

Ein Beitrag zur verteilten und adaptiven
Materialflusssteuerung für Stückguttransport

Dissertation

zur Erlangung des akademischen Grades

Doktoringenieur
(Dr.-Ing.)

von Dipl.-Wirtsch.-Ing. Kai Lorentz
geb. am 17.12.1969 in Braunschweig

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

Gutachter: Prof. Dr.-Ing. Hermann Kühnle
Prof. Dr. techn. Dietmar Dietrich

Promotionskolloquium am 4.07.2003

[...]

Und keine Zeit und keine Macht zerstückelt
Geprägte Form, die lebend sich entwickelt.

ΔΑΙΜΩΝ, Dämon

Urworte. Orphisch

Johann Wolfgang Goethe

Vorwort des Verfassers

Die vorliegende Arbeit entstand während meiner Tätigkeit am Fraunhofer-Institut für Fabrikbetrieb und -automatisierung IFF und am Institut für Arbeitswissenschaft, Fabrikautomatisierung und Fabrikbetrieb (IAF) der Otto-von-Guericke-Universität Magdeburg.

Dem Leiter des Instituts für Arbeitswissenschaft, Fabrikautomatisierung und Fabrikbetrieb und meinem Doktorvater, Herrn Prof. Dr.-Ing. Hermann Kühnle, bin ich für die stete Diskussionsbereitschaft und beständige Förderung dieser Arbeit zu besonderem Dank verpflichtet.

Ebenso gilt mein besonderer Dank Herrn Prof. Dr. techn. Dietmar Dietrich, der die Zweitbegutachtung und den langen Weg aus Wien auf sich genommen hat.

Darüber hinaus gilt mein Dank allen Mitarbeitern des Centers Verteilte Systeme, dabei insbesondere Herrn Dr. Axel Klostermeyer, Herrn Dr. Arndt Lüder und Herrn Ralf Messerschmidt, für die unermüdliche Diskussionsbereitschaft. Nicht unerwähnt sollen in dieser Aufzählung die Herrn Thomas Fuchs, Mario Lucke und Sascha Hildebrand bleiben, die als studentische Hilfskräfte eine große Hilfe waren.

Mein ganz besonderer Dank gilt meiner Familie, ohne deren Unterstützung ich dies alles nicht geschafft hätte.

Magdeburg, im Juli 2003

Kai Lorentz

VERZEICHNIS DER TABELLEN UND ABBILDUNGEN	VI
ABKÜRZUNGEN UND FORMELZEICHEN	VIII
1. EINLEITUNG	1
2. STEUERUNG DES MATERIALFLUSSES VON STÜCKGUTTRANSPORTEN .	4
2.1. Steuerungsaspekte	4
2.2. Adaptive Materialflusssteuerung	7
2.3. Stückguttransporte	9
2.4. Anforderungen an eine verteilte und adaptive Steuerungsmethode für Stückgüter	11
3. STAND DER TECHNIK	14
3.1. Materialflusssteuerung	14
3.2. Modellierungsverfahren	16
3.3. Verteilte Steuerungen	20
3.4. Adaptive Routing-Algorithmen	23
3.5. Zusammenfassende Betrachtung des Standes der Technik .	26
4. ZIELSETZUNG UND VORGEHENSWEISE	28
5. GRUNDLAGEN	31
5.1. Materialflussmodule	31
5.2. Softwarekomponenten	34
5.3. Modellierung	37
5.4. Algorithmus	42
5.5. Protokoll	48
6. DIE SOFTWAREKOMPONENTEN	52
6.1. Definition der Funktionen	52
6.2. Datenhaltung	56
6.3. Plug-and-Play	58
6.3.1. <i>Anmeldung im System (Funktion P1)</i>	58
6.3.2. <i>Entgegennahme Anmeldung im System (Funktion P2)</i> .	58
6.3.3. <i>Verschicken von Heartbeats (Funktion P3)</i>	60
6.3.4. <i>Entgegennahme von Heartbeats (Funktion P4)</i>	63
6.4. Topographiebildung	67

6.5.	Wegfindung	69
7.	MFM ÜBERGREIFENDE ABLÄUFE	70
7.1.	Systemstart	71
7.2.	Normaler Betrieb	75
7.3.	Hinzufügen eines MFM	77
7.4.	Ausfall eines MFM	79
7.5.	Heartbeat zu spät	81
8.	ADAPTIVES MATERIALFLUSSSYSTEM	84
8.1.	Beschreibung des Referenz-MFS	85
8.2.	Modellbildung	89
8.3.	Simulation	92
8.4.	Umsetzung am Referenz-MFS	97
8.5.	Auswertung	102
9.	ZUSAMMENFASSUNG UND AUSBLICK	104
10.	GLOSSAR	107
11.	LITERATURVERZEICHNIS	109
ANHANG A: XML-BESCHREIBUNGEN DER DATEN UND MESSAGES		131
ANHANG B: PROTOKOLLE DER SIMULATION		136
ANHANG C: PROTOKOLLE DER UMSETZUNG		141

Verzeichnis der Tabellen und Abbildungen

Abbildung 2-1: Einordnung verschiedener Steuerungsarchitekturen in das Enslowsche Modell.....	6
Abbildung 2-2: Denkbare Hardwarekonfiguration und Steuerungsarchitektur der Materialflussteuerung.....	8
Abbildung 3-1: Stand der Technik in der Steuerungsarchitektur modularer MFS.....	16
Abbildung 3-2: Irreduzibler Transportknoten.....	18
Abbildung 4-1: Aufbau und Vorgehensweise der Arbeit.....	30
Abbildung 5-1: Ein modulares Materialflusssystem.....	32
Abbildung 5-2: Erweiterte Steuerungsarchitektur innerhalb eines MFM.....	36
Abbildung 5-3: Modell eines MFM.....	39
Abbildung 5-4: Beispiel eines MFS.....	42
Abbildung 5-5: Sequenzdiagramm der Messages.....	51
Abbildung 6-1: Übersicht entwickelter Softwarekomponenten und Funktionen, in Relation zur gesamten Steuerungsfunktion des MFM.....	55
Abbildung 6-2: Funktionen und Datennutzung.....	57
Abbildung 6-3: Pseudocode <i>Entgegennahme Anmeldungen</i>	60
Abbildung 6-4: Pseudocode <i>Heartbeat - Klasse 1</i>	61
Abbildung 6-5: Pseudocode <i>Heartbeat - Klasse2</i>	62
Abbildung 6-6: Pseudocode <i>Heartbeat - Klasse3</i>	63
Abbildung 6-7: Pseudocode <i>Entgegennahme Heartbeats - Knoten_streichen</i>	64
Abbildung 6-8: Pseudocode <i>Entgegennahme Heartbeats - Aktualisieren</i>	66
Abbildung 7-1: Sequenzdiagramm Szenario 1.....	73
Abbildung 7-2: Sequenzdiagramm Szenario 2.....	76
Abbildung 7-3: Sequenzdiagramm Szenario 3.....	78

Abbildung 7-4:	Sequenzdiagramm Szenario 4.....	80
Abbildung 7-5:	Sequenzdiagramm Szenario 5.....	82
Abbildung 8-1:	Referenz-MFS.....	84
Abbildung 8-2:	Aufteilung des Referenz-MFS in MFM.....	86
Abbildung 8-3:	Steuerungstechnik des MFS.....	88
Abbildung 8-4:	Graph des Referenz-MFS.....	91
Abbildung 8-5:	Modulmonitor-Fenster.....	93
Abbildung 8-6:	Zieleingabe-Fenster.....	94
Abbildung 8-7:	Transportobjektmonitor-Fenster.....	94
Abbildung 8-8:	Screenshot Simulationslauf.....	95
Abbildung 8-9:	Implementierung der Softwarekomponenten innerhalb der SPS.....	99
Tabelle 7-1:	Erläuterungen zum Sequenzdiagramm 1.....	74
Tabelle 7-2:	Erläuterungen zum Sequenzdiagramm 2.....	77
Tabelle 7-3:	Erläuterungen zum Sequenzdiagramm 3.....	79
Tabelle 7-4:	Erläuterungen zum Sequenzdiagramm 4.....	81
Tabelle 7-5:	Erläuterungen zum Sequenzdiagramm 5.....	83
Tabelle 8-1:	Typisierung der implementierten Messages....	100

Abkürzungen und Formelzeichen

Abkürzung / Formelzeichen	Bedeutung
A	Ausgangsknoten eines MFM
a	Ausgangsast eines MFM
AN	Ausgangsast-Nachfolgertabelle
B	Menge der Äste von T
c()	Kostenfunktion eines Astes
COB	Cyclic Organisation Block
DIN	Deutsches Institut für Normung
DP	Dezentrale Peripherie
DTD	Document Type Definition
E	Eingangsknoten eines MFM
e	Eingangsast eines MFM
EN	Eingangsast-Nachfolgertabelle
FB	Function Block
G	Graph
IDA	Interface for Distributed Automation e.V.
IEC	International Electrotechnical Commission
IP	Internet Protocol
j	Laufvariable
K	Zentralknoten eines MFM
LON	Local Operating Network
LVR	Lagerverwaltungsrechner

m	Laufvariable
MFM	Materialflussmodul
MFR	Materialflussrechner
MFS	Materialflusssystem
n	Laufvariable
o	Laufvariable
P1	Funktion <i>Anmeldung im System</i>
P2	Funktion <i>Entgegennahme Anmeldung im System</i>
P3	Funktion <i>Verschicken von Heartbeats</i>
P4	Funktion <i>Entgegennahme von Heartbeats</i>
PAS	Public Available Specification
PB	Program Block
PLC	Programmable logic controller
r	Anzahl der Eingänge eines MFM
s	Anzahl der Ausgänge eines MFM
SB	Sequential Block
SGML	Standard Generalized Markup Language
SPS	Speicherprogrammierbare-Steuerung
T	Graph eines MFM
T1	Funktion <i>Lege Ausgangsast-Nachfolgertabelle ab</i>
T2	Funktion <i>Lösche Ausgangsast-Nachfolgertabelle</i>
T3	Funktion <i>Erstelle Eingangsast-Nachfolgertabelle</i>
T4	Funktion <i>Liefere Ausgangsast-Nachfolgertabellen</i>
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

UML	Unified Modelling Language
UST	Untergelagerte Steuerung
V	Menge der Knoten von T
VDE	Verein Deutscher Elektroingenieure
VDI	Verein Deutscher Ingenieure
VDMA	Verein Deutscher Maschinen und Anlagenbauer
w	Laufvariable
W1	Funktion <i>Wegberechnung</i>
W2	Funktion <i>Zuordnung</i>
x	Laufvariable
XML	Extensible Markup Language
y	Laufvariable
z	Laufvariable
ZN	Zentralknoten-Nachfolgertabelle

Die Zitationsweise der Arbeit entspricht Teil 2 *Zitierregeln* /DIN 84/ und Teil 3 *Literaturverzeichnisse* /DIN 95/ der *DIN 1505 Titelangaben von Dokumenten*. Verweise auf Quellen des Internets erfolgen - in Ermangelung eines deutschen Standards - entsprechend den Vorgaben des Teils 2 *Electronic documents and parts thereof* der ISO 690 /ISO 97/.

1. Einleitung

Die Anforderungen an die Logistik produzierender Unternehmen in Bezug auf Flexibilität¹ und Geschwindigkeit /Gudehus 99, Graf 00, Eisenschmid 01/ sind in den letzten Jahren aufgrund neuer Trends wie Mass-Customization, Virtuelles Fabrizieren /Kühnle 99, Lorentz 99, Behrens 00, Kühnle 00b/ und E-Commerce /Rupp 00, Gora 01, Wildemann 01/ stetig gestiegen. Vor diesem Hintergrund reichen innovative Organisationskonzepte, wie sie seit Anfang der 90er Jahre vertreten werden /Womack 92, Warnecke 93/, allein nicht aus. Es fehlt an den geeigneten technischen Lösungen, die einerseits eine gleichwertige Alternative zu bestehenden Konzepten darstellen und andererseits die Wandlungsfähigkeit der Unternehmensstrukturen durch technische Anpassungsfähigkeit² unterstützen /Westkämper 97, Wiendahl 98, Kühnle 99, Reinhard 99/.

Eine Verbesserung der Anpassungsfähigkeit muss also auch für die Betriebsmittel der innerbetrieblichen Logistik erreicht werden, stellt diese doch das Bindeglied zwischen den einzelnen Bearbeitungsstätten innerhalb eines Betriebes dar /Sommerer 98, Jünemann 00/. Ein integraler Bestandteil der innerbetrieblichen Logistik sind dabei Materialflusssysteme³. Diese werden im Normalfall spezifisch für die zu fertigenden Produkte und die damit verbundenen Abläufe ausgelegt. Da MFS üblicherweise über verzweigte Streckenführungen verfügen, um Variantenvielfalt, Redundanz oder Sortierfunktionen zu erlauben /Pfeifer 98/, ist die Materialflusssteuerung des Systems von entscheidender Bedeutung. Umbaumaßnahmen, die durch eine Variation des

¹ Flexibilität muss als Flexibilität des gesamten Systems Logistik verstanden werden /Göpfert 00/.

² Anpassungsfähigkeit wird hier nach /Hartmann 95/ als die Fähigkeit eines Systems zur aktiven, schnellen Anpassung der Strukturen auf zeitlich nicht vorhersehbar wechselnde Aufgaben aus eigener Substanz verstanden.

³ Materialfluss ist die Verkettung aller Vorgänge beim Gewinnen, Be- und Verarbeiten sowie bei der Verteilung von Gütern innerhalb festgelegter Bereiche /DIN 83, VDI 73/. Im weiteren Verlauf soll der Begriff Materialflusssysteme durch MFS abgekürzt werden.

Materialflusses⁴ entstehen, sind derzeit sehr aufwendig, weshalb eine materialflusssseitige Anpassungsfähigkeit⁵ nur sehr bedingt gegeben ist /Dürschmidt 97/. Dies ist auf eine unbefriedigende Modularisierung der Anlagenteile und der diese steuernden Steuerungstechnik⁶ zurückzuführen /Krämer 99/. Daraus ergibt sich die Notwendigkeit für eine neue Steuerungsmethodik des Materialflusses, die die allgemeine Anpassungsfähigkeit der logistischen Betriebsmittel unterstützt⁷. Diese geforderte Anpassungsfähigkeit findet als Umsetzung von verteilter Intelligenz⁸ neuerdings Eingang in die Überlegungen der industriellen Steuerungstechnik /Kühnle 00, IDA 01, Patz 01, Kühnle 01, Lorentz 01b/. Die Studie *PLCs: Is There a Future?* der ARC Advisory Group sieht darin den wichtigsten Trend für die industrielle Steuerungstechnik der Zukunft /ARC 01/. Die Einführung verteilter Intelligenz in die Steuerungstechnik stellt somit die Grundlage⁹ für eine anpassungsfähige Materialflussteuerung und muss deshalb in engem Zusammenhang mit der (übergeordneten) Steuerung des Materialflusses gesehen werden. Die vorliegende Arbeit

⁴ Für eine Übersicht der Ursachen für Änderungen an MFS vgl. /Brinkmann 87/.

⁵ Im Kontext der Steuerung des Materialflusses ist Anpassungsfähigkeit als die Fähigkeit eines MFS zu verstehen, einerseits ohne großen Konfigurationsaufwand in Betrieb genommen werden zu können, andererseits neue Module - beispielsweise neue Teilstrecken nach einem Umbau - ohne Rekonfiguration des bestehenden MFS integrieren zu können /Messerschmidt 02b/.

⁶ Der bisherige zentralistische Ansatz innerhalb der Steuerungstechnik von MFS /Düll 77/ führt zu Lösungen mit geringer Anpassungsfähigkeit, da eine Veränderung des MFS eine manuelle Adaption der Materialflussteuerung nach sich zieht /Brinkmann 87, Lang 97/.

⁷ Marke stellt diese Forderung bereits 1991 auf /Marke 91/. Siehe dazu auch z.B. /Bosch 02/.

⁸ Unter verteilter Intelligenz versteht man im Rahmen der Steuerungstechnik die Aufteilung der bisher zentral erfolgten Steuerung auf mehrere, miteinander vernetzte, intelligente Steuerungsmodule /Buchwitz 01/.

⁹ Verteilte Steuerungssysteme sind durch ihre Skalierbarkeit, die es ermöglicht, ein System beliebig in seiner Größe zu verändern, gekennzeichnet.

entwickelt eine Steuerungsmethode¹⁰ zur verteilten und adaptiven Materialflussteuerung, die die Anpassungsfähigkeit der logistischen Betriebsmittel entsprechend steigert. Dabei setzt sich diese Methode aus einer Definition der Materialflussmodule, einem Algorithmus¹¹ zur verteilten und adaptiven Wegfindung und einem, diesen Algorithmus einschließenden, Protokoll¹² zusammen.

¹⁰ Methode ist laut Brockhaus /Brockhaus 98/ als ein nach Gegenstand und Ziel planmäßiges (methodisches) Verfahren, die Kunstfertigkeit einer Technik zur Lösung praktischer und theoretischer Aufgaben, speziell das Charakteristikum für wissenschaftliches Vorgehen definiert.

¹¹ Ein Algorithmus soll in diesem Zusammenhang als ein endliches Verfahren zur Lösung von Problemen verstanden werden, dessen elementare Schritte und deren Reihenfolge angegeben sind /Hering 95, Jungnickel 99/.

¹² Unter einem Protokoll wird im folgenden ein Satz von Regeln zum Austausch von Nachrichten zwischen Partnerinstanzen verstanden /Stehle 01/.

2. Steuerung des Materialflusses von Stückguttransporten

2.1. Steuerungsaspekte

Laut DIN 19226 Teil 1 ist Steuerung als der Vorgang in einem System definiert, bei dem eine oder mehrere Größen als Eingangsgrößen aufgrund der dem System eigentümlichen Gesetzmäßigkeiten andere Größen als Ausgangsgrößen beeinflussen /DIN 94/. Dabei weist die DIN jedoch explizit darauf hin, dass der Begriff Steuerung nicht nur für den oben beschriebenen Vorgang Verwendung findet, sondern auch allgemeingültig für die elektrische Anlage verwendet wird, in der die Steuerung stattfindet¹³. Aufgrund der steigenden Systemkomplexität geht der Trend seit einiger Zeit zu dezentralen und räumlich verteilt aufgebauten Steuerungssystemen¹⁴ /Langmann 96, MOVA 00/. Innerhalb der industriellen Steuerungstechnik finden auch teilweise die Begriffe „Verteilte Intelligenz“ /IDA 01, Wenzel 01, Plagemann 01/ oder dezentrale intelligente Automatisierungstechnik /Bruland 95/ Verwendung. Dabei handelt es sich analog zur Informatik um autonome, an ein Kommunikationsnetz angeschlossene Steuerungen und Systeme für die Datenspeicherung, die im Sinne der Erfüllung einer gemeinsamen Steuerungsaufgabe miteinander agieren /Dietrich 98/. Nach Enslow /Enslow 74, Enslow 78/ lassen sich generell fünf Merkmale zur Charakterisierung von Steuerungssystemen feststellen:

- Hardwarearchitektur,

¹³ Vgl. /Konhäuser 98/ zum übergreifenden Zusammenhang der Steuerungstechnik.

¹⁴ Dieser Begriff entstammt der Informatik und wurde ursprünglich bezüglich Mehrrechner- und Multiprozessorenarchitekturen verwendet, wobei sich diese grundsätzlich durch die folgenden drei Komponenten auszeichnen: Eine Menge autonomer Prozessoren, ein Kommunikationsnetzwerk und Software, die diese Komponenten zu einem kommunizierenden System integriert /Schroeder 93, Teil 00/.

- Art der Datenverarbeitung,
- Art der Datenspeicherung,
- Systemsteuerung,
- Systemtransparenz.

Die Punkte eins bis vier sind unter Zusammenfassung der Punkte zwei und drei in einem dreidimensionalen Koordinatensystem in Abbildung 2-1 dargestellt. Derzeit am Markt befindliche Steuerungskonzepte, die auf traditionellen Feldbussen¹⁵ wie Profibus oder Interbus¹⁶ und damit auf einer homogenen Hardware basieren, verwenden Master-Slave Verfahren. Davon ausgehend fand eine Migration der Funktionen sowohl hinsichtlich der Verteilung der Verarbeitung¹⁷ als auch hinsichtlich der zum Einsatz kommenden Hardware¹⁸ statt. Das Konzept der Local Operating Networks (LON)¹⁹ ist bereits soweit verteilt, dass die Datenverarbeitung dezentral ohne die Übergabe von Zugriffsrechten erfolgt. Dabei sind jedoch die einzelnen zum Einsatz kommenden Module so klein, dass eine lokale Datenspeicherung nicht möglich ist. Vollständig verteilte Steuerungssysteme, deren Eigenschaften das Zusammenspiel autonomer, heterogen aufgebauter Steuerungen mit lokalen Datenbeständen sind, befinden sich zur Zeit noch in der Spezifikation /IDA 01, Wenzel 01/. Die genannten

¹⁵ Unter Feldbussen werden hier analog zur VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik alle Bussysteme im prozessnahen Anwendungsbereich verstanden /Kriesel 98/. Für einen Überblick und eine Einordnung vgl. /Gruhler 00/.

¹⁶ Profibus ist eine Entwicklung der Firma Siemens, Interbus ist eine Entwicklung der Firma Phoenix Contact. Beide Bussysteme wurden in der internationalen Norm *Feldbusse für industrielle Leitsysteme* /IEC 02/ normiert.

¹⁷ Vgl. hierzu die Erweiterung zu Profibus DP (Dezentrale Peripherie).

¹⁸ Hier seien die vielen am Markt erhältlichen Gateways (vgl. z.B. /WT 01/) genannt, die unterschiedlichste Hardware überbrücken.

¹⁹ Bei LON handelt es sich um ein Produkt der Firma Echelon. Es ist das erste und heute weitestgehend in der Gebäudeautomatisierung eingesetzte Konzept verteilter Steuerungssysteme /Tönnissen 97, Knackfuß 97/.

Architekturkonzepte sind wie folgt gegen vollständig verteilte Systeme abzugrenzen:

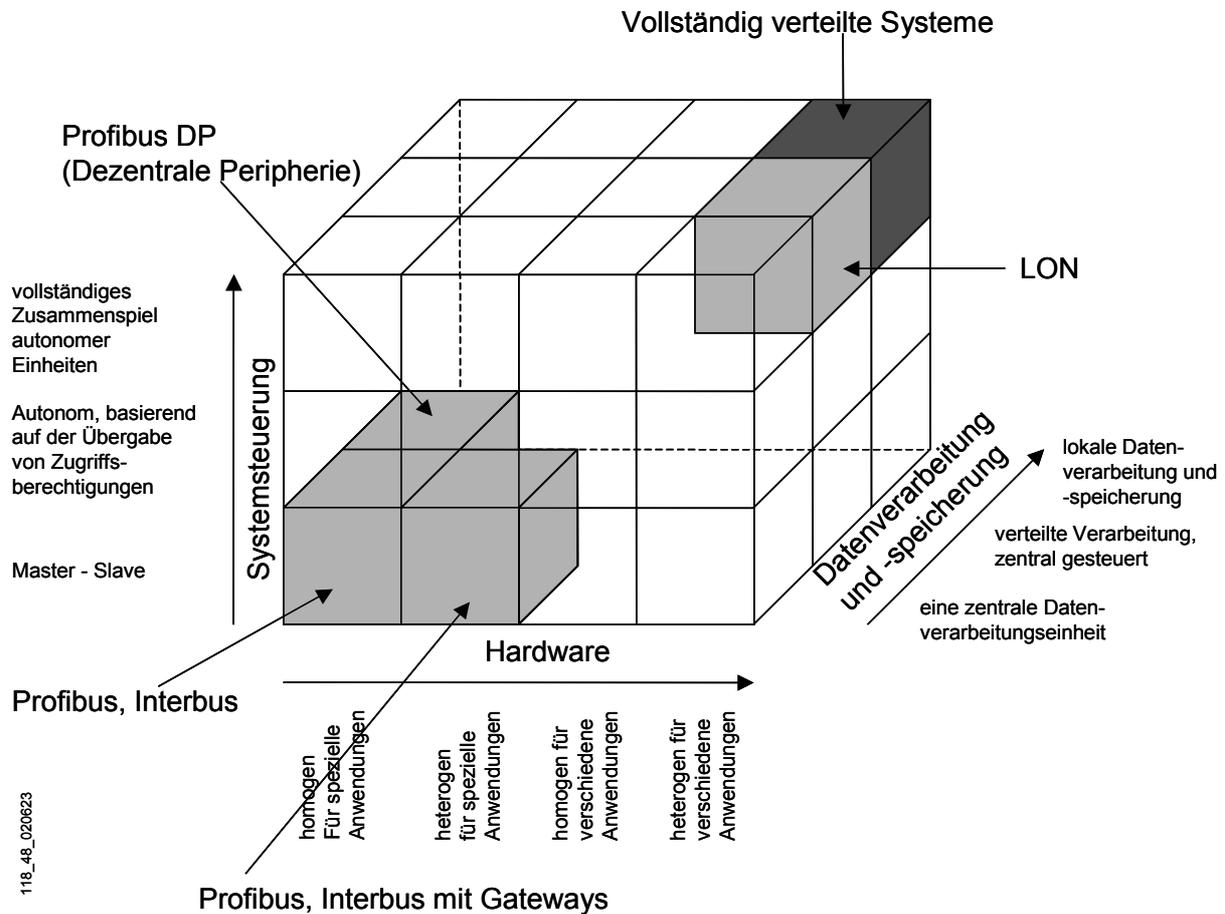


Abbildung 2-1: Einordnung verschiedener Steuerungsarchitekturen in das Enslowsche Modell

Bezüglich verteilter Steuerungssysteme ist darüber hinaus die Granularität²⁰ der einzelnen Komponenten ein entscheidendes Beschreibungskriterium, da sich damit die Größe der verwendeten

²⁰ Granularität, von lat. Granulum (Körnchen). Ein Granulat zeichnet sich durch Gleichartigkeit in Korngröße und Kornform aus /Brockhaus 98/. Im folgenden soll der Begriff Granularität Verwendung finden, um die Aufteilung der Steuerungsfunktionen zu beschreiben. Eine Aufteilung in mehrere kleinere Steuerungen besitzt also eine feinere Granularität als einige wenige große Steuerungen. Dabei muss die Granularität vor dem Hintergrund der betrachteten Funktion verstanden werden: Die Funktion „Motor ansteuern“ lässt sich wiederum in die Sub-Funktionen Temperaturfühler überwachen, Schütz schalten, etc. auflösen. Panreck /Panreck 02/ verwendet den Begriff analog für die zunehmende Zerlegung von Systemen in Subsysteme im Rahmen einer Modellierung.

Steuerungen beschreiben lässt. Die betrachteten, vollständig verteilten Systeme zeichnen sich also durch heterogene, autonome Einheiten mit lokaler Datenverarbeitung und -speicherung aus, deren Granularität von der zu erfüllenden Steuerungsfunktion abhängig ist. Die Eingrenzung der Steuerungsfunktion - und damit implizit der Granularität - erfolgt im nächsten Kapitel.

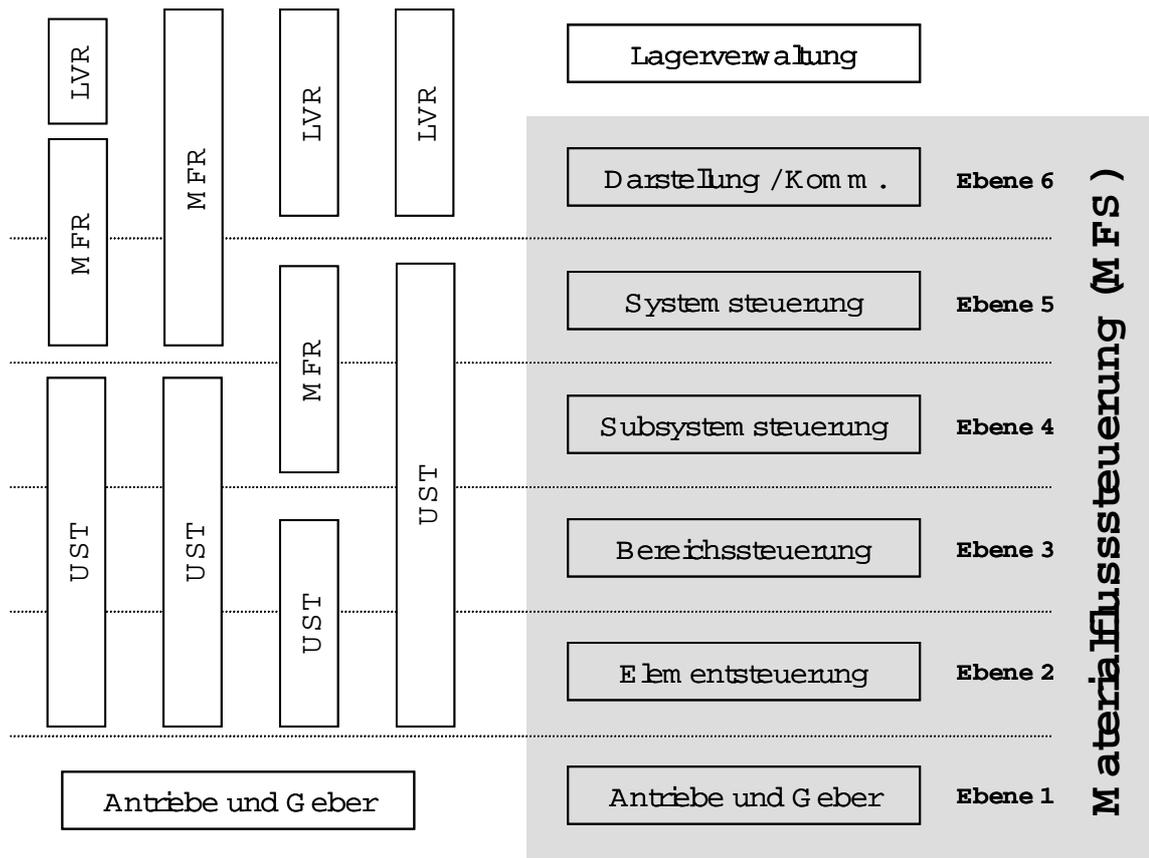
2.2. Adaptive Materialflusssteuerung

Im VDMA-Einheitsblatt *Datenschnittstellen in Materialflusssteuerungen* /VDMA 94/ werden die unterschiedlichen Funktionen²¹ von Materialflusssteuerungen definiert und diesen Funktionen unterschiedliche Ebenen für ihre Implementierung zugewiesen. Wie aus Abbildung 2-2 ersichtlich ist, wird aber im Rahmen des VDMA-Einheitsblattes die reine Materialflusssteuerung mit den übrigen Steuerungsfunktionen²² innerhalb der Steuerung vermischt. Im Gegensatz zu der VDI-Richtlinie 3628 /VDI 96/²³ wird die Verbindung der Steuerungsfunktionen mit der Materialflusssteuerung dabei auf allen Ebenen des Steuerungssystems zugelassen. Materialflusssteuerungen werden also entsprechend der DIN 19226 Teil 1 als Gesamtsystem, bestehend aus Steuerungshardware und -Software, Aktoren und Sensoren, verstanden /VDMA 94/.

²¹ Für eine weiterführende Diskussion der Funktionen der Materialflusssteuerung vgl. /Härdtner 91/.

²² Zu diesen gehören beispielsweise Notausfunktionen, Verriegelungen, etc.

²³ Die VDI-Richtlinie 3628 *Schnittstellen zwischen den Funktionen im Automatisierungsmodell* schlägt eine Implementierung nur innerhalb der unteren Ebenen vor.



118_03_021023

LVR : Lagerverwaltungsrechner
 MFR : Materialflussrechner
 UST : Untergeordnete Steuerung

Abbildung 2-2: Denkbare Hardwarekonfiguration und Steuerungsarchitektur der Materialflusssteuerung /VDMA 94/

Bezüglich der Funktionen der Materialflusssteuerung soll im weiteren Verlauf dieser Arbeit die Wegfindung²⁴ für das Transportgut betrachtet werden. Die zu erreichende Adaptivität²⁵

²⁴ Grundsätzlich kann die Wegfindung des Materialflusses als Generierung entsprechender Sub-System Koordinaten aus Quelle-Ziel Koordinaten der darüber liegenden Ebene verstanden werden. Dies erfolgt durch Verknüpfung der Daten des Transportauftrages mit den Daten, die das spezifische MFS informationstechnisch als zeitlich veränderliches Modell abbilden /VDMA 94/.

²⁵ Adaptivität ist als Fähigkeit eines technischen Systems definiert, mit Sensoren automatisch eine Korrektur der Sollvorgabe vorzunehmen und sich Änderungen in der Umgebung anzupassen /Böge 98/.

bezieht sich also auf die Wegfindung, die adaptiv in Bezug auf das Gesamt-MFS ermöglicht wird. Die Funktion der adaptiven Wegfindung ist vor allem bei der paketorientierten²⁶ Vermittlungsweise innerhalb der Kommunikationstechnik zum Einsatz gekommen. Dabei passt sich die Wegfindung vor allem der Topologie und Netzlast an: Adaptive Verfahren für die Wegfindung innerhalb von Kommunikationsnetzen sind als Verfahren definiert, die versuchen, die aktuelle Topologie des Netzes oder die momentanen Lastverhältnisse²⁷ in die Berechnungen einzubringen /Lang 97, Rauber 00/.

2.3. Stückguttransporte

Bei einem MFS handelt es sich notwendigerweise um ein offenes System²⁸, da das MFS grundsätzlich als zwischen Quelle und Senke liegend zu verstehen ist. Materialfluss kann dabei laut VDI Richtlinien 2411 /VDI 70/ und 2689 /VDI 74/ in sechs Grundfunktionen aufgeteilt werden, wobei im Rahmen dieser Arbeit der Aspekt „Fördern“ untersucht wird. Die verschiedenen Formen der Förderer als Verteileinrichtungen werden von Axmann /Axmann 93/ und der VDI Richtlinie 2340 /VDI 69/ dargestellt. Üblicherweise werden Förderer in Unstetigförderer und Stetigförderer differenziert /Brinkmann 87/. Beide Kategorien können nochmals in flurfrei und flurgebunden differenziert werden.

²⁶ Generell wird zwischen verbindungsorientierten und verbindungslosen Verfahren unterschieden. Bei der verbindungsorientierten Verbindung wird eine Verbindung von Sender zu Empfänger „durchgeschaltet“. Bei der verbindungslosen Vermittlung wird immer nur ein Paket durch das Kommunikationsnetz gelenkt /Meyer 99/.

²⁷ Vgl. Scheideler /Scheideler 97/ zu den Vorteilen von adaptiven Routing-Algorithmen.

²⁸ Ausgehend von der allgemeinen Definition eines System als „die Summe seiner Teile und als die Summe der Beziehungen zwischen den Teilen“ wird unter einem offenen System ein System verstanden, das in Austauschbeziehung mit seiner Umgebung steht /Ropohl 99/. Im Falle der Materialflusssysteme bestehen Austauschbeziehungen bezüglich Materie, Energie und Informationen /Bahke 74/.

Als Stückgüter werden in der Fördertechnik gemeinhin alle Gegenstände bezeichnet, die ohne Rücksicht auf Form und Gewicht während des Förderns als Einheit betrachtet werden können, wobei diese Definition bezüglich Gewicht, Maßen und Beschaffenheit eingegrenzt werden kann /Axmann 93/. Darauf aufbauend kann zwischen MFS mit nicht identifizierten Stückgütern und solchen mit identifizierten Stückgütern unterschieden werden. Die Identifikation erfolgt dabei über Barcodes, induktive Sensoren, oder Transponder /Jansen 01/. Transponder bieten dabei die Möglichkeit, die gespeicherten Daten auf dem Werkstück dynamisch anzupassen²⁹. Im Rahmen dieser Arbeit werden Stetigförderer³⁰ mit identifizierten Stückgütern betrachtet, da diese notwendigerweise einer Wegfindung bedürfen.

²⁹ Für einen vollständigen Überblick über die möglichen Steuerungsmittel vgl. VDI-Richtlinie 2339 *Zielsteuerungen für Förder und Materialflußsysteme* /VDI 99/ und VDI-Richtlinie 2515 Blatt 2: *Identträger in Stückgutfördersystemen - Mobile Datenspeicher* /VDI 98/.

³⁰ Eine genauere Einteilung der Stetigförderer soll im Rahmen dieser Arbeit unterbleiben, sie kann der DIN 15201 *Stetigförderer - Benennungen* entnommen werden /DIN 94a/.

2.4. Anforderungen an eine verteilte und adaptive Steuerungsmethode für Stückgüter

Wie in Kapitel 2.2 beschrieben wurde, basiert die Wegfindung des Materialflusses grundsätzlich auf einer Verknüpfung von Zieldaten mit Daten, die das Modell des MFS enthalten. Diese Funktion kann als eine das MFS überspannende Gesamtfunktion betrachtet werden und erfolgt deshalb derzeit innerhalb einer zentralen, das gesamte MFS steuernden Instanz - dem Materialflussrechner. Für die Entwicklung einer verteilten und adaptiven Steuerungsmethode ist es notwendig, eine dynamische Modellbildung zu ermöglichen. Das zur Steuerung herangezogene Modell muss durch das MFS dynamisch erweiterbar sein, um die geforderte Anpassungsfähigkeit des MFS an Erweiterungen, Umbauten, etc. zu ermöglichen. Weiterhin kann diese Anpassungsfähigkeit nur durch konsequente Modularisierung des MFS erreicht werden, was eine Verteilung der bisher zentralen Funktion „Steuerung des Materialflusses“ unausweichlich macht. Daraus ergeben sich die folgenden Anforderungen an die zu entwickelnde Steuerungsmethode:

Die Steuerungsmethode muss eine vollständige Modularisierung des MFS erlauben: Die Steuerungsmethode soll eine derartige Verteiltheit besitzen, dass das Gesamtsystem ohne zentrale Einheit auskommt.

Die Erfüllung der Funktion Wegfindung muss bei einem verteilten System gewährleistet sein: In Kapitel 2.1 wurde ausgeführt, dass in einem vollständig verteilten System ausschließlich ein Zusammenspiel autonomer Einheiten mit lokalen Datenbeständen besteht. Daraus ergibt sich für die Erfüllung der gemeinsamen Aufgabe der Wegfindung somit auch die Notwendigkeit für einen verteilten Algorithmus³¹. Wie bereits dargelegt wurde, werden für die Wegfindung Daten über das Modell des MFS benötigt. Zur

³¹ Laut Maurer ist ein Algorithmus verteilt, wenn er für die Bearbeitung durch mehrere in einem Netz verbundene Rechner, ohne Rückgriff auf gemeinsamen Speicher entworfen ist /Maurer 99/.

Erstellung dieser Daten und ebenso für die Wegfindung selbst bedarf es des verteilten Algorithmus. Da in einem verteilten System keine bevorzugten Teile existieren, muss weiterhin jeder einzelne Teil des MFS alle zur Erfüllung der Materialflusssteuerung relevanten Funktionen enthalten³².

Die Steuerung des MFS soll selbstkonfigurierend erfolgen: Der verteilte Algorithmus der Steuerungsmethode muss die Modellbildung selbständig ausführen. Dafür müssen die einzelnen Module innerhalb des MFS erkennen, wann eine neue Konfiguration erforderlich ist und entsprechend den Algorithmus aufrufen. Dafür ist ein Protokoll nötig, das die Anmeldung neuer Teile im MFS erlaubt und die Überwachung des MFS durch sich selbst übernimmt.

Die Wegfindung soll wegoptimal erfolgen: Die Steuerungsmethode soll bei alternativen Wegmöglichkeiten eine Optimierung des Transportweges durchführen. Dabei ist es unerheblich, ob die Optimierung nach Auslastung oder Kosten erfolgen soll.

Die Steuerungsmethode muss die erforderliche Robustheit besitzen: Die mit verteilten Systemen zusammenhängenden Problemstellungen wie Ausfälle einzelner Teile, Ausfälle des Kommunikationsnetzes oder Verspätungen einzelner Nachrichten müssen innerhalb der Steuerungsmethode, also innerhalb des Protokolls und des Algorithmus, berücksichtigt werden³³.

Die Steuerungsmethode soll eine leichtgewichtige Implementierung erlauben: Da die Steuerungsmethode für MFS mit teilweise sehr kleinen Systembausteinen zum Einsatz kommen kann, und die industrielle Steuerungstechnik üblicherweise mit wesentlich kleineren Systemressourcen auskommen muss als die

³² Diese Forderung bedingt sich aus dem Wegfall einer zentralen Instanz bei vollständig verteilten Systemen.

³³ Diese genannten Anforderungen sind dabei nicht als Ausnahmen oder Fehler des Systems zu betrachten, sondern müssen als normale Betriebszustände verstanden werden.

Informatik, muss so wenig Speicherplatz, Rechenkapazität und Bandbreite des Netzwerkes wie möglich verbraucht werden³⁴.

³⁴ Da der Entwurf der Steuerungsmethode entscheidenden Einfluss auf die Leichtgewichtigkeit der späteren Umsetzung hat, muss dieser Aspekt vor dem Hintergrund einer industriellen Anwendung schon beim Entwurf berücksichtigt werden.

3. Stand der Technik

3.1. Materialflusssteuerung

Der verbreitete Ansatz zur Steuerung beliebiger technischer Anlagen sind zentrale Systeme /Gülich 95, Heeg 97, Konhäuser 98/. Von einer Zentrale ausgehend, meist einer Speicherprogrammierbaren-Steuerung (SPS), werden Daten von Sensoren oder Geräten abgerufen und nach der Bearbeitung wieder an diese zurück übermittelt. Die Sensoren oder Aktoren selbst verfügen über keinerlei eigene Intelligenz in Form von Microchips oder ähnlichen Bausteinen; Sensoren und Aktoren sind somit rein ausführende Elemente, welche auf die zentrale Steuerintelligenz angewiesen sind³⁵. Die Steuerungen von Materialflussanlagen müssen dabei die Funktionen Lagerverwaltung, Materialflusssteuerung, Systemsteuerung und Visualisierung zur Verfügung stellen /VDMA 94, VDI 96/.

Grundsätzlich kann bei MFS zwischen zentralen und dezentralen Steuerungskonzepten unterscheiden werden /Kief 98, Käppner 95/. Zentrale Konzepte zeichnen sich durch eine einzelne, das gesamte Materialflusssystem steuernde Steuerung aus. Dezentrale Konzepte verlagern die Funktionen in dezentrale, im Feld befindliche Steuerungen, wobei der Grad der Dezentralität der einzelnen Konzepte stark schwanken kann. Dabei weist keines der bisher vorgestellten Konzepte³⁶ eine vollständig Verteilung der Materialflusssteuerung auf, üblicherweise werden einzelne Teilstücke des MFS von dezentralen Steuerungen, die mit einer zentralen Steuerung kommunizieren, gesteuert.

³⁵ In Bereichen der Fördertechnik, in denen es gilt, große Strecken zu überwinden, setzen sich mittlerweile auch Konzepte mit dezentralen, an die Motoren angebundenen Steuerungen durch /Bohl 01/. Hierbei wird größtenteils die Schaltung der Antriebe zusammen mit der Steuerungsintelligenz in einem dezentralen Konzept zusammengefasst.

³⁶ Vgl. dazu /Schwarzer 98, Bächter 00, Bosch 00, Beckhoff 01, Bohl 01, Montrac 01, Stein 01/

Bezüglich dieser Einordnung kann nochmals zwischen direkter und indirekter Wegfindung unterschieden werden /Jünemann 98, Dubbel 97/. Direkte Wegfindungskonzepte sehen eine dem Werkstückträger aufgeprägte Weginformation vor, die an den Verzweigungsstellen mittels eines Identifikationssystems ausgelesen wird. Diese Methode ist insofern inflexibel und benötigt bei einer Veränderung des MFS eine Neuprogrammierung dieser Informationen, als dass Abläufe exakt vorgeschrieben werden /Brinkmann 87/. Indirekte Konzepte basieren deshalb auf einer Beschreibung der Zielinformation /Patent 01/. Bei diesen Konzepten wird jedoch auch eine zentrale Steuerung benötigt, da die Zuordnung, wie an einer bestimmten Weiche zu verfahren ist, von einem zentralen Rechner in die einzelnen Teile heruntergeladen wird. Schwarzer /Schwarzer 98/ und Gausemeier /Gausemeier 98/ stellen jeweils ähnlich steuerungstechnisch verteilte Konzepte vor, die auf dem indirekten Konzept beruhen.

Direkte, zentrale Steuerungskonzepte arbeiten nach dem selben Prinzip wie die indirekten Konzepte, nur wird die Information zentral verarbeitet. Indirekte, zentrale Steuerungskonzepte arbeiten nicht mehr mit Zielinformationen, sondern mit Identnummern /Offenlegung 01/. Die übergeordnete Steuerung entscheidet dann über das anzufahrende Ziel /VDMA 94/³⁷. Dies erlaubt bei Störungen oder Materialstauungen an redundanten Systemen eine gewisse Flexibilität.

Alle zur Zeit vorgeschlagenen Konzepte weichen hinsichtlich ihrer Granularität und auch hinsichtlich der verwendeten Steuerungskonzepte stark voneinander ab. Grundsätzlich können sie jedoch darauf reduziert werden, dass die einzelnen Teilsteuerungen aus einer Steuerung, angeschlossenen Sensoren und Aktoren und einem die einzelnen Teilsteuerungen miteinander verbindenden Bussystem bestehen. Dabei übernimmt die Steuerung nur sehr hardwarenahe Funktionen, wie sie in Abbildung 2-2 für

³⁷ Als Beispiel kann hierfür das Gepäck-Transfersystem des Flughafens Köln/Bonn genannt werden /Thomas 01/.

die Ebenen 1 und 2 vorgesehen sind³⁸. Dies ist in der folgenden Abbildung dargestellt:

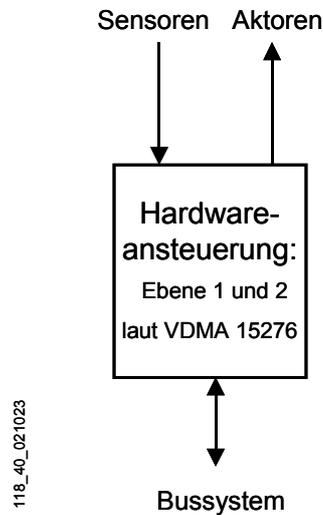


Abbildung 3-1: Stand der Technik in der Steuerungsarchitektur modularer MFS

Somit stellt Abbildung 3-1 den Stand der Technik für die Steuerungsarchitektur von MFS dar und weist dabei auch die derzeit existierenden Begrenzungen aus: Eine Verteilung von übergeordneten Steuerungsfunktionen, speziell der Wegfindung, steht noch aus. Damit können diese Ansätze keinerlei Selbstkonfigurationsfähigkeiten bezüglich des zur Wegfindung verwendeten Modells beinhalten, was die Grundlage für die geforderte Anpassungsfähigkeit ist. Da dies auch in den bisher verwendeten Modellierungsverfahren für MFS begründet ist, sollen diese in einem nächsten Schritt diskutiert werden.

3.2. Modellierungsverfahren

Für die Wegfindung des Materialflusses ist ein Modell des zu steuernden MFS unerlässlich /VDMA 94/, wobei ein Modell nach

³⁸ Im weiteren soll in diesem Zusammenhang deshalb von „Hardwareansteuerung“ die Rede sein.

Minsky /Minsky 65/ als ein Objekt³⁹ verstanden wird, das von einem Betrachter genutzt werden kann, um ihn interessierende Fragen bezüglich des realen Objektes⁴⁰ zu beantworten⁴¹. Für die Modellierung von MFS findet hauptsächlich die Graphentheorie⁴² /Diestel 00, Nägler 96/ Anwendung, wobei der Modellierungsansatz auf das Gesamtsystem bezogen ist. Gudehus /Gudehus 77, Gudehus 77b/, Großeschallau /Großeschallau 84/, Brinkmann /Brinkmann 87/ und Arnold /Arnold 98/ diskutieren umfassend die möglichen Dekompositionen eines MFS, dessen kleinstes Subsystem der „Irreduzible Transportknoten“⁴³ ist, der aus einer Anzahl Eingänge und Ausgänge und einem zusammenführenden Element besteht /Gudehus 77/. Ein „Irreduzible Transportknoten“ nach Gudehus /Gudehus 77/ mit n Eingängen und m Ausgängen - also vom Typ (n,m) - ist in der folgenden Abbildung dargestellt:

³⁹ Für eine umfassende Diskussion der unterschiedlichen Aspekte der technischen Modellbildung vgl. /Panreck 02/.

⁴⁰ Vgl. dazu auch /VDI 96b/.

⁴¹ Bezüglich der Steuerung des Materialflusses ist damit die Modellierung der Topographie entscheidend. Topographie erfasst sowohl ein Gelände als auch die sich darauf befindlichen Objekte, im Gegensatz zur Topologie, die nur die Lage und Lagebeziehungen einzelner Objekte erfasst /Brockhaus 98/. Im Rahmen dieser Arbeit soll die Struktur des Transportsystems als Topographie bezeichnet werden, da ein MFS durch unterschiedliche MFM gekennzeichnet ist. Dies dient ebenfalls der Abgrenzung gegenüber des im Rahmen der Kommunikationstechnik verwendeten Begriffs „Topologie“ /Stehle 01/.

⁴² Bezüglich mathematischer Modelle lassen sich relationale Strukturen und Algebren unterscheiden /Kreisel 72/.

⁴³ Arnold verwendet synonym den Begriff „Komplexknoten“ /Arnold 98/.

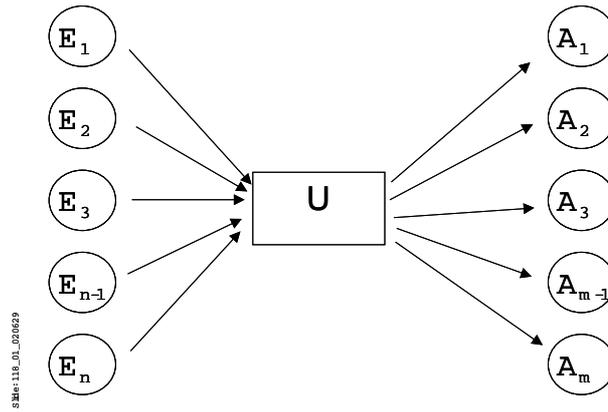


Abbildung 3-2: Irreduzibler Transportknoten

Die Rückbezüglichkeit des Modells auf die Anwendung bewirkt unterschiedliche Schwerpunkte der Modellierung, die einerseits in planungs- und simulationsbezogen und andererseits in ausführungsbezogen eingeteilt⁴⁴ werden können. Die Modellierung von MFS für planungsbezogene Aufgaben⁴⁵ dient hauptsächlich der Kapazitätsauslegung /Großeschallau 84/, wobei dies auf die Netzwerkflussplanung bezüglich der einzelnen Teile des MFS zurückgeführt werden kann /Arlt 93/. Gudehus /Gudehus 77, Gudehus 77b, Gudehus 99/ schlägt dafür die Anwendung einer Input / Output bezogenen Modellierung vor. Den einzelnen Ein- und Ausgängen des „Irreduziblen Transportknotens“ werden entsprechende Partialströme zugeordnet. Großeschallau /Großeschallau 84/ und Arnold /Arnold 98/ erweitern diese Modellierung um entsprechende Warteschlangenmodelle. Gueorguiev und Ziems schlagen zur Leistungsanalyse eine gemeinsame Modellierung von Material- und Informationsfluss und Ressourcen in einem Petri-Netz-Modell vor /Gueorguiev 99/. Darüber hinaus werden noch die Materialflussmatrizen als Modellierungsformen genannt /VDI 75, Arnold 98/. Alle darüber hinaus für Touren-

⁴⁴ Die VDI-Richtlinie 3633 - Teil 1 diskutiert umfassend die möglichen Nutzungsmöglichkeiten der Simulation im Verlauf des Lebenszykluses /VDI 00/.

⁴⁵ Für die durchgängige Informationsverarbeitung im Planungsprozess schlägt Dobrowolny ein semantisches Modell vor /Dobrowolny 99/.

und Transportplanung verwendeten Modellierungsverfahren basieren auf Graphen /Domschke 95, Domschke 97/.

Die ausführungsbezogene Modellierung bildet den kleinsten Teil der Modellierungsansätze für MFS. Gudehus und Brinkmann schlagen beide vor, die In / Output orientierte Modellierung in einen Graphen zu überführen /Gudehus 99, Brinkmann 87/. Die Verbindungsstruktur der Subsysteme einer Dekomposition des MFS wird in einem solchen Graphen als nicht hierarchische „Topologie“ des Systems verstanden, wobei die Subsysteme als Knoten und deren Transportbeziehungen als Kanten modelliert werden /Panreck 02/. Anhand dieses Graphen erfolgt die Wegfindung für jeden Transportauftrag.

Gueorguiev und Ziems /Gueorguiev 99/ gestalten als einzige für jeden irreduziblen Transportknoten ein eigenständiges Petri-Netz-Modell, dessen Eingangs- und Ausgangsschnittstellen als Transition-Stelle-Transition (T-S-T)-Kette entwickelt sind. Die allen anderen Modellierungsansätzen gemeinsame Vorgehensweise der Modellierung des Gesamtsystems durch Dekomposition kann nur für eine Betrachtung des Gesamtsystems zum Ziel führen, eine Modellierung einzelner, alleinstehender Subsysteme - wie für diese Arbeit notwendig - ist mit diesen Ansätzen nicht möglich.

3.3. Verteilte Steuerungen

Wie in Kapitel 2.1 ausgeführt wurde, geht der Trend im Bereich der Informatik seit einiger Zeit zu verteilten Systemen, wobei die dort zum Einsatz kommenden Technologien auch für die Automatisierungswelt immer relevanter werden /Freese 97, Thrun 97, Kabitzsch 99, Wurmus 99, Harnischer 01, Tauchnitz 01, Wenzel 01, ARC 01b, Klostermeyer 01, Kühnle 01/. Verteilte Systeme, und damit verteilte Steuerungen, können auf die verschiedensten Arten und Weisen kategorisiert werden. Jensen /Jensen 02/ weist auf die zugrunde liegenden Programmiermodelle als Unterscheidungskriterium hin. Farley /Farley 98/ hingegen schlägt hinsichtlich der Kommunikation vier Aspekte vor. Eine grundsätzlicher Unterscheidung lässt sich in synchrone und asynchrone verteilte Systeme vornehmen⁴⁶ /Coulouris 01/, wobei synchrone verteilte Systeme nur unter erheblichem Aufwand herstellbar sind und die Forderungen nach solchen im industriellen Einsatz derzeit nur in sehr bestimmten Teilbereichen, wie zum Beispiel bei der *Drive-Synchronization* zu finden sind /Laux 01/. Eine dafür notwendige Uhren-Synchronisation lässt sich durch den Bezug auf eine externe Zeitquelle oder aber durch verschiedene Algorithmen und Protokolle⁴⁷ realisieren /Tanenbaum 95/. Alle restlichen verteilten Systeme können als asynchron verstanden werden /Babaoglu 93/. Grundsätzlich kann festgestellt werden, dass bei verteilten Systemen die Programmier- und Kommunikationsweise eng miteinander verbunden sind.

⁴⁶ Bei synchronen Systemen geht man davon aus, dass sowohl die Ausführungszeit eines Programms als auch die Laufzeit einer Message in bestimmten Grenzen voraussagbar ist. Außerdem werden lokale, genaue Uhren vorausgesetzt. Asynchrone Systeme gehen davon aus, dass keine der drei Forderungen erfüllt wird.

⁴⁷ Hier sei auf den *Draft Standard for a Precision Clock Synchronisation Protocol for Network Measurement and Control Systems* (P1588) der IEEE verwiesen /P1588 02/.

Das Konzept der Verteilung von Steuerungsfunktionen hat direkte Auswirkungen auf die Programmierweise. Eingang in die Normung⁴⁸ für Programmiersysteme von Steuerungen haben verteilte Systeme im Rahmen der IEC-PAS 61499-1 gefunden⁴⁹ /IEC 00/. Dieser Normungsvorschlag enthält ein auf Events basierendes Programmiermodell /IEC 00/. Java, die Programmiersprache des Internets /Boger 99/, findet ebenfalls vor diesem Hintergrund bisher ungewohnten Einsatz im Bereich der Steuerungstechnik /Freitag 01, Lorentz 01, Grundmann 02/. Dies wird ermöglicht durch zwei Spezifikationen, die Java um Echtzeitfähigkeiten erweitern /RTDA 01, RTS 02/.

Im Rahmen der derzeitigen Diskussion kristallisiert sich die sogenannte *Middleware*⁵⁰ als Ort der Implementierung der wichtigsten Kommunikationsfunktionen⁵¹ zwischen den einzelnen Teilen des verteilten Systems heraus /Coulouris 01/. Dies schlägt sich ebenfalls bei den Feldbussen und den damit verbundenen Protokollspezifikationen nieder, welche für die Übertragung der Steuerungsinformationen zu den Sensoren und Aktoren zuständig sind /Dietrich 98/. Zu nahezu allen Systemen gibt es mittlerweile Derivate, die die Vorteile verteilter Systeme mehr oder minder gut umzusetzen in der Lage sind /Kriesel 98/. Im Unterschied zu den Feldbussen arbeiten die dezentralen Steuerungseinheiten der LON-Technologie ereignisgesteuert und befinden sich damit in unmittelbarer Nähe zu der weiter oben vorgestellten IEC-PAS 61499-1. Die Grundlage von LON kann als beispielhaft für verteilte Steuerungstechnik

⁴⁸ Vgl. Enste und Epple /Enste 99/ zu ersten Umsetzungen in Klassenstrukturen für den Entwurf von Prozeßsteuerungen als speziellem Typ der Objekt-Orientierung.

⁴⁹ Darüber hinaus wurde auch die Internationale Norm für die Programmierung von SPSen um einen Teil für Funktionsblöcke erweitert /EN 01/.

⁵⁰ Früher wurde in diesem Zusammenhang auch von einem *Netzwerk-Betriebssystem* gesprochen, vgl. /Tanenbaum 95/. Heute findet teilweise der Begriff *Internet Operating System* Anwendung /Razafimahefa 00/.

⁵¹ Dazu gehört primär die Implementierung des Protokolls, aber auch Funktionen wie Namensdienste etc.

angesehen werden, jedoch erfüllt es die Anforderung des Determinismus⁵² nicht und hat deshalb nur sehr begrenzte Nutzung in diesem Feld erfahren /LON 98/. Die drei zur Zeit in unterschiedlicher Anwendungsreife befindlichen Protokollspezifikationen für verteilte Automatisierung *Interface for Distributed Automation* (IDA) /IDA 01, IDA 01a/, *EtherNet/IP* /EtherNet/IP 01/ und *ProfiNet* /Wenzel 01/ müssen als *Middleware* verstanden werden und können nicht mehr nur den Bussystemen oder den Programmiersystemen zugeordnet werden⁵³.

Abschließend kann festgehalten werden, dass der Trend zu verteilten Systemen innerhalb der industriellen Steuerungstechnik erst an seinem Anfang steht, die entsprechenden Normen und Produkte stecken noch in ihren Kinderschuhen. Für die Entwicklung verteilter Applikationen sind dabei beide Aspekte, die Programmierung und die Kommunikation, zu berücksichtigen.

⁵² Unter Determinismus versteht man im Rahmen der Steuerungstechnik ein vorhersagbares und reproduzierbares Systemverhalten /Messerschmidt 02/. Aufgrund der bei LON zum Einsatz kommenden Bustechnologie kann dieses Verhalten nicht garantiert werden. In diesem Zusammenhang sei auf Tsichritzis hingewiesen, der die Forderung nach Determinismus in parallelen Systemen vor dem Hintergrund Biologischer Systeme generell in Zweifel stellt /Tsichritzis 99/.

⁵³ Ihnen allen ist gemeinsam, dass Sie auf Ethernet TCP/IP als Kommunikationsmedium der Zukunft setzen /Lorentz 02/.

3.4. Adaptive Routing-Algorithmen

Abschließend für den Stand der Technik sollen die für die Wegfindung nutzbaren Algorithmen betrachtet werden. Dabei soll der Schwerpunkt auf adaptiven und verteilten Algorithmen liegen. Hier ist eine Betrachtung von Applikationen jenseits der Steuerung von MFS unausweichlich, da die weitreichendsten Anstrengungen in den Anwendungsfeldern Routing⁵⁴ von Fahrzeugen /Ball 95, Jörns 97/ und Routing von Informationen /Lang 97/ erfolgten. Diese Algorithmen zerfallen laut Lang /Lang 97/ grundsätzlich in zwei Kategorien, einerseits in nicht adaptive und andererseits in adaptive Routing-Algorithmen. Hinsichtlich der adaptiven Routing-Algorithmen differenziert Lang wiederum zwischen zentralen, isolierten und verteilten, wobei sich die verteilten Algorithmen durch Kooperation, also den Datenaustausch untereinander, auszeichnen. Eine umfassende Diskussion der Unterschiede zwischen verteilten und zentralen Algorithmen findet sich bei /Tel 00/. Unabhängig von dieser Klassifizierung lassen sich Routing Algorithmen noch in Shortest-Path-Algorithmen oder in Optimal-Routing-Algorithmen⁵⁵ einteilen /Schwarz 87/. Da die adaptiven Algorithmen auf den Shortest-Path Algorithmen aufbauen, sollen hier im folgenden die wichtigsten Shortest-Path-Algorithmen⁵⁶ diskutiert werden. Im Anschluss daran werden die verschiedenen bekannten Ansätze adaptiver Algorithmen vorgestellt.

Der sicherlich bekannteste Shortest-Path Routing Algorithmus ist der *Dijkstra-Algorithmus* /Dijkstra 59/. Bei diesem

⁵⁴ Der Begriff des Routing wird hier synonym zu dem in der Logistik üblichen Begriff der Wegfindung verwendet.

⁵⁵ Diese Algorithmen finden üblicherweise Anwendung innerhalb der Sprachübertragung. Entscheidendes Merkmal ist dabei die verbindungsorientierte Vermittlung im Gegensatz zur paketorientierten Vermittlung in anderen Telekommunikationsnetzen wie dem Internet /Frikha 99/.

⁵⁶ Auf eine Vorstellung der Optimal-Routing-Algorithmen wird verzichtet, da diese für den Gang der Arbeit unerheblich sind, weil für MFS keine vorab bekannten Verkehrsflüsse bestimmbar sind, beziehungsweise das Transportvolumen für eine statistische Auswertung nicht hoch genug ist.

Algorithmus wird für einen bekannten Topologiegraphen der kürzeste Weg für ein Quelle-Ziel-Paar errechnet. Der *Floyd-Warshall-Algorithmus* liefert im Gegensatz dazu als Ergebnis die Entfernung zwischen allen Quelle-Ziel-Paaren /Jungnickel 99/. Ebenfalls ein Vertreter der Shortest-Path-Algorithmen ist der *Bellmann-Ford-Algorithmus* /Bertsekas 87/. Dabei wird die Anzahl der Kanten zwischen den Knoten als weiteres Minimierungskriterium eingefügt⁵⁷. Grundlage für diese Algorithmen ist jeweils ein vorliegender, gewichteter Graph⁵⁸, der das Gesamtsystem beschreibt.

Die weitestgehende Umsetzung des Gedankens der adaptiven Steuerung findet sich im Internet. Das Routing der Datenpakete wird üblicherweise mittels dynamischer Tabellen organisiert. Diese Tabellen passen sich automatisch neuen Gegebenheiten⁵⁹ an. Der *Backward-learning Algorithmus*, das *Delta-routing*, *Distance-Vektor-Protokolle* und das *Link-State-Protokoll*⁶⁰ stellen Vertreter adaptiver Verkehrssteuerungsmethodiken in der Kommunikationstechnik dar. Die letzten beiden genannten Protokolle sind dabei als verteilt kooperierende Algorithmen konzipiert /Lang 97, Tel 00/.

Derzeit in Internet-Routern eingesetzte Algorithmen sind in das *Distance-Vektor-Protokoll* und in das *Link-State-Protokoll* /Halabi 98/ eingebettete Algorithmen, wobei das *Distance-Vektor-Protokoll* eine verteilte Adaption des *Bellman-Ford-Algorithmus* ist /Tanenbaum 00, Bruns 01/. Ursprünglich wurde

⁵⁷ Intention dieses weiteren Optimierungskriteriums ist der Versuch, die Anzahl der Sprünge von einem Internet-Router zu dem nächsten (Hops) innerhalb eines Weges zu minimieren /Bertsekas 87/.

⁵⁸ Ein Graph ist ein Paar $G = (V, E)$ disjunkter Mengen, wobei die Elemente von V üblicherweise Ecken und die Elemente von E ($E \subseteq V \times V$) üblicherweise Kanten genannt werden /Diestel 00, Steger 01/.

⁵⁹ Optimierungskriterien können beispielsweise die Entfernung, die Anzahl der Teilstrecken oder aber die Übertragungszeit sein.

⁶⁰ Im weiteren Verlauf soll dem allgemeinen Sprachgebrauch folgend von „Protokollen“ die Rede sein.

dieses Protokoll zuerst im ARPANET⁶¹ eingesetzt, mittlerweile ist es jedoch größtenteils durch das *Link-State-Protokoll* ersetzt worden /McQuillan 80/. Auch bei diesem Verfahren wird vorausgesetzt, dass jeder Knoten über vollständige Informationen hinsichtlich der Topologie verfügt. Adaptivität existiert in diesem Protokoll nur hinsichtlich des kürzesten Weges. Ebenso wie das *Distance-Vektor-Protokoll* baut das *Link-State-Protokoll* auf einem zentralen Ansatz auf, in diesem Fall dem Algorithmus von Dijkstra /Tanenbaum 00/. Durch Austausch mit anderen Routern erweitert jeder Router seine Tabellen dynamisch und kann sich somit an Änderungen im Netzwerk anpassen.

Beide im Internet hauptsächlich verwendeten Algorithmen bauen ihre Modelle des Systems, die zum Routing der Datenpakete genutzt werden, dynamisch auf. Erfolgt von einem Knoten für eine bestimmte Zeit keine Sendung, wird er aus dem Modell entfernt, was eine gewisse Selbstwartung erlaubt. Der für die Steuerung von MFS zu entwickelnde Algorithmus kann sich in diesem Aspekt an beide Algorithmen anlehnen, muss diese jedoch darüber hinaus um die Funktion der Sicherstellung der Existenz der Nachbarn erweitern.

⁶¹ ARPANET steht für *Advanced Research Projects Agency Network*, ein von dem amerikanischen Verteidigungsministerium unterstütztes Computernetzwerk, das die Grundlage des heutigen Internets legte.

3.5. Zusammenfassende Betrachtung des Standes der Technik

Sowohl innerhalb der allgemeinen Steuerungstechnik als auch speziell für MFS ist grundsätzlich der Trend zu verteilten Steuerungssystemen zu identifizieren. Dabei basieren alle vorgeschlagenen Ansätze zur Steuerung von MFS noch auf einem zentralistisch geprägten Konzept, das weder komplett modularisiert und verteilt ist, noch eine Selbstkonfiguration erlaubt. Die Robustheit dieser Ansätze ist ebenfalls durch ihre Abhängigkeit von einer zentralen Steuerung begrenzt. Dies wird durch die Implementierungsvorschläge der entsprechenden technischen Richtlinien unterlegt. Die Ursache für diese noch stark zentralistisch geprägten und wenig anpassungsfähigen Steuerungskonzepte findet sich begründet in der zugrundeliegenden Technologie: Die allgemeine Steuerungstechnik bewegt sich erst seit einigen Jahren in Richtung verteilter Steuerungssysteme. Deshalb sind die Gesamtkonzepte, die Kommunikations- und Programmierweise vereinen, erst im Entstehen, wobei auf die entwickelten, grundlegenden Mechanismen⁶² im Rahmen der Arbeit zurückgegriffen werden kann. Darüber hinaus ist jedoch eine Steuerungsmethode zur verteilten Wegfindung, die derzeit nicht vorliegt und im Rahmen dieser Arbeit entwickelt wird, nötig. Diese Steuerungsmethode kann in Ansätzen an die Routing-Protokolle des Internets angelehnt werden⁶³, jedoch sind für den Materialfluss spezifische Aspekte⁶⁴ zu berücksichtigen. Daraus folgt, dass ein spezieller Algorithmus für die Wegfindung zu entwickeln ist, der die speziellen Eigenschaften von MFS berücksichtigt. Für einen

⁶² Synchronisationsmechanismen seien hier genannt.

⁶³ Betrachtet man die Routing-Algorithmen, so kann man feststellen, dass die beiden derzeit in Routern eingesetzten Algorithmen, das *Distance-Vektor*- und das *Link-State-Protokoll*, für ein verteiltes Routing geeignet sind. Die Algorithmen sind adaptiv, eingeschränkt selbstwartend und besitzen Optimierungsfunktionen.

⁶⁴ Für Routing-Algorithmen der Kommunikationstechnik sind Transportgut und Nachricht identisch, dies gilt nicht für MFS, der physische Transport der Nachricht und des Transportgutes fallen auseinander. Weiterhin sind innerhalb der Kommunikationstechnik Transportbeziehungen aller mit allen üblich, was auch nicht für MFS zutrifft.

derartigen, neuen Algorithmus muss auch ein neues Modellierungsverfahren erstellt werden, das es erlaubt, eigenständige Module des MFS zu modellieren.

Für die Umsetzung einer verteilten Steuerung ist darüber hinaus ein Kommunikationsprotokoll erforderlich, das die Kommunikation aller Module spezifiziert. Da dieses Protokoll spezifisch für die Funktion „Wegfindung“ zu erstellen ist, kann es sich nur an den bereits für verteilte Systeme bestehenden Mechanismen orientieren, bzw. teilweise auf diese aufbauen.

.

4. Zielsetzung und Vorgehensweise

Die konsequente Nutzung der sich aus dem Trend zu verteilten Steuerungssystemen ergebenden Vorteile, wie Anpassungsfähigkeit, Modularität und Komplexitätsreduzierung⁶⁵, kann innerhalb der MFS nur durch eine neue, verteilte Steuerungsmethode erfolgen, die durch ihre Verteiltheit und Adaptivität eine maximal mögliche Anpassungsfähigkeit an Änderungen im Aufbau des MFS, also der Topographie, erlaubt. Dies wird durch vollständige Auflösung eines MFS in autarke und dezentral gesteuerte Module ermöglicht. Dafür verfügen die einzelnen Module des Systems bei Systemstart über ein Topographie-Abbild, das die direkt angrenzenden Module enthält. Darauf aufbauend wird durch einen Datenaustausch mit diesen Nachbarn iterativ eine Konfiguration des Systems vollzogen, die damit endet, dass in allen Modulen ausreichende Daten zur Erstellung eines Modells des MFS - und damit zur Wegfindung - vorliegen. Die Verteilung des Abbildes der Topographie des MFS und die Verteilung der Wegfindung im MFS, die in traditionellen Steuerungsansätzen immer zentral erfolgt, sind dabei die entscheidenden Punkte. Ausgehend von einer Definition der Module als atomare Ausprägung einer Transportstrecke und einer Beschreibung der grundsätzlichen Softwarekomponenten zusammen mit der zugrunde liegenden Software-Architektur⁶⁶ innerhalb dieser Module werden in Kapitel 5 eine neue Form der Modellierung des Systems, ein entsprechend verteilter Algorithmus und ein Kommunikationsprotokoll angegeben. Dabei sind die in einem verteilten System notwendigen Funktionen in diesem Algorithmus zu berücksichtigen und der Algorithmus ist in das Protokoll einzubetten. Die bisher bekannten und

⁶⁵ Laut Ashby muss ein steuerndes System mindestens ebenso komplex wie das zu steuernde System sein /Ashby 56/. Durch die Verteilung von Steuerungsfunktionen auf mehrere Steuerungen kann die Komplexität innerhalb der einzelnen Module gesenkt werden.

⁶⁶ Laut Balzert beschreibt eine Software-Architektur die Struktur des Software-Systems durch Systemkomponenten und ihre Beziehung untereinander und eine hierarchische Anordnung derselben /Balzert 00/.

verwendeten Modellierungen von MFS sind dabei zu erweitern, da sie dem Aspekt der Verteilung nicht gerecht werden können.

Eine detaillierte Konkretisierung der in Kapitel 5 beschriebenen Steuerungsmethode erfolgt in Kapitel 6 entlang der definierten Softwarekomponenten. Eingangs werden die Softwarekomponenten in einzelne, zu erfüllende Funktionen unterteilt und darauf aufbauend die Datenhaltung in jedem Modul dargestellt. Diese beiden Konkretisierungen bilden die Grundlage für eine detaillierte Beschreibung der einzelnen Funktionen, deren Abhängigkeiten untereinander und der verschickten Messages. Da verteilte Systeme sich nur in dem Zusammenspiel mehrerer Module darstellen lassen, sollen fünf alle Systemzustände abdeckende Szenarien vorgestellt werden. Abschließend soll die Steuerungsmethode an einem Referenzobjekt umgesetzt werden. Dies geschieht anhand einer Simulation der grundsätzlichen Funktionen und einer kompletten Implementierung in einem MFS. Dieses Vorgehen ist in Abbildung 4-1 dargestellt.

Methodisch lehnt sich diese Arbeit an die in der Softwareentwicklung gängige Vorgehensweise an⁶⁷. Zur Beschreibung von Daten dient die XML⁶⁸, um eine schnittstellenfreie Umsetzung der Arbeit in die Praxis zu gewährleisten. Die Sequenzdiagramme werden mittels der UML⁶⁹ dargestellt, für die Darstellung von Algorithmen wurde auf einen Pseudocode zurückgegriffen, der sich an der objektorientierten Programmiersprache Java /Weinert 01, Krüger 00, Jobst 99/ orientiert.

⁶⁷ Für eine Übersicht und Diskussion der verschiedenen Ansätze vgl. /Gumm 00/.

⁶⁸ Die *Extensible Markup Language (XML)* /Eckstein 00/ ist eine sogenannte Metasprache, das heißt mit ihrer Hilfe lassen sich neue Sprachen /Konventionen definieren. Eine Weiterentwicklung der XML stellen die sogenannten *Schemata* dar /W3C 00/.

⁶⁹ Für eine umfassende Einführung in die *Unified Modelling Language (UML)* vgl. /Oesterreich 99, Balzert 01/; bezüglich der spezifischen Nutzung für Steuerungsanwendungen vgl. /Douglass 99, Douglass 99b/.

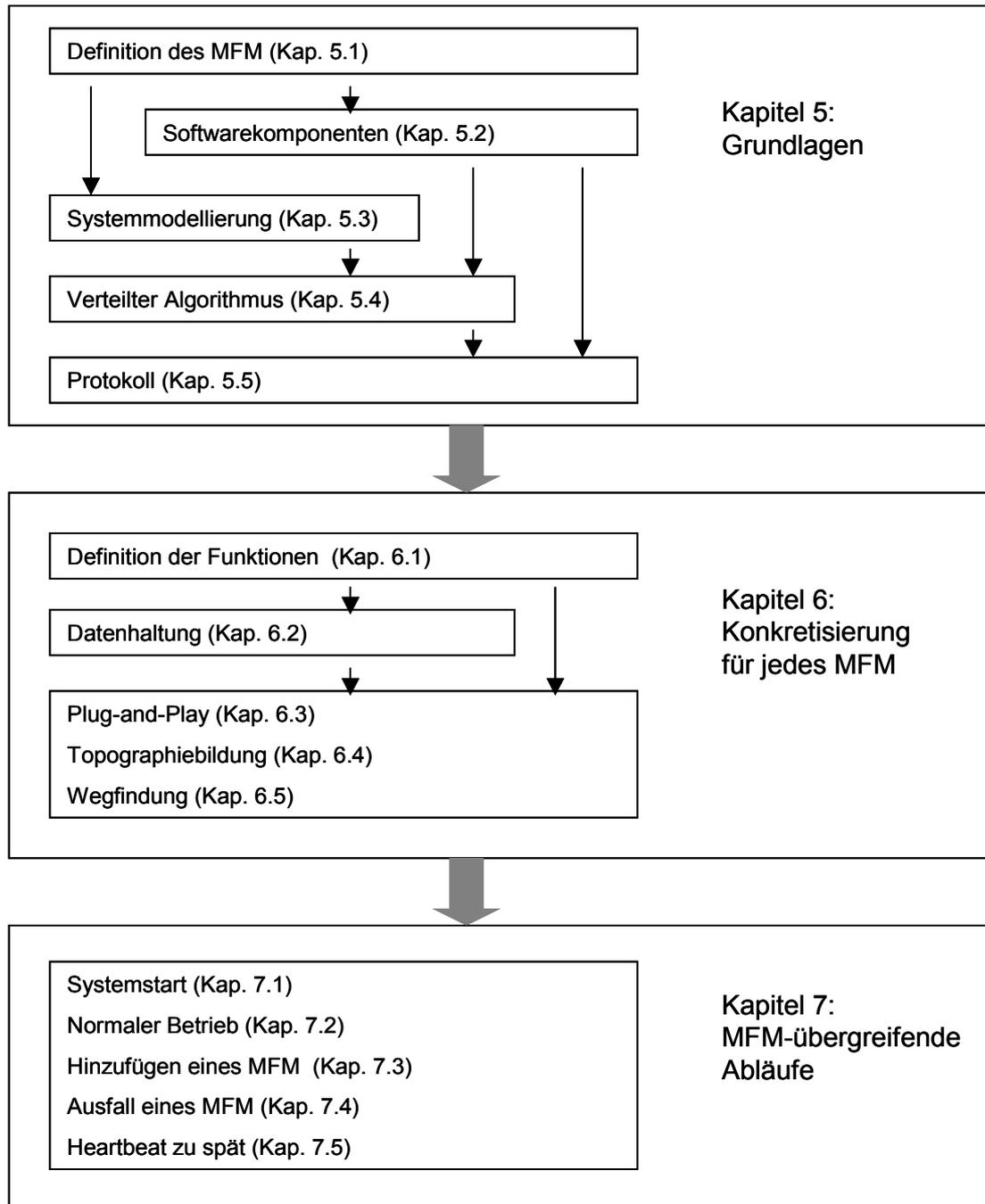


Abbildung 4-1: Aufbau und Vorgehensweise der Arbeit

5. Grundlagen

Wie in Kapitel zwei dargelegt wurde, ist zur Erhöhung der Anpassungsfähigkeit von MFS eine Methode zur verteilten Steuerung des Materialflusses notwendig. Diese Steuerungsmethode kann sich in Teilen an den Stand der Technik anlehnen, muss dabei jedoch die unterschiedlichen Aspekte von verteilten Systemen und Materialflusssystemen berücksichtigen. Im nachfolgenden Kapitel sollen deshalb die Grundlagen für die verteilte Steuerungsmethode beschrieben werden. Dafür erfolgt eingangs eine Definition der Module eines MFS in Form von Prämissen, die sowohl die technischen Grundlagen der Module als auch die zu entwickelnden Steuerungsfunktionen betreffen. Darauf aufbauend und an die geforderten Funktionen des MFS angelehnt, erfolgt die Beschreibung der innerhalb der Module benötigten Steuerungsfunktionen. Diese Funktionen werden durch ein Modell für modulare MFS zusammen mit einem Algorithmus zur verteilten Wegfindung weiter unterlegt. Abschließend erfolgt die Spezifikation des zwischen den einzelnen Modulen zur Anwendung kommenden Kommunikationsprotokolls.

5.1. Materialflussmodule

In Kapitel 2.3 wurde bereits ausgeführt, dass es sich bei MFS um offene Systeme handelt. Die nachfolgende Abbildung stellt ein modulares MFS und dessen Schnittstellen zur Systemumgebung dar:

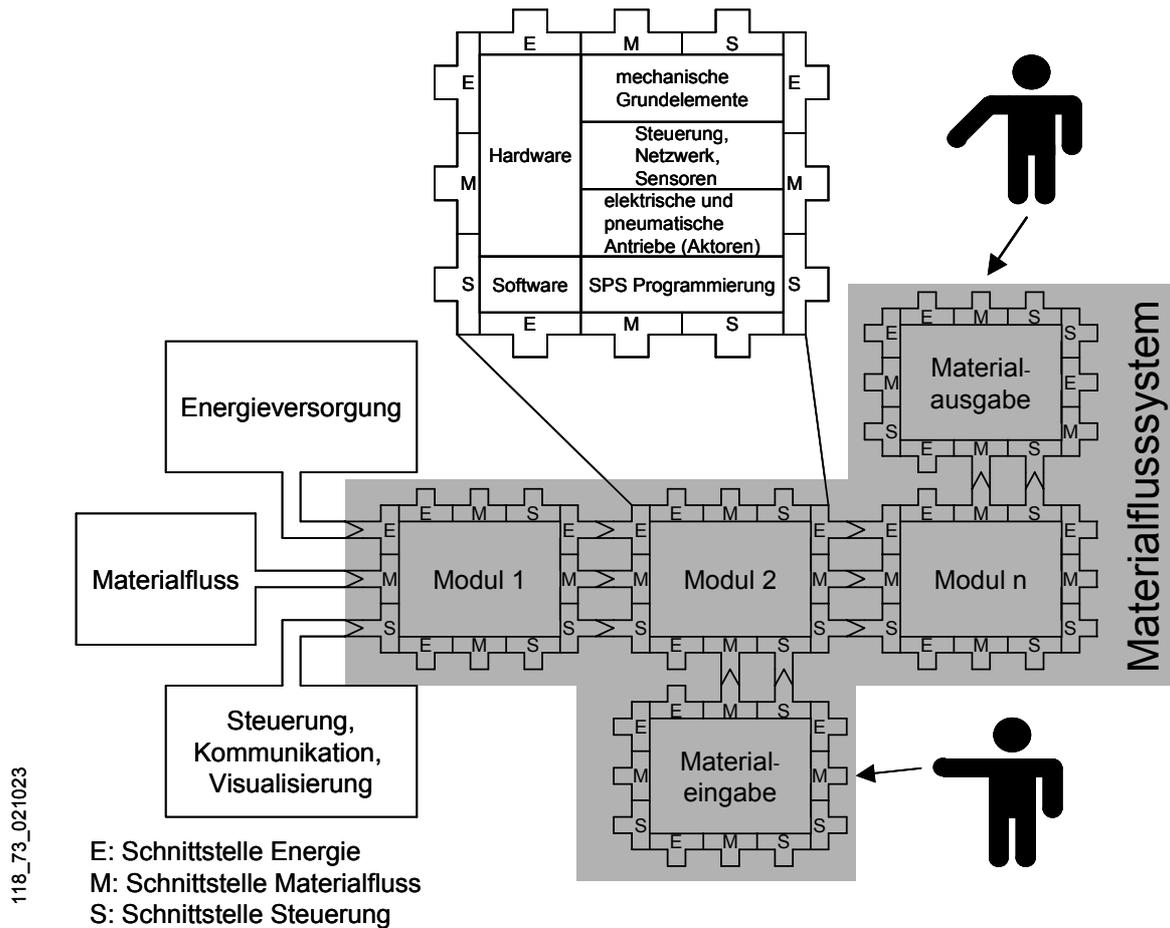


Abbildung 5-1: Ein modulares Materialflusssystem /Lorentz 02c/

Jedes MFS besitzt die Schnittstellen Energie-, Material- und Steuerungsinformationsfluss - sowohl nach außen als auch innerhalb des MFS, zwischen seinen einzelnen Modulen. Jedes dieser Module besteht aus mechanischen Elementen der Fördertechnik und Steuerungen mit darin enthaltenen Steuerungsprogrammen und den entsprechenden Aktoren und Sensoren. Im Rahmen der Darstellung besitzt jedes Modul jeweils vier Schnittstellen bezüglich Energie, Materialfluss und Steuerung zu anderen Modulen, bzw. nach außen; dies kann jedoch auch variieren /Lorentz 02c/. Aufbauend auf dieser allgemeinen Darstellung von Modulen in Materialflusssystemen gilt es, diese Module genauer bezüglich ihrer Struktur und Interaktion zu hinterlegen. Die Prämissen der Materialflussmodule (MFM) des in dieser Arbeit entwickelten vollständig verteilten MFS sind:

1. Ein MFS besteht nur aus einzelnen MFM. Das MFS erfüllt seine Aufgabe durch das Zusammenspiel der einzelnen MFM, es gibt kein einzelnes, das Gesamtsystem steuerndes MFM.
2. Jedes MFM entspricht genau einer Förderstrecke, wobei unerheblich ist, ob es sich bei der Förderstrecke um eine einfache Strecke, eine Weiche, eine Zusammenführung oder ähnliches handelt⁷⁰.
3. Jedes MFM ist hinsichtlich seiner eigenen Hardware statisch. Dies betrifft sowohl die Fördertechnik als auch die Steuerung und Sensoren / Aktoren⁷¹.
4. Jedes MFM hat genau eine, von den anderen Steuerungen unabhängige Steuerung⁷², wobei deren Steuerungsprogramme jeweils der entsprechenden Fördertechnik innerhalb des MFM direkt zugeordnet sind. Die steuernden Programme sind statisch und fertig implementiert, Adaptivität wird ausschließlich durch Parametrisierung erzeugt.
5. Jedes MFM ist in der Lage, mit anderen MFM zu kommunizieren⁷³. Es gibt in dem System kein bevorzugtes Modul, das den Informationsfluss mit den anderen MFM regelt. Bezüglich der Steuerungen der einzelnen MFM erfolgt eine globale Uhren-Synchronisation.

⁷⁰ Vgl. dazu Gudehus' Definition des irreduziblen, also nicht weiter zerlegbaren „Transportknotens“ /Gudehus 77/.

⁷¹ Z.B. ist eine Weiche bereits als Weiche in Hardware aufgebaut. Adaptivität bezieht sich rein auf das Zusammenspiel der MFM.

⁷² Im weiteren soll davon ausgegangen werden, dass es sich bei den beschriebenen verteilten Systemen tatsächlich um von einander unabhängige (in Hardware ausgeprägte) Steuerungen handelt und nicht, wie auch von Tel /Tel 00/ zugelassen, um von einander unabhängige Softwareprozesse. Weiterhin ist die Art der Steuerung im Nachfolgenden nicht relevant, es wird aber davon ausgegangen, dass die Steuerungen frei programmierbar sind.

⁷³ Die Frage der Bustechnologie soll im Rahmen dieser Arbeit nicht behandelt werden, es wird nur davon ausgegangen, dass der zum Einsatz kommende Bus

6. Jedes MFM wird bei der Inbetriebnahme mit grundlegenden Daten ausschließlich in Bezug auf seine Vorgänger und Nachfolger MFM initialisiert⁷⁴.

7. Die Zielinformationen der einzelnen Stückgüter sind direkt eindeutig diesen zu zuordnen und für jedes einzelne MFM und seine Steuerung zugänglich⁷⁵.

Anhand dieser Prämissen ist ein vollständig modulares und vollständig verteilt gesteuertes MFS in seinen Grundzügen festgelegt. Die Bestimmung der Funktionen der Steuerungen der MFM soll in den nächsten Kapiteln erfolgen.

5.2. Softwarekomponenten

Eingangs wurde das MFS als Gesamtheit dargestellt und die MFM definiert. Mit dieser Definition und den in Kapitel 2.4 festgestellten Anforderungen gilt es, die für alle MFM gleiche Steuerung bezüglich der durch sie zu erfüllenden Funktionen festzulegen.

Wie bereits im vorigen Kapitel festgehalten wurde, soll es kein das Gesamtsystem steuerndes MFM geben. Damit muss in jedem einzelnen MFM die Hauptaufgabe eines MFS, die Wegfindung, implementiert werden. Für die Wegfindung muss das MFM notwendigerweise über Informationen bezüglich der Topographie

hinreichend schnell ist (vgl. /Kriesel 98/) und eine Master-Master Kommunikation zulässt.

⁷⁴ Um die angestrebte Anpassungsfähigkeit oder Adaptivität weitestgehend zu ermöglichen, gilt es, die Menge der anfänglich für eine Initialisierung benötigten Informationen möglichst gering zu halten.

⁷⁵ Die Art der Bereitstellung der mit dem Stückgut verbundenen Zielinformationen soll im Rahmen dieser Arbeit nicht diskutiert werden. Vorstellbar ist die direkte Aufprägung dieser Informationen auf das Stückgut mittels Transpondern, die Weitergabe der Zielinformationen jeweils von einer beteiligten Station zur nächsten (Agenten-Architektur, vgl. dazu /Lüth 98/) oder auch ein (zentraler) Zielinformationsserver, der auf Anfrage und Identifizierung des Stückgutes dessen Zielinformationen bekannt gibt (Client-Server-Architektur, vgl. dazu /Flanagan 99/).

des MFS verfügen. Da das Ziel der Arbeit die sich aus der Anpassungsfähigkeit ergebende Adaptivität an Änderungen dieser Topographie ist, kann diese Topographie nicht von vornherein in dem MFM vorliegen, sie muss zur Laufzeit erstellt werden. Um der in Kapitel 2.4 definierten Forderung nach Selbstkonfiguration genüge zu tun, muss also ebenfalls in jedem MFM die Aufgabe der eigenständigen Topographiebildung gelöst sein. Dies ermöglicht es, Änderungen an dem MFS ohne manuelle Konfigurationen in einzelnen MFM vorzunehmen.

Da es sich bei den zu entwickelnden MFM um gleichberechtigte, verteilte Module handelt, muss die Interaktion der MFM untereinander als spontane Vernetzung gelöst werden: Die MFM müssen in der Lage sein, sich im MFS bei den richtigen anderen MFM anzumelden und entsprechend die Anmeldungen anderer MFM entgegenzunehmen. Dies beinhaltet ebenfalls weiteren Kommunikationsbedarf und die verteilte Überwachung des MFS durch jedes MFM, da es keine zentrale Überwachung mehr gibt. Diese spontane Vernetzung muss dabei ohne großen Parametrierungsaufwand erfolgen und wird als Plug-and-Play bezeichnet.

Die drei Aspekte Wegfindung, Topographiebildung und Plug-and-Play sollen im weiteren Verlauf der Arbeit als Softwarekomponenten bezeichnet werden, die in jedem MFM implementiert werden. Die nachfolgende Darstellung zeigt die in Kapitel 3.1 eingeführte Abbildung 3-1 um diese drei Softwarekomponenten erweitert:

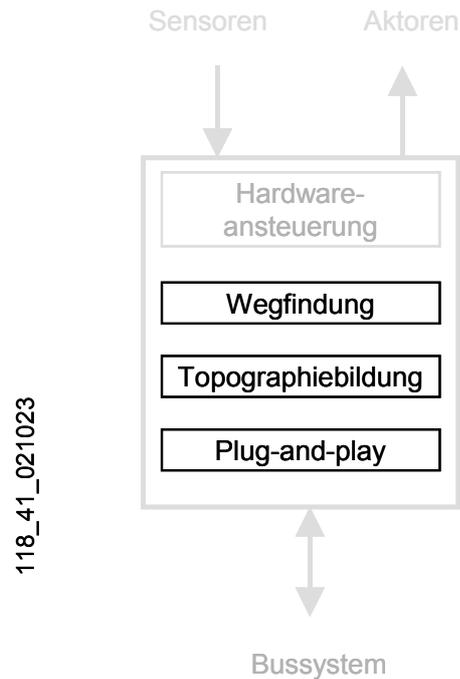


Abbildung 5-2: Erweiterte Steuerungsarchitektur innerhalb eines MFM

Diese vier Softwarekomponenten können jeweils voneinander getrennt behandelt werden, beeinflussen sich jedoch gegenseitig. Nur in ihrer Gesamtheit ermöglichen die drei neu vorgeschlagenen Softwarekomponenten das gesetzte Ziel einer verteilten und adaptiven Steuerung.

Die Softwarekomponente Hardwareansteuerung, die nicht Teil der Arbeit ist, übernimmt dabei die gesamte Aufgabe der Ansteuerung der MFM-Hardware⁷⁶. Die Schnittstelle zwischen dieser und den neu eingeführten Softwarekomponenten liegt dabei einzig in einem Aufruf der Softwarekomponente Wegfindung durch die Komponente Hardwareansteuerung. Dabei wird davon ausgegangen, dass die Hardwareansteuerung durch die mit ihr verbundenen Sensoren von einem neuen Transportgut eine neue Zielinformation

⁷⁶ Darüber hinaus sollen in dem Modul Hardwareansteuerung auch alle herkömmlichen Funktionen wie Verriegelung etc., die nicht Bestandteil dieser Arbeit sind, implementiert sein.

erhalten hat und diese anschließend der Wegfindung übergibt. Als Antwort erhält sie von der Wegfindung die Information, über welchen Ausgang das Transportgut zu leiten ist⁷⁷.

Die eingeführten Softwarekomponenten lassen sich auf allen möglichen Zielplattformen implementieren, deshalb muss keine spezielle Anforderung an die Steuerungshardware innerhalb eines MFMs gestellt werden.

5.3. Modellierung

Für die Wegfindung innerhalb eines MFS muss ein Modell der Topographie des MFS vorliegen. In Kapitel 5.1 wurde definiert, dass ein MFS ausschließlich aus MFM besteht und diese MFM zu einem bestehenden MFS hinzugefügt werden können. Deshalb gilt es zu Anfang, eine Modellierung der Topographie für ein MFM unabhängig von dem MFS zu finden. Darauf aufbauend und aus den Modellen der MFM zusammengesetzt, kann ein MFS modelliert werden. Diese Modellierung kann sich an den bestehenden und im Stand der Technik vorgestellten Ansätzen zur Modellierung von MFS orientieren, muss diese jedoch um den Aspekt der Eigenständigkeit eines MFM erweitern: Bisherige Modellierungen beschreiben ein MFS als Gesamtheit, für das Ziel der Arbeit ist es jedoch unerlässlich, ein MFM mit einer Beschreibung seiner selbst auszustatten.

Aus der ebenfalls in Kapitel 5.1 festgelegten Prämisse, dass ein MFM immer nur aus einer Förderstrecke besteht, lässt sich ableiten, dass ein MFM bezüglich des Materialflusses eine bestimmte Anzahl von physikalisch vorhanden Schnittstellen aufweist⁷⁸, wobei diese Schnittstellen Eingänge, Ausgänge oder

⁷⁷ Denkbar ist in diesem Zusammenhang auch, dass die Hardwareansteuerung mehrere Zielinformationen für ein Stückgut abfragt, um alternative Bearbeitungsmöglichkeiten zu prüfen.

⁷⁸ Z.B. ergibt sich daraus für eine einfache Abzweigung ein Eingang und zwei Ausgänge.

Ein- und Ausgänge sein können⁷⁹. Diese Ein- und Ausgänge können mit Ein- und Ausgängen anderer MFM verbunden sein. Die derartig verbundenen MFM ergeben zusammen ein MFS. Die physikalischen Schnittstellen lassen sich in logische Ein- und Ausgangsschnittstellen überführen. Dabei wird jedem MFM eine Anzahl Eingangsschnittstellen r und eine Anzahl Ausgangsschnittstellen s zugeordnet. Für eine mathematische Darstellung des MFM kann auf die Graphentheorie /Diestel 00, Nägler 96/ zurückgegriffen werden. Ein MFM kann als gerichteter Graph T beschrieben werden:

$$T := (V, B) \quad (5-1)$$

Wobei V für die Knoten des MFM und B für die Äste⁸⁰ steht. Die logischen Eingangs- und Ausgangsschnittstellen werden als Eingangsknoten E und Ausgangsknoten A in diesem Graphen abgebildet⁸¹, wobei der Graph in Abhängigkeit von dem abzubildenden MFM r Eingangsknoten E und s Ausgangsknoten A enthalten kann. Weiterhin enthält der Graph einen Zentralknoten K für jedes MFM. Somit ergibt sich für V und B :

$$V = \{E_1, \dots, E_r, A_1, \dots, A_s, K\}$$

$$B = \{(E_1, K), \dots, (E_r, K), (K, A_1), \dots, (K, A_s)\}$$

wobei

$$(E_1, K) = e_1$$

⁷⁹ In diesem Fall arbeitet die Förderstrecke im bidirektionalen Betrieb.

⁸⁰ Im Rahmen dieser Arbeit soll der Begriff Ast für die Zu- und Abförderstrecken eines MFMs Verwendung finden. In der Graphentheorie wird in diesem Zusammenhang von einer Kante gesprochen.

⁸¹ Zu einer Diskussion der Notwendigkeit, physikalische Systemstrukturen im Modell wiederzugeben vgl. /Panreck 02/. Diese Vorgehensweise kann analog zu der von Gueorguiev und Ziems /Gueorguiev 99/ vorgeschlagenen Modellierung verstanden werden.

$$(K, A_1) = a_1$$

Vereinbarungsgemäß sollen nur der Zentralknoten K und die Ausgangsknoten A der Wegfindung dienen. Im nachfolgenden Bild ist dieses Modell dargestellt:

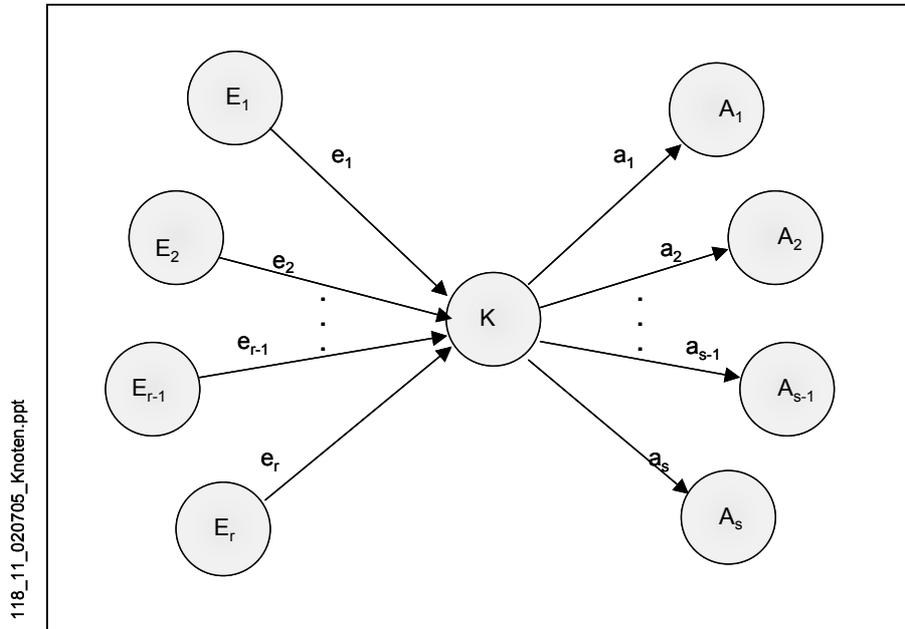


Abbildung 5-3: Modell eines MFM

Agrund der aufgestellten Forderung, dass das MFS eine Wegoptimierung vornehmen soll, muss sowohl für die Eingangs- als auch für die Ausgangsäste die einfache Abbildung derselben um eine qualitative Bestimmung dieser Äste erweitert werden, um eine Vergleichbarkeit bei Alternativwegen zu ermöglichen. Diese qualitative Bestimmung soll in Form von Kosten, die jedem Ast eindeutig zugeordnet werden, erfolgen. Kosten sind in diesem Falle nicht notwendigerweise als Kosten im Sinne der betriebswirtschaftlichen Kostenrechnung zu verstehen, sondern beziehen sich auf die Frage, welche Belastungen verursacht werden. Diese Kosten für jeden Eingangs- und Ausgangsast lassen

sich als eine Funktion c darstellen, die diesen Ästen jeweils eine reelle Zahl größer oder gleich Null⁸² zuordnet:

$$\text{kost}_m = c(a_m) \quad (5-2)$$

und

$$\text{kost}_n = c(e_n) \quad (5-3)$$

Ein bestimmtes MFM x kann mit dieser Modellierung als eine Menge von Knoten und Ästen alleinständig dargestellt werden. Die einzelnen Äste können mit Kosten belastet werden, was der späteren Wegoptimierung zugrunde liegt. Das damit aufgestellte Modell kann ein reales MFM beschreiben, wobei noch keinerlei Beziehungen zu anderen MFM enthalten sind. Diese Beschreibung entspricht also einem gefertigten MFM, dessen konkreter Einsatz innerhalb eines MFS noch unbekannt ist. Damit ist der Anspruch der verteilten Modellierung als Basis für die Adaptivität der Module erfüllt.

Da sich ein MFS aus mehreren MFM zusammensetzt und die Funktion der Wegfindung sich auf das MFS bezieht, müssen die eigenständige Modelle der MFM in ein Gesamtmodell für das MFS überführt werden können. Dabei soll folgende Vereinbarung gelten:

Ein MFM x ist ein Vorgänger zu einem anderen MFM y wenn gilt

$$A_m^x = E_n^y. \quad (5-4)$$

Für diesen Fall soll der daraus resultierende Graph vereinbarungsgemäß nur noch den Ausgangsknoten A_m^x enthalten. Im Rahmen der Modellierung werden also die logischen Ausgangsschnittstellen und der Zentralknoten erhalten. Dies

⁸² Negative Kosten für Wege sollen in diesem Zusammenhang nicht betrachtet werden, da diese für ein reines Transportproblem nicht relevant sind.

entspricht der Vereinbarung, nur Ausgangsknoten und Zentralknoten als Ziele für die Wegfindung zuzulassen, Eingangsknoten können nur an den Systemgrenzen des MFS existieren. Da die Eingangsknoten unmittelbar Abstraktionen von Eingängen einer Förderstrecke sind, können diese, wenn sie sich auf der Systemgrenze des MFS befinden, notwendigerweise keine Ziele sein.

Die Zuordnung eines bestimmten Ausgangsknotens A_m^x zu einem bestimmten Eingangsknoten E_n^y eines anderen MFM erweitert das bisherige, nur auf das MFM bezogene Modell um dessen Beziehungen zu anderen MFM innerhalb des MFS. Da damit die MFS spezifische Nutzung eines MFM ermöglicht wird, müssen diese Informationen innerhalb eines MFM ebenfalls vorgehalten werden. Über die einfache Kenntnis aller Nachfolger und Vorgänger MFM hinaus muss ebenso bekannt sein, über welche Äste diese zu erreichen sind. Dies bezieht sich sowohl auf den Ast des eigenen MFM als auch auf den Ast des anderen MFM⁸³.

Ein einfaches Beispiel dafür ist in der folgenden Abbildung graphisch dargestellt:

⁸³ Die Kenntnis beider Äste ist für eine verteilte Topographiebildung unabdingbar, beispielsweise sei die Neuintegration eines weiteren MFM in ein bestehendes MFS genannt: Das neue MFM muss dem vor ihm liegenden MFM nicht nur mitteilen, dass es sein neuer Nachfolger ist, sondern darüber hinaus über welchen eigenen Ast der Vorgänger das neue MFM erreichen kann.

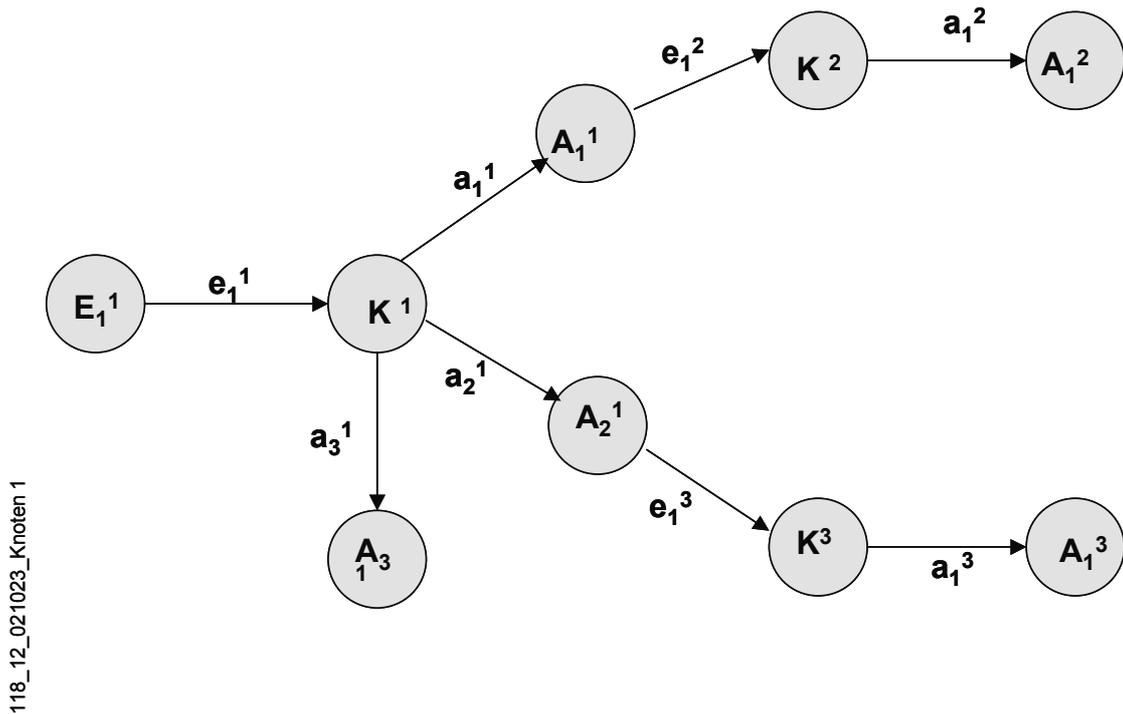


Abbildung 5-4: Beispiel eines MFS

Zusammenfassend ist also festzustellen, dass für eine verteilte Topographiebildung jedes MFM x einerseits bezüglich T , andererseits aber auch bezüglich seiner Beziehungen zu allen Vorgänger MFM y und allen Nachfolger MFM z vorab parametrisiert sein muss⁸⁴.

5.4. Algorithmus

Der in jedem MFM implementierte verteilte Algorithmus berechnet, wie in genau diesem MFM mit dem Transportgut zu verfahren ist, also an welchen der Ausgangsäste a_m das Transportgut optimalerweise weiterzuleiten ist. Entlang des Weges des Transportgutes erfolgt diese Berechnung jeweils neu aufgrund der dem jeweiligen MFM vorliegenden Daten. Um eine

⁸⁴ Wie weiter oben festgestellt wurde, ist T dabei statisch. Alle Beziehungen zu den Vorgängern und Nachfolgern sind dabei dynamisch. Dies ergibt sich aus der geforderten Anpassungsfähigkeit des Systems, zusätzliche MFM können jederzeit hinzukommen.

Wegoptimierung des Gesamtweges sicherzustellen, müssen die Einzel-Berechnungen der Wegfindung in jedem MFM auf konsistenten Daten beruhen. Die Konsistenz der Daten der MFM wird durch eine Aggregation der einzelnen Daten der MFM entgegen dem Materialfluss erreicht. Die von einem MFM erreichbaren anderen MFM ergeben sich aus der Aggregation der von seinen direkten Nachfolgern erreichbaren MFM.

Für die Wegfindung innerhalb des MFM muss für jeden Ausgangsast a_m bekannt sein, welche MFM über ihn zu erreichen sind und mit welchen Kosten dies verbunden ist. Diese Daten werden für jeden a_m in der Ausgangsast-Nachfolgertabelle AN_m abgelegt und stellen das Modell des MFS aus Sicht des MFM dar⁸⁵:

$$AN_m = \{(K^y, \text{kost}^y), \dots, (A_o^z, \text{kost}_o^z)\} \quad (5-5)$$

wobei:

K^y = Ein beliebiger Zentralknoten des MFM y

kost^y = Kosten zu dem Zentralknoten des MFM y

A_o^z = Ein beliebiger Ausgangsknoten o eines beliebigen MFM z

kost_o^z = Kosten zu dem Ausgangsknoten o des MFM z

In jedem MFM liegt also für jeden seiner Ausgangsäste m eine Tabelle AN_m vor. Damit ist der Ausgangsast o optimal, also kostengünstigst⁸⁶ für das Ziel⁸⁷ K^y eines Transportgutes wenn gilt:

⁸⁵ Hierbei muss es sich nicht um ein Gesamtmodell des MFS handeln. Das Modell erfolgt aus Sicht des MFM „Materialflussabwärts“, d.h. das Modell beinhaltet nur den Teil des MFS, der von dem MFM aus erreichbar ist.

⁸⁶ Für den Fall, dass zwei Ausgangsäste die selben Kosten zu einem gegebenen Ziel K^y haben, kann die Entscheidung, welcher Ausgangsast zu wählen ist, zufällig erfolgen.

$$\text{kost}^y (AN_o) < \text{kost}^y (AN_m) \text{ für } m \text{ über alle Ausgangsäste} \quad (5-6)$$

Da die Adaptivität bezüglich Änderungen des MFS Ziel der Arbeit ist, muss jedes MFM in der Lage sein, die Ausgangsast-Nachfolgertabellen (und damit sein Modell des MFS) dynamisch zu erstellen. Für die Aufstellung der Tabellen kann grundsätzlich zwischen zwei verschiedenen Fällen unterschieden werden:

1) Das MFM x verfügt an dem Ausgangsast m über kein MFM als Nachfolger. In diesem Fall ist der einzige, über den Ausgangsast zu erreichende Knoten der Ausgangsknoten A_m^x mit den entsprechenden Kosten des Astes:

$$AN_m^x = \{(A_m^x, \text{kost}_m^x)\} \quad (5-7)$$

2) Das MFM x besitzt an dem Ausgangsast m das MFM y als Nachfolger. In diesem Fall übergibt das MFM y seinem Vorgänger x eine Tabelle mit den über ihn zu erreichenden Knoten. Diese Tabelle soll als Eingangsast-Nachfolgertabelle des MFM y bezeichnet werden, da sie alle über dieses MFM erreichbaren Knoten bezogen auf dessen Eingangsast n angibt:

$$EN_n^y = \{\{K^j, \text{kost}^j\}, \dots, \{A_o^z, \text{kost}_o^z\}\} \quad (5-8)$$

⁸⁷ Für eine Analyse der verschiedenen Möglichkeiten, Zielinformationen anzugeben vgl. /Schwarzer 98, Gausemeier 98/.

Wobei:

n = die Nummer des Eingangsastes des Nachfolger-MFM y mit dem x verbunden ist

K^j = Ein beliebiger Zentralknoten des MFM j

$Kost^j$ = Kosten zu dem Zentralknoten des MFM j

A_o^z = Ein beliebiger Ausgangsknoten o eines beliebigen MFM z

$kost_o^z$ = Kosten zu dem Ausgangsknoten o des MFM z

Das MFM x transformiert EN_n^y für den Ausgangsast m in AN_m^x , indem die Kosten in EN_n^y jeweils um die Kosten des Ausgangsastes m erhöht werden:

$$AN_m^x = \{(K^j, Kost^j) / \exists (K^j, Kost_{alt}^j) \in EN_n^y \wedge Kost^j = Kost_{alt}^j + Kost_n^x\} \\ \cup \{(A_o^z, Kost_o^z) / \exists (A_o^z, Kost_{o,alt}^z) \in EN_n^y \wedge Kost_o^z = Kost_{o,alt}^z + Kost_n^x\}$$

(5-9)

Nachdem gezeigt wurde, wie die Ausgangsast-Nachfolgertabellen zu erstellen sind, und dabei die Eingangsast-Nachfolgertabelle EN_m^y (für MFM y über Eingangsast n) eingeführt wurde, ist es abschließend notwendig, deren Berechnung zu bestimmen. Dabei ist zu beachten, dass diese Tabelle individuell für jeden Eingangsast n existiert, da diese mit unterschiedlichen Kosten belastet sein können. Zuerst wird dafür die Zentralknoten-Nachfolgertabelle ZN^x vorgestellt und dann um die Kosten der individuellen Eingangsäste n erweitert werden.

Die Zentralknoten-Nachfolgertabelle ZN^x eines MFM x errechnet sich aus den Ausgangsast-Nachfolgertabellen AN_m^x aller Ausgangsäste m und dem Zentralknoten K^x mit Kosten 0. Wie oben beschrieben wurde, dienen die auf ZN^x basierenden EN_n^y dem Vorgänger-MFM zur optimalen Wegfindung für das Transportgut. Um eine optimale Lösung für das Problem der Wegfindung zu gewährleisten, muss deshalb ZN^x als kostenoptimale Aggregation

aller in den AN_m^x enthaltenen Nachfolger-Knoten erstellt werden. Für ZN^x gilt, dass im Falle von mehreren Ausgangsästen, die als Nachfolger den Knoten y anbieten, der Knoten y mit den Kosten $kost_n^y$ des Astes n in ZN_x^y versehen wird, wenn dies die minimalsten Kosten sind. Es gilt also:

$$\begin{aligned} ZN^x = & \{(K^j, kost_y^j) / \exists AN_y: (K^j, kost_y^j) \in AN_y \wedge Kost_y^j = \min_n Kost_n^j\} \\ & \cup \{(A_o^z, kost_y^z) / \exists AN_y: (A_o^j, kost_y^j) \in AN_y \wedge Kost_y^j = \min_n Kost_n^j\} \\ & \cup \{(K^x, 0)\} \end{aligned} \quad (5-10)$$

wobei:

K^j = Ein beliebiger Zentralknoten eines MFM j

$Kost_w^j$ = Kosten zu dem Zentralknoten des MFM j für den Ausgangsast y

y = alle Ausgangsäste

A_o^z = Ein beliebiger Ausgangsknoten o eines beliebigen MFM z

$kost_w, o^z$ = Kosten zu dem Ausgangsknoten o von MFM z für den Ausgangsast y

Da die Erstellung der Zentralknoten-Nachfolgertabelle in jedem MFM alle im Materialfluss nachfolgenden MFM betrifft, muss auch in jedem MFM überprüft werden, ob Schleifen vorliegen. Dies erfolgt, indem das MFM x bei Berechnung von ZN^x seinen Zentralknoten mit dem Kostensatz Null berücksichtigt: $\{K^x, 0\}$. Damit werden in jedem Fall die Kosten für K^x Null gesetzt.

Basierend auf ZN^x kann für jeden speziellen Eingangsast n des MFM x EN_n^x errechnet werden. Dies geschieht einfach, indem alle enthaltenen Kosten um die Kosten des Astes n erhöht werden:

$$EN_n^x = \{(K^j, Kost^j + c(e_n^x))\} \quad (5-11)$$

Zusammenfassend ist festzustellen, dass der vorgestellte Algorithmus in jedem MFM x existiert. Die Wegfindung beruht auf dem Vergleich der Tabellen AN_m^x , die für jeden Ausgangsast m des MFM x existieren, und die die über diesen Ausgangsast m erreichbaren Knoten zusammen mit den dabei über diesen Ast entstehenden Kosten enthalten. Aus den Ausgangsast-Nachfolgertabellen AN_m^x wird die Zentralknoten-Nachfolgertabelle ZN^x berechnet, wobei Schleifen berücksichtigt werden. Diese ZN^x wird um die den einzelnen Eingangsästen n zugeordneten Kosten erhöht für jeden Eingangsast n als EN_n^x erstellt und dann an die Vorgänger MFM gesendet. In den Vorgänger MFM wird die EN_n^x um die Kosten der dortigen Ausgangsäste erhöht und in die AN_m^y für die Vorgänger MFM y umgewandelt⁸⁸.

Die der Wegfindung als Modell des MFS zugrundeliegenden Tabellen AN_m^x ergeben sich also als Summe der Kosten der jeweilig billigsten Wege und erfüllen damit das Kriterium der Wegoptimierung. Die entsprechenden EN_n^x müssen jedem MFM durch seine Nachfolger im Materialfluss bereitgestellt werden, wofür das im folgenden Kapitel beschriebene Protokoll Verwendung findet.

⁸⁸ Vgl. dazu auch den erfolgten Patentantrag /Lorentz 02b/.

5.5. Protokoll

Abschließend für die Beschreibung der Grundlagen gilt es, das der Steuerungsmethode zugrunde liegende Protokoll, das im Austausch zwischen den einzelnen MFM Anwendung findet, darzustellen. Aus der Forderung nach einer Selbstkonfiguration des MFS, als deren Resultat ein Zustand im System vorliegt, in dem jedes MFM in der Lage ist, die Wegfindung wahrzunehmen⁸⁹, ergibt sich, dass einerseits geänderte Systemkonfigurationen kommuniziert werden müssen, andererseits müssen diese Änderungen vom Protokoll auch erkannt werden. Das Protokoll muss also Nachrichten bezüglich der folgenden Aspekte unterstützen:

Anmeldung im System: Nachdem ein MFM gestartet wird, muss es sich bei seinen direkten Vorgängern und Nachfolgern (die ihm durch die Initialisierung bekannt sind) anmelden. Bei dieser Anmeldung übergibt der Anmeldende entsprechende Daten über sein Topographie-Abbild⁹⁰. Das MFM, bei dem diese Anmeldung eingegangen ist, führt damit eine Aktualisierung seiner Daten durch. Dieser Zustand stellt für das entsprechenden MFM eine *Änderung im System* dar und löst die entsprechenden, weiter unten beschriebenen Vorgänge aus.

Im Rahmen der an die Methode gestellten Anforderung *Robustheit* wurde eingangs festgestellt, dass bei verteilten Systemen die Reihenfolge der eingehenden Nachrichten nicht von vornherein festliegt. Da die Initialisierungsdaten eines jeden MFM zur Abstimmung mit den benachbarten MFM herangezogen werden, kann dies zu einem unvorhersehbaren Verhalten führen, dem zu begegnen ist. Dieses Verhalten⁹¹ tritt bei inkonsistenten

⁸⁹ Bezugnehmend auf Kapitel 5.4 bedeutet dies, dass in allen MFM die zur Wegfindung notwendigen Ausgangsast-Nachfolgertabellen AN vorliegen.

⁹⁰ Es wird eine erste Eingangsast-Nachfolgertabelle, die jedoch nur den Zentralknoten und seine Initialisierungsdaten enthält, übergeben.

⁹¹ Beispielhaft sei das Einfügen eines weiteren MFM zwischen zwei bereits existierende MFM genannt: Die beiden ursprünglichen MFM sind miteinander als Vorgänger und Nachfolger initialisiert. Sollte bei einem Systemstart

Initialisierungsdaten auf⁹². Es wurde deshalb ein Datumsstempel für jeden Initialisierungsdatensatz eingefügt. Dieser Datumsstempel muss eine hinreichend hohe zeitliche Auflösung beinhalten, da die MFM durchaus im Bereich von Millisekunden im Netz hochfahren können. Der in dieser Arbeit vorgeschlagene Datumsstempel besitzt das Format Jahr, Monat, Tag, Stunde, Minute, Sekunde, Millisekunde und löst damit hinreichend genau auf.

Überwachung des Systems: Da es keine zentrale, überwachende Einheit innerhalb des Systems gibt, müssen die einzelnen MFM, die das MFS ausmachen, sich gegenseitig überwachen. Dies geschieht durch Vereinbarung eines regelmäßigen Heartbeat⁹³ zwischen einzelnen MFM. Dabei ist immer das MFM verpflichtet, den Heartbeat zu liefern, das im Materialfluss nachfolgend ist. Dieser Heartbeat soll im weiteren Verlauf als Heartbeat Klasse 1 bezeichnet werden und trägt außer der Angabe des Absenders keine weiteren Informationen. Sobald dieser Heartbeat ausbleibt, muss das überwachende MFM davon ausgehen, dass ein Ausfall vorliegt, was eine *Änderung im System* darstellt.

Als Ergebnis der beiden vorab dargestellten Sachverhalte ergeben sich *Änderungen im System*: Im Fall des Eintreffens einer Anmeldung im System muss der Empfänger diese neuen Daten weiterkommunizieren. Dieses Signal wird als Heartbeat Klasse 2 bezeichnet. Dieser Heartbeat enthält die komplette Eingangsst-Nachfolgertabelle des versendenden MFM. Der Eingang dieser

das neue MFM sich vor den beiden anderen (alten) MFM anmelden, würden die Anmeldungen der alten MFM die Initialisierungsdaten des neuen MFM unter diesen Umständen überschreiben.

⁹² Unter inkonsistenten Initialisierungsdaten ist der Fall zu verstehen, wenn beispielsweise für das MFM x in Bezug auf seinen Vorgänger y $E_n^x = A_m^y$ gilt, dies jedoch für das MFM y nicht gilt.

⁹³ Da in einem verteilten System damit gerechnet werden muss, dass ein Teil des Systems ausfällt, muss die Verfügbarkeit benötigter Komponenten ständig überprüft werden. Ähnliche Mechanismen sehen Suns Jini /Edwards 00/ und NDDS /Pardo-Castellote 99/ ebenfalls vor. Im weiteren Verlauf wird dieses regelmäßige Signal Heartbeat genannt.

Nachricht bei den Empfängern stellt dort eventuell⁹⁴ wiederum eine *Änderung im System* dar, wird also wiederum weitergeleitet. Dabei geschieht diese Kommunikation immer in dem Materialfluss entgegengesetzter Richtung⁹⁵.

Für den Fall, dass ein Überwachungssignal ausbleibt, muss das MFM seine eigenen Ausgangsast-Nachfolgertabellen löschen, es muss aber auch dafür sorgen, dass alle anderen MFM diese Daten löschen. Dafür wird ein Heartbeat Klasse 3, der eine Umkonfiguration des Systems anzeigt, verschickt.

Die *Anmeldung im System* wird vom MFM nach seinem Start regelmäßig verschickt, solange bis diese Nachricht angekommen ist. Sollte die Nachricht aber nicht ankommen, bedeutet dies für das MFM keinen Fehler. Die Heartbeats, die zur Überwachung dienen, werden von den MFM regelmäßig verschickt. Die Nachrichten über eine *Änderung im System* erfolgen nach Bedarf, also wenn ein solcher Zustand eintritt. Daraus folgt, dass die Konfiguration des Systems iterativ erfolgt⁹⁶. Ein MFM wird seine Vorgänger über die erste Anmeldung eines Nachfolgers informieren und die Nachricht wird von diesen entsprechend weiter propagiert. Sobald alle MFM gestartet sind und die Nachrichten darüber entsprechend propagiert sind, liegt die Systemkonfiguration vor. Das nachfolgende Sequenzdiagramm stellt die oben beschriebenen Nachrichten zusammenfassend in einem Beispiel für zwei MFM dar. Die in den MFM implementierten Software-Funktionen sind mit ihren Abkürzungen (P1-P4, T1-T4, W1 & W2) abgekürzt:

⁹⁴ Im Falle von vermaschten MFS würde dies zu ewig kreisenden Nachrichten führen, dies wird daher abgefangen, indem geprüft wird, ob die eingehende Nachricht tatsächlich eine *Änderung im System* bedeutet.

⁹⁵ Dies ist darin begründet, dass nur für die im Materialfluss weiter vorne liegenden MFM eine solche Änderung Relevanz hat.

⁹⁶ Dies ist bedingt durch die in einem verteilten System nicht festlegbare Reihenfolge des Hochfahrens der einzelnen MFM.

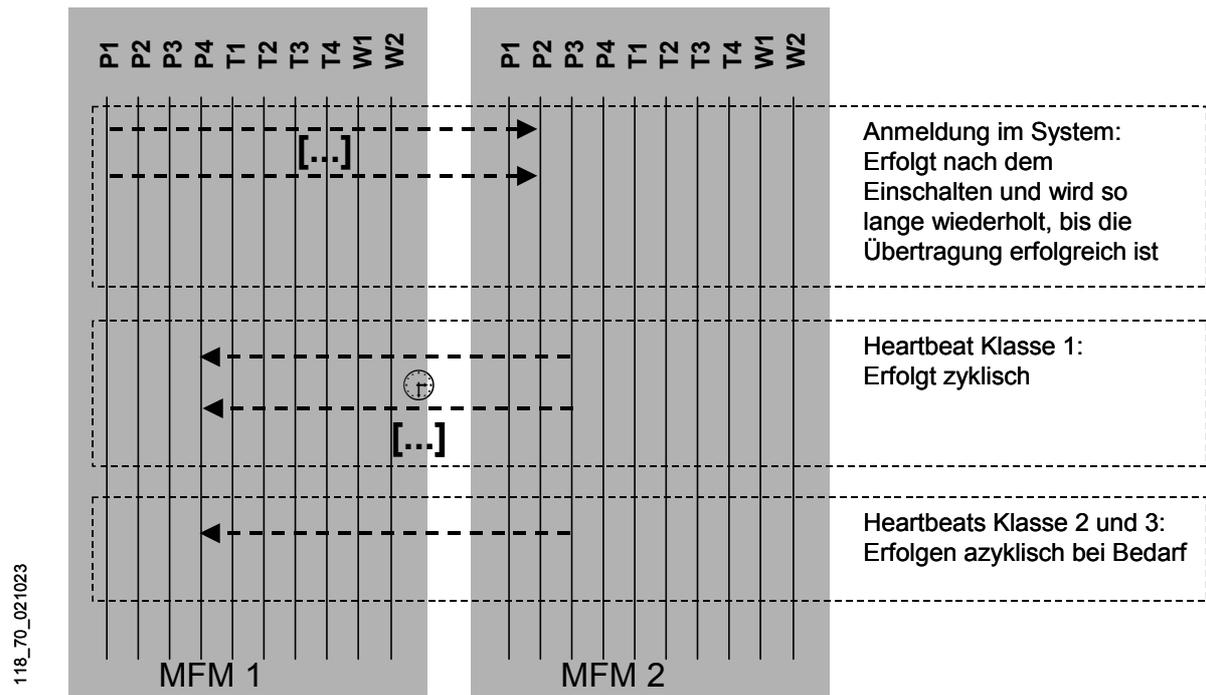


Abbildung 5-5: Sequenzdiagramm der Messages

6. Die Softwarekomponenten

An die funktionale Unterlegung der in Kapitel 5 vorgestellten Softwarekomponenten erfolgt anschließend eine Darstellung der einzelnen, innerhalb der Steuerung zu verwaltenden Daten. Abschließend werden die einzelnen Funktionen der Komponenten und ihre Austauschbeziehungen detailliert beschrieben.

6.1. Definition der Funktionen

Die in Kapitel 5.2 vorgestellten drei Komponenten Plug-and-Play, Topographiebildung und Wegfindung können als hierarchisch angeordnet verstanden werden. Die Softwarekomponente Plug-and-Play ist ausschließlich für die Kommunikation über das Netzwerk zuständig. Innerhalb des MFM kommuniziert sie nur mit der Softwarekomponente Topographiebildung⁹⁷. Diese Komponente kommuniziert darüber hinaus mit der Wegfindung, der sie auf Anfrage die Ausgangsast-Nachfolgertabellen zur Verfügung stellt. Die Aufgabe der Softwarekomponente Topographiebildung ist also, aus Informationen über Topographieänderungen die Zentralknoten-Nachfolgertabelle und die Eingangsast-Nachfolgertabellen zu generieren und diese, zusammen mit den Ausgangsast-Nachfolgertabellen, den beiden anderen Softwarekomponenten zur Verfügung zu stellen. Somit sind die Teile des Algorithmus, die die Topographie bilden, in dieser Softwarekomponente implementiert. Die Komponente Wegfindung wird von der Hardwareansteuerung unter Angabe von Zielinformationen nach Stellinformationen angefragt und nimmt dabei den Vergleich der Ausgangsast-Nachfolgertabellen vor.

Die Softwarekomponente Plug-And-Play verfolgt die Anmeldung und den Verbleib anderer MFM. Außerdem meldet sie das eigene MFM im MFS bei den direkten Vorgängern und Nachfolgern an. Daraus

⁹⁷ Einerseits liefert die Softwarekomponente Plug-and-Play die Eingangsast-Nachfolgertabellen, die es von den Nachfolgern erhält, andererseits benötigt es selber die Eingangsast-Nachfolgertabellen für die Kommunikation mit den Vorgängern.

ergeben sich vier Funktionen, die diese Komponente ausführen muss:

1. Anmeldung des eigenen MFM bei seinen eigenen Vorgängern und Nachfolgern. Dafür muss das MFM mit diesen Informationen initialisiert werden.
2. Entgegennahme und Bestätigung der Anmeldung von anderen MFM. Diese Funktion muss ständig im MFM laufen, da eine derartige Anmeldung zu jedem Zeitpunkt erfolgen kann.
3. Verschicken von Heartbeats. Die regelmäßigen Heartbeats der Klasse 1 werden automatisch verschickt, die unregelmäßigen Heartbeats der Klassen 2 und 3 werden nach Bedarf verschickt.
4. Entgegennahmen der Heartbeats. Im MFM müssen sowohl nicht regelmäßige Heartbeats der Klassen 2 und 3 verarbeitet als auch regelmäßige Heartbeats der Klasse 1 überwacht werden.

Die Softwarekomponente Topographiebildung erstellt aus den ankommenden Eingangsst-Nachfolgertabellen der direkten Nachfolger des MFM die Zentralknoten- und Eingangsst-Nachfolgertabellen. Somit ergeben sich die folgenden Funktionen des Moduls Topographiebildung:

1. Lege Ausgangsst-Nachfolgertabelle ab. Wenn sich ein MFM neu anmeldet, wird für dieses MFM dessen Eingangsst-Nachfolgertabelle in die Ausgangsst-Nachfolgertabelle umgerechnet und abgelegt. Im Falle eines Heartbeats, der über eine Änderungen bei diesem Nachfolger informiert, wird ebenfalls die neue Ausgangsst-Nachfolgertabelle abgelegt.
2. Lösche Ausgangsst-Nachfolgertabellen. Im Falle des Ausbleibens eines Heartbeats von einem Nachfolger stößt die Komponente Plug-and-Play diese Funktion an, um die Ausgangsst-Nachfolgertabelle löschen zu lassen.

3. Erstelle Eingangsst-Nachfolgertabelle. Aus den einzelnen Ausgangsst-Nachfolgertabellen wird die Zentralknoten-Nachfolgertabelle errechnet, die als Grundlage für die Eingangsst-Nachfolgertabellen dient.
4. Liefere Ausgangsst-Nachfolgertabellen. Die Ausgangsst-Nachfolgertabellen werden der Funktion Wegberechnung zurückgeliefert.

Die Softwarekomponente Wegfindung errechnet anhand der ihr von der Hardwareansteuerung übergebenen Zielinformationen und der Ausgangsst-Nachfolgertabellen den Ast des eigenen MFM, über den der Transport erfolgen soll⁹⁸. Da die Hardwareansteuerung die direkte Hardware des MFM steuert, muss noch eine Zuordnung des Asts zu der direkten Hardware erfolgen.

1. Wegberechnung: Aus der Zielinformation und den Ausgangsst-Nachfolgertabellen wird ein optimaler Weg und der dahin führende eigene Ast des MFM berechnet.
2. Zuordnung: Der von der Funktion Wegberechnung gelieferte Ast wird der direkten Hardware zugeordnet und zurück geliefert.

In der nachfolgenden Darstellung sind diese Funktionen in die Abbildung 5-2 eingeordnet. Anhand der Pfeile zwischen den einzelnen Softwarekomponente lässt sich der Kontrollfluss zwischen den Komponenten erkennen. Die Funktionen der Komponente Hardwareansteuerung sind nicht näher beschrieben, da dies nicht Inhalt dieser Arbeit ist. Die einzelnen Funktionen sind mit Kurzbezeichnungen (W1 & W2, T1 - T4 und P1 - P4) versehen, die im weiteren Verlauf der Arbeit verwendet werden.

⁹⁸ Bei der Berechnung des Weges sind vielfältige Optimierungen vorstellbar, im Rahmen dieser Arbeit soll der kürzeste Weg gesucht werden.

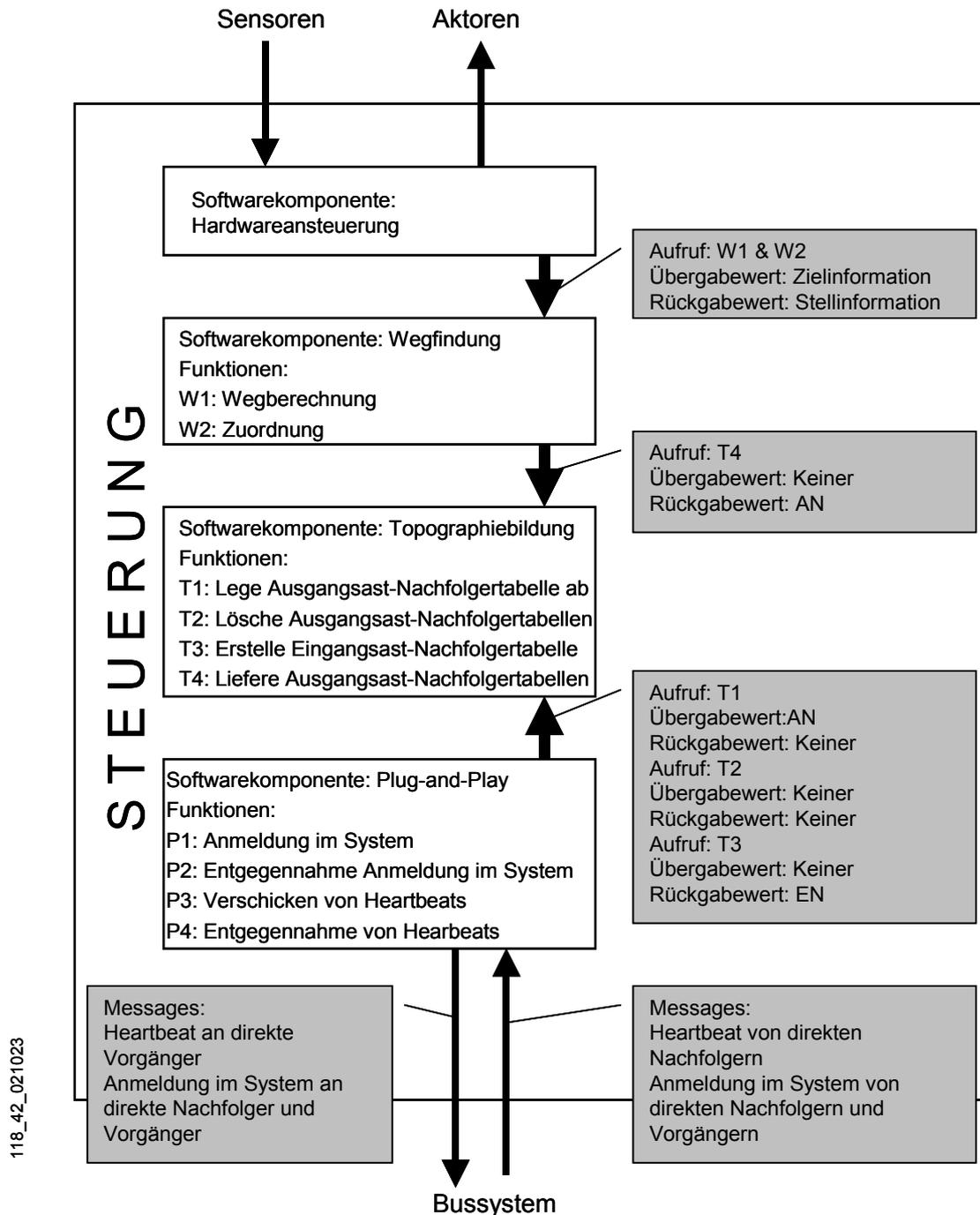


Abbildung 6-1: Übersicht entwickelter Softwarekomponenten und Funktionen, in Relation zur gesamten Steuerungsfunktion des MFV

6.2. Datenhaltung

Für die Ausführung der verschiedenen Softwarekomponenten ist es unerlässlich, Daten in jedem MFM lokal zu speichern. Diese Daten können in persistente, also auch nach einem Abschalten des MFM noch vorhandene, und nicht-persistente Daten unterschieden werden, wobei die persistenten Daten dynamischen Charakter haben können.

Zu den persistenten Daten gehören die Hardwaretabelle und die Initialisierungsdaten. Die Hardwaretabelle stellt eine Zuordnung der zu schaltenden Ausgänge zu einem Ast dar und ist persistent und statisch im Rahmen des Lebenszyklus des MFM⁹⁹. Die Initialisierungsdaten enthalten alle direkten Vorgänger und Nachfolger des MFM und müssen persistent im MFM vorgehalten werden, damit dieses nach einem Aus- und Einschalten in der Lage ist, mit seinen Nachbarn zu kommunizieren¹⁰⁰. Die nicht-persistenten Daten fallen im Rahmen des normalen Betriebs des MFS an und gehen bei dem Ausschalten des MFM verloren¹⁰¹. Die Zielinformation ist die von jedem Transportgut individuell gelieferte Information bezüglich seines Ziels¹⁰². Die Eingangsst-Nachfolgertabelle ist die Information, die das MFM an seine Vorgänger weitergibt. Die Liste der Ausgangsst-Nachfolgertabellen enthält die Daten der Nachfolger. Die beiden Timer für Vorgänger- und Nachfolger-Heartbeats stellen die Überwachungszeit bis zum Ablauf der Timer dar. Nachfolgende

⁹⁹ Eine Änderung dieser Tabelle ist mit dem physikalischen Umbau des betreffenden MFM verbunden und kann deshalb nicht automatisch erfolgen.

¹⁰⁰ Um der Forderung nach Adaptivität gerecht werden zu können, müssen diese Daten dynamisch verwaltet werden, da eine spätere Änderung von Vorgängern und Nachfolgern möglich ist.

¹⁰¹ Der Verlust der Daten ist notwendig, da nach einem Ausschalten unbedingt eine neue Konfiguration durchgeführt werden muss.

¹⁰² Diese Daten sind grundsätzlich dem Transportgut direkt zugeordnet (z.B. mittels eines Transponders), sie müssen jedoch auch im MFM verarbeitet und deshalb zwischengespeichert werden.

Darstellung ordnet die verschiedenen Daten den einzelnen Softwarekomponenten und ihren Funktionen zu:

		Daten						
		Hardwaretabelle (Persistent)	Zielinformation	Eingangsst-Nachfolgertabellen	Ausgangsst-Nachfolgertabellen	Initialisierungs-Daten (Persistent)	Vorgänger Heartbeat Timer	Nachfolger Heartbeat Timer
Komponente: Wegfindung								
W1	Wegberechnung		X		X			
W2	Zuordnung	X						
Komponente: Topographiebildung								
T1	Lege AN ab				X			
T2	Lösche AN				X			
T3	Erstelle EN			X	X	X		
T4	Liefere AN				X			
Komponente: Plug-and-Play								
P1	Anmeldung im System			X		X		
P2	Entgegennahme Anm. im System						X	X
P3	Verschicken von Heartbeats			X		X	X	X
P4	Entgegennahme von Heartbeats						X	X

Abbildung 6-2: Funktionen und Datennutzung

Eine detaillierte Spezifikation der vorzuhaltenden Daten kann den XML-Beschreibungen in Anhang A entnommen werden.

6.3. Plug-and-Play

Die Spezifikation aller durch die Softwarekomponente Plug-and-Play verschickten Messages ist dem Anhang A zu entnehmen.

6.3.1. Anmeldung im System (Funktion P1)

Die Anmeldung im System erfolgt durch das MFM automatisch bei jedem Hochfahren aus dem ausgeschalteten Zustand. Dabei werden die direkten Vorgänger und Nachfolger, die aus den Initialisierungsdaten bekannt sind (Kapitel 6.1), über die Existenz des MFM informiert.

Den Vorgängern wird die Existenz des eigenen MFM zusammen mit dem Ast des Vorgänger-MFM, über den das eigene MFM zu erreichen ist, mitgeteilt. Mit diesen Informationen kann der Vorgänger seine eigenen Initialisierungsdaten anpassen und eine entsprechende Timeoutverfolgung verwalten. Den Nachfolgern wird die Existenz des MFM zusammen mit der Timeoutzeit, nach der das MFM davon ausgeht, dass die Nachfolger-MFM nicht mehr leben, mitgeteilt. Damit gilt die Zeit, innerhalb derer der Nachfolger sich mittels eines Heartbeat Klasse 1 melden muss, als vereinbart. Der Versand der Telegramme erfolgt durch die Funktion P1 ohne Überwachung eventuell auftretender Fehler während der Kommunikation¹⁰³.

6.3.2. Entgegennahme Anmeldung im System (Funktion P2)

Um die Adaptivität des MFS zu gewährleisten, muss jedes MFM jederzeit in der Lage sein, Anmeldungen neuer Vorgänger oder Nachfolger entgegenzunehmen und seine eigenen Initialisierungsdaten und Nachfolgertabellen entsprechend zu ändern. Daraus ergibt sich, dass die Funktion P2 in einer

¹⁰³ Für den Fall, dass ein Fehler eintritt, ist keine weitere Maßnahme planbar. Aufgrund der Verteiltheit des Systems ist es durchaus möglich, dass die Empfänger der Messages noch nicht gestartet sind. Dieses Problem wird behoben, wenn diese MFM starten und ihrerseits eine Message verschicken.

ständigen Schleife auf neue Anmeldungen warten muss. Im Falle des Eingangs einer neuen Anmeldung durch einen Vorgänger oder Nachfolger wird in die Funktion *Aktualisieren_Daten* verzeigt. Abhängig davon, ob die Anmeldung von einem Vorgänger oder Nachfolger erfolgt, sind zwei verschiedene Reaktionen möglich.

In beiden Fällen muss das neue MFM in die Initialisierungsdaten des empfangenden MFM übernommen werden. Vorher erfolgt jedoch ein Vergleich der Zeitstempel; nur wenn der neue Wert jünger ist als der alte, kann er übernommen werden.

Im Falle der Neuanschmeldung eines Vorgängers muss für diesen Vorgänger der Heartbeat-Timer entsprechend gesetzt werden. Außerdem muss an dieses MFM ein Heartbeat Klasse 2 verschickt werden, um ihm die aktuelle Eingangsast-Nachfolgertabelle mitzuteilen¹⁰⁴.

Im Falle der Neuanschmeldung eines Nachfolgers muss dessen EN in die entsprechende AN umgerechnet und abgelegt werden. Dies geschieht durch Aufruf der Funktion *Lege Ausgangsast-Nachfolgertabelle ab (Funktion T1)*. Nachdem dies geschehen ist, wird ein Heartbeat Klasse 2 an alle Vorgänger geschickt, um diese über die Änderung der Eingangsast-Nachfolgertabelle zu informieren. Außerdem wird der Heartbeat-Timer entsprechend gesetzt.

Der folgende Pseudocode stellt diesen Vorgang dar:

Funktion P2: Entgegennahme Anmeldung im System

```
VOID Aktualisieren_Daten (Anmeldung_im_System)
```

```
{
```

```
Wenn Datum (Anmeldung_im_System(MFM)) > Datum (Initialisierungsdaten(MFM))
```

```
    Dann Ersetze Initialisierungsdaten (MFM) mit (Anmeldung_im_System(MFM))
```

¹⁰⁴ Der Heartbeat Klasse 2 muss nur an diesen Vorgänger erfolgen, da sich die Eingangsast-Nachfolgertabellen nicht ändern, somit eventuell andere vorhandene Vorgänger nicht informiert werden müssen.

```
Setze Heartbeat (Knoten)
Wenn Anmeldung_im_System = Vorgänger
    Dann Heartbeat_Klasse2 (MFM)
Wenn Anmeldung_im_System = Nachfolger
    Dann {
        Lege Ausgangsast-Nachfolgertabelle ab (Anmeldung_im_System)
        Heartbeat_Klasse2 ()
    }
}
```

Abbildung 6-3: Pseudocode *Entgegennahme Anmeldungen*

6.3.3. *Verschicken von Heartbeats (Funktion P3)*

Es existieren drei verschiedene Arten von Heartbeats:

Heartbeat Klasse 1: Diese Message enthält als Inhalt den Nachrichtentyp, Empfängerknoten und Absenderknoten.

Heartbeat Klasse 2: Diese Message enthält die gesamte Eingangsast-Nachfolgertabelle.

Heartbeat Klasse 3: Die Nachricht enthält als Angabe den empfangenden Knoten, den initiierenden Knoten und einen Zeitstempel.

Die Beschreibung der Abläufe innerhalb des MFM erfolgt getrennt für alle drei Heartbeat-Typen¹⁰⁵. Dabei werden die speziellen „Sende“- Methoden und die Methoden, die die eigentlichen Messages erzeugen nicht näher beschrieben¹⁰⁶. Heartbeat Klasse 1 wird immer durch einen Timer ausgelöst, die Abläufe für den Fall, dass der Heartbeat-Timer eines MFMs herunterläuft, sind in folgendem Pseudocode dargestellt:

¹⁰⁵ Grundsätzlich kann die Implementierung innerhalb einer Klasse mit verschiedenen Konstruktoren erfolgen.

```
Funktion P3: Verschicken von Heartbeats
VOID Heartbeat_Klasse1 ():
{
Heartbeat HB1
Solange in Initialisierungsdaten noch Vorgänger
    {
    Nimm MFM aus Initialisierungsdaten
    Erzeuge HB1 (Klasse1, Eigenes_MFM, MFM)
    Schicke HB1 (MFM)
    Setze Heartbeat-Timer(MFM)
    }
}
```

Abbildung 6-4: Pseudocode *Heartbeat* - Klasse 1

Heartbeats der Klasse 2 können sowohl von der Funktion *Entgegennahme Anmeldung im System (Funktion P2)* und von der Funktion *Entgegennahme von Heartbeats (Funktion P4)* aufgerufen werden. Im ersten Fall muss der Heartbeat nur an ein MFM verschickt werden (den Vorgänger, von dem die Anmeldung eingegangen ist), im zweiten Fall müssen alle Vorgänger über die Topographieänderung informiert werden:

¹⁰⁶ Die Sende-Methoden sind stark implementierungsabhängig, und die einzelnen Messages sind dem Anhang zu entnehmen. Deshalb soll hier auf eine detaillierte Darstellung verzichtet werden.

```

Funktion P3: Verschicken von Heartbeats
VOID Heartbeat_Klasse2 (MFM):
{
Heartbeat HB2
Eingangsst-Nachfolgertabelle = Erstelle Eingangsst-Nachfolgertabelle ()
Wenn MFM = 0 Dann
    {
        Solange in Initialisierungsdaten noch Vorgänger(MFM)
            {
                Nimm MFM aus Initialisierungsdaten
                Erzeuge HB2 (Klasse2, Eigenes_MFM, MFM, Eingangsst-Nachfolgertabelle)
                Sende HB2 (MFM)
                Setze Heartbeat-Timer(MFM)
            }
        }
    Andernfalls
        {
            Erzeuge HB2 (Klasse2, Eigenes_MFM, MFM, Eingangsst-Nachfolgertabelle)
            Sende HB2 (MFM)
        }
    }
}

```

Abbildung 6-5: Pseudocode *Heartbeat - Klasse2*

Ein Heartbeat Klasse 3 wird in zwei Fällen verschickt, einerseits, wenn ein Knoten ausgefallen ist und andererseits, wenn ein Heartbeat Klasse 3 eingegangen ist. Im ersten Fall erzeugt das MFM eine neue Message für einen Heartbeat Klasse 3, wobei es sich selbst als initiierendes MFM einträgt und den Zeitstempel setzt. Im zweiten Fall wird geprüft, ob der Heartbeat Klasse 3 bereits vorliegt, sollte dies nicht der Fall sein, wird der Heartbeat Klasse 3 weiter verschickt¹⁰⁷.

¹⁰⁷ Die Prüfung, ob der Heartbeat Klasse 3 bereits vorliegt, erfolgt anhand des initiierenden MFM und des Zeitstempels und dient dazu, ein Umlaufen von Heartbeats Klasse 3 in vermaschten MFS zu verhindern.

```
Funktion P3: Verschicken von Heartbeats
VOID Heartbeat_Klasse3 (Heartbeat):
{
Heartbeat HB3
Wenn Heartbeat = nil Dann
    {
    Erzeuge HB3 (Klasse3, Eigenes_MFM, Datum)
    }
Lösche Ast-Nachfolgertabellen
Letzter_Heartbeat = Heartbeat
Solange in Initialisierungsdaten noch Vorgänger(MFM)
    {
    Nimm MFM aus Initialisierungsdaten
    Sende HB3 (MFM)
    Setze Heartbeat-Timer(MFM)
    }
}
```

Abbildung 6-6: Pseudocode *Heartbeat* - *Klasse3*

6.3.4. *Entgegennahme von Heartbeats (Funktion P4)*

Die Funktion P4 ist ebenso wie die Funktion P2 für die Entgegennahme von Messages anderer MFM verantwortlich und damit in einem ständigen Wartezustand auf eintretende Messages. Da diese Funktion aber auch die Überwachung der Nachfolger-MFM übernimmt, muss gleichzeitig auch das Ausbleiben von Heartbeats Klasse 1 durch die Überwachung eines Timers zu einer Aktion führen. Grundsätzlich kann also in der Funktion P4 zwischen zwei Ereignissen unterschieden werden. Einerseits wird das Ausbleiben eines erwarteten Heartbeats (also der Ablauf eines Heartbeat-Timers) und andererseits wird der Eingang eines Heartbeats als Ereignis aufgefasst. Im Falle des Eingangs eines Heartbeats sind für die drei verschiedenen Heartbeat-Klassen unterschiedliche Reaktionen nötig, wobei diese sich ebenfalls für verschiedene Zeitpunkte des Eingangs der Heartbeats unterscheiden können.

Die Funktion *Knoten_Streichen* wird ausgeführt, wenn ein Heartbeat eines Nachfolgers ausbleibt. Die Ausgangsast-Nachfolgertabellen werden durch den Aufruf der Funktion *Lösche Ausgangsast-Nachfolgertabellen* (Funktion T2) gestrichen und die direkten Vorgänger des MFM durch Aufruf der Funktion *Verschicken von Heartbeats* (Funktion P3) mittels eines Heartbeat Klasse 3 über diesen Vorgang informiert.

Funktion P4: Entgegennahme von Heartbeats

VOID Knoten_streichen (Timer):

```
{  
  Lösche Ausgangsast-Nachfolgertabellen  
  Heartbeat_Klasse3 ()  
}
```

Abbildung 6-7: Pseudocode *Entgegennahme Heartbeats* - *Knoten_streichen*

Im Falle des Eingangs eines Heartbeats sind drei verschiedene Fälle zu unterscheiden, die ausgeführt werden müssen:

Fall 1 behandelt den Eingang eines Heartbeat Klasse 3: Dieser Heartbeat weist auf den Ausfall eines Knotens hin. Im Falle des Eingangs prüft das MFM, ob es diesen Heartbeat bereits erhalten hat. Sollte dies der Fall sein, wird der Heartbeat verworfen. Andernfalls müssen die eigenen Ausgangsast-Nachfolgertabellen gelöscht werden und ein Heartbeat Klasse 3 an alle Vorgänger verschickt werden.

Fall 2 behandelt den rechtzeitigen, also innerhalb der Überwachungszeit liegenden, Eingang der Heartbeats Klasse 1 und 2: Der Heartbeat Klasse 1 enthält das Signal für keine Änderungen. In diesem Fall ist keine weitere Aktion durch das MFM erforderlich. Lediglich der Timer, der die Zeit bis zum Eingang des Heartbeats überwacht, muss wieder zurückgesetzt werden. Der Heartbeat Klasse 2 enthält das Signal für Änderungen zusammen mit der neuen Eingangsast-Nachfolgertabelle aus Sicht des Nachfolgers. In diesem Fall muss die in dem Heartbeat enthalten Eingangsast-Nachfolgertabelle durch die

Funktion T1 in die entsprechende Ausgangsast-Nachfolgertabelle umgewandelt und abgelegt werden. Als nächstes muss ein Heartbeat Klasse 2 an alle Vorgänger verschickt werden, um diese über die Änderungen zu informieren. Abschließend muss ebenso der Timer zurückgesetzt werden.

Fall 3 behandelt den nicht rechtzeitigen, also nach Ende der Überwachungszeit erfolgenden Eingang der Heartbeats Klasse 1 und 2: Für Heartbeats beider Klassen, die nach Ablauf der Überwachungszeit eingehen, muss das MFM eine Neuansmeldung an das absendenden MFM verschicken. Außerdem wird ebenso der Timer zurückgesetzt. Dieser Sachverhalt ist in folgendem Pseudocode dargestellt:

```

Funktion P4: Entgegennahme von Heartbeats
VOID Aktualisieren (Heartbeat):
{
  Wenn Heartbeat = Klasse 3
    Dann
      {
        Wenn Heartbeat <> Letzter_Heartbeat
          Dann
            {
              Heartbeat_Klasse 3 (Heartbeat)
            }
          }
        }
  Wenn Knoten(Heartbeat) nicht in Initialisierungsdaten
    Dann
      {
        Anmeldung_im_System(MFM)
        Setze Heartbeat-Timer(MFM)
      }
  Andernfalls {
    Wenn Heartbeat = Klasse 1
      Dann {Setze Heartbeat-Timer(MFM)}
      Andernfalls
        {
          Lege_Ausgangsast-Nachfolgertabelle_ab (MFM)
          Heartbeat_Klasse2()
          Setze Heartbeat-Timer(MFM)
        }
      }
  }
}

```

Abbildung 6-8: Pseudocode *Entgegennahme Heartbeats - Aktualisieren*

6.4. Topographiebildung

Der in Kapitel 5.4 eingeführte Algorithmus wird bezüglich seiner die Topographie bildenden Teile im Rahmen der Softwarekomponente Topographiebildung implementiert. Dies erfolgt aus Daten, die von den Nachfolgern des MFM geliefert werden und innerhalb des MFM von der Komponente Plug-and-Play der Softwarekomponente Topographiebildung zur Verfügung gestellt werden. Hauptaufgabe der Softwarekomponente ist es also, die Ausgangsast-Nachfolgertabellen zu erstellen und zu verwalten und basierend darauf, die Eingangsast-Nachfolgertabellen¹⁰⁸ zu generieren.

Da davon auszugehen ist, dass die von den Nachfolgern gelieferten¹⁰⁹ Eingangsast-Nachfolgertabellen zeitlich unsynchronisiert eintreffen, müssen diese Tabellen verwaltet werden, was die zweite wichtigen Aufgabe der Softwarekomponente Topographiebildung ist. Die jeweils aktuelle Ausgangsast-Nachfolgertabelle muss für jeden Ausgangsast gespeichert werden und für das Auftreten eines Heartbeat Klasse 2 durch die neue Tabelle ersetzt werden. Im Falle des Ausbleibens eines Heartbeats werden die Ausgangsast-Nachfolgertabellen ebenfalls gelöscht.

Der Aufruf der Funktion *Lege Ausgangsast-Nachfolgertabelle ab* (Funktion T1) erfolgt entweder durch die Funktion P2, wenn sich ein neuer Nachfolger anmeldet, oder durch die Funktion P4, wenn ein bereits gemeldeter Nachfolger die Änderung seiner Eingangsast-Nachfolgertabelle mitteilt. Die Funktion T1 erhält dann die von dem Nachfolger des eigenen MFM übermittelte Eingangsast-Nachfolgertabelle und berechnet aus ihr analog der

¹⁰⁸ Entsprechend dem in Kapitel 5.4 eingeführten Algorithmus sind die Ausgangsast-Nachfolgertabellen Grundlage der Eingangsast-Nachfolgertabellen.

¹⁰⁹ Dies geschieht im Rahmen der Anmeldung im System und der Heartbeats Klasse 2.

Gleichung 5.8 die entsprechende Ausgangsast-Nachfolgertabelle und legt diese ab.

Für den Fall, dass der Heartbeat eines MFM ausbleibt, müssen durch die Funktion *Lösche Ausgangsast-Nachfolgertabellen* (Funktion T2) alle Ausgangsast-Nachfolgertabellen gelöscht werden. Dies wird durch die Funktion *Entgegennahme von Heartbeats* (Funktion P4) ausgelöst.

Die Funktion *Erstelle Eingangsast-Nachfolgertabelle* (Funktion T3) errechnet aus den Ausgangsast-Nachfolgertabellen die Zentralknoten-Nachfolgertabelle entsprechend Gleichung 5-9 und darauf aufbauend die Eingangsast-Nachfolgertabellen. Sie wird von der Funktion P4 im Falle des Eingangs eines Heartbeats Klasse 2 aufgerufen.

Von der Funktion *Wegberechnung* werden die Ausgangsast-Nachfolgertabellen der Nachfolger benötigt. Die Funktion *Liefere Ausgangsast-Nachfolgertabellen* (Funktion T4) liefert diese im Rahmen ihres Aufrufes zurück.

6.5. Wegfindung

Die Softwarekomponente Wegfindung beinhaltet die Implementierung der Teile des in Kapitel 5.4 vorgestellten Algorithmus, die sich auf die reine Wegfindung beziehen. Damit stellt sie die Schnittstelle zu der Komponente Hardwareansteuerung dar, da die Funktion *Wegberechnung* (Funktion W1) von dieser aufgerufen wird. In dem Aufruf von W1 wird die Zielinformation des Transportgutes übergeben, und daraufhin ruft die Funktion W1 die Funktion T4 auf und erhält als Rückgabewerte die Ausgangsast-Nachfolgertabellen für alle Ausgangsäste des MFM. Da die eigentliche Wegfindung auf den bereits vorberechneten Ausgangsast-Nachfolgertabellen beruht, handelt es sich um den einfachen Vergleich der vorhandenen AN analog zu (5-5). Anschließend wird der errechnete Ausgangsast durch Aufruf der Funktion *Zuordnung* (Funktion W2) in die entsprechend anzusteuernde Hardware aufgelöst. Diese Auflösung ist notwendig, da der Ausgangsast nicht direkt an die Hardwareansteuerung zurückgegeben werden kann (da dieser Wert dort nicht verarbeitet werden kann) und geschieht durch Auslesen der entsprechenden Informationen aus der Hardwaretabelle. Diese Tabelle stellt somit die Schnittstelle zwischen der implementierungsabhängigen Hardware / Hardwareansteuerung und der in dieser Arbeit entwickelten Methode dar.

7. MFM übergreifende Abläufe

In den vorangegangenen Kapiteln wurden die drei Softwarekomponenten Plug-and-Play, Topographiebildung und Wegfindung jeweils einzeln mit ihren Funktionen vorgestellt. Da aber bei einem verteilten Steuerungssystem die einfache Darstellung der Abläufe innerhalb der Subsysteme nicht ausreichend für das Verständnis des zeitlichen Verhalten des Systems ist¹¹⁰, soll im Rahmen dieses Kapitels das Zusammenwirken aller drei Komponenten über mehrere MFM hinweg dargestellt werden. Diese Darstellung soll anhand von Anwendungsszenarien erfolgen. Die folgenden fünf Szenarien decken alle auftretenden Zustände des Systems ab:

- *Systemstart*: In diesem Kapitel soll gezeigt werden, wie sich das Gesamtsystem im Rahmen des Neustarts des MFS verhält. Dies bezieht sich speziell auf den Ablauf des Austausches von Topographiedaten.
- *Normaler Betrieb*: Der Ablauf innerhalb eines MFM, der ausgelöst wird, wenn ein Transportstück an dem MFM ankommt, wird dargestellt.
- *Hinzufügen eines MFM*: Hier wird der Umbau eines MFS anhand des Zufügen eines zusätzlichen MFM gezeigt.
- *Ausfall eines MFM*: Diese Szenario gilt sowohl für den Umbau als auch für den Ausfall eines MFM.
- *Heartbeat zu spät*: Dieses spezielle Szenario beschreibt das Verhalten, wenn ein MFM - beispielsweise aufgrund einer Busstörung - nicht in der Lage war, seinen Heartbeat rechtzeitig zu verschicken, das MFM jedoch trotzdem nicht ausgefallen ist.

¹¹⁰ Für eine umfangreiche Einführung in diese Problematik vgl. /Wellings 02/.

Die nachfolgenden Darstellungen beschränken sich darauf, nur die Anzahl der MFM darzustellen, die für das Verständnis des Systemverhaltens notwendig sind. Eine größere Anzahl von MFM in diesem Rahmen darzustellen wäre der Verständlichkeit der Vorgänge eher abträglich und wurde daher unterlassen.

Zu Anfang wird in jedem Kapitel das Szenario kurz vorgestellt und dann anhand einer graphischen Darstellung der einzelnen Vorgänge und einer diese beschreibenden Tabelle erläutert. Die graphische Darstellung erfolgt in Anlehnung an das aus der UML bekannte Sequenzdiagramm¹¹¹. Für Messages von einem MFM zum nächsten wurden gestrichelte Pfeile verwendet, für Messages innerhalb des MFMs durchgängige Pfeile. Weiterhin wurde ein asynchroner, paralleler Ablauf zugelassen, da mit dem Verschicken der Messages zwischen den MFM der Kontrollfluss das MFM nicht verlässt. Für alle Szenarien, außer Szenario 2, wurde auf die Darstellung der Softwarekomponente Hardwareansteuerung verzichtet, da dies nicht Teil der Arbeit ist. Innerhalb der Darstellungen wurden die Software-Funktionen der Softwarekomponenten in der abgekürzten Form (P1-P4, T1-T4, W1 & W2) angegeben.

7.1. Systemstart

Das Szenario *Systemstart* beschreibt die in jedem einzelnen MFM ablaufenden Vorgänge beim Einschalten des MFS. Dabei ist zu beachten, dass keine vorher festgelegte Einschaltreihenfolge für die einzelnen MFM existiert. Das heißt, dass die Reihenfolge, in der die einzelnen MFM verfügbar werden, nicht vorausgesehen werden kann. Das nachfolgende Szenario stellt also eine von mehreren Möglichkeiten dar, kann aber als repräsentativ für alle Möglichkeiten betrachtet werden.

¹¹¹ Für einen Überblick der Applikationsarten und der damit verbundenen vorgeschlagenen Erweiterungen vgl. /Braatz 01/.

Im dem folgenden Beispiel sind drei MFM dargestellt, wobei diese die (zufällige) Einschaltreihenfolge 2 - 1 - 3 haben. Die angenommene Förderrichtung ist 1 - 2 - 3.

Im Rahmen des *Systemstarts* liegen den MFM nur Informationen bezüglich ihrer direkten Vorgänger und Nachfolger vor¹¹². Um lokal im MFM Entscheidungen bezüglich der Wegfindung treffen zu können, müssen diese Informationen um Informationen der jeweiligen Nachfolger erweitert werden. Weiterhin muss für jedes MFM festgestellt werden, welche eigenen Nachfolger es bezüglich ihrer Heartbeats überwachen muss und an welche eigenen Vorgänger es ein Heartbeat schicken muss.

Dafür tauschen die MFM untereinander die Message *Anmeldung im System* aus. Darin enthalten übermitteln die MFM die ihnen vorliegende Eingangsast-Nachfolgertabelle an ihre Vorgänger. Sobald sich diese lokal in einem MFM vorliegende Tabelle ändert (weil beispielsweise dessen Vorgänger eine geänderte Eingangsast-Nachfolgertabelle mitgeteilt hat), wird diese neue EN ebenfalls an die Vorgänger weitergeleitet. In dem Szenario *Systemstart* werden nur die beiden Softwarekomponenten Plug-and-Play und Topographiebildung durchlaufen. Das Szenario kann als abgeschlossen angesehen werden, wenn die Konfiguration des Systems bezüglich aller in MFM lokal vorliegenden Nachfolgertabellen als auch bezüglich aller in MFM lokal vorliegenden Informationen für Heartbeats vorliegt.

¹¹² Im Gegensatz zum Szenario *Hinzufügen eines MFM* wird in diesem Szenario von konsistenten Initialisierungsdaten bezüglich Vorgängern und Nachfolgern ausgegangen.

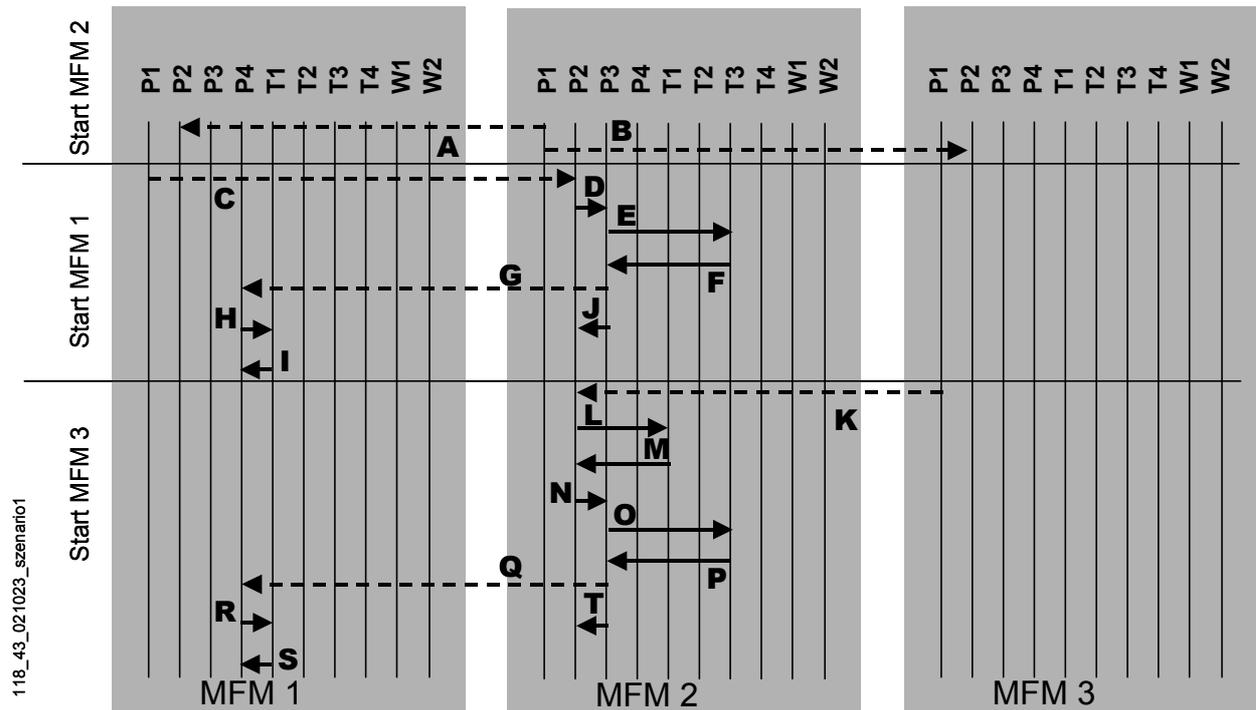


Abbildung 7-1: Sequenzdiagramm Szenario 1

Schritt	Auslöser	MFM	Beschreibung	Daten
A ¹¹³		2	P1: Message Anmeldung_im_System von MFM2 an MFM1	
B ¹¹⁴		2	P1: Message Anmeldung_im_System von MFM2 an MFM3	
C ¹¹⁵		1	P1: Message Anmeldung_im_System von MFM1 an MFM2	

¹¹³ MFM 1 ist zu diesem Zeitpunkt noch nicht aktiv, empfängt die Message also nicht.

¹¹⁴ MFM 3 ist zu diesem Zeitpunkt noch nicht aktiv, empfängt die Message also nicht.

¹¹⁵ MFM 1 ist zu diesem Zeitpunkt gestartet.

D	C	2	P2: Ruft P3 auf	HB-Timer gesetzt
E	D	2	P3: Verschicken von Heartbeats: Ruft T3 auf	
F	E	2	T3: Gibt EN an P3 zurück	EN erstellt
G	F	2	P3: Message Heartbeat Klasse 2 an MFM 1	
H	G	1	P4: Entgegennahme von Heartbeats: Ruft T1 auf	HB-Timer gesetzt
I	H	1	T1: Legt AN ab. Kehrt zu P4 zurück	AN abgelegt
J	G	2	P3: Kehrt zu P2 zurück	
K ¹¹⁶		3	P1: Message Anmeldung_im_System von MFM3 an MFM2	%
L	K	2	P2: Ruft T1 auf	HB-Timer gesetzt
M	L	2	T1: Lege Ausgangsast-Nachfolgertabelle ab. Kehrt zu P2 zurück	AN abgelegt
N	M	2	P2: Ruft P3 auf	
O	N	2	P3: Verschicken von Heartbeats: Ruft T3 auf	
P	O	2	T3: Erstellt neue EN und gibt diese an P3 zurück	EN erstellt
Q	P	2	P3: Message Heartbeat Klasse 2 an MFM 1	
R	Q	1	P4: Entgegennahme von Heartbeats: Ruft T1 auf	
S	R	1	T1: Lege Ausgangsast-Nachfolgertabelle ab. Kehrt zu P4 zurück	AN abgelegt
T	Q	2	P3: Kehrt zu P2 zurück	

Tabelle 7-1: Erläuterungen zum Sequenzdiagramm des Szenarios 1

¹¹⁶ MFM 3 ist zu diesem Zeitpunkt gestartet.

7.2. Normaler Betrieb

Das Szenario *Normaler Betrieb* beschreibt den Betrieb eines MFS, bei dem ein Stückgut an einem einzelnen MFM ankommt und dieses eine Entscheidung bezüglich des weiteren Weges für dieses Stückgut treffen muss. Dabei wird davon ausgegangen, dass das Ziel des Stückgutes, beispielsweise mittels eines Transponders, von der Hardwareansteuerung ausgelesen wird. Die Softwarekomponente Hardwareansteuerung fragt bei der Wegfindung die entsprechenden Stellinformationen an. Diese fragt ihrerseits die aktuellen Ausgangsast-Nachfolgertabellen bei der Topographiebildung an und berechnet damit zusammen mit der Zielinformation den Ast. Damit ruft die Funktion *Wegberechnung* die Funktion *Zuordnung* auf, die die Stellinformationen zurückliefert. Diese Stellinformationen leitet die Softwarekomponente Wegfindung an die Hardwareansteuerung zurück. Die darunter liegende Komponente Plug-and-Play wird im normalen Betrieb nicht berührt. Bei dieser Darstellung wurde darauf verzichtet, eventuell gleichzeitig stattfindende Heartbeats Klasse 1, die bei dem MFM eingehen oder von ihm verschickt werden, darzustellen, da der gesamte dargestellte Ablauf innerhalb einer so kurzen Zeit stattfindet, dass die Heartbeats nicht berücksichtigt werden müssen.

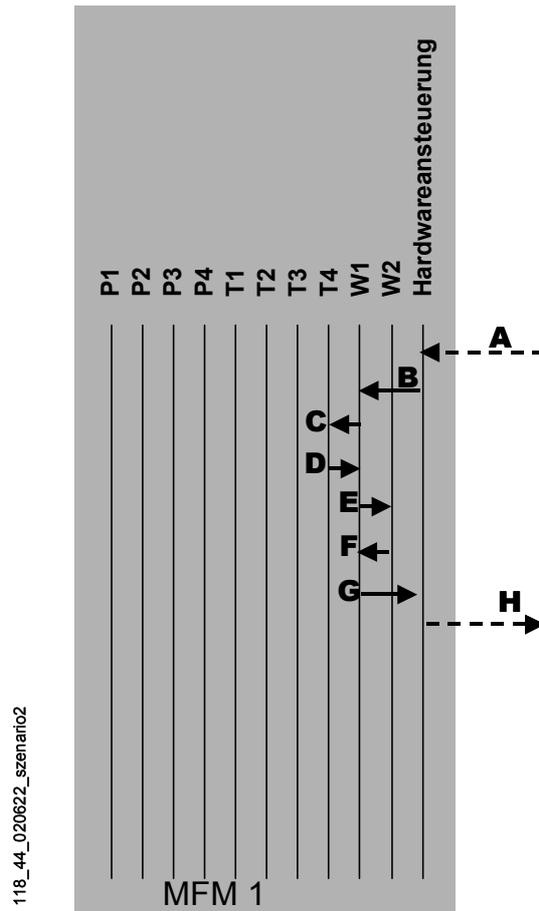


Abbildung 7-2: Sequenzdiagramm Szenario 2

Schritt	Auslöser	MFM	Beschreibung	Daten
A ¹¹⁷		1	Hardwareansteuerung: Liest Zielinformationen über Sensoren	Zielinformation wird gegeben
B	A	1	Hardwareansteuerung: Ruft W1 auf und übergibt Zielinformation	
C	B	1	W1: Ruft T4 auf	
D	C	1	T4: Gibt AN an W1 zurück	

¹¹⁷ Am MFM 1 kommt ein Werkstück an.

E	D	1	W1: Errechnet anhand der Ausgangsast-Nachfolgertabellen und der Zielinformation den Ast. Ruft W2 mit diesem Ast auf.	
F	E	1	W2: Liest anhand des Astes aus der Hardwaretabelle die zu schaltenden Ausgänge aus und gibt diese an W1 zurück.	
G	F	1	W1 gibt die zu schaltenden Ausgänge an die Hardwareansteuerung zurück.	
H	G	1	Hardwareansteuerung: Setzt die Ausgänge	

Tabelle 7-2: Erläuterungen zum Sequenzdiagramm des Szenarios 2

7.3. Hinzufügen eines MFM

Das *Hinzufügen eines MFM* in ein bestehendes und eventuell laufendes MFS stellt das Szenario eines Umbaus dar. Es ist dabei unerheblich, ob dieser Umbau tatsächlich während des Betriebs oder im Stillstand geschieht. Dieses Szenario zeichnet sich durch nicht konsistente Initialisierungsdaten in den einzelnen MFM bezüglich derer Vorgänger und Nachfolger aus, da das neu hinzuzufügende MFM nicht in den Initialisierungsdaten des bestehenden MFS enthalten sein kann.

Im folgenden Szenario sind drei MFM mit der Förderrichtung 1 - 2 - 3 dargestellt. Diese MFM laufen in der Sequenz 1 - 3 - 2 hoch, wobei 2 erst startet, wenn 1 und 3 vollständig initialisiert sind. Die Initialisierung von 1 und 3 erfolgt analog dem Szenario *Systemstart* und ist hier nicht abgebildet. Die inkonsistenten Initialisierungsdaten ergeben sich in diesem Beispiel dadurch, dass das MFM 1 mit dem MFM 3 als Nachfolger initialisiert ist und dies auch umgekehrt für das MFM 3 gilt. MFM 2 ist so initialisiert, dass es MFM 1 als Vorgänger und MFM 3 als Nachfolger hat. In diesem Punkt unterscheidet sich dieses Szenario von dem Szenario *Systemstart* und macht eine getrennte Behandlung notwendig, da das MFM 2 die Initialisierungsdaten von MFM 1 und 3 ändern muss.

Das neu zu dem MFS hinzugekommene MFM sendet an seine Vorgänger und Nachfolger die Message *Anmeldung im System*. Die Empfänger dieser Information ändern damit ihre eigenen Initialisierungsdaten, nachdem sie vorher den Datumstempel dieser Daten mit dem Datumstempel der eigenen Daten verglichen haben. Diese Verhaltensweise stellt die Adaptivität an Änderungen im Layout des MFS sicher.

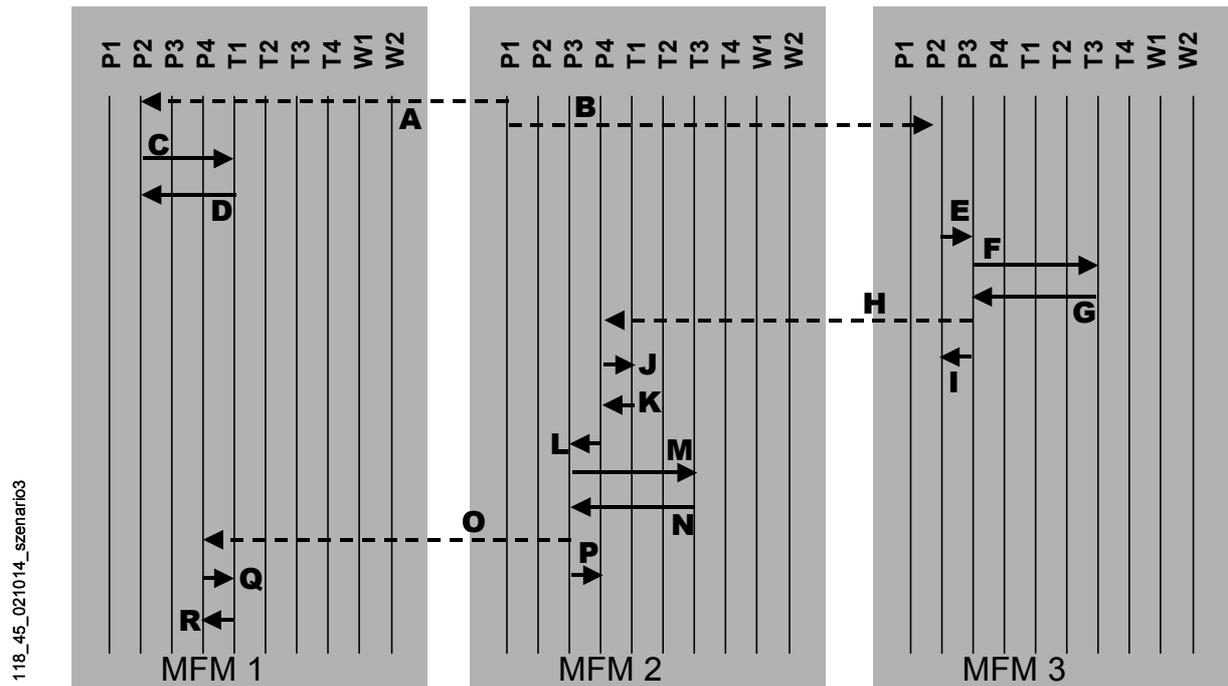


Abbildung 7-3: Sequenzdiagramm Szenario 3

Schritt	Auslöser	MFM	Beschreibung	Daten
A		2	P1: Anmeldung im System: Message Anmeldung_im_System von MFM 2 an MFM 1	
B		2	P1: Anmeldung im System: Message Anmeldung_im_System von MFM 2 an MFM 3	
C	A	1	P2: Entgegennahme Anmeldung im System: Ruft T1 auf	HB-Timer gesetzt Initialisierungs- daten geändert

D	C	1	T1: Legt AN ab. Kehrt zu P2 zurück.	
E	B	3	P2: Ruft P3 auf	HB-Timer gesetzt Initialisierungsdaten geändert
F	E	3	P3: Ruft T3 auf	
G	F	3	T3: EN wird erstellt und an P3 zurückgegeben	EN erstellt
H	G	3	P3: Message Heartbeat Klasse 2 an MFM 2	
I	H	3	P3: Kehrt zu P2 zurück.	
J	H	2	P4: Ruft T1 auf	HB-Timer gesetzt
K	J	2	T1: Legt AN ab. Kehrt zu P4 zurück	AN abgelegt
L	K	2	P4: Ruft P3 Heartbeat Klasse 2 auf	
M	L	2	P3: Ruft T3 auf	
N	M	2	T3: Erstellt EN und gibt diese an P3 zurück	EN erstellt
O	N	2	P3: Message Heartbeat Klasse 2 an MFM 1	
P	O	2	P3: Kehrt zu P2 zurück.	
Q	O	1	P4: Ruft T1 auf	HB-Timer gesetzt
R	Q	1	T1: Legt AN ab, kehrt zu P4 zurück ¹¹⁸	AN abgelegt

Tabelle 7-3: Erläuterungen zum Sequenzdiagramm des Szenarios 3

7.4. Ausfall eines MFM

Der *Ausfall eines MFM* kann durch den Umbau des MFS verursacht oder durch einen tatsächlichen Ausfall eines MFM begründet sein. Die Ursache des Ausfalls ist für das MFS unerheblich, das MFM steht nicht mehr zur Verfügung und seine Daten müssen gelöscht werden, um das MFS der neuen Situation anzupassen. Das nachfolgende Szenario besteht aus drei MFM, die bereits eine Konfiguration des Systems durchgeführt haben. Die Förderrichtung der drei MFM ist 1 - 2 - 3. MFM drei fällt nach

¹¹⁸ Keine weiteren Aktionen, da kein Vorgänger vorhanden ist.

erfolgter Konfiguration zu einem nicht näher spezifizierten Zeitpunkt aus, dies wird von MFM zwei aufgrund des Ausbleibens des Heartbeats erkannt. Dieses MFM löscht daraufhin die Ausgangsast-Nachfolgertabellen und verschickt einen Heartbeat Klasse 3 an seine Vorgänger. Dieser löscht daraufhin ebenfalls seine Ausgangsast-Nachfolgertabellen. Beide MFM müssen nun ihre Ausgangsastnachfolger-Tabellen wieder aufbauen, dies geschieht analog dem Szenario *Systemstart* und soll deshalb hier nicht weiter ausgeführt werden.

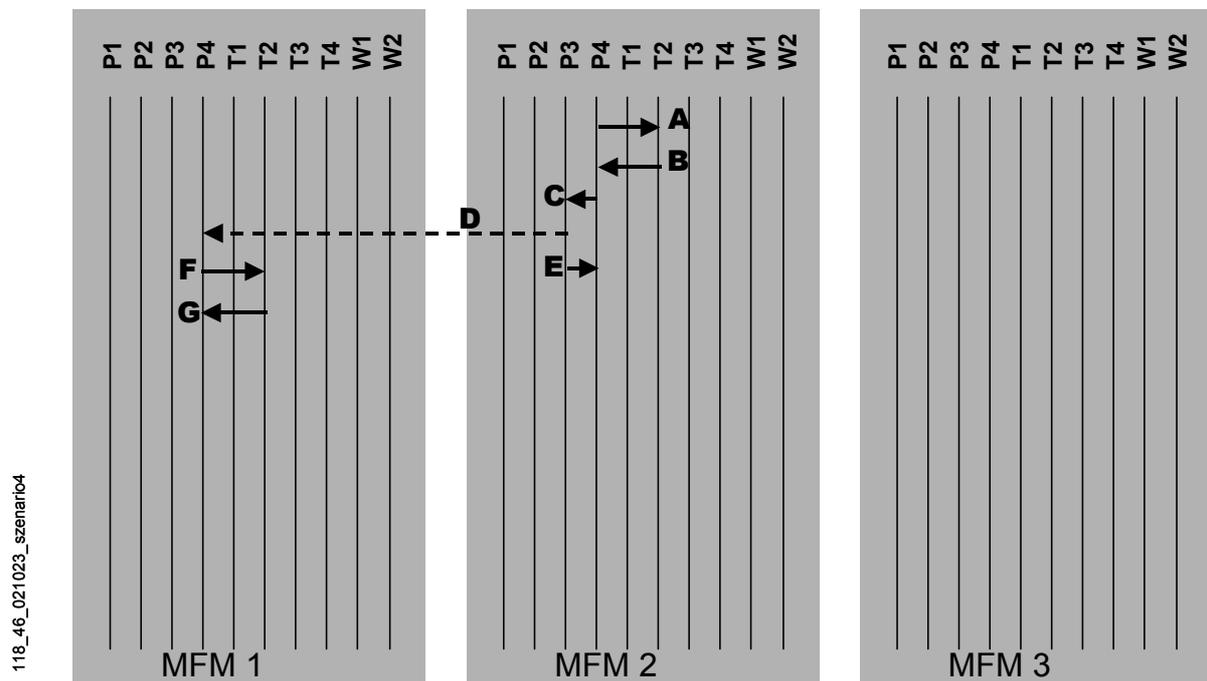


Abbildung 7-4: Sequenzdiagramm Szenario 4

Schritt	Auslöser	MFM	Beschreibung	Daten
A ¹¹⁹		2	P4: Ruft T2 auf	

¹¹⁹ P4 wird durch den Ablauf des Heartbeat-Timers und das Ausbleiben des entsprechenden Heartbeats von MFM 3 gestartet.

B	A	2	T2: Löscht alle AN. Kehrt zu P4 zurück	AN gelöscht
C	B	2	P4: Ruft P3 auf	HB-Timer gesetzt, Init-daten geändert
D	C	2	P3: Message Heartbeat Klasse 3 an MFM 1	
E	D	2	P3: Kehrt zu P4 zurück.	
F	E	1	P4: Ruft T2 auf	HB-Timer gesetzt
G	F	1	T2: Löscht alle AN. Kehrt zu P4 zurück. ¹²⁰	

Tabelle 7-4: Erläuterungen zum Sequenzdiagramm des Szenarios 4

7.5. Heartbeat zu spät

Dieses Szenario beschreibt den Fall, dass ein MFM nicht in der Lage ist, in der vorgesehenen Zeit seinen Heartbeat an den Vorgänger zu schicken, dies aber nicht aufgrund eines Ausfalls des MFM geschieht¹²¹. In diesem Fall hat der Vorgänger bereits die Daten des MFM gelöscht (vgl. dazu das Szenario *Ausfall eines MFM*), erhält dann jedoch (zu spät) den Heartbeat des MFM.

Ebenso wie das vorangegangene Szenario, beinhaltet dieses drei MFM, die bereits eine Konfiguration des Systems durchgeführt haben. Die Förderrichtung der drei MFM ist 1 - 2 - 3. MFM drei war nicht in der Lage, seinen Heartbeat Klasse 1 pünktlich an MFM 2 zu verschicken und ist von MFM 2 entsprechend dem Szenario *Ausfall eines MFM* gestrichen worden. Nachdem MFM 3 gestrichen ist, geht bei MFM 2 der Heartbeat Klasse 1 zu spät ein. In diesem Fall schickt MFM 2 die Message *Neuanmeldung im System* an MFM 3, die dieses dann mit einem Heartbeat Klasse 2 beantwortet. Damit wird MFM 2 wieder (die bereits gelöschte) Eingangsst-Nachfolgertabelle von MFM 3 mitgeteilt. Darauf hin

¹²⁰ Keine weiteren Aktionen, da kein Vorgänger vorhanden ist.

¹²¹ Grund dafür könnte beispielsweise ein Ausfall oder eine Überlastung des Bussystems sein.

ändert MFM 2 seine eigene Eingangsast-Nachfolgertabelle und teilt dies mittels eines Heartbeat Klasse 2 MFM 1 mit.

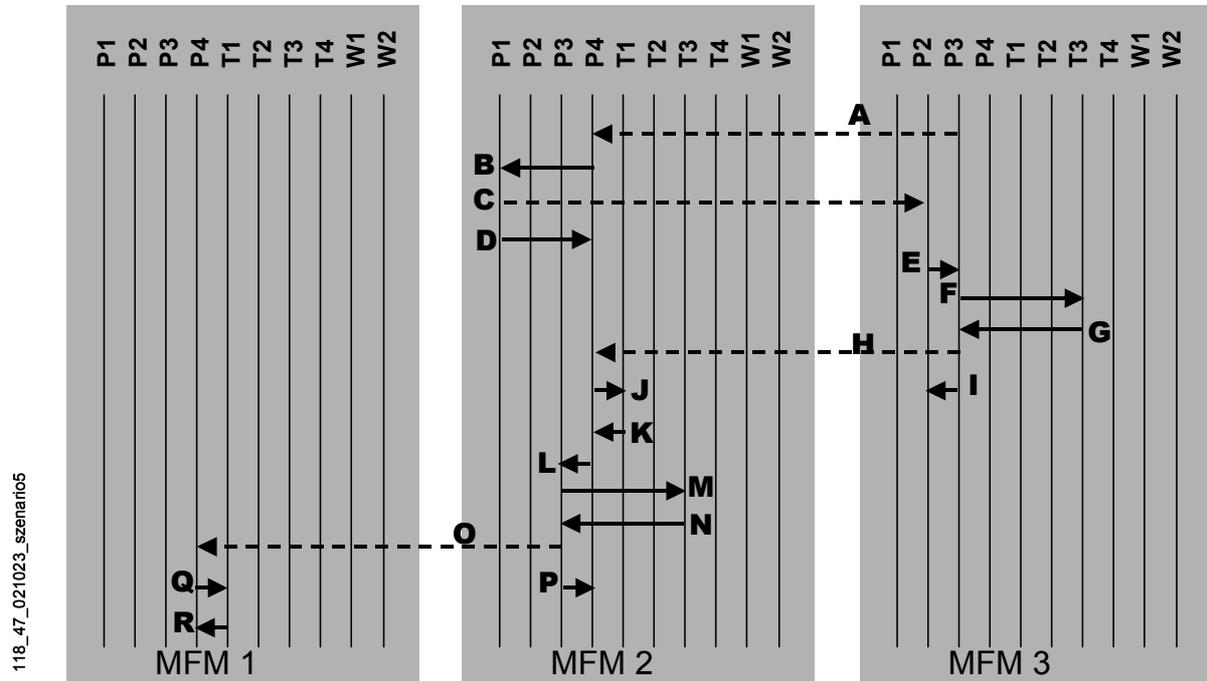


Abbildung 7-5: Sequenzdiagramm Szenario 5

Schritt	Auslöser	MFM	Beschreibung	Daten
A ¹²²		3	P3: Message Heartbeat Klasse 1 an MFM 2	
B	A	2	P4: Aufruf von P1 unter Angabe von MFM 3	HB-Timer gesetzt
C	B	2	P1: Anmeldung_im_System von MFM 2 an MFM 3	
D	C	2	P1: Kehrt zu P4 zurück.	

¹²² Der Heartbeat-Timer in MFM 3 ist abgelaufen und MFM 3 verschickt entsprechend einen Heartbeat Klasse 1.

E	C	3	P2: Ruft P3 auf	HB-Timer gesetzt
F	E	3	P3: Verschicken Heartbeats: Ruft T3 auf	
G	F	3	T3: Gibt EN an P3 zurück	EN erstellt
H	G	3	P3: Message Heartbeat Klasse 2 an MFM 2	
I	H	3	P3: Kehrt zu P2 zurück.	
J	H	2	P4: Entgegennahme Heartbeats: Ruft T1 auf	HB-Timer gesetzt
K	J	2	T1: Legt die AN ab. Kehrt zu P4 zurück	AN abgelegt
L	K	2	P4: Ruft P3 auf	
M	L	2	P3: Ruft T3 auf	
N	M	2	T3: Erstellt EN & gibt diese an P3 zurück.	EN erstellt
O	N	2	P3: Verschickt Heartbeat Klasse2 an MFM 1	HB-Timer gesetzt
P	O	2	P3: Kehrt zu P4 zurück.	
Q	O	1	P4: Entgegennahme Heartbeats: Ruft T1 auf	AN abgelegt
R	Q	1	T1: Legt die AN. Kehrt zu P4 zurück.	AN abgelegt

Tabelle 7-5: Erläuterungen zum Sequenzdiagramm des Szenarios 5

8. Adaptives Materialflusssystem

Für den Nachweis der praktischen Anwendbarkeit soll die entwickelte Steuerungsmethode für eine verteilte und adaptive Materialflusssteuerung innerhalb eines Materialflusssystems vom Typ Bosch TS2¹²³ umgesetzt werden. Durch seinen Aufbau mit mehreren Ausschleusestationen (die Bearbeitungsplätze darstellen) und seinem Trägerumlaufsystem entspricht es den in der industriellen Praxis zum Einsatz kommenden MFS und kann somit als Referenzplattform dienen. Das nachfolgende Bild zeigt das Referenz-MFS:



Abbildung 8-1: Referenz-MFS

¹²³ Bei diesem MFS handelt es sich um den Vorgänger des im Stand der Technik angerissenen MTS von Bosch. Das TS2 zeichnete sich generell durch ein zentrales Steuerungskonzept und nicht auf Modularisierung ausgelegte Komponenten aus.

Ursprünglich wurde dieses MFS mittels einer einzigen Speicherprogrammierbaren-Steuerung (SPS) gesteuert und stellte damit die Implementierung eines vollständig zentralistischen Steuerungskonzeptes dar. Im Rahmen der Umsetzung wurde das MFS komplett auf das in dieser Arbeit entwickelte verteilte Steuerungskonzept umgestellt. Die zentrale Steuerung wurde entfernt, die Verdrahtung entflochten und in dezentralen Schaltkästen aufgelegt, die jeweils eine Steuerung für ein Modul aufnehmen.

Im ersten Schritt der Umsetzung erfolgt eine Simulation der Steuerungsmethode, angewandt auf das existierende MFS. Diese Simulation wird auf einem Standard-PC durchgeführt, die einzelnen Steuerungsprogramme werden durch entsprechende Java-Implementierungen simuliert. Durch diese Simulation wird der Beweis der generellen Funktion der Steuerungsmethode unter verschiedenen Betriebsbedingungen erbracht. Im zweiten Schritt erfolgt die Umsetzung in das umgebaute MFS durch Programmierung von verteilten Speicherprogrammierbaren-Steuerungen. Aufgrund der - mit einer Umsetzung in SPSen verbundenen - erhöhten Anforderungen an die Leichtgewichtigkeit der entwickelten Steuerungsmethode, kann damit der Nachweis der Praxistauglichkeit erbracht werden.

8.1. Beschreibung des Referenz-MFS

Das Referenz-MFS besteht aus Standardkomponenten der Baureihe TS2 der Firma Bosch. Diese Baureihe basiert auf der Stauförderung mittels Doppelgurtbändern. Zwei stetig angetriebene, parallel verlaufende Gurtbänder fördern das auf Werkstückträgern befindliche Transportgut. An Verzweigungen oder Bearbeitungsstationen wird der Werkstückträger mittels Stoppern angehalten, wobei die Gurtbänder unter dem Werkstückträger weiter laufen¹²⁴.

¹²⁴ Diese Technologie kommt üblicherweise in der Montage-Automatisierung mit Bauteilen kleinerer Gewichte zum Einsatz.

Das genutzte MFS ist als ein geschlossener Kreislauf¹²⁵ mit drei Ausschleusstrecken¹²⁶ (MFM 6, 7, 8) und einer internen Umgehung (MFM 5) aufgebaut. Die Aufteilung des MFS in MFM und die Förderrichtung sind der Abbildung 8-2 zu entnehmen. Die drei Ausschleusstrecken können das Fördergut bidirektional fördern. Weiterhin kann an dem MFM 1 ein Werkstückträger eingeschleust und an dem MFM 3 wieder ausgeschleust werden¹²⁷:

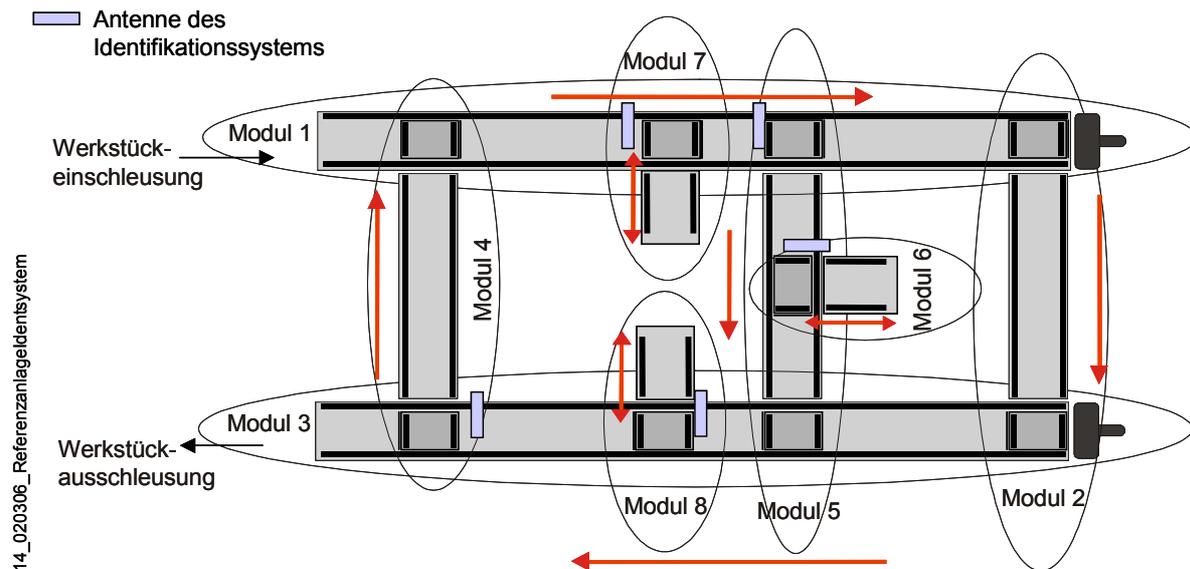


Abbildung 8-2: Aufteilung des Referenz-MFS in MFM

In der Darstellung sind die acht MFM gekennzeichnet, in die das bisher monolithische System aufgeteilt wurde. Jedes dieser MFM ist mit einer eigenen Speicherprogrammierbaren-Steuerung versehen und verfügt deshalb über einen eigenen kleinen Schaltschrank. In einem ersten Schritt wurden alle acht Module mit Steuerungen der Firma Saia-Burgess vom Typ SAIA-Burgess

¹²⁵ Dieser Aufbau ist typisch für die Verwendung von Werkstückträgern und dient der Rückführung der Werkstückträger.

¹²⁶ Diese Ausschleusstrecken stellen jeweils Bearbeitungsstationen dar.

¹²⁷ In der Darstellung sind ebenfalls die vorhandenen Antennen für die Transponder gekennzeichnet.

PCD2.M150¹²⁸ ausgerüstet. Diese können - ausgestattet mit einem intelligenten Co-Prozessormodul für die Kommunikation - Daten durchgängig vom standardisierten Feldbus bis zum Ethernet austauschen. Dieses Kommunikationsmodul wird auf den entsprechenden Steckplatz des jeweiligen Prozessormoduls gesteckt. Über RJ45-Stecker (Kategorie 5) kann ein Ethernetkabel an die Steuerung angeschlossen werden. Damit stehen die Transport-Protokolle UDP/IP bzw. TCP/IP mit den Übertragungsraten von 10 bzw. 100 MBit/s mit Autosensing (Ethernet 10 Base-T / 100 Base-TX) zur Verfügung /SAIA 01/. Alle Steuerungen der MFM sind mittels dieser Ethernet-TCP/IP-Verbindung miteinander vernetzt. Die Netzwerkarchitektur entspricht der üblicherweise bei Base-T Netzwerken zum Einsatz kommenden Sternarchitektur. Als Infrastrukturkomponente des Netzwerks kommt ein 12-Port Switch¹²⁹ zum Einsatz, an den alle Steuerungen und der PC für die Programmierung angeschlossen sind. Die grundsätzliche Architektur dieses Aufbaus ist in dem folgenden Bild dargestellt.

¹²⁸ Diese homogene Architektur kann später auch durch den Ersatz einiger Steuerungen durch andere in eine heterogene Architektur überführt werden.

¹²⁹ Ein Switch ist ein Infrastrukturelement auf der Schicht 2 des OSI 7-Schichten Modells /Seifert 00/.

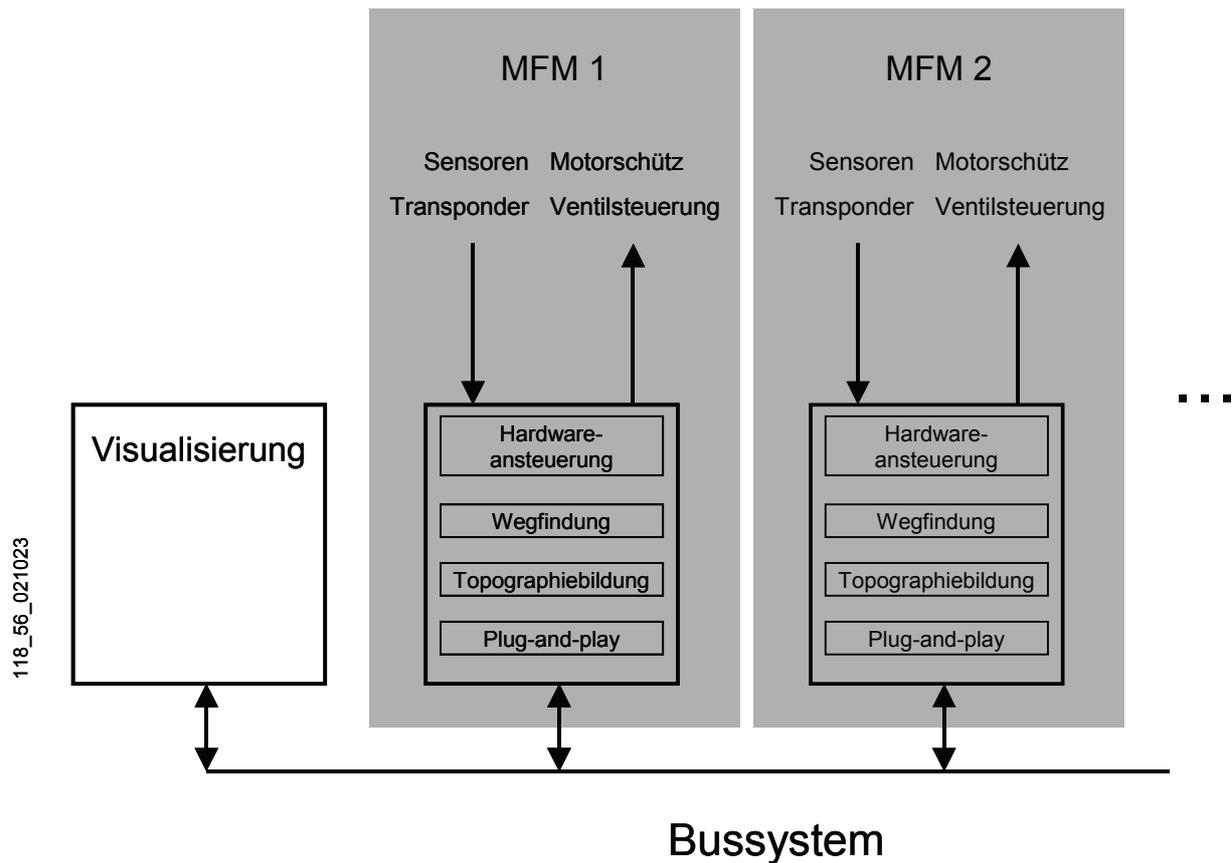


Abbildung 8-3: Steuerungstechnik des MFS

Die umlaufenden Werkstückträger verfügen jeweils über einen Multipage-Transponder der Firma Hermos. Dabei handelt es sich um einen Transponder, der auf Mikrowellen-Basis arbeitet und deshalb keine eigene Stromversorgung benötigt¹³⁰. Der Multipage-Transponder hat maximal 136 Byte Datenspeicher in 17 Seiten mit jeweils 8 Byte. Diese Daten können von den in der Anlage befindlichen Lese- und Schreibstationen sowohl gelesen als auch beschrieben werden /Hermos 01/. An allen fünf Stellen, an denen Verzweigungen möglich sind, befinden sich jeweils Stationen zum Lesen und Schreiben dieser Transponder. Auf den Transpondern wird die dem Werkstückträger zugeordnete Zielinformation gespeichert.

Somit entsprechen das Referenz-MFS und seine acht MFM vollständig den in Kapitel 5.1 eingeführten Prämissen und sind damit zum Nachweis der Funktion der entwickelten Methode geeignet.

8.2. Modellbildung

Die Bildung der MFM des Referenz-MFS orientiert sich an den einzelnen Doppelgurtbändern. Jedes dieser Bänder wird von einem Motor angetrieben und wurde deshalb innerhalb eines MFM abgebildet. Für jedes einzelne MFM x existiert ein Graph T^x , der dieses MFM losgelöst von dem Gesamtsystem beschreibt. Bei der Bezeichnung der logischen Knoten wurden die MFM-Nummern beibehalten. Die Bezeichnung der Eingangs- und Ausgangsäste erfolgte jeweils im Uhrzeigersinn, startend bei 12:00 Uhr:

$$T^1 = \{ \{E_1^1, E_2^1, E_3^1, A_1^1, A_2^1, A_3^1, K^1\}, \{e_1^1, e_2^1, e_3^1, a_1^1, a_2^1, a_3^1\} \}$$

$$T^2 = \{ \{E_1^2, A_1^2, K^2\}, \{e_1^2, a_1^2\} \}$$

$$T^3 = \{ \{E_1^3, E_2^3, E_3^3, A_1^3, A_2^3, A_3^3, K^3\}, \{e_1^3, e_2^3, e_3^3, a_1^3, a_2^3, a_3^3\} \}$$

$$T^4 = \{ \{E_1^4, A_1^4, K^4\}, \{e_1^4, a_1^4\} \}$$

$$T^5 = \{ \{E_1^5, E_2^5, A_1^5, A_2^5, K^5\}, \{e_1^5, e_2^5, a_1^5, a_2^5\} \}$$

$$T^6 = \{ \{E_1^6, A_1^6, K^6\}, \{e_1^6, a_1^6\} \}$$

$$T^7 = \{ \{E_1^7, A_1^7, K^7\}, \{e_1^7, a_1^7\} \}$$

$$T^8 = \{ \{E_1^8, A_1^8, K^8\}, \{e_1^8, a_1^8\} \}$$

Die für jedes einzelne MFM benötigten Initialisierungsdaten ergeben sich aus diesen Graphen und der Zuordnung der einzelnen

¹³⁰ Die Lesestation sendet über ihre Antenne einen Energieimpuls aus, wodurch sich ein Kondensator in dem Transponder auflädt, der dann die Energie für die Rückübermittlung der Daten zur Verfügung stellen kann.

Äste e_n^x , a_m^x zu den jeweiligen realen Steuerungsfunktionen des MFM x , was innerhalb der Hardwaretabelle erfolgt.

Durch Anwendung der in Kapitel 5.3 eingeführten Regeln lassen sich diese Einzelgraphen T^x in einen, das gesamte MFS beschreibenden Graphen T überführen:

$$T = \{ \{ E_2^1, A_1^1, A_2^1, A_3^1, A_1^2, A_1^3, A_2^3, A_3^3, A_1^4, A_1^5, A_2^5, A_1^6, A_1^7, A_1^8, K^1, K^2, K^3, K^4, K^5, K^6, K^7, K^8 \}, \{ e_1^1, e_2^1, e_3^1, e_1^2, e_1^3, e_2^3, e_3^3, e_1^4, e_1^5, e_2^5, e_1^6, e_1^7, e_1^8, a_1^1, a_2^1, a_3^1, a_1^2, a_1^3, a_2^3, a_3^3, a_1^4, a_1^5, a_2^5, a_1^6, a_1^7, a_1^8 \} \}$$

Die folgende Abbildung stellt T dar:

118_66_020705_Modellierung_Referenzanlage

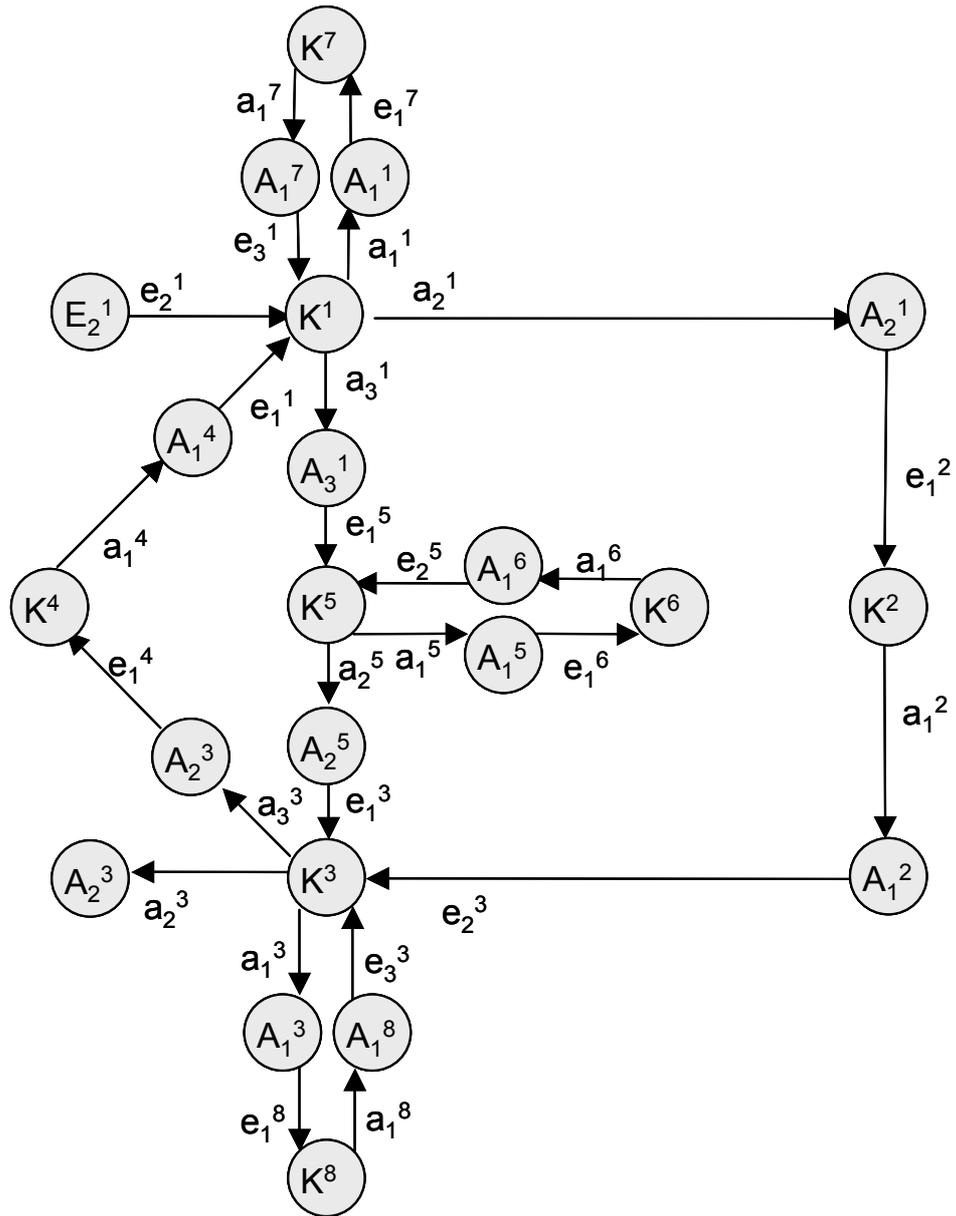


Abbildung 8-4: Graph des Referenz-MFS

Anhand dieser Modellierung erfolgt die Parametrierung sowohl der Simulation als auch der realen MFM im weiteren Verlauf der Arbeit.

8.3. Simulation

Als erster Schritt der Umsetzung wurde im Rahmen einer Simulation¹³¹ die Funktion der Methode getestet und nachgewiesen. Dabei erfolgte die Simulation unter Zugrundelegung des in Kapitel 8.1 beschriebenen MFS. Die Implementierung der Simulation erfolgte in Java¹³² auf einem handelsüblichen PC. Eine Java Klasse *MFM*, die alle drei beschriebenen Softwarekomponenten mit ihren Funktionen enthält, bildet die Basis dieser Simulation. Diese Klasse stellt bereits eine erste Umsetzung der in dieser Arbeit entwickelten Methode dar und könnte auf einer Steuerung, die Java implementiert, zur Wegfindung eingesetzt werden. Jedes einzelne MFM des in Kapitel 8.1 beschriebenen MFS wird im Rahmen der Simulation als ein Objekt instanziiert, das in einem eigenen Thread läuft¹³³. Diese Objekte kommunizieren über ein *Messenger* Objekt, das das Kommunikationsnetzwerk simuliert und das in einer realen, Java-basierten Anwendung nicht notwendig wäre. Weiterhin wird ein *Transportobjekt* erzeugt, das ein zu transportierendes Transportgut darstellt. Diese Objekte sind jeweils mit einer graphischen Benutzerschnittstelle versehen, um eine Überwachung und Parametrierung der Objekte im Verlauf der Simulation zu ermöglichen. Nachfolgend sollen diese Schnittstellen und ihre Funktionen kurz dargestellt werden:

Für jedes instanziierte MFM existiert ein spezielles Fenster, das dessen aktuelle Initialisierungsdaten, seine Zentralknoten-

¹³¹ Lat. Die Vorspiegelung. Unter einer Simulation versteht man die modellhafte Darstellung oder Nachbildung bestimmter Aspekte eines vorhandenen Systems /Brockhaus 98/.

¹³² Auf den Einsatz bestehender Simulationsumgebungen wie z.B. Simulink (vgl. Hoffmann /Hoffmann 99/) wurde verzichtet, da diese einerseits keine Reduktion des Aufwandes für die Simulation bedeutet hätten, andererseits auch nicht für die gleichzeitige Simulation von sowohl Protokollen als auch Algorithmen in verteilten Systemen ausgelegt sind.

¹³³ Dieser Aufbau erlaubt einen quasi-parallelen Betrieb der verschiedenen simulierten Knoten und erlaubt damit eine Simulation der „Verteiltheit“ des Systems.

Nachfolgertabelle¹³⁴ mit Kosten und die aktuell eingegangene Message anzeigt. Außerdem kann jedes MFM über einen Schalter in diesem Fenster vom Bediener an- und ausgeschaltet werden¹³⁵ und der aktuelle Zustand abgelesen werden. Dieses Fenster ist in der folgenden Abbildung zu sehen:

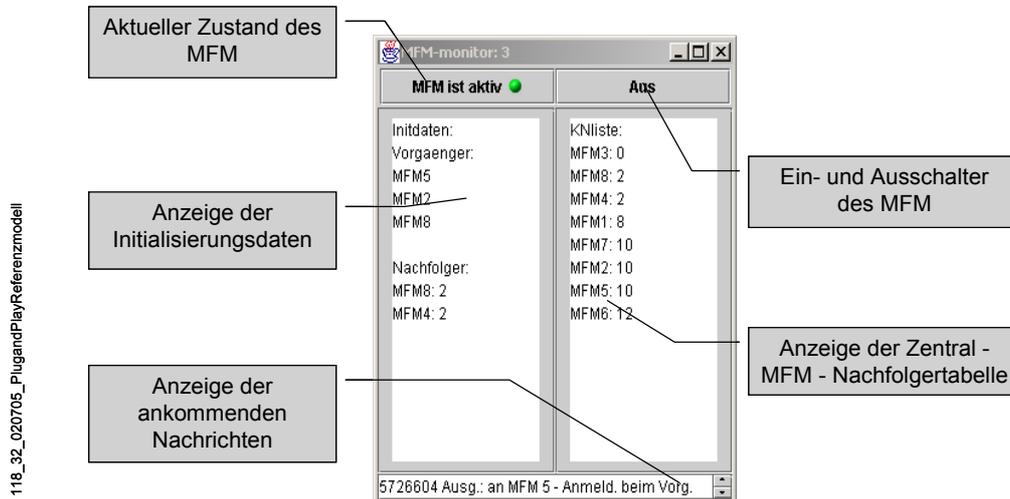


Abbildung 8-5: Modulmonitor-Fenster

Für unterschiedliche Simulationsläufe kann ein Transportobjekt in einem eigenen Fenster erzeugt werden, wobei es möglich ist, eine Abfolge von Zielen einzugeben und diese Ziele noch mit Alternativzielen zu hinterlegen. In diesem Fenster wird das simulierte Transportgut auch gestartet:

¹³⁴ Die Darstellung der Zentralknoten-Nachfolgerliste erfolgte, um die Darstellung auf dem Bildschirm zu vereinfachen, da die ZN die Aggregation der AN ist.

¹³⁵ Dies erlaubt die Simulation verschiedenster Reihenfolgen beim Hochfahren des Systems oder von Teilausfällen. Im realen MFM kann dies durch Abklemmen der Stromversorgung erfolgen.

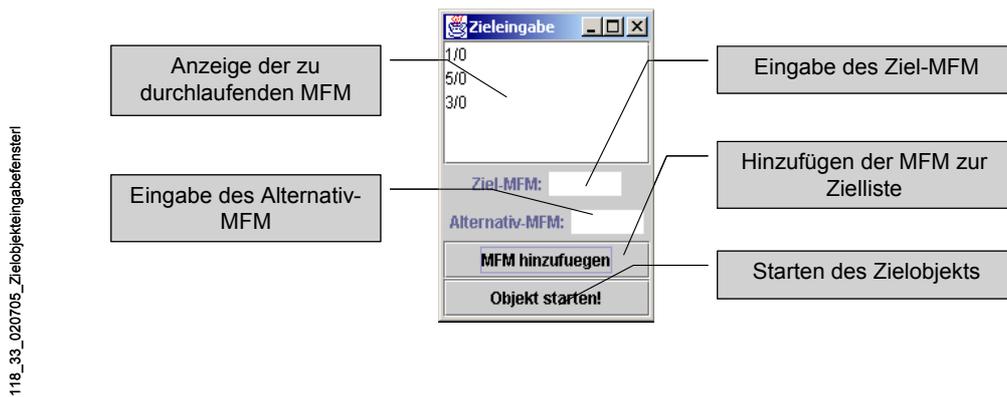


Abbildung 8-6: Zieleingabe-Fenster

In einem weiteren Fenster kann der Verlauf des Transportobjektes durch das simulierte MFS am Bildschirm online verfolgt werden:

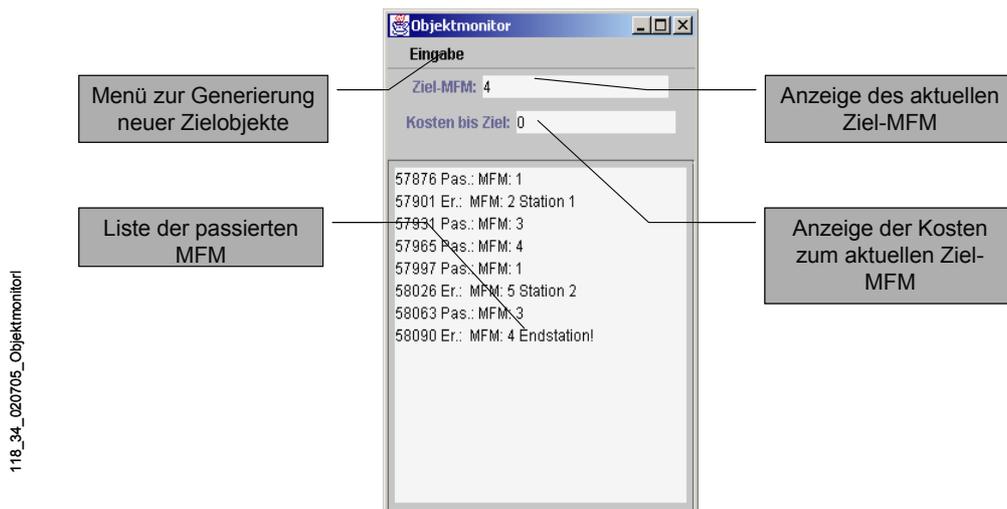
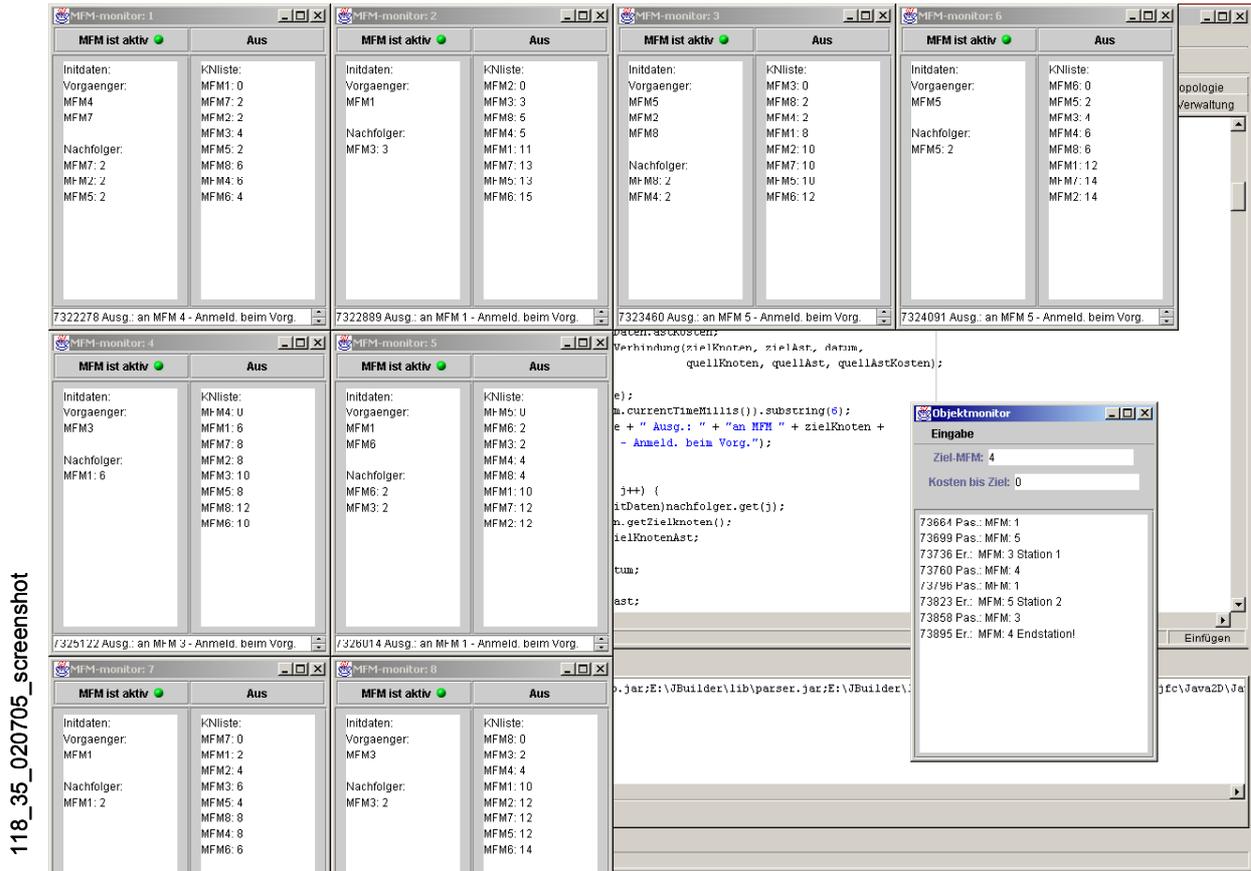


Abbildung 8-7: Transportobjektmonitor-Fenster

Diese Fenster lassen sich gleichzeitig auf einem PC darstellen und ermöglichen die Überwachung des Ablaufs sowohl in den einzelnen MFM als auch im Transportgut. Das nachfolgende Bild zeigt die Bildschirmdarstellung des in Anhang A (Protokoll 1) protokollierten Simulationslaufes:



118_35_020705_screenshot

Abbildung 8-8: Screenshot Simulationslauf

Mittels dieser Darstellung lässt sich das Verhalten des gesamten MFS übergreifend betrachten und analysieren. Basierend auf dieser Implementierung wurde das in Abbildung 8-4 dargestellte Modell des Referenz-MFS simuliert. Dazu wurden acht Instanzen der allgemeinen Klasse *MFM* mit den entsprechenden Initialisierungsdaten erzeugt. In Abbildung 8-8 kann man die acht Fenster der Modulmonitore¹³⁶, die jeweils ein MFM des Modells repräsentieren, erkennen. Abhängig von den zu simulierenden Szenarien wurden verschiedene *Transportobjekte* mit unterschiedlichen Zielen erzeugt. Das Transportobjektmonitor-Fenster ist in der Abbildung 8-8 rechts am Rand zu sehen. Zum Nachweis der Robustheit,

¹³⁶ Wie aus der Abbildung zu erkennen ist, sind alle simulierten MFM aktiv, und das Gesamtsystem ist konfiguriert.

Selbstkonfiguration und Adaptivität der entwickelten Steuerungsmethode wurden verschiedene Simulationsläufe, die nur die MFM ohne ein Transportgut betreffen, durchgeführt. Verschiedene Einschaltreihenfolgen der simulierten MFM wurden für das Referenz-MFS mit konsistenten und nicht konsistenten Initialisierungsdaten in mehreren Läufen simuliert. Bei diesen Simulationsläufen konnte der zu erwartende Aufbau der Nachfolgertabellen aller MFM beobachtet werden und damit der Nachweis der Selbstkonfiguration und Robustheit erbracht werden. Bei der Simulation nicht konsistenter Initialisierungsdaten konnte darüber hinaus noch die entsprechende Korrektur der Initialisierungsdaten in den MFM Daten beobachtet werden. Außerdem wurden in verschiedenen Simulationsläufen die Ausfälle von MFM innerhalb des Referenz-MFS und deren anschließendes Hochfahren getestet. Im Falle des Ausfalles von MFM wurden die Nachfolgerlisten der übrigen MFM richtig gelöscht und anschließend wieder richtig aufgebaut. Nach dem Hochfahren von ausgefallenen MFM wurden die Nachfolgertabellen ebenfalls richtig aktualisiert. In allen Fällen waren die simulierten MFM in der Lage, die entsprechenden Initialisierungsdaten und die Nachfolgertabellen entsprechend den Erwartungen richtig nachzuführen. Die Ergebnisse dieser Simulationsläufe sind im Anhang B in den Protokollen 1 und 2 zusammenfassend dargestellt.

Darüber hinaus wurden für den normalen Betrieb (also den Transport von Gütern) verschiedene Simulationsläufe, die dem Nachweis der Selbstoptimierung dienten, durchgeführt. Für das Referenz-MFS wurden verschiedene Transportobjekte erstellt, die unterschiedliche Routen beinhalteten, dabei fuhr das Transportobjekt in jedem Simulationslauf die korrekten Ziele an. Im Rahmen weiterer Simulationsläufe wurden die Kosten der simulierten MFM 2 und 5 so geändert, dass eine Wegoptimierung entweder über das eine oder das andere MFM führte. Die Beobachtungen ergaben, dass das Transportobjekt immer über den kostengünstigeren Weg geleitet wurde, was den Nachweis der Wegoptimierung der Steuerungsmethode erbrachte. In abschließenden Simulationsläufen wurde getestet, wie sich ein Transportobjekt verhält, wenn ein Teil des MFS, während es noch

unterwegs ist, ausfällt. Dabei konnte beobachtet werden, dass das Transportobjekt in diesen Fällen entweder seine Ausweichziele anfuhr, oder, wenn das aufgefallene MFM wieder hochgefahren war, seinen Weg fortsetzte, was ein zusätzlicher Nachweis der Robustheit der Steuerungsmethode ist. Auch für diese Szenarien waren die Simulierten MFM vollständig in der Lage, alle Wege korrekt zu berechnen. Eine Dokumentation der Ergebnisse erfolgt in Anhang B, Protokoll 3.

Abschließend kann festgestellt werden, dass die Simulationsläufe die grundlegende Funktionsfähigkeit der Steuerungsmethode nachgewiesen haben. Die simulierten MFM des Referenz-MFS waren innerhalb weniger Sekunden in der Lage, sich an Ausfälle oder Neustarts einzelner MFM entsprechend anzupassen. Diese Anpassungen erfolgen durch die Weiterreichung der Informationen von einem MFM zum nächsten im Rahmen der Heartbeats. Da in dem simulierten MFS die Heartbeat-Zeit eine Sekunde betrug, kann die Steuerungsmethode auch problemlos für um Zehnerpotenzen größere MFS genutzt werden, indem die Heartbeat-Zeit entsprechend herabgesetzt wird¹³⁷.

8.4. Umsetzung am Referenz-MFS

Die zuvor mittels der Simulationsläufe getestete Steuerungsmethode wird anschließend in dem eingangs beschriebenen Referenz-MFS umgesetzt. Die Programmierung wird dafür von dem in der Simulation verwendeten objektorientierten Java-Code in eine für die Speicherprogrammierbaren-Steuerungen (SPS) verständliche Sprache übertragen. Darüber hinaus wird für die Ansteuerung der realen Hardware die Softwarekomponente Hardwareansteuerung entsprechend der Hardware programmiert und um die Softwarekomponente Logistik erweitert. Im Gegensatz zu anderen SPSen arbeitet die PCD von SAIA nicht mit

¹³⁷ Die Busbelastungen kann dabei aufgrund der Größe der Nachrichten vernachlässigt werden.

Prozessabbildern¹³⁸ und grundsätzlich zyklisch, d.h. es werden alle Programm-Blöcke nacheinander in einer endlosen Schleife abgearbeitet. Für die Programmierung der PCD stehen deshalb verschiedene Arten von Programmblöcken zur Verfügung:

Die COBs (Cyclic Organisation Block) sind für dauernd abzuarbeitende Programmteile ohne jegliche Wartefunktionen vorgesehen und stellen das Grundgerüst der Programmierung dar. Es können bis zu 16 verschiedene COBs programmiert werden, die alle nacheinander in einem Zyklus abgearbeitet werden. Alle Prozess-Signale müssen innerhalb eines COB bearbeitet werden. Um bestimmte Programmteile mehrmals aufrufbar zu machen, stellt die PCD den Program Block (PB) und den Function Block (FB) zur Verfügung. PBs und FBs werden nur dann aktiv, wenn diese aufgerufen werden, wobei der Aufruf unbedingt oder bedingt erfolgen kann. Der FB bietet über den PB hinausgehend die Möglichkeit an, bei seinem Aufruf Parameter zu übergeben. Für die Verarbeitung genau definierter, nacheinander auftretender Ereignisse ist der Sequential Block (SB) vorgesehen. Ein SB wird entlang von STEPs, die entsprechende Ereignisse darstellen, programmiert. Dabei wird jeder SB aus einem COB aufgerufen. Ist das Ereignis, auf das der SB wartet, nicht erfüllt, kehrt das Programm zum aufrufenden COB zurück und arbeitet dort weiter. Andernfalls erfolgt die Bearbeitung des SB bis zu dem nächsten nicht erfüllten Ereignis.

Für jedes MFM werden alle in Kapitel 5.2 definierten Funktionen implementiert. Die Funktionen der Wegfindung und Topologiebildung wurden aufgrund des zyklischen Charakters ihrer Bearbeitung als FBs implementiert. Die Funktionen der Komponente Plug-and-Play wurden als SBs implementiert, da diese Funktionen azyklisch vom Netzwerk abhängig sind:

¹³⁸ Bei einer SPS mit Prozessabbild werden zu Beginn jedes Programmzyklus' der Status aller definierten Ein- und Ausgänge gelesen und abgelegt. Während des Zyklus werden die Ein- und Ausgänge nicht mehr behandelt, sondern es wird das Abbild verarbeitet. Am Ende des Zyklus' wird das aufgefrischte Abbild zu den Ausgängen übertragen. Für eine umfassende Einführung in die SPS-Technologie vgl. /Konhäuser 98/.

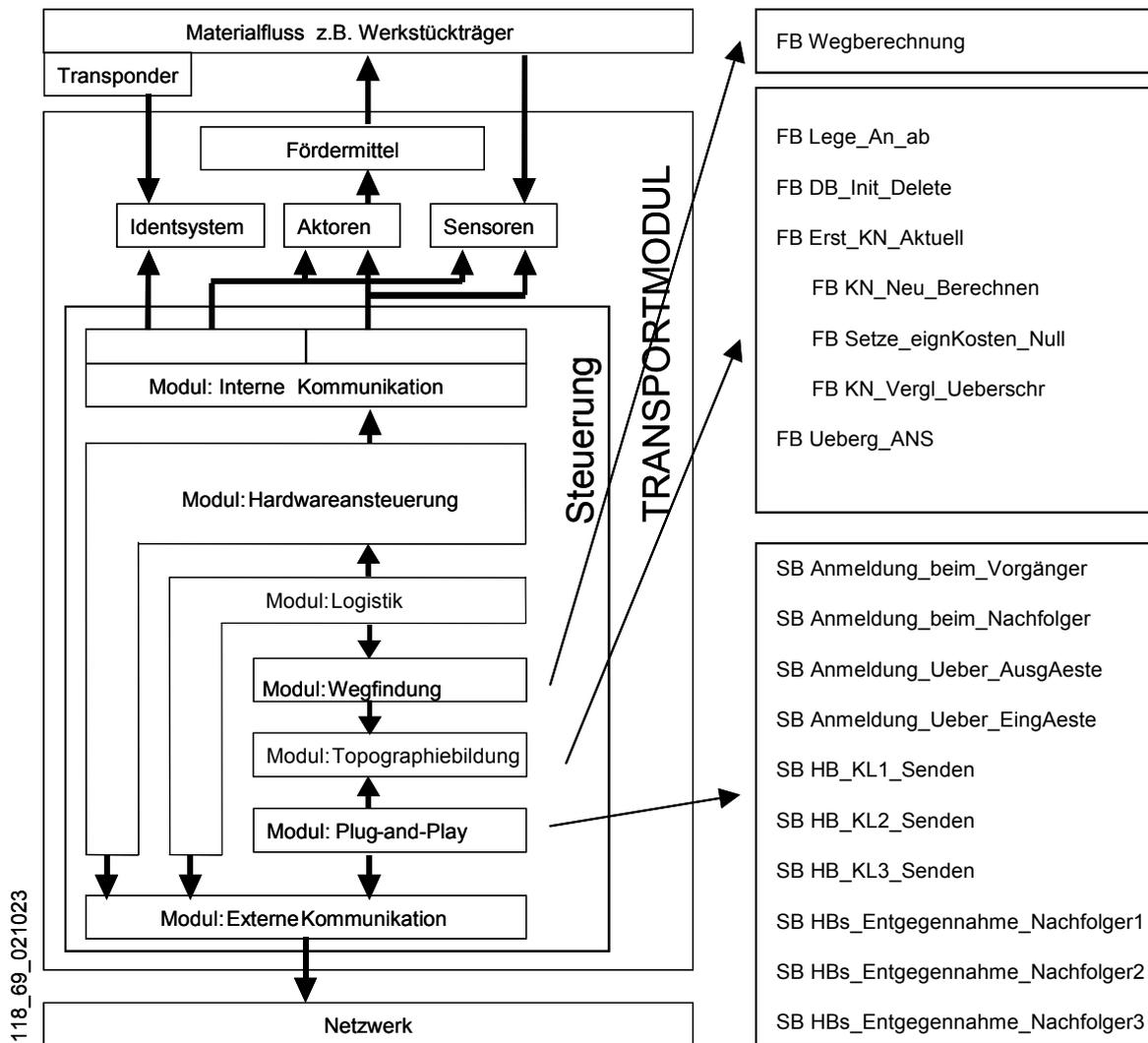


Abbildung 8-9: Implementierung der Softwarekomponenten innerhalb der SPS

Aufgrund der mechanischen Gegebenheiten des TS2 kann bei der Anwendung das allgemeine Modell auf jeweils drei mögliche Vorgänger-MFM und drei Nachfolger-MFM begrenzt werden. Um eine der Referenzimplementierung angemessene Flexibilität bezüglich der vorhandenen MFM zu gewährleisten, wurden die Eingangsast-Nachfolgertabellen, die zwischen den einzelnen MFM ausgetauscht werden, für 32 Module vorbereitet¹³⁹. Bei der Implementierung

¹³⁹ Dabei handelt es sich um 32 Bytes, wobei die Position des Bytes in der Liste die MFMnummer angibt und der Inhalt des Bytes die Kosten zu diesem MFM.

wurde - um Ressourcen zu sparen - Wert auf die Kürze der ausgetauschten Messages gelegt. Das Byte Null einer jeden Message definiert den Typen der Message wie folgt:

Wert des Byte 0	Message-Typ
1	Anmeldung bei Vorgänger
2	Anmeldung bei Nachfolger
3	Heartbeat Klasse 1
4	Heartbeat Klasse 2
5	Heartbeat Klasse 3

Tabelle 8-1: Typisierung der implementierten Messages

Auch bei der Implementierung der Datenhaltung wurde auf eine ressourcenarme Konzeption Wert gelegt, da Speicherprogrammierbare-Steuerungen im Verhältnis zu PCs über wenig Speicherplatz verfügen.

Ebenso wie bei den Simulationsläufen wurde bei der Umsetzung im Referenz-MFS in einem ersten Schritt die grundsätzliche Funktionsweise der Steuerungsmethode getestet. Dafür wurden die nach dem Hochlaufen des MFS in jedem MFM vorliegenden ZN ausgelesen. Diese Tabellen entsprachen jeweils den zu erwartenden Werten und erbrachten somit den Nachweis der Selbstkonfiguration und Adaptivität. Dies konnte für unterschiedliche Reihenfolgen des Hochfahrens nachgewiesen werden, in Anhang C befindet sich Tabelle 1 mit den ausgelesenen Werten für die ZN. Daran anschließend wurde getestet, ob die MFM in der Lage sind, den Ausfall eines MFMs zu erkennen und sich entsprechend neu zu konfigurieren. Dafür wurde ein beliebiges MFM per Hand ausgeschaltet. Vereinbarungsgemäß ist dabei für ein nicht erreichbares MFM der Wert 255 vorgesehen. Die in Anhang C befindliche Tabelle 2 belegt die Fähigkeit zur Selbstkonfiguration und Robustheit der Steuerungsmethode exemplarisch für den Ausfall des MFM 2. Abschließend wurde das Hochfahren eines ausgefallenen MFM erprobt und durch Auslesen der MFM-ZN überprüft. Auch dieser Test verlief erfolgreich, das MFS stellte sein Fähigkeit zur

Selbstkonfiguration auch dabei unter Beweis. Auf eine Darstellung des Ergebnisses im Anhang soll verzichtet werden, da die Werte den Werten der Tabelle 1 entsprechen.

Der normale Betrieb - und damit die grundsätzlich Funktionsweise der Steuerungsmethode im realen MFS - wurde durch verschiedene Transportaufträge getestet und erfolgreich abgeschlossen. Die Optimierungsfunktion der Steuerungsmethode wurde analog der Simulation anhand der Variation der Kosten in MFM 2 und 5 nachgewiesen. Im Rahmen des Nachweises wurde ein Transportgut mehrfach mit dem Ziel MFM 8 eingeschleust und die Kosten der MFM 2 und 5 so variiert, dass der (theoretisch) kostenoptimale Weg entweder über das MFM 2 oder über das MFM 5 führte. Die Förderung es Transportgutes erfolgte entlang des theoretisch zu erwartenden Weges, womit die Funktionsfähigkeit der Optimierung belegt ist.

Die Umsetzung innerhalb des realen MFS erbrachte den Nachweis der grundsätzlichen Umsetzbarkeit der Steuerungsmethode in ihrer vollen Funktionalität in SPSen. Nachdem die grundsätzliche Funktionsweise der Steuerungsmethode im Rahmen der Simulation nachgewiesen werden konnte, war dies der ausstehende Nachweis für ihre Leichtgewichtigkeit und damit Praxistauglichkeit. Das eingesetzte 10/100 Base-TX Ethernet Netzwerk erfüllte die für die Steuerung notwendigen Anforderungen ebenfalls. Am realen MFS konnte das Hochfahren in unterschiedlichen Reihenfolgen durch das kontrollierte Anschalten verschiedener Steuerungen per Hand erprobt werden. Außerdem wurden verschiedene Steuerungen ebenfalls per Hand ausgeschaltet (und wieder eingeschaltet), um den Ausfall eines MFM darzustellen. Ebenfalls konnte die Wegoptimierung getestet werden. Alle drei Punkte wurden am realen MFS - ebenso wie innerhalb der Simulationsläufe - vollständig erfüllt.

8.5. Auswertung

Das Ziel, ein MFS vollständig auf verteilte und adaptive Materialflusssteuerung umzustellen, konnte durch Anwendung der in der Arbeit vorgestellten Steuerungsmethode erfüllt werden. Dabei haben sowohl die vorgenommene Simulation als auch die darauf folgende, reale Umsetzung dies bestätigende Ergebnisse geliefert.

Im Rahmen der Simulationsläufe wurden alle relevanten Szenarien für den Betrieb eines MFS getestet und damit die Fähigkeit der Steuerungsmethode zur verteilten Selbstkonfiguration und zur Wegoptimierung sowie ihre Robustheit verifiziert. Bei einem Neustart der Simulation hatten die einzelnen, simulierten MFM innerhalb kürzester Zeit ihre Daten soweit ausgetauscht, dass jedes einzelne MFM über ein für die Wegfindung ausreichendes Modell der Topographie verfügte. Ebenso zeigte sich ein verändertes MFS innerhalb kürzester Zeit in der Lage, sich an die Änderungen anzupassen und seinen Betrieb aufrecht zu erhalten, soweit dies möglich war. Die Optimierung des Transportweges sowohl für alternative Wege als auch bei Ausfällen von MFM konnte in verschiedenen Simulationsläufen beobachtet werden. Das von der Simulation gezeigte Zeitverhalten war vor dem Hintergrund einer mit einer Sekunde angesetzten Heartbeat-Zeit mehr als befriedigend.

Durch die Umsetzung in dem Referenz-MFS konnte die in der Simulation nachgewiesene Funktionsfähigkeit zusammen mit der geforderten Leichtgewichtigkeit an einem realen MFS dargestellt werden. Da für die Umsetzung in handelsüblichen SPSen, deren Speicherplatz, Rechenpower und Kommunikationsfähigkeit noch relativ eingeschränkt ist, gerade die Leichtgewichtigkeit ein entscheidendes Kriterium ist, erbrachte die reale Umsetzung diesen Nachweis. Dabei weisen die zum Einsatz kommenden SPSen mit Hinblick auf grundsätzliche Adaptivität noch Potenzial aus. Zum einen muss es möglich werden, die Kommunikationsverbindungen über das Bussystem dynamisch zu konfigurieren, zum anderen ist eine dynamischer Datenhaltung innerhalb der SPS wünschenswert.

Zusammenfassend kann also festgestellt werden, dass im Rahmen der Simulation und Umsetzung der entwickelten Steuerungsmethode der vollständige Nachweis ihrer Funktionsfähigkeit erbracht wurde. Dabei kann die Steuerungsmethode alle eingangs geforderten Anforderungen erfüllen, indem sie Selbstkonfiguration erlaubt, Modularität durch vollständige Verteiltheit unterstützt, eine Wegoptimierung durchführt und robust und leichtgewichtig genug für den industriellen Einsatz ist.

9. Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurde eine Methode zur Steuerung von MFS erarbeitet, die es ermöglicht, modular aufgebaute MFS vollständig verteilt und adaptiv hinsichtlich des Materialflusses zu steuern. Eine zentrale Steuerung, wie sie bei bestehenden Systemen notwendig ist, zusammen mit einer umfangreichen Programmierung der Materialflusssteuerung entfallen somit. Dabei benötigen die einzelnen MFM eines auf der beschriebenen Methode aufgebauten MFS nur eine Parametrierung hinsichtlich ihrer unmittelbaren Vorgänger und Nachfolger und sind damit in der Lage, durch Austausch von Informationen mit diesen Vorgängern und Nachfolgern die Wegfindung vorzunehmen. Dies ermöglicht eine maßgebliche Erhöhung der Anpassungsfähigkeit von MFS und wird sowohl anhand einer Simulation als auch Anhand eines Beispielfalles eines MFS verifiziert.

Grundlage der Arbeit sind sowohl der verteilte Algorithmus zur Wegfindung als auch das Protokoll zum Austausch der Daten zwischen den einzelnen MFM. Für den verteilten Algorithmus galt es zuerst, die bisher bekannten Modelle von MFS zu erweitern, da diese von einem vollständigen MFS ausgehen und nicht die Modellierung von Teilmodulen erlauben. Da aber die Arbeit eine Methode entwickelt, die es unter anderem ermöglicht, neue MFM in bestehende MFS einzufügen, ohne das bestehende MFS umkonfigurieren zu müssen, ist es für diese MFM unerlässlich, ihr eigenes, nur das MFM und seine Schnittstellen beschreibende Abbild zu führen. Ausgehend von diesen Modellen aller MFM kann der entwickelte verteilte Algorithmus eine Wegfindung für die Transportaufgabe vornehmen. Dabei werden die Kosten der einzelnen, erreichbaren MFM durch den Algorithmus als Abbildung der Topographie in den sogenannten Nachfolgertabelle errechnet. Diese Nachfolgertabellen werden von den MFM an ihre Vorgänger weitergegeben und sind dort Grundlage für die Berechnung der dortigen Nachfolgertabellen.

Im Zusammenspiel zwischen Algorithmus und Protokoll ergibt sich die enge Verbindung beider Bestandteile der Methode. So sichert

das Protokoll nicht nur den Datenaustausch zwischen angrenzenden MFM, sondern erzeugt durch seine Überwachungsfunktion selbst Ereignisse, die durch den Algorithmus erfasst werden müssen. Dies gilt für das Ausbleiben eines Heartbeats eines überwachten MFM, in diesem Fall muss der Algorithmus entsprechend der Annahme, dass das überwachte MFM ausgefallen ist, dieses MFM aus seinen Daten streichen. Aus diesem beschriebenen Zusammenspiel von Algorithmus und Protokoll ergibt sich die Fähigkeit der Steuerungsmethode, sowohl neue MFM jederzeit integrieren zu können als auch ausgefallene MFM zu erkennen und entsprechend systemweit zu löschen. Die erarbeitete Auflösung der beiden Kernbestandteile in die drei Softwarekomponenten *Wegfindung*, *Topographiebildung* und *Plug-and-Play* und deren Untersetzung mit einzelnen Funktionen sind von großer Bedeutung, da somit eine klare Abgrenzung der einzelnen Implementierungen erlaubt wird.

Mit der vorgestellten Arbeit erfolgt ein weiterer Schritt in Richtung der Vision des Plug-and-Produce - also des Zusammenwirkens intelligenter Maschinen und Geräte in der Produktion ohne Konfigurationsaufwand. Mit der grundlegenden Auflösung der bisherigen Gesamtsystem-Modelle in modulbezogene Modelle wird ein wichtiger Schritt in diese Richtung getan, der in andere Funktionsbereiche der Fertigung übertragen werden kann. Dabei ist besonders die Schnittstellendefinition zwischen diesen Modulen als ausschlaggebender Punkt hervorzuheben.

Für den Einsatz im Individualverkehr wird die Anwendbarkeit auf sehr große Transportsysteme mit mehreren zehntausend Modulen zu prüfen und verifizieren sein, wobei dies anderen Arbeiten überlassen werden muss. Dies bezieht sich insbesondere auf das Zeitverhalten bei Umkonfigurationen, wobei eventuell eine Erweiterung der Methode um eine Segmentierung der erreichbaren Ziele zu prüfen ist. Die Fragestellung der verteilten Datenhaltung in sehr großen Transportsystemen ist in diesem Falle ebenfalls zu untersuchen. Als grundsätzlicher Forschungsbedarf in verteilten Systemen mit ihrer Asynchronität und ihrer Parallelität sind bessere Beschreibungsmittel

auszuweisen. Die in der Arbeit verwendeten (und an die UML angelehnten) Diagramme gilt es zu erweitern.

10.Glossar

Ausgangsast- Nachfolgertabelle (AN)	Die in einem MFM abgelegte und um die Kosten des Ausgangsasts erhöhte Eingangsast-Nachfolgertabelle eines Nachfolgers. AN sind immer einem Ausgangsast des MFM direkt zugeordnet.
Eingangsast- Nachfolgertabelle (EN)	Die in einem MFM abgelegte und um die Kosten des Eingangsasts erhöhte Zentralknoten-Nachfolgertabelle. EN sind immer einem Eingangsast des MFM direkt zugeordnet.
Funktion	Ein Teil einer Softwarekomponente, der eine spezielle Aufgabe ausführt.
Hardwaretabelle	Zuordnung der direkten Eingänge und Ausgänge der individuellen Steuerung zu den Ästen des MFMs.
Initialisierungsdaten	Die in einem MFM persistent gespeicherten Daten bezüglich seiner direkten Vorgänger und Nachfolger .
Konfiguration des MFS	Die Konfiguration des MFS ist dann erfolgt, wenn in allen MFM sowohl jeweils die Nachfolgertabellen vorliegen, als auch alle Heartbeats vereinbart sind.

Materialflussmodul (MFM)	Ein MFM wird als die kleinste Einheit innerhalb eines MFS verstanden. Es besteht aus einer unteilbaren Förderstrecke, wobei diese auch eine Weiche, Zusammenführung etc. sein kann.
Materialflusssystem (MFS)	Ein MFS setzt sich ausschließlich aus einzelnen MFM zusammen.
Messages	Zwischen einzelnen MFM über das Bussystem ausgetauschte Messages: <i>Anmeldung im System</i> und <i>Heartbeats</i> .
Softwarekomponente	Bezeichnung für einen in sich abgeschlossenen Teil der Software: <i>Wegfindung</i> , <i>Topographiebildung</i> und <i>Plug-and-Play</i> .
Zentralknoten- Nachfolgertabelle (ZN)	Die in einem MFM abgelegte Tabelle, die alle von ihm erreichbaren Nachfolger im System enthält. Die ZN stellt eine Zusammenfassung der AN dar.
Zielinformation	Information, die bestimmt, wohin das Fördergut transportiert werden soll.

11.Literaturverzeichnis

- /ARC 01/ ARC Advisory Group: PLCs: Is There a Future? ARC Strategies. Dedham (MA), USA: ARC, 2001
- /ARC 01b/ ARC Advisory Group: Web-hosted Solution Strategies Enabling E-Manufacturing, ARC Strategies. Dedham (MA), USA: ARC, 2001
- /Arlt 93/ Arlt, C.: Netzwerkflussprobleme. Wiesbaden: Deutscher Universitäts-Verlag, 1993
- /Arnold 98/ Arnold, D.: Materialflusslehre. Braunschweig: Vieweg Verlag, 1998
- /Ashby 56/ Ashby, W. R.: Introduction to Cybernetics. New York: John Wiley & Sons, 1956
- /Axmann 93/ Axmann, N.: Stückgutförderer: Handbuch für Materialflusstechnik. Böblingen: expert Verlag, 1993
- /Babaoglu 93/ Babaoglu, Ö; Marzullo, K.: Consistent Global States of Distributed Systems: Fundamental Concepts and Mechanisms, in: Mullender, S.: Distributed systems. New York: Addison-Wesley, 1993
- /Bahke 74/ Bahke, E.: Materialflusssysteme. Band 1: Materialflusstechnik. Mainz: Krausskopf-Verlag, 1974
- /Ball 95/ Ball; M.; Magnanti; T.; Monma, C.; Nemhause, G.: Handbooks in Operations Research and Management Science: Network Routing. Amsterdam: Elsevier, 1995
- /Balzert 00/ Balzert, H.: Lehrbuch der Software-Technik. Heidelberg: Spektrum Akademischer Verlag, 2000

- /Balzert 01/ Balzert, H.: UML Kompakt. Heidelberg: Spektrum Akademischer Verlag, 2001
- /Beckhoff 01/ N.N.: Just in Time - Durch Ethernet vernetzte dezentrale Steuerungstechnik, in: PC-Control - The New Automation Technology Magazine, Nr. 1. Verl: Beckhoff Industrie Elektronik, 2001
- /Behrens 00/ Behrens, S.: Produktionstheoretische Perspektiven der Virtuellen Unternehmung, in: Virtuelle Unternehmen, Ergänzungsheft 2/2000, ZfB. Wiesbaden: Verlag Gabler, 2000
- /Bertsekas 87/ Bertsekas, D.; Gallager, R.: Data Networks. Englewood Cliffs (NJ), USA: Prentice Hall, 1987
- /Böge 98/ Böge, A.: Vieweg Lexikon Technik. Braunschweig: Vieweg Verlag, 1998
- /Boger 99/ Boger, M.: Java in verteilten Systemen. Nebenläufigkeit, Verteilung, Persistenz. Heidelberg: dpunkt-Verlag, 1999
- /Bohl 01/ Bohl, A.: Transportsystem für Flaschen - Gesteuert wird dezentral, in: F+H Fördern und Heben, Jahrgang 51, Nr. 1-2. Mainz: Vereinigte Fachverlage, 2001
- /Bosch 00/ N.N.: Modulares TransferSystem MTS 2. Stuttgart: Robert Bosch GmbH, 2000
- /Bosch 02/ N.N.: Sitzt, passt, wackelt und schafft Luft - Flexibles Transfersystem aus dem Baukasten, in: Der Betriebsleiter, Ausgabe 4. Mainz: Verlag für Technik und Wirtschaft, 2002

- /Braatz 01/ Braatz, A.; Groß-Rohde, M.; Ehrig, H.; Westkämper, E.: UML-basierte Software-Spezifikation und Entwicklungswerkzeuge für Systeme der Automatisierungstechnik, in: Schnieder, E. (Hrsg.): Engineering komplexer Automatisierungssysteme, EKA 2001. Braunschweig: Technische Universität Braunschweig, 2001
- /Brinkmann 87/ Brinkmann, H.-G.: Beitrag zur Steuerung von Mikroprozessorgeführten Stückgutfördersystemen. Dortmund: Universitätsdissertation, 1987
- /Brockhaus 98/ N.N.: Brockhaus - Die Enzyklopädie. Leipzig: Brockhaus Verlag, 1998
- /Bruland 95/ Bruland, M.: DIAtec und DIALoc. Dezentrale intelligente Automation mit LON, in: Automatisierungskonzepte mit dezentraler Intelligenz (LonWorks). Dresden: Tagungsband LonWorks, 1995
- /Bruns 01/ Bruns, K.; Klimsa, P.: Informatik für Ingenieure kompakt. Braunschweig: Vieweg Verlag, 2001
- /Büchter 00/ Büchter, H.: Steuerungen für Stückgutfördersysteme: Entwicklungspotenziale und Trends, in: F+H Fördern und Heben, Jahrgang 50, Nr. 3. Mainz: Vereinigte Fachverlage, 2001
- /Buchwitz 01/ Buchwitz, M.: Die verteilte Intelligenz, in: Computer & Automation, Ausgabe 6. Poing: WEKA Fachzeitschriftenverlag, 2001
- /Coulouris 01/ Coulouris, G.; Dollimore, J.; Kindberg, T.: Distributed Systems, Concepts and Design. London: Addison-Wesley, 2001

- /Diestel 00/ Diestel, R.: Graphentheorie. Berlin: Springer Verlag, 2000
- /Dietrich 98/ Dietrich, D.; Loy, D.; Schweinzer, H.-J.: LON-Technologie - Verteilte Systeme in der Anwendung. Heidelberg: Hüthig-Verlag, 1998
- /Dijkstra 59/ Dijkstra, E.: A Note on Two Problems in Connection with Graphs, Numerische Mathematik, Vol. 1, 1959
- /DIN 83/ Deutsches Institut für Normung: DIN 30781 - Transportkette: Grundbegriffe. Berlin: Beuth Verlag, 1983
- /DIN 84/ Deutsches Institut für Normung: DIN 1505 Teil 2: Titelangaben von Dokumenten - Zitierregeln. Berlin: Beuth Verlag, 1984
- /DIN 94/ Deutsches Institut für Normung: DIN 19226 Teil 1: Regelungstechnik und Steuerungstechnik - Allgemeine Grundbegriffe. Berlin: Beuth Verlag, 1994
- /DIN 94a/ Deutsches Institut für Normung: DIN 15201 Teil 1: Stetigförderer - Benennungen. Berlin: Beuth Verlag, 1994
- /DIN 95/ Deutsches Institut für Normung: DIN 1505 Teil 3: Titelangaben von Dokumenten - Literaturverzeichnisse. Berlin: Beuth Verlag, 1995
- /Dobrowolny 99/ Dobrowolny, V.; Wolter, O.: Zur Anwendung semantischer Modelle bei der Planung logistischer Systeme, in: Ziems, D. (Hrsg.): Magdeburger Schriften zur Logistik - Heft 3. Magdeburg: LOGiSCH GmbH, 1999

- /Domschke 95/ Domschke, W.: Logistik - Transport. München: Oldenbourg Verlag, 1995
- /Domschke 97/ Domschke, W.: Logistik - Rundreisen und Touren. München: Oldenbourg Verlag, 1997
- /Douglass 99/ Douglass, B. P.: Real Time UML. New-York (NY), USA: Addison Wesley, 1999
- /Douglass 99b/ Douglass, B. P.: Doing Hard Time: Developing Real-Time Systems with UML, Objects, Frameworks and Patterns. New-York (NY), USA: Addison Wesley, 1999
- /Dubbel 97/ Beitz, W.; Grote, K.-H.: Taschenbuch für den Maschinenbau - Dubbel. Berlin: Springer Verlag, 1997
- /Düll 77/ Düll, E.: Prozessautomatisierung mit Mikroprozessoren am unteren Ende der Prozessrechner, Automatische Technik im Wandel durch Mikroprozessoren, Interkama Kongress. Berlin: Springer Verlag, 1977
- /Dürschmidt 97/ Dürschmidt, S.: Wandlungsfähigkeit im Layout, in: Logistik Heute, Ausgabe 11. München: hussverlag, 1997.
- /Eckstein 00/ Eckstein, R.: XML - kurz & gut. Köln: O'Reilly Verlag, 2000
- /Edwards 00/ Edwards, W. K.: Core Jini™, The Sun Microsystems Press Java Series. Englewood Cliffs (NJ), USA: Prentice Hall, 2000
- /Eisenschmid 01/ Eisenschmid, M.: Supply-Chain-Collaboration bei Daimler-Chrysler, in: Innovative Logistik in der Automobilindustrie, VDI-Bericht 1628. Düsseldorf: VDI Verlag, 2001

- /EN 01/ Deutsches Institut für Normung: DIN EN 61131
Speicherprogrammierbare Steuerungen - Teil 5:
Kommunikation. Berlin: Beuth Verlag, 2001
- /Enslow 74/ Enslow, P. H.: Multiprocessors and Parallel
Processing - Comtre Corporation. New York (NY), USA:
John Wiley & Sons, 1974
- /Enslow 78/ Enslow, P.H.: What is a distributed data
processing system. Computer, January 1978
- /Enste 99/ Enste, U.; Epple, U.: Komponentenbasierte
Funktionsbausteintechnik, in: Schnieder, E. (Hrsg.):
Entwicklung und Betrieb komplexer
Automatisierungssysteme. Braunschweig: Technische
Universität Braunschweig, 1999
- /EtherNet/IP 01/ N.N.: EtherNet/IP Specification, Release 1.0,
June 5, 2001. Boca Raton (FL), USA: ControlNet
International and Open DeviveNet Vendor Association,
2001
- /Farley 98/ Farley, J.: Java Distributed Computing. Sebastopol
(CA), USA: O'Reilley & Associates, 1998
- /Flanagan 99/ Flanagan, D.; Farley, J.; Crawford, W.;
Magnusson, K.: Java Enterprise in a Nutshell.
Sebastopol (CA), USA: O'Reilly & Associates, 1999
- /Freese 97/ Freese, H.; Varnholt, R.: Dezentral vor Ort an CAN
oder AS-i. In: Elektrotechnik + Automation
Elektrotechnische Zeitung Nr. 21. Berlin: VDE-Verlag,
1997

- /Freitag 01/ Freitag, J.: Java in der Automatisierungstechnik, in: Schraft, R.; Bender, K.; Brandenburg, G. (Hrsg.): Tagungsband SPS/IPC/Drives 2001. Heidelberg: Hüthig Verlag, 2001
- /Frikha 99/ Frikha, M.: Performance-Optimierung und Simulation von GSM-Festnetzen mit nichthierarchischer Verkehrslenkung. Aachen: Shaker Verlag, 1999
- /Gausemeier 98/ Gausemeier, J.; Förste, S.; Gerdes K.-H.: Dezentrale Steuerungsarchitektur für modulare Materialflusssysteme, in: Schraft, R.; Bender, K.; Brandenburg, G.: Tagungsband SPS/IPC/Drives 1998. Heidelberg: Hüthig Verlag, 1998
- /Göpfert 00/ Göpfert, I.: Die Anwendung der Zukunftsforschung für die Logistik, in: Göpfert (Hrsg.): Logistik der Zukunft. Wiesbaden: Verlag Gabler, 2000
- /Gora 01/ Gora, W.; Scheid, E.: E-Commerce: Auswirkungen, Chancen und Risiken, in: Logistik Spektrum, Supplement zu Fördern und Heben und Distribution, Ausgabe 13. Mainz: Vereinigte Fachverlage, 2001
- /Graf 00/ Graf, I.: Die Entwicklung der Logistik im Einfluss der Visionen - Ein Beitrag der Adolf Würth GmbH & Co.KG, in: Göpfert (Hrsg.): Logistik der Zukunft. Wiesbaden: Verlag Gabler, 2000
- /Großeschallau 1984/ Großeschallau, W.: Materialflussrechnung - Modelle und Verfahren zur Analyse und Berechnung von Materialflusssystemen. Berlin: Springer Verlag, 1984
- /Gruhler 00/ Gruhler, G. (Hrsg.): Feldbusse und Gerätekommunikationssysteme - Einführende Darstellung und detaillierter Vergleich. Reutlingen: Steinbeis-Transferzentrum Automatisierung (STA), 2000.

- /Grundmann 02/ Grundmann, U.: ARC Essay - Java betritt das Parkett, in: Computer & Automation. Ausgabe 1. Poing: WEKA Fachzeitschriftenverlag, 2002
- /Gudehus 77/ Gudehus, T.: Strukturanalyse der Transportsysteme. Teil 1: Die System-Elemente, in: Fördern und Heben 27 (Nr.1). Mainz: Vereinigte Fachverlage, 1977
- /Gudehus 77b/ Gudehus, T.: Strukturanalyse der Transportsysteme. Teil 2: Die System-Struktur, in: Fördern und Heben 27 (Nr. 2). Mainz: Vereinigte Fachverlage, 1977
- /Gudehus 99/ Gudehus, T.: Logistik - Grundlagen, Strategien, Anwendungen. Berlin: Springer Verlag, 1999
- /Gueorguiev 99/ Gueorguiev, M.; Ziems, D.: Gestaltung von timed Petri-Netz Modellen und Leistungsanalyse der Knoten der Materialflusssysteme, in: Ziems, D. (Hrsg.): Magdeburger Schriften zur Logistik Heft 2. Magdeburg: LOGiSCH GmbH, 1999
- /Gülich 95/ Gülich, E.: Dezentral besser gesteuert, in: Logistik Heute, Ausgabe 11. München: Huss-Verlag, 1995
- /Gumm 00/ Gumm, H.-P.; Sommer, M: Einführung in die Informatik. München: Oldenbourg Verlag, 2000
- /Halabi 98/ Halabi, Bassam: Internet-Routing-Architekturen: Grundlagen, Design und Implementierung. München: Hanser Verlag, 1998

- /Härdtner 91/ Härdtner, M.: Materialflusssteuerung als integraler Bestandteil eines flexiblen Fertigungssystems, in: Steuerung von Materialflusssystemen, VDI-Bericht 881. Düsseldorf: VDI-Verlag, 1991
- /Harnischer 01/ Harnischer, G.; Lupp, K.; Arph, J.: Was bedeutet .NET für die Leit- und Automatisierungstechnik? Internet als Automation-Middleware, in: SPS Magazin Nr. 10. Würzburg: TeDo Verlag, 2001
- /Hartmann 95/ Hartmann, M.: Merkmale zur Wandlungsfähigkeit von Produktionssystemen (Univ. Dissertation). Magdeburg: OvG Univ. Magdeburg, 1995
- /Heeg 97/ Heeg, F.J.; Knackfuß, P.: Automatisierungstechnik für Produktionstechniker. Aachen: Verlag Mainz, 1997
- /Hering 95/ Hering, E.; Gutekunst, J.; Dyllong, U.: Informatik für Ingenieure. Düsseldorf: VDI Verlag 1995
- /Hermos 01/ Hermos Informatik GmbH: Technical Reports: Transponder Reader RS 232 ASCII-Protocol. Release 1.3. Mistelgau: Hermos Informatik GmbH, 2001
- /Hoffmann 99/ Hoffmann, J.: MATLAB und SIMULINK in Signalverarbeitung und Kommunikationstechnik. München: Addison-Wesley, 1999
- /IDA 01/ N.N.: IDA White Paper V 1.0. Blomberg: IDA Group, April 2001
- /IDA 01a/ N.N.: Interface for Distributed Automation - Architecture Description and Specification. Blomberg: IDA Group, November 2001

- /IEC 00/ N.N.: IEC-PAS 61499-1, Function blocks for industrial-process measurement and control systems, Edition 1.0, 2000-09. Düsseldorf: VDI Verlag, 2000
- /IEC 02/ Deutsches Institut für Normung: DIN IEC 61158 Feldbus für industrielle Leitsysteme - Teil 2: Spezifikation der Bitübertragungsschicht (Physical Layer) und Teil 3: Definition der Dienste der Sicherungsschicht (Data Link Layer). Berlin: Beuth Verlag, 2002
- /ISO 97/ International Standards Organisation: ISO 690 Information and documentation - Bibliographic references - Part 2: Electronic documents or parts thereof. Genf: International Standards Organisation, 1997
- /Jansen 01/ Jansen, R.; Krabs, A.: Transpondereinsatz in der Logistik nimmt zu, in: Logistik Jahrbuch 2001. Düsseldorf: Verlagsgruppe Handelsblatt, 2001
- /Jensen 02/ Jensen, E. D.: The Distributed Real-Time Specification for Java - An Initial Proposal. Burlington (MA), USA: The MITRE Corporation, 2002
- /Jobst 99/ Jobst, F.: Programmieren in Java. München: Carl Hanser Verlag, 1999
- /Jörns 97/ Jörns, C.; Kühner, R.: Hybride Modellierung des innerstädtischen Verkehrs auf der Basis adaptiver interpretierender Petri-Netze, in: at - Automatisierungstechnik, Nr. 45. München: Oldenbourg Verlag, 1997
- /Jünemann 98/ Jünemann, R.; Beyer A.: Steuerung von Materialfluss- und Logistiksystemen. Berlin: Springer Verlag, 1998

- /Jünemann 00/ Jünemann, R.; Schmidt, T.: Materialflusssysteme - Systemtechnische Grundlagen. Berlin: Springer Verlag, 2000
- /Jungnickel 99/ Jungnickel, D.: Graphs, Networks and Algorithms. Berlin: Springer Verlag, 1999
- /Kabitzsch 99/ Kabitzsch, K.: Telediagnose an Speicherprogrammierbaren Steuerungen, in: VDI-Berichte 1515, Industrielle Automation und Internet / Intranet Technologie. Düsseldorf: VDI Verlag, 1999
- /Käppner 95/ Käppner, M.: Computergestützte Steuerungs- und Positionierverfahren für Elektrohängebahnen. Dortmund: Verlag Praxiswissen, 1995
- /Kief 98/ Kief, H. B.: FFS-Handbuch: Einführung in flexible Fertigungssysteme und deren Komponenten. München: Hanser Verlag, 1998
- /Klostermeyer 01/ Klostermeyer, A.; Lorentz, K.: WIM - Web Integrated Manufacturing, in: Praxis Profiline - Industrial Ethernet. Würzburg: Vogel Verlag, 2001
- /Knackfuß 97/ Knackfuß, P.; Detken, K.-O.: Vernetzte Werkzeugmaschinen und Leitreechner. In: Heeg, Hirsch, Knackfuß (Hrsg.): Automatisierungstechnik für Produktionstechniker. Aachen: Verlag Mainz, 1997.
- /Konhäuser 98/ Konhäuser, W.: Industrielle Steuerungstechnik - Grundlagen und Anwendungen. München: Hanser Verlag, 1998.
- /Krämer 99/ Krämer, K.: Informationslogistik und Auto-ID, in: Krämer, K.; Wölker, M.; Jünemann, R. (Hrsg.): Basis der Unternehmensprozesse - Identifikationstechnik. Frankfurt am Main: Umschau Zeitschriftenverlag, 1999

- /Kreisel 72/ Kreisel, G.; Krivine, J.: Modelltheorie - eine Einführung in die mathematische Logik und Grundagentheorie. Berlin: Springer Verlag, 1972
- /Kriesel 98/ Kriesel, W.; Heimbold, T.; Telschow, D.: Bustechnologien für die Automation. Heidelberg: Hüthig Verlag, 1998
- /Krüger 00/ Krüger, G.: GoTo Java 2 - Handbuch der Java Programmierung. München: Addison-Wesley, 2000
- /Kühnle 99/ Kühnle, H.: Produktion 2010, in: Markt und Mittelstand, Ausgabe 7. München: Markt und Mittelstand Verlag, 1999
- /Kühnle 00/ Kühnle, H.; Klostermeyer, A.; Lorentz, K.: Neue Wege in der Fabrikautomation, in: wt Werkstattstechnik, Nummer 91, Heft 3. Düsseldorf: Springer VDI Verlag, 2001
- /Kühnle 00b/ Kühnle, H.; Rietz, S.: Dynamische Strukturen - Anspruch und Chancen, in: Dück, O.: Praxishandbuch Plus Materialwirtschaft und Logistik in der Praxis - Konzepte, Methoden, Techniken erfolgreich umsetzen. Augsburg: WEKA Verlag, 2000
- /Kühnle 01/ Kühnle, H.; Klostermeyer, A.; Lorentz, K.: A Paradigm Shift to Distributed Systems in Plant Automation. In: Fares Sebaaly, M. (Hrsg.). Dubai: Proceedings of The International NAISO Congress on Information Science Innovations ISI 2001, March 17-21, 2001
- /Lang 97/ Lang, M.: Effizienz von Verfahren zur Adaptiven und verteilten Verkehrslenkung in Packetvermittlungsnetzen (Univ. Dissertation). Stuttgart: Univ. Stuttgart, 1997

- /Langmann 96/ Langmann, R.: Prozesslenkung – Grundlagen zur Automatisierung technischer Prozesse. Braunschweig: Vieweg Verlag, 1997
- /Laux 01/ Laux, T.: Es wird Echt Zeit, in: IEE Automatisierung und Datentechnik, Ausgabe 4, Jahrgang 45. Heidelberg: Hüthig Verlag, 2001
- /LON 98/ N.N.: Objektnahe Systembusse: LON, in: VDI Nachrichten, Nr. 44. Düsseldorf: VDI Verlag, 1998
- /Lorentz 99/ Lorentz, K.; Martinetz, J.; Markfort, D.: Gnosis VF – Deliverable 1.1, Distributed Architectures and Organisation of the Virtual Factory. Magdeburg: Gnosis VF Consortium, 1999
- /Lorentz 01/ Lorentz, K.; Peschke, J.; Lüder, A.: Echtzeit-Java in der Automatisierungstechnik – Spezifikationen und praktische Anwendungen, in: Schraft, R.; Bender, K.; Brandenburg, G. (Hrsg.): Tagungsband SPS/IPC/Drives 2001. Heidelberg: Hüthig Verlag, 2001
- /Lorentz 01b/ Lorentz, K.; Lüder, A.; Peschke, J.: PABADIS – Fabrikautomatisierung basierend auf verteilten Systemen, in: Schraft, R.; Bender, K.; Brandenburg, G. (Hrsg.): Tagungsband SPS/IPC/Drives 2001. Heidelberg: Hüthig Verlag, 2001
- /Lorentz 02/ Lorentz, K.; Klostermeyer, A.: Anwendungsorientierte Ethernet-Lösungen – Zusammenführung verschiedener ethernetbasierter Ansätze, in: A&D Kompendium 2002. München: publish-industry Verlag, 2002

- /Lorentz 02b/ Lorentz, K.: Verfahren zur adaptiven Zielsteuerung von Transportgütern in Materialflusssystemen, Deutsche Patentanmeldung DE 102 53 105.6, 2002
- /Lorentz 02c/ Lorentz, K.; Messerschmidt, R.; Schwab, C.: Das Wandlungsfähige Transportsystem - Entwicklung modularer Komponenten für Stückgut-Materialflusssysteme. Projektdokumentation, Forschungsprojekt S508 der Stiftung Industrierforschung. Magdeburg: Magdeburg: Otto-von-Guericke-Universität, 2002
- /Lüth 98/ Lüth, T.: Technische Multi-Agenten-Systeme: Verteilte autonome Roboter- und Fertigungssysteme. München: Hanser Verlag, 1998
- /Marke 91/ Marke, K.: Rechner- und Kommunikationsstrukturen in Materialflusssystemen, in: Steuerung von Materialflusssystemen, VDI-Bericht 881. Düsseldorf: VDI Verlag, 1991
- /Maurer 99/ Maurer, C.: Grundzüge der nichtsequentiellen Programmierung. Berlin: Springer Verlag, 1999
- /McQuillan 80/ McQuillan, J.; Richter, I.; Rosen, E.: The New Routing Algorithm for the ARPANET, in: IEEE Transactions on Communications, Vol. 28, Nr. 5. New York (NY), USA: The Institute of Electrical and Electronics Engineers, 1980
- /Messerschmidt 02/ Messerschmidt, R.; Dreher, A.: Echtzeit mit Ethernet, in: Industrial Ethernet - Produkte und Applikationen. Würzburg, Vogel Verlag, 2002

- /Messerschmidt 02b/ Messerschmidt, R.; Lorentz, K.:
Flexibilität durch Plug & Play, in: SPS-Magazin, HMI-Special 2002. Marburg: TeDo-Verlag, 2002
- /Meyer 99/ Meyer, M.: Kommunikationstechnik. Braunschweig:
Vieweg Verlag, 1999
- /Minsky 65/ Minsky, M.: Matter, Mind and Models, in:
Kalenichi, W. (Hrsg.): Information Processing.
Washington (DC), USA: Spartan Books, 1965
- /Montrac 01/ N.N.: Produktbeschreibung Montrac. Derendingen,
Schweiz: Montech AG, 2001
- /MOVA 00/ N.N.: MOVA - Modulare Offene Verteilte
Funktionsblocksysteme für die Automatisierungstechnik;
Abschlusspräsentation des BMBF-Verbundprojektes.
Frankfurt am Main: VDMA Verlag, 2000
- /Nägler 96/ Nägler, G.; Stopp, F.: Graphen und Anwendungen:
Eine Einführung für Studierende der Natur-, Ingenieur-
und Wirtschaftswissenschaften. Stuttgart: B. G.
Teubner Verlagsgesellschaft, 1996
- /Oestereich 99/ Oestereich, B.: Objektorientierte
Softwareentwicklung: Analyse und Design mit der
Unified Modelling Language. Stuttgart: Oldenbourg
Verlag, 1999
- /Offenlegung 01/ N.N.: Anlage zur Bearbeitung von Wafern,
Offenlegungsschrift DE 199 52 195 A1, Deutsches
Patent- und Markenamt, Bundesdruckerei, 2001
- /P1588 02/ IEEE: Draft Standard for a Precision Clock
Synchronisation Protocoll for Network Measurement and
Control Systems (P1588). New York (NY), USA: Institute
of Electrical and Electronics Engineers, 2002

- /Panreck 02/ Panreck, K.: Rechnergestützte Modellbildung physikalisch-technischer Systeme. Fortschr.-Ber. VDI Reihe 8 Nr. 945. Düsseldorf: VDI Verlag, 2002
- /Pardo-Castellote 99/ Pardo-Castellote, G. et al.: NDDS: The Real-Time Publish-Subscribe Middleware. Sunnyvale (CA), USA: Real-Time Innovations, Inc., 1999
- /Patent 01/ N.N.: Transportsystem und Verfahren zur Steuerung des Transportsystems, Europäische Patentschrift EP 0 894 751 B1, Europäisches Patentamt, 2001
- /Patz 01/ N.N.: Verteilte Intelligenz in der Steuerungstechnik - Interview mit Dr. Manfred Patz, Softing AG, in: Praxis Profiline - Industrial Ethernet. Würzburg: Vogel Verlag, 2001.
- /Pfeifer 98/ Pfeifer, H.; Kabisch, G.; Lautner, H.: Fördertechnik: Konstruktion und Berechnung. Braunschweig: Vieweg Verlag, 1998
- /Plagemann 01/ N.N.: Das Konzept der verteilten Intelligenz ist für Festo ein wichtiger Ansatzpunkt - Im Interview: B. Plagemann und M. Prinzen, in: A&D Newsletter, 3. Jahrgang, Ausgabe 6. München: Publish-Industry Verlag, 2001
- /Rauber 00/ Rauber, T.; Rüniger, G.: Parallele und verteilte Programmierung. Berlin: Springer Verlag, 2000
- /Razafimahefa 00/ Razafimahefa, C.; Bryce, C.: Towards the Design of an Internet Operating System, in: Tsichritzis, D. (Editor): Internet Objects. Genève: Centre Universitaire d'Informatique, Université de Genève, 2000

- /Reinhard 99/ Reinhard, G.: Wandel in der Montage - Flexibilität ist gut, Reaktionsfähigkeit noch besser, in: wt, Werkstattstechnik 89. Berlin: Springer Verlag, 1999
- /Ropohl 99/ Ropohl, G.: Eine Systemtheorie der Technik. München: Hanser Verlag, 1999
- /RTDA 01/ N.N.: Real-Time Data Access, Draft International JConsortium Spezifikation, Rev. 1.8. Cupertino (CA), USA: JConsortium, 2001
- /RTS 01/ Bollela, G.; Brosgol, B.; Furr, S.; Hardin, D.; Dibble, P.; Gosling, J.; Turnbull, M.: The Real-Time Specification for Java, First Public Release. New-York (NY), USA: Addison-Wesley, 2001
- /Rupp 00/ Rupp, T.; Löh, H.; Ristic, M.: Decentralised Manufacturing Co-ordination in Complex Supply Networks, in: Weber, F.; Pawar, K. (Editors): Proceedings of the 6th International Conference on Concurrent Enterprising. Nottingham: University of Nottingham, 2000
- /SAIA 01/ N.N.: Handbuch der Baureihen PCD1 und PCD2 - Hardware, Ausgabe 26/737 D6. Murten (Schweiz): Saia-Burgess Controls AG, 2001
- /Scheideler 97/ Scheideler, C.: Universal Routing Strategies for interconnection Networks (LNCS). Berlin: Springer Verlag, 1997
- /Schroeder 93/ Schroeder, M.: A State-of-the-Art Distributed System: Computing with BOB, in: Mullender, S.: Distributed systems. New-York (NY), USA: Addison-Wesley, 1993

- /Schwarz 87/ Schwarz, M.: Telecommunication Networks: Protocols, Modeling and Analysis. New-York (NY), USA: Addison-Wesley, 1987
- /Schwarzer 98/ Schwarzer, W.: Moderne Steuerungskonzepte optimieren den Materialfluss zum Nutzen des Kunden, in: VDI Bericht 1395: Schnell - Flexibel - Kostengünstig. Düsseldorf: VDI Verlag, 1998
- /Seifert 00/ Seifert, R.: The Switch Book. New York (NY), USA: John Wiley & Sons, 2000
- /Sommerer 98/ Sommerer, G.: Unternehmenslogistik. München: Carl Hanser Verlag, 1998
- /Steger 01/ Steger, A.: Diskrete Strukturen 1. Berlin: Springer Verlag, 2001
- /Stehle 01/ Stehle, W.: Digitale Netz: Grundlagen, Protokolle, Anwendungen. Weil der Stadt: J. Schlembach Verlag, 2001
- /Stein 01/ N.N.: Das Werkstückträger Transportsystem - Basic Components. Villingen-Schwenningen: Stein Automation GmbH, 2001
- /Tanenbaum 00/ Tanenbaum, A.: Computernetzwerke. München: Pearson Studium, 2000
- /Tanenbaum 95/ Tanenbaum, A.: Verteilte Betriebssysteme. München: Prentice Hall Verlag, 1995
- /Tauchnitz 01/ Tauchnitz, T.: Mit Internet-Technologie zur verteilten Automatisierung, in: SPS Magazin, Ausgabe 12. Marburg: TeDo-Verlag, 2001

- /Tel 00/ Tel, Gerard: Introduction to distributed Algorithms.
Cambridge: Cambridge University Press, 2000
- /Thomas 01/ Thomas, K.; Klockerbeck, G.; Hartweg, H.: Hohe
Performance im 24-Stunden-Betrieb, in: F+H Fördern und
Heben, Jahrgang 51, Nr. 5. Mainz: Vereinigte
Fachverlage, 2001
- /Thrun 97/ Thrun, W.; Stern, S.: Steuerungstechnik im
Maschinenbau. Braunschweig: Vieweg Verlag, 1997
- /Tönnißen 97/ Tönnißen, A.: Re-engineering - die sanfte
Operation. Beitrag auf der LNO-Roadshow, Stuttgart
1997.
- /Tsichritzis 99/ Tsichritzis, D: BIO / COM: Mixing Water with
Fire, in: Tsichritzis, D. (Editor): Trusted Objects.
Genève: Centre Universitaire d'Informatique,
Université de Genève, 1999
- /VDI 69/ Verein Deutscher Ingenieure (Hrsg.): VDI-Richtlinie
2340, Verteileinrichtungen. Berlin: Beuth Verlag, 1969
- /VDI 70/ Verein Deutscher Ingenieure: VDI 2411 - Begriffe und
Erläuterungen im Förderwesen. Düsseldorf: VDI Verlag,
1970
- /VDI 73/ Verein Deutscher Ingenieure: VDI 3300 -
Materialflussuntersuchungen. Düsseldorf: VDI Verlag,
1973
- /VDI 74/ Verein Deutscher Ingenieure: VDI 2689 - Leitfaden für
Materialflussuntersuchungen. Düsseldorf: VDI Verlag,
1974

- /VDI 75/ Verein Deutscher Ingenieure: VDI-Richtlinie 3596,
Darstellungsmethoden für den Materialfluss. Berlin:
Beuth Verlag, 1975
- /VDI 96/ Verein Deutscher Ingenieure: VDI-Richtlinie 3628,
Automatisierte Materialflusssysteme - Schnittstellen
zwischen den Funktionsebenen im Automatisierungsmodell
(Entwurf). Berlin: Beuth Verlag, 1996
- /VDI 96b/ Verein Deutscher Ingenieure: VDI-Richtlinie 3633,
Simulation von Logistik-, Materialfluss- und
Produktionsanlagen - Begriffsdefinitionen (Entwurf).
Berlin: Beuth Verlag, 1996
- /VDI 98/ Verein Deutscher Ingenieure: VDI-Richtlinie 2515,
Blatt 2, Identträger in Stückgutfördersystemen -
Mobile Datenspeicher. Berlin: Beuth Verlag, 1998
- /VDI 99/ Verein Deutscher Ingenieure: VDI-Richtlinie 2339,
Zielsteuerungen für Förder- und Materialflusssysteme.
Berlin: Beuth Verlag, 1999
- /VDI 00/ Verein Deutscher Ingenieure: VDI-Richtlinie 3633,
Blatt1: Simulation von Logistik-, Materialfluss- und
Produktionsanlagen - Grundlagen (Entwurf). Berlin:
Beuth Verlag, 2000
- /VDMA 94/ Verein Deutscher Maschinen und Anlagenhersteller:
VDMA- Einheitsblatt 15276: Datenschnittstellen in
Materialflusssystemen. Berlin: Beuth Verlag, 1994
- /W3C 00/ World Wide Web Consortium: XML Schema - Part 0:
Primer, W3C Recommendations. Sankt Augustin: Deutsches
Büro am IMK (FhG), 2000. [zitiert am 14.03.2002].
Erhältlich im World Wide Web:
<<http://www.w3.org/TR/2000/xmlschema-0/>>

- /Warnecke 93/ Warnecke, H. J.: Revolution in der Unternehmenskultur: Das Fraktale Unternehmen. Berlin: Springer Verlag, 1993
- /Weinert 01/ Weinert, A.: Java für Ingenieure. München: Hanser Verlag, 2001
- /Wellings 02/ Wellings, A.; Clark, R.; Jensen, E. D.; Wells, D.: A Framework for Integrating the Real-Time Specification for Java and Java's Remote Method Invocation. San Diego (CA), USA: Proc. International Symposium on Object Oriented Real-Time Distributed Computing, 2002.
- /Wenzel 01/ Wenzel, P.: Verteilte Automatisierung mit Profinet, in: SPS Magazin, Ausgabe 12: Marburg: TeDo-Verlag, 2001
- /Westkämper 97/ Westkämper, E.: Wandlungsfähige Unternehmensstrukturen - Teil I, in: Logistik-Spektrum Nr. 3. Mainz: Vereinigte Fachverlage, 1997
- /Wiendahl 98/ Wiendahl, H.-P.; Hartmann, M.: Produktion 2000plus. Visionen und Forschungsfelder für die Produktion in Deutschland. Untersuchungsbericht zur Definition neuer Forschungsfelder für die Produktion nach dem Jahr 2000. Aachen: RWTH Aachen, 1998
- /Wildemann 01/ Wildemann, H.: Supply Chain Management mit E-Technologien, in: E-Business Management mit E-Technologien, Ergänzungsheft 3/2001, ZfB. Wiesbaden: Verlag Gabler, 2001
- /Womack 92/ Womack, J.P.; Jones, D.T.; Roos, D.: Die zweite Revolution in der Autoindustrie. Frankfurt: Campus-Verlag, 1992

/WT 01/ N.N.: TCP/IP und Web-IO. Wuppertal: Wieseemann und Theis GmbH, 2001

/Wurmus 99/ Wurmus, H.; Wagner, B.: Verteilte Steuerung in Java mit Interbus Anbindung, in: Industrielle Automation und Internet/Intranet-Technologie, VDI Berichte, Nr. 1515. Düsseldorf: VDI Verlag, 1999

Anhang A: XML-Beschreibungen der Daten und Messages

Hardwaretabelle

```
<Hardwaretabelle>
  <Ast>
    <AstNummer>3</AstNummer>
    <FB> >1969</FB>
    [...]
  </Ast>
  [...]
</Hardwaretabelle>
```

Zielinformation mit Bearbeitungszustand

```
<Zielinformation >
  <Knoten>
    <KnotenNummer>3</KnotenNummer>
    <Bearbeitet_time>20011001_12:00</Bearbeitet_time>
  </Knoten>
</Zielinformation >
```

Zentralknoten-Nachfolgertabelle

```
<Zentralknoten-Nachfolgertabelle>
  <Knoten>
    <KnotenNummer>3</KnotenNummer>
    <Kosten>2</Kosten>
  </Knoten>
  [...]
</Zentralknoten-Nachfolgertabelle>
```

(EN und AN sind analog der ZN aufgebaut)

Initialisierungsdaten

```
<Initialisierungsdaten>
<Nachfolger>
  <Knoten>
    <KnotenNummer>3</KnotenNummer>
    <Ast>2</Ast>
  </Knoten>
  <Ast Kosten=4>2</Ast>
  <Datum>20011231160059000</Datum>
</Nachfolger>
[...]
// Dies sind die Vorgaenger
<Vorgaenger>
  <Knoten>
    <KnotenNummer>4</KnotenNummer>
    <Ast>2</Ast>
  </Knoten>
  <Ast Kosten=2>1</Ast>
  <Datum>20011231160059000</Datum>
</Vorgaenger>
[...]
</Initialisierungsdaten>
```

Anmeldung im System

```
<Message>
<Typ>Anmeldung bei Vorgaenger</Typ>
<Empfängerknoten>
  <KnotenNummer>3</KnotenNummer>
  <Ast>2</Ast>
  <Datum>20011231160059000</Datum>
</Empfängerknoten>
<Absenderknoten>
  <KnotenNummer>4</KnotenNummer>
  <Ast Kosten=4>1</Ast>
</Absenderknoten>
</Message>
oder
<Message>
<Typ>Anmeldung bei Nachfolger</Typ>
<Empfängerknoten>
  <KnotenNummer>5</KnotenNummer>
  <Timeout>200</Timeout>
  <Ast>7</Ast>
  <Datum>20011231160059000</Datum>
</Empfängerknoten>
<Absenderknoten>
  <KnotenNummer>4</KnotenNummer>
</Absenderknoten>
</Message>
```

Heartbeat Klasse 1

```
<Message>
<Typ>Heartbeat Klasse 1</Typ>
<Empfängerknoten>
  <KnotenNummer>1</KnotenNummer>
</Empfängerknoten>
<Absenderknoten>
  <KnotenNummer>2</KnotenNummer>
</Absenderknoten>
</Message>
```

Heartbeat Klasse 2:

```
<Message>
<Typ>Heartbeat Klasse 2</Typ>
<Empfängerknoten>
  <KnotenNummer>1</KnotenNummer>
</Empfängerknoten>
<Absenderknoten>
  <KnotenNummer>4</KnotenNummer>
</Absenderknoten>
<Knoten>
  <KnotenNummer Kosten=4>3</KnotenNummer>
</Knoten>
[...]
```

Heartbeat Klasse 3:

```
<Message>
<Typ>Heartbeat Klasse 3</Typ>
<Empfängerknoten>
  <KnotenNummer>1</KnotenNummer>
</Empfängerknoten>
<Initiierender Knoten>
  <KnotenNummer>2</KnotenNummer>
  <Datum>20011231160059000</Datum>
</Initiierender Knoten>
</Message>
```

Anhang B: Protokolle der Simulation

Protokoll 1: Hochfahren der Anlage

Die folgende Liste ist ein (um die Heartbeats 1 gekürzter) Auszug aus allen ankommenden und abgehenden Messages des MFM1, die den Prozess des Hochfahrens protokollieren. Das erste Datum stellt den Zeitstempel dar, mit „Ausg.“ werden ausgehende Messages und mit „Eing.“ werden eingehende Messages gekennzeichnet, „an MFM (x)“ spezifiziert die Adresse und „von MFM (x)“ den Absender der Nachricht. Der Text nach dem Strich beinhaltet eine kurze Erklärung des Nachrichtentyps. Dahinter, in eckigen Klammer, ist der aktuelle Zustand der MFM-ZN zu sehen.

```
76914 Ausg.: an MFM 4 - Anmeld. beim Vorg. [1]
76915 Ausg.: an MFM 0 - Anmeld. beim Vorg. [1]
76915 Ausg.: an MFM 7 - Anmeld. beim Vorg. [1]
76915 Ausg.: an MFM 7 - Anmeld. beim Nachf.[1]
76915 Ausg.: an MFM 2 - Anmeld. beim Nachf.[1]
76915 Ausg.: an MFM 5 - Anmeld. beim Nachf.[1]
76915 Ausg.: an MFM 4 - HB2 versendet [1]
76915 Ausg.: an MFM 0 - HB2 versendet [1]
76915 Ausg.: an MFM 7 - HB2 versendet [1]
76925 Eing.: von MFM 2 - Anmeld. von Nachf. [1]
76925 Ausg.: an MFM 4 - HB2 versendet [1, 2]
76925 Ausg.: an MFM 0 - HB2 versendet [1, 2]
76925 Ausg.: an MFM 7 - HB2 versendet [1, 2]
76925 Eing.: von MFM 2 - HB2 erhalten [1, 2]
76956 Eing.: von MFM 2 - HB2 erhalten [1, 2]
76956 Eing.: von MFM 4 - Anmeld. von Vorg. [1, 2]
76956 Ausg.: an MFM 4 - HB2 versendet [1, 2]
76956 Ausg.: an MFM 0 - HB2 versendet [1, 2]
76956 Ausg.: an MFM 7 - HB2 versendet [1, 2]
76956 Eing.: von MFM 5 - Anmeld. von Nachf. [1, 2]
76956 Ausg.: an MFM 4 - HB2 versendet [1, 2, 5]
76957 Ausg.: an MFM 0 - HB2 versendet [1, 2, 5]
76957 Ausg.: an MFM 7 - HB2 versendet [1, 2, 5]
76956 Eing.: von MFM 5 - HB2 erhalten [1, 2, 5]
76957 Eing.: von MFM 7 - Anmeld. von Nachf. [1, 2, 5]
76957 Eing.: von MFM 7 - Anmeld. von Vorg. [1, 2, 5]
76957 Ausg.: an MFM 4 - HB2 versendet [1, 2, 5]
76957 Ausg.: an MFM 0 - HB2 versendet [1, 2, 5]
76957 Ausg.: an MFM 7 - HB2 versendet [1, 2, 5]
76957 Ausg.: an MFM 4 - HB2 versendet [1, 7, 2, 5]
76957 Ausg.: an MFM 0 - HB2 versendet [1, 7, 2, 5]
76957 Ausg.: an MFM 7 - HB2 versendet [1, 7, 2, 5]
76957 Eing.: von MFM 7 - HB2 erhalten [1, 7, 2, 5]
76987 Ausg.: an MFM 4 - HB2 versendet [1, 7, 2, 5, 3]
76988 Ausg.: an MFM 0 - HB2 versendet [1, 7, 2, 5, 3]
76988 Ausg.: an MFM 7 - HB2 versendet [1, 7, 2, 5, 3]
76987 Eing.: von MFM 5 - HB2 erhalten [1, 7, 2, 5, 3]
76988 Eing.: von MFM 5 - HB2 erhalten [1, 7, 2, 5, 3]
76988 Ausg.: an MFM 4 - HB2 versendet [1, 7, 2, 5, 6, 3]
76988 Ausg.: an MFM 0 - HB2 versendet [1, 7, 2, 5, 6, 3]
```

Anhang

76988	Ausg.:	an	MFM 7	-	HB2	versendet	[1, 7, 2, 5, 6, 3]
76988	Eing.:	von	MFM 5	-	HB2	erhalten	[1, 7, 2, 5, 6, 3]
76988	Eing.:	von	MFM 2	-	HB2	erhalten	[1, 7, 2, 3, 5, 6]
76988	Eing.:	von	MFM 7	-	HB2	erhalten	[1, 7, 2, 3, 5, 6]
77019	Eing.:	von	MFM 7	-	HB2	erhalten	[1, 7, 2, 5, 3, 6]
77019	Ausg.:	an	MFM 4	-	HB2	versendet	[1, 7, 2, 5, 3, 6, 4]
77019	Ausg.:	an	MFM 0	-	HB2	versendet	[1, 7, 2, 5, 3, 6, 4]
77019	Ausg.:	an	MFM 7	-	HB2	versendet	[1, 7, 2, 5, 3, 6, 4]
77019	Eing.:	von	MFM 5	-	HB2	erhalten	[1, 7, 2, 5, 3, 6, 4]
77019	Eing.:	von	MFM 2	-	HB2	erhalten	[1, 7, 2, 5, 3, 4, 6]
77049	Eing.:	von	MFM 5	-	HB2	erhalten	[1, 7, 2, 5, 3, 4, 6]
77049	Ausg.:	an	MFM 4	-	HB2	versendet	[1, 7, 2, 5, 3, 4, 6, 8]
77049	Ausg.:	an	MFM 0	-	HB2	versendet	[1, 7, 2, 5, 3, 4, 6, 8]
77049	Ausg.:	an	MFM 7	-	HB2	versendet	[1, 7, 2, 5, 3, 4, 6, 8]
77049	Eing.:	von	MFM 5	-	HB2	erhalten	[1, 7, 2, 5, 3, 4, 6, 8]
77049	Eing.:	von	MFM 2	-	HB2	erhalten	[1, 7, 2, 5, 3, 4, 6, 8]
77049	Eing.:	von	MFM 2	-	HB2	erhalten	[1, 7, 2, 5, 3, 8, 4, 6]
77049	Eing.:	von	MFM 7	-	HB2	erhalten	[1, 7, 2, 5, 3, 8, 4, 6]
77050	Eing.:	von	MFM 7	-	HB2	erhalten	[1, 7, 2, 5, 6, 3, 8, 4]
77080	Eing.:	von	MFM 5	-	HB2	erhalten	[1, 7, 2, 5, 6, 3, 8, 4]
77080	Eing.:	von	MFM 2	-	HB2	erhalten	[1, 7, 2, 5, 6, 3, 8, 4]
77080	Eing.:	von	MFM 2	-	HB2	erhalten	[1, 7, 2, 5, 6, 3, 8, 4]
77080	Eing.:	von	MFM 7	-	HB2	erhalten	[1, 7, 2, 5, 3, 6, 4, 8]
77112	Eing.:	von	MFM 2	-	HB2	erhalten	[1, 7, 2, 5, 3, 6, 4, 8]
77112	Eing.:	von	MFM 7	-	HB2	erhalten	[1, 7, 2, 5, 3, 4, 6, 8]

Protokoll 2: Ausfall eines MFM

Die Beschreibung der Protokollstruktur ist dem Anhang weiter oben zu entnehmen, die folgende Liste enthält die an MFM1 eingehenden und ausgehenden Nachrichten und beginnt mit dem Eingang des Heartbeat Klasse 3.

```
92084 Eing.: HB3 erhalten
92084 Ausg.: an MFM 4 - HB2 versendet [1]
92084 Ausg.: an MFM 0 - HB2 versendet [1]
92084 Ausg.: an MFM 7 - HB2 versendet [1]
92114 Ausg.: an MFM 4 - HB2 versendet [1, 2]
92114 Ausg.: an MFM 0 - HB2 versendet [1, 2]
92114 Ausg.: an MFM 7 - HB2 versendet [1, 2]
92114 Eing.: von MFM 2 - HB2 erhalten [1, 2]
92114 Eing.: von MFM 2 - HB2 erhalten [1, 2]
92114 Eing.: von MFM 2 - HB2 erhalten [1, 2]
92114 Eing.: von MFM 2 - HB2 erhalten [1, 2]
92114 Ausg.: an MFM 4 - HB2 versendet [1, 7, 2]
92114 Ausg.: an MFM 0 - HB2 versendet [1, 7, 2]
92114 Ausg.: an MFM 7 - HB2 versendet [1, 7, 2]
92114 Eing.: von MFM 7 - HB2 erhalten [1, 7, 2]
92115 Eing.: von MFM 7 - HB2 erhalten [1, 7, 2]
92115 Eing.: von MFM 7 - HB2 erhalten [1, 7, 2]
92115 Eing.: von MFM 7 - HB2 erhalten [1, 7, 2]
92145 Eing.: von MFM 2 - HB2 erhalten [1, 7, 2]
92145 Ausg.: an MFM 4 - HB2 versendet [1, 7, 2, 3]
92145 Ausg.: an MFM 0 - HB2 versendet [1, 7, 2, 3]
92145 Ausg.: an MFM 7 - HB2 versendet [1, 7, 2, 3]
92145 Eing.: von MFM 2 - HB2 erhalten [1, 7, 2, 3]
92145 Eing.: von MFM 7 - HB2 erhalten [1, 7, 2, 3]
92145 Eing.: von MFM 7 - HB2 erhalten [1, 7, 2, 3]
92175 Ausg.: an MFM 4 - HB2 versendet [1, 7, 2, 3, 4]
92175 Ausg.: an MFM 0 - HB2 versendet [1, 7, 2, 3, 4]
92176 Ausg.: an MFM 7 - HB2 versendet [1, 7, 2, 3, 4]
92175 Eing.: von MFM 2 - HB2 erhalten [1, 7, 2, 3, 4]
92176 Ausg.: an MFM 4 - HB2 versendet [1, 7, 2, 3, 8, 4]
92176 Ausg.: an MFM 0 - HB2 versendet [1, 7, 2, 3, 8, 4]
92176 Ausg.: an MFM 7 - HB2 versendet [1, 7, 2, 3, 8, 4]
92176 Eing.: von MFM 2 - HB2 erhalten [1, 7, 2, 3, 8, 4]
92176 Eing.: von MFM 7 - HB2 erhalten [1, 7, 2, 3, 8, 4]
92206 Eing.: von MFM 2 - HB2 erhalten [1, 7, 2, 3, 8, 4]
92206 Eing.: von MFM 7 - HB2 erhalten [1, 7, 2, 3, 8, 4]
92237 Eing.: von MFM 2 - HB2 erhalten [1, 7, 2, 3, 8, 4]
92237 Eing.: von MFM 7 - HB2 erhalten [1, 7, 2, 3, 4, 8]
92237 Eing.: von MFM 7 - HB2 erhalten [1, 7, 2, 3, 8, 4]
```

Protokoll 3: Wegfindung

Dieses Protokoll zeigt den Weg eines Zielobjektes mit den Zielen 3 - 5 - 1 - 3. Das erste Datum ist der Zeitstempel des Ereignisses. Die Abkürzung „Pas.“ bedeutet, dass das Objekt das MFM auf dem Weg zu seinem Ziel nur passiert. „Er.“ heißt das Objekt erreicht ein Ziel. Mit „MFM: (x)“ wird der Aufenthaltsort des Objektes spezifiziert.

```
99791 Pas.: MFM: 1
99815 Pas.: MFM: 2
99852 Er.: MFM: 3 Station 1
Alternativ-MFM: 0 ausgewaehlt!
99880 Pas.: MFM: 8
Alternativ-MFM: 0 ausgewaehlt!
99913 Pas.: MFM: 3
Alternativ-MFM: 0 ausgewaehlt!
99950 Pas.: MFM: 8
Alternativ-MFM: 0 ausgewaehlt!
99975 Pas.: MFM: 3
Alternativ-MFM: 0 ausgewaehlt!
00011 Pas.: MFM: 8
Alternativ-MFM: 0 ausgewaehlt!
00046 Pas.: MFM: 3
Alternativ-MFM: 0 ausgewaehlt!
00072 Pas.: MFM: 8
00107 Pas.: MFM: 3
00139 Pas.: MFM: 4
00168 Pas.: MFM: 1
00203 Er.: MFM: 5 Station 2
00230 Pas.: MFM: 3
00261 Pas.: MFM: 4
00300 Er.: MFM: 1 Station 3
00325 Pas.: MFM: 5
00364 Er.: MFM: 3 Endstation!
```

Zu Beginn des Transportes ist MFM5 noch nicht aktiviert. Deshalb wählt die Wegfindung das Alternativziel aus der Liste der Ziel-MFM. Da kein Alternativ-MFM eingegeben wurden, wird die Standardeingabe Null ausgewählt, was in diesem Fall bedeutet, dass das Objekt in eine Kreisbewegung zwischen MFM8 und MFM3 geschickt wird. Sobald MFM5 hochfährt und in den Nachfolgerlisten erscheint, wird das Objekt auf den richtigen Weg gebracht. Die nachfolgende Liste stellt die an MFM 1 ankommenden und abgehenden Nachrichten dar:

```
99301 Ausg.: an MFM 4 - Anmeld. beim Vorg. [1]
99301 Ausg.: an MFM 0 - Anmeld. beim Vorg. [1]
99301 Ausg.: an MFM 7 - Anmeld. beim Vorg. [1]
```

Anhang

99301 Ausg.: an MFM 7 - Anmeld. beim Nachf. [1]
99301 Ausg.: an MFM 2 - Anmeld. beim Nachf. [1]
99301 Ausg.: an MFM 5 - Anmeld. beim Nachf. [1]
99301 Ausg.: an MFM 4 - HB2 versendet [1]
99301 Ausg.: an MFM 0 - HB2 versendet [1]
99301 Ausg.: an MFM 7 - HB2 versendet [1]
99311 Eing.: von MFM 2 - Anmeld. von Nachf. [1]
99311 Ausg.: an MFM 4 - HB2 versendet [1, 2]
99311 Ausg.: an MFM 0 - HB2 versendet [1, 2]
99312 Ausg.: an MFM 7 - HB2 versendet [1, 2]
99311 Eing.: von MFM 2 - HB2 erhalten [1, 2]
99343 Eing.: von MFM 2 - HB2 erhalten [1, 2]
99343 Eing.: von MFM 4 - Anmeld. von Vorg. [1, 2]
99343 Ausg.: an MFM 4 - HB2 versendet [1, 2]
99343 Ausg.: an MFM 0 - HB2 versendet [1, 2]
99343 Ausg.: an MFM 7 - HB2 versendet [1, 2]
99343 Eing.: von MFM 7 - Anmeld. von Nachf. [1, 2]
99343 Eing.: von MFM 7 - Anmeld. von Vorg. [1, 2]
99343 Ausg.: an MFM 4 - HB2 versendet [1, 2]
99343 Ausg.: an MFM 0 - HB2 versendet [1, 2]
99343 Ausg.: an MFM 7 - HB2 versendet [1, 2]
99343 Ausg.: an MFM 4 - HB2 versendet [1, 7, 2]
99343 Ausg.: an MFM 0 - HB2 versendet [1, 7, 2]
99343 Ausg.: an MFM 7 - HB2 versendet [1, 7, 2]
99343 Ausg.: an MFM 7 - HB2 versendet [1, 7, 2]
99343 Eing.: von MFM 7 - HB2 erhalten [1, 7, 2]
99375 Ausg.: an MFM 4 - HB2 versendet [1, 7, 2, 3]
99375 Ausg.: an MFM 0 - HB2 versendet [1, 7, 2, 3]
99375 Ausg.: an MFM 7 - HB2 versendet [1, 7, 2, 3]
99375 Eing.: von MFM 2 - HB2 erhalten [1, 7, 2, 3]
99375 Eing.: von MFM 7 - HB2 erhalten [1, 7, 2, 3]
99406 Ausg.: an MFM 4 - HB2 versendet [1, 7, 2, 3, 4]
99406 Ausg.: an MFM 0 - HB2 versendet [1, 7, 2, 3, 4]
99406 Ausg.: an MFM 7 - HB2 versendet [1, 7, 2, 3, 4]
99406 Eing.: von MFM 2 - HB2 erhalten [1, 7, 2, 3, 4]
99437 Ausg.: an MFM 4 - HB2 versendet [1, 7, 2, 3, 8, 4]
99437 Ausg.: an MFM 0 - HB2 versendet [1, 7, 2, 3, 8, 4]
99437 Ausg.: an MFM 7 - HB2 versendet [1, 7, 2, 3, 8, 4]
99437 Eing.: von MFM 2 - HB2 erhalten [1, 7, 2, 3, 8, 4]
99437 Eing.: von MFM 2 - HB2 erhalten [1, 7, 2, 3, 8, 4]
99437 Eing.: von MFM 7 - HB2 erhalten [1, 7, 2, 3, 8, 4]
99478 Eing.: von MFM 2 - HB2 erhalten [1, 7, 2, 3, 8, 4]
99478 Eing.: von MFM 7 - HB2 erhalten [1, 7, 2, 3, 4, 8]
99508 Eing.: von MFM 7 - HB2 erhalten [1, 7, 2, 3, 8, 4]
99791 Eing.: Obj. mit Ziel MFM 3 pass. MFM [1, 7, 2, 3, 8, 4]
99944 Eing.: von MFM 5 - Anmeld. von Nachf. [1, 7, 2, 3, 8, 4]
99944 Ausg.: an MFM 4 - HB2 versendet [1, 7, 2, 3, 8, 4, 5]
99944 Ausg.: an MFM 0 - HB2 versendet [1, 7, 2, 3, 8, 4, 5]
99944 Ausg.: an MFM 7 - HB2 versendet [1, 7, 2, 3, 8, 4, 5]
99944 Eing.: von MFM 5 - HB2 erhalten [1, 7, 2, 3, 8, 4, 5]
00007 Ausg.: an MFM 4 - HB2 versendet [1, 7, 2, 3, 8, 4, 5, 6]
00007 Ausg.: an MFM 0 - HB2 versendet [1, 7, 2, 3, 8, 4, 5, 6]
00007 Ausg.: an MFM 7 - HB2 versendet [1, 7, 2, 3, 8, 4, 5, 6]
00007 Eing.: von MFM 5 - HB2 erhalten [1, 7, 2, 3, 8, 4, 5, 6]
00007 Eing.: von MFM 7 - HB2 erhalten [1, 7, 2, 3, 8, 4, 5, 6]
00078 Eing.: von MFM 2 - HB2 erhalten [1, 7, 2, 3, 8, 4, 5, 6]
00078 Eing.: von MFM 7 - HB2 erhalten [1, 7, 2, 3, 8, 4, 5, 6]
00078 Eing.: von MFM 7 - HB2 erhalten [1, 7, 2, 3, 8, 4, 5, 6]
00138 Eing.: von MFM 2 - HB2 erhalten [1, 7, 2, 3, 8, 4, 5, 6]
00168 Eing.: Obj. mit Ziel MFM 5 pass. MFM [1, 7, 2, 3, 8, 4, 5, 6]
00300 Eing.: Obj. mit Ziel MFM 1 pass. MFM [1, 7, 2, 3, 8, 4, 5, 6]

Anhang C: Protokolle der Umsetzung

Tabelle 1: ZN der MFM nach dem Hochfahren

ZN								
Nachfolger-MFM	MFM 1	MFM 2	MFM 3	MFM 4	MFM 5	MFM 6	MFM 7	MFM 8
1	0	11	8	6	10	12	2	10
2	2	0	10	8	12	14	4	12
3	4	3	0	10	2	4	6	2
4	6	5	2	0	4	6	8	4
5	2	13	10	8	0	2	4	12
6	4	15	12	10	2	0	6	14
7	2	13	10	8	12	14	0	12
8	6	5	2	12	4	6	8	0

Tabelle 2: ZN der MFM nach dem Ausfall von MFM 2

ZN								
Nachfolger-MFM	MFM 1	MFM 2	MFM 3	MFM 4	MFM 5	MFM 6	MFM 7	MFM 8
1	0	X	8	6	10	12	2	10
2	255	X	255	255	255	255	255	255
3	4	X	0	10	2	4	6	2
4	6	X	2	0	4	6	8	4
5	2	X	10	8	0	2	4	12
6	4	X	12	10	2	0	6	14
7	2	X	10	8	12	14	0	12
8	6	X	2	12	4	6	8	0

Lebenslauf

Name: Kai Lorentz
Geburtsdatum und -ort: 17.12.1969 in Braunschweig
Familienstand: ledig
Eltern und Geschwister: Ingrid Lorentz; eine Schwester, Inga Weimar-Lorentz

Schulbildung:

1976 - 1980 Grundschole Destedt am Elm
1980 - 1982 Orientierungsstufe Friesenstraße in Braunschweig
1982 - 1989 Gymnasium Christophorusschule in Braunschweig; Abschluß Abitur
1986 - 1987 Schüleraustausch Amerika
Stipendiat an der Hill School, Pottstown

Wehrdienst

06/89 - 08/90 1./25 Panzerartillerie

Studium

10/90 - 11/95 Wirtschaftsingenieurwesen mit Fachrichtung Maschinenbau an der Technischen Universität Carolo Wilhelmina zu Braunschweig und an dem Politecnico di Milano, Italien; Abschluss Diplom

Beruflicher Werdegang

12/95 - 05/98 Projektingenieur bei der Firma Lorentz Mühlenbau GmbH, Braunschweig.
06/98 - 12/01 Mitarbeiter am Fraunhofer-Institut für Fabrikbetrieb und -automatisierung IFF, Magdeburg; zuletzt Abteilungsleiter
seit 01/02 Leiter des Centers Verteilte Systeme am IAF der Otto-von-Guericke-Universität Magdeburg
ebenfalls seit 07/00 Geschäftsführer der IAONA Europe e.V.