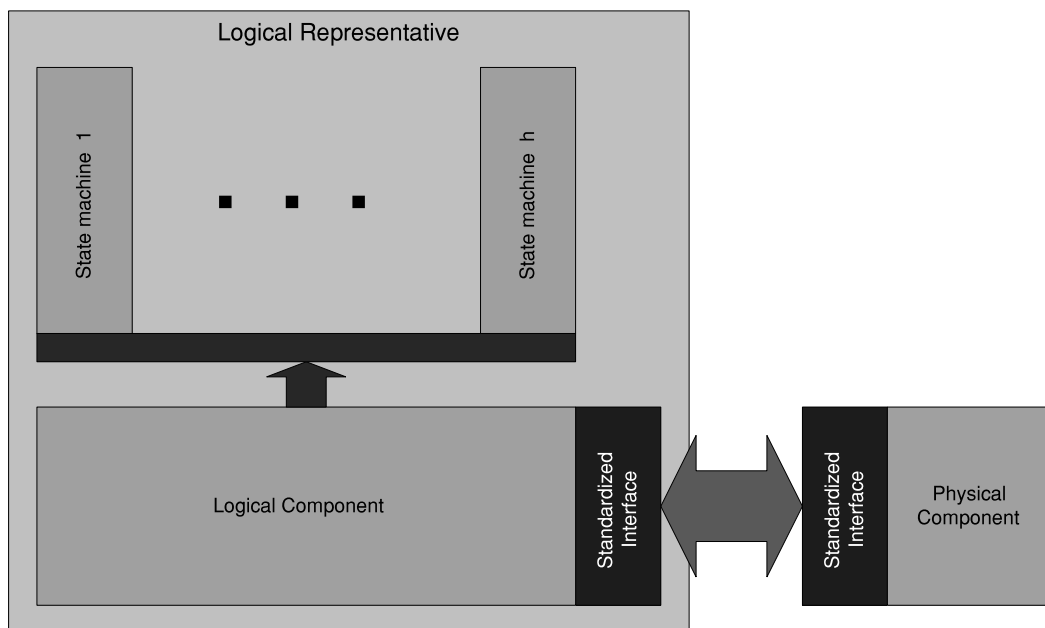


Fig. 9.1: *Logical Representative with logical component corresponding to the physical component and state machines for each possible job*

The control system allows configuring the assembly cell for different products. Since the assembly cell consists of different components, for each of these components orders can be defined. Every component order executes a well defined job of the corresponding physical component. After the execution of such an order, many states will change within the logical representative of the corresponding component. In addition to the state machines for each job the logical representative of physical component comprises of many other state machines, e.g. product state machine, product position state machine (see Fig. 9.2).

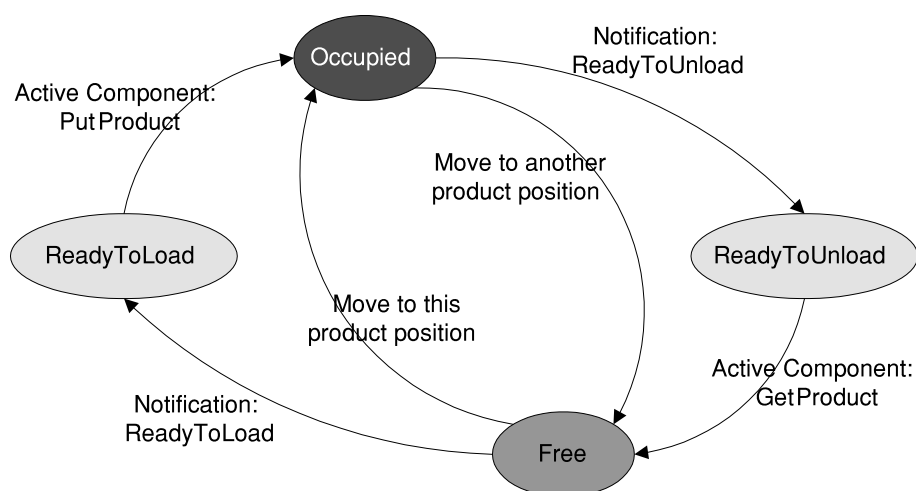


Fig. 9.2: *State machine of a product load/unload position within a component*

All these state machines are grouped in a so called state class. If the execution of an order within the logical representative results in a state change of any state machine, this leads to a change of the state class itself. Each change within the state class sends a state change notification to all the components (see Fig. 9.3).

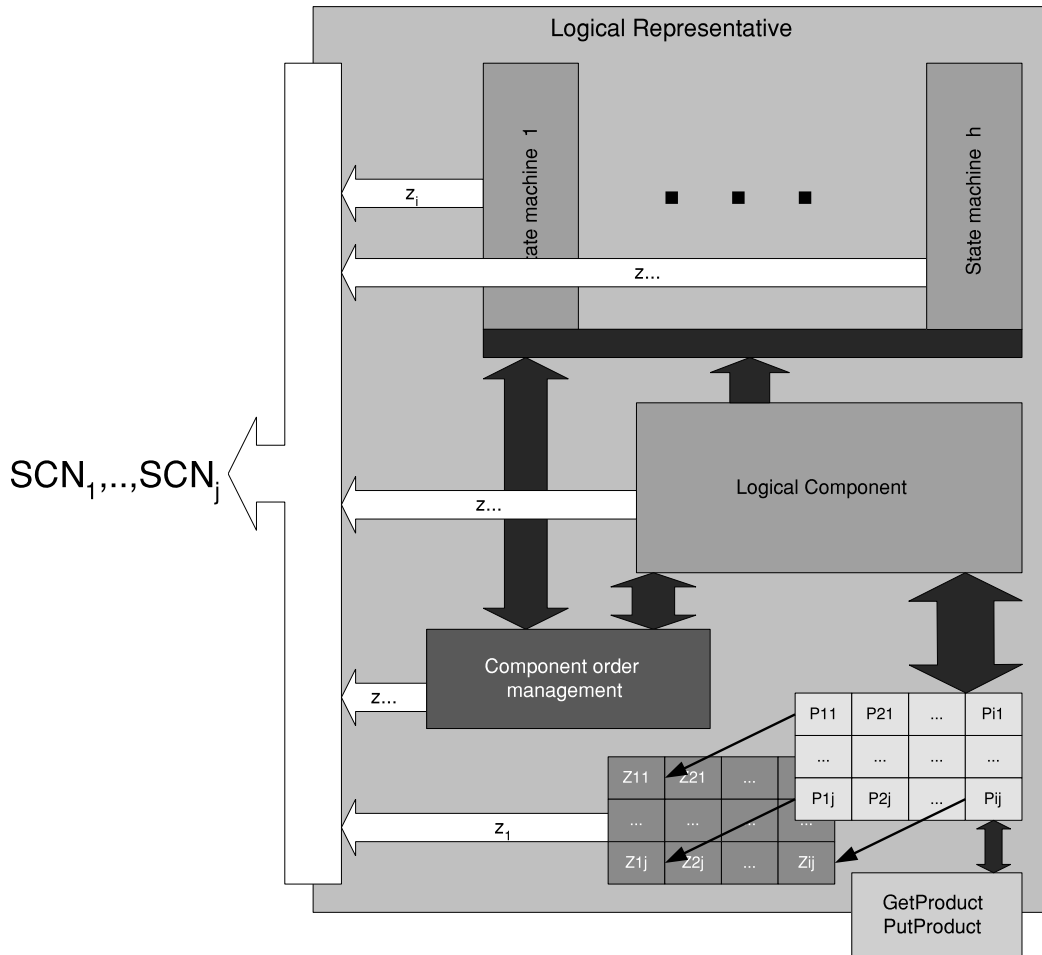


Fig. 9.3: Creation of different state change notifications (SCN) within the logical representative

The execution of a particular component order depends upon a set of many such state change notifications. These are configured as state change notification requirements for each component order. To start a job in a particular component therefore depends not only upon a definite state or notification but also on a sum of states distributed among different components (see Fig. 9.4).

The configuration of the control system is done through assigning an individual component order with the corresponding state change notifications requirements. To reduce the complexity the state change notifications were limited. This means that the following state change notifications *ReadyToLoad* and *ReadyToUnload* (for a particular product) as well as *ComponentOrderFinished* and *Alarm* are to be

configured for every individual component order. As these state change notifications have information about the different product states and also about the product itself this leads to a tracking mechanism of the individual products during the assembly process.

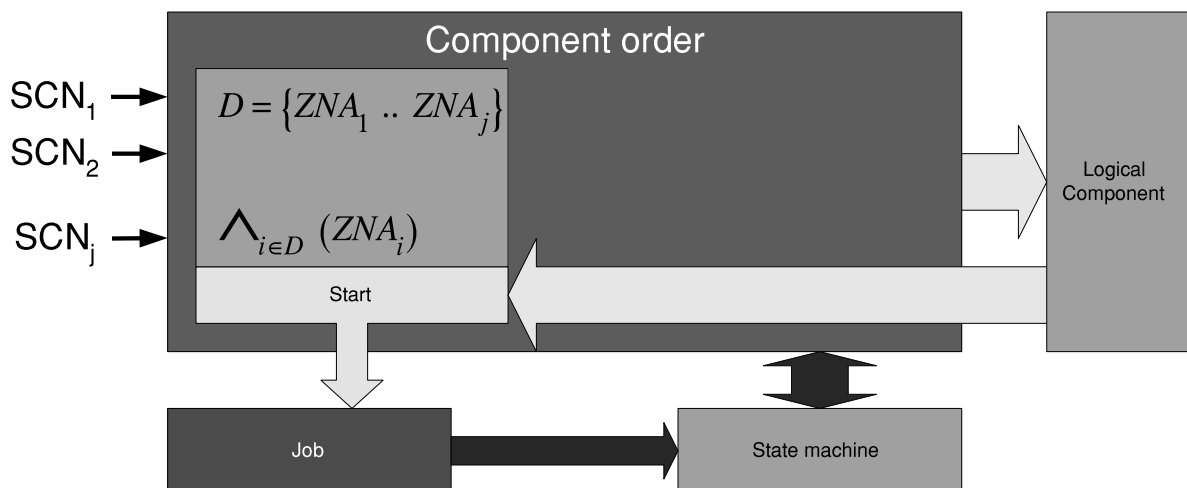


Fig. 9.4: Component order with incoming state change notifications

The deployment of the developed configurable control system is shown by an implementation of a prototype. The realization of the prototype i.e. of configurable control system was done for one assembly cluster with different assembly cells. Configuration examples for three different assembly cells were shown. Each of these three assembly cells has a typical characteristic which is shown in the corresponding configuration. Even though the state change notifications were limited to four different types the configurable control system developed here was in the position to cope with the challenges, which were encountered during the hard disc assembly.

Further research work is required to develop a high level planning system within the assembly cell. The testing showed that it is reasonable not to start a component order immediately even though all its state change notification requirements are fulfilled. For the overall sequence waiting to start an order on other components can result to an optimized assembly sequence. Although through a skillful configuration of state change notification requirements it was possible to achieve an optimized assembly sequence, but for the future research work a system which is in the position to plan beforehand will simplify the configuration of control system and thus will be less fault-prone.

Kapitel 10

Literaturverzeichnis

- [Angerer03] Angerer, T.:
'Komplettmontage mit verteilten NC-Achsen'
In: Mechatronik News (2003)2, S. 4-5
http://www.bayern-mechatronik.de/download/newsletter_2_2003.pdf, (2003)
Letzter Zugriff: 17.06.2004
- [Aspen93] Aspern, J. v.:
'SPS-Softwareentwicklung mit Petrinetzen'
Heidelberg: Hüthig, 1993
- [Bader99a] Bader, U.; Schließer, J.; Weinmann, A.:
'Prozess 2000 - Modulares Fertigungssystem in der Mikrotechnik'
In: wt Werkstattstechnik 89 (1999), Nr. 11/12, S. 485-488
- [Bader99b] Bader, U.; Schließer, J.; Hägele, K.-D.; Weinmann, A.:
'Standardisierungslösungen für ein modulares Fertigungssystem in der Mikrotechnik'
In: VTE 11 (1999), Nr. 6, S. 321-325
- [Bader99c] Bader, U.; Schließer, J.; Weinmann, A.; Schwertschlager, O.:
'Modulares Fertigungssystem'
In: Maschinenmarkt, 105 (1999) Nr. 44, S. 46-48
- [Bader99d] Bader, U.; Schließer, J.; Hägele, K.-D.; Weinmann, A.:
'Offenes Softwarekonzept für ein modulares Fertigungssystem'
In: Elektronik 48 (1999) Nr. 24, S. 20-22

- [Balachandra02] Balachandra, R.:
'Modular Desing and Technological Innovation: The Case of the Hard Disk Drives'
Report 2002-02, The Information Storage Industry Center
<http://isic.ucsd.edu>
Letzter Zugriff: 17.06.2004
- [Bär03] Bär, T.:
'Festplatten: Gegenwart & Zukunft'
In: Windows 2000 Magazin, (2003) Nr. 3, S. 43-46
- [Bartels99] Bartels, J.; Scherff, B.:
'Automatisierungskonzepte im Vergleich'
<http://www.elektroniknet.de/topics/automatisieren/fachthemen/print/1999/99015.htm>, (1999)
Letzter Zugriff: 17.06.2004
- [Bengel02] Bengel, M.; Braatz, A.:
'XML- und Java-basiertes modulares Produktionssystem'
In: Schraft, Rolf-Dieter (Hrsg.) u.a.:
SPS/IPC/Drives 2002: Tagungsband 13. Fachmesse,
26.-28. November 2002, Nürnberg
Heidelberg: Hüthig, 2002, S. 293-301
- [Bengel03] Bengel, M.:
'Modellbasierte Steuerungsarchitektur'
In: wt Werkstattstechnik 93 (2003) Nr. 3, S. 150-152
- [BLE01] N. N.:
'Erstellung eines Tool Controllers als Steuerungssoftware für das Acrobat-Equipment'
Interner Projektbericht des Fraunhofer IPA über ein Projekt mit der Firma B.L.E., Radolfzell, 2000-2001

- [Bohn99]** Bohn, R. E.; Chea, K. S.; Terwiesch C.:
'An Exploration Study of International Product Transfer and Production Ramp-up in the Data Storage Industry'
ISIC, 1999
<http://www-irps.ucsd.edu/irps/faculty/bohn.pdf>
Letzter Zugriff: 22.07.2003
- [Braatz00]** Braatz, A.:
'Spezifikation von dezentralen und verteilten Automatisierungssystemen mit der UML'
In: Westkämper, E. (Hrsg.) u.a.:
Innovative Softwarelösungen für die Automatisierung
Stuttgart, 2000, S. 333-359
(Fraunhofer IPA - Technologie Forum F56)
- [Brandin96]** Brandin, B. A.:
'The Real-Time Supervisory Control of an Experimental Manufacturing Cell'
In: IEEE Transactions on Robotics and Automation, 12 (1996)
Nr. 1, S. 1-14
- [Bullinger86]** Bullinger, H.J. (Hrsg.) u.a.:
'Systematische Montageplanung'
München u.a.: Hanser, 1986
- [Colombo98]** Colombo, A.W.:
'Development and Implementation of Hierarchical Control Structures of Flexible Production Systems Using High-Level Petri Nets'
Bamberg: Meisenbach Verlag, 1998
(Fertigungstechnik - Erlangen 82)
Erlangen-Nürnberg, Universität, Dissertation, 1998

- [Colombo02] Colombo, A. W.; Süssmann, B.; Neubert, R.:
'Eine agentenbasierte Steuerungsarchitektur für intelligente Produktionssysteme'
In: Schraft, Rolf-Dieter (Hrsg.) u.a.:
SPS/IPC/Drives 2002: Tagungsband 13. Fachmesse,
26.-28. November 2002, Nürnberg
Heidelberg: Hüthig, 2002, S. 284-292
- [Delchambre92] Delchambre, A.:
'Computer-aided Assembly Planning'
London: Chapman & Hall, 1992
- [Desrochers95] Desrochers, A. A.:
'Applications of Petri Nets in Manufacturing Systems'
Piscataway: IEEE Press, 1995
- [Dillmann91] Dillmann, R.; Huck, M.:
'Informationsverarbeitung in der Robotik'
Berlin u.a.: Springer, 1991
- [DIN5893] DIN 5893-Teil 1 2003-09:
Fertigungsverfahren Fügen, 2003
- [DIN6789] DIN 6789-Teil 2 1990-09:
Dokumentationssystematik; Dokumentensätze Technischer
Produktdokumentation, 1990
- [DIN19233] DIN 19233 Vornorm 1998-07:
Leittechnik-Prozessautomatisierung-Automatisierung mit
Prozessrechensystemen, 1998
- [DIN61131] DIN 61131 Blatt 3 2003-12:
Speicherprogrammierbare Steuerungen-Teil 3:
Programmiersprachen, 2003
- [Donovan03] Donovan, J.; Kim, J.:
'How did the HDD industry fare in 2002?'
<http://www.idema.org/i4a/pages/index.cfm?pageid=3324>
Letzter Zugriff: 17.06.2004

- [Dutzler02] Dutzler, C.; Glanzer, K.; Zeichen, G.:
'Hochflexible Steuerungsarchitektur für Montageprozesse auf Basis verteilter holonischer Systeme'
In: Schraft, Rolf-Dieter (Hrsg.) u.a.:
SPS/IPC/Drives 2002: Tagungsband 13. Fachmesse,
26.-28. November 2002, Nürnberg
Heidelberg: Hüthig, 2002, S. 203-211
- [EETimes03] www.eet.com:
'Hard disk'
<http://www.eet.com/leftResources/encycSearch?term=Hard+Disk>
Letzter Zugriff: 07.07.2003
- [Ezpeleta97] Ezpeleta, J.; Colom, J. M.:
'Automatic Synthesis of Colored Petri Nets for the Control of FMS'
In: IEEE Transactions on Robotics and Automation, 13 (1997)
Nr. 3, S. 327-337
- [Föederal04] N. N.:
Forschungsprojekt Föederal
<http://www.foederal.org>
Letzter Zugriff: 17.06.2004
- [Freund00] Freund, E.; Theis, K.:
'Verteilte Steuerung flexibler Fertigungszellen'
In: Schraft, Rolf-Dieter (Hrsg.) u.a.:
SPS/IPC/Drives 2000: Tagungsband 11. Fachmesse,
28.-30. November 2000, Nürnberg
Heidelberg: Hüthig, 2000, S. 59-67
- [Frey01] Frey, G.:
'Integration von Petrinetzen in den Steuerungsentwurf nach IEC 61131'
In: Schraft, Rolf-Dieter (Hrsg.) u.a.:
SPS/IPC/Drives 2001: Tagungsband 12. Fachmesse,
27.-29. November 2001, Nürnberg
Heidelberg: Hüthig, 2001, S. 197-205

- [Grochowski03] Grochowski, E.; Halem, R. D.:
'Technological impact of magnetic hard disk drives on storage systems'
IBM Systems Journal, Vol. 42, No. 2, 2003
- [Heger98] Heger, R.:
'Entwicklung eines Systems zur interaktiven Gestaltung und Auswertung von manuellen Montagetätigkeiten in der virtuellen Realität'
Berlin u.a.: Springer, 1998
(IPA-IAO Forschung und Praxis 273)
Stuttgart, Univ., Diss., 1998
- [IBM02a] Röhrsheim, O.; Kobusch, I.:
IBM, Mainz
Gespräch: *'Produktvielfalt bei Festplatten'*
08.02.2002, 9:30h, Mainz
- [IBM02b] Röhrsheim, O.; Kobusch, I.:
IBM, Mainz
Gespräch: *'Manuelle und Halbautomatische Montage von Festplatten'*
21.02.2002, 9:30h, Mainz
- [IBM02c] Röhrsheim, O.; Kobusch, I.:
IBM, Mainz
Gespräch: *'Aufbau einer Festplatte'*
28.02.2002, 9:30h, Mainz
- [IBM02d] Röhrsheim, O.; Kobusch, I.:
IBM, Mainz
Gespräch: *'Verbesserung der Montage von Festplatten'*
19.03.2002, 9:30h, Mainz
- [IBM02e] Röhrsheim, O.; Kobusch, I.:
IBM, Mainz
Gespräch: *'Vorhandene Montagefunktionen bei der Montage von Festplatten'*
09.10.2002, 9:30h, Mainz

- [IBM02f] Kobusch, I.; Kreckel, A.; Lautenbach, F.; Röhrsheim, O.:
IBM, Mainz
Gespräch: *'Aufbau einer Montageanlage für die
Festplattenmontage'*
10.06.2002, 9:30h, Mainz
- [IBM02g] Kobusch, I.; Kreckel, A.; Lautenbach, F.; Röhrsheim, O.:
IBM, Mainz
Gespräch: *'Aufbau einer Montageanlage für die
Festplattenmontage'*
09.07.2002, 9:30h, Mainz
- [IBM03a] Kobusch, I.; Kreckel, A.:
IBM, Mainz
Gespräch: *'Genauigkeit bei der Festplattenmontage'*
21.01.2003, 9:30h, Mainz
- [IBM03b] Kobusch, I.; Kreckel, A.; Lautenbach, F.; Röhrsheim, O.:
IBM, Mainz
Gespräch: *'Montage auf unterschiedlichen Montagezellen'*
28.01.2003, 9:30h, Mainz
- [IBM03c] Lautenbach, F.; Röhrsheim, O.:
IBM, Mainz
Gespräch: *'Verwendung einer Scriptsprache zur Konfiguration
des Leitsystems'*
04.02.2003, 9:30h, Mainz
- [IBM03d] Kreckel, A.; Kobusch, I.; Lautenbach, F.; Röhrsheim, O.:
IBM, Mainz
Gespräch: *'Test des Prototypen MoF'*
12.05.2003, 9:30h, Mainz
- [Idema01] www.idema.org
'Industry Summary/Overview'
<http://www.idema.org/i4a/pages/index.cfm?pageid=3259> (2001)
Letzter Zugriff: 17.06.2004

- [IEE99]** N. N.:
'Durchgängige Offenheit'
IEE 44, (1999), Nr. 9, S. 44-46
- [Iwanitz99]** Iwanitz. F.:
'Katalysator der Automation'
IEE 44, (1999), Nr. 9, S. 16-18
- [Karbo01a]** Karbo, M. B.:
'Hard disks - an introduction'
<http://www.karbosguide.com/hardware/module4b1.htm>
(1996-2001)
Letzter Zugriff: 17.06.2004
- [Karbo01b]** Karbo, M. B.:
'Festplatten - der physikalische Aspekt'
<http://www.karbosguide.com/de/modulg4b2.htm>
(1996-2001)
Letzter Zugriff: 17.06.2004
- [Karbo01c]** Karbo, M. B.:
'Hard disks - areas of developments'
<http://www.karbosguide.com/hardware/module4b3.htm>
(1996-2001)
Letzter Zugriff: 17.06.2004
- [Karbo01d]** Karbo, M. B.:
'Harddisks developments'
<http://www.karbosguide.com/hardware/module4b4.htm>
(1996-2001)
Letzter Zugriff: 17.06.2004
- [Karbo01e]** Karbo, M. B.:
'Hard disks (continued)'
<http://www.karbosguide.com/hardware/module4b5.htm>
(1996-2001)
Letzter Zugriff: 17.06.2004

- [Kaun97] Kaun, R.:
'Verfahren zur Konzeption automatischer reinraumtauglicher Fertigungsanlagen und -zellen'
Berlin u.a.: Springer, 1997
(IPA-IAO Forschung und Praxis 253)
Stuttgart, Univ., Diss., 1997
- [Khan04] Khan, A., California, USA
Gespräch: *'Requirements for a new hard disk manufacturing'*
03.02.2004, 19:00h,
- [Kilian03] Kilian, U.:
'Das magnetische Gedächtnis'
In: Physik Journal, 2 (2003) Nr. 2, Seite 50-51
- [Kobusch03] Kobusch, I.; Röhrsheim, O.; Muckenhirn, R.; u.a.:
'Advanced Recipe Management for Hard Disk Manufacturing'
In: Future Fab international, (2003) 14, S. 67-74
- [Köcher85] Köcher, M.:
'Arbeitsbuch der Montagetechnik' 2. Aufl.
Berlin u.a.: Springer 1985
- [Kozierok01a] Kozierok, C. M.:
'Construction and Operation of the Hard Disk'
The PC Guide, Version 2.2.0, 17. April 2001
<http://www.pcguides.com/ref/hdd/op/index-c.html>
Letzter Zugriff: 17.06.2004
- [Kozierok01b] Kozierok, C. M.:
'Hard Disk Operational Overview'
The PC Guide, Version 2.2.0, 17. April 2001
<http://www.pcguides.com/ref/hdd/op/over-c.html>
Letzter Zugriff: 17.06.2004

- [**Kozierok01c**] Kozierok, C. M.:
'A Brief History of the Hard Disk Drive'
The PC Guide, Version 2.2.0, 17. April 2001
<http://www.pcguid.com/ref/hdd/hist-c.html>
Letzter Zugriff: 17.06.2004
- [**Kozierok01d**] Kozierok, C. M.:
'Hard Disk Trends'
The PC Guide, Version 2.2.0, 17. April 2001
<http://www.pcguid.com/ref/hdd/histTrends-c.html>
Letzter Zugriff: 17.06.2004
- [**Kozierok01e**] Kozierok, C. M.:
'Key Technological Firsts'
The PC Guide, Version 2.2.0, 17. April 2001
<http://www.pcguid.com/ref/hdd/histFirsts-c.html>
Letzter Zugriff: 17.06.2004
- [**Kozierok01f**] Kozierok, C. M.:
'Hard Disk Platters and Media'
The PC Guide, Version 2.2.0, 17. April 2001
<http://www.pcguid.com/ref/hdd/op/media-c.html>
Letzter Zugriff: 17.06.2004
- [**Kozierok01g**] Kozierok, C. M.:
'Platter Size'
The PC Guide, Version 2.2.0, 17. April 2001
<http://www.pcguid.com/ref/hdd/op/mediaSize-c.html>
Letzter Zugriff: 17.06.2004
- [**Kozierok01h**] Kozierok, C. M.:
'Head Actuator'
The PC Guide, Version 2.2.0, 17. April 2001
<http://www.pcguid.com/ref/hdd/op/actActuator-c.html>
Letzter Zugriff: 17.06.2004

- [Kozierok01i] Kozierok, C. M.:
'Early Disk Drives'
The PC Guide, Version 2.2.0, 17. April 2001
<http://www.pcguides.com/ref/hdd/histEarly-c.html>
Letzter Zugriff: 17.06.2004
- [Kursawe98] Kursawe, S.:
'Festplatten'
Poing: Franzis, 1998
- [Licha03] Licha, A.:
'Flexible Montageautomatisierung zur Komplettmontage flächenhafter Produktstrukturen durch kooperierende Industrieroboter'
Bamberg: Meisenbach, 2003
(Fertigungstechnik - Erlangen 138)
Erlangen-Nürnberg, Univ., Diss., 2003
- [Lipschutz89] Lipschutz, S.:
'Lineare Algebra: Theorie und Anwendung'
Hamburg u.a.: McGraw-Hill, 1989
- [Lotter82] Lotter, B.:
'Arbeitsbuch der Montagetechnik'
Mainz: Vereinigte Fachverlage, Krausskopf Ingenieur-Digest,
1982
- [Lotter86] Lotter, B.:
'Wirtschaftliche Montage - Ein Handbuch für Elektrogerätebau und Feinwerktechnik'
Düsseldorf: VDI-Verlag, 1986
- [Lüth98] Lüth, T. C.:
'Technische Multi-Agenten-Systeme'
München: Hanser, 1998

- [**Massengill03**] Massengill, M.:
'The Wave of Change: Storage Rises to New Challenges and Opportunities'
IDC Storage Forum, 20. Mai 2003
<http://www.idema.org/i4a/pages/index.cfm?pageid=3350> (2003)
Letzter Zugriff: 17.06.2004
- [**MOWIMA97**] FISW GmbH; MoWiMa-Konsortium:
'MoWiMa - Modellierung und Wiederverwendung objektorientierter Maschinensoftware'
http://www.isw.uni-stuttgart.de/projekte/mowima/index_d.htm
(1997)
Letzter Zugriff: 17.06.2004
- [**Muckenhirn03**] Muckenhirn, R.; Honisch, F.:
'Manufacturing of the Future'
Interaktiv, Fraunhofer IPA, (2003) Nr. 2, S. 12-13
- [**Muckenhirn04**] Muckenhirn, R.; Dreiss, P.; Dorner, J.; u.a.:
'Future Clinical Analysis: Component Based Framework for Modular Hardware and Software Integration of Different Clinical Equipment'
In: Association for Laboratory Automation:
LabAutomation 2004: The premier international conference on laboratory automation
Feb. 1. - 5., 2004, San José, California, USA
San Diego, CA, USA, 2003, S. 107
- [**Munoz00**] Munoz, P:
'Zeit und Kostenreduktion bei der Projektierung verteilter Automatisierungsaufgaben durch geräteübergreifende komponentenbasierte Workbench'
In: Schraft, Rolf-Dieter (Hrsg.) u.a.:
SPS/IPC/Drives 2000: Tagungsband 11. Fachmesse,
28.-30. November 2000, Nürnberg
Heidelberg: Hüthig, 2000, S. 327-335

- [Oestereich04] Oestereich, B.:
'Objektorientierte Softwareentwicklung: Analyse und Design mit der UML 2.0'
München u.a.: Oldenbourg, 2004
- [OMG04] Object Management Group
'Unified Modeling Language (UML) Version 1.5'
<http://www.omg.org/technology/documents/formal/uml.htm>
(1997-2004)
Letzter Zugriff: 17.06.2004
- [OPC04] OPC Foundation
<http://www.opcfoundation.org>
Letzter Zugriff: 17.06.2004
- [Paschen99] Paschen, H.:
'Vollständig verteiltes SPS-System auf der Basis einer horizontalen Kommunikation in der Sensor-/Aktorebene'
Aachen: Shaker, 2000
(Schriftenreihe des Lehrstuhls für Produktionssysteme 3/99)
Bochum, Ruhr-Univ., Diss., 1999
- [Prehn02] Prehn, E.:
'Realisierung des OMAC Maschinen-Modells mit Zustandsgraphen und der Komponententechnologie'
In: Schraft, Rolf-Dieter (Hrsg.) u.a.:
SPS/IPC/Drives 2002: Tagungsband 13. Fachmesse,
26.-28. November 2002, Nürnberg
Heidelberg: Hüthig, 2002, S. 319-326
- [Pritschow91] Pritschow, G. (Hrsg.):
'Leit- und Steuerungstechnik in flexiblen Produktionsanlagen'
München u.a.: Hanser, 1991

- [Quantum03] Quantum:
'Under data pressure - Data Storage Issues across Europe'
Executive Summary, Juni 2003
<http://www.speicherguide.de/service/download.asp>
Letzter Zugriff: 17.06.2004
- [Ritter01] Ritter, A.; Schaeffer C.:
'Agenten zur Steuerung flexibler Fertigungssysteme'
wt Werkstattstechnik 91, (2001) Nr. 6, Seite 333-337
- [Rostky98] Rostky, G.:
'Disk drives take eventful spin'
<http://www.disktrend.com/disk3.htm> (1998)
Letzter Zugriff: 17.06.2004
- [Schließer98] Schließer, J.:
'Untersuchung von Reinheitssystemen zur Herstellung von Halbleiterprodukten'
Berlin u.a.: Springer, 1998
(IPA-IAO Forschung und Praxis 281)
Stuttgart, Univ., Diss., 1998
- [Schoop00] Schoop, R.; Neubert, R.; Süßmann, B.:
'Flexible Fertigungssteuerung durch SPS, CNC und Software-Agenten'
In: Schraft, Rolf-Dieter (Hrsg.) u.a.:
SPS/IPC/Drives 2000: Tagungsband 11. Fachmesse,
28.-30. November 2000, Nürnberg
Heidelberg: Hüthig, 2000, S. 199-207
- [Schraft03] Schraft, R.-D.; Univ. Stuttgart:
'Automatisierung in der Montage und der Handhabungstechnik'
Vorlesungsskript, Stuttgart, 2003
- [Schuster04] Schuster, A.; Dette, A.:
'Zukünftige Technologien für Massenspeicher'
<http://www.bode.cs.tum.edu/archiv/seminare/Rechnerarch/Platten1.doc>
Letzter Zugriff: 17.06.2004

- [Seagate01] Seagate Technology Media Coverage Details
'This is the factory of the future'
<http://www.seagate-asia.com/sgt/asean/mediacoveragecnt.jsp?articleId=454>
(August 2001)
Letzter Zugriff: 17.06.2004
- [Seagate02] Seagate Technology Media Coverage Details
'Factory of the future aims to enhance our core competence in storage technology'
<http://www.seagate-asia.com/sgt/india/mediacoveragecnt.jsp?articleId=936>, (April 2002)
Letzter Zugriff: 17.06.2004
- [Seagate03a] Seagate Technology
'2003 Summary Annual Report'
http://www.seagate.com/newsinfo/invest/annualreport/2003_report.html
Letzter Zugriff: 11.08.2003
- [Seagate03b] Seagate Technology
'The science of Storage'
<http://www.seagate.com/newsinfo/technology/storage/index.html>
Letzter Zugriff: 11.08.2004
- [Seagate03c] Seagate Technology
'Advanced Concepts: Pushing the Envelope'
<http://www.seagate.com/newsinfo/technology/research/D4b3.html>
Letzter Zugriff: 11.08.2003
- [Seagate03d] Seagate Technology
'Questions and Answers with Mark Kryder'
<http://www.seagate.com/newsinfo/technology/research/D4b4.html>
Letzter Zugriff: 11.08.2003

- [Seagate03e] Seagate Technology
'Investigating Alternative Technologies'
<http://www.seagate.com/newsinfo/technology/research/D4b2.html>
Letzter Zugriff: 11.08.2004
- [Seagate04] Seagate Produkt Informationen
<http://www.seagate.com/products/discfamily/cheetah/index.html>
Letzter Zugriff: 17.06.2004
- [Seitz03] Seitz, M.:
'Speicherprogrammierbare Steuerungen'
München: Hanser, 2003
- [SEMI30] SEMI E30-1000:
'Generic Model for Communications and Control of Manufacturing Equipment'
2000
- [Speicherguide03] speicherguide.de:
'Festplatten - Preise tendieren weiter nach unten'
http://www.speicherguide.de/include/module/mod_print_artikel.asp?todo=1&theID=896
Letzter Zugriff: 11.08.2003
- [Stroustrup97] Stroustrup, B.:
'The C++ Programming Language' 3. Aufl.
Reading u.a.: Addison-Wesley, 1997
- [VDI2083_1] VDI 2083 Blatt 1 2003-06
'Reinraumtechnik: Partikelreinheitsklassen der Luft'
VDI-Gesellschaft Technische Gebäudeausrüstung
1996
- [VDI2083_6] VDI 2083 Blatt 6 1996-11:
'Reinraumtechnik: Personen am Reinen Arbeitsplatz'
1996

- [Vogel-Heuser00] Vogel-Heuser, B.:
'UML/RT for Automation Systems - Introduction and Evaluation'
In: Schraft, Rolf-Dieter (Hrsg.) u.a.:
SPS/IPC/Drives 2000: Tagungsband 11. Fachmesse,
28.-30. November 2000, Nürnberg
Heidelberg: Hüthig, 2000, S. 117-124
- [Wattrodt98] Wattrodt, K.:
'Speichertechnologie und -herstellung im globalen Wettbewerb'
[http://www.informatik.uni-mainz.de/VDI/
veranstaltungen98.html](http://www.informatik.uni-mainz.de/VDI/veranstaltungen98.html) (1998)
Letzter Zugriff: 17.06.2004
- [Watts00] Watts, H.:
'Improving Fab Performance'
In: Future Fab international, (2000), 9
- [Wenk02] Wenk, M.:
*'Entwicklung eines konfigurierbaren Steuerungssystems für die
flexible Sensorführung von Industrierobotern'*
Bamberg: Meisenbach, 2002
(Fertigungstechnik - Erlangen, 131)
Erlangen-Nürnberg, Univ., Diss., 2002
- [WesternDigital04] Western Digital Produkt Informationen
<http://www.westerndigital.com> (2001-2004)
Letzter Zugriff: 17.06.2004
- [Westkämper01a] Westkämper, E.:
*'Modulare Konzepte senken die Kosten in der Anlaufphase neuer
Produkte'* (Interview)
In: Industrieanzeiger 123 (2001) Nr. 36, S. 137

- [Westkämper01b]** Westkämper, E.; Braatz, A.:
'Eine Methode zur objektorientierten Softwarespezifikation von dezentralen Automatisierungssystemen mit der Unified Modeling Language (UML)'
In: *Automatisierungstechnik* at 49 (2001), Nr. 5, S. 225-232
- [Westkämper02]** Westkämper, E.; Warnecke, H.-J.:
'Einführung in die Fertigungstechnik' 5. Aufl.
Stuttgart: Teubner, 2002
- [Wonham03]** Wonham, W. M.:
'Notes on control of Discrete-Event Systems'
ECE 1636F / 1637S 2003-04
Toronto, Univerity
Edward S. Rogers Sr. Dept. of Electrical & Computer
Engineering, Systems Control Group