

Lehrstuhl für
Fördertechnik Materialfluss Logistik
der Technischen Universität München

Konzeption und Entwicklung von Modulen für das Internet der Dinge

Dipl.-Ing. (Univ.) Florian Alexander Kuzmany

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

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzende:

Univ.-Prof. Dr.-Ing. Birgit Vogel-Heuser

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing. Willibald A. Günthner
2. Univ.-Prof. Dr. med. Michael ten Hompel,
Technische Universität Dortmund

Die Dissertation wurde am 29.10.2009 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 11.02.2010 angenommen.

Herausgegeben von:

Univ.-Prof. Dr.-Ing. Dipl.-Wi.-Ing. Willibald A. Günthner
fml – Lehrstuhl für Fördertechnik Materialfluss Logistik
Technische Universität München

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, der Entnahme von Abbildungen, der Wiedergabe auf fotomechanischem oder ähnlichen Wege und der Speicherung in Datenverarbeitungsanlagen bleiben – auch bei nur auszugsweiser Verwendung – vorbehalten.

Layout und Satz: Florian Kuzmany

Copyright © Florian Kuzmany 2010

ISBN: 978-3-941702-06-6

Printed in Germany 2010

Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Assistent am Lehrstuhl für Fördertechnik Materialfluss Logistik (fml) der Technischen Universität München. Ich danke Herrn Prof. Dr. Willibald A. Günthner für das in mich gesetzte Vertrauen, für die Unterstützung bei meiner Forschungsarbeit und für die mir gewährte Freiheit, mich auf das Thema Internet der Dinge zu konzentrieren zu können. Bei Herrn Prof. Dr. Michael ten Hompel bedanke ich mich für die Übernahme des Koreferats bei der Verteidigung meiner Dissertation und die gute Zusammenarbeit mit dem Fraunhofer Institut für Materialfluss und Logistik (IML). Bei Frau Prof. Dr. Birgit Vogel-Heuser bedanke ich mich für die Übernahme des Vorsitzes des Prüfungsausschusses.

Mein Dank gilt ebenso dem Bundesministerium für Bildung und Forschung für die Förderung des Forschungsprojekts Internet der Dinge und den beteiligten Industriepartnern für die stets gute Zusammenarbeit.

Ganz besonders bedanke ich mich bei Herrn Razvan Chisu, der mir in fachlichen Fragen stets ein kompetenter Berater war. Ich möchte mich auch bei meinen Bürokollegen Herrn Michael Schipplick und Herrn Peter Tenerowicz bedanken. Sie haben zu einer sehr freundlichen und konzentrierten Arbeitsatmosphäre beigetragen.

Vor allem zu Dank verpflichtet bin ich meinen Eltern Günther und Christl Kuzmany, welche mich immer unterstützt und mir in allen Lebenslagen Rückhalt gegeben haben. Obwohl mein Vater den Abschluss meiner Arbeit leider nicht erleben durfte, hat er doch durch die von ihm vermittelten Wertvorstellungen den Grundstein dafür gelegt, dass ich dieses Ziel erreichen konnte. Meiner Mutter danke ich dafür, dass Besuche in meinem Elternhaus mich stets zur Ruhe kommen ließen und ich dort immer wieder die Energie tanken konnte, um meine Arbeit fortzusetzen.

Schließlich möchte ich mich noch bei meinem Bruder Stefan Kuzmany und meinem guten Freund Dr. Mark Perić für die abschließende Korrektur bedanken.

München, im Februar 2010


Florian Kuzmany

Kurzzusammenfassung

Konzeption und Entwicklung von Modulen für das Internet der Dinge

Florian Kuzmany

Weltweit genießt Deutschland den Ruf, im Maschinen- und Anlagenbau und in der Automatisierungstechnik qualitativ hochwertige Spitzentechnologie zu produzieren. In der Vergangenheit ermöglichte es die hochautomatisierte Massenproduktion von Gütern, im Hochlohnland Deutschland der stärker werdenden weltweiten Konkurrenz zu begegnen. Mittlerweile stellen die Kunden jedoch höhere Anforderungen. Sie wollen individuelle Produkte zu einem niedrigen Preis und diese in kürzester Zeit geliefert bekommen. Die gewünschten zahlreichen Produktvarianten erhöhen die Anzahl der innerbetrieblichen Transporte und führen zu nicht prognostizierbaren Nachfrageschwankungen.

Klassische Materialflusssteuerungen haben jedoch längst einen Komplexitätsgrad erreicht, der nur noch schwer zu handhaben ist. Sie lassen sich nur mit großem Aufwand verändern. Auf unerwartete Anforderungen sind sie nicht vorbereitet.

Das Internet, das über eine Milliarde Teilnehmer miteinander vernetzt und trotzdem flexibel und schnell erweiterbar geblieben ist, ist das Vorbild für ein Internet der Dinge, in dem, ähnlich wie die Router eine E-Mail weiterleiten, die Fördertechnikmodule selbstständig über den Weg der Transporteinheiten bestimmen.

Dazu wird das Materialflusssystem nach den Vorgaben der funktionsorientierten Modularisierung strukturiert und die Module werden mit intelligenten Softwareagenten versehen. Ein Modulaufbau in Schichten, ein Vererbungsmodell für Modulfunktionen und eine Vorgehensweise zur Strategieentwicklung erlauben die schnelle Konzeption und Entwicklung dieser Module. So gefundene Lösungen sind in hohem Maße wiederverwendbar. Eine vergleichende Untersuchung der heute ablaufenden und in Zukunft möglichen Schritte bei der Realisierung eines Materialflusssystems zeigt die Vorteile des neuen Konzepts: Das Internet der Dinge-Materialflusssystem ist leicht erweiterbar, wandelbar und kann auf neue Anforderungen selbstständig reagieren. Es ist ein System, das auch auf das Unerwartete vorbereitet ist.

Summary

Design and Development of Modules for the Internet of Things

Florian Kuzmany

Germany has established a worldwide good reputation by producing high quality technologies for automated production machines and plants. The highly automated production of goods empowered Germany to compete with low-wage countries in the rest of the world. Meanwhile, customers have higher expectations. They demand individual products at low prices and fast delivery. The many versions of any product increase the number of transports in a production plant and lower the predictability of demand for each product.

Classic material flow systems have reached a level of complexity which is difficult to handle. Modifications cause high costs because these systems are not primed for unexpected requirements. The internet which has remained flexible and extensible in spite of its size serves as a model for an internet of things, where packets are transported by autonomous conveying modules, just like the email is forwarded independently by the routers of the internet. For that purpose the material flow system is divided into modules using the function oriented modularization approach and an intelligent software agent is assigned to each module. The design and development of these modules is supported by a multi layer structure, an inheritance model for module functions and a procedural method for the development of material flow strategies. Solutions developed in this way are highly reusable. A comparative analysis of present and future activities necessary for the realization of a material flow system shows the benefits of the new concept: The internet of things is easy to extend and to modify and can autonomously react to new requirements. It is primed for the unexpected.

Inhaltsverzeichnis

1	Einführung	1
1.1	Ausgangssituation	1
1.2	Anforderungen an zukünftige Materialflusssysteme	3
1.2.1	Flexibilität	4
1.2.2	Wandlungsfähigkeit und Wandelbarkeit	5
1.2.3	Komplexitätsreduktion	6
1.3	Zielsetzung und Vorgehensweise	8
2	Stand der Technik und Forschung	11
2.1	Materialflusssteuerung	11
2.1.1	Aufbau herkömmlicher Materialflusssteuerungen	11
2.1.2	Defizite aktueller Steuerungssysteme	15
2.2	Erstellung, Betrieb und Erweiterung eines Materialflusssystems	18
2.2.1	Bisherige Vorgehensweise	18
2.2.2	Optimierungspotenzial	28
2.3	Verfügbare Technologien	33
2.3.1	Automatisierungstechnik	33
2.3.2	Radio Frequenz Identifikation	37
2.3.3	Internet und TCP/IP	40
2.3.4	Multiagentensysteme	44
2.4	Stand der Forschung	48
3	Das Internet der Dinge	53
3.1	Die Vision vom Internet der Dinge	53
3.2	Die Bausteine des Internet der Dinge	57
3.2.1	Transporteinheiten	59

3.2.2	Module	60
3.2.3	Dienste	60
4	Konzeption und Entwicklung von Modulen	61
4.1	Gestaltungsmethodik und Vorgehensweise	61
4.1.1	Entwicklungsmethodik für mechatronische Systeme	61
4.1.2	Methodik zur Modulentwicklung für das Internet der Dinge	65
4.1.3	Funktionsorientierte Modularisierung	68
4.1.4	Funktionsklassen für Module	73
4.2	Grundsätzlicher Aufbau von Modulen	77
4.2.1	Schnittstellen eines Moduls	80
4.2.2	Steuerungssoftware	84
4.2.3	Steuerungshardware	87
4.2.4	Additive Komponenten	94
4.3	Entwicklung der Modulsteuerung	96
4.3.1	Vererbungsmodell für Modulfunktionen	96
4.3.2	Entwicklung von Materialflussstrategien	104
4.3.3	Konfigurierung / Parametrierung	118
4.4	Funktionsabsicherung	121
4.4.1	Emulation	121
4.4.2	Visualisierung	125
4.4.3	Fehlersuche und -behebung	128
5	Realisierung am Lehrstuhl für Fördertechnik Materialfluss Logistik	133
5.1	Fördertechnische Einrichtungen und Layout	133
5.2	Szenario	137
5.3	Realisierte Module	140
5.3.1	Modulklassen und -funktionen	141

5.3.2	Auftragsdisposition	142
5.3.3	Steuerungshardware und -software	144
5.3.4	Schnittstellen und Kommunikation	147
6	Auswirkungen des Internet der Dinge-Konzepts	153
6.1	Planungsphase	153
6.2	Realisierungsphase	154
6.3	Inbetriebnahme- /Hochlaufphase	156
6.4	Betrieb	158
6.5	Erweiterung / Modernisierung	159
7	Zusammenfassung und Ausblick	161
7.1	Zusammenfassung der Ergebnisse	161
7.2	Ausblick	165
8	Literaturverzeichnis	167

Abbildungsverzeichnis

Abb. 1.1: Aufwandsverschiebung	9
Abb. 1.2: Vorgehensweise.....	10
Abb. 2.1: Hardwarekonfiguration und Steuerungsarchitektur der Materialflusssteuerung nach VDMA15276 [VDMA-15276]	12
Abb. 2.2: Hardwarekonfiguration und Steuerungsarchitektur in einem Lebensmittellager (Quelle: Stöcklin Logistik GmbH).....	14
Abb. 2.3: Steuerungspyramide nach Bender[Ben-08a] mit Ebenen nach VDMA 15276.....	15
Abb. 2.4: Steuerungspyramide: Beispielhafter Schnitt entlang der Anlagenfunktionen	17
Abb. 2.5: Tätigkeiten von Planung bis Betrieb einer Materialflussanlage und die wichtigsten Meilensteine (Quelle: Forschungsprojekt Internet der Dinge) 19	
Abb. 2.6: Zeitlicher Ablauf der Tätigkeiten in der Planungsphase (qualitativ).....	20
Abb. 2.7: Zeitlicher Ablauf der Tätigkeiten in der Realisierungsphase (qualitativ)	21
Abb. 2.8: Zeitlicher Ablauf der Tätigkeiten in der Inbetriebnahmephase (qualitativ) .	24
Abb. 2.9: Die Erweiterung / Modernisierung im Lebenszyklus eines Materialflusssystems	27
Abb. 2.10: Fehlerverursachung und Fehlerentdeckung bei Entwicklungsprojekten [Sch-08b]	28
Abb. 2.11: Steuerung.....	33
Abb. 2.12: Regelung.....	33
Abb. 3.1: Eine neue Dimension der Kommunikation [ITU-05]	54
Abb. 3.2: Analogie zwischen Internet und Intra-logistik	56
Abb. 3.3: Die Grundeinheiten des Internet der Dinge sind kooperierende Entitäten.	58
Abb. 4.1: V-Modell als Makrozyklus zur Entwicklung mechatronischer Systeme [VDI- 2206].....	62
Abb. 4.2: Allgemeiner Problemlösungszyklus [Dae-94].....	64
Abb. 4.3: Angepasstes V-Modell für die Entwicklung und Integration von Modulen für das Internet der Dinge	65
Abb. 4.4: Vorgehensweise bei der Modulentwicklung	67
Abb. 4.5: Methode der Aufgliederung und Verknüpfung zur Problem- und Systemstrukturierung [VDI-2221].....	68
Abb. 4.6: Allgemeine Eigenschaften und Beschreibungsmerkmale eines Moduls [Hil- 05].....	69
Abb. 4.7: Geeignete Modularisierungsebene.....	71
Abb. 4.8: Modulaufbau und Schnittstellen	77

Abb. 4.9: Trennung von logistischer Funktion und konkreter Ausprägung einer Technologie durch eine Middleware	83
Abb. 4.10: Aufbau der Steuerung eines Moduls als hybrider Softwareagent.....	84
Abb. 4.11: Mögliche Hardwarekonzepte im Internet der Dinge.....	91
Abb. 4.12: Vererbungsmodell für Module	97
Abb. 4.13: Entitätsklasse	98
Abb. 4.14: Modulklassse	99
Abb. 4.15: Funktionsklasse.....	100
Abb. 4.16: Standardgeräteklasse	101
Abb. 4.17: Geräteklasse	102
Abb. 4.18: Maschinennahe Steuerung	103
Abb. 4.19: Deadlocksituation	105
Abb. 4.20: Vorgehensweise bei der Simulation von Logistiksystemen [VDI-3633].	107
Abb. 4.21: Taxonomie von Lösungen des Koordinationsproblems [Huh-99]	108
Abb. 4.22: Vorgehensweise bei der Entwicklung von Strategien.....	110
Abb. 4.23: Querverschiebewägen	113
Abb. 4.24: Blockadeauflösung.....	115
Abb. 4.25: Durchsatzsteigerung durch Anwendung von Verhaltensregeln (einzeln)	117
Abb. 4.26: Durchsatzsteigerung durch Anwendung von Verhaltensregeln (kombiniert)	117
Abb. 4.27: Emulation der Modulhardware	123
Abb. 5.1: Versuchshalle des Lehrstuhls fml.....	133
Abb. 5.2: Materialflusssystem am Lehrstuhl fml (schematisch)	134
Abb. 5.3: Wege, Ziele und Entscheidungspunkte im Materialflusssystem des Lehrstuhls fml	138
Abb. 5.4: Auftragsdisposition	143
Abb. 5.5: Beckhoff IPC CX9010	145
Abb. 5.6: Konfigurationsdatei für eine reale EHB-Katze	148
Abb. 5.7: Konfigurationsdatei für den Emulator einer EHB-Katze.....	150
Abb. 5.8: Kommunikation zwischen strategischer und operativer Steuerung.....	151
Abb. 6.1: Erwartete Veränderung des Zeitaufwands in der Planungsphase(qualitativ)	153
Abb. 6.2: Erwartete Veränderung des Zeitaufwands in der Realisierungsphase(qualitativ).....	155
Abb. 6.3: Erwartete Veränderung des Zeitaufwands in der Inbetriebnahme- /Hochlaufphase (qualitativ)	157

Tabellenverzeichnis

Tab. 2.1: Zykluszeiten von unterschiedlichen Industrial Ethernet Lösungen [Koh-07]	43
Tab. 4.1: Gegenüberstellung SPS - PC	90
Tab. 4.2: Additive Komponenten und ihre Verwendbarkeit in versch. Lebenszyklusphasen	95
Tab. 5.1: Am Lehrstuhl fml realisierte Modulfunktionen.....	142

Abkürzungsverzeichnis

Abkürzung	Bedeutung
EHB	Elektrohängebahn
ERP	Enterprise Resource Planning System
GLT	Großladungsträger
IPC	Industrie Personal Computer
KLT	Kleinladungsträger
LVS	Lagerverwaltungssystem
MFR	Materialflussrechner
OOP	Objektorientierte Programmierung
QVW	Querverschiebewagen
RBG	Regalbediengerät
RFID	Radio Frequenz Identifikation
SPS	Speicherprogrammierbare Steuerung
TE	Transporteinheit
VDA	Verband der Automobilindustrie
WE	Wareneingang

1 Einführung

1.1 Ausgangssituation

Noch nie war es für Unternehmen so schwer, im Wettbewerb zu bestehen, wie heute. Der freie Fluss von Waren, Dienstleistungen und Kapital über alle Landesgrenzen hinweg - gemeinhin als Globalisierung bezeichnet - hat den Konkurrenzdruck unheimlich erhöht: Dienstleister und produzierendes Gewerbe kämpfen nicht mehr nur mit ihren Nachbarn um Marktanteile. Heute wird der Kampf um die Kunden weltweit ausgetragen. Bisherige Schwellenländer wandeln sich zu Industrienationen, der Aufstieg Chinas ist dafür nur ein Beispiel. Indien ist bereits auf dem Sprung, andere werden folgen. Mehr denn je gilt für das produzierende Gewerbe in Deutschland: Seine Produkte müssen besser sein als jene der Konkurrenz, sie müssen günstiger sein, und sie müssen schneller zur Verfügung stehen. Wer langfristig auf dem Markt bestehen will, muss in der Lage sein, sein Angebot in immer kürzeren Zeitabständen der Marktentwicklung anzupassen. Nur mit neuen, technologisch weiter entwickelten und qualitativ hochwertigeren Produkten wird er seine Marktpositionen halten oder verbessern können [Bir-07].

Die im weltweiten Vergleich relativ hohen deutschen Lohnkosten zwingen hier ansässige Unternehmen, diese Produkte in hoch automatisierten Produktionssystemen herzustellen. Die kostengünstige Massenproduktion war lange Zeit lukrativ, doch sie hat sich überlebt: Immer mehr Menschen besitzen die gleichen oder ähnliche Gegenstände [Fou-94]. Der moderne Mensch sucht jedoch den Distinktionsgewinn, er will sich von der Masse unterscheiden und abheben. Gerade kaufkräftige Konsumenten versuchen, ihre Persönlichkeit durch eine individuelle Produktwahl zu demonstrieren [Rei-06].

Dieser Wunsch nach Individualität kann nur durch eine Vielzahl an Produktvarianten erfüllt werden. Steigt die Zahl der Varianten, sinkt das Bestellvolumen je Produkt. Innerhalb der Produktionsstätte müssen dann relativ viele Stückguttransporte in kleinen Losgrößen durchgeführt werden [Weh-04]. Der reibungslose und schnelle Transport erfordert Materialflusssysteme mit zunehmend komplexen Steuerungsmechanismen.

Die Hersteller und Inbetriebnehmer aktueller Materialflusssysteme sind demnach mit immer komplexeren Abläufen und damit auch Steuerungsaufgaben konfrontiert. Diese müssen in Abstimmung mit Kunden und zahlreichen Zulieferern organisiert werden. Der dabei auftretende Mehraufwand bei Planung, Realisierung und Inbetriebnahme entsteht unter anderem, weil die Steuerung herkömmlicher Materialflusssysteme in einer hierarchischen Ebenenstruktur organisiert ist. Hier sind zwar vertikale Schnittstellen definiert, also z.B. zwischen dem Materialflussrechner und den speicherprogrammierbaren Steuerungen, jedoch besteht in der Horizontalen, also zwischen der Steuerung der einzelnen Anlagenteile, keine klare Trennung. Auf diese Weise entstehen Wechselwirkungen, welche für Programmierer und Inbetriebnehmer nur noch schwer zu überblicken sind. Funktioniert ein Anlagenteil nicht, so kann die Ursache in jeder Ebene der Steuerungsarchitektur angesiedelt sein. Zwar verfolgen einige Hersteller die Strategie, ein Materialflusssystem aus Baukastenelementen zusammenzusetzen. Diese Strukturierung bezieht sich jedoch in der Regel nur auf den mechanischen Aufbau der Systeme [Gün-08]. Ein Modulbaukasten, wie er von vielen Softwareherstellern für die Erstellung der Steuerungssoftware verwandt wird, wird in der Regel unabhängig vom physischen Aufbau des Materialflusssystems entwickelt. Daher verfügt er meist über vollkommen andere Schnittstellen als die Mechanikbaukästen.

Die funktionsorientierte Modularisierung, welche die gleichen Schnittstellen für Mechanik, Energie und Steuerungssoftware für ein definiertes Funktionsmodul fordert [Wil-06], ist ein wichtiger Ansatz, um in Zukunft der beschriebenen Komplexität zu begegnen. Ein Hersteller von Anlagenkomponenten wird damit in die Lage versetzt, nicht nur die Mechanik zu entwickeln und zu testen, sondern auch die dazu gehörende Steuerungslogik. Er wird so zum Anbieter einer mechatronischen Einheit, welche direkt und ohne Programmieraufwand einsetzbar ist. Klare Schnittstellen verhindern dann eine Fortpflanzung von Fehlfunktionen in die restliche Anlage und erleichtern so die Inbetriebnahme. Standardisierte und damit wiederverwendbare Module ermöglichen zudem, einen größeren Anteil der anfallenden Arbeitsaufwände projektunabhängig vorzubereiten und so die projektspezifischen Entwicklungskosten zu senken.

Doch mechatronische Einheiten haben ein noch größeres Potenzial, wenn man sie zusätzlich mit der Fähigkeit versieht, sich selbst zu organisieren und sich mit anderen

Systemteilnehmern abzustimmen. So bekommt jede Einheit quasi eine eigene Intelligenz. Solche intelligenten Einheiten reagieren auf Ereignisse, ohne dass eine übergeordnete Leitinstanz notwendig ist. Dies ermöglicht einen neuen Denkansatz in der Logistik: weg vom vorgeplanten Prozess, hin zu einem System, das sich selbst organisiert und auch auf ungeplante Ereignisse sinnvoll reagiert. Hin zu einem System, das auch auf das Unerwartete vorbereitet ist.

Diese Entwicklung ist notwendig, da sich Ereignisse in einem Materialflusssystem wegen der vielen Einflussfaktoren und Wechselwirkungen nicht mehr mit ausreichender Genauigkeit vorhersagen lassen. So ist laut ten Hompel der bisherige, deterministische Ansatz, "durch die Standardisierung der Vergangenheit, [...] Modelle und Handlungsanweisungen für zukünftige Ereignisse abzuleiten, [...] systemimmanent zum Scheitern verurteilt." [Hom-08].

In der Informatik sind intelligente, verteilte Systeme schon seit längerem Untersuchungsgegenstand. Ein Beispiel dafür ist die Technik der Multiagentensysteme, die auch in der vorliegenden Arbeit Verwendung finden soll. Bei Agenten handelt es sich in diesem Zusammenhang um autonome Softwareprogramme, welche in der Lage sind, eine Aufgabe selbständig oder in Kooperation mit anderen zu lösen. In der Produktionstechnik wird die Agententechnik eingesetzt, um mehr Flexibilität, eine erhöhte Fehlertoleranz im Anlagenbetrieb sowie ein deutlich einfacheres Engineering zu erreichen [Rit-03].

Die Kombination aus Agententechnik und funktionsorientierter Modularisierung ist ein vielversprechender Ansatz zur Modellierung von zukünftigen Materialflusssystemen. Dieser wird in der vorliegenden Arbeit entwickelt und umgesetzt.

1.2 Anforderungen an zukünftige Materialflusssysteme

Zukünftige Materialflusssysteme müssen schneller als bisher geplant, realisiert und in Betrieb genommen werden können. Denn um heutzutage am Markt zu bestehen, ist eine schnelle Reaktionsfähigkeit auf wechselnde Kundenbedürfnisse zu akzeptablen Kosten eine unabdingbare Voraussetzung. Darüber hinaus sollen bereits bestehende Anlagen einfacher auf neue Produktspektren und Durchsätze umgestellt werden können. Dies betrifft zum einen die schnelle Umsetzung neuer Materialflussstrategien und zum anderen die Möglichkeit, das System ohne großen Aufwand

physikalisch zu verändern oder zu erweitern. Diese Anforderungen lassen sich nur erfüllen, wenn gleichzeitig die Komplexität des Systems aus Sicht des Anlagenerbauers sinkt. Ein großes Materialflusssystem mit seinen Abhängigkeiten zu überblicken und zu beherrschen, ist sonst nur noch schwer möglich. Daher sollen im Folgenden die Begriffe Flexibilität, Wandlungsfähigkeit und Wandelbarkeit als zukünftige Anforderungen an Materialflusssysteme näher definiert und Wege zur Komplexitätsreduktion aufgezeigt werden.

1.2.1 Flexibilität

In der Fachliteratur existieren vielfältige Definitionen der Flexibilität, welche sich zwar nicht grundsätzlich widersprechen, jedoch auf die verschiedenen Fachgebiete zugeschnitten sind.

Laut Ost ist Flexibilität bezogen auf die Anpassungsfähigkeit eines Fertigungssystems die „... Fähigkeit, eine vorgegebene Vielfalt von Fertigungsaufgaben bei zufälligen oder systembedingten Änderungen der Eingangsgrößen zu bewältigen, ohne dass das System in seiner Grundkonzeption verändert werden muss.“ [Ost-93].

Heinecker greift diese Definition auf und teilt die Flexibilität zusätzlich in die Basisflexibilität und die erweiterte Flexibilität ein [Hei-06]. Hier entscheidet der Zeitpunkt, an welchem die Flexibilität genutzt wird. Die *Basisflexibilität* ist diejenige Flexibilität, welche bereits beim Neubau einer Anlage bereit gestellt wird und die Anlage daher dazu befähigt, auf wechselnde Anforderungen im laufenden Betrieb zu reagieren. Die *erweiterte Flexibilität* wird erst dadurch nutzbar, dass eine Anlage durch einen Umbau verändert wird.

Bezogen auf Materialflusssysteme lässt sich die Flexibilität mit drei Kriterien beurteilen. Dabei handelt es sich um die Fördergut-, die Layout- und die Durchsatzflexibilität [Han-01, Gün-02, Gün-03]:

- **Fördergutflexibilität**
Das System muss in der Lage sein, unterschiedliche Produkte, unabhängig von deren Abmessung und Gewicht, transportieren zu können.
- **Layoutflexibilität**
Jeder Ort der Fabrik muss bedient werden können bzw. das Materialflusssystem muss sich schnell auf veränderte Materialflusswege einstellen können.

- Durchsatzflexibilität

Durchsatzflexible Materialflusssysteme müssen sich den Leistungsänderungen der Produktion durch geeignete Maßnahmen anpassen können.

Den Entwicklungen flexibler Lösungen ist gemein, dass bereits bei deren Planung die Wirkrichtung, also die Dimension der Flexibilität, festgelegt wird [Sua-96]. Flexibilität ist in den genannten Definitionen also nur die Fähigkeit, auf bereits *vorgeplante* Ereignisse oder Anforderungen zu reagieren. Die Prognostizierbarkeit von Markt- und Nachfrageentwicklungen ist jedoch in den letzten Jahren immer weiter gesunken. Ein Produkt, das heute durch eine stark vernetzte Konsumentengruppe in die Höhe gebubelt wird und daher eine Nachfrageexplosion innerhalb kürzester Zeit erfährt, kann morgen schon wieder „out“ sein. Der Markt orientiert sich nicht an der betrieblichen Flexibilität, die in vorgehaltenen Grenzen existiert - er stellt unerwartete Anforderungen.

1.2.2 Wandlungsfähigkeit und Wandelbarkeit

Soll ein System in der Lage sein, auch auf ungeplante Ereignisse zu reagieren, so reicht Flexibilität alleine nicht mehr aus. Daher ist eine zweite Fähigkeit gefordert: die Reaktionsfähigkeit. Diese ermöglicht es, auch jenseits vorgedachter Dimensionen, Korridore und Szenarien auf plötzliche Veränderungen zu reagieren [Ble-99].

Die Erweiterung der Flexibilität um die Reaktionsfähigkeit wird als Wandlungsfähigkeit oder Wandelbarkeit bezeichnet. Westkämper definiert Wandlungsfähigkeit folgendermaßen:

„Ein System wird als wandlungsfähig bezeichnet, wenn es aus sich selbst heraus über gezielt einsetzbare Prozess- und Strukturvariabilität sowie Verhaltensvariabilität verfügt. Wandlungsfähige Systeme sind in der Lage, neben reaktiven Anpassungen auch antizipative Eingriffe vorzunehmen. Diese Aktivitäten können auf Systemveränderungen wie auch auf Umweltveränderungen hinwirken.“ [Wes-00].

Laut Wilke muss ein wandelbares Materialflusssystem die drei Flexibilitätskriterien Fördergut-, Layout- und Durchsatzflexibilität voll erfüllen [Wil-06]. Diese Kriterien werden auch als operative Flexibilitätskriterien bezeichnet. Ein wirtschaftlicher Betrieb kann unter dieser Voraussetzung jedoch nur schwer realisiert werden, da ein Materialflusssystem dann schon beim Bau umfangreiche Reserven vorhalten muss-

te. Daher fordert Wilke zwei weitere Kriterien, welche Wandelbarkeit auch unter wirtschaftlichen Gesichtspunkten erreichbar machen, nämlich die Erweiterungsfähigkeit und die Integrationsfähigkeit.

- Erweiterungsfähigkeit

Hierunter wird die Möglichkeit verstanden, ein bestehendes Materialflusssystem durch Hinzufügen von zusätzlichen Komponenten zu erweitern.

- Integrationsfähigkeit

Sie beschreibt die Fähigkeit, technische und steuerungstechnische Verbindungen mit anderen Materialflusssystemen eingehen zu können. Um eine hohe Integrationsfähigkeit zu erreichen, müssen die Systemschnittstellen in Bezug auf Mechanik, Energieversorgung und Informationsübertragung standardisiert sein.

Diese beiden Fähigkeiten werden bei Wilke als strategische Wandlungsfähigkeit bezeichnet.

Zusammenfassend lässt sich sagen: Die *operative Flexibilität* befähigt ein Materialflusssystem, sich während des Betriebs dynamisch und ohne manuellen Eingriff an neue Umweltbedingungen, z.B. neue Durchsatzanforderungen oder Ausfälle von Komponenten anzupassen. Dies bedeutet eine höhere Adaptivität und Robustheit als sie von heutigen Systemen geboten wird und ist vor allem für den Anlagenbetreiber von Vorteil.

Die *strategische Wandlungsfähigkeit* verringert in erster Linie den Aufwand für Planung, Anlagenbau bzw. -umbau und Inbetriebnahme und führt dadurch zu verringerten Kosten für Anlagenbauer. Um diese zu realisieren, benötigt man einen Baukasten standardisierter und projektunabhängig realisierbarer funktionaler Einheiten wie Softwaretools und Fördertechnikmodule. Dadurch können Materialflusssysteme nach dem „Plug and Play“ bzw. in diesem Zusammenhang „Plug and Convey“ Prinzip erstellt werden

1.2.3 Komplexitätsreduktion

Der Begriff „Komplexität“ (von complexus, lat.: verwoben, vernetzt) wird von Dörner folgendermaßen definiert: "Die Existenz von vielen voneinander abhängigen Merkmalen in einem Ausschnitt der Realität [...]" [Dör-08]. Die Komplexität eines Reali-

tätsausschnittes ist also umso höher, je mehr Merkmale vorhanden sind und je mehr diese voneinander abhängig sind. Für Malik sind komplexe Systeme durch eine hohe Vielfalt auf der einen Seite und eine hohe Dynamik auf der anderen Seite gekennzeichnet [Mal-06].

Der Mensch kann seine Umwelt nur selektiv wahrnehmen und ist in seiner Fähigkeit begrenzt, Beziehungen zu dieser herzustellen. In der subjektiven Wahrnehmung hängt die Klassifizierung der Komplexität unmittelbar davon ab, inwieweit der Betrachter mit dem System vertraut ist, ihm also die Struktur transparent erscheint [Dör-89]. Damit „setzt Komplexität [...] immer ein Reduktionsverfahren voraus, das ein Muster der Selektion von Beziehungen festlegt und andere Möglichkeiten vorläufig ausschließt“ [Kle-96].

Diese Komplexitätsreduktion kann u.a. durch drei Prinzipien erreicht werden [Fis-06]:

- Abstraktion

Abstraktion bedeutet eine Loslösung von konkreten Eigenschaften. Stattdessen werden gleiche Merkmale und die für das konkrete Problem wichtigen Fragestellungen hervorgehoben. Alles Unwesentliche kann vernachlässigt werden.

- Modularisierung

Ein Gesamtsystem oder eine umfangreiche Problemstellung soll nach der Devise „Teile und Herrsche“ in überschaubare, abgeschlossene und aufeinander abgestimmte Module bzw. Teilaufgaben zerlegt werden. Diese Module sollen möglichst mehrfach verwendbar sein, denn je weniger unterschiedliche Varianten zum Einsatz kommen, desto einfacher bleibt ein Gesamtsystem zu überblicken und zu verstehen.

- Geheimnisprinzip

Das Geheimnisprinzip wird auch als Kapselung oder Black-Box-Prinzip bezeichnet und bedeutet, dass der Nutzer über die in den Modulen existierenden Algorithmen und Datenstrukturen nicht informiert sein muss [Par-72]. Daher werden Schnittstellen definiert, über die auf die Module zugegriffen werden kann. Eine klare Schnittstellendefinition soll bewirken, dass Module nur im beabsichtigten Sinn verwendet werden [Bal-94].

Da ein wichtiger Schlüssel zur Verkürzung von Entwicklungs-, Realisierungs- und Umbauphasen von Materialflusssystemen in der Reduktion der Systemkomplexität liegt, befasst sich die vorliegende Arbeit mit der Konzeption und der Entwicklung von mechatronischen Modulen. Diese sollen in ihrem mechanischen Aufbau variabel, steuerungsseitig aber unabhängig von der konkreten Ausprägung der eingesetzten Fördertechnik einsetzbar sein. Der innere Aufbau der Steuerungslogik kann dabei für den Anwender verborgen bleiben.

1.3 Zielsetzung und Vorgehensweise

Das Ziel dieser Arbeit ist die Entwicklung eines Konzeptes für die Materialflusststeuerung, das für zukünftige Materialflusssysteme einen hohen Grad an Wandelbarkeit ermöglicht. Hersteller von Materialflussanlagen sollen so in die Lage versetzt werden, neue Anlagen schneller als bisher zu planen, zu realisieren und in Betrieb zu nehmen und auch Erweiterungen oder Umbauten einfacher durchführen zu können.

Das hier vorgestellte Steuerungskonzept nimmt das Internet zum Vorbild, da es sich schon seit vielen Jahren als äußerst flexibles und erweiterbares System bewährt hat und vollkommen dezentral organisiert ist. In dieser Analogie entsprechen die Transporteinheiten den im Internet übertragenen Datenpaketen und die Fördertechnik eines Materialflusssystems den Datenleitungen und Routern. In diesem Internet der Dinge spielen Fördertechnikmodule, welche mechatronisch gestaltet sind und über eindeutige Schnittstellen sowohl im Inneren als auch zur Außenwelt verfügen, eine zentrale Rolle. Die klaren Systemgrenzen dieser Module vermindern die Komplexität und verhindern z.B. eine Ausbreitung von Fehlfunktionen, da Abhängigkeiten und Wechselwirkungen reduziert werden.

Diese Module werden so gestaltet, dass sie als Bestandteile eines Baukastens in einem hohen Maße wiederverwendet werden können. Auf diese Weise soll eine Verschiebung eines möglichst großen Anteils der bisher bei Erstellung bzw. Erweiterung und Umbau anfallenden Arbeiten in eine projektunabhängige Vorphase erreicht (siehe Abb. 1.1) und so der Aufwand bei zukünftigen Projekten deutlich reduziert werden.

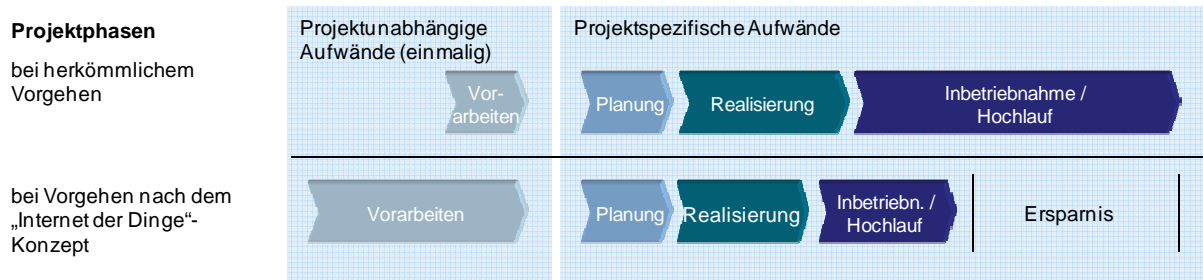


Abb. 1.1: Aufwandsverschiebung

Die vorliegende Arbeit gliedert sich in drei Themenbereiche (siehe Abb. 1.2). In einem ersten Teil werden die Grundlagen heutiger Materialflusssysteme und die derzeit eingesetzten Steuerungsarchitekturen vorgestellt. Eine Betrachtung der heute durchzuführenden Tätigkeiten bei der Erstellung eines Materialflusssystems gibt Aufschluss darüber, an welchen Stellen eine neue Steuerungsarchitektur Optimierungspotentiale freisetzen kann. Um diese neue Architektur umsetzen zu können, soll auf neueste Technologien zurück gegriffen werden. Daher schließt dieser erste Themenblock mit einer Beschreibung der für die angestrebten Entwicklungen relevanten Technologien, nämlich Automatisierungstechnik, RFID, Internet und Agententechnologie.

Der zweite Teil stellt das Konzept des Internet der Dinge vor, beschreibt die Vorgehensweise bei der Modularisierung einer Materialflussanlage und die Bausteine, welche dafür benötigt werden bzw. dabei entstehen. Der wichtigste und gleichzeitig komplexeste Baustein ist ein Fördertechnikmodul. Daher wird auf die Konzeption und die Entwicklung dieser Module ein besonderer Schwerpunkt gelegt. Dies umfasst nicht nur ein Vererbungsmodell, das die Entwicklungszeiten für Module verkürzen soll, sondern auch die Vorgehensweise bei der Implementierung einer dezentralen Materialflusstrategie. Das Internet der Dinge-Konzept wurde in der Versuchsanlage des Lehrstuhls fml der TU München realisiert. Daher wird anhand einer Beschreibung der konkreten Umsetzung in der dort befindlichen Elektrohängebahnanlage die Machbarkeit der neuen Steuerungsarchitektur aufgezeigt.

Der dritte und letzte Themenbereich der vorliegenden Dissertation beschreibt die zu erwartenden Auswirkungen der vorgestellten Lösung auf die Tätigkeiten eines Anlagenherstellers und greift so den ersten Teil der Arbeit nochmals auf.

Eine Zusammenfassung und ein Ausblick auf zukünftige Entwicklungen schließen die Arbeit ab.

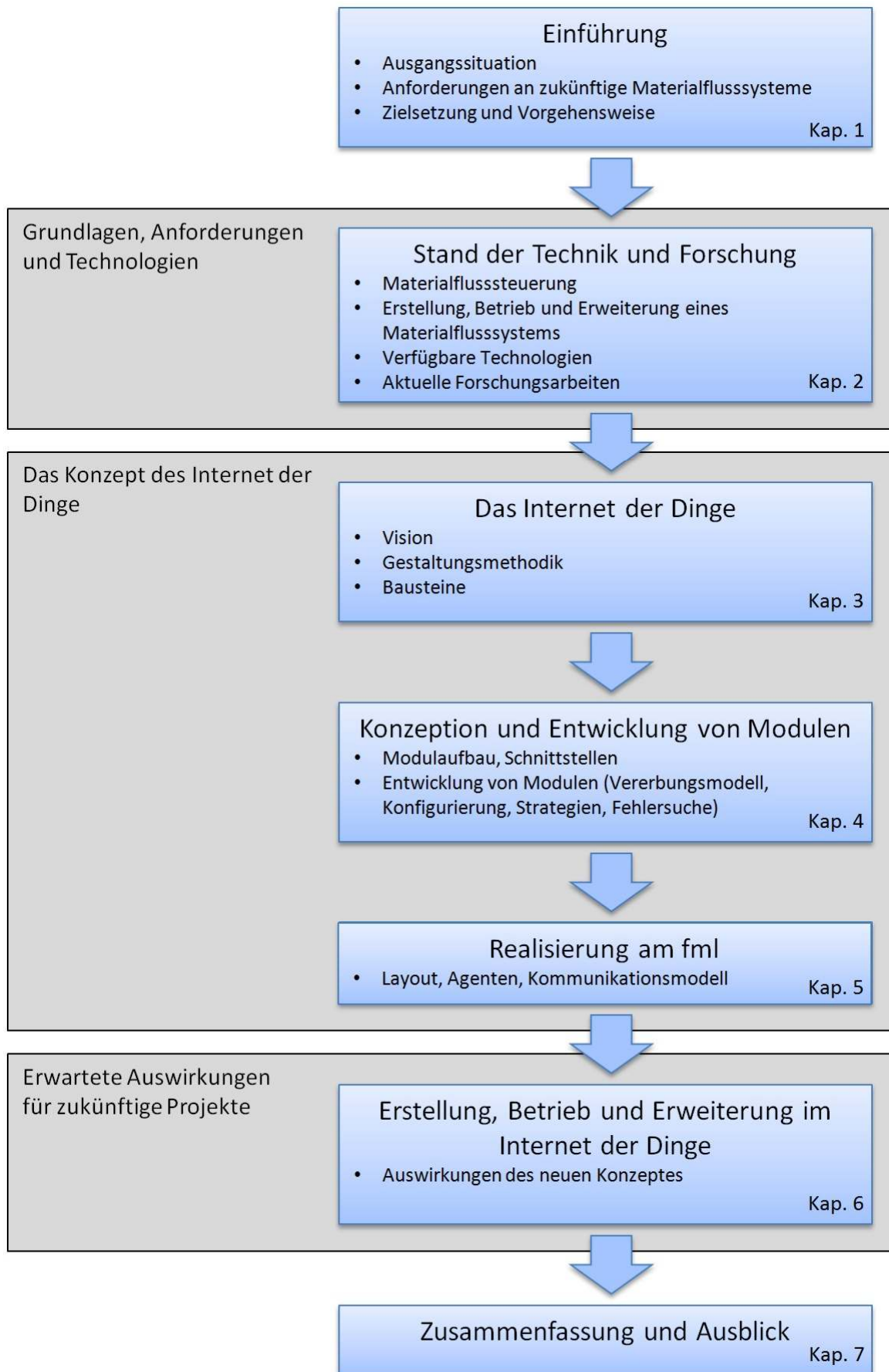


Abb. 1.2: Vorgehensweise

2 Stand der Technik und Forschung

2.1 Materialflusssteuerung

Die Materialflusssteuerung hat laut VDI-Richtlinie 3961 [VDI-3961] das Ziel, auf Basis einer Synchronisation von Informationsfluss und Materialfluss Transporteinheiten zur rechten Zeit, am richtigen Ort, in richtiger Qualität und Menge zur Verfügung zu stellen.

Die heutige Architektur der Materialflusssteuerung ist derzeit noch ein großes Hindernis bei der Gestaltung wandlungsfähiger Materialflusssysteme. Im Folgenden wird der Aufbau heute üblicher Materialflusssteuerungen beschrieben. Anhand dieser Beschreibung werden anschließend Defizite aufgezeigt, welche bei der Gestaltung einer neuen Steuerungsarchitektur ausgeräumt werden sollen.

2.1.1 Aufbau herkömmlicher Materialflusssteuerungen

Hersteller und Betreiber von Materialflusssystemen setzen meist noch auf die herkömmliche Zentralsteuerung, welche hierarchisch organisiert ist und den größten Teil der Steuerungslogik in einem Materialflussrechner konzentriert. Das VDMA-Einheitsblatt 15276 "Datenschnittstellen in Materialflusssystemen" [VDMA-15276] unterteilt den Aufbau der Materialflusssteuerung in die folgenden sechs Ebenen (siehe Abb. 2.1):

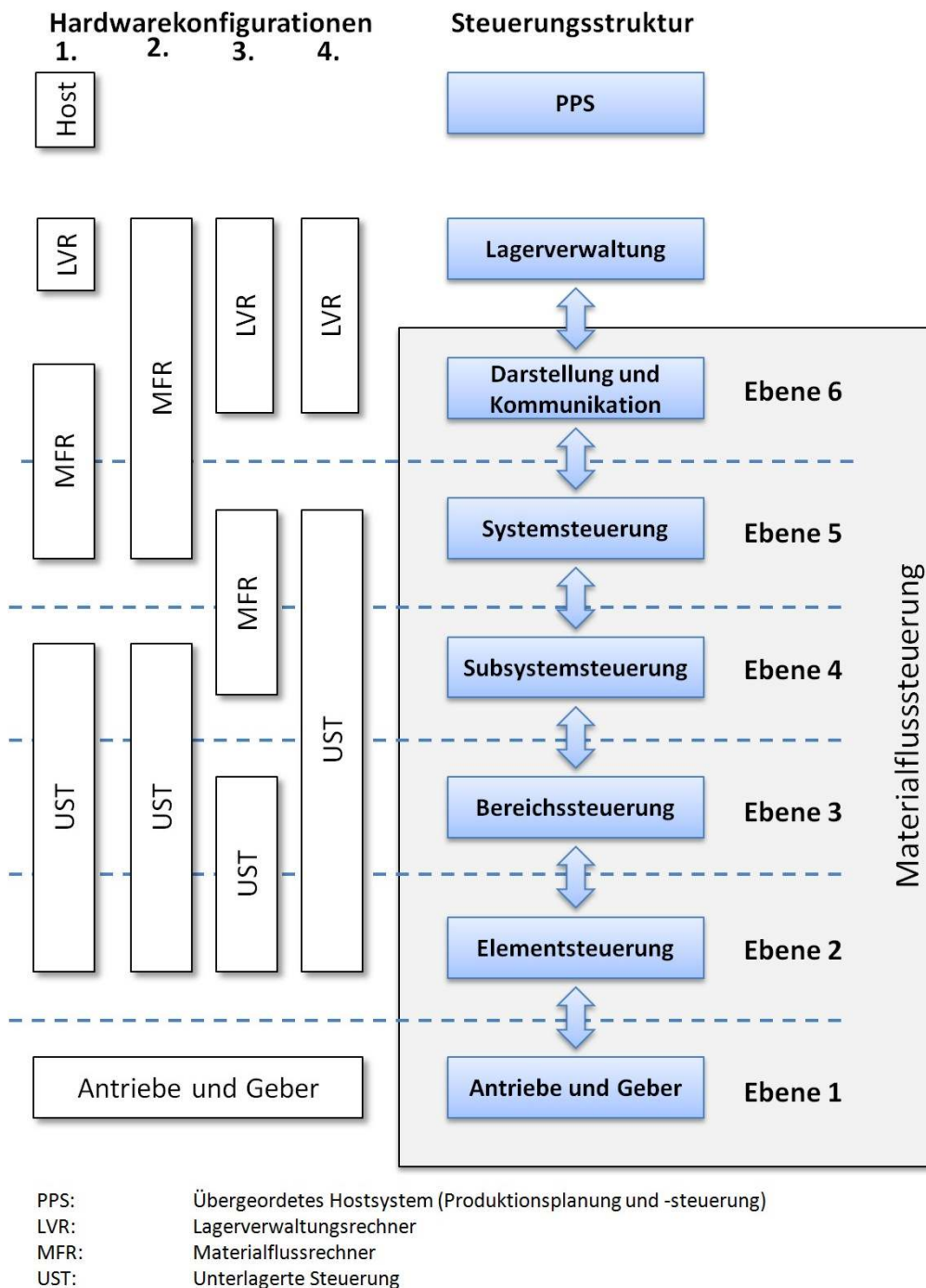


Abb. 2.1: Hardwarekonfiguration und Steuerungsarchitektur der Materialflusssteuerung nach VDMA 15276 [VDMA-15276]

Ebene 6, Darstellung und Kommunikation

Diese Ebene bildet die Schnittstelle zum übergeordneten EDV-System, aus welchem die Transportaufträge an das Materialflusssystem übermittelt werden. Hier findet eine Übersetzung eines Auftrags in die „Sprache“ (also z.B. die Koordinaten) des Materialflusssystems statt. Zudem werden Rückmeldungen wie z.B. Vollzugs- oder Störungsmeldungen erzeugt und weitergegeben.

Eine in dieser Ebene implementierte Visualisierungsumgebung verfügt meist auch über die Möglichkeit, in die Steuerung einzugreifen oder Statusmeldungen und Berichte abzufragen.

Ebene 5, Systemsteuerung

Innerhalb dieser Ebene werden sämtliche Transportoperationen des Materialflusssystems durchgeführt. Dazu werden Informationen zu Topologie, aktuellen Zuständen und Transportauftragsbeständen vorgehalten. Zusätzlich sind hier Strategien zur Abarbeitung der Aufträge hinterlegt. Dies umfasst auch die Routenplanung.

Ebene 4, Subsystemsteuerung

Bereiche des Logistiksystems, welche logisch als Einheit betrachtet werden, wie z.B. ein Wareneingang, verfügen jeweils über eine Subsystemsteuerung. Diese ist so aufgebaut, dass ein solcher Bereich autark betrieben werden kann. Daher müssen die Topologiedaten des Bereichs auch in dieser Ebene redundant zur Systemsteuerungsebene vorgehalten werden.

Ebene 3, Bereichssteuerung

Die Bereichssteuerung betrachtet ein Fördermittel bzw. Gerät als Teil einer Anlage. Für dieses werden die Plätze verwaltet und der Transportablauf gesteuert und überwacht. Die Bedienoberfläche wird speziell für Wartungsarbeiten ausgelegt.

Ebene 2, Elementsteuerung

In dieser Ebene werden ein oder mehrere Antriebe für einen Fördermittelplatz überwacht. Dies umfasst auch die Erzeugung von Sollwerten, die Synchronisation mit anderen Förderelementen und die Ansteuerung von Sicherheitsverriegelungen.

Ebene 1, Antriebe und Geber

Die Antriebe befinden sich auf der untersten Ebene der Materialflusssteuerung. Sie werden mit Energie versorgt und über einfache digitale oder analoge Ein-/Ausgänge angesteuert. Diese Ebene wird sehr stark antriebspezifisch ausgeführt.

Es ist erlaubt, die Ebenen des Steuerungsmodells, je nach Anforderungen, in unterschiedlichen Hardwarekonfigurationen zusammen zu fassen. In Abb. 2.2 ist beispielhaft die Konfiguration in einem Lebensmittellager zu sehen. Diese Steuerung orientiert sich an der Hardwarekonfiguration 3 nach VDMA 15276 (siehe Abb. 2.1).

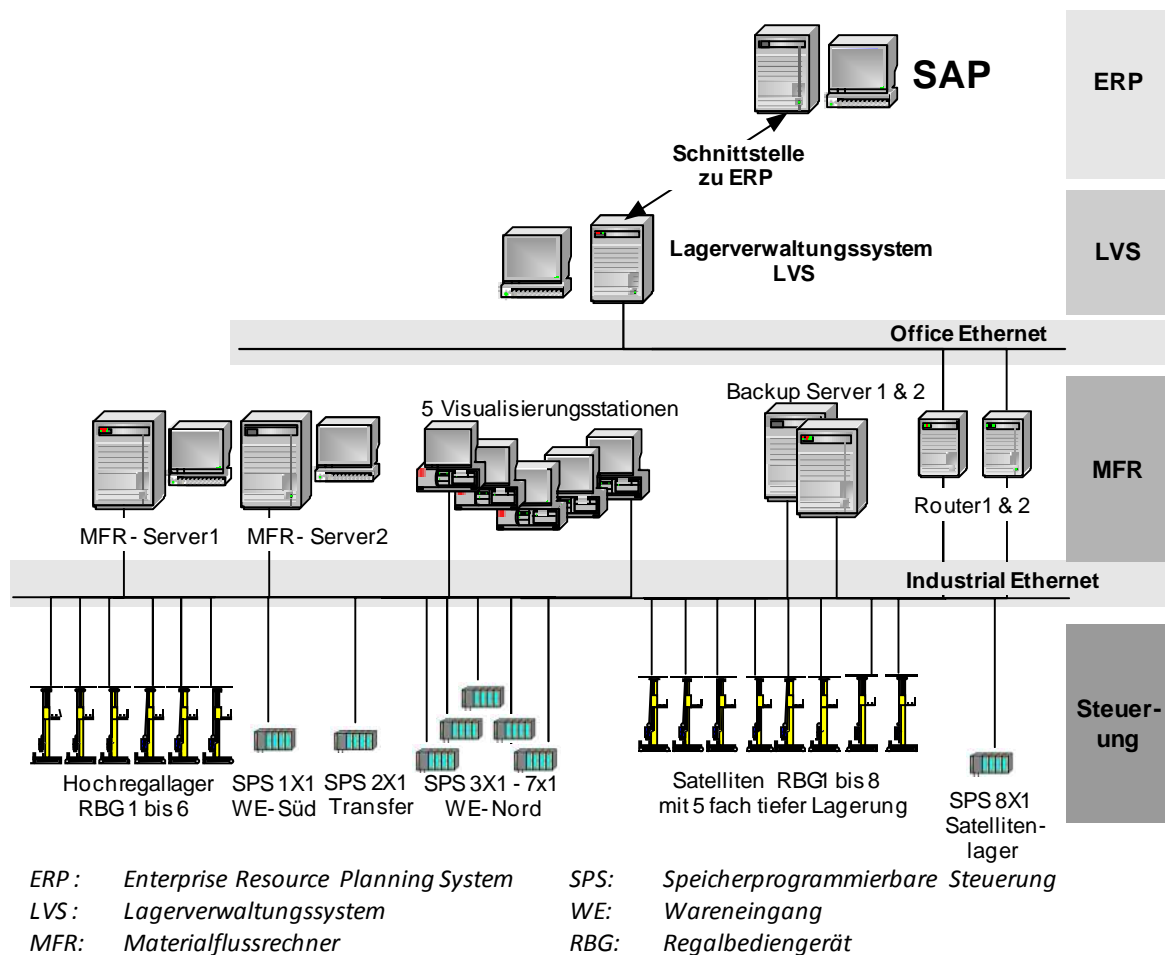


Abb. 2.2: Hardwarekonfiguration und Steuerungsarchitektur in einem Lebensmittellager
(Quelle: Stöcklin Logistik GmbH)

Das bedeutet, dass die Ebenen Element- und Bereichssteuerung in Speicherprogrammierbaren Steuerungen (SPS) implementiert sind, während ein Materialflussrechner (MFR) die Subsystem- und Systemsteuerung realisiert.

Vereinfacht lässt sich das Ebenenmodell der VDMA 15276 auch als Steuerungspyramide darstellen (siehe Abb. 2.3). Diese Darstellung zeigt, dass auf den oberen Ebenen in der Regel dispositive und strategische Aufgaben (wie z.B. die Lagerplatzzuordnung oder die Erzeugung der Transportaufträge) ausgeführt werden. Die unteren Ebenen bilden hingegen die operative Anlagensteuerung bis hin zum Schalten der Aktoren ab.

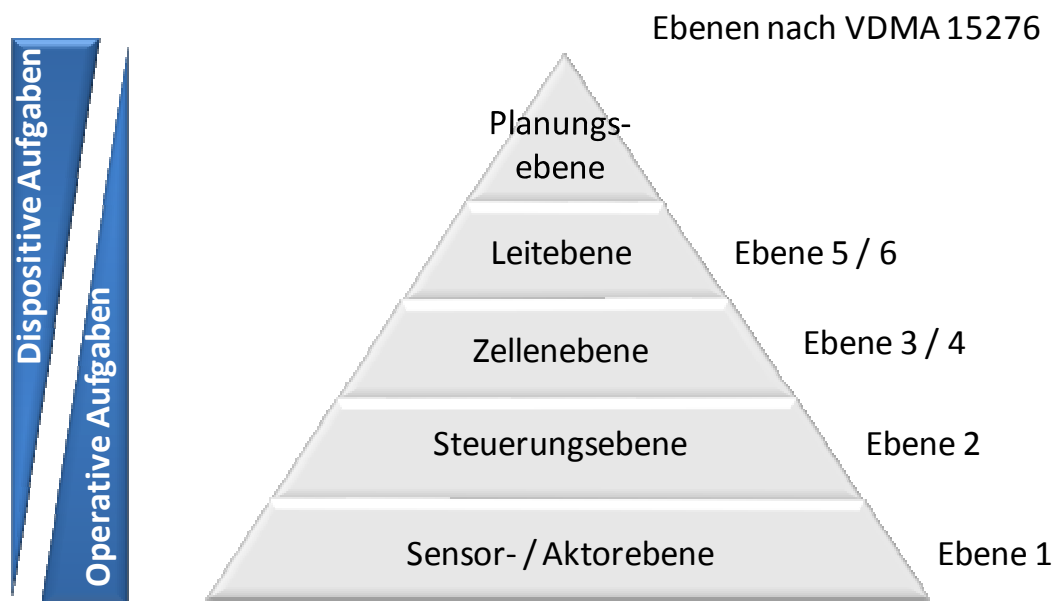


Abb. 2.3: Steuerungspyramide nach Bender[Ben-08a] mit Ebenen nach VDMA 15276

2.1.2 Defizite aktueller Steuerungssysteme

Eine klare und einheitliche Trennung der einzelnen Steuerungsebenen wird laut VDMA 15276 nicht vorgegeben. Dies führt dazu, dass jeder Hersteller von Materialflusssystemen seine eigene Ebenenstruktur mit den damit einhergehenden Schnittstellen einsetzt. So wird es für ein Konkurrenzunternehmen sehr schwer, eigene Lösungen in ein bestehendes System zu implementieren. Dies ist für den Hersteller von Materialflussanlagen zwar einerseits von Vorteil, da er damit seine Kunden an sich bindet. Weil diese Struktur die Einbindung von Zulieferern erschwert, ist diese Kundenbindung allerdings nur eine vermeintliche. Der Kunde wird sich in Zukunft eher für einen Hersteller von Anlagen mit standardisierten Schnittstellen entscheiden,

um sich auch für spätere Aufträge die maximale Entscheidungsfreiheit zu sichern. Die Hersteller haben diese Entwicklung erkannt und beteiligen sich daher an Standardisierungsprojekten wie z.B. der Systemarchitektur für die Intralogistik (SAIL), welche vom Forum Intralogistik des VDMA bis zu einem Richtlinienentwurf vorangetrieben wurde [VDI-5100].

Betrachtet man die Verfügbarkeit eines hierarchisch strukturierten Systems, so ist zu beachten, dass der Ausfall einer Komponente auf einer oberen Ebene weitreichende Folgen bis hin zu einem kompletten Systemausfall haben kann. Die Hersteller setzen daher auf redundante Komponenten, um dieses Risiko in Grenzen zu halten. Damit steigen die Erstellungskosten.

Die Steuerung heutiger Materialflussanlagen hat zudem einen Komplexitätsgrad erreicht, welcher nur noch schwierig zu beherrschen ist (s. Komplexität Kap. 1.2.3). Dies ist auch dadurch bedingt, dass eine Änderung der Steuerungslogik auf einer höheren Ebene immer auch zu Auswirkungen auf allen unterlagerten Ebenen führen kann. Ein Programmierer kann die Folgen einer solchen Änderung also nur noch schwer absehen. Die Fehlersuche bei der Inbetriebnahme einer Anlage wird aus diesem Grund unnötig erschwert. Kommt es zu einem Umbau oder einer Erweiterung einer bestehenden Anlage, so ist die Steuerungsprogrammierung meist ebenfalls auf jeder Ebene zu verändern. Die Schnittstellen auf der Vertikalen, also zwischen den Steuerungsebenen, sind zwar innerhalb einer Anlage meist eindeutig definiert, dies trifft aber auf der Horizontalen, also zwischen den einzelnen Funktionen einer Anlage, nicht zu (siehe Abb. 2.4).

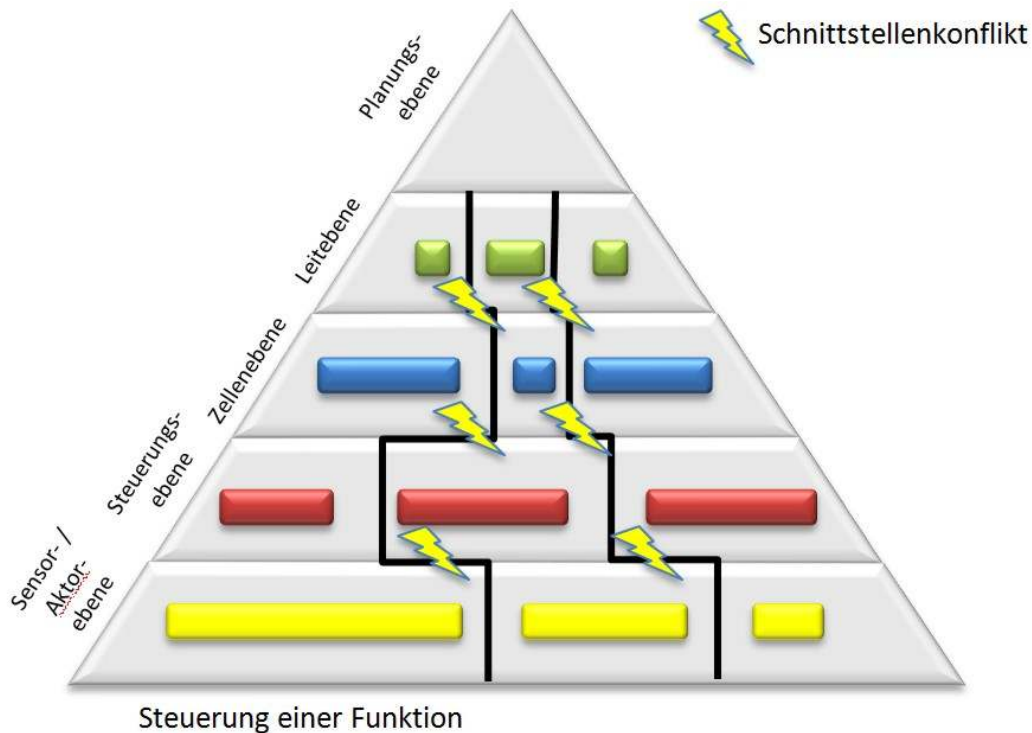


Abb. 2.4: Steuerungspyramide: Beispielhafter Schnitt entlang der Anlagenfunktionen

Beispielsweise wird die Inbetriebnahme eines neuen Regalbediengeräts eine Strategieänderung auf der Leitebene erfordern, aber auch die Implementierung eines neuen SPS-Programms auf der Steuerungsebene. Diese Eingriffe setzen voraus, dass die Anlage zum Zeitpunkt der Änderungen außer Betrieb gesetzt wird. Der entsprechende Programmierer muss über den detaillierten Aufbau aller Steuerungsebenen informiert sein, wobei Fehler in der neuen Implementierung auch zu Problemen beim Betrieb der restlichen Anlage führen können.

Zusammenfassend kann also gesagt werden, dass die aktuell eingesetzte Steuerungsarchitektur noch starke Defizite bzgl. der Zusammenarbeit mit Zulieferern, der Verfügbarkeit, der Beherrschung der Komplexität und der Flexibilität gegenüber Umbau und Erweiterung aufweist und daher neue Steuerungsarchitekturen für Materialflusssysteme zu entwickeln sind.

Hier setzt die bereits erwähnte VDI/VDMA-Richtlinie 5100 an, indem sie objektorientierte Konzepte auf die Materialflusssteuerung überträgt. Dabei ist folgendes Vorgehen maßgeblich:

- primäre Anlagenzerlegung nach Funktionen und nicht nach Ebenen
- Kapselung der gefundenen Funktionen in Komponenten
- Beschreibung der Schnittstellen der Komponenten

- Bereitstellung von Steuerungskomponenten analog zu verfügbaren Mechanikkomponenten

Die dabei entstehenden Komponenten lassen sich wiederum in Elemente und Aggregationsbausteine gliedern. Letztere bestehen entweder aus Elementen oder werden aus anderen Aggregationsbausteinen zusammengesetzt. Diese Verschachtelung kann jedoch wiederum zu relativ komplexen Aggregationsbausteinen führen.

Dieser Richtlinienentwurf ist zwar eine große Weiterentwicklung gegenüber den herkömmlichen Steuerungsarchitekturen, geht aber noch nicht weit genug. Eine weitgehend hierarchielose Steuerungsarchitektur würde die Komplexität reduzieren. Der Einsatz neuester Technologien (siehe Kap. 2.3) kann eine zusätzliche Optimierung bewirken.

2.2 Erstellung, Betrieb und Erweiterung eines Materialflusssystemes

Die Entwicklung einer neuen Steuerungsarchitektur sollte in erster Linie das Ziel haben, die Kosten bei Planung, Realisierung, Inbetriebnahme und Umbau bzw. Erweiterung einer Materialflussanlage zu senken, aber auch Vorteile beim Betrieb der Anlage nach sich ziehen. Dies bedeutet einerseits, den Aufwand zu senken, welcher bei der Durchführung eines Anlagenbaus zu bewältigen ist. Andererseits sind auch die Tätigkeiten soweit wie möglich zu vereinfachen, welche zum Betrieb einer Anlage durchgeführt werden müssen. Im Folgenden wird daher die bisherige Vorgehensweise von der Planung bis zum Betrieb und Umbau einer Materialflussanlage betrachtet und es werden Optimierungspotenziale identifiziert. Letztere bilden zugleich auch die Anforderungen für die Entwicklung einer zukünftigen Materialflussteuerung.

2.2.1 Bisherige Vorgehensweise

Bei der Entstehung einer konventionellen, zentral gesteuerten Materialflussanlage fallen vielfältige Tätigkeiten an, welche unterschiedliche Anforderungen an Technik und Personal von Anlagenbauern stellen (s. Abb. 2.5).

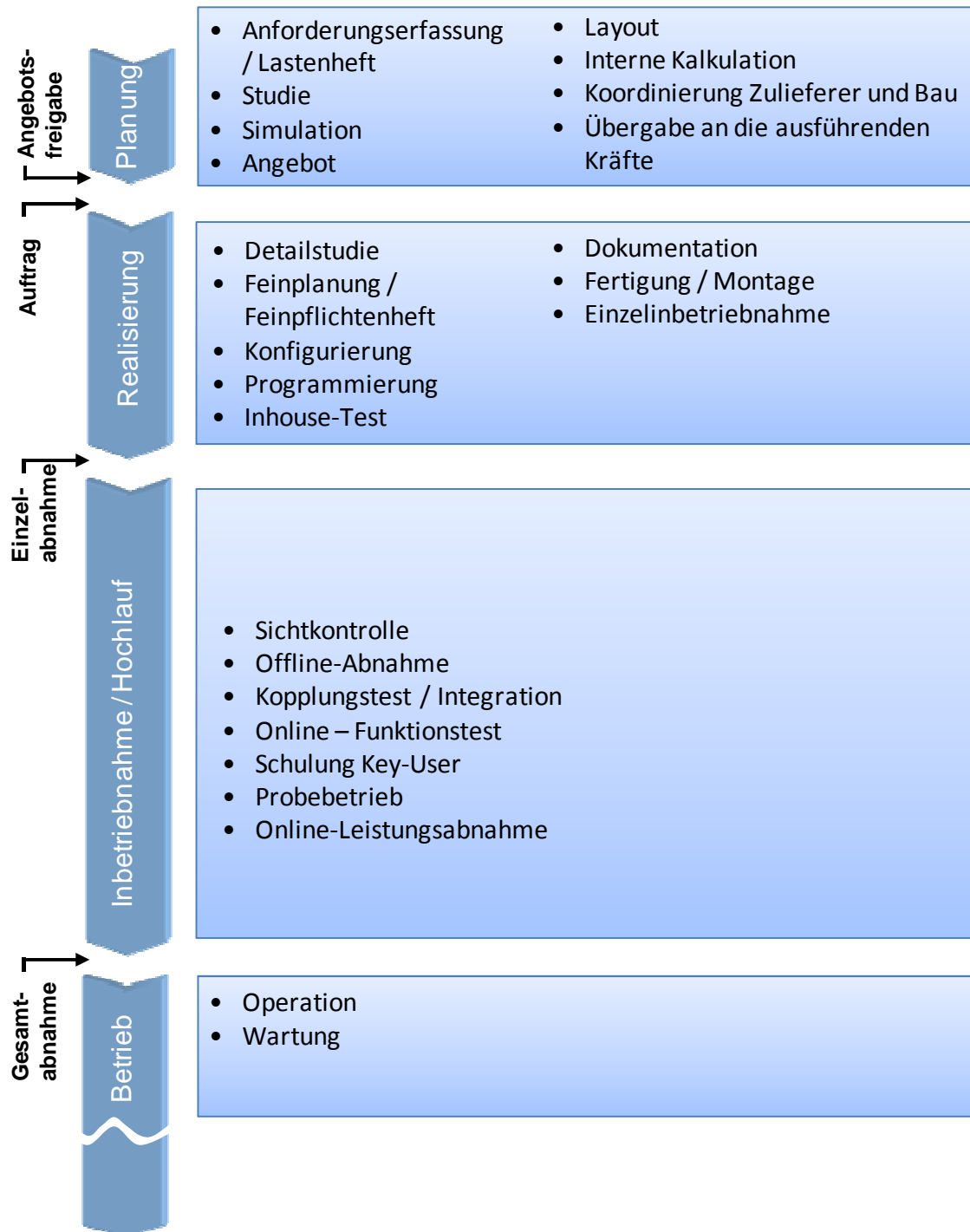


Abb. 2.5: Tätigkeiten von Planung bis Betrieb einer Materialflussanlage und die wichtigsten Meilensteine (Quelle: Forschungsprojekt Internet der Dinge)

2.2.1.1 Planungsphase

Am Beginn jeder Planung steht der Wunsch eines Kunden nach der Neuplanung oder Veränderung seiner Logistikprozesse. Unabhängig davon, ob es sich um eine Neu- oder Umplanung des Materialflusses handelt, ist nun ein Anbieter von Material-

flusssystemen gefragt, um den Kundenwunsch zu akzeptablen Kosten zu bedienen. Abb. 2.6 zeigt die dann auf Anbieterseite ablaufenden Tätigkeiten, welche teils sequentiell und teils parallel ausgeführt werden.

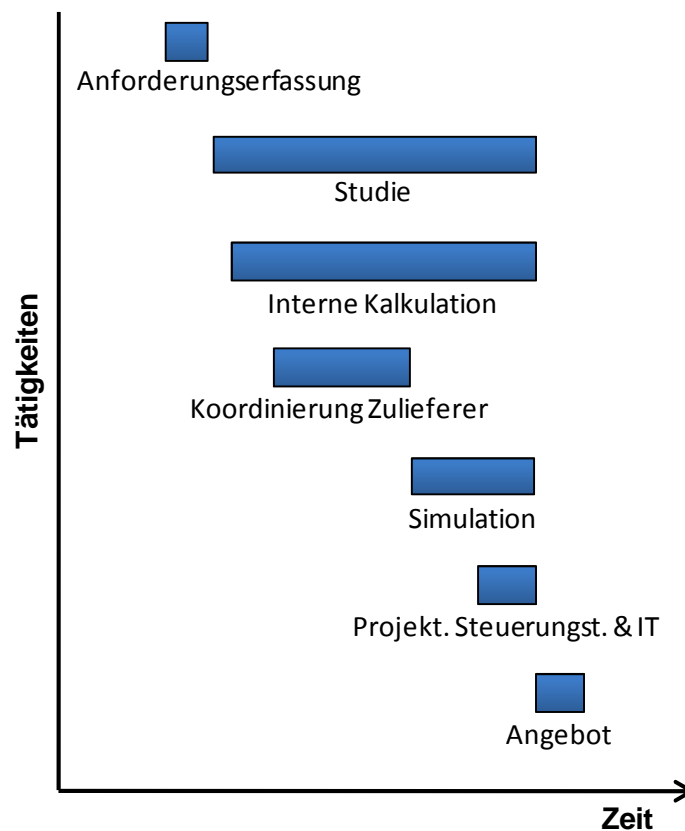


Abb. 2.6: Zeitlicher Ablauf der Tätigkeiten in der Planungsphase (qualitativ)

In der Planungsphase erfasst der Anbieter gemeinsam mit dem Kunden die Anforderungen an das System. Nun werden im Rahmen einer Studie erste Lösungskonzepte erarbeitet und auf ihre Machbarkeit überprüft. Bei größeren Projekten werden zudem ein oder mehrere Simulationsmodelle erstellt. Diese Simulation implementiert weder die genaue Anlagentopologie noch alle Materialflusstategien. Sie soll lediglich die prinzipielle Tauglichkeit eines Lösungskonzeptes aufzeigen. Auf eine Bewertung der verschiedenen Konzepte bezüglich der Erfüllung der vom Kunden gestellten Anforderungen folgt die Bestimmung eines bzw. weniger Lösungskonzepte, welche weiter betrachtet werden. Den Ausschlag für die Auswahl eines konkreten Konzeptes gibt u.a. eine erste Kostenrechnung, welche auch die Grundlage für das spätere Angebot an den Kunden bildet. Für jedes Teilsystem wird eine Make-Or-Buy-Entscheidung getroffen bzw. die Kosten für dessen Erstellung ermittelt. Nun sind die für einen Zukauf in Frage kommenden Zulieferer an den Gesprächen zu beteiligen, damit diese

ebenfalls eine Kostenabschätzung abgeben können. Eine erste Projektierung der benötigten Steuerungstechnik und IT liefert weitere Informationen, welche schließlich in ein Angebot an den Kunden einbezogen werden. Diese erste Aufwandsabschätzung für das zu realisierende System ist umso genauer bzw. einfacher, je höher der Anteil an Lösungen ist, welche aus bereits realisierten Projekten wiederverwendbar sind. Denn hier liegt belastbares Zahlenmaterial vor und daher sinkt das Risiko einer Fehleinschätzung. In der Regel können ca. 60-70% aller geplanten Umfänge beibehalten bzw. wieder verwendet werden. Diese Aussage trifft insbesondere dann zu, wenn bereits umfangreiche Zusatzinformationen wie z.B. die Dokumentation für die einzusetzende Mechanik, Steuerungstechnik und für die Verkabelung vorhanden sind.

2.2.1.2 Realisierungsphase

Nachdem der Kunde den Auftrag erteilt hat, kann an die Realisierung des Projektes gegangen werden. Die entsprechenden Tätigkeiten sind in Abb. 2.7 dargestellt.

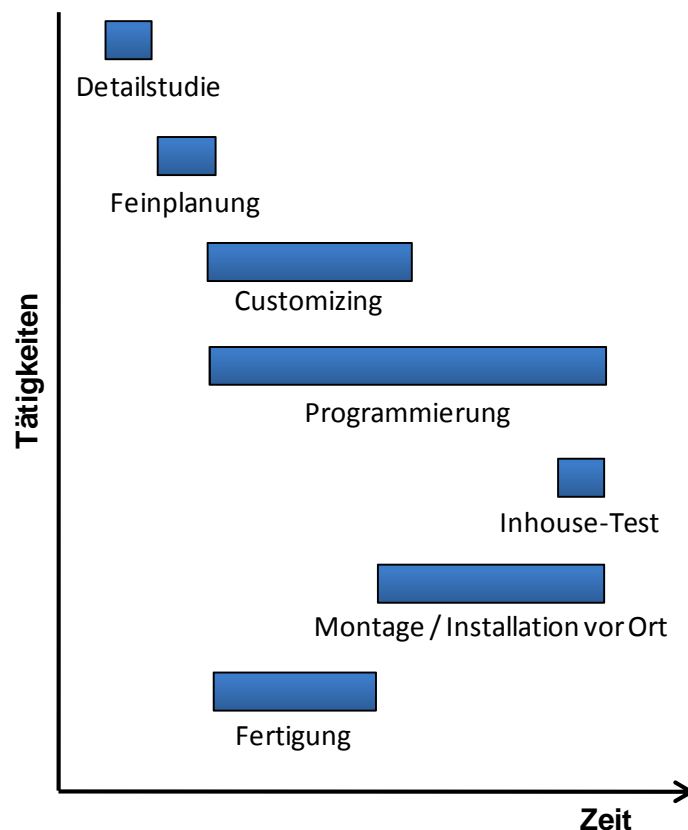


Abb. 2.7: Zeitlicher Ablauf der Tätigkeiten in der Realisierungsphase (qualitativ)

Die Realisierungsphase beginnt mit der Erstellung einer Detailstudie, in welcher ein evtl. vorhandenes Simulationsmodell aus der Planungsphase nun weiter detailliert

wird, um spezielle Materialflussstrategien zu entwickeln und zu überprüfen. Dies umfasst z.B. die Betriebsstrategien von Regalbediengeräten und Querverschiebewägen aber auch die Auftragsdisposition oder die Wegplanung. Bei der Erstellung dieser Simulation wird meist ein Baukasten verwendet, welcher Modelle für häufig eingesetzte Fördertechnikmodule enthält. Durch die Simulation kann der später erreichbare Durchsatz an bestimmten Punkten geprüft werden. Sind zusätzlich komplexe Fragestellungen wie z.B. Prioritätsregelungen oder die Steuerung von Transportreihenfolgen zu klären, dann müssen die entsprechenden Simulationsbausteine geändert bzw. neu implementiert werden. Bei kleinen Projekten, deren Steuerungsaufgabe überschaubar ist, wird in der Regel ganz auf die Erstellung einer Simulation verzichtet oder nur eine sehr grobe Simulation durchgeführt.

Weiterhin wird das Layout im Rahmen eines Feinlayoutplans genauer ausgearbeitet. Dazu wird die konkrete Ausprägung der eingesetzten Fördertechnik bestimmt und die Anzahl der Fördertechnikabschnitte auf den einzelnen Wegstrecken festgelegt. Für diese Aufgabe verfügen die Hersteller meist über einen Mechanikbaukasten, welcher gängige Fördertechnikmodule enthält, dessen Schnittstellen jedoch oftmals anders gezogen sind, als die der Softwarebaukästen für Steuerung und Simulation des Systems. Die so ausgewählten Fördertechnikmodule werden nun auf dem Layoutplan platziert und die Raumeinteilung bzw. Zugänglichkeit abgestimmt. Aus dem Layout lassen sich weitere Festlegungen ableiten. Beispiele hierfür sind

- die Festlegung der Punkte, an welchen die Transporteinheiten identifiziert werden müssen,
- die Bestimmung von Bereichen, welche in einem Notauskreis (siehe Sicherheitssteuerung Kap. 4.2.1.1) zusammen geschaltet werden müssen.

Zusätzlich wird auch die Systemarchitektur aus der Planungsphase weiter detailliert. Das Steuerungskonzept bestimmt, welche Steuerungsaufgaben auf welcher Ebene der klassischen Automatisierungspyramide (s. Abb. 2.3) implementiert werden sollen. Außerdem wird die Systemhardware (Leitrechner, Materialflussrechner, Automatisierungsgeräte) festgelegt, und es werden Datensicherheitskonzepte bzw. Notstrategien für den Ausfall dieser Komponenten entwickelt.

Nun werden die Systemschnittstellen sowohl zwischen den verschiedenen Steuerungsebenen als auch zu übergeordneten Systemen definiert. Die vorgesehenen

Abläufe des Systems werden in Form von Ablauf- und Arbeitsplatzbeschreibungen festgehalten. Gleichzeitig kann nun mit der Programmierung der Steuerung und der Fertigung der Fördertechnikkomponenten begonnen werden.

Ebenso wie bei der Simulationserstellung werden auch bei der Steuerungsprogrammierung in der Regel vorgefertigte Bausteine verwendet. Auch in diesem Fall sind meist umfangreiche kundenspezifische Änderungen und Anpassungen durchzuführen, welche einen erheblichen Arbeitsaufwand bedeuten (sog. Customizing). Die Bausteine aus einem evtl. vorhandenen Simulationsmodell können dabei jedoch nicht direkt wieder verwendet werden, da diese im Allgemeinen eine andere Ablaufumgebung und -logik besitzen als das tatsächlich eingesetzte System. Die Ebenen der Automatisierungspyramide sind dort ebenfalls nicht abgebildet, beispielsweise wird die Aktorik- und Sensorik-Ebene in diesem Rahmen nicht simuliert.

Außerdem benötigt ein Materialflusssystem eine Visualisierungsumgebung, welche dem Betreiber sowohl die Überwachung aktueller Systemzustände als auch den Eingriff in den aktuellen Betrieb der Anlage ermöglichen muss. Diese Visualisierungsumgebung wird ebenfalls im Rahmen der Realisierungsphase erstellt.

Die Steuerungsprogramme auf Materialflussrechner- und SPS-Ebene werden einzeln und im Gesamtzusammenspiel bereits beim Softwareersteller im Rahmen des sogenannten Inhouse-Tests geprüft und um Fehler bereinigt, so dass beim späteren Einspielen der Software auf der Baustelle möglichst wenige Nacharbeiten durchzuführen sind.

Während diese Programme noch erstellt und getestet werden, kann mit dem Aufbau der Anlage beim Kunden begonnen werden. Im Anschluss an die Errichtung der Mechanik (z.B. Förderer) erfolgt der Aufbau der Elektrik (z.B. Motoren, Sensoren, Umrichter) und der Steuerungstechnik einschließlich Verkabelung.

Der Aufbau der physischen Anlagenkomponenten wird dabei gewerkeweise (z.B. mechanischer Aufbau, Verkabelung, Steuerungstechnik) und sequenziell durchgeführt.

2.2.1.3 Inbetriebnahme-/Hochlaufphase

Nun folgt die Inbetriebnahme-/Hochlaufphase, deren Tätigkeiten in Abb. 2.8 zu sehen sind. Diese Phase ist gekennzeichnet von mehreren aufeinander aufbauenden

Test- und Abnahmevorgängen, die letzte Fehler im Aufbau, der Steuerungsprogrammierung und beim späteren Betrieb der Anlage aufzeigen sollen. Treten hier noch Fehler auf, wird umgehend nachgebessert. Dies führt dazu, dass speziell in dieser Phase, je nach Anlagengröße, eine mehr oder weniger umfangreiche Expertenmannschaft vorgehalten werden muss.

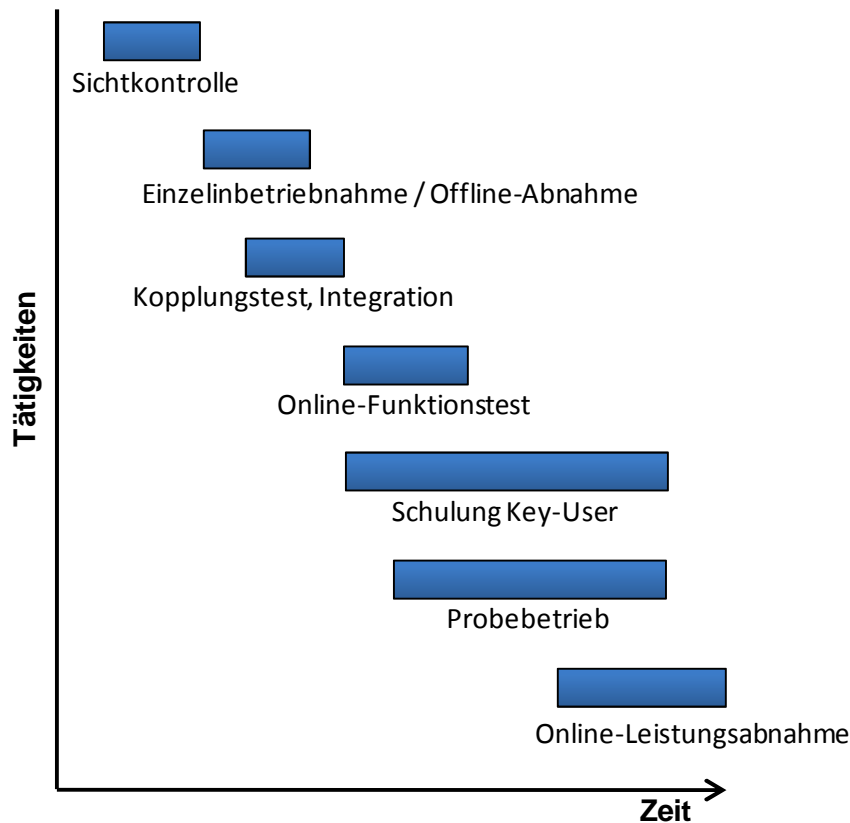


Abb. 2.8: Zeitlicher Ablauf der Tätigkeiten in der Inbetriebnahmephase (qualitativ)

Vor Beginn dieser Phase wird, wie bereits beschrieben, das Materialflusssystem mechanisch aufgebaut, die Verkabelung durchgeführt und die Steuerungssoftware eingespielt. Im Rahmen einer Sichtkontrolle wird die Vollständigkeit dieser Arbeiten bzgl. der handwerklichen Qualität und der statischen Eigenschaften der Fördertechnik geprüft. Nun werden die einzelnen Anlagenteile im Rahmen einer ersten Inbetriebnahme in Bewegung versetzt und auf korrekte Funktionalität überprüft. Die Offline-Abnahme zeigt die Funktionsfähigkeit einzelner Anlagenteile ohne Anbindung des Materialflussrechners. Hier werden Handbetrieb- und Halbautomatikfunktionen getestet und die Einzelleistung dieser Anlagenteile geprüft. Der daran anschließende Kopplungstest ist besonders wichtig, um den fehlerfreien Nachrichtenaustausch zu

überprüfen. Dies betrifft z.B. den Auf- und Abbau von Kommunikationsverbindungen bzw. das Verhalten bei doppelten oder fehlenden Telegrammen.

Der Online-Funktionstest kommt der Betriebsphase der Anlage schon recht nahe, indem hier detaillierte Testszenarien gefahren werden, welche in der Realität auch zu erwarten sind. Treten hier noch Fehler auf, welche sich auf die Koordination und Kooperation von Anlagenteilen – beispielsweise auf die Strategie zur Steuerung mehrerer Verschiebewägen auf derselben Schiene – beziehen, so können Änderungen sehr aufwändig werden. Hier kann es vorkommen, dass die Steuerungsarchitektur in allen Abhängigkeiten und über alle Ebenen der Automatisierungspyramide angepasst werden muss. Dies bedeutet Eingriffe sowohl in die Programmierung des Materialflussrechners als auch in die der Speicherprogrammierbaren Steuerungen.

Bereits während diese letzten Tests durchlaufen werden, kann mit der Schulung der späteren Bediener der Anlage begonnen werden. Dazu werden sogenannte Key-User ausgebildet, welche ihr Wissen als Multiplikatoren an die restlichen Anlagenbediener weiter geben. Zu diesem Zeitpunkt findet meist auch der Gefahrenübergang vom Anlagenhersteller auf den Betreiber statt. Der Betreiber der Anlage übernimmt damit die Verantwortung für eventuelle Schäden, welche durch die Bediener verursacht werden.

Im Rahmen der Schulung erlernen die Bediener den Umgang mit den Mensch-Maschine-Schnittstellen und die Arbeitsabläufe an ihren Arbeitsplätzen. Besonderes Augenmerk gilt hierbei der Bedienung der Visualisierungsumgebung zur Erkennung von Fehlfunktionen und zur geplanten Veränderung von Anlagenstrategien für bestimmte Szenarien. Dies ist beispielsweise dann relevant, wenn zu einer bestimmten Tageszeit eine hohe Auslastung von Streckenabschnitten zu erwarten ist, und daher der Materialfluss manuell auf weitere Strecken verteilt werden muss. Dies ist in einem Versandzentrum für Lebensmittel der Fall, wenn die frische Ware morgens angeliefert, während des Tages kommissioniert und abends versandt wird.

Der letzte Schritt vor der Gesamtabnahme ist die Online-Leistungsabnahme. Im Gegensatz zu allen vorherigen Tests wird die gesamte Anlage dabei an ihre in der Ausschreibung geforderte Leistungsgrenze gebracht. Es wird anhand von vordefinierten Szenarien gezeigt, dass das System die dort definierten Durchsätze und

Antwortzeiten erbringen kann und über einen definierten Zeitraum keine den Betrieb deutlich behindernde Fehler auftreten.

Die hohe Komplexität dieser Aufgabe besteht darin, hunderte Materialflusselemente und ihr Zusammenspiel zu koordinieren. Auch hier gilt: Je später eine Fehlfunktion entdeckt wird, desto aufwändiger ist ihre Behebung. Daher sollte in allen vorhergehenden Phasen so gearbeitet werden, dass hier keine bzw. nur noch geringe Fehler auftreten können. Dies ist meist noch nicht der Fall, was sich daran zeigt, dass Experten bei Befragungen die Online-Leistungsabnahme als die arbeitsaufwändigste Tätigkeit der Inbetriebnahme-/Hochlaufphase beschreiben. Üblicherweise schließt sich an den beschriebenen Leistungstest ein Probetrieb an, der mehrere Wochen dauert und in welchem die Anlage neben dem Durchsatz auch eine festgelegte Verfügbarkeit erreichen muss.

Aufgrund des hohen Integrationsrisikos und der Vielzahl der durchzuführenden Tests ist die Inbetriebnahme-/Hochlaufphase für den Anlagenersteller eine der aufwändigsten Phasen im gesamten Lebenszyklus eines Materialflusssystemes.

2.2.1.4 Betrieb

In der Betriebsphase muss der Anlagenersteller im Rahmen seiner Gewährleistungspflicht noch die von ihm zu verantwortenden Mängel beseitigen. Darüber hinaus wechselt er, wenn vom Kunden gewünscht und bestellt, in die Rolle des Service-Dienstleisters. Ein Service-Vertrag legt dabei den Umfang dieser Dienstleistung fest.

Der Betreiber kann die Betriebszustände der Anlage überwachen, Fehlfunktionen feststellen und kleinere Störungen selbst beheben. Trifft er auf ein größeres Problem, ist er auf die schnelle Hilfe des Anlagenerstellers angewiesen.

In der Regel wird dabei in einem dreistufigen System gearbeitet, das sich nach der Problemkomplexität richtet. Die erste Anlaufstelle für relativ einfache Problemstellungen ist eine Hotline, welche vom Anlagenersteller zur Verfügung gestellt wird. Ist das Problem in dieser Stufe noch nicht zu lösen, so wird in einem zweiten Schritt z.B. anhand von Logdateien eine genauere Analyse des Fehlerzustands vorgenommen und eventuell eine neue Software zur Fehlerbehebung eingespielt. Dies kann über eine Fernwartungsverbindung geschehen oder, falls dies nicht möglich ist, einen Besuch des Wartungspersonals vor Ort erfordern. Die dritte Stufe des Wartungskonzeptes ist komplexen Problemstellungen vorbehalten. Hier müssen sich die System-

entwickler selbst mit der Fehlerbehebung (vor Ort oder über Fernwartung) auseinandersetzen.

Die Kosten für diese Serviceleistungen hängen daher stark von einer klaren Strukturierung der Systemarchitektur ab. Sie muss es erlauben, das Problem schnell und strukturiert einzugrenzen und zu identifizieren. Außerdem spielt die Fernwartbarkeit des Systems eine zentrale Rolle: Je besser ein Fehler bereits via Ferndiagnose eingegrenzt werden kann, desto gezielter und damit kürzer gestaltet sich die Suche des Personals vor Ort.

Eine weitere Aufgabe des Anlagenerstellers ist die Wartung des Systems. Zwar sollten in vorher festgelegten Wartungsintervallen mechanische und steuerungstechnische Komponenten getauscht bzw. überprüft werden, um eine hohe Anlagenverfügbarkeit zu garantieren. In der Realität wird die Anlage jedoch häufig bis zum Ausfall betrieben. Je nach Wartungsarbeit müssen dann Teile oder die komplette Anlage still gelegt werden. Dies wird meist nur in betriebsfreien Zeiten, d.h. nachts oder an Wochenenden und Feiertagen möglich sein und erhöht daher zusätzlich die Kosten für das eingesetzte Wartungspersonal.

2.2.1.5 Erweiterung / Modernisierung

Nach einigen Jahren wird in vielen Fällen eine Erweiterung oder Modernisierung einer bestehenden Anlage notwendig. Dies wird z.B. durch erhöhte Anforderungen an Durchsatz, Lagerkapazität oder ein verändertes Produktspektrum ausgelöst. Prinzipiell wird in einem solchem Fall aus Sicht des Anlagenerstellers der Lebenszyklus der Anlage nochmals durchlaufen. Jedoch betrifft dies speziell den neu zu erstellen- den bzw. zu modernisierenden Teil der Anlage (siehe Abb. 2.9).

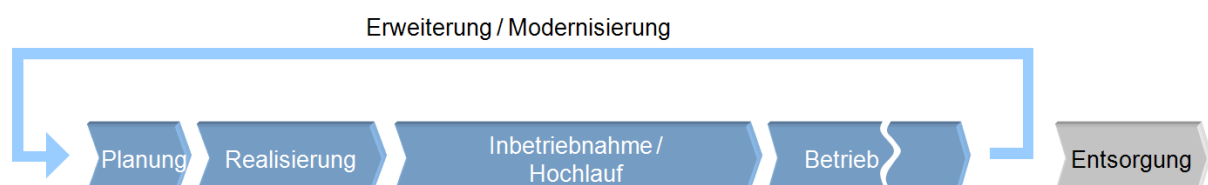


Abb. 2.9: Die Erweiterung / Modernisierung im Lebenszyklus eines Materialflusssystemes

Der maßgebliche Unterschied bei den nun ablaufenden Tätigkeiten ist die Integration der neuen Gewerke in bereits bestehende Anlagenteile. Dies ist sowohl mechanisch als auch softwaretechnisch zu realisieren. Speziell die Anpassung einer bestehenden Materialflussrechnersoftware verursacht dabei einen hohen Aufwand, da in alle

steuerungslogischen Ebenen eingegriffen werden muss. Je nach Tätigkeit ist dazu in vielen Fällen eine komplette Stilllegung der Materialflussanlage notwendig, was wiederum zu erheblichen Kosten führen kann.

Zeit- und kostenintensiv ist in dieser Phase an erster Stelle die Inbetriebnahme, an zweiter Stelle folgt die Umprogrammierung des Materialflussrechners [Gün-08]. Es bietet sich also an, ein Materialflusssystem bereits bei der ersten Erstellung auf diesen Fall vorzubereiten, um diese Aufwände so weit wie möglich zu reduzieren.

2.2.2 Optimierungspotenzial

Das unter Kap. 2.2.1 beschriebene Vorgehen kann an verschiedenen Stellen verbessert werden. Grundsätzlich muss in Zukunft dafür Sorge getragen werden, dass ein möglichst großer Anteil der anfallenden Arbeiten in frühe Projektphasen oder sogar in eine projektunabhängige Vorphase verlegt wird. Dies ist zum einen damit zu begründen, dass ein Fehler umso kostenintensiver zu beheben ist, je später er zutage tritt (s. Abb. 2.10).

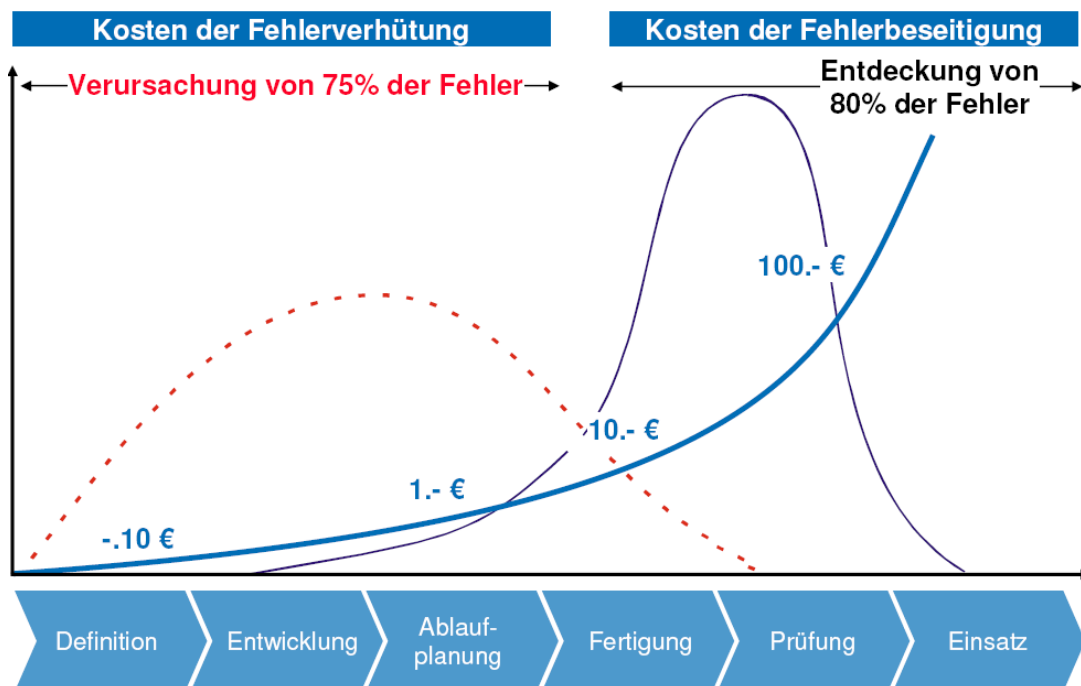


Abb. 2.10: Fehlerverursachung und Fehlerentdeckung bei Entwicklungsprojekten [Sch-08b]

Zum anderen kann durch einen hohen Anteil an vorgefertigten, wiederverwendbaren Arbeitsumfängen das Risiko eines Fehlers begrenzt und der gesamte Arbeitsumfang bei einem konkreten Projekt verringert werden.

Zukünftige Materialflussanlagen sollten also so gestaltet sein, dass ihr Aufbau und ihre Logik für den Anlagenerbauer jederzeit transparent bleiben.

Darüber hinaus gelten für die Tätigkeiten bei der Realisierung eines Materialflusssystemes folgende Anforderungen und Randbedingungen:

Planungsphase

- Eine Simulation sollte so früh bzw. so genau wie möglich durchgeführt werden, ohne dass bei der Erstellung des Simulationsmodells ein nicht mehr vertretbarer Aufwand entsteht.
- Die Kostenkalkulation ist umso belastbarer, je mehr Erfahrungswerte vorliegen und je besser sich die Zahlen aus vergangenen Projekten auch auf zukünftige Realisierungen übertragen lassen.
- Das Risiko einer Fehleinschätzung sinkt, wenn auch die Zulieferer früh in die Gespräche mit einbezogen werden, und eine leichte Integration der zugekauften Lösungen in die restlichen Anlagenteile gewährleistet ist.
- Die reibungslose Zusammenarbeit mit Zulieferern kann durch klar abgegrenzte Zuständigkeitsbereiche und eine gemeinsame Sprache unterstützt werden.
- Eine umfangreiche Dokumentation vergangener Projekte in den verschiedensten Bereichen vom Verkabelungsplan bis zum Steuerungsprogramm verringert erheblich die Gesamtkosten des Projektes und erleichtert die Planung.

Realisierungsphase

- Der Programmieraufwand für Steuerungssoftware und Simulation sollte so gering wie möglich gehalten werden. Das bedeutet, dass ein möglichst großer Anteil bestehender Implementierungen wiederverwendbar sein muss.
- Der ausführliche Test von Hardware, Software und Mechanik sollte bestenfalls bereits beim Hersteller erfolgen und nicht erst auf der Baustelle, wo die Fehlerbehebungskosten höher sind.
- Insgesamt muss die Möglichkeit gegeben sein, Fehler schnell zu lokalisieren bzw. eine Fehlerbehebung durchzuführen, ohne dass dazu die restlichen Anlagenteile außer Betrieb genommen werden müssen.

- Grundsätzlich sollen Programmierarbeiten so weit wie möglich vermieden werden, daher sollten möglichst viele Aufgabenstellungen durch eine Veränderung von Parametern lösbar sein. Daher sollte in Zukunft der Anteil konfigurierbarer Programmteile so hoch wie möglich sein.
- Die Implementierung von Strategien ist eine besonders aufwändige Tätigkeit, bei welcher heutzutage nur auf wenige vorgefertigte Bausteine zurückgegriffen werden kann. Hier ist in Zukunft eine einfachere Umsetzung von Strategien und ein höherer Grad an Konfigurierbarkeit gefordert.
- Bisher wird die Visualisierungsumgebung sehr stark kundenspezifisch gestaltet, was dazu führt, dass meist große Teile neu programmiert werden müssen. Eine generische Umgebung, welche beliebige Anlagentopologien und -zustände darstellen kann, wäre wünschenswert.
- Gerade in der Realisierungsphase sollten so viele Arbeiten wie möglich parallel durchgeführt werden. Dies senkt den Zeitaufwand und damit die Realisierungskosten.

Inbetriebnahme-/Hochlaufphase

- Die größte Herausforderung bei der Inbetriebnahme einer Materialflussanlage ist der Kopplungstest, da hier zum ersten Mal das Zusammenspiel der einzelnen Systemelemente geprüft wird. Klare Schnittstellen und möglichst wenige Wechselwirkungen zwischen den einzelnen Elementen helfen an dieser Stelle, das Risiko zu vermindern.
- Die Inbetriebnahme von Anlagenbereichen sollte in Zukunft parallel durchgeführt werden können.
- Treten in dieser Phase noch Fehler auf, so müssen sie schnell auffindbar und eingrenzbar sein, da gerade hier die Fehlerbehebungskosten sehr hoch sind.
- Eine klare Trennung von Automatik- und Handbetrieb bei der Implementierung der Steuerung erleichtert den Test einzelner Anlagenteile und ermöglicht getrennte Tests von Steuerungslogik und mechanischer Funktion.
- Häufig fordert der Kunde in dieser Phase des Projektes noch Änderungen an Steuerung bzw. Aufbau der Anlage. Diese späten Änderungen müssen leicht durchzuführen sein, ohne dass das bisher umgesetzte Gesamtkonzept verändert werden muss.

- Zur Abnahme werden bestimmte Testszenarien an der Anlage gefahren. Diese Abläufe müssen für den Kunden einfach nachvollziehbar sein. Daher muss sichergestellt sein, dass das Anlagenverhalten z.B. über Logdateien dokumentiert wird.
- Im Kopplungstest wird der fehlerfreie Nachrichtenaustausch geprüft. Hier sind unterstützende Werkzeuge gefordert, welche den Nachrichtenverkehr visualisieren und umfangreiche Filtermöglichkeiten zur gezielten Analyse bereit stellen.
- Die Schulung der späteren Bediener an den Benutzerschnittstellen findet derzeit erst statt, wenn die Anlage bereits den Online-Funktionstest bestanden hat. Durch Simulationsunterstützung könnte diese Schulung zumindest teilweise auf einen früheren Zeitpunkt verlegt werden.

Betriebsphase

- Tritt eine Störung oder eine unerwartete Veränderung der Anlagenauslastung auf, so muss bei herkömmlichen Anlagen eine entsprechende Strategie für genau diesen Fall implementiert worden sein, damit eine adäquate Reaktion erfolgt. Eine zukünftige Materialflussteuerung sollte so gestaltet sein, dass sie im Sinne der Wandelbarkeit (siehe Kap. 1.2.2) auch auf Unvorhergesehenes reagieren kann.
- Eine nachvollziehbare Fehlerberichterstattung erleichtert dem Anlagenbetreiber die Behebung kleinerer Störungen. Darüber hinaus ist eine genaue Dokumentation notwendig, damit der Servicedienstleister schnell entscheiden kann, welches Ausmaß die Störung hat bzw. ob ein Besuch vor Ort notwendig ist. Die Visualisierungsumgebung sollte entsprechende Hilfsmittel zur Verfügung stellen.
- Eine Fernwartungsverbindung kann zum einen den Aufwand für den Servicedienstleister bei Fehlersuche und -behebung erheblich reduzieren. Zum anderen verringert sich auch die Zeit, bis der Fehler behoben ist, da die Anfahrtszeiten des Servicepersonals entfallen. Daher ist bei der Erstellung zukünftiger Materialflussanlagen auf eine gute Fernwartbarkeit zu achten.

- Der Bereich, welcher für die Behebung eines Fehlers oder zur Wartung außer Betrieb genommen werden muss, sollte so klein wie möglich sein, um die Ausfallkosten für den Betreiber niedrig zu halten.

Erweiterung / Modernisierung

- Meist ist es sinnvoll, bereits im Vorhinein die Auswirkungen eines Umbaus genau zu prüfen. An dieser Stelle kann eine Simulation der erweiterten Gesamtanlage Antworten liefern. Zum einen sollte diese Simulation schnell und einfach zu erstellen sein, zum anderen muss das genaue Verhalten der vorhandenen Anlage modelliert sein. Wird dabei auf ein in der Realisierungsphase erstelltes Simulationsmodell zurück gegriffen, so besteht die Gefahr, dass bei der Inbetriebnahme bereits Änderungen am Anlagenverhalten vorgenommen wurden, welche in diesem Modell nicht implementiert wurden. Es wäre also wünschenswert, das Simulationsmodell aus dem aktuellen Steuerungsprogramm der Anlage zu erzeugen.
- Die bestehenden Steuerungsprogramme müssen um die Steuerung der neuen Anlagenteile erweitert werden. Die bestehende Software muss also so gestaltet sein, dass eine problemlose Erweiterung möglich ist und nur ein geringer Anteil geändert werden muss, um die neue Steuerungslogik zu integrieren.
- Häufig ist eine Außerbetriebsetzung der bestehenden Anlage notwendig, da eine Umprogrammierung, z.B. des Materialflussrechners, weitreichende Auswirkungen auf die gesamte Anlage hat. Der Zeitraum und Umfang eines bei einem Umbau oder einer Erweiterung notwendigen Ausfalls ist so gering wie möglich zu halten.

2.3 Verfügbare Technologien

In den vorangegangenen Kapiteln wurden sowohl die Defizite aktueller Steuerungs-lösungen als auch das Optimierungspotenzial bei der Vorgehensweise zur Erstellung eines Materialflusssystems betrachtet. Im Folgenden sollen nun ausgewählte Tech-nologien betrachtet werden, welche zur Schaffung einer neuen Steuerungs-lösung heran gezogen werden können.

2.3.1 Automatisierungstechnik

Laut DIN 19233 ist ein Automat ein selbsttätig arbeitendes künstliches System, des-sen Verhalten entweder schrittweise durch vorgegebene Entscheidungsregeln oder zeitkontinuierlich nach festgelegten Beziehungen bestimmt wird und dessen Aus-gangsgrößen aus seinen Eingangs- und Zustandsgrößen gebildet werden.

Automatisierungstechnik wird zur Steuerung bzw. Regelung von Prozessen einge-setzt. Steuerung ist nach DIN 19226 definiert als Vorgang in einem System, bei dem eine oder mehrere Größen als Eingangsgrößen die Ausgangsgrößen auf Grund der im System eigentümlichen Gesetzmäßigkeiten beeinflussen (siehe Abb. 2.11).



Abb. 2.11: Steuerung

Im Gegensatz zur Steuerung erfolgt bei der Regelung eine Rückkopplung von Aus-gangsgrößen. Es werden bestimmte Zustände aus dem Prozess erfasst und verar-beitet, so dass sie wiederum als Ausgangsgröße auf den Prozess einwirken (s. Abb. 2.12).

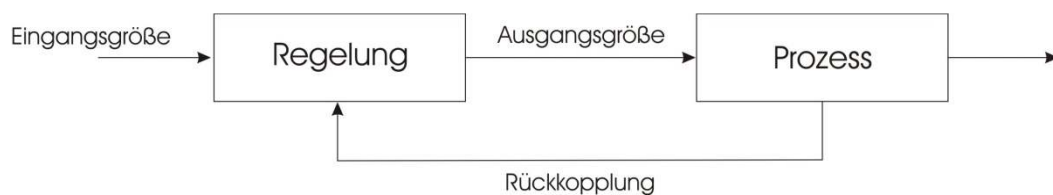


Abb. 2.12: Regelung

Nach DIN 19226 ist die Regelung ein Vorgang, bei dem die zu regelnde Größe fortlaufend erfasst, mit der Führungsgröße verglichen und im Sinne einer Angleichung an diese Führungsgröße beeinflusst wird. Kennzeichen für das Regeln ist ein geschlossener Wirkungsablauf, bei dem sich die Regelgröße fortlaufend selbst beeinflusst.

Der Begriff Steuerung bezeichnet heute spezielle Rechner, welche den Betriebsablauf einzelner Maschinen oder einer Anlage überwachen, steuern und regeln. Dazu sind neben der eigentlichen Recheneinheit auch Ein-/Ausgangsmodule bzw. Schnittstellenmodule zu Feldbussystemen und Ethernet notwendig. Dabei sind die Hauptaufgaben der Steuerungen Schaltfunktionen (ein, aus), Zustandsüberwachung (z.B. Druck, Schaltzustand), Sollwertgenerierung (translatorisch, rotatorisch) sowie die Informationsverarbeitung in bzw. aus Maschinen (z.B. Lagerrückmeldung).

Ursprünglich wurden Steuerungsrechner als Ersatz für Relaissteuerungen entwickelt, bei welchen eine Steuerungsaufgabe durch relativ einfache elektrische Schaltungen ausgeführt werden konnte. Bei den heute in der Automatisierungstechnik eingesetzten Steuerungsrechnern handelt es sich meist um Speicherprogrammierbare Steuerungen (SPS) deren Steuerungslogik in einer Programmiersprache formuliert wird und somit im Gegensatz zur Logik der Verbindungsprogrammierten Steuerungen jederzeit veränderbar ist. Inzwischen haben sich SPS zu leistungsfähigen Rechnern entwickelt, welche zur Steuerung von der Sensor-/Aktorebene bis hin zur Zellenebene eines Materialflusssystems (siehe Kap. 2.1) eingesetzt werden können.

Die Gerätetechnik und die Funktionsweise auf der einen Seite und die entwickelten Programmiersprachen auf der anderen Seite sind auf die Realisierung prozessnaher Aufgabenstellungen abgestimmt.

Typische Kennzeichen einer SPS sind:

- Sicherheit und Zuverlässigkeit

In erster Linie muss sichergestellt sein, dass eine Fehlfunktion einer SPS nicht zur Verletzung oder sogar Tötung von Menschen führt. Daher müssen SPS in dieser Hinsicht besonders sicher arbeiten (siehe Sicherheitssteuerung Kap. 4.2.3).

Da heutzutage ganze Anlagenbereiche von einer SPS gesteuert werden, kann ein Ausfall weitreichende und teure Konsequenzen haben. Aus diesem Grund

haben Speicherprogrammierbare Steuerungen hohen Anforderungen bzgl. Störfestigkeit z.B. gegenüber Strahlung zu genügen und eine hohe Zuverlässigkeit und Verfügbarkeit aufzuweisen.

- **Möglichkeit der Anbindung von Ein-/Ausgängen oder Feldbussen**
Moderne Steuerungssysteme sind meist modular aufgebaut. So kann die Recheneinheit um vielfältige Module erweitert werden. Dies können analoge und/oder digitale Ein-/Ausgangsmodule sein, die eine direkte Verbindung zur Sensor-/Aktorebene der Fördertechnik erlauben. Meist sind darüber hinaus zahlreiche Schnittstellenmodule zu Feldbussystemen wie CAN- oder Profibus oder sogar für drahtlose Kommunikationsverbindungen wie z.B. Bluetooth und ZigBee verfügbar.
- **Robuster Geräteaufbau und keine oder einfache Bedienperipherie**
In der Regel verfügen SPS über wenige Möglichkeiten zur direkten Benutzerinteraktion. Dies ist zum einen damit zu begründen, dass diese Geräte häufig zur Ansteuerung sicherheitskritischer Steuerungsaufgaben eingesetzt werden und der Eingriff eines Unbefugten verhindert werden soll. Zum anderen werden SPS auch in rauen Produktionsumgebungen eingesetzt. Umfangreiche Bedienperipherie würde in diesen Umgebungen zu schnell beschädigt werden.
- **Anwendungsorientierte Programmiersprachen**
Nach IEC 61131-3 sind fünf Programmiersprachen (Anweisungsliste, Strukturierter Text, Kontaktplan, Funktionsbausteine und Ablaufsprache) genormt, die dafür entwickelt wurden, den Programmierer bei der Umsetzung einer steuerungstechnischen Aufgabe zu unterstützen. Hier sind sowohl Sprachen vertreten, welche einfache logische Zusammenhänge übersichtlich darstellen als auch solche, die einen vergleichbaren Funktionsumfang zu dem einer klassischen Hochsprache wie z.B. C bereitstellen.

- **Programmierung mittels PC über Fernzugriff**
Der Zugriff auf Speicherprogrammierbare Steuerungen erfolgt in der Regel über TCP/IP bzw. über ein Feldbussystem. Der Ort, von dem aus die Steuerung programmiert wird, ist daher beinahe beliebig. Zudem stellen die dazu eingesetzten Programmierumgebungen umfangreiche Möglichkeiten zur Diagnose und Überwachung der Steuerung bereit. So kann im Gegensatz zu herkömmlichen Programmierumgebungen jeder Variablenzustand auch zur Laufzeit angezeigt und sogar verändert werden.
- **Zyklische Programmabarbeitung und Echtzeitfähigkeit**
Viele Steuerungsaufgaben, beispielsweise die Ansteuerung eines Antriebs, erfordern eine garantierte Reaktion der Steuerungstechnik innerhalb einer festgelegten Zeitspanne. Diese auch als Echtzeitfähigkeit bekannte Eigenschaft wird durch eine zyklische Abarbeitung des Steuerungsprogramms garantiert. Innerhalb eines Steuerungszyklus werden die Eingänge ausgelesen, das Steuerungsprogramm abgearbeitet und die Ausgänge gesetzt. Die minimale Reaktionszeit einer SPS entspricht der doppelten maximalen Zeit, in der ein Zyklus abgearbeitet wird. Dies lässt sich damit erklären, dass eine Änderung eines Eingangszustandes die kurz nach dem letzten Auslesen der Eingänge auftritt, nicht sofort verarbeitet werden kann. Das Steuerungsprogramm muss erst noch einmal mit den alten Werten abgearbeitet werden, bevor die Änderung wirksam wird.

Aktuelle Entwicklungen

Bender [Ben-08] führt vier Trends auf, welche die weitere Entwicklung der Automatisierungstechnik beeinflussen werden:

- **Miniaturisierung**
Die Komponenten werden bei vergleichbarer Leistungsfähigkeit immer kleiner und ermöglichen so dezentrale Automatisierung (s.u.).
- **Dezentrale Automatisierung**
Verteilung der Intelligenz und Verlagerung der Steuerungslogik in Prozessnähe. Anstatt immer größere Rechenleistung zentral zur Verfügung zu stellen,

sollen in Zukunft kleine Recheneinheiten direkt am Ort des Geschehens eingesetzt werden.

- Offene Systeme

Ein System kann als offen bezeichnet werden, wenn drei Eigenschaften gegeben sind. Erstens sollen alle Schnittstellen an den Systemgrenzen bekannt sein (Interkonnektivität). Zweitens müssen die auszutauschenden Daten in ihrer Bedeutung festgelegt sein (Interoperabilität). Und drittens dürfen keine Unterschiede zwischen den Fähigkeiten der Komponenten bestehen (Austauschbarkeit).

- Verschmelzung von IPC, SPS und NC

Automatisierungstechnische Komponenten werden immer vielseitiger einsetzbar. Die Trennung zwischen Industrie-PC (IPC), Speicherprogrammierbarer Steuerung und Numerischer Steuerung (NC) wird sich immer mehr auflösen. Schon heute bieten viele Hersteller beispielsweise eine Kombination aus IPC mit herkömmlichem Betriebssystem (Windows, Linux) und einer SPS an, welche mit IEC 61131-3 Sprachen programmiert werden kann und echtzeitfähig ist.

2.3.2 Radio Frequenz Identifikation

Die sogenannte Radio Frequenz Identifikation oder RFID-Technologie ist eine Identtechnologie, bei welcher die Daten berührungslos mittels elektromagnetischer Wellen oder induktiver Kopplung übertragen werden. Sie kann wie ein Barcode dazu verwendet werden, eine Transporteinheit eindeutig zu identifizieren. Dazu werden mindestens zwei Komponenten benötigt. Ein Schreib-/Lesegerät mit Antenne, um die Daten zu erfassen bzw. zu senden und ein elektronischer Datenträger, welcher seine Daten per Funkverbindung zur Verfügung stellen kann. Dieser Datenträger wird in Anlehnung an die lateinischen Begriffe transmittere (übertragen, hinüber schicken) und respondere (antworten) als Transponder bezeichnet.

Transponder gibt es als aktive und als passive Ausführung. Passive Transponder beziehen die Energie zum Versenden ihrer Nachrichten aus der Energie des elektromagnetischen Felds, das vom Schreib-/Lesegerät aufgebaut wird. Bei aktiven Transpondern muss ein Energiespeicher z.B. eine Batterie, welche im Transponder verbaut ist, die Energie bereitstellen. Die Lebensdauer der Transponder hängt zum

einen von den Umgebungsbedingungen am Einsatzort und zum anderen von der eingesetzten Energieversorgung ab. Aktive Transponder sind zudem darauf angewiesen, dass der verbaute Energiespeicher regelmäßig getauscht bzw. aufgeladen wird.

RFID-Systeme werden auf den unterschiedlichsten Frequenzen von Langwelle (135kHz) bis Mikrowelle (5,6GHz) betrieben. Die Reichweite dieser Systeme kann je nach Frequenzspektrum, Sendeleistung und Umgebungseinflüssen nur bei wenigen Millimetern liegen, jedoch können unter günstigen Bedingungen Reichweiten von 15m und darüber erreicht werden.

Als Close-Coupling-Systeme werden Systeme mit Reichweiten von bis zu einem Zentimeter bezeichnet. Hierbei muss der Transponder direkt an die Antenne des Lesegeräts gehalten werden. Gerade bei erhöhten Sicherheitsanforderungen sind Close-Coupling-Systeme das Mittel der Wahl, da ein unerlaubter Zugriff auf den Transponder aus größerer Entfernung nicht möglich ist. Dies ist zum Beispiel bei der Eingangskontrolle am Skilift oder dem Bezahlssystem in einer Mensa der Fall.

Spricht man von Remote-Coupling-Systemen, so sind Reichweiten bis zu einem Meter gemeint. Hier kommen einerseits Sendefrequenzen unterhalb von 135kHz und andererseits 13,56MHz zum Einsatz. Gerade diese Systeme haben inzwischen ihren Weg in zahlreiche industrielle Anwendungen gefunden und sind häufig anzutreffen. Eine Vielzahl technischer Lösungen ist daher bereits am Markt verfügbar.

Die sogenannten Long-Range-Systeme mit Reichweiten über 1m werden in Europa auf der UHF-Frequenz 868MHz bzw. der Mikrowellenfrequenz 2,6GHz betrieben, sind jedoch im Vergleich zu Close-Coupling-Systemen sehr viel seltener anzutreffen.

Je nach Frequenz und Bauart verfügen einige Transponder über die sogenannte Pulkfähigkeit. Diese Fähigkeit erlaubt, mehrere Transponder, welche sich im gleichen Schreib-/Lesefeld befinden, gleichzeitig auszulesen. So ist es in vielen Fällen möglich, den Inhalt einer kompletten Palette am Wareneingang zu bestimmen, wenn die einzelnen Packstücke mit Transpondern ausgestattet sind.

An diesem Beispiel ist zu erkennen, dass RFID viel mehr Möglichkeiten bietet, als nur den Barcode zu ersetzen. Finkenzeller [Fin-06] benennt folgende Vorteile des Einsatzes von RFID in der Produktion:

- **Qualitätskontrolle**

Die Qualitätsdaten eines jeden Fertigungsschrittes können mittels RFID direkt am Produkt gespeichert werden. Dadurch ist gewährleistet, dass die Information eindeutig dem Produkt zugeordnet wird, das sie betrifft.
- **Systemsicherheit**

Die Speicherung von Daten direkt am Objekt und nicht, wie bisher üblich, in einer zentralen Datenbank, erhöht die Datensicherheit erheblich. Auch bei einem Ausfall des Zentralrechners gehen die Daten nicht verloren, sondern sind noch am Objekt vorhanden.
- **Datensicherheit**

Die Daten auf einem Transponder sind durch verschiedene Prüfsummenverfahren geschützt. Werden diese Daten fehlerhaft ausgelesen, so kann dies in jedem Fall festgestellt und darauf reagiert werden.
- **Flexibilität**

Transponder ermöglichen das Abspeichern fertigungsrelevanter Informationen direkt an der Transporteinheit. So können jeder Transporteinheit individuelle Daten mitgegeben werden, wodurch eine Fertigung in Losgröße eins einfacher realisierbar ist. Eine aufwändige Kommunikation zu einem Zentralrechner entfällt.
- **Raue Umgebungsbedingungen**

RFID-Systeme sind unempfindlich gegen Staub, Feuchtigkeit, Öle, Gase, hohe Temperaturen und ähnliche Beeinträchtigungen. Glas- oder Kunststofftransponder sind in der Regel vollkommen staub- und wasserdicht.

Speziell mit einer Anwendung im intralogistischen Bereich befasst sich Günthner [Gün-05a] und bemerkt dazu:

„Transponder fassen auch größere Datenmengen [als ein Barcode] und können immer wieder neu beschrieben werden. Damit ermöglichen sie zum ersten Mal eine effiziente dezentrale Datenhaltung direkt am Transportgut. Die RFID-Technologie eignet sich deshalb ideal zur effizienten Verfolgung, Steuerung und Begleitung logistischer Objekte und Prozesse in Wertschöpfungsnetzen.“

Die genannten Gründe sprechen dafür, dass die RFID-Technologie eine wichtige Grundlage für zukünftige Steuerungsarchitekturen von Materialflusssystemen sein wird.

2.3.3 Internet und TCP/IP

Im Jahr 1958 gründete das amerikanische Verteidigungsministerium eine Forschungsbehörde mit dem Namen Advanced Research Projects Agency (ARPA). ARPA gab den Anstoß zur Entwicklung des Computernetzwerkes, das heute als Internet bekannt ist. Die folgende Darstellung der Entwicklung des Internet stützt sich auf Muschs Arbeit über „Die Geschichte des Netzes“ [Mus-97]. Die ARPA hatte den Auftrag, innovative militärisch nutzbare Projekte finanziell zu unterstützen. Sie wurde dadurch zu einem der wichtigsten Förderer der Informationstechnik in den Vereinigten Staaten. Zu dieser Zeit waren leistungsfähige Rechner, welche – für damalige Verhältnisse – große Datenmengen verarbeiteten und erzeugten, schon verbreitet. Der Datenaustausch allerdings war beschwerlich. Die Rechner konnten nicht direkt miteinander kommunizieren und die Daten mussten daher in Stapeln von Lochkarten übertragen werden.

Neue Betriebssysteme erlaubten es mehreren Nutzern gleichzeitig auf einen leistungsfähigen Prozessor zuzugreifen. Hier wurden mehrere einfache Terminalrechner in einer Sterntopologie an einen gemeinsamen Hostrechner angeschlossen. Dadurch konnte auf die aufwändige Vorbereitung von Lochkartenstapeln für eine spätere Verarbeitung verzichtet werden. In einem nächsten Schritt sollte die Nutzung von Diensten räumlich verteilter Rechner ermöglicht werden. Hierzu war eine Änderung der Netzwerktopologie notwendig, da die Störung einer Datenverbindung zum Ausfall des angeschlossenen Terminals führte, und der Ausfall des Zentralrechners das gesamte Netzwerk außer Funktion setzte. Bei zu vielen Teilnehmern stieß der Zentralrechner bald an seine Grenzen. Deshalb konnte ein vorhandenes System nicht beliebig erweitert werden.

Die US Air Force befürchtete zu dieser Zeit, dass ein sowjetischer Nuklearangriff den Ausfall ihrer Kommandostruktur zur Folge haben könnte. Daher sollte anstelle einer Sterntopologie eine Netzwerktopologie eingesetzt werden, welche zu jedem Kommunikationspartner redundante Verbindungen möglich macht. Zusätzlich sollte das Datenaufkommen gleichmäßig auf das Netzwerk verteilt werden, um Überlastungen einzelner Leitungen zu verhindern. Dazu wurden die Nachrichten in kleine Pakete zerteilt, welche auf verschiedenen Wegen übertragen wurden.

Das erste Netzwerk, das diese Ideen konsequent umsetzte und zusätzlich noch die Integration von Hardware unterschiedlicher Hersteller erlaubte, war das 1968 entwickelte ARPANET. Die beiden Hauptanwendungsgebiete waren die Fernsteuerung von Rechnern (Telnet) und der Austausch von Dateien mit anderen Rechnern (file transfer protocol - ftp). Auch die Hersteller von Rechnerhardware wie IBM und Xerox hatten inzwischen eigene Netzwerke entwickelt, welche jedoch nicht in der Lage waren, mit dem ARPANET zu kommunizieren. Es lag also nahe, ein neues Netzwerk zu schaffen, das auch den Datenaustausch zwischen heterogenen Netzen erlauben sollte. Diese Entwicklung wurde als INTERNET bezeichnet und benötigte einen neuen Standard zur Kommunikation, der unter dem Namen „Transmission Control Protocol“ (TCP) bekannt wurde. Dieses überwacht in einer überarbeiteten Form noch heute die Übermittlung der Datenpakete und garantiert eine gleichmäßige Netzwerkauslastung. Ursprünglich hatte TCP auch die Aufgabe, die angeschlossenen Rechner zu adressieren. In den 80er Jahren wurde diese Aufgabe in ein neues Protokoll ausgelagert – das „Internet Protocol“ (IP). Im IP wird jedem Rechner eine hierarchisch aufgebaute Adresse zugewiesen. In der vierten Version des IP gibt das erste Byte das Netzwerk an, in welchem sich der Rechner befindet und die letzten drei Teile die Adresse innerhalb des Netzwerks. Die Länge der Adresse wird nun auf 16Bytes umgestellt (IPv6), um mehr Teilnehmer adressieren zu können. Der hierarchische Aufbau der IP vereinfacht das Routing, also das Auffinden eines Weges für Nachrichtenpakete enorm, da nicht jeder Zwischenstation auf dem Weg der endgültige Zielrechner explizit bekannt sein muss.

Das Internet hat in den letzten Jahren ein exponentielles Wachstum erfahren, wobei verschiedenste Netzwerktechnologien und Rechenplattformen zu einem weltumspannenden Netz zusammengewachsen sind. Millionen von Teilnehmern vernetzen sich, ohne dass dafür eine zentrale Leitinstanz notwendig ist. Das ursprüngliche Ziel, ein hochgradig robustes und beinahe beliebig erweiterbares System zu schaffen, wurde erreicht.

Das Internet ist inzwischen aus der modernen Welt nicht mehr wegzudenken. In Deutschland nutzten im Jahr 2007 bereits beinahe 70% der Bevölkerung im Alter über zehn Jahren das Internet [Sta-07]. Durch diese große Verbreitung konnten die Hardwarekomponenten in größeren Stückzahlen produziert werden, was zu einem

Preisverfall dieser Komponenten führte. Zeitgleich wurden neue Domänen für die Nutzung des TCP/IP erschlossen und spezielle Hardwarelösungen entwickelt.

Die wichtigste für die technische Umsetzung des Internet verwendete kabelgebundene Übertragungstechnologie ist das Ethernet. Ethernet ist in der IEEE-Norm 802.3 standardisiert und legt beispielsweise Stecker- und Kabeltypen zur Datenübertragung fest. Ethernet beginnt die klassischen Feldbussysteme wie CAN oder PROFIBUS zur Vernetzung steuerungstechnischer Komponenten abzulösen. Die Hersteller steuerungstechnischer Komponenten wie SPS, Frequenzumrichter oder Sensorik haben inzwischen das Potenzial des TCP/IP erkannt und bieten entsprechende Schnittstellen an.

Das sogenannte Industrial Ethernet soll den Ethernet-Standard auch für die Anwendung im industriellen Umfeld nutzbar machen. Dazu werden spezielle Komponenten entwickelt, welche für raue Umgebungsbedingungen im industriellen Umfeld geeignet sind, Schutz vor Staub und Spritzwasser bieten, besonderen EMV-Anforderungen genügen und meist auf eine Hutschiene montierbar sind.

Klassische Feldbusse können aber erst dann abgelöst werden, wenn die neuen Übertragungsprotokolle eine wichtige Eigenschaft aufweisen, welche im Automatisierungsumfeld unerlässlich ist – die Echtzeitfähigkeit (siehe Kap. 2.3.1). In diesem Zusammenhang bedeutet Echtzeitfähigkeit, dass eine Nachricht innerhalb einer definierten Zeitspanne ihren Empfänger garantiert erreichen muss.

Die Anbieter von Industrial Ethernet Technologien haben bereits verschiedene Lösungen entwickelt, welche echtzeitfähig sind und auf den Ethernet Standard aufbauen. Um Echtzeitfähigkeit zu erreichen werden folgende Mittel eingesetzt:

- **Netzwerksegmentierung**
Durch das Segmentieren des Ethernet-Netzwerkes lassen sich Kollisionen von Nachrichten einschränken oder sogar komplett vermeiden. Eine Möglichkeit der Segmentierung ist, eine überschaubare Anzahl von Kommunikationsteilnehmern in einem getrennten Netzwerk miteinander zu verbinden. Die Anbindung an das übrige Netzwerk wird über einen Router geregelt.
- **Nachrichtenpriorisierung**
Durch den Einsatz von Nachrichtenspeichern kann dafür gesorgt werden, dass höher priorisierte Nachrichten zuerst weiter geleitet werden.

- **Kommunikationsorganisation und Uhrzeitsynchronisation**
Ein Beispiel für Kommunikationsorganisation ist die Festlegung von Zeitfenstern, in welchen die Kommunikationsteilnehmer berechtigt sind, Nachrichten abzusetzen. Damit dieses Verfahren eingesetzt werden kann, ist es sinnvoll, eine gemeinsame und synchrone Zeitbasis zur Verfügung zu stellen. Zur Uhrzeitsynchronisation kann beispielsweise auf den Standard IEEE 1588 [IEEE-1588] zurück gegriffen werden.
- **Protokollmodifikation**
Die lokale Telegrammverarbeitung wird durch ein modifiziertes Ethernetprotokoll unterstützt. Dieses ist speziell auf eine hohe Verarbeitungsgeschwindigkeit optimiert.

In Tab. 2.1 sind aktuelle Industrial Ethernet Lösungen und ihre minimalen Zykluszeiten aufgeführt. Die Zykluszeit bezeichnet die Zeitspanne, welche verstreicht, bis ein Teilnehmer, der gerade eine Nachricht verschickt hat, erneut die Erlaubnis zum Senden erhält. Sie ist ein Maßstab für die Echtzeitfähigkeit.

Industrial Ethernet Lösung	minimale Zykluszeit [ms]
Modbus/TCP	<100
FoundationFieldbus HSE	>100
Powerlink	>0,2
PROFINet CBA (NRT)	100
PROFINet CBA (RT)	<10
PROFINet IO (RT)	<10
PROFINet CBA (IRT)	>0,2
SERCOS III	>0,03
EtherCAT	>0,03
SynqNet	<0,1

Tab. 2.1: Zykluszeiten von unterschiedlichen Industrial Ethernet Lösungen [Koh-07]

Im Vergleich zu klassischen Feldbusssystemen haben viele der aufgeführten Lösungen den Vorteil, dass eine Kombination von herkömmlichen Ethernetkomponenten und Industrial Ethernet möglich ist. Anders als bisher ist es also ausreichend, nur ein Netzwerk aufzubauen, das die gesamte Kommunikation in einem Materialflusssystem überträgt. Damit bietet sich Industrial Ethernet als grundlegende Übertragungstechnologie für moderne Materialflusssysteme an.

2.3.4 Multiagentensysteme

Im Folgenden sollen der Begriff des Softwareagenten näher definiert, grundlegende Agentenarchitekturen vorgestellt und Multiagentensysteme beschrieben werden.

Definition eines Softwareagenten

Der Begriff Agent wird in der Informatik vielfältig verwendet, so dass keine einheitliche Definition existiert. Eine Auswahl der in der Literatur vorhandenen Definitionen liefert die folgende Aufzählung:

- Etzioni [Etz-95]

Ein Software Agent ist laut Etzioni ein Computerprogramm, das sich ähnlich wie ein hilfreicher Mensch bzw. ein Assistent verhält. Er führt Arbeiten aus, die für den Auftraggeber entweder nicht durchführbar sind oder ihn zu viel Zeit kosten würden.

- Odell [Ode-00]

Odell nimmt die Objektorientierung in der Softwareentwicklung als Definitionsgrundlage und beschreibt einen Agenten als ein Softwareobjekt, das selbst entscheiden kann, ob es eine Aufgabe bearbeitet oder nicht.

- Woolridge [Woo-95]

Für Woolridge hat ein Agent folgende Eigenschaften:

- Autonomie

Agenten operieren ohne einen direkten Eingriff von außen und kontrollieren ihre Handlungen und ihren internen Zustand selbst.

- Kommunikationsfähigkeit

Agenten interagieren mit anderen Agenten (und evtl. Menschen) in einer eigenen Agentenkommunikationssprache.

- Reaktivität

Agenten nehmen ihre Umwelt wahr – das kann die reale Welt, ein Benutzer, der über eine entsprechende Schnittstelle interagiert, andere Agenten, das Internet oder eine Mischung aus dem Vorgenannten sein – und reagieren in einer angemessenen Zeitspanne auf Veränderungen dieser Umwelt.

- Proaktivität
Agenten reagieren nicht nur auf ihre Umwelt, sondern sind in der Lage, zielgerichtetes Verhalten an den Tag zu legen und selbständig die Initiative zu ergreifen.
- Dumke [Dum-03]
Dumke beschreibt einen Agenten wie folgt: „Ein Software-Agent (software agent) ist ein Software-System, welches innerhalb einer definierten Umgebung, dessen Bestandteil er auch ist, in einer bestimmten Zeit, mit einer ihm immanenten Verhaltensweise und einer vorgegebenen Zielstellung wirkt bzw. agiert.“ Hierzu ist sowohl autonomes Handeln als auch Intelligenz notwendig, um den Agenten dazu in die Lage zu versetzen, seine Aufgabe zu bewältigen. Hinzu kommt laut Dumke die Fähigkeit, sich in seiner Umgebung zu bewegen (Mobilität), falls dies innerhalb der Agentenplattform, die ihm als Umwelt dient, vorgesehen ist.

In der vorliegenden Arbeit wird der Begriff Agent im Sinne der Definition von Dumke angewandt.

Agentenarchitekturen

Es gibt zwei grundsätzliche Architekturen von Agenten und eine Mischform aus diesen:

- Reaktiver Agent

Reaktive Agenten leiten aus Reizen, die aus der Umwelt kommen, mittels einfacher Verhaltensregeln mehr oder weniger direkt eine Aktion ab. Sie sind nicht in der Lage, komplexe Schlussfolgerungen aus ihren Informationen zu ziehen und verfügen über kein Modell ihrer Umwelt.

Als Vorbild für diese Art der Aufgabenbewältigung dienen Vorbilder aus der Natur. Ameisen und Bienen organisieren sich in einem Staat und erzeugen durch die Handlungen einzelner, einfacher Individuen ein geordnetes und komplexes Gesamtverhalten. Der Mehrwert, welcher durch dieses Zusammenspiel entsteht, wird als Emergenz bezeichnet.

- Kognitiver Agent

In dieser Ausprägung, die auch als deliberativ bezeichnet wird, werden Agenten als intelligente Individuen betrachtet, welche über ausreichendes Wissen verfügen, um ihre Aufgabe selbständig zu erledigen. Kognitive Agenten sind sehr viel komplexer aufgebaut als reaktive. Ein kognitiver Agent verfolgt Pläne, um sein Ziel zu erreichen. Zusätzlich sind kognitive Agenten in der Lage, mit anderen Agenten zusammenzuarbeiten. Ihre Architektur wird meist nach dem BDI-Modell (Belief-Desire-Intention-Modell) gestaltet, welches von Rao [Rao-91] entwickelt wurde:

- Belief (Glaube)

Hiermit sind alle Informationen gemeint, welche ein Agent über seine Umwelt besitzt. Dies umfasst auch Regeln, welche es ermöglichen, aus vorhandenen Informationen Schlüsse zu ziehen. Bewusst wurde dabei nicht der Begriff Knowledge (Wissen) gewählt. So soll zum Ausdruck gebracht werden, dass die Informationen, über welche ein Agent verfügt, nicht unbedingt wahr sein müssen, sondern lediglich seiner subjektiven Sichtweise entsprechen.

- Desire (Wunsch, Verlangen)

Ein Agent hat den Wunsch, besondere Zustände oder Situationen herbei zu führen. Ein Bietagent, welcher bei Ebay den Menschen in einer Auktion vertritt, hat beispielsweise den Wunsch, die Auktion zu gewinnen und einen möglichst niedrigen Preis zu erzielen. Desires dürfen prinzipiell im Widerspruch zueinander stehen. In diesem Fall muss der Agent abwägen, welchem Desire er folgt. Bei Goals (Zielen) handelt es sich um einen Sonderfall der Desires. Goals müssen immer konsistent zueinander sein.

- Intention (Vorhaben)

Dieser Begriff beschreibt die Handlung, zu welcher sich ein Agent entschlossen hat. Dabei können auch komplexe Pläne verfolgt werden. Pläne sind eine Aneinanderreihung von Handlungen.

- Hybrider Agent

Bei hybriden Agenten handelt es sich um Mischformen aus kognitiven und reaktiven Konzepten. Dabei steht es dem Entwickler frei, wie er die Gewichtung

zwischen diesen beiden Ausprägungen festlegt. Hybride Agenten werden häufig in einer Schichtarchitektur aufgebaut, in welcher die oberen Schichten eher die strategisch / planenden Aufgaben übernehmen und die unteren Schichten für reaktive Aufgaben zuständig sind.

Multiagentensystem

Multiagentensysteme sind Systeme, deren Hauptakteure Softwareagenten sind. Ferber definiert ein Multiagentensystem (MAS) als ein System, das die folgenden Elemente enthält [Fer-99]:

- Eine Umgebung e , die ein Raum ist und allgemein ein Volumen hat.
- Eine Menge O von Objekten. Diese Objekte sind in e befindlich und es ist möglich, diesen Objekten jederzeit eine Position in e zuzuweisen. Die Objekte sind passiv (bis auf eine Ausnahme, siehe nächsten Punkt), d. h. sie können wahrgenommen, hergestellt, verändert und zerstört werden.
- Eine Menge $\Gamma \subseteq O$ von Agenten. Diese sind die aktiven Objekte des Systems.
- Eine Menge von Relationen R , welche Objekte und Agenten in Beziehung zueinander setzen.
- Eine Menge von Operationen OP , welche Agenten erlauben, Objekte zu modifizieren, zu kreieren und zu zerstören.
- Die Regeln des Universums, welche die Aufgabe haben, den Effekt von Operationen und die Reaktion der Umwelt darauf zu repräsentieren.

Unter Umständen kann gelten: $\Gamma = O$. Außerdem kann e aus der leeren Menge bestehen. Ist beides der Fall, dann definiert R ein Netzwerk von Agenten, welches jeweils einen Agenten mit einer Menge anderer Agenten direkt verbindet.

Seit dem Jahr 1996 existiert eine einheitliche Spezifikation für Agentensprachen, -interaktionsprotokolle und -plattformen, welche für die Entwicklung von Multiagentensystemen eingesetzt werden können. Die FIPA (Foundation for Intelligent Physical Agents) [FIPA-05], welche diese Spezifikation erarbeitet hat, wurde inzwischen offizieller Teil der Normungsorganisation IEEE (Institute of Electrical and Electronics Engineers) und bietet damit eine einheitliche Grundlage für die Entwicklung von Multiagentensystemen.

2.4 Stand der Forschung

Die Logistik produzierender Unternehmen ist in Zukunft mit neuen Anforderungen an Flexibilität und Geschwindigkeit [Gud-99, Gra-00, Eis-01] konfrontiert. Viele Kunden bestellen via Internet und erwarten eine Lieferung innerhalb der nächsten 24 Stunden.

Die Produzenten versuchen, den schnellen Veränderungen der Märkte mit einer stärkeren Vernetzung ihrer logistischen Strukturen zu begegnen [Ber-08]. Unabhängige Unternehmen schließen sich kurzfristig zu sogenannten virtuellen Unternehmen bzw. globalen Verbänden zusammen, um ihre Kompetenzen zu bündeln und gemeinsam schneller auf eine neue Nachfrage reagieren zu können. Dies wiederum führt zu einer Zunahme an komplexen, unternehmensinternen und -übergreifenden logistischen Prozessen [Fre-04]. Die Unternehmen müssen also künftig ein hohes Maß an operativer, struktureller und vor allem strategischer Beweglichkeit an den Tag legen [Vol-96].

Das Produkt der Zukunft sollte aber nicht nur schnell geliefert werden, es sollte auch genau auf den Kunden zugeschnitten sein (Mass-Customization). Die dazu notwendige kundenindividuelle Produktion unterscheidet sich grundlegend von der bisherigen Massen- bzw. Serienproduktion und hat damit auch Auswirkungen auf die Logistik. Das Bestellvolumen pro Produkt nimmt immer mehr ab, was zu einer Atomisierung der Losgrößen in der Produktion führt [Weh-04] und eine Vervielfachung der innerbetrieblichen Transporte zur Folge hat [Wil-06].

Damit wird auch die Steuerung der innerbetrieblichen Logistik immer schwieriger, da deren Komplexität direkt mit der Anzahl der Transporte, aber auch mit ihren Beziehungen und Abhängigkeiten untereinander korreliert. Herkömmliche automatisierte Materialflusssysteme gelten aber als eher unflexibel und sind ab einer gewissen Größe bzw. Komplexität nur noch schwer zu beherrschen [Gün-05b]. Vor allem bis Mitte der 80er Jahre gab es keine Alternativen zu zentralen, streng hierarchischen Steuerungsarchitekturen. Ein Beispiel aus dem Bereich der Fertigung ist Computer Integrated Manufacturing (CIM) [Mes-90, Noa-90]. Hier wurde versucht, durch eine strenge Hierarchie mit starkem Determinismus und zentraler Datenhaltung Planungs- und Fertigungsaufgaben zu beherrschen. Gerade dieser Determinismus wird als Ursache dafür betrachtet, dass das Konzept heute als gescheitert gilt [Lan-99].

Es müssen also technische Lösungen gefunden werden, welche die Wandlungsfähigkeit in Unternehmensstrukturen durch technische Anpassungsfähigkeit unterstützen und nicht zu starr vorher bestimmt sind [Wes-97, Wie-98, Küh-99, Rei-99]. Selbststeuerende logistische Prozesse, welche sich durch ein teilautonomes Handeln von Objekten auszeichnen, gelten als Lösung für diese Problematik [Ber-08]. Aktuelle Forschungsarbeiten bereiten schon einen Paradigmenwechsel hin zu dezentral organisierten und selbstgesteuerten Materialflusssystemen vor [Weh-08]:

- Variable Fertigungsstrukturen

Das Forschungsprojekt „MATVAR“ erarbeitete Materialflusssysteme für variable Fertigungsstrukturen im dynamischen Produktionsumfeld [Gün-00a]. Hier wurde die Fertigung in dezentrale Segmente unterteilt, welche leicht an veränderte Anforderungen anpassbar sind. Im Mittelpunkt des Forschungsprojektes stand die Untersuchung, Konzeption, Bewertung und Erprobung flurfreier Materialflusstechnik und deren Schnittstellen zur Fertigung [Gün-00b]. Dazu wurde am Lehrstuhl für Fördertechnik Materialfluss Logistik der Technischen Universität München eine Elektrohängebahnanlage in Betrieb genommen, welche in mehreren Stufen automatisierbar ist [Gün-98].

- Wandlungsfähige Unternehmensstrukturen

Im SFB 467 „Wandlungsfähige Unternehmensstrukturen für die Serienproduktion“ wurden Methoden, Modelle und Verfahren zur Erhöhung der Wandlungsfähigkeit von Unternehmen erforscht. Dabei geht es um die Fragestellung, wie man Unternehmensstrukturen so gestalten kann, dass sie auf die schnellen und unvorhersagbaren Schwankungen der für den Unternehmenserfolg wesentlichen Faktoren reagieren können [SFB-467]. Dabei werden Organisationsformen eingesetzt, welche auf autonomen, selbständigen Leistungseinheiten basieren und die durch gelenkte Selbstorganisation ein wandlungsfähiges Gesamtsystem bilden.

- Wandelbare Logistikstrukturen

Der SFB 582 „Marktnahe Produktion individualisierter Produkte“ lieferte im Teilbereich M2 „Wandelbare Logistikstrukturen in Minifabriken“ Anforderungen und Lösungen für eine innerbetriebliche Logistik, welche speziell auf die kundenindividuelle Produktion von Gütern ausgelegt ist. Die auf diesen Ergebnissen aufbauende Dissertation von Wilke [Wil-06] entwickelt dazu ein Konzept

für dezentral gesteuerte Materialflussmodule, welche nach den Vorgaben der funktionsorientierten Modularisierung gestaltet sind (s. Kap. 4.1.3). Dabei setzt Wilke sich im Speziellen mit den externen Schnittstellen und Wegplanungsalgorithmen für diese Module auseinander.

Heinecker entwirft eine „Methodik für die Gestaltung und Bewertung wandelbarer Materialflusssysteme“ [Hei-06]. Hier werden Kennzahlen entwickelt, die eine Beurteilung der Wandlungsfähigkeit eines Materialflusssystems ermöglichen sollen.

Agentensysteme bilden in einigen Forschungsarbeiten die Grundlage für selbststeuernde Einheiten. Das Ziel dieser agentenbasierten Ansätze ist es, Heterogenität und Komplexität von verteilten Systemen beherrschbar zu machen [Her-03]. Die Folgende Aufzählung gibt einen Überblick über die Forschungsaktivitäten in diesem Bereich:

- **Multi-Agenten-System für Produktionssysteme**
Ritter [Rit-03] entwirft „Ein Multi-Agenten-System für mobile Einrichtungen in Produktionssystemen“. Schwerpunkt seiner Arbeit ist die Entwicklung von Anforderungen, Interaktionsprotokollen und Konzepten für eine speziell auf Produktionssysteme ausgerichtete Agentensteuerung.
- **Selbststeuerung logistischer Prozesse**
Der SFB 637 „Selbststeuerung logistischer Prozesse – Ein Paradigmenwechsel und seine Grenzen“, hat das Ziel, die Selbststeuerung als ein neues Paradigma für logistische Prozesse systematisch zu erforschen. Hierbei wird besonders der überbetriebliche Transport betrachtet, aber auch die Auswirkungen und die Weiterentwicklung der sozio-technischen Logistiksysteme durch veränderte Steuerungsparadigmen und -prozesse werden näher beleuchtet [Fre-04].
- **Dezentral gesteuerte Stetigförderermodule**
Mayer [May-09] entwickelt dezentrale, modulare Stetigförderer, welche beliebig hinter- bzw. nebeneinander angeordnet werden, die Topologie selbständig erkennen und blockadefreie Wege für Transporteinheiten berechnen können.

- Internet der Dinge

Das BMBF-Forschungsprojekt „Internet der Dinge – Wandelbare Echtzeit-Logistiksysteme auf Basis Intelligenter Agenten für den produktionsnahen Bereich“ kombiniert RFID-Technologien mit der Agententechnik. Frei nach dem Motto „Selbst ist das Paket“ sollen sich die logistischen Objekte in Zukunft selbst ihren Weg zum Ziel suchen, genauso wie die Email im Internet. Dabei werden Kommunikation, spontane Vernetzung und gemeinsame, domänen- und anwendungsspezifische Ontologien untersucht und definiert.

Zusammenfassend lässt sich sagen, dass die meisten bisher genannten Forschungsarbeiten sich mit der übergeordneten Struktur einer dezentralen Steuerung oder mit speziellen Insellösungen befassen und die Module als Black-Box betrachten. Die Auswirkungen, welche ein solches Konzept auf die Tätigkeiten bei Realisierung oder Umbau bzw. Erweiterung einer Materialflussanlage haben, werden dabei nicht genauer beleuchtet.

Die vorliegende Arbeit ist im Rahmen des Forschungsprojektes Internet der Dinge entstanden. Der Schwerpunkt liegt sowohl auf der Entwicklung und dem internen Aufbau von selbstgesteuerten Modulen auf Basis von Softwareagenten als auch auf den Auswirkungen eines solchen Konzeptes, welche über den Lebenszyklus eines Materialflusssystemes betrachtet werden.

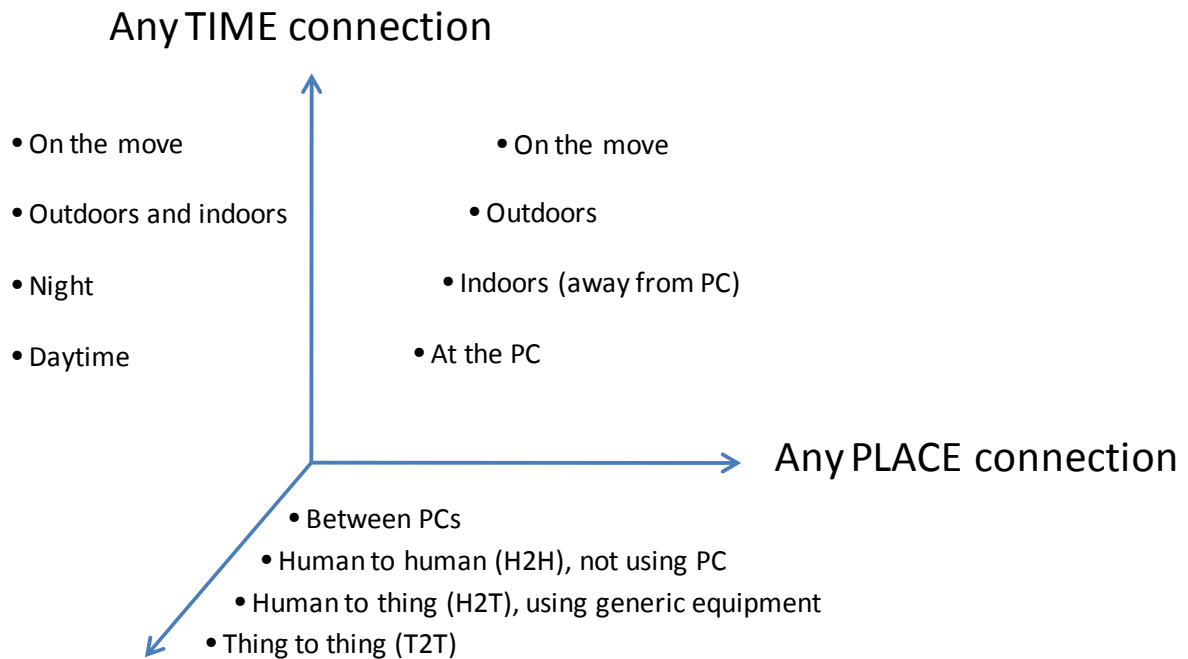
3 Das Internet der Dinge

3.1 Die Vision vom Internet der Dinge

Weiser beschreibt bereits 1991 die Vision einer von vielen kleinen Computern durchsetzten Welt, die den Menschen unmerklich unterstützen sollen [Wei-91]. Es sei, so Weiser, nur eine Frage der Zeit, bis die bislang bekannten Personalcomputer in den Hintergrund rücken. Organizer waren für ihn nur der erste Schritt hin zu einer neuen Technologie. Stark vernetzte Kleinstrechner sollen die Bedürfnisse des Menschen selbständig erkennen und ihm eine Hilfestellung bei alltäglichen Problemen geben. Bekanntestes Beispiel für diese Idee ist wohl der Kühlschrank, welcher seinen Inhalt überwacht und bei Bedarf Milch nachbestellt. Diese Vision allgegenwärtig verteilter, aber im Hintergrund wirkender Rechner wird als Ubiquitous Computing oder UbiComp bezeichnet. UbiComp ist ein relativ weit gefasster Oberbegriff für eine Technologie, welche auf folgenden vier Grundprinzipien beruht [Die-03]:

- **Dezentralisierung**
Funktionalitäten und Aufgaben werden auf viele kleine Recheneinheiten verteilt.
- **Diversifikation**
Die Recheneinheiten haben eine streng abgegrenzte Funktionalität, und Soft- und Hardware sind genau auf diese abgestimmt.
- **Konnektivität**
Die Einheiten des UbiComp können miteinander kommunizieren. Hierzu sind sowohl Technologien als auch Protokolle notwendig.
- **Simplizität**
Die Recheneinheiten sind einfach zu bedienen bzw. arbeiten im Hintergrund, ohne dass der Benutzer eingreifen oder auch nur darüber nachdenken muss.

Diese Entwicklung spannt eine neue Dimension in der Art der Kommunikation auf. Bisher wurde daran gearbeitet, dass jedermann und zu jeder Zeit kommunizieren konnte. Nun sollen sich auch die Dinge miteinander „unterhalten“ (s. Abb. 3.1).



Any THING connection

Abb. 3.1: Eine neue Dimension der Kommunikation [ITU-05]

In diesem Zusammenhang entwickelte sich die Vision eines Internet der Dinge, in welchem sich auch die Gegenstände des Alltags genauso wie die Rechner im Internet miteinander vernetzen und untereinander abstimmen können. Der Bezug zum Internet wurde gewählt, weil sich die dort eingesetzten Technologien bewährt haben:

- Das Internet verzeichnete in den letzten Jahren ein exponentielles Wachstum. Dies gilt sowohl für die Anzahl der angeschlossenen Teilnehmer als auch für die Zunahme der Übertragungsgeschwindigkeit auf den schnellsten Verbindungen im Netz. Trotzdem haben sich für mehr als zehn Jahre die Internet-Protokolle nicht signifikant verändert [Han-06] und das Netz bleibt dennoch einsatzfähig.
- Es gibt keine zentrale, koordinierende Instanz im Internet. Das Internet ist daher ein hochgradig robustes System. Ausfälle von einzelnen Teilnehmern beeinträchtigen das Gesamtsystem nicht.
- Verschiedene Kommunikationstechnologien, wie zum Beispiel Ethernet [IEEE-802.3] oder Wireless Local Area Network WLAN [IEEE-802.11], die auch für das Internet Verwendung finden, wurden genau definiert. Die dafür benötigte Hardware kann daher in großen Stückzahlen hergestellt werden, was wiederum zu niedrigen Preisen führt.

- Ein Internet-User muss über relativ wenige Kenntnisse verfügen, um am Internet teilzunehmen. Durch eine klare Schichtenarchitektur (ISO/OSI Referenzmodell [ISO-7498]) muss sich der Anwender nicht um physikalische Prozesse oder Datenkodierungen kümmern. Er bewegt sich nur auf der Anwendungsschicht und muss die unterlagerten Schichten nicht kennen.

Eine besondere Ausprägung des Internet der Dinge bezogen auf die Anwendung in der Logistik beschreibt Fleisch [Fle-05]. Eine Verbindung aus Internettechnologien und der Radio Frequenz Identifikation (RFID), die eine berührungslose Datenübertragung gestattet (s. Kap. 2.3.1), soll zukünftige Geschäftsprozesse und logistische Steuerungsaufgaben einfacher, robuster und besser beherrschbar machen. Dazu wird jede Transporteinheit mit einem eigenen Transponder ausgestattet. Dieser kann zusätzliche Daten über den Produktions- und Transportvorgang entweder direkt speichern oder es wird eine Identifikationsnummer, der sogenannte Electronic Product Code (EPC), abgelegt. Dieser ermöglicht eine Zuordnung von Daten, welche beispielsweise in einer Datenbank gespeichert sind, zur Transporteinheit. Durch diese eindeutige Zuordnung von Informations- und Warenfluss wird z.B. die Dokumentation über den Lebenszyklus auch über verschiedene Ebenen der Supply Chain erleichtert. Außerdem können Prozesse, welche bisher manuell durchgeführt werden, beispielsweise die Verbuchung der Waren am Wareneingang, beschleunigt werden.

Die vorliegende Dissertation bezieht in den Begriff Internet der Dinge zusätzlich die Agententechnologie (s. Kap. 2.3.4) ein. Alle Bestandteile eines Materialflusssystems, also sowohl die Fördertechnik als auch die Transporteinheiten, werden mittels Softwareagenten mit einer eigenen Intelligenz ausgestattet. Diese Agenten steuern im Zusammenspiel den Transport von Waren vom Start- zum Zielpunkt.

Eine Analogie (s. Abb. 3.2) soll den Zusammenhang zwischen dem Transport von Einheiten in der Intralogistik und demjenigen im Internet verdeutlichen:

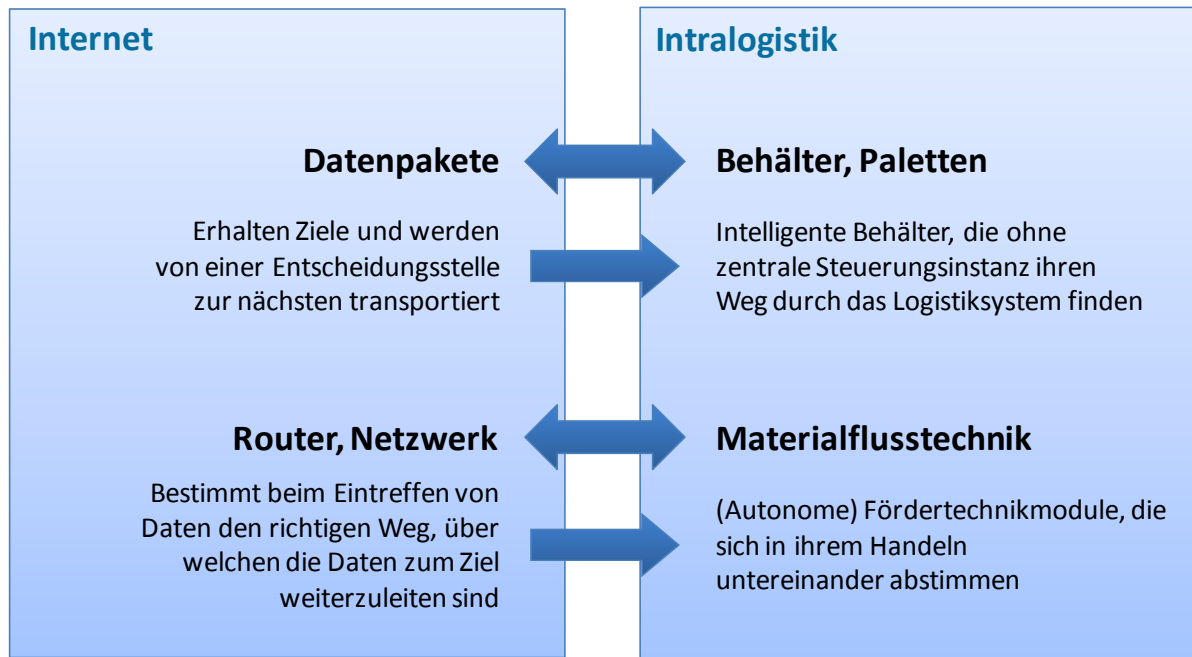


Abb. 3.2: Analogie zwischen Internet und Intralogistik

Genau so wie im Internet die Datenpakete ohne eine übergeordnete Steuerungsinstanz transportiert werden, sollen auch die Transporteinheiten in einem Materialflusssystem ohne das Eingreifen einer Zentralinstanz zum Ziel kommen. Jeweils ein Softwareagent pro Transporteinheit koordiniert alle dazu notwendigen Schritte. Ebenso wie jede Email im Internet über einen Header verfügt, welcher z.B. Daten zum Adressaten speichert, wird an jeder Transporteinheit ein Transponder angebracht, welcher die für den Transport notwendigen Informationen enthält.

Die Materialflusstechnik entspricht in dieser Analogie den Datenleitungen bzw. Routern im Internet. Letztere sind intelligente Komponenten, welche im Internet die gleichmäßige Verteilung der Datenpakete übernehmen und auch den Weg bestimmen, den z.B. eine Email zum Ziel nimmt. Genau so sollen im Internet der Dinge die Fördertechnikmodule dezentral über den Weg einer Transporteinheit entscheiden. Dazu verfügen sie, genauso wie die Transporteinheiten, über einen Softwareagenten, der für alle Belange eines Fördertechnikmoduls zuständig ist. Beispielsweise könnte der Ablauf an einer Fördertechnikweiche folgendermaßen aussehen: Die Weiche liest, während sie von einer Transporteinheit passiert wird, deren Transponder aus und kennt damit das nächste anzufahrende Ziel. Unter der Voraussetzung, dass dem Softwareagenten der Weiche die Anlagentopologie und deren aktuelle Auslastung bekannt sind, kann dieser die Richtung bestimmen, in welche die Trans-

porteinheit weiter geleitet werden muss. Die Weiche stellt sich also, ohne Abstimmung mit einer übergeordneten Instanz, vollkommen autonom und hat damit eine ähnliche Funktion wie ein Router im Internet.

Zusammenfassend lässt sich sagen, dass anstelle hochkomplexer und damit fehleranfälliger zentraler Steuerungssysteme (s. Kap. 2.1.2) dezentral gesteuerte Einheiten zum Einsatz kommen sollen, welche genügend Intelligenz aufweisen, um auf die jeweiligen Situationen selbständig reagieren zu können. Materialflussmodule wie Weichen oder Verschiebewägen sollen so intelligent werden, dass sie Staus und Blockaden selbstständig vermeiden und auf schwankende Auftragslasten oder Störungen reagieren. Alle Einheiten des Systems müssen dazu definierte Zuständigkeiten, Aufgaben und Funktionen haben. Zusätzlich müssen sie über die Fähigkeit verfügen, diese eigenständig zu erfüllen bzw. sich dazu mit anderen Einheiten abzustimmen. Anders als bisher muss einem solchen System also nur noch mitgeteilt werden, „WAS“ es zu tun hat, z.B. durch entsprechende Transportaufträge, während sich das „WIE“ bzw. die dafür notwendigen Einzelschritte aus dem gerichteten Zusammenspiel der individuellen Einheiten ergeben.

Ein Materialflussrechner als zentrale, steuernde Komponente ist in einem solchen System nicht mehr notwendig. Der Ausfall einer oder einiger weniger Komponenten kann somit nicht zu einem Gesamtzusammenbruch des Systems führen. Sind Komponenten vorhanden, welche die Arbeit übernehmen können, so ist das System aus sich selbst heraus in der Lage, die Transportströme entsprechend umzuleiten. Diese Eigenschaft macht das System wandelbar, da auch auf neue Ereignisse reagiert werden kann, ohne dass diese explizit vorgeplant wurden (s. Kap. 1.2.2).

3.2 Die Bausteine des Internet der Dinge

Ein erster Schritt zur Verwirklichung dieser Vision ist die Zerlegung eines Materialflusssystemes in dezentrale Einheiten. Wie bereits beschrieben gibt es im Internet der Dinge autonome Fördertechnikmodule und mit RFID-Transpondern ausgestattete Transporteinheiten. Diese beiden Einheiten reichen aber noch nicht aus, um alle Funktionalitäten eines Materialflusssystemes abzubilden.

Denn in vielen Materialflusssystemen werden Aufgaben erledigt, welche keiner expliziten mechanischen Repräsentanz wie Fördertechnikmodulen oder Transporteinhei-

ten zugeordnet werden können. Ein Beispiel dafür ist die Visualisierungsumgebung, welche es dem Anlagenbetreiber ermöglicht, die aktuellen Systemzustände zu überwachen und Fehlersituationen zu erkennen. Eine solche Visualisierungsumgebung ist lediglich ein Softwaredienst.

Zur Steuerung eines Materialflusssystems sind somit drei verschiedene Typen von autonomen Einheiten mit jeweils eigenen Fähigkeiten und Eigenschaften notwendig, nämlich:

- Module
- Transporteinheiten
- Dienste

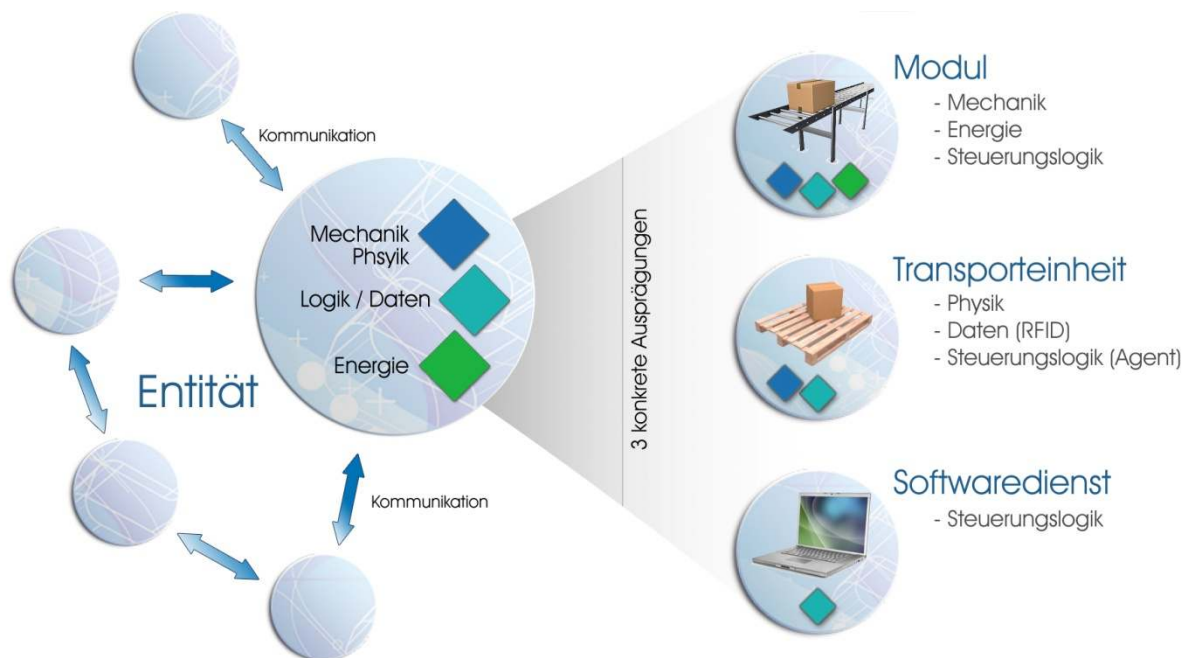


Abb. 3.3: Die Grundeinheiten des Internet der Dinge sind kooperierende Entitäten

In Anlehnung an das Forschungsgebiet der Multiagentensysteme sollen im Folgenden diese nicht mehr weiter zerlegbaren Funktionseinheiten im Internet der Dinge als Entität bezeichnet werden: Entitäten erfüllen autonom eine bestimmte logistische Funktion und interagieren zu diesem Zweck mit anderen gleichberechtigten Entitäten (s. Abb. 3.3). Jeder Entität wird eine eigene Intelligenz in Form eines Softwareagenten zugeordnet. Dieser befähigt sie, ihre eigenen Aufgaben wahrzunehmen, mit anderen zu kommunizieren oder Verhaltensstrategien umzusetzen. In den folgenden

Kapiteln werden die drei Ausprägungen einer Entität für das Internet der Dinge definiert.

3.2.1 Transporteinheiten

Eine Transporteinheit (TE) ist die kleinste Menge, welche im Materialflusssystem einzeln bewegt wird, also z.B. VDA-Kleinladungsträger, Behälter, Paletten oder Kartons. TE verfügen auch im Internet der Dinge weder über eine Mechanik, wie Antriebe oder Sensoren, noch über eine eigene Recheneinheit oder eine Energieversorgung. Jeder TE wird ein Softwareagent zugeordnet, der beispielsweise auf einem stationären Rechner ausgeführt wird. Zwar ist es theoretisch denkbar, jede TE mit einem eigenen Controller auszustatten, allerdings ist dies aus Kostengründen derzeit noch nicht sinnvoll. Der Agent ermöglicht es einer Transporteinheit, mit anderen Entitäten zu kommunizieren, die Ausführung von Funktionen anzufordern und Daten auszutauschen.

Im Internet der Dinge sind alle TE mittels RFID-Transpondern eindeutig identifizierbar. Auf diesen sollen so viele Informationen wie möglich dezentral abgelegt werden, mindestens aber die Zieladresse und auch Zwischenziele bzw. anzufahrende Bearbeitungsstationen. Zusätzlich ist es denkbar, Informationen über das Packmuster einer Palette oder die Geometrie eines Behälters zu speichern, um problemlose Lastübergaben zu ermöglichen. Darüber hinaus kann sogar die Programmlogik des Softwareagenten so lange auf dem Transponder gespeichert werden, bis sie wieder benötigt wird, sofern der Speicher dafür ausreichend ist.

Die klassische Lösung, alle Daten zentral in einer Datenbank abzulegen und anhand einer eindeutigen Nummer bei Bedarf abzufragen (Data-On-Network), soll hier bewusst vermieden werden. Einerseits verringert eine solche Zentralkomponente die Robustheit des Gesamtsystems, andererseits kann durch eine lokale Datenhaltung (Data-On-Chip) auch das Kommunikationsaufkommen reduziert werden, indem die Daten vor Ort verarbeitet bzw. verdichtet werden.

Verhindern beispielsweise unzureichende Speicherkapazitäten heutiger RFID-Systeme, dass alle Informationen direkt an der Transporteinheit abgelegt werden, so können diese vom Agenten der TE verwaltet und auf Anfrage bereitgestellt werden.

3.2.2 Module

Module sind autonom agierende Fördertechnikelemente wie z.B. Regalbediengeräte, Rollenbahnen, Elektrohängebahnkatzen, Verschiebewagen oder Kommissionierstationen, deren Systemgrenzen nach den Regeln der funktionsorientierten Modularisierung gezogen werden sollen (siehe 4.1.3). Soweit wirtschaftlich und technisch machbar, ist daher eine klare Kapselung der Elektrik, Mechanik, Steuerungshardware und Steuerungssoftware anzustreben. In Ausnahmefällen ist es erlaubt, die Agenten mehrerer Module auf einer gemeinsamen Rechenplattform auszuführen – was zu geringeren Kosten, aber auch zu einer niedrigeren Robustheit und Transparenz des Systems führt.

Module erfüllen eine logistische Funktion wie z.B. Transportieren, Sortieren oder Lagern und sind als Dienstleister aufzufassen, die den Transporteinheiten die zur Erfüllung ihres Arbeitsablaufs notwendigen Funktionen anbieten. Außerdem handelt es sich meist um Punkte, an welchen Entscheidungen (z.B. Wegentscheidungen) getroffen werden, wobei es in diesem Zusammenhang keine Rolle spielt, ob diese Entscheidungen mittels Softwarealgorithmen oder über eine Benutzerschnittstelle von einem Menschen getroffen werden.

3.2.3 Dienste

Dienste sind reine Softwareprogramme und übernehmen Aufgaben, die nicht einem einzelnen Modul oder einer TE zugeordnet werden können. Beispiele für Dienste sind eine Visualisierungsumgebung, eine Datenaustauschplattform oder ein Verkehrsleitsystem. Sie treten lediglich in Form eines Softwareagenten auf und dienen z.B. der Koordination, Optimierung, Überwachung und Visualisierung. Dienste können auch eine Schnittstelle zu externen Systemen wie dem Lagerverwaltungsrechner realisieren.

4 Konzeption und Entwicklung von Modulen

Fördertechnikmodule spielen beim Aufbau eines Materialflusssystems eine zentrale Rolle und sind im Vergleich zu den übrigen Entitäten die komplexesten Systembestandteile. Daher konzentriert sich die vorliegende Arbeit auf die Betrachtung dieser Entitäten. Im Folgenden wird die Gestaltungsmethodik zum Systementwurf und zur Modulentwicklung vorgestellt, und sowohl der allgemeine Aufbau von Modulen als auch Vorgehensweisen zu ihrer Entwicklung und ihrer Integration in das Gesamtsystem erläutert.

4.1 Gestaltungsmethodik und Vorgehensweise

Das Internet der Dinge ist eine neue Denkweise in der Intralogistik. Trotzdem kann auf bereits existierende Methoden zurückgegriffen werden, welche sich gut auf die Konzeption und Entwicklung von Fördertechnikmodulen für eine dezentrale Materialflussteuerung anwenden lassen. Aufbauend auf die Entwicklungsmethodik für mechatronische Systeme nach [VDI-2206] wird ein Vorgehensmodell für das Internet der Dinge vorgestellt, welches drei Schritte vorsieht. Den ersten Schritt bildet die funktionsorientierte Modularisierung, welche anschließend erläutert wird. Sie bildet auch die Grundlage für eine Strukturierung der Module nach Funktionsklassen, welche zum Abschluss des Kapitels beschrieben wird.

4.1.1 Entwicklungsmethodik für mechatronische Systeme

Der Begriff Mechatronik beschreibt ursprünglich die Verschmelzung der Fachdisziplinen Mechanik bzw. Maschinenbau und Elektronik bzw. Elektrotechnik. Mit dem Aufkommen der Mikroelektronik und der Mikroprozessortechnik ist die Informationstechnik als weiterer Bestandteil der Mechatronik hinzu gekommen. Eine einheitliche Definition des Begriffes ist nicht vorhanden, vielmehr ist eine ständige Weiterentwicklung des Begriffes im Sinne einer Technologieerweiterung zu beobachten [VDI-2206].

In der vorliegenden Arbeit soll die Definition von Isermann [Ise-99] Verwendung finden: „Mechatronik ist ein interdisziplinäres Gebiet, bei dem folgende Disziplinen zusammenwirken: mechanische und mit ihnen gekoppelte Systeme, elektronische

Systeme, Informationstechnik. Dabei ist das mechanische System im Hinblick auf die Funktionen dominierend. Es werden synergetische Effekte angestrebt, die mehr beinhalten als die reine Addition der Disziplinen.“

Ein Materialflusssystem ist ein mechatronisches System, da hier mechanische, elektrotechnische und informationstechnische Komponenten miteinander interagieren müssen. Zudem sind die Module im Internet der Dinge ebenso mechatronische Systeme, denn auch diese verfügen über die erwähnten drei Schichten.

Die VDI-Richtlinie 2206 beschreibt eine „Entwicklungsmethodik für mechatronische Systeme“ und stellt als allgemeinen Handlungszyklus auf der Makroebene das sogenannte V-Modell vor (siehe Abb. 4.1).

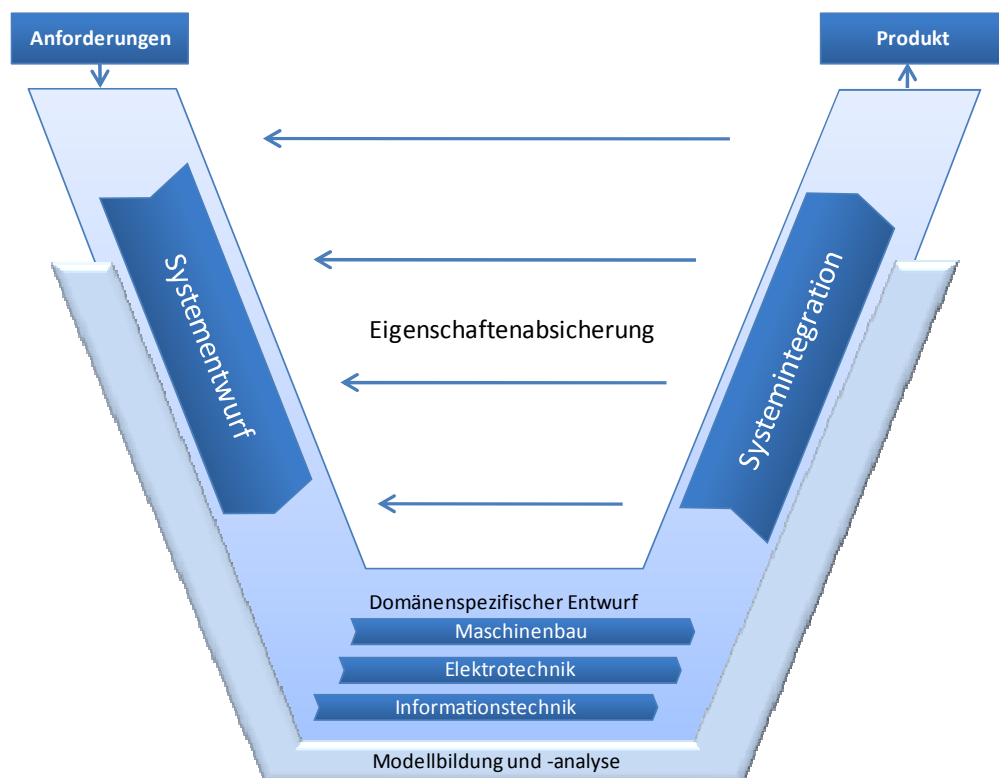


Abb. 4.1: V-Modell als Makrozyklus zur Entwicklung mechatronischer Systeme [VDI-2206]

Das V-Modell ist ein Vorgehensmodell, das eine logische Abfolge wesentlicher Teilschritte bei der Entwicklung mechatronischer Systeme beschreibt. Dabei werden die entworfenen Lösungen schrittweise anhand der vorgegebenen Anforderungen abgesichert. Die Phasen bzw. Elemente dieses Modells sind:

Anforderungen

Die Anforderungen an das zu entwickelnde System sind der Ausgangspunkt der Entwicklung. Diese Anforderungen bilden gleichzeitig den Maßstab, nach welchem das spätere Produkt bewertet werden soll.

Systementwurf

In der Systementwurfsphase wird ein domänenübergreifendes, also von Maschinenbau, Elektrotechnik und Informationstechnik unabhängiges Lösungskonzept entworfen, das die wesentlichen Wirkungsweisen des zukünftigen Produktes beschreibt. Dazu wird das Gesamtsystem in wesentliche Teilfunktionen zerlegt, für welche wiederum Wirkprinzipien und Lösungselemente gesucht werden. Diese werden auf ihre Funktionserfüllung im Gesamtzusammenhang geprüft.

Domänenspezifischer Entwurf

Nun können, meist getrennt nach den jeweiligen Domänen, konkrete Lösungen entwickelt und weiter konkretisiert werden. Dazu werden detaillierte Berechnungen und Auslegungen durchgeführt, um die spätere Funktionserfüllung der Teilsysteme sicherzustellen.

Systemintegration

In dieser Phase wird das Gesamtsystem aus den Teillösungen zusammengesetzt, damit das Zusammenspiel untersucht werden kann.

Eigenschaftsabsicherung

Jeder Fortschritt beim Entwurf wird anhand des Lösungskonzepts und der gestellten Anforderungen überprüft, damit später die gewünschten mit den realen Systemeigenschaften übereinstimmen.

Modellbildung und -analyse

Begleitend werden die Systemeigenschaften mit Modellen und rechnerunterstützten Werkzeugen untersucht.

Produkt

Das Produkt ist das Ergebnis eines durchlaufenen Entwicklungszyklus nach dem V-Modell. Darunter können auch Zwischenentwicklungen verstanden werden, welche noch nicht marktreif sind und noch weitere Zyklen des V-Modells benötigen.

Auf der Mikroebene, also in den einzelnen Phasen des V-Modells, soll der allgemeine Problemlösungszyklus Verwendung finden, wie er aus der Systemtechnik bekannt ist (s. Abb. 4.2).

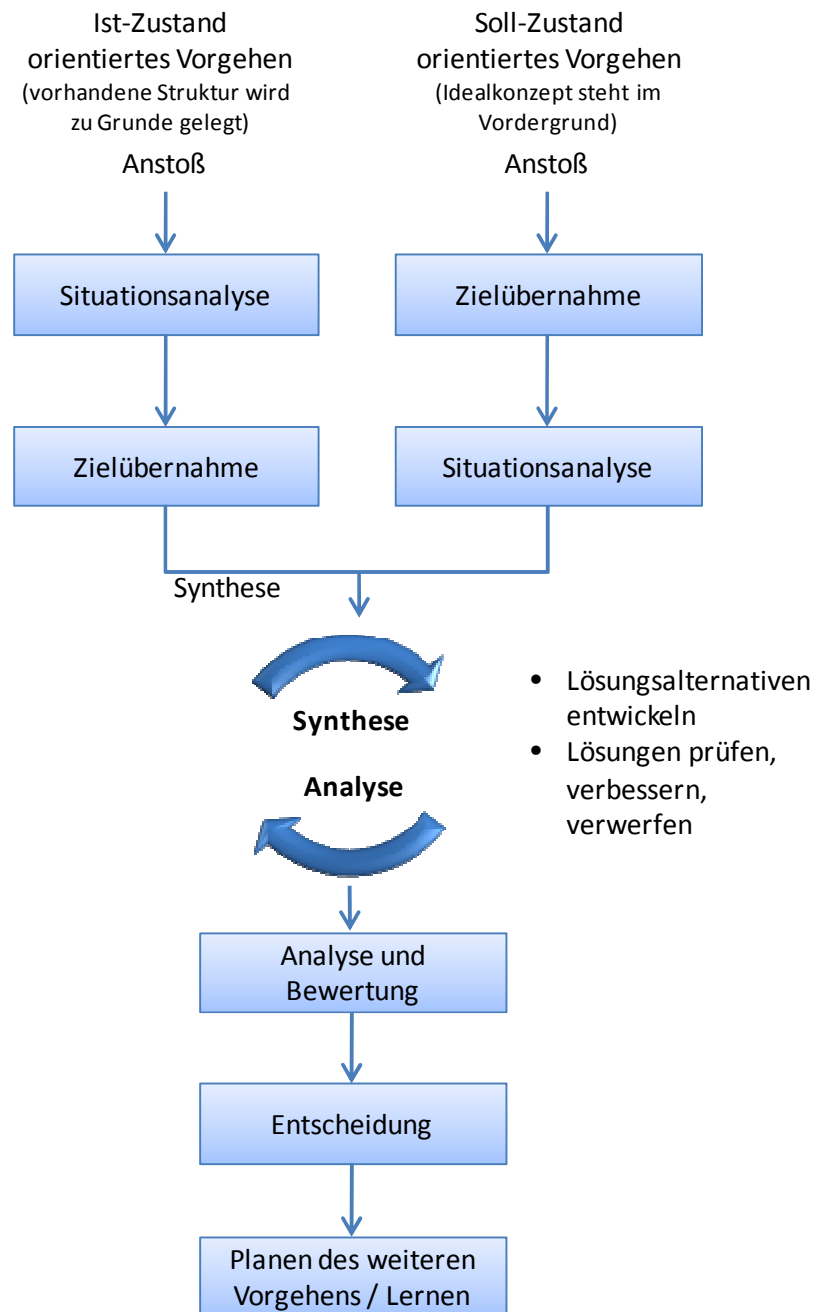


Abb. 4.2: Allgemeiner Problemlösungszyklus [Dae-94]

Für eine genauere Beschreibung sei auf die einschlägige Literatur der Systemtechnik verwiesen [Dae-94].

4.1.2 Methodik zur Modulentwicklung für das Internet der Dinge

Das im letzten Kapitel beschriebene V-Modell dient als generische Vorgehensweise zur Entwicklung mechatronischer Produkte. Es bietet sich an, diese Vorgehensweise auch für die Entwicklung von Modulen im Internet der Dinge anzuwenden. Dazu müssen die einzelnen Vorgehensschritte an die konkrete Aufgabe angepasst werden (s. Abb. 4.3)

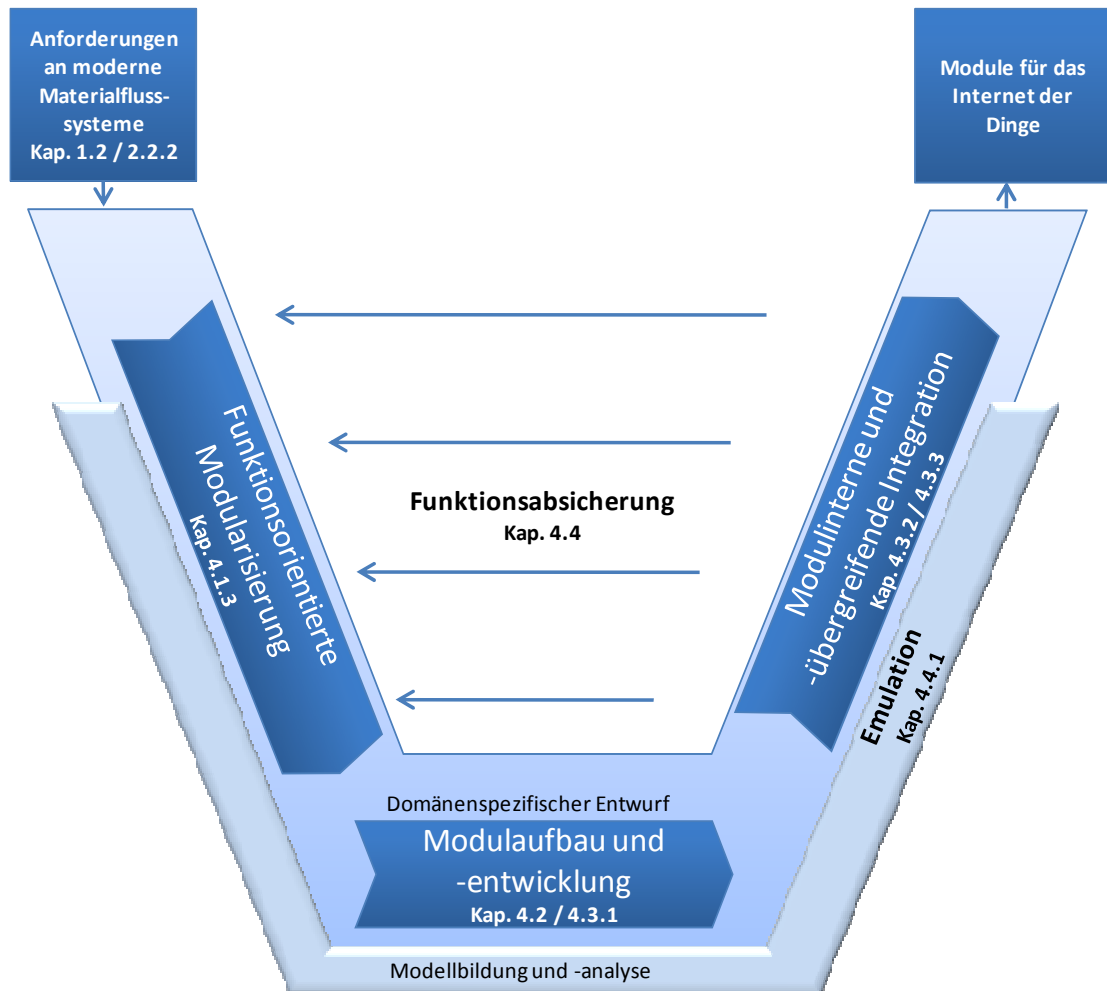


Abb. 4.3: Angepasstes V-Modell für die Entwicklung und Integration von Modulen für das Internet der Dinge

Funktionsorientierte Modularisierung

Ausgehend von den Anforderungen an moderne Materialflusssysteme, wie sie in Kap. 1.2 und Kap. 2.2.2 bereits formuliert wurden, wird eine funktionale Aufteilung eines Materialflusssystems vorgenommen. Diese Aufteilung wird nach den Regeln der funktionsorientierten Modularisierung (s. Kap. 4.1.3) durchgeführt. Dabei entste-

hen Module, welche über je eine energetische, informationstechnische und mechanische Schicht verfügen.

Modulaufbau und Entwicklung

Die im vorherigen Schritt entstandenen Module werden nun weiter konkretisiert und ihr grundsätzlicher Aufbau festgelegt. Die vorliegende Arbeit legt dabei einen Schwerpunkt auf die informationstechnische Gestaltung der Module, denn sowohl für die energetische als auch für die mechanische Schicht von funktionalen Modulen sind bereits praxistaugliche Lösungen vorhanden.

Diese informationstechnische Schicht umfasst sowohl die Steuerungssoftware, welche die Logik eines Moduls abbildet als auch die Steuerungshardware, auf welcher diese ausgeführt wird. Ein Modul ist aus verschiedenen steuerungstechnischen Ebenen aufgebaut. In Kap. 4.2.1 werden diese Ebenen sowie ihre Schnittstellen untereinander und zur Außenwelt definiert. Zudem werden zu verwendende Software und die notwendigen Hardwareplattformen vorgestellt.

Eine der wichtigsten Anforderungen bei der Entwicklung mechatronischer Module im Internet der Dinge ist deren Wiederverwendbarkeit für möglichst viele Materialflussanlagen. Daher wird ein Wiederverwendungskonzept entwickelt, das an die objektorientierte Softwareentwicklung angelehnt ist (s. Kap. 4.3.1).

Modulinterne und -übergreifende Integration

Nachdem nun der grundsätzliche Aufbau eines Moduls fest steht, müssen die Module sowohl intern als auch in ihrem Zusammenspiel integriert werden.

Ein wichtiger Aspekt beim Aufbau eines dezentral gesteuerten Materialflusssystems ist die Realisierung von Funktionen, welche nicht von einem Modul alleine zur Verfügung gestellt werden können oder die Abstimmung mehrerer Module untereinander benötigen. Damit diese Funktionen dennoch realisiert werden können, sind Regeln zu entwickeln, nach welchen sich die Einzelnen zu verhalten haben, damit ein gerichtetes Zusammenspiel oder auch spezielle Materialflusstrategien realisierbar werden. Die beispielhafte Entwicklung einer Strategie für eine gemeinsame Ressourcennutzung ist in Kap. 4.3.2 beschrieben.

Die interne Integration umfasst eine Prüfung der Schnittstellen zwischen den Domänen Elektrotechnik, Informationstechnik und Maschinenbau. Zudem müssen speziell

auf der steuerungstechnischen Ebene die einzelnen Schichten sowohl getrennt voneinander als auch in ihrer gemeinsamen Funktionalität getestet werden. Dazu sind Hilfsmittel für die Fehlersuche und Fehlerbehebung notwendig, wie sie in Kap. 4.4.3 vorgestellt werden.

Emulation

Für die Überprüfung eines modulübergreifenden Zusammenspiels wird die Emulation der realen Modulphysik, wie sie in Kap. 4.4.1 dargestellt wird, empfohlen.

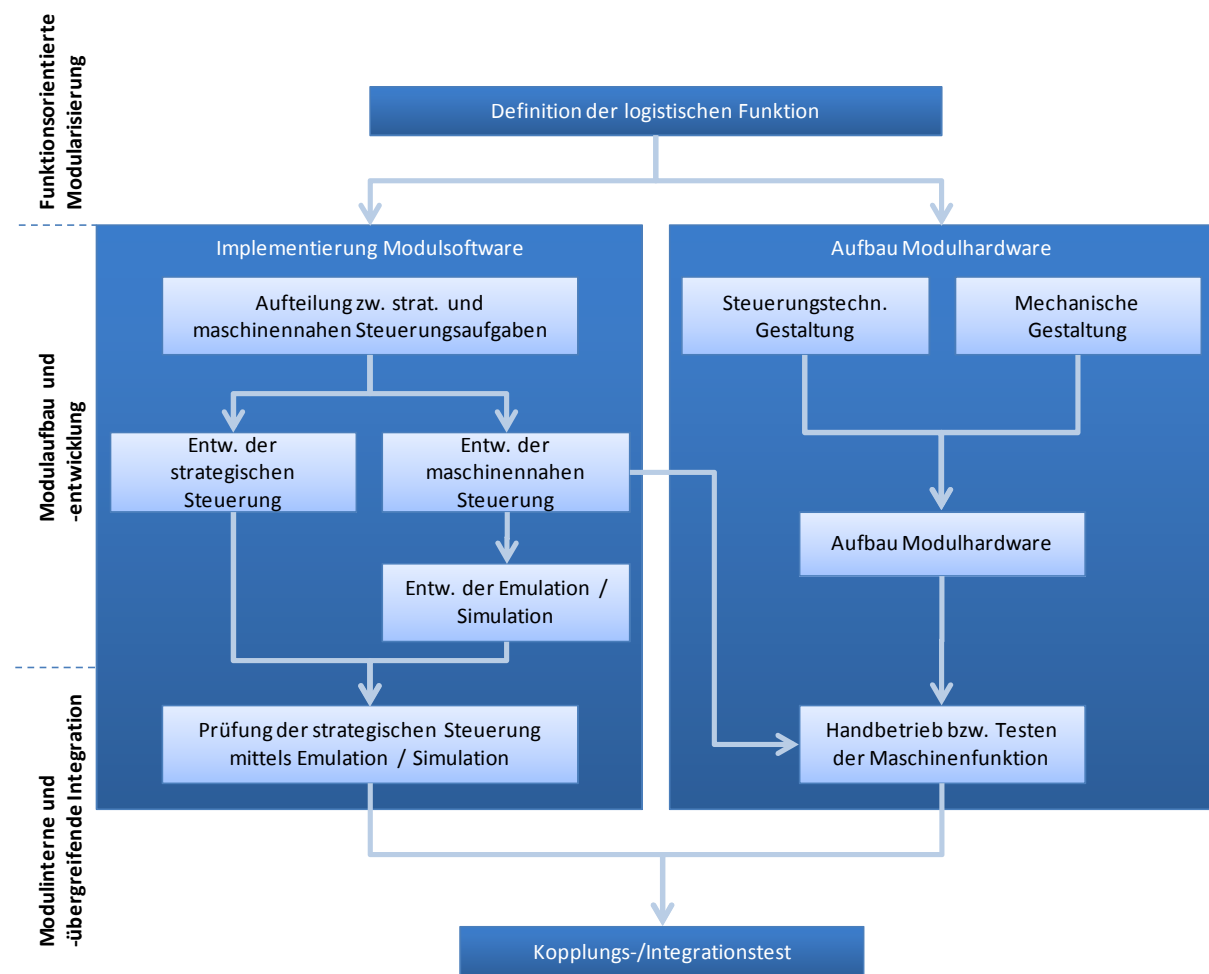


Abb. 4.4: Vorgehensweise bei der Modulentwicklung

In Abb. 4.4 sind die bei der Modulentwicklung ablaufenden Tätigkeiten in ihrer Reihenfolge bzw. ihren Abhängigkeiten zueinander zu sehen. Dabei zeigt sich, dass durch eine klare Gliederung der Steuerungssoftware und den Einsatz von Simulationstools eine Parallelisierung von Tätigkeiten möglich wird. Die Modulhardware kann unabhängig von der Modulsoftware aufgebaut werden. Zudem stellt die maschinen-

nahe Steuerung bereits einen Handbetrieb des Moduls zur Verfügung und ermöglicht damit einen von der strategischen Steuerung unabhängigen Test. Die Emulation wiederum erlaubt eine Prüfung der strategischen Steuerung, so dass hierfür keine Maschinenhardware verfügbar sein muss.

Es ist davon auszugehen, dass diese stark parallelisierte Vorgehensweise den Zeitaufwand bei der Realisierung- und Inbetriebnahme eines Materialflusssystems senken kann.

4.1.3 Funktionsorientierte Modularisierung

Um die Idee des Internet der Dinge zu verwirklichen, müssen die in herkömmlichen Materialflusssystemen gezogenen Systemgrenzen neu definiert werden. Der Modularisierung des Systems kommt damit eine zentrale Bedeutung zu. Im Folgenden wird das Thema Modularisierung im Allgemeinen dargestellt und, darauf aufbauend, die funktionsorientierte Modularisierung beschrieben.

Modularisierung

Die Planung von Materialflusssystemen berücksichtigt eine Vielzahl von Systemelementen und Abhängigkeiten unter diesen. Daher bietet es sich an, den Entwicklungsprozess in mehrere parallel ablaufende Prozesse zu unterteilen, die für die Planer noch handhabbar sind (s. Abb. 4.5)

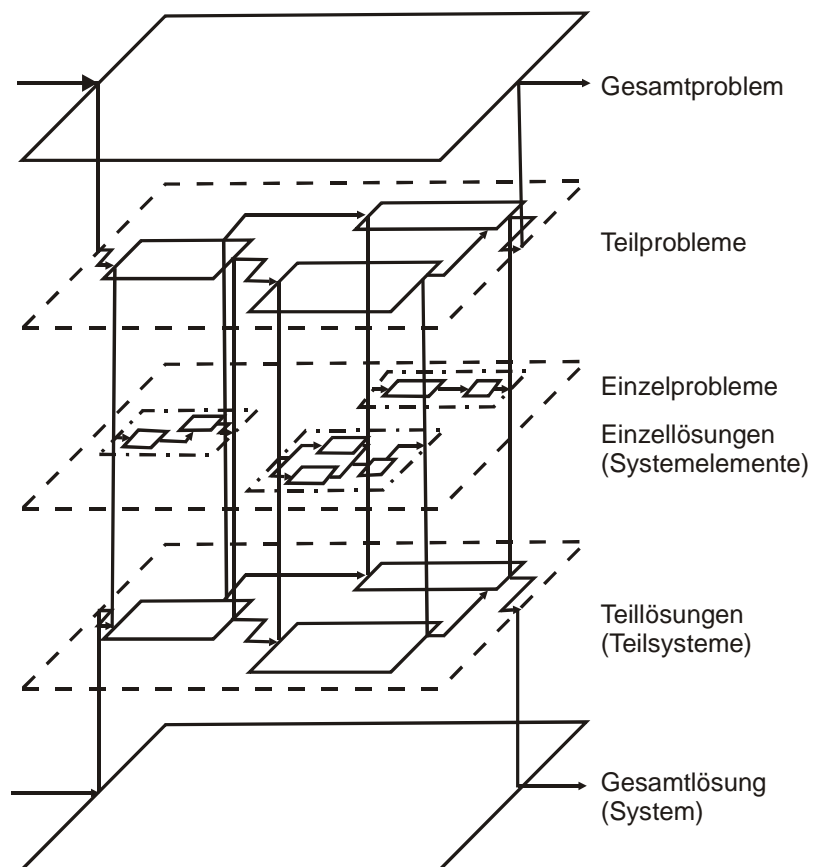


Abb. 4.5: Methode der Aufgliederung und Verknüpfung zur Problem- und Systemstrukturierung [VDI-2221]

Bei dieser Vorgehensweise wird ein Gesamtproblem in Teilprobleme gegliedert, die wiederum jeweils einzeln gelöst werden können. Diese Teillösungen werden anschließend kombiniert. Betrachtet man nun mehrere Planungsprozesse, so fällt auf, dass viele Einzellösungen an mehreren Orten bzw. in unterschiedlichen Planungsprojekten Verwendung finden können. Diese Wiederverwendbarkeit von Teillösungen legt die Verwendung eines Baukastens nahe, in dem häufig wiederkehrende Systemelemente hinterlegt sind, die bei Bedarf mit neuen Lösungen kombiniert werden können. Voraussetzung dafür ist, dass die einzelnen Systemelemente über standardisierte Schnittstellen verfügen, die eine problemlose Kombinierbarkeit sicherstellen. Ist dieses Baukastensystem aus standardisierten Einzelkomponenten aufgebaut, die auch gegeneinander ausgetauscht werden können, so kann man von einem modularisierten System sprechen. Die allgemeinen Eigenschaften und Beschreibungsmerkmale von Modulen nach Hildebrand sind in Abb. 4.6 dargestellt.

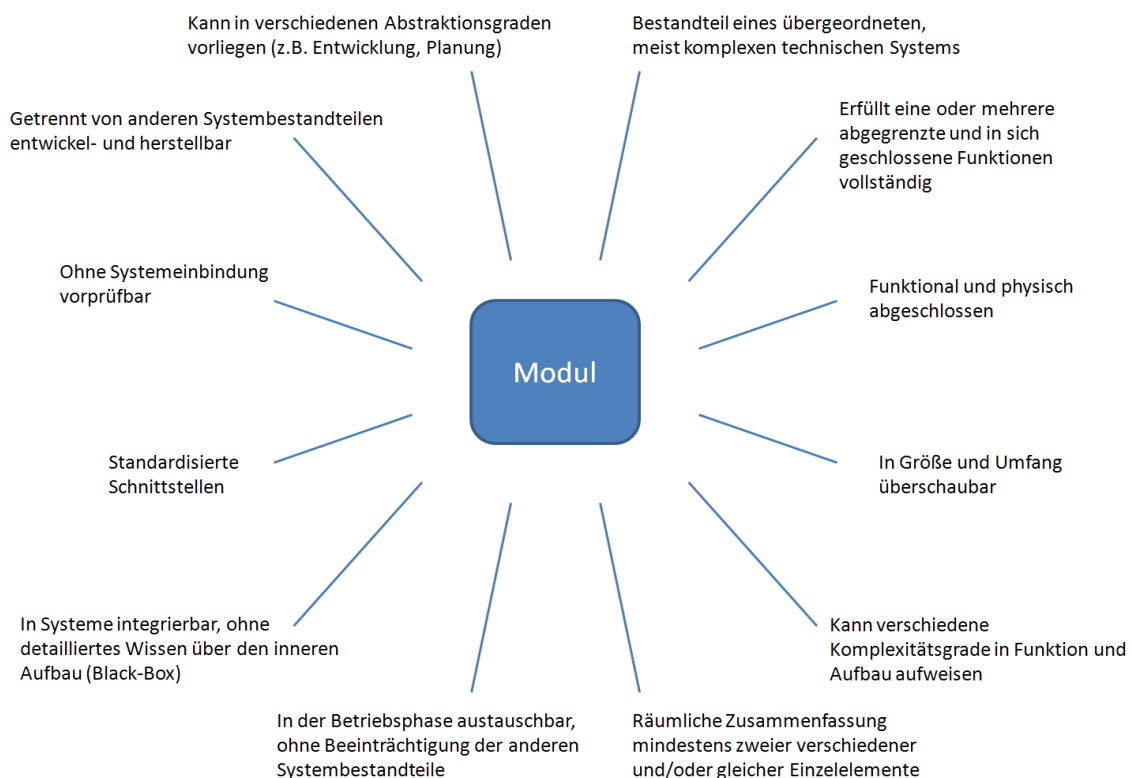


Abb. 4.6: Allgemeine Eigenschaften und Beschreibungsmerkmale eines Moduls [Hil-05]

Der Gedanke der Modularisierung und der Einsatz von Baukastensystemen, vor allem bezüglich der mechanischen Gestaltung von Fördertechnik, wird von Herstellern bereits seit Langem verfolgt – denn die Vorteile, ein komplexes Logistiksystem aus einer überschaubaren Anzahl standardisierter Module wie Weichen, Rollenförde-

ern und Drehtischen flexibel realisieren zu können, liegen auf der Hand: die hohen Stückzahlen der Module bringen Kostenvorteile und die standardisierten Schnittstellen erleichtern die Integration [Gün-06].

So werden zeit- und personalaufwändige Arbeiten, wie z.B. Entwicklung und Inbetriebnahme von abgeschlossenen Modulen in eine projektunabhängige Vorbereitungsphase verschoben. Dies kann den Aufwand bei späteren Projekten drastisch senken (s. Abb. 1.1).

Ein wichtiger Aspekt bei dieser dezentralen, aus eigenständigen Modulen bestehenden Systemarchitektur ist die Granularität der Modularisierung. Die Systemgrenzen der Module bestimmen über die Wiederverwendbarkeit von Lösungen für unterschiedliche Anwendungen. Prinzipiell gilt: Je kleiner und je universeller die Bausteine eines Systems, desto mehr Kombinationsmöglichkeiten gibt es. Man ist bei der Planung freier und die Wiederverwendbarkeit der Bausteine verbessert sich. Werden jedoch die Module zu klein oder in ihrer Ausprägung zu spezialisiert, dann vermehren sich auch die Schnittstellen und das Kommunikationsaufkommen steigt stark an. Daher soll im Folgenden genauer betrachtet werden, nach welchen Regeln die Systemgrenzen der Module in einem Materialflusssystem sinnvollerweise gezogen werden sollten, um eine hohe Wiederverwendbarkeit zu sichern, ohne dabei eine zu hohe Kommunikationslast und Schnittstellenkomplexität zu verursachen.

Prinzipiell bieten sich zur Modularisierung in einem Materialflusssystem verschiedene Ebenen an, welche wiederum über die Zuständigkeiten eines Moduls und seine Größe entscheiden. In Abb. 4.7 sind ausgewählte Aufgaben einer Materialflussteuerung zu sehen, welche in einem herkömmlichen, zentral gesteuerten System auf unterschiedlichen Ebenen der Automatisierungspyramide angeordnet sind (siehe Kap. 2.1.1). Dies sind:

- **Geschäftslogik**
Kunden- oder projektspezifische Abläufe, z.B. Stationen, welche in einer bestimmten Reihenfolge von einer Transporteinheit passiert werden müssen und Restriktionen, die dabei einzuhalten sind.
- **Materialflusstrategien**
Vorschriften und Regeln, nach welchen die Transporteinheiten ihren Weg nehmen sollen oder die Fördertechnik ihre Transportaufträge abarbeiten soll.

- **Logistische Funktionen**
Grundfunktionen der physischen Logistik sind laut Günthner [Gün-08b] Fördern / Transportieren, Verteilen / Zusammenführen, Lagern und Handhaben.
- **Technische Funktionen**
Hierbei handelt es sich um Funktionen, welche notwendig sind, um mechanische Vorgänge anzusteuern und zu überwachen, und die sich direkt auf die Umwelt auswirken oder Veränderungen in dieser registrieren. Dies ist z.B. die Regelung eines Antriebsmotors oder die Meldung über die Ankunft einer Transporteinheit.

Eine Modularisierung auf der Ebene der technischen Funktionen führt zu relativ kleinen und damit sehr zahlreichen Modulen. Dies sind z.B. einzelne Sensoren und Aktoren wie Lichtschranken oder Antriebsmotoren. Auf dieser Ebene sind die Module sehr stark von der eingesetzten Technologie abhängig. Im schlechtesten Fall muss für jedes Fabrikat eines Sensors oder Aktors ein neuer Modultyp generiert werden. Auf diese Weise sinken sowohl die verbaute Stückzahl pro Modultyp als auch die Wiederverwendbarkeit eines Moduls.



Abb. 4.7: Geeignete Modularisierungsebene

Alternativ kann eine Modularisierungsebene gewählt werden, welche einen hohen Anteil an dispositiven Aufgaben wie z.B. besondere Materialflussstrategien oder die Bearbeitungsschritte und Randbedingungen für die zu transportierenden Waren (Geschäftslogik) enthält. Diese Wahl führt aber dazu, dass die Module relativ groß

ausfallen und ihre Übertragbarkeit von einem realisierten Projekt zum nächsten nur gering ist, da gerade diese Aufgaben in ihrer Ausprägung sehr projektspezifisch sind.

In der Konstruktionslehre werden bei der Modularisierung von Produkten Lösungsalternativen abgeleitet, indem die zu erfüllenden Funktionen analysiert werden [VDI-2221]. Im Bezug auf Materialflusssysteme verspricht eine Modularisierung auf der Ebene der logistischen Funktion praxistaugliche Lösungen. Einerseits werden damit Modulgrenzen gezogen, welche meist schon heute für die Gestaltung von mechanischen Fördertechnikbaukästen festgelegt werden. Andererseits bewegt sich damit die Modularisierung auf einer Ebene, welche eine hohe Übertragbarkeit der Module auf verschiedenste Projekte ermöglicht, da beinahe jeder Materialfluss als eine sequenzielle und/oder parallele Aneinanderreihung dieser logistischen Grundfunktionen dargestellt werden kann.

Funktionsorientierte Modularisierung

Wilke [Wil-06] konkretisiert die beschriebene Modularisierung unter dem Namen „funktionsorientierte Modularisierung“ und entwickelt dazu Regeln für die Gestaltung von Materialflussmodulen. Dabei wird das Materialflusssystem aus mechatronischen Modulen aufgebaut, deren Modulgrenzen entsprechend der logistischen Funktionalität gezogen werden, und die nach dem Konzept der verteilten Automatisierung automatisiert sind.

Bei der verteilten Automatisierung sind alle Systemelemente autonome Module. Sie verfügen über eigene Mechanik sowie energetische und steuerungstechnische Komponenten.

Kriterien für eine funktionsorientierte Modularisierung nach Wilke sind:

- Zerlegbarkeit
Ein System lässt sich in Subsysteme/Module zerlegen.
- Kombinierbarkeit
Module lassen sich frei und unabhängig voneinander zu neuen Systemen und Subsystemen kombinieren.

- **Verständlichkeit**
Die Funktion eines Moduls muss für einen Anwender (Bediener/Planer) verständlich sein. Die Verwendung eines Moduls erfordert keine Kenntnisse über seinen inneren Aufbau (Black-Box-Prinzip)
- **Stetigkeit und Geschützttheit (Kapselung)**
Änderungen im Inneren eines Moduls, welche seine Schnittstellen unverändert lassen, dürfen keine Rückwirkungen auf das übrige System haben. Fehler und Störungen in einem Modul sollten auf das Modul beschränkt bleiben.
- **Identität der Systemgrenzen**
Die Systemgrenzen von Mechanik, Energie und Steuerungstechnik eines Moduls müssen identisch sein.
- **Funktionsorientierte Betrachtungsweise**
Modulgrenzen werden entsprechend der Funktionalität des Moduls gezogen. Das Modul kann seine Funktion ohne Einbettung in ein Gesamtsystem erfüllen. Es lässt sich unabhängig von anderen Modulen testen und in Betrieb nehmen.

In der vorliegenden Arbeit wird die funktionsorientierte Modularisierung als Gestaltungsmethodik für Materialflusssysteme nach dem Konzept des Internet der Dinge gewählt.

4.1.4 Funktionsklassen für Module

Die Module eines Internet der Dinge-Materialflusssysteme lassen sich sehr gut strukturieren, indem man eine Gruppe von Modulen, welche ähnliche logistische Funktionen zu erbringen haben, zu einer Klasse zusammenfasst. Die Bildung von Klassen erleichtert dem Entwickler die Strukturierung seiner Aufgabe und erhöht die Wiederverwendbarkeit von Lösungen.

Kein Differenzierungsmerkmal sind dabei Funktionen, welche ausnahmslos von allen Modulen erbracht werden müssen. Alle Module müssen beispielsweise ihre eigenen Fähigkeiten und Funktionen bekannt geben und den eigenen Zustand verwalten können. Dies betrifft sowohl Belegungs- als auch Fehlerzustände und das Absenden von Rück- und Statusmeldungen. Zudem soll jedes Modul in der Lage sein, einen

Lastwechsel mit einem anderen Modul abzustimmen, um eine kollisionsfreie Übergabe von Transporteinheiten zu gewährleisten.

Sieht man von diesen Grundfunktionen ab, so ergeben sich Klassen, welche eine ähnliche Funktionalität bereitstellen müssen. Dazu ist zu bemerken, dass es dem jeweiligen Entwickler frei steht, die hier aufgeführten Funktionen noch zu erweitern, um Module an die eigenen Bedürfnisse anzupassen. Die Funktionsklassen für die Module des Internet der Dinge sind folgende:

Unstetigförderer

Unstetigförderer sind flurgebundene, aufgeständerte oder flurfreie Fördereinrichtungen, bei denen das Fördergut von Aufgabe- zu Abgabestelle in einzelnen Arbeitspielen bewegt wird. Die Be- und Entladung erfolgt dabei während des Stillstandes des Fördermittels [FML-09].

Alle Unstetigförderer stellen mindestens folgende Funktionen zur Verfügung:

- Auftragsdisposition
- Ausführen der Transportaufträge
- Kollisionskontrolle
- Verwaltung der eigenen Plätze
- Ermittlung aller für die Modulfunktion relevanten Eigenschaften einer Transporteinheit (entweder durch Anfrage bei anderen Entitäten oder durch Identifikation direkt auf dem Förderelement)

Wichtige Zusatzfähigkeiten von Unstetigförderern sind die Abstimmung mit anderen Modulen, um die gemeinsame Nutzung von Ressourcen zu koordinieren und die Wegplanung im eigenen Aktionsbereich.

Beispiele für Unstetigförderer sind Katzen einer Elektrohängebahnanlage, Stapler mit Leitsystem, Regalbediengeräte oder Fahrerlose Transportfahrzeuge (FTF).

Stetigförderer und Schienen

Stetigförderer sind mechanische, pneumatische und hydraulische Fördereinrichtungen, bei denen das Fördergut auf festgelegtem Förderweg von Aufgabe- zu Abgabestelle stetig, mit wechselnder Geschwindigkeit oder im Takt bewegt wird [DIN-15201]. Schienen dienen dem Tragen und der Führung eines Laufwerks [VDI-3647]. Dies kann z.B. ein Unstetigförderer sein. Ebenso wie ein Stetigförderer das Beförde-

rungsmittel für eine TE ist, transportiert eine Schiene einen Unstetigförderer. Daher erfüllt eine Schiene aus der Perspektive des Internet der Dinge eine ähnliche Funktion wie die Stetigförderer und die beiden Elemente können einer einzigen Kategorie zugeordnet werden. Alle Stetigförderer und Schienen stellen folgende Funktion zur Verfügung:

- Auftragsdisposition
- Transport von Transporteinheiten bzw. Führen von Unstetigförderern von der Aufgabe- zur Abgabestelle des Moduls

Beispiele für Stetigförderer sind Senkrechtförderer, Gurtförderer, Rollenbahnen oder Elektrohängebahnstrecken.

Verzweigung / Zusammenführung

Verzweigungs- und Zusammenführungsstellen leiten Transporteinheiten von einer Strecke auf eine andere um [VDI-3647]. Sie stellen dabei folgende Funktionen zur Verfügung:

- Herstellen von unterschiedlichen Quelle-Ziel Verbindungen (z.B. Schalten von Streckenverbindungen)
- Prioritätenregelung und Kollisionsschutz (bei Zusammenführungen)
- Verwalten / Bereitstellen eines Aufsetzpunktes / Übergabeplatzes

Dabei wird zwischen zwei Varianten unterschieden. Einerseits gibt es Verzweigungen/Zusammenführungen, die lediglich eine Streckenverbindung herstellen, welche dann von einem Unstetigförderer befahren wird. Dies sind z.B. Weichen in einem Elektrohängebahnsystem. Hier reichen die bereits erwähnten Funktionen aus, da die Transport bzw. Koordinationsaufgabe von dem Unstetigförderer übernommen wird. Andererseits werden Verzweigungen/Zusammenführungen eingesetzt, welche die Transporteinheiten in einem stetigen Fördergutstrom selbst befördern. Dies sind z.B. Ein-/Ausschleuser auf einem Rollenförderer. Eine solche Verzweigung/Zusammenführung sollte zusätzlich folgende Funktionen erfüllen:

- Ermittlung aller für die Modulfunktion relevanten Eigenschaften einer Transporteinheit (entweder durch Anfrage bei anderen Entitäten oder durch Identifikation direkt auf dem Verzweigungs-/Zusammenführungselement)
- Ausführen der Transportaufträge

Beispiele für diese Sorte von Verzweigungs-/Zusammenführungselementen sind z.B. Rollenbahnweichen, Querverschiebewagen oder Drehtische.

Arbeitsstation

Arbeitsstationen dienen der Veränderung oder Bestimmung von Eigenschaften der Transporteinheiten. Dies kann eine Veränderung des Inhalts einer TE während der Kommissionierung oder die Identifikation einer TE mittels eines RFID-Lesegerätes sein. Arbeitsstationen stellen folgende Funktionen zur Verfügung:

- Auftragsdisposition
- Meldungen über Eigenschaften der TE
- Einlasten von TE in den Materialfluss (über einen Übergabepplatz)

Darüber hinaus können beispielsweise folgende Funktionen notwendig sein:

- Verwaltung der gerade verfügbaren TE bzw. Artikel
- Anzeige von Auftragsinformationen für einen Bediener
- Meldungen über Fehlmengen bei der Kommissionierung

Beispiele für Arbeitsstationen sind Kommissionierarbeitsplätze oder Produktionsmaschinen.

4.2 Grundsätzlicher Aufbau von Modulen

Der interne Aufbau eines Moduls lässt sich in zwei Ebenen untergliedern (siehe Abb. 4.8). Zum einen die mechanische Ebene, welche die mechanische Konstruktion und alle Sensoren und Aktoren umfasst. Dazu gehören beispielsweise tragende Teile, Getriebe, Motoren und Lichtschranken.

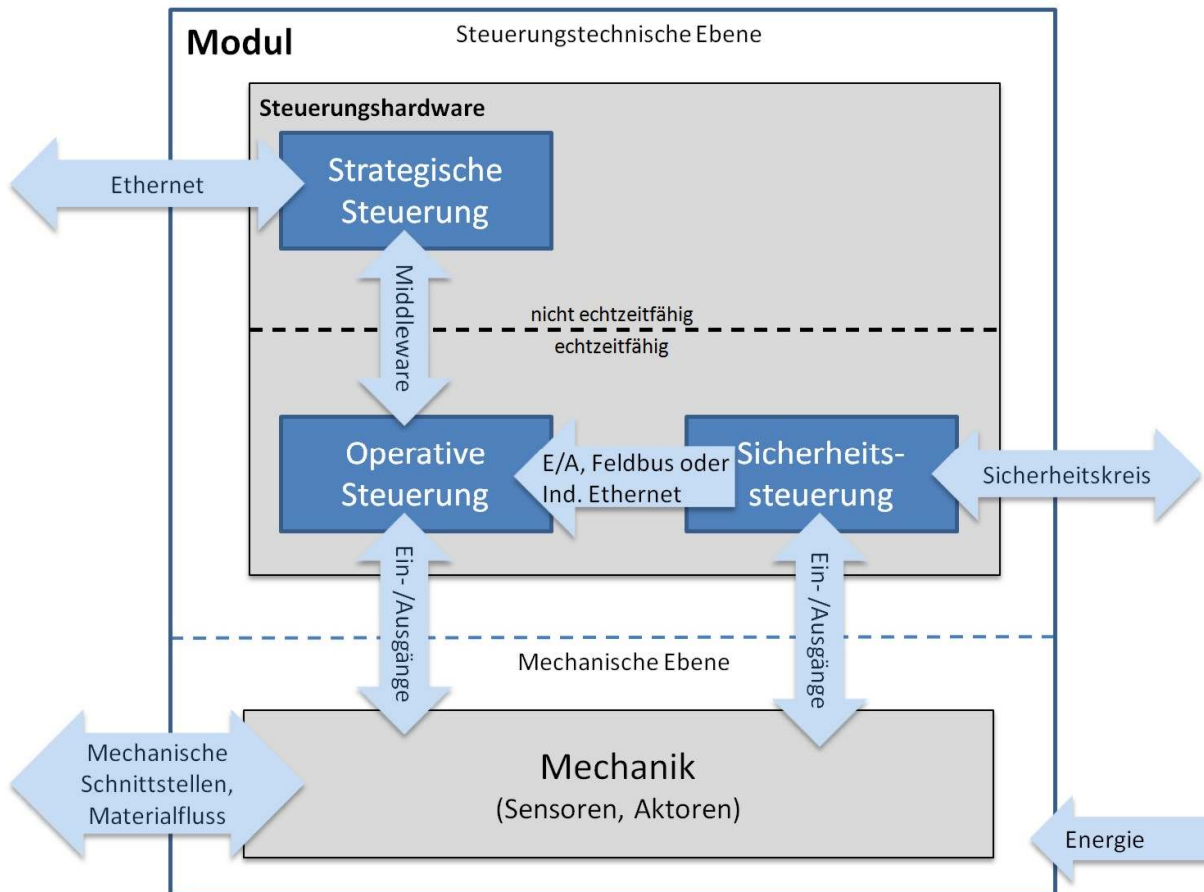


Abb. 4.8: Modulaufbau und Schnittstellen

Zum anderen sind auf der steuerungstechnischen Ebene alle Komponenten anzusiedeln, welche die Sensoren überwachen, mit der Umwelt kommunizieren und Aktoren in Bewegung versetzen. Die Steuerungshardware dient als Plattform zur Ausführung der Steuerungsprogramme und stellt Schnittstellen sowohl zu den Sensoren und Aktoren der mechanischen Ebene als auch zu anderen Modulen zur Verfügung (siehe Kap. 4.2.1).

Die auf der Steuerungshardware implementierte Logik kann je nach auszuführender Steuerungsaufgabe in die Sicherheitssteuerung, die operative und die strategische Steuerung unterteilt werden:

Sicherheitssteuerung

Bei der Sicherheitssteuerung handelt es sich um eine Komponente, welche unabhängig von der restlichen Anlagensteuerung dafür zuständig ist, gefährliche Zustände zu erkennen, notwendige Maßnahmen einzuleiten und diese Information an andere Sicherheitssteuerungen weiterzugeben. Typische sicherheitsrelevante Ereignisse sind z.B. die Betätigung eines Notausschalters oder das unbefugte Betreten eines Gefahrenbereichs. Üblicherweise wird für die Übertragung dieser Information ein sogenannter Sicherheitskreis verlegt. Eine abfallende Potenzialflanke zeigt den Notaus an und führt dazu, dass die Geräte innerhalb eines betroffenen Bereichs abgeschaltet bzw. in einen sicheren Zustand versetzt werden.

Die IEC 61501 „Standard for Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems“ gilt weltweit als Basis für Spezifikationen, Entwurf und Betrieb von sicherheitstechnischen Systemen (Safety Instrumented Systems, SIS). Anhand einer Gefährdungs- und Risikoanalyse werden dabei sämtliche Gefahren ermittelt, die von einer Anlage bzw. einem Steuerungssystem ausgehen. Dadurch wird analysiert, ob funktionale Sicherheit erforderlich ist, um einen angemessenen Schutz gegen mögliche Gefährdungen zu gewährleisten. Je nach Steuerungsaufgabe müssen also spezielle Geräte eingesetzt werden, welche über eine entsprechende Schutzklasse nach IEC 61501 verfügen.

Operative Steuerung

Die operative Steuerung dient der Überwachung von Sensoren und dem Ansteuern von Aktoren. Diese Steuerungsaufgabe wird abgearbeitet, ohne dass eine Kommunikation mit anderen Modulen notwendig ist. Besonders wichtig ist hier die schnelle und zuverlässige Erledigung der gestellten Aufgabe. Die Aufgaben der maschinennahen Steuerung sind sehr eng an die konkrete technologische bzw. mechanische Gestaltung eines Moduls gebunden. So ist es beispielsweise wahrscheinlich, dass Rollenförderer verschiedener Hersteller auch über unterschiedliche Aktoren bzw. Sensoren verfügen. Die Anbindung dieser Komponenten an die maschinennahe Steuerung wird dabei herstellerspezifisch ausfallen. Ebenso muss das Steuerungsprogramm speziell auf die Positionen der Sensoren und die Geschwindigkeiten der Aktoren abgestimmt werden. Die Positionsanfahrt eines Regalbediengerätes ist ein

Beispiel dafür: Die Steuerung setzt den Motor in Bewegung, überwacht über entsprechende Sensorik die aktuelle Position und regelt nach einem vorgegebenen Algorithmus das Antriebsmoment bzw. die Drehzahl während der Fahrt nach. Dabei muss sichergestellt sein, dass auf eine Veränderung des Ist-Zustands innerhalb einer garantierten Zeitspanne von z.B. maximal 10ms reagiert werden kann. Die Aufgaben der operativen Steuerung erfordern also eine Abarbeitung in Echtzeit (siehe dazu Kap. 2.3.1).

Da sowohl die Sicherheitssteuerung als auch die operative Steuerung direkt mit den Sensoren und Aktoren der Mechanik verbunden sind, sollen diese im Folgenden als maschinennahe Steuerung bezeichnet werden.

Strategische Steuerung

Aufgaben, welche in der klassischen Automatisierungspyramide (siehe Kap. 1.2) in den oberen Ebenen angesiedelt sind und bisher von einem Materialflussrechner übernommen wurden, wandern nach den Vorgaben der funktionsorientierten Modularisierung ebenfalls in den Zuständigkeitsbereich eines Moduls. Dies sind z.B. die dynamische Routenplanung für Transporte, die Auftragsdisposition oder die Ressourcenallokation. Hierzu muss in der Regel mit anderen Modulen kommuniziert werden, um Abstimmungsmechanismen bzw. Auktionen zu realisieren.

Diese Aufgaben sind zwar nicht sicherheits- oder echtzeitrelevant, sollten jedoch so schnell abgearbeitet werden, dass keine Stauungen bzw. unnötige Wartezeiten im Materialfluss entstehen.

Die Steuerungslogik der strategischen Steuerung lässt sich im Unterschied zur maschinennahen Steuerung sogar auf Förderer ähnlichen Typs (z.B. Rollenförderer und Bandförderer) übertragen. Eine klare Trennung zwischen strategischer und maschinennaher Steuerung ist also wünschenswert und würde erlauben, das gleiche strategische Steuerungsprogramm für verschiedene Technologieausprägungen zu verwenden.

4.2.1 Schnittstellen eines Moduls

Die Schnittstellen eines Moduls dienen der Interaktion mit der Umwelt z.B. durch die Übertragung von Energie und Informationen, der Aufnahme bzw. Abgabe von Transporteinheiten und der mechanischen Verbindung.

4.2.1.1 Externe Modulschnittstellen

Die externen Modulschnittstellen lassen sich in mechanische, energetische und steuerungstechnische Schnittstellen aufteilen.

Mechanische und energetische Modulschnittstellen

Die mechanischen Schnittstellen eines Moduls umfassen alle festen Verbindungen zur umgebenden Fördertechnik oder dem Fabrikgebäude. Auch die Weitergabe einer Transporteinheit von einem Modul zum nächsten wird durch eine mechanische Schnittstelle sichergestellt. Die energetischen Schnittstellen eines Moduls dienen der Zuführung von elektrischer Energie, Druckluft oder Flüssigkeiten (Hydraulik).

Sowohl für mechanische als auch für die energetischen Schnittstellen existieren bzgl. ihrer Realisierung bereits vielfältige praxistaugliche Lösungen, welche auch die Grundlage für die bisherigen Baukastensysteme von Fördertechnikherstellern bilden. Aus diesem Grund wird in der vorliegenden Arbeit nicht genauer darauf eingegangen.

Sicherheitssteuerung

Auch die Sicherheitssteuerung eines Moduls stellt Schnittstellen zur Außenwelt bereit. Der sogenannte Sicherheitskreis dient dazu, besondere Gefahrensituationen oder Systemzustände zu übermitteln, welche nach IEC 61501 eine besondere Schutzklasse erfordern. Die klassische Lösung wäre hier eine einfache elektrische Verbindung zwischen den einzelnen Sicherheitskomponenten. Inzwischen sind aber auch herstellereigenspezifische Industrial-Ethernet-Lösungen (s. Kap. 2.3.3) verfügbar, die diese Anforderungen erfüllen und eine komfortable Verschaltung der Sicherheitstechnik ermöglichen. Ein Beispiel dafür ist das Safety-Busklemmen-Konzept der Firma Beckhoff, das über ein auf Ethernet basierendes Busprotokoll kommuniziert und die Programmierung einfacher Regeln zur Signalverarbeitung, wie z.B. UND- bzw. ODER-Schaltungen erlaubt.

Operative Steuerung

Die operative Steuerung kommuniziert lediglich mit der Sicherheitssteuerung und der strategischen Steuerung und stellt keine externen Schnittstellen bereit.

Strategische Steuerung

Die strategische Steuerung benötigt eine Schnittstelle zu anderen Entitäten. Diese Schnittstelle dient u. A. der Übertragung aller Inhalte, welche bisher laut VDMA 15276 zwischen Ebene 6 „Darstellung und Kommunikation“ und Ebene 5 „Systemsteuerung“ ausgetauscht wurden. Die dort aufgeführten Inhalte lassen sich ebenso auf das hier dargestellte dezentrale Steuerungsmodell übertragen. Dabei handelt es sich z.B. um Transportaufträge, Vollzugs- und Statusmeldungen und Daten für Programmupdates.

Diese relativ komplexen Dateninhalte legen nahe, eine Schnittstelle zu wählen, welche eine gute Strukturierung der Daten bei der Übertragung erlaubt und auch größere Datenmengen problemlos übermitteln kann. Ethernet verfügt über diese Eigenschaften und soll daher als Schnittstelle dienen.

Wilke [Wil-06] schlägt für die Datenübertragung eine ethernetbasierte XML-Kommunikation vor und beschreibt diese Schnittstelle eines autonomen Moduls ausführlich. Diese Grundlagen sind daher vorhanden und nicht Gegenstand der vorliegenden Arbeit.

4.2.1.2 Interne Modulschnittstellen

Ein Modul verfügt über diverse interne Schnittstellen, welche im Folgenden näher erläutert werden.

Maschinennahe Steuerung - Sensoren / Aktoren

Die Hardwarekomponenten für die maschinennahe Steuerung verfügen entweder selbst über digitale und / oder analoge Ein- und Ausgänge oder bieten die Möglichkeit, Busklemmen anzuschließen, welche E/As bereitstellen.

Dabei sind alle für die ordnungsgemäße Abarbeitung der Steuerungsaufgabe notwendigen Sensoren und Aktoren an die operative Steuerung angeschlossen, während Sensoren, welche einen Gefahrenzustand oder eine sicherheitsrelevante Fehlfunktion detektieren, mit der Sicherheitssteuerung verbunden sind. Erkennt die Sicherheitssteuerung einen Gefahrenzustand, so leitet sie selbstständig adäquate

Schritte ein, indem sie beispielsweise die Stromversorgung der Aktoren unterbricht und damit die Antriebe des Moduls still legt. Gleichzeitig werden Informationen über diese Maßnahmen an die operative Steuerung weitergegeben. Da diese Informationen in Echtzeit übertragen werden müssen, müssen für diese Datenverbindung entweder klassische Feldbusse oder Industrial Ethernet Lösungen eingesetzt werden. Die operative Steuerung wiederum kann nun ihre eigenen Abläufe unterbrechen und das Modul in einen Zustand versetzen, der einen schnellen Wiederanlauf ermöglicht, nachdem die Störung behoben wurde, und die Sicherheitssteuerung wieder einen ordnungsgemäßen Zustand meldet.

Operative Steuerung - Strategische Steuerung

Jegliche Störungszustände aber auch Meldungen über erledigte Steuerungsaufgaben werden von der operativen Steuerung an die strategische Steuerung gemeldet. Die Aufgaben der strategischen Steuerung unterliegen keinen Echtzeitanforderungen, daher muss diese Datenverbindung ebenfalls nicht echtzeitfähig sein.

Diese Schnittstelle zwischen strategischer und maschinennaher Steuerung ist von zentraler Bedeutung für eine Wiederverwendbarkeit der strategischen Steuerungsprogramme. Obwohl die strategische Steuerung im Normalfall komplexere Aufgaben zu bewältigen hat als die maschinennahe Steuerung, kann sie relativ abstrakt implementiert werden und ist daher gut wiederverwendbar. Die maschinennahe Steuerung ist im Gegensatz dazu sehr stark von der eingesetzten technologischen Ausprägung der Fördertechnik abhängig (s. Kap. 4.2). Daher wird zwischen diesen beiden Schichten eine Middleware zur Datenübertragung eingesetzt (s. Abb. 4.9). Dabei handelt es sich um eine konfigurierbare Software, welche zwischen den beiden Steuerungsebenen vermitteln kann. Diese Middleware stellt eine einheitliche Schnittstelle zwischen strategischer und operativer Steuerung zur Verfügung und erlaubt trotzdem eine herstellereigenspezifische Ausführung der maschinennahen Steuerung.

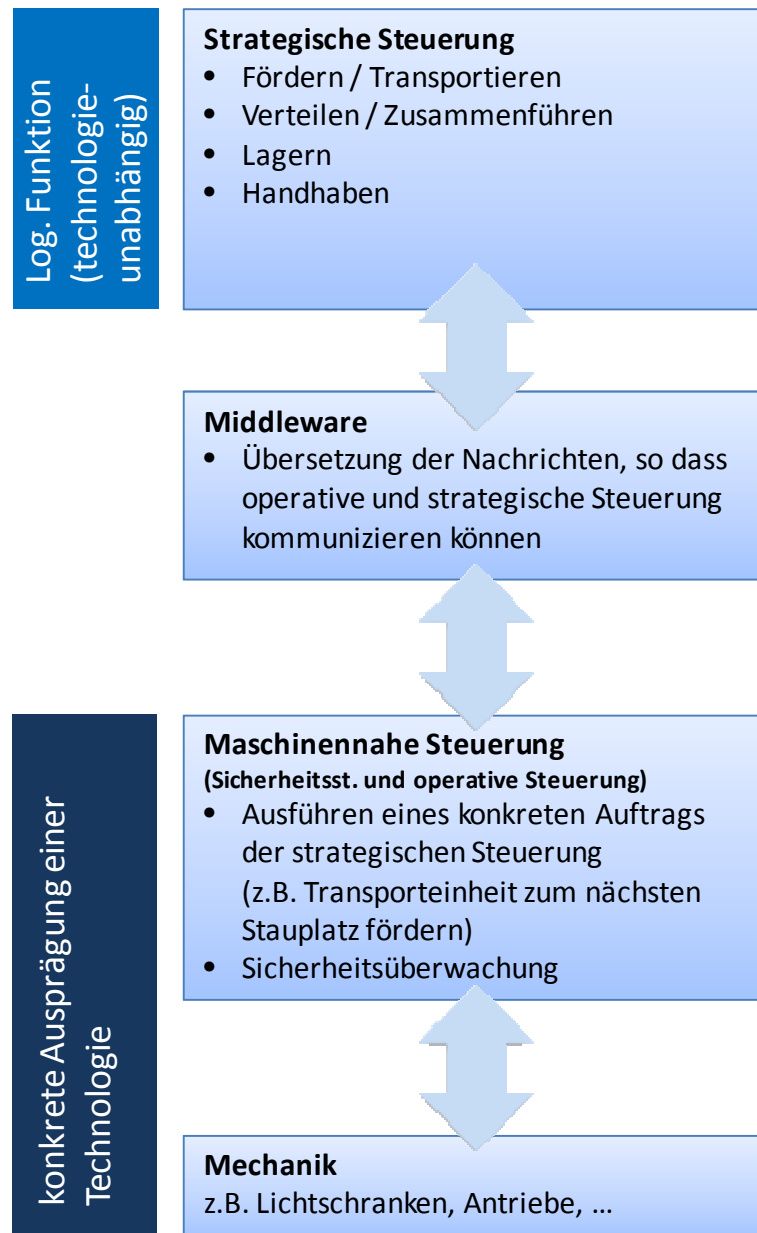


Abb. 4.9: Trennung von logistischer Funktion und konkreter Ausprägung einer Technologie durch eine Middleware

Wird die maschinennahe Steuerung emuliert, so ist es sogar möglich, die strategischen Steuerungsaufgaben getrennt vom realen System zu entwickeln und zu testen (siehe Kap. 4.4.1).

4.2.2 Steuerungssoftware

Wie in Kap. 4.2 beschrieben, verfügt ein Modul im Internet der Dinge über drei Ausprägungen von Steuerungen, welche sich in ihren Aufgaben unterscheiden.

Trotzdem lässt sich die gesamte Steuerung eines Moduls als hybrider Softwareagent betrachten (siehe Kap. 2.3.4), welcher aus zwei Schichten aufgebaut ist. Dabei sind die maschinennahen Steuerungen, also die operative Steuerung und die Sicherheitssteuerung, auf einer Schicht angeordnet (s. Abb. 4.10).

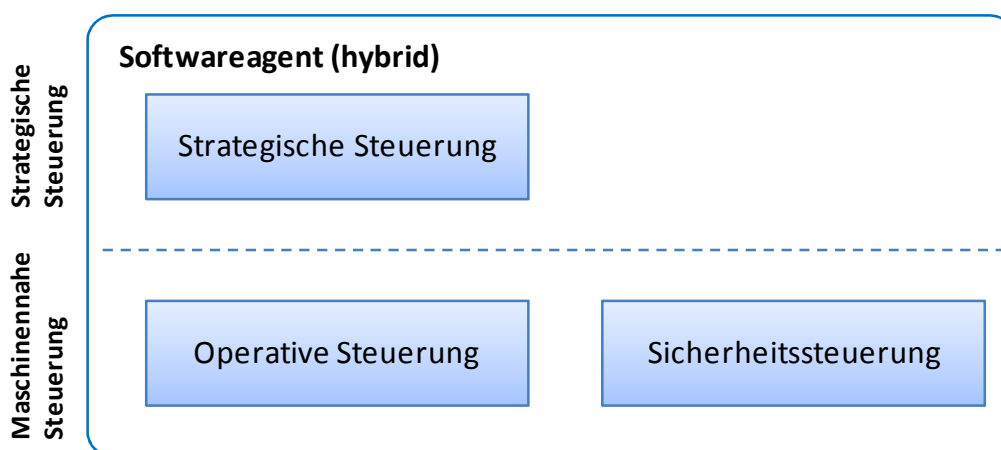


Abb. 4.10: Aufbau der Steuerung eines Moduls als hybrider Softwareagent

Für jede dieser Schichten bzw. Steuerungsarten sind unterschiedliche Programmiersprachen sowie Entwicklungs- und Ausführungsumgebungen geeignet:

Sicherheitssteuerung

Die Aufgabe der Sicherheitssteuerung ist, wie bereits beschrieben, die Erkennung von Gefahrensituationen und das Einleiten von Maßnahmen zum Entschärfen dieser Gefahren. Die dafür notwendige Steuerungslogik ist im Normalfall relativ einfach aufgebaut. Hier handelt es sich meist um eine Aneinanderreihung von Bedingungen, welche erfüllt sein müssen, damit der aktuelle Zustand der Anlage als sicher betrachtet wird. Beispielsweise dürfen Endschalter nicht betätigt oder Schutztüren nicht geöffnet sein. Die Verknüpfung der einzelnen Bedingungen wird dann über eine boolesche Logik realisiert.

Die Sicherheitssteuerung kann auf einer eigenen Hardwarekomponente oder zusammen mit der operativen Steuerung einer gemeinsamen Komponente angesiedelt sein. Die Sicherheitssteuerung kann durch eine Verbindungsprogrammierte Steuerung (VPS) realisiert werden. Dies ist z.B. eine Relaischaltung, welche eine einfache

che Logik implementiert. Alternativ gibt es programmierbare Sicherheitskomponenten, welche mit herstellerspezifischen Hilfsmitteln programmiert werden. Diese stellen einen relativ geringen, aber für die meisten sicherheitstechnischen Aufgaben ausreichenden Funktionsumfang bereit. Die Implementierung der Logik ist dabei meist grafisch unterstützt: Der Programmierer ordnet Logikbausteine auf einer Oberfläche an und verknüpft die Ein- und Ausgänge dieser Bausteine mit den physischen Ein- und Ausgängen der Steuerung. Auf diese Weise können einfache Schaltsammenhänge schnell und übersichtlich realisiert werden und das Risiko eines Programmierfehlers sinkt erheblich.

In vielen Fällen kann, sofern die Sicherheitsklasse der Steuerungsaufgabe es zulässt, die Sicherheitssteuerung auch als Unterprogramm der operativen Steuerung ausgeführt werden.

Operative Steuerung

Die klassischen Aufgaben einer operativen Steuerung sind z.B. die Verarbeitung von Eingangssignalen, Zeitschaltmechanismen, eine Antriebsregelung oder die Kommunikation über Feldbusnetzwerke, um zusätzliche Peripherie anzusteuern. Diese werden in der Regel mit den unter [IEC 61131-3] genormten Programmiersprachen umgesetzt. SPS werden zusammen mit einer Entwicklungsumgebung für diese Sprachen angeboten und liefern bereits Bausteine, welche genau auf die erwähnten Einsatzfelder zugeschnitten sind. Darüber hinaus sind zahlreiche herstellerspezifische Lösungen verfügbar, welche sich leicht in eigene Programme einbinden lassen. Zusätzlich können Fehler in der Programmlogik oder falsch angeschlossene Sensoren und Aktoren sehr leicht aufgefunden werden, da die Entwicklungsumgebungen umfangreiche Hilfsmittel zur Überwachung und Fehlerbehebung zur Verfügung stellen. Die Visualisierung aktueller Ein- und Ausgangs- sowie Variablenzustände während der Laufzeit ist nur eine der komfortablen Möglichkeiten zum Debugging von SPS-Programmen. Es ist sogar vorgesehen, neue Programmlogik auch während des Betriebs einzuspielen. Grundsätzlich sollten also Aufgaben der operativen Steuerung in IEC 61131-3 Sprachen realisiert werden.

Strategische Steuerung

Eine der wichtigsten Aufgaben der strategischen Steuerung ist die Kommunikation mit anderen Modulen, um Handlungen abzustimmen bzw. mit anderen Modulen zu kooperieren oder Rückmeldungen an die Visualisierungsumgebung zu geben. Zudem wird in der strategischen Steuerung die Logik für bestimmte Materialflusstategien wie z.B. das Sortieren oder die auslastungsabhängige Wegplanung implementiert. Diese Aufgaben weisen im Vergleich zur maschinennahen Steuerung einen höheren Abstraktionsgrad auf und sind daher auch einfacher auf andere Module übertragbar. Der Einsatz desselben strategischen Steuerungsprogramms für die verschiedenen technologischen Ausprägungen von unterschiedlicher Fördertechnik ist daher denkbar und wünschenswert (siehe Kap. 4.1.4).

Dieses Anforderungsspektrum legt den Einsatz einer Hochsprache nahe. Moderne Hochsprachen wie C++, Java oder C# sind objektorientiert und bieten daher sehr gute Möglichkeiten, ein abstraktes Problem zu strukturieren. Für die Entwicklung von Programmen kann hier auf umfangreiche Hilfsmittel, wie z.B. Editoren zur Erstellung von Klassendiagrammen und sogar der automatischen Codegenerierung daraus, zurück gegriffen werden. Zudem stehen für die etablierten Hochsprachen umfangreiche Klassenbibliotheken zur Verfügung, welche z.B. die Kommunikation von komplexen Nachrichteninhalten wie XML unterstützen. Speziell im Bereich der Multiagentensysteme sind Bibliotheken verfügbar, welche die Kommunikationsprotokolle zwischen Agenten implementieren, eine Agentenumgebung bereitstellen oder Hilfsmittel zum Debugging von Agenten liefern. Ein Beispiel dafür ist das Agentenframework JADE (Java Agent Development Framework) [Jad-09]. Dieses Framework implementiert die nach den Vorgaben der Standardisierungsorganisation FIPA [FIPA-05] genormten Spezifikationen für ein Multiagentensystem und ist eine sehr gute Grundlage für die Entwicklung eigener Lösungen.

Denkbar ist auch, jedes Modul mit einer eigenen Visualisierung auszustatten, welche die Überwachung des aktuellen Betriebszustands bzw. von Störungsmeldungen oder aktuellen Auftragsdaten ermöglicht. Hier bieten die Entwicklungsumgebungen der Hochsprachen Editoren, welche auf eine umfangreiche Bibliothek von grafischen Benutzerelementen zurück greifen und so die Gestaltung von Benutzeroberflächen erheblich erleichtern. Diese Benutzeroberflächen können dann z.B. über eine Web-

schnittstelle verfügbar gemacht werden, so dass jedes Modul quasi über seine eigene Homepage ansprechbar wird.

Zusammenfassend kann also gesagt werden, dass speziell die Aufgaben einer strategischen Steuerung eine Implementierung in Hochsprachen nahe legen.

4.2.3 Steuerungshardware

Neben der Modularisierung der Steuerungssoftware nach dem Konzept des Internet der Dinge ist die Hardware, auf der die jeweiligen Steuerungsagenten ausgeführt werden sollen, ebenfalls anzupassen. Dazu erfolgt parallel zur Softwareseite eine Modularisierung der Hardware. Die klassische Architektur, welche vorsieht, dass eine SPS einen kompletten Anlagenbereich steuert, soll so durch eine verteilte Architektur abgelöst werden. Diese ordnet, soweit technisch und wirtschaftlich sinnvoll, jedem Modul eine eigene Hardwareplattform zu (siehe Kap.3.2.2). Hier kann der Planer grundsätzlich zwischen zwei Hardwareausprägungen, nämlich PC und Speicherprogrammierbarer Steuerung (SPS), wählen. Die folgenden Abschnitte beschreiben die Anforderungen und den möglichen Aufbau einer Hardwareumgebung für die Module des Internet der Dinge, wobei die aktuellen Entwicklungen im PC- und SPS-Bereich in die Betrachtung mit einbezogen werden.

Die Anforderungen an die Steuerungshardware für die Elemente eines Materialflusssystems lassen sich in wirtschaftliche und technische Anforderungen sowie Anforderungen an die Benutzerfreundlichkeit unterteilen.

Wirtschaftliche Anforderungen

Für den Hersteller und späteren Betreiber steht der Preis der benötigten Hardware oftmals im Vordergrund. Laut Moores Gesetz [Moo-65] verdoppelt sich die Leistungsfähigkeit der Prozessoren alle anderthalb Jahre, wobei meist der Preis und die Baugröße abnehmen. Dieses Gesetz hat in den letzten Jahren seine Gültigkeit behalten und es haben sich große Fortschritte speziell im Bereich der PC-Technologie ergeben. Bei den heute verfügbaren SPS-Lösungen kann dieser Trend ebenfalls festgestellt werden, jedoch nicht ganz so dramatisch wie bei den PC. Hersteller von SPS verwenden beispielsweise meist nicht die neueste Prozessorgeneration, sondern Vorgängergenerationen, die bereits seit längerer Zeit im Einsatz sind, und so eine robuste Technologiebasis bieten. Zusätzlich verfügen die Hersteller über ausrei-

chende Lagerbestände, um sicherzustellen, dass ein Kunde auch nach einigen Jahren Ersatzteile für die Baugruppen erhält. Die gesicherte Ersatzteilversorgung ist ein nicht zu vernachlässigender Vorteil beim Einsatz einer SPS.

Technische Anforderungen

Eine klassische Anforderung an die in Materialflusssystemen eingesetzte Steuerungstechnik ist eine hohe Verfügbarkeit und Robustheit der Komponenten, denn ein Ausfall führt in der Regel zum Stillstand eines Teils der Produktionsanlage und zieht daher hohe Kosten nach sich. SPS wurden speziell vor diesem Hintergrund entwickelt und können daher eine hohe Verfügbarkeit aufweisen. Zudem sind SPS so ausgelegt, dass sie eine sehr kurze Hochlaufzeit nach einem Stromausfall oder einer geplanten Abschaltung garantieren. Hier können die Betriebssysteme und Hardwarekomponenten der PC, die zumeist aus angepassten Standardsystemen bestehen, nicht mithalten.

Die Hardware einer feldnahen Steuerung muss in der Lage sein, rechtzeitig auf Veränderungen in ihrem Zuständigkeitsbereich zu reagieren (siehe Echtzeitfähigkeit Kap. 2.3.1). Die Zykluszeit, also die Zeit, die für das Einlesen der Eingangszustände, die Verarbeitung des Steuerungsprogramms und das Schreiben der Ausgänge benötigt wird, ist stark von der Hardware und der Komplexität der Steuerungsaufgabe abhängig. Für die Ansteuerung der Fördertechnik eines Materialflusssystems bewegt sie sich in der Regel zwischen 3ms und 100ms. PC-basierte Systeme dagegen benötigen echtzeitfähige Betriebssysteme wie Windows CE oder Realtime-Linux, um dieser Anforderung gerecht zu werden.

Zudem muss die Rechengeschwindigkeit des verwendeten Systems groß genug sein, um eine für die jeweilige Aufgabe ausreichende Reaktionszeit garantieren zu können. Diese Anforderung erfüllen für herkömmliche Anwendungen sowohl eine SPS als auch ein PC.

Da die Kommunikation zwischen den strategischen Steuerungen ausschließlich über Ethernet durchgeführt wird (siehe Kap. 4.2.1.1), muss die eingesetzte Hardware auch über entsprechende Schnittstellen verfügen. Hier haben die SPS in den letzten Jahren aufgeholt. Inzwischen gilt eine Ethernetschnittstelle nach IEEE-Norm 802.3 auch in diesem Bereich als Standard. Viele Hersteller bieten zudem Industrial Ethernet-Schnittstellen an, welche sogar Echtzeitkommunikation ermöglichen (s. Kap.

2.3.1). Zusätzlich sind nach wie vor klassische Feldbussysteme wie z.B. CAN oder PROFIBUS im Einsatz, die auf absehbare Zeit weiterhin Verwendung finden werden. Dies ist bei SPS kein Hindernis, und bei PC sind entsprechende Schnittstellenkarten auf dem Markt erhältlich. Einige Hersteller bieten auch integrierte Kleinstrechner an, welche über klassische Feldbusschnittstellen verfügen.

Mit zunehmender Vernetzung und der Anbindung an das World Wide Web rückt auch die Sicherheit eines Steuerungssystems immer mehr in den Vordergrund. Zum einen muss die Steuerungskomponente nach außen mit Firewalls gesichert werden, zum anderen sollte böswillige Software auch direkt auf den eingesetzten Systemen auffindbar und zu beseitigen sein. Bei PC wird dieser Problematik mit Virenscannern begegnet, welche mit geringem administrativem Aufwand aktuell gehalten werden und regelmäßige Scanvorgänge im Hintergrund durchführen. Für SPS sind bisher keine Viren bekannt und daher auch keine Virenscanner verfügbar. Ob sich dieser Zustand in Zukunft ändern wird, kann noch nicht vorhergesagt werden. Eine Sicherung der Programme gegen die Veränderung durch unbefugte Personen, welche sich bereits im eigenen Unternehmen befinden, wird auf beiden Systemen durch die Verwendung von Passwörtern garantiert.

Anforderungen an die Benutzerfreundlichkeit

Eine Hardwarekomponente sollte, um den Programmier- und Inbetriebnahmeaufwand zu begrenzen, möglichst einfach handhabbar sein und dem Benutzer bzw. Programmierer seine Arbeit erleichtern. Dies beginnt bereits bei der richtigen Programmiersprache und -umgebung. Wie in Kap. 4.2.2 beschrieben, sollte die Implementierung der strategischen Steuerung in einer Hochsprache erfolgen, während für die Maschinensteuerungsebene Steuerungssprachen nach IEC 61131-3 von Vorteil sind.

Generell ist es auch nicht mehr notwendig, für die Fehlerbehebung vor Ort zu sein. Moderne SPS werden grundsätzlich über eine Netzwerkverbindung angesprochen, so dass der Standort des Programmierers unerheblich ist. Auf dem PC übliche Entwicklungsumgebungen für Hochsprachen wie C++, C# oder Java, bieten ähnliche Möglichkeiten zur Fehlersuche und für den Fernzugriff. Eine Darstellung aller Umgebungsvariablen während der Laufzeit oder das Einspielen neuer Programme im Online-Betrieb ist hier jedoch in der Regel nicht möglich. Trotzdem kann der PC gerade

bei Aufgaben sein Potenzial ausspielen, welche nicht direkt im Zusammenhang mit der eingesetzten Sensorik und Aktorik stehen. Die Verarbeitung komplexer Daten, wie z.B. die Interpretation einer XML-codierten Nachricht, ist auf einer SPS, wenn überhaupt, nur mit großem Aufwand zu realisieren. Wie bereits beschrieben verfügen Hochsprachen über Klassenbibliotheken, welche diese Aufgabe recht einfach implementierbar machen. Auch ist eine Anbindung an Datenbanksysteme oder ein Routingalgorithmus für Transporteinheiten in Hochsprachen sehr leicht umzusetzen, während hier IEC 61131-3 Sprachen schnell an ihre Grenzen stoßen.

Für SPS existieren Softwarepakete zur Visualisierung von Anlagenzuständen. Diese sind für einfache Visualisierungsaufgaben, z.B. für die Darstellung des Zustands einzelner Anlagenelemente, recht gut vorbereitet. Eine eher abstrakte Sicht auf ein komplexes Materialflusssystem lässt sich damit jedoch nur eingeschränkt bewerkstelligen.

Tab. 4.1 bietet einen Überblick der Vor- und Nachteile von PC und SPS:

Anforderung	SPS	PC
Preis	0	+
Nachkaufsicherheit	++	-
Verfügbarkeit / Robustheit	++	0
Hochlaufzeit	+	-
Echtzeitfähigkeit	++	+
Rechengeschwindigkeit	0	+
Ethernetanbindung	+	+
Feldbusanbindung	++	0
Sicherheit gegen Angriffe	0	-
Debugging	++	+
Fernzugriff	++	+
Überwachung / Steuerung von Ein- / Ausgängen	++	0
Kommunikation komplexer Datenstrukturen (z.B. XML)	--	++
Anbindung an Datenbanken	-	++
Umsetzung komplexer Steuerungsalgorithmen (z.B. Routing)	-	++
Darstellung einer Benutzerschnittstelle / Visualisierung	0	++

Tab. 4.1: Gegenüberstellung SPS - PC

An die einzusetzende Steuerungshardware sind also unterschiedliche Anforderungen gestellt. Für maschinennahe Aufgaben ist die SPS nach wie vor die beste Lösung. Übergeordnete Aufgaben, welche bisher von einem Materialflussrechner übernommen wurden, lassen sich jedoch besser auf einem PC-basierten System abbilden.

Architektur

Die in Kapitel 4.2.2 vorgestellte Softwarearchitektur beschreibt ein Ebenenmodell für Module im Internet der Dinge, das über eine strategische Steuerungsebene und eine Maschinensteuerungsebene verfügt. Die strategische Ebene führt dabei eher Aufgaben wie die Koordination und Kommunikation mit anderen Modulen durch, während die Maschinensteuerungsebene alle maschinennahen und echtzeitkritischen Aufgaben wahrnimmt. Durch die Unabhängigkeit der Ebenen untereinander sind verschiedene Ausprägungen der tatsächlichen Steuerungshardware denkbar. Eine vorgegebene universelle und optimale Lösung existiert nicht, da diese für jeden Einsatzfall und -bereich unterschiedlich ausfallen kann. Neben den bereits erwähnten Hardwareausführungen SPS und PC existieren Mischformen, welche diese beiden miteinander in einem Gerät vereinen.

In den folgenden Abschnitten werden mehrere Konzepte, die auf SPS, PC und/oder EPC (Embedded PC) basieren, kurz vorgestellt (siehe Abb. 4.11):

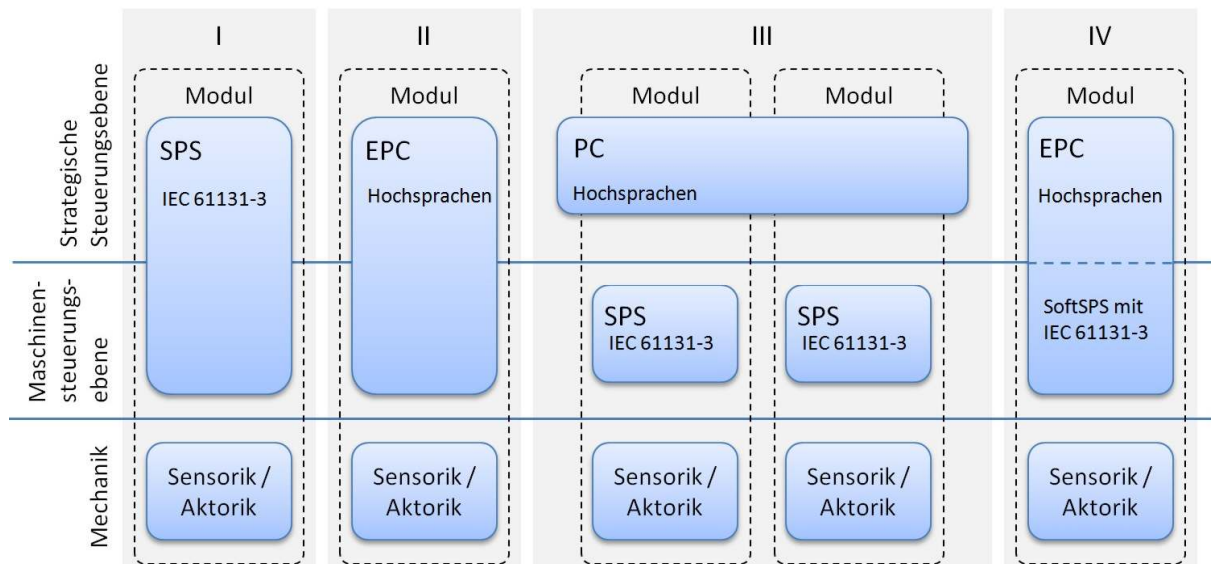


Abb. 4.11: Mögliche Hardwarekonzepte im Internet der Dinge

Konzept I: Einsatz einer SPS

Grundsätzlich kann ein Internet der Dinge-Materialflusssystem ausschließlich mit der SPS-Technologie aufgebaut werden. Hier müssen sowohl die Agentenebene als auch die Maschinensteuerungsebene auf einer SPS implementiert werden.

Die Vorteile liegen in der Reduzierung des Kommunikationsaufkommens und dem Einsatz einer einheitlichen Programmierumgebung. Der große Nachteil dieser Lösung liegt in der Umsetzung der strategischen Steuerung in IEC 61131-3. Für den Programmierer stellt sich hier das Problem, eine Agentenumgebung und deren Kommunikation in Sprachen zu implementieren, die für diese Aufgaben nicht geeignet sind. Auf die beim PC bereits vorhandenen Agentenframeworks kann hier nicht zurück gegriffen werden.

Konzept II: Einsatz eines Embedded-PC ohne Soft-SPS

Ein Embedded-PC ohne Soft-SPS vereint die Agenten- und die Maschinensteuerungsebene auf einer Plattform und einem Betriebssystem. Eventuell muss ein Betriebssystem mit Echtzeitfähigkeit eingesetzt werden, um die Funktionen der steuerungstechnischen Ebene korrekt umsetzen zu können.

Der Vorteil, der durch den Einsatz eines Embedded-PC ohne Soft-SPS entsteht, ist die Reduzierung der Kosten des Moduls, da das Soft-SPS-Betriebssystem nicht lizenziert werden muss. Ein Nachteil, der durch diese Vorgehensweise entsteht, ist die Umsetzung der steuerungstechnischen Ebene in Hochsprachen, da diese Umsetzung neue softwaretechnische Konzepte erfordert und nicht in IEC 61131-Sprachen erfolgen kann.

Konzept III: Kombination aus PC und SPS

Die Kombination aus einem Standard-PC und einer SPS kann als Übergangslösung für die Migration eines bestehenden zentral gesteuerten Materialflusssystems auf das dezentrale Konzept des Internet der Dinge gesehen werden. Der PC stellt dabei die Rechenplattform für die Softwareagenten sowie die Ethernet-Anbindung zur Kommunikation mit anderen Modulen zur Verfügung. Der PC deckt die Agentenebene ab. Die Kommunikation zur SPS, welche die Maschinensteuerungsebene abbildet, kann beispielsweise über ein Feldbussystem oder ebenfalls über Ethernet erfolgen.

Die Vorteile dieser Lösung liegen in dem relativ geringen Aufwand zur Einrichtung des Agentensystems auf dem PC und der einfachen Feldsteuerung über die SPS. Nachteile ergeben sich durch den hohen Kommunikationsaufwand zwischen PC und SPS.

Konzept IV: Einsatz eines Embedded-PC mit Soft-SPS

Ein Embedded-PC (EPC) mit Soft-SPS ist ein modular aufgebauter, kleiner Industrie-PC, der sowohl über ein herkömmliches Betriebssystem (beispielsweise Windows oder Linux) als auch über ein SPS-Betriebssystem, das mit IEC 61131-3 Sprachen programmiert werden kann, verfügt. Diese Variante vereint die Vorteile der beiden vorher genannten Lösungen. Die Soft-SPS bildet die Maschinensteuerungsebene samt der echtzeitrelevanten Vorgänge ab, die Agentenebene wird im herkömmlichen (PC)-Betriebssystem in Hochsprachen implementiert.

Die Vorteile liegen in der einfachen Programmierung der maschinennahen Funktionen in IEC 61131-3 und der Softwareagenten in Hochsprachen. Durch den Einsatz einer einzigen Hardware-Plattform entfällt die externe Schnittstelle. Der Nachteil besteht allerdings im noch relativ hohen Preis der Hardware.

Fazit

Ein Embedded-PC mit Soft-SPS erfüllt, abgesehen vom Kaufpreis, die eingangs beschriebenen Anforderungen am besten. Daher kann diese Lösung durchaus als Grundlage für das Internet der Dinge empfohlen werden. Es ist jedoch im Einzelfall zu prüfen, ob diese Investition notwendig ist, oder ob aus Kostengründen andere Konzepte angewandt werden sollen. Dies kann z.B. der Fall sein, wenn die Maschinenebene nur einen geringen Komplexitätsgrad und Umfang hat. Diese Ebene in Hochsprachen statt in IEC 61131-3 zu implementieren, wäre dann keine große Einschränkung und es kann das Konzept II eingesetzt werden.

Die Agentenebene sollte grundsätzlich in Hochsprachen implementiert werden, da der Aufwand, diese auf einer SPS zu implementieren, in keinem Verhältnis zum Nutzen steht. Aus diesem Grund sollte vom Konzept I prinzipiell Abstand genommen werden.

4.2.4 Additive Komponenten

In den vorangegangenen Kapiteln wurde die Gestaltung eines Moduls bezüglich logischer und hardwaretechnischer Aspekte erläutert. Darüber hinaus gibt es jedoch noch zusätzliche Komponenten, welche einem Modul im Hinblick auf eine Absenkung des Neuentwicklungsaufwands bei Folgeprojekten zugeordnet werden sollten. Hierbei handelt es sich z.B. um Dokumentationen für die verschiedensten Teile eines Moduls, welche je nach Lebenszyklusphase eine eigene Sicht auf das Modul gestatten, und um Hilfsmittel, die an verschiedenen Stellen im Lebenszyklus einer Materialflussanlage eingesetzt werden können. In Tabelle Tab. 4.2 ist beispielhaft eine Zusammenstellung aufgeführt, welche eine Zuordnung bzgl. der Verwendbarkeit dieses Zusatzmaterials vornimmt und auf Nieke [Nie-09] aufbaut. Die hier aufgeführten Punkte sind lediglich eine Auswahl der möglichen additiven Komponenten eines Moduls und können nach Bedarf erweitert werden.

Die in Kap. 4.1.3 beschriebene mechatronische Gestaltung eines Moduls erleichtert die eindeutige Zuordnung von Informationen zum jeweiligen Modul und erhöht somit die Wiederverwendbarkeit von Arbeitsumfängen erheblich. Es bietet sich für einen Hersteller von Materialflussmodulen an, zur Verwaltung aller Informationen ein Produktdatenmanagementsystem einzusetzen. Dieses dient der strukturierten Aufbereitung und Speicherung aller zu einem Produkt gehörenden Daten.

	Planung	Realisierung	Inbetriebn./ Hochlauf	Betrieb
Allgemein				
Bedienungsanleitungen		x	x	x
Verwendete Normen und Konformitätserkl.	x	x		
Berechnungsunterlagen (z.B. bzgl. Durchsatz)	x	x		
Arbeitspläne		x	x	
Qualitätssicherungsunterlagen		x	x	
Studien	x	x		
Mechanik / Elektrotechnik				
Konstruktionsplan	x	x		
Stückliste		x		
Verkabelungspläne		x		
Inbetriebnahme- und Installationsanleitung		x	x	
Einstell- und Prüfvorschriften		x	x	
Sicherheitsanalyse			x	
Wartungsvorschriften				x
Dokumentation für Zukaufteile	x	x	x	
Ersatzteillisten				x
Logik / Software				
Steuerungsprogramme und -dokumentation	x	x	x	x
Prüf- und Emulationssoftware		x	x	
Simulationsbausteine	x	x		
Strategiebeschreibungen	x	x	x	
Hilfsmittel zum Debugging		x	x	x
Vertriebsmaterial				
Kalkulationsunterlagen	x			
Werbematerial	x			

Tab. 4.2: Additive Komponenten und ihre Verwendbarkeit in versch. Lebenszyklusphasen

4.3 Entwicklung der Modulsteuerung

Nachdem der grundsätzliche Aufbau von Modulen für das Internet der Dinge beschrieben wurde, soll nun auf Methoden eingegangen werden, welche die Entwicklung der Steuerungssoftware für diese Module erleichtern und deren Wiederverwendungsgrad erhöhen. Dabei werden ein Vererbungsmodell für Modulfunktionen vorgeschlagen, die Entwicklung von Materialflusstategien beschrieben, und Möglichkeiten zur Anpassung der Modulsoftware mittels Konfigurierung und Parametrierung aufgezeigt.

4.3.1 Vererbungsmodell für Modulfunktionen

Vergleicht man die Konzepte des Internet der Dinge mit denjenigen der objektorientierten Programmierung (OOP), so lassen sich folgende Ähnlichkeiten feststellen:

- Klassenbildung

Genauso wie in der OOP lassen sich die Module im Internet der Dinge in Klassen einteilen (siehe Kap. 4.1.4). Unter einer Klasse wird in der Informatik ein abstraktes Modell bzw. ein Bauplan für eine Reihe von ähnlichen Objekten verstanden.

- Abstraktion

Die OOP ähnelt kognitionspsychologisch gesehen der menschlichen Denkweise [Sch-93]. So wie der Mensch Dinge benutzt, ohne dass er deren innere Struktur kennt, soll ein Objekt in der OOP einsetzbar sein, ohne dass der Anwender die zugrunde liegende Implementierung verstehen muss. Das Objekt kann losgelöst von seinen konkreten Eigenschaften betrachtet werden. Diese Abstraktion ist ebenso ein Grundkonzept für die Module des Internet der Dinge.

- Kapselung

Die Module des Internet der Dinge verwenden genauso wie die Objekte der OOP das Geheimnisprinzip (siehe Kap. 1.2.3). Das bedeutet, dass der interne Zustand entweder für andere nicht sichtbar ist oder nur über klar definierte Schnittstellen abgerufen werden kann.

Es bietet sich an, auch das Konzept der Vererbung aus der OOP auf die strategische Steuerung des Internet der Dinge anzuwenden, um die Wiederverwendbarkeit der Lösungskonzepte zu erhöhen. Die Vererbung ist eine Vorgehensweise zur Schöpfung neuer Klassen. Hier werden Funktionen von bereits existierenden Klassen übernommen und lediglich Spezialisierungen neu implementiert. Übergeordnete Klassen können auf diese Weise Schnittstellen und Grundkonzepte implementieren.

Für die Module im Internet der Dinge wird eine Vererbung in vier Stufen vorgeschlagen, wobei auf den oberen Ebenen eher abstrakte, vom konkreten Gerät unabhängige Aufgaben implementiert werden, und nach unten eine Spezialisierung vorgenommen wird (s. Abb. 4.12).

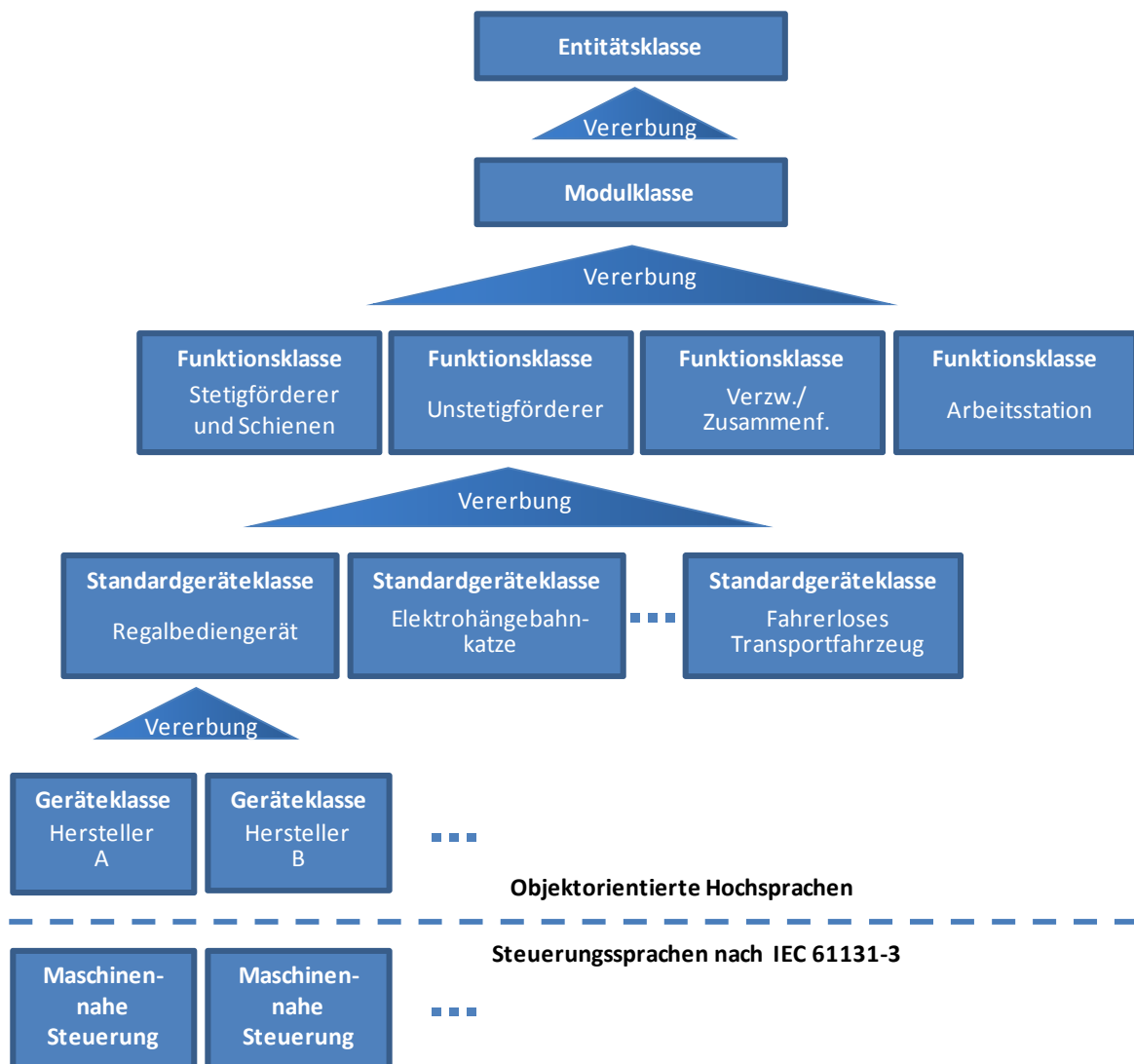


Abb. 4.12: Vererbungsmodell für Module

Das Konzept der Vererbung wird in der OOP noch um das Konzept der Polymorphie erweitert. Polymorphie bedeutet, dass eine Nachricht, welche an verschiedene Objekte untergeordneter Klassen verschickt wird, unterschiedliche Reaktionen auslösen kann, obwohl die Nachrichtenschnittstelle in einer übergeordneten Klasse identisch definiert wurde.

Übertragen auf die Module im Internet der Dinge bedeutet Polymorphie, dass beispielsweise verschiedene Unstetigförderer ihre Transportfunktion unterschiedlich erbringen, obwohl die Transportanfrage immer den gleichen Aufbau hat. Auf diese Weise kann der Aufbau einer Transportanfrage und deren Schnittstelle nur einmal in der Funktionsklasse, also gültig für alle Unstetigförderer, implementiert werden, die konkrete Ausführung des Transports wird dann in einer untergeordneten Klasse gesteuert. Polymorphie ist also ein besonderes Vererbungskonzept, welches die Wiederverwendbarkeit der Programmlogik zusätzlich erhöht.

Im Folgenden werden die Klassen des Vererbungsmodells und ihre Funktionen näher erläutert:

Entitätsklasse

Die Entitätsklasse implementiert eine allgemeingültige Schnittstelle zu anderen Entitäten und nimmt eine erste Sortierung eingehender Nachrichten vor (siehe Abb. 4.13). Abgeleitete Klassen sind damit in der Lage, diese Nachrichten auszulesen und auf Methoden der Entitätsklasse zurückgreifen, um ihre Antworten zusammenzustellen.

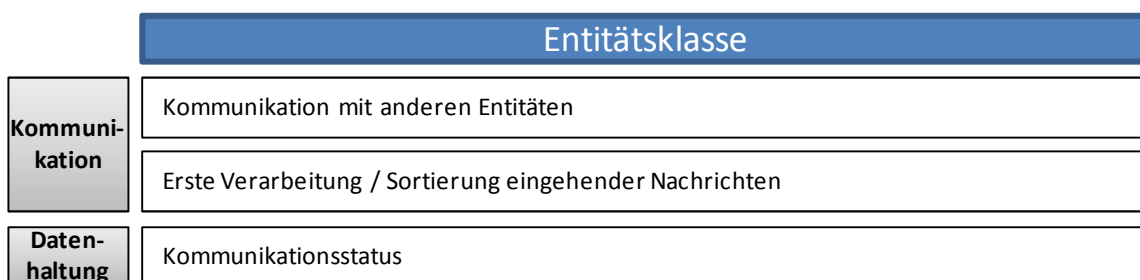


Abb. 4.13: Entitätsklasse

Modulklasse

Die Modulklasse verwaltet ein aktuelles Umweltmodell. Dieses umfasst die Topologie des Systems und weitere für alle Module relevanten Systeminformationen wie z.B. die aktuelle Auslastung, die Positionen anderer Module oder eventuelle Ressourcen-

reservierungen. Zudem stellt die Modulkasse eine generische Schnittstelle zur Maschinensteuerung zur Verfügung (Middleware). Diese ermöglicht allen untergeordneten Klassen mit der Maschinensteuerung oder einem Emulator (siehe Kap. 4.4.1) zu kommunizieren. Sie wird so gestaltet, dass sie leicht an unterschiedliche Hardwareausführungen anpassbar ist und die strategische Steuerung unabhängig vom Typ der maschinennahen Steuerung entwickelt werden kann.

Zudem verwaltet die Modulkasse den allgemeinen Zustand des Moduls. Letzterer gibt beispielweise an, ob das Modul gerade einsatzbereit ist oder ob eine Störung vorliegt. In Abb. 4.14 sind diese Funktionen abgebildet.

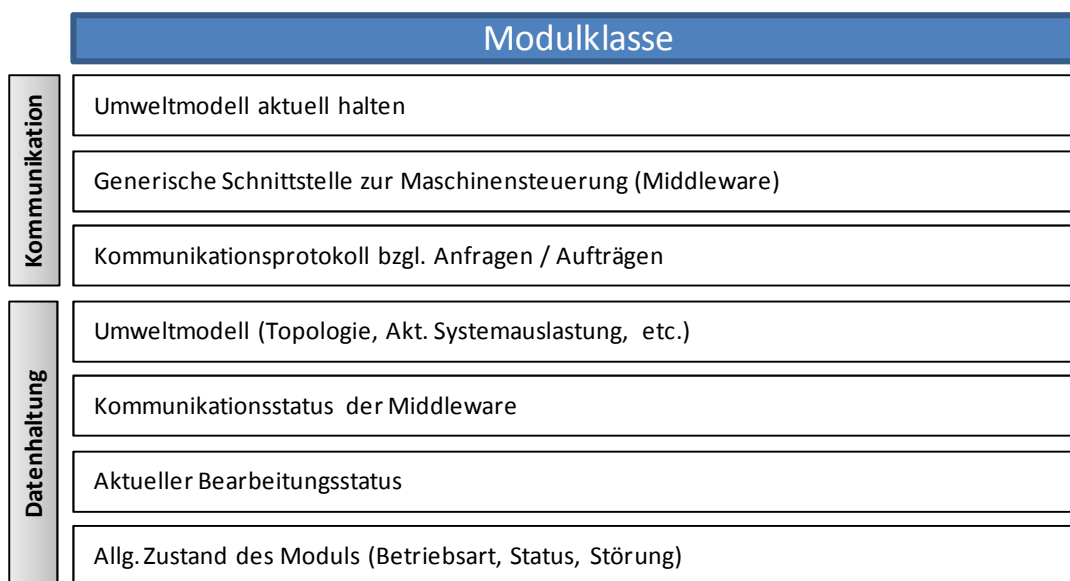


Abb. 4.14: Modulkasse

Funktionsklasse

Die Funktionsklasse wird von der Modulkasse abgeleitet und dient der Verwaltung von Anfragen bzw. Aufträgen anderer Module. Diese Aufgabe ist stark von der Zugehörigkeit zu einer der vier Funktionskategorien für Module (siehe 4.1.4) abhängig. Ein Schaltauftrag für eine Verzweigungsstelle verfügt über andere Daten als ein Transportauftrag für einen Unstetigförderer. Daher wird die Verarbeitung der eingehenden Anfragen bzw. Aufträge für jede der Funktionsklassen unterschiedlich implementiert. Allen Funktionsklassen ist gemein, dass sie prinzipiell eine Verarbeitung eines eingehenden Auftrags anbieten bzw. den jeweiligen Auftragsbearbeitungsstatus verwalten (siehe Abb. 4.15).

Funktionsklasse					
	Allg. Beschreibung	Stetig. und Schienen	Unstetig-förderer	Verzw. / Zusammenf.	Arbeitsstation
Kommunikation	Funktionsspez. Auftragsstruktur	Transportauftrag	Transportauftrag	Schaltaufträge	Bearbeitungs- / Kommissionierauftrag
Datenhaltung	Funktionsspez. Auftragsdaten	Start, Ziel	Start, Ziel, Art d. Lastwechsels, ...	Gewünschte Schaltstellung	Arbeitsauftrag für den Bediener

Abb. 4.15: Funktionsklasse

Standardgeräteklasse

Die Standardgeräteklasse wird von der Funktionsklasse abgeleitet und verfügt daher über alle Funktionen der übergeordneten Klassen. Sie implementiert in erster Linie das strategische Verhalten eines Moduls und die Überwachung eines Arbeitsspiels.

Das strategische Verhalten wird dabei sowohl durch Verhaltensregeln und Kostenfunktionen repräsentiert als auch durch spezielle Algorithmen, welche auf das jeweilige Gerät zugeschnitten sind. In Abb. 4.16 sind für vier Geräte, welche unterschiedlichen Funktionsklassen zugeordnet werden, beispielhafte Funktionen der Geräteklasse dargestellt. Bei einem Kommissionierplatz ist ein besonderer Auktionsmechanismus zu integrieren, welcher dafür sorgt, dass die Kommissionierer gleichmäßig mit Nachschub versorgt werden. Im Unterschied dazu sind für einen Querverschiebewagen Abstimmungsmechanismen zu implementieren, welche eine Kollision mit anderen QVW verhindern bzw. die Zuteilung der Transportaufträge zu den jeweiligen QVW steuern. Die Entwicklung einer Materialflussstrategie für Querverschiebewagen ist in Kap. 4.3.2.4 beschrieben.

Es ist davon auszugehen, dass ein Arbeitsspiel von verschiedenen Geräten unterschiedlicher Hersteller bezüglich seines grundsätzlichen Ablaufs gleich gestaltet ist. Ein Einzel- bzw. Doppelspiel für Regalbediengeräte wird beispielsweise in [VDI-2516] allgemeingültig definiert. Die in der Standardgeräteklasse implementierte Logik initiiert die Einzelschritte, welche für ein Arbeitsspiel ausgeführt werden müssen, und überwacht die ordnungsgemäße Abarbeitung dieser Schritte.

Standardgeräteklasse					
Allg. Beschreibung	Beispiele				
	Rollenbahn (Stetigförderer)	Querver- schiebe- wagen (Unstetigförd.)	Rollenbahn- weiche (Verzweigung)	Komm. Platz (Arbeitsstation)	
Kommunikation	Kooperations- protokolle für Strategien		Abstimmung mit anderen QVW		Auktions- mechanismus f. Zuteilung der Aufträge
	Aufträge an andere Module veranlassen	Weitertransport anfordern	Weitertransport anfordern		Nachschub anfordern
Steuerung	Ansteuern / Überwachen des Arbeitsspiels	Fördern bis Lichtschranke	Verfahren, Lastwechsel	Schaltvorgang	z.B. Pick by Light
	Strategie		Auftragsdisp., Tourenplanung, Zeitl. Koord.	Kollisions- kontrolle	Gleichm. Auslastung auf versch. Komm. Plätzen
Datenhaltung	Gerätespez. Zustand	In Bewegung	Position auf der Schiene	Akt. Schaltzustand	Komm. wartet auf neuen Auftrag
	Eigene Belegung, Reservierungen	Stauplatz X belegt	Belegt mit einer TE	Weiche frei	Komm- issionierer ist neuem Auftr. zugeteilt

Abb. 4.16: Standardgeräteklasse

Geräteklasse

In der Geräteklasse hat der Entwickler die Möglichkeit, herstellereigene Sonderfunktionen zu implementieren. Dies kann z.B. eine Konturenkontrolle auf einem Querverschiebewagen oder eine Wiegefunktion sein. Außerdem implementiert die Geräteklasse die Kommunikationsschnittstelle zur Maschinensteuerung, indem sie, je nach konkretem Steuerungstyp entsprechende Funktionen der Middleware aufruft, welche von der Modulkategorie abgeleitet wurden (siehe Abb. 4.17).

Geräteklasse			
Allg. Beschreibung		Beispiele	
		QVW Hersteller A	QVW Hersteller B
Kommunikation	Komm. mit Maschinensteuerung	Komm. mit Steuerung Typ A	Komm. mit Steuerung Typ B
	Komm. bzgl. Sonderfunktionen	Waage	Konturenkontrolle
Steuerung	Sonderf. veranlassen/überwachen	Wiegen	Scannen
Datenhaltung	Zustand bzgl. Sonderfunktionen	Gewicht	Profil i.O.

Abb. 4.17: Geräteklasse

Maschinennahe Steuerung

Die maschinennahe Steuerung muss, wie in Kap. 4.2.3 beschrieben, auf einer echtzeitfähigen Plattform implementiert werden. Hier finden normalerweise die nach IEC 61131-3 genormten Programmiersprachen Verwendung. Aus diesem Grund ist die Maschinensteuerung nicht als Klasse ausgeführt und ist kein Teil des Vererbungsmodells. Sie wird durch ein SPS-Programm realisiert, welches stark auf das jeweilige Gerät abgestimmt ist. Die Maschinensteuerung kann Arbeitsaufträge ausführen, welche von der strategischen Steuerung angewiesen werden und Teilschritte eines Arbeitsspiels darstellen. Dies ist zum Beispiel die Fahrt eines Regalbediengerätes vom E/A-Punkt zu einem Lagerplatz in der Gasse oder eine Wiegefunktion, die auf einem Querverschiebewagen realisiert ist.

Maschinennahe Steuerung			
Allg. Beschreibung		Beispiele	
		QVW Hersteller A	QVW Hersteller B
Kommunikation	Kommunikation mit der strat. Steuerung	Variable an Adresse X setzen	Variable an Adresse Y setzen
	Steuerung	Teile eines Arbeitsspiels durchführen / überwachen	Wiegen
Sicherheitssteuerung		Position X nicht überfahren	Position Y nicht überfahren
Handbetrieb		Verfahren mit Motor X	Verfahren mit Motor Y
Datenhaltung	Auftragsstatus	Wiegen abgeschlossen	Konturenkontrolle abgeschlossen
	Maschinenzustand (Sensoren / Aktoren)	Zustand von Lichtschranke X	Zustand von Lichtschranke Y

Abb. 4.18: Maschinennahe Steuerung

Die maschinennahe Steuerung ist zudem zuständig für die Überwachung aller sicherheitsrelevanten Funktionen (siehe dazu Kap. 4.2) und implementiert den Handbetrieb eines Moduls. Zusätzlich verwaltet die maschinennahe Steuerung den aktuellen Maschinenzustand, also das Abbild aller Sensoren und Aktoren eines Moduls. Damit stellt die Maschinensteuerung alle Funktionen bereit, um ein Modul unabhängig von der restlichen Anlage aufzubauen und die ordnungsgemäße Funktion seiner mechanischen Komponenten bzw. einzelner Schritte eines Arbeitsspiels zu prüfen.

Das vorgeschlagene Vererbungsmodell erleichtert einerseits die Wiederverwendung von Steuerungsprogrammen, so dass ein höherer Anteil an bereits getestetem Code

eingesetzt werden kann, und sorgt andererseits dafür, dass Programmieraufgaben parallelisiert werden können.

4.3.2 Entwicklung von Materialflusstategien

An vielen Stellen in einem Materialflusssystem ist es notwendig, die Aufteilung von Ressourcen zwischen verschiedenen Systemteilnehmern zu organisieren oder Regeln festzulegen, nach welchen die einzelnen Transportaufträge abgearbeitet werden sollen. Dies kann z.B. eine Regelung für das Wegerecht auf gemeinsam genutzten Fahrwegen, die Verfolgung einer LIFO- bzw. FIFO-Strategie oder die bevorzugte Abarbeitung von Eilaufträgen sein.

Eine Grundvoraussetzung für den störungsfreien Betrieb ist die Verhinderung von Blockadesituationen, den sogenannten Deadlocks. Ein Deadlock ist eine Situation, in welcher ein oder mehrere gleichzeitig ablaufende Prozesse eines Systems unendlich lange blockiert werden, weil diese Prozesse Ressourcen benötigen, welche ihnen niemals zur Verfügung stehen werden [Kim-97]. Ein Deadlock entsteht nach Coffman [Cof-71] und Liu [Liu-01], wenn folgende Bedingungen gleichzeitig erfüllt sind:

1. Die umstrittenen Ressourcen sind nur exklusiv von einem Prozess nutzbar.
2. Die Prozesse blockieren die ihnen bereits zugewiesenen Ressourcen, während sie auf zusätzliche Ressourcen warten.
3. Die Ressourcen können keinem Prozess entzogen werden.
4. Es gibt eine geschlossene Kette von sich gegenseitig blockierenden Prozessen.

Diese Bedingungen können beispielsweise dann erfüllt sein, wenn zwei Unstetigförderer als Fahrtziel jeweils den Standort des anderen anzufahren haben (siehe Abb. 4.19) und vor ihrer Abfahrt darauf warten, dass dieser frei wird.

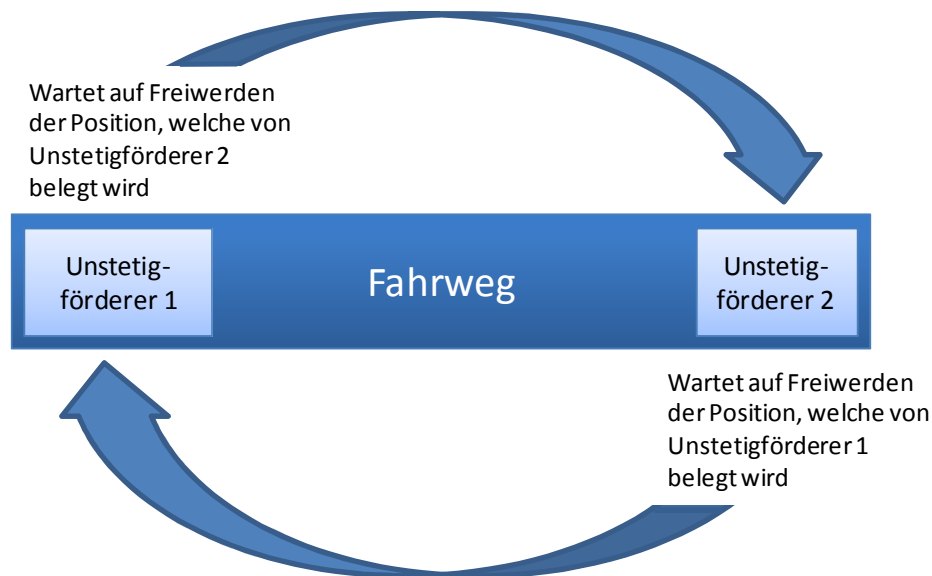


Abb. 4.19: Deadlocksituation

Hier ist der Fahrweg eine begrenzte Ressource, da niemals zwei Förderer an derselben Position stehen können. Eine solche Situation kann entweder im Vorfeld durch entsprechende Regeln bei der Auftragsvergabe vermieden werden, oder es werden Maßnahmen zur Auflösung der Blockade ergriffen. Eine solche Maßnahme ist z.B., einem Unstetigförderer seine Ressource zu entziehen, er also für den anderen Platz machen muss. Welcher Förderer Vorfahrt gewähren muss, kann dabei durch Prioritäten geregelt werden, welche zusätzlich durch übergeordnete Ziele beeinflusst sind. Das Erreichen einer bestimmten Durchsatzleistung oder die gleichmäßige Auslastung der Unstetigförderer wären solche übergeordneten Ziele.

4.3.2.1 Fragestellungen bei der Entwicklung

Prinzipiell sind bei der Entwicklung einer Materialflussstrategie folgende Fragestellungen zu beantworten:

- Welches Gerät soll welchen Auftrag bearbeiten?
(Auftragsdisposition)
- Welches ist die günstigste Route, um die Ziele anzufahren und welche Auftragsreihenfolge erzielt den besten Durchsatz?
(Routing bzw. Tourenbildung)
- Wann werden vorliegende Aufträge am besten bearbeitet, so dass möglichst wenige Behinderungen zu erwarten sind?
(Zeitliche Koordination)

In herkömmlichen Materialflusssystemen werden diese Vorgaben durch einen Bereichs- oder Materialflussrechner umgesetzt, der gegenüber dezentral gesteuerten Systemen den Vorteil hat, jederzeit auf ein konsistentes Datenmodell zugreifen und wie ein „Absolutherrscher“ die ihm unterstellten Fördertechnikelemente befehligen zu können. Dies kann jedoch, wie eingangs beschrieben, zu komplexen und schwer überschaubaren Steuerungsprogrammen führen, welche nur schwer erweiterbar sind.

Je nachdem, welcher Teil einer Materialflussanlage im Fokus der Betrachtung steht, können für die erwähnten Fragestellungen unterschiedliche Lösungen gefunden werden. Diese Materialflussstrategien bieten in der Regel für den praktischen Einsatzfall ausreichende, aber nicht unbedingt mathematisch optimale Ergebnisse. Es gibt zwar meist optimale Lösungen, wenn man jede Fragestellung unabhängig von den anderen betrachtet. Diese Teiloptima führen jedoch noch nicht automatisch zu einem Gesamtoptimum. Zudem hat die Auswahl einer Handlungsalternative im Transportprozess Auswirkungen auf alle folgenden Entscheidungen. Daher sind analytische Lösungsverfahren, welche alle drei Fragestellungen und ihre Auswirkungen aufeinander gleichermaßen berücksichtigen, nicht vorhanden. Ebenso gestaltet sich eine simulationsbasierte Suche nach dem Optimum als schwierig, da schnell eine sehr große Anzahl von kombinatorischen Möglichkeiten entsteht, die mit aktuellen Rechnern nicht mehr in einem vertretbaren Zeitraum zu bewältigen ist. Dies führt dazu, dass es sich bei der Entwicklung von Materialflussstrategien meist um einen iterativen Prozess handelt, bei welchem der Programmierer die Logik schrittweise implementiert und simulativ die Leistungsverbesserung für eine größere Anzahl von Transportaufträgen überprüft (siehe Abb. 4.20).

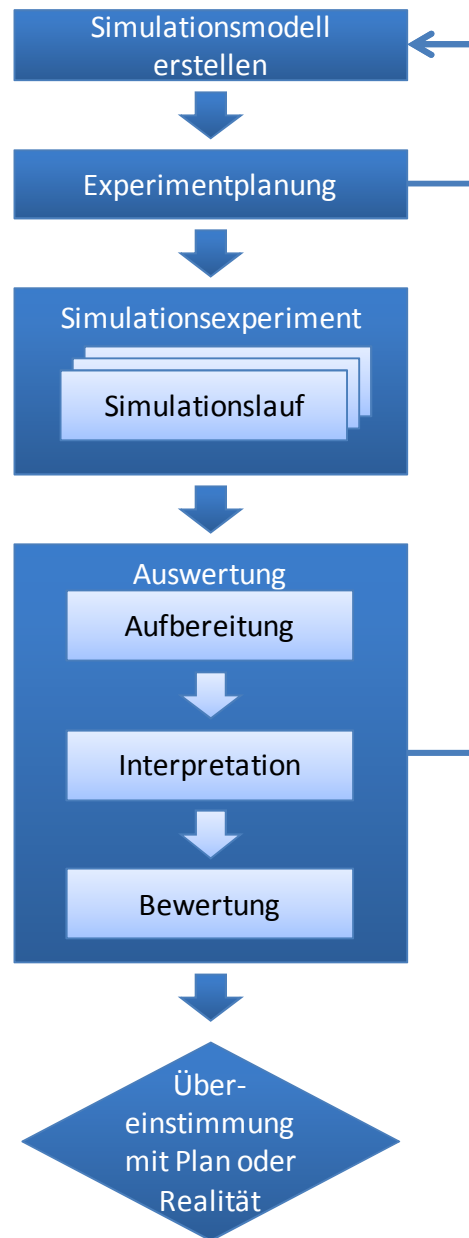


Abb. 4.20: Vorgehensweise bei der Simulation von Logistiksystemen [VDI-3633]

Dabei gibt es, wegen des Fehlens analytischer Verfahren, keine zuverlässige Möglichkeit zu beurteilen, wie weit man noch von der optimalen Materialflussstrategie entfernt ist.

4.3.2.2 Dezentrale Koordination

Auch im Internet der Dinge wird die optimale Materialflussstrategie schwer zu finden sein. Dennoch erleichtert die dezentrale Organisation im Internet der Dinge die Entwicklung der Strategien erheblich, da die Steuerungsaufgabe in überschaubare Teile strukturiert, und so die Komplexität reduziert werden kann (siehe Kap. 1.2.3). Die in

den vorangegangenen Kapiteln beschriebenen Module müssen dabei so gestaltet werden, dass sie über Abstimmungsmechanismen verfügen, welche eine Zusammenarbeit ermöglichen. Diese werden in der strategischen Steuerung eines Moduls implementiert (siehe Kap. 4.2).

Neben der grundsätzlichen Anforderung, dass lokale Zielkonflikte niemals zu einer Blockade führen dürfen, ist die angestrebte Materialflussstrategie mittels Verhaltensregeln und Verhandlungsmechanismen so umzusetzen, dass sich ein gerichtetes Zusammenspiel ergibt. Nach Huhns [Huh-99] kann diese Koordination durch die zwei Prinzipien Kooperation und Wettbewerb erreicht werden (siehe Abb. 4.21).

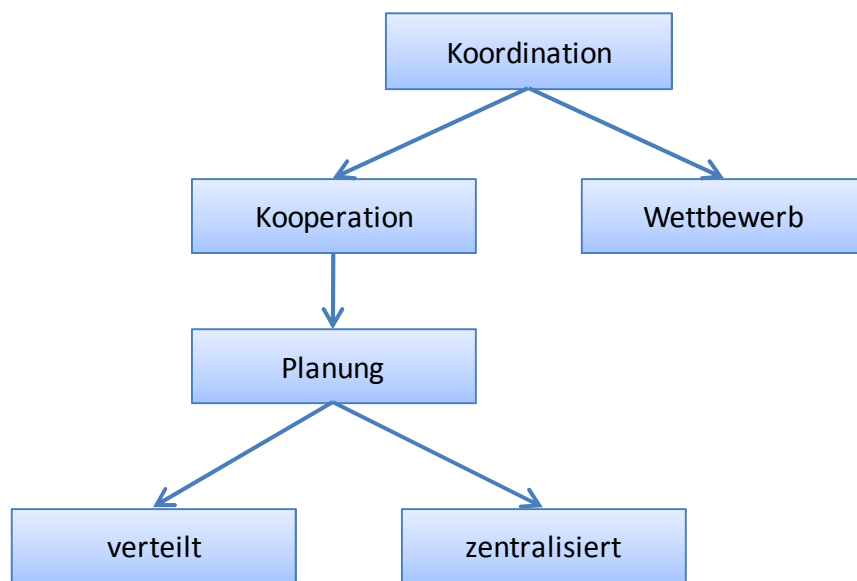


Abb. 4.21: Taxonomie von Lösungen des Koordinationsproblems [Huh-99]

Eine Wettbewerbssituation liegt u.a. dann vor, wenn mehrere Module um eine Ressource oder einen Transportauftrag konkurrieren. Die Entscheidung darüber, welcher Wettbewerber den Zuschlag erhält, wird dann beispielsweise durch eine Auktion geregelt, in der derjenige den Zuschlag erhält, der die niedrigsten Kosten für die Erledigung der Aufgabe veranschlagt.

Kooperation wiederum setzt entweder verteilte oder zentralisierte Planung voraus. In einem zentral gesteuerten Materialflusssystem kann diese Planungsaufgabe durch den Materialflussrechner oder eine Bereichs-SPS übernommen werden, welche beispielsweise festlegt, auf welche Weise Transporteinheiten von einem Querverschiebewagen auf die verschiedenen Bahnen zu sortieren sind.

Im Internet der Dinge kommt eine verteilte Planung zum Einsatz. Dazu müssen die verschiedenen Module entweder direkt miteinander kommunizieren und ihre Handlungen untereinander abstimmen, oder sie nutzen dazu eine gemeinsame Datenaustauschplattform.

Im Forschungsgebiet der Multiagentensysteme wurden bereits zahlreiche Arbeiten veröffentlicht, welche Koordinationsprotokolle und -mechanismen behandeln, um Konflikte zu vermeiden bzw. aufzulösen:

Huhns und Stephens schlagen spezielle Interaktionsprotokolle für unterschiedliche Koordinationsprobleme wie z.B. die Knowledge Query and Manipulation Language (KQML) vor und entwickeln diese weiter [Huh-99]. Eymann beschreibt einen agentenbasierten dezentralen Koordinationsmechanismus für elektronische Märkte [Eym-00]. Muscholl verfolgt die Grundidee, Kooperation zu erzielen, indem die Agenten einen Teil ihrer Autonomie an ein Interaktionsverfahren abtreten und sich diesem unterordnen [Mus-01]. Gerber untersucht die dynamische Kooperation in holonischen Multiagenten-Systemen und setzt dazu die Vektorrechnung ein [Ger-04]. Wellner zeigt die grundsätzlichen Lösungsverfahren für Koordinations- und Kooperationsprobleme auf und setzt seinen Schwerpunkt auf die Betrachtung selbstorganisierender Kommunikationssysteme für Multiagentensysteme [Wel-03]. Timm beschreibt die Autonomie und dynamische Koordination in kooperativen Systemen und stellt eine Agentenarchitektur zur Konfliktbehandlung vor [Tim-04].

Obwohl die erwähnten Arbeiten bereits umfangreiche Lösungsansätze zur Behandlung des Kooperationsproblems bieten, besteht die kreative Leistung eines Entwicklers nach wie vor darin, eine spezielle Koordinationsaufgabe auf die verschiedenen dezentralen Entitäten aufzuteilen. Dazu sind Regeln zu entwickeln bzw. Kostenfunktionen aufzustellen, welche ein gewünschtes Gesamtverhalten erzeugen. Diese Regeln müssen auf die jeweiligen Anwendungsgebiete genau eingestellt und an die speziellen Anforderungen angepasst werden.

4.3.2.3 Mögliche Vorgehensweise für das Internet der Dinge

Der hier vorgeschlagene Entwicklungsprozess für dezentrale Materialflussstrategien ist ebenso wie das herkömmliche Vorgehen nach [VDI-3633] ein iterativer Prozess. Die Steuerungslogik wird schrittweise implementiert, durch eine Emulation überprüft

und für den Fall, dass das gewünschte Ziel nicht erreicht wurde, nochmals überarbeitet. Die Entwicklungsaufgabe lässt sich dabei in zwei Schritte unterteilen:

Der erste Schritt ist die Entwicklung von Grundregeln, im zweiten Schritt werden spezielle Verhaltensregeln für strategische Handlungen definiert (siehe Abb. 4.22).

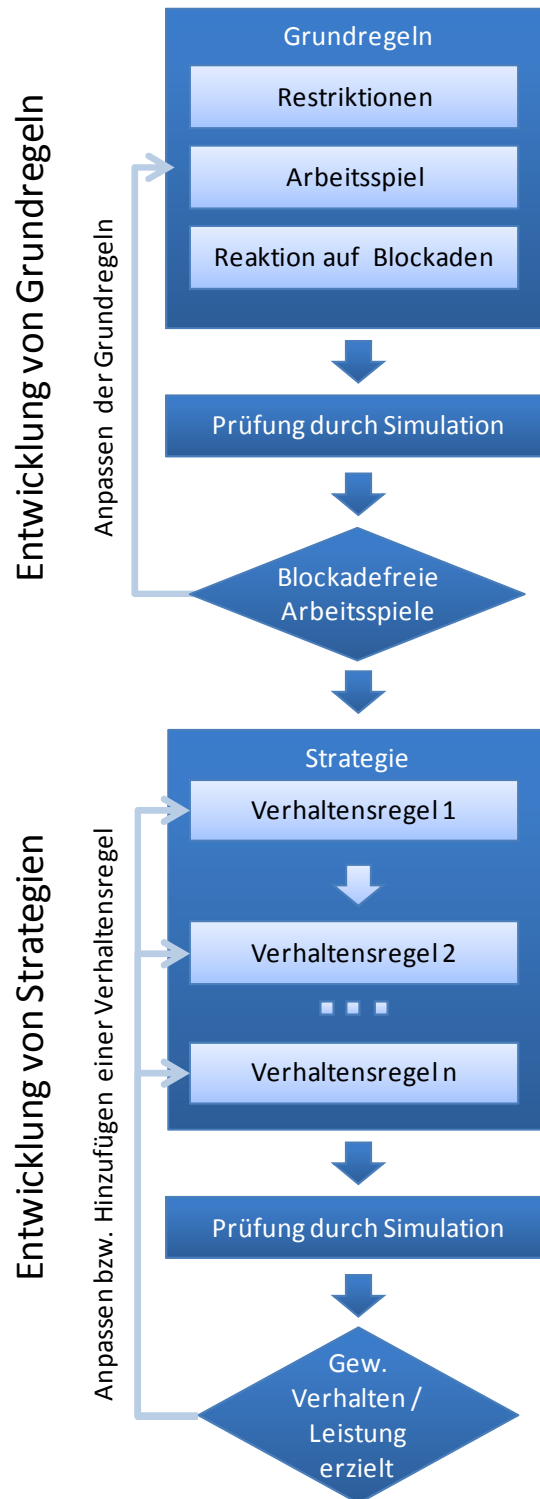


Abb. 4.22: Vorgehensweise bei der Entwicklung von Strategien

Als Grundregeln werden alle Regeln bezeichnet, welche notwendig sind, um das Arbeitsspiel des Moduls festzulegen und seine blockadefreie Abarbeitung zu ermöglichen. Ein Arbeitsspiel ist der Bewegungsablauf, den ein Modul zur Erledigung seiner logistischen Funktion durchzuführen hat. Dabei werden auch Restriktionen eines Moduls in seiner Umgebung berücksichtigt und definiert, z.B. welche Ziele ein Modul anfahren soll, welche Aufträge es übernehmen kann oder welcher Sicherheitsabstand zu anderen Modulen einzuhalten ist. Darüber hinaus legen die Grundregeln auch Handlungen fest, die im Konfliktfall als Reaktion auf eine Blockadesituation durchgeführt werden müssen, um die Blockade aufzulösen oder zu verhindern. Dabei muss beispielsweise entschieden werden, welches Modul einem anderen den Vortritt bei der Abarbeitung eines Auftrags zu lassen hat, ob ein Kompromiss gefunden werden kann oder ob eine gestellte Aufgabe eventuell kooperativ zu lösen ist.

Sind diese Grundregeln in der strategischen Steuerung eines Moduls implementiert, so kann mittels Emulation (siehe Kap. 4.4.1) geprüft werden, ob bereits blockadefreie Arbeitsspiele durchgeführt werden können, oder ob eine weitere Überarbeitung der Regeln notwendig ist. Handelt es sich um Module, welche lediglich einfache logistische Funktionen ausführen, so können eventuell bereits die Grundregeln ausreichend sein, um eine ausreichend leistungsfähige Modulfunktion bereit zu stellen. In diesem Fall ist die Logikentwicklung abgeschlossen.

Normalerweise wird ein Modul, das alleine durch die Grundregeln gesteuert wird, aber noch nicht das gewünschte Verhalten oder eine gute Durchsatzleistung erzielen. Daher sind zusätzlich noch spezielle Verhaltensregeln für strategische Handlungen zu entwickeln. Diese Verhaltensregeln sind sehr stark von der Aufgabe und den Anforderungen abhängig, welche an sie gestellt werden. Sie sind Optimierungsmaßnahmen für das reine Verhalten nach den Grundregeln. Hier werden u.a. proaktive Verhaltensweisen festgelegt, welche bereits im Vorfeld eine Blockadesituation vermeiden. Dies können z.B. Reservierungsmechanismen für gemeinsam genutzte Wege oder Abstimmungsverhandlungen mit anderen Modulen sein. Dabei werden so lange neue Regeln hinzugefügt bzw. vorhandene optimiert, bis das gewünschte Systemverhalten oder eine geforderte Systemleistung erreicht ist. Diese Verhaltensregeln sollten so gestaltet sein, dass sie unabhängig voneinander eingesetzt werden können und einzeln aktivierbar sind. Dies erleichtert einerseits die simulative Über-

prüfung der Effekte einzelner Regeln und führt andererseits zu einer besseren Übertragbarkeit auf unterschiedliche Einsatzfälle.

Im Normalfall kann schon mit wenigen einfachen Regeln eine starke Leistungsverbesserung erzielt werden. Dieser Steigerungseffekt wird jedoch mit jeder zusätzlichen Regel immer schwieriger zu erreichen. Um schließlich bei vielen Regeln überhaupt noch eine Verbesserung zu erzielen, müssen äußerst komplexe Algorithmen entwickelt werden, welche zudem aufwändig zu implementieren sind. Die Suche nach dem optimalen Verhalten könnte daher unbegrenzt viel Zeit in Anspruch nehmen.

Es ist aber davon auszugehen, dass eine praxistaugliche Lösung für eine dezentrale Steuerung mit vertretbarem Aufwand und in einem überschaubaren Zeitraum erstellt werden kann. Denn im Vergleich zur zentralen Strategieimplementierung, welche ganz ähnliche Fragestellungen zu lösen hat, kann diese im Internet der Dinge besser strukturiert und übersichtlicher gestaltet werden.

Sowohl die Grundregeln als auch die strategischen Verhaltensregeln sollten so gestaltet sein, dass sie einfach an eine möglichst große Anzahl von konkreten Einsatzfällen angepasst werden können. Hier bietet es sich z.B. an, mit Kostenfunktionen zu arbeiten, welche über einstellbare Parameter verfügen, die es ermöglichen, das Verhalten an den jeweiligen Einsatzfall anzupassen. Zusätzlich wird die Wiederverwendbarkeit der Implementierung stark erhöht, wenn die Regeln so aufgebaut sind, dass sie unabhängig von der Anzahl der Module in einem System funktionieren.

4.3.2.4 Beispielhafte Strategieentwicklung für einen Querverschiebewagen

Im Folgenden soll am Beispiel mehrerer, auf einer gemeinsamen Schiene fahrender Querverschiebewägen gezeigt werden, dass in diesem Fall eine dezentrale und auf relativ einfachen Regeln aufbauende Steuerung möglich ist.

Ein Querverschiebewagen (QVW) hat die Aufgabe, Transporteinheiten (TE) von einer oder mehreren Ankunftsstrecken auf eine oder mehrere Abtransportstrecken zu befördern. Dazu bewegt er sich auf einer Schiene senkrecht zu den Zuführ-/Abtransportstrecken und verfügt selbst über einen Förderer, welcher die Lastübergabe durchführt.

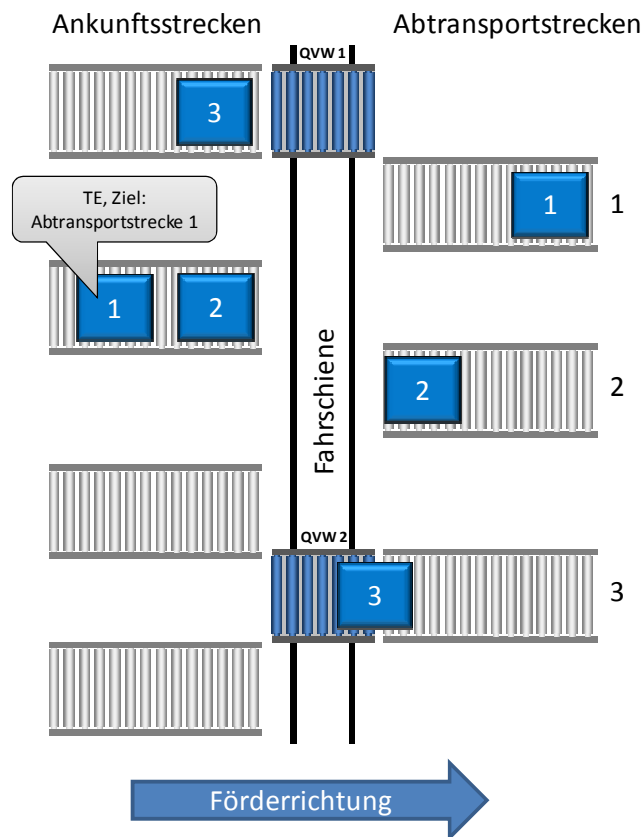


Abb. 4.23: Querverschiebewägen

Liegt ein Transportauftrag vor, so sei das Arbeitsspiel eines QVW wie folgt:

1. Fahrt zur Ankunftsstrecke
2. Lastaufnahme
3. Fahrt zur Abtransportstrecke
4. Lastabgabe

Im konkreten Beispiel soll davon ausgegangen werden, dass die ankommenden Transporteinheiten nach dem LWF-Prinzip (Longest Wait First) weitergefördert werden. Warten also auf verschiedenen Ankunftsstrecken TEs auf ihren Transport, so ist diejenige bevorzugt zu befördern, welche am längsten gewartet hat. Zusätzlich soll immer derjenige freie QVW den Auftrag übernehmen, welcher den kürzesten Anfahrtsweg hat, um so die Transportwege und damit die Transportzeit zu minimieren. Diese Vorgaben führen zu einer ersten Prioritätsregelung:

$$P_{\text{Auftrag}}(TE, QVW) = t_W(TE) - t_{\text{Anfahrt}}(QVW)$$

Die Auftragsdisposition läuft damit wie folgt ab:

Jede Transporteinheit überwacht ihre Wartezeit t_W und meldet diese an die Querverschiebewägen. Diese berechnen ihre Anfahrtszeit t_{Anfahrt} und die Priorität

$P_{\text{Auftrag}}(TE, QVW)$, welche wiederum an die Transporteinheit übermittle wird. Auf diese Weise verfügen zum einen die Transporteinheiten über eine priorisierte Liste von möglichen Transporteuren und zum anderen die Querverschiebewägen über eine Liste von bevorzugten TE. Der QVW kann dann unter den TE, welche sein Gebot angenommen haben, die für ihn günstigste auswählen. Nun wird die TE zum Transport reserviert und der QVW beginnt mit der Startanfahrt.

Während der Startanfahrt kann es bereits zu konkurrierenden Zielen der QVW und dadurch zu Blockadesituationen kommen. Ab jetzt gilt:

$$P_{\text{Fahren}}(QVW) = t_W(TE) + c * t_{QVW}(TE) - \rho(QVW) + R$$

Hierbei fließt neben der Wartezeit $t_W(TE)$ zusätzlich mit Gewichtung $c > 1$ die Zeit ein, welche eine TE bereits auf dem QVW verbringt $t_{QVW}(TE)$. Dies führt dazu, dass ein beladener QVW immer eine höhere Priorität besitzt, als ein unbeladener, so dass ein bereits begonnener Auftrag beendet werden kann, ohne dass ein freier QVW die Ausführung behindert. $\rho(QVW)$ beschreibt die Auslastung eines QVW und sorgt in der Prioritätsgleichung dafür, dass bei einem Patt immer der QVW zum Zuge kommt, welcher bisher die geringere Auslastung hatte, was eine gleichmäßige Auslastung der QVW garantiert. Um den unwahrscheinlichen Fall zu verhindern, dass trotz allem zwei QVW die gleiche Priorität erhalten, geht schließlich eine Zufallszahl R in die Berechnung mit ein. Diese sollte so klein gewählt werden, dass sie zwar eine Pattsituation verhindert, die anderen Faktoren aber nicht überragt.

Tritt nun ein Zielkonflikt auf, so verhält sich jeder QVW nach folgenden Regeln:

- Falls der blockierende QVW eine höhere Priorität hat, fahre auch in dessen Richtung, d.h.:

$$P_{QVW} \leq P_{\text{Blockierer}} \quad \rightarrow \text{In Richtung des Blockierers fahren}$$

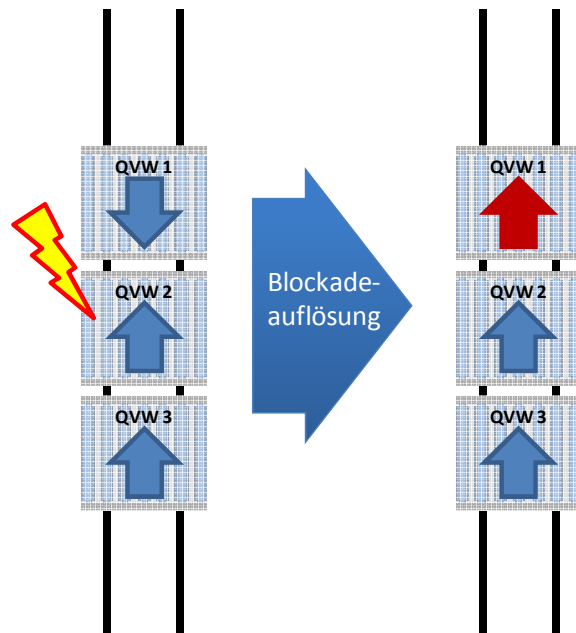
$$P_{QVW} > P_{\text{Blockierer}} \quad \rightarrow \text{Eigene Fahrtrichtung beibehalten}$$

- Wenn einem Blockierer ausgewichen werden muss, so wird die Priorität des Blockierers zur eigenen Priorität addiert, d.h.:

$$P_{\text{Fahren}}(QVW) = t_W(TE) + c * t_{QVW}(TE) - \rho(QVW) + R + P_{\text{Fahren}}(\text{Blockierer})$$

Die zweite Regel führt in größeren Systemen dazu, dass die Priorität quasi von einem Fahrzeug zum nächsten „vererbt“ wird und auf diese Weise eine gewichtete Mehrheitsentscheidung bzgl. der Fahrtrichtung getroffen wird. Ein Beispiel dafür ist in Abb. 4.24 zu sehen. Zwar wäre QVW 1 in der Lage, QVW 2 zu verschieben. Dazu

müsste aber zusätzlich QVW 3 seine geplante Fahrtrichtung ändern. Die beiden QVW 2 und 3 sind aber gemeinsam „stark genug“, um QVW 1 zu überstimmen, so dass dieser ausweichen muss.



Wobei:

$$P_{\text{Zielanfahrt}}(\text{QVW3}) > P_{\text{Zielanfahrt}}(\text{QVW1}) > P_{\text{Zielanfahrt}}(\text{QVW2})$$

$$P_{\text{Zielanfahrt}}(\text{QVW1}) < P_{\text{Zielanfahrt}}(\text{QVW2}) + P_{\text{Zielanfahrt}}(\text{QVW3}) = P_{\text{Blockade}}(\text{QVW1})$$

Abb. 4.24: Blockadeauflösung

Sobald die Blockade beendet ist, bzw. ein Blockierer seinen Auftrag erledigt hat, sinkt dessen Priorität schlagartig, da er nun unbeladen ist. Nun kann der bis dato unterlegene QVW zum Zuge kommen.

Die bisher beschriebenen Prioritäts- und Verhaltensregeln führen zwar schon zu einem blockadefreien Ablauf, jedoch keineswegs zu einem hohen Durchsatz, da ein Konflikt erst aufgelöst wird, wenn die QVW schon auf Kollisionskurs sind. Es ist daher möglich, das bisherige Verhalten noch zu optimieren. Dazu sind verschiedene zusätzliche Verhaltensregeln denkbar, welche sowohl einzeln als auch in Kombination einsetzbar sind:

1. Auftrag bei Blockade stornieren

Falls auf dem Weg zur Ankunftsstrecke ein anderer QVW den Weg blockiert, so soll, falls möglich, ein neuer Auftrag gesucht werden, dessen Startpunkt in Fahrtrichtung des Blockierers liegt. Diese Regel verhindert, dass die Ziele der einzelnen QVW über Kreuz liegen, so dass sie sich nicht unnötig gegenseitig behindern.

2. Anfahrtsweg reservieren

Der Weg zur Ankunftsstrecke wird im Voraus reserviert, so dass andere QVW diesen nur noch in der gleichen Fahrtrichtung befahren dürfen. Diese Regel verfolgt die Idee, Blockaden nicht aufzulösen, sondern sie bereits im Vorfeld zu verhindern.

3. Zielanfahrt reservieren

Der Weg von der Ankunftsstrecke bis zum Zielpunkt wird im Voraus reserviert, so dass andere QVW diesen nur noch in der gleichen Fahrtrichtung befahren dürfen.

4. Auftragsbearbeitungszeit schätzen

Die eingangs beschriebene Auftragsdisposition bezieht ausschließlich freie QVW mit ein. Diejenigen, welche gerade mit der Abarbeitung eines Auftrags beschäftigt sind, sind daran noch nicht beteiligt. Daher kann die Berechnung der Priorität zur Auftragsdisposition $P_{\text{Auftrag}}(TE, QVW)$ so erweitert werden, dass sich auch belegte QVW an der Auftragsvergabe beteiligen können. Die Berechnung der Priorität lautet dann wie folgt:

$$P_{\text{Auftrag}}(TE, QVW) = t_W(TE) - t_{\text{Anfahrt}}(QVW) - t_{\text{Auftragsbearbeitung}}(QVW)$$

Wobei $t_{\text{Auftragsbearbeitung}}$ der Zeit entspricht, die voraussichtlich verstreicht, bis der QVW wieder zur Verfügung steht. Je nachdem, in welcher Phase seines Arbeitsspiels sich dieser befindet, gehen dabei Lastübergabezeiten und/oder noch verbleibende Fahrzeiten in die Berechnung mit ein.

Im Rahmen einer Simulation wurde die Leistungsverbesserung durch diese Strategien anhand eines Systems mit drei QVW, zehn Ankunftsstrecken und acht Abtransportstrecken überprüft. Dazu wurde der Durchsatz für 5000 Transportaufträge beim Betrieb ohne Verhaltensregeln mit demjenigen, der durch Anwendung einer zusätzlichen Verhaltensregel erzielt wird, verglichen (siehe Abb. 4.25).

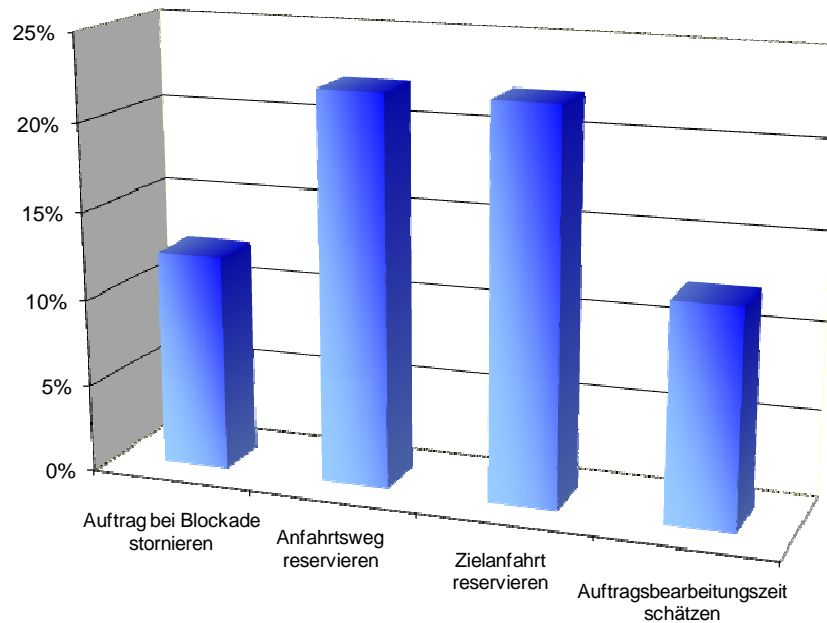


Abb. 4.25: Durchsatzsteigerung durch Anwendung von Verhaltensregeln (einzeln)

Dabei zeigt sich, dass die Verhaltensregeln 2 und 3 jeweils die größten Auswirkungen auf den Durchsatz haben. Damit bestätigt sich, dass diejenigen Regeln, die Blockaden schon im Vorfeld verhindern, am wirkungsvollsten sind.

Anschließend wurde eine kombinierte Anwendung der Verhaltensregeln untersucht. So bewirkt die Kombination von Regel 1 & 2 bereits eine Durchsatzsteigerung von 40%. Alle Regeln zusammen erreichen sogar 54% (siehe Abb. 4.26).

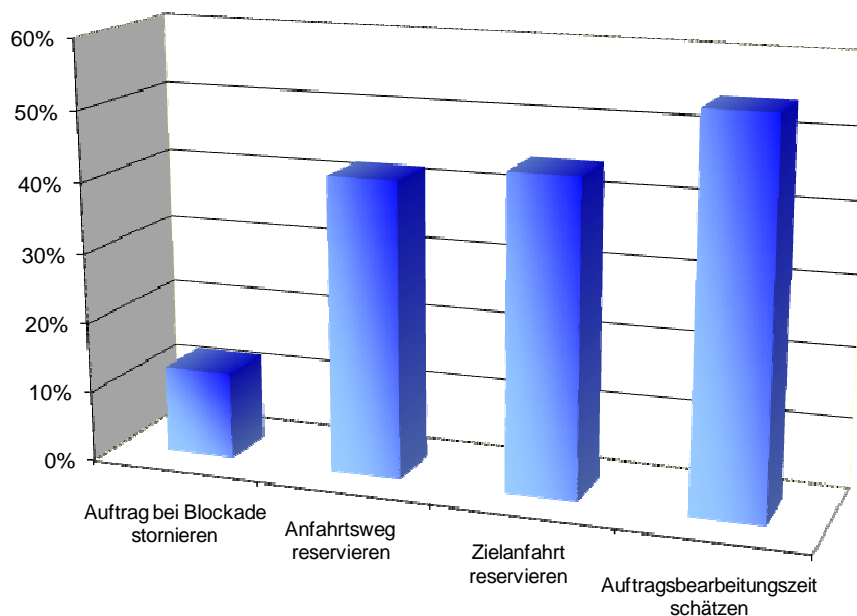


Abb. 4.26: Durchsatzsteigerung durch Anwendung von Verhaltensregeln (kombiniert)

Die durchschnittliche Auslastung der QVW liegt in diesem System, das unter Überlast betrieben wurde (theoretisch unendlicher Stau auf den Ankunftsstrecken), bei ca. 76%. Dieser, im Vergleich zu einem einzeln betriebenen QVW, niedrige Auslastungsgrad ist darauf zurückzuführen, dass sich Zielkonflikte und dadurch bedingte Behinderungen nicht gänzlich vermeiden lassen, wenn drei Fahrzeuge um einen Schienenweg konkurrieren.

Zusammenfassend lässt sich festhalten, dass sich auch in einem dezentral gesteuerten System mit relativ einfachen Mitteln komplexe Verhaltensweisen umsetzen lassen. Darüber hinaus haben dezentral gesteuerte Systeme verglichen mit herkömmlichen Systemen den Vorteil einer übersichtlicheren Programmierung und der problemlosen Erweiterbarkeit. Die vorgestellten Regeln sind unabhängig von der Anzahl der QVW und der Ankunfts- bzw. Abtransportstrecken im System. Die Implementierung der Steuerungslogik ist damit nicht nur auf unterschiedliche Einsatzszenarien von Querverschiebewägen übertragbar, sondern ist auch für die Ansteuerung von Regalbediengeräten oder Elektrohängebahnkatzen einsetzbar.

4.3.3 Konfigurierung / Parametrierung

Das Konzept des Internet der Dinge ist ein Ansatz, in dessen Fokus die Wiederverwendbarkeit der entwickelten Lösungen für möglichst viele Anlagenrealisierungen steht. Um in der Softwareentwicklung eine gute Wiederverwendbarkeit zu erzeugen, sind die Konfigurierbarkeit bzw. Parametrierbarkeit von Software wichtige Konzepte, denn diese erlauben es, ein Programm an einen neuen Einsatzfall anzupassen, ohne dass das Programm selbst geändert werden muss. Hierbei wird zwischen der Offline- und der Online-Parametrierung unterschieden.

Offline-Parametrierung

Der Begriff Offline-Parametrierung bzw. Konfiguration beschreibt in der vorliegenden Arbeit die Anpassung eines Programms an den jeweiligen Einsatzfall, noch bevor das Programm gestartet wird. Dazu werden die Initialwerte von Programmvariablen verändert oder bestimmte Programmbausteine speziell für die konkrete Aufgabe zusammen gestellt. Die Offline-Parametrierung ist also eine Vorgehensweise zur Anpassung von Steuerungssoftware an die statischen Eigenschaften eines Moduls.

Anwendungsgebiete sind:

- Mechanische Ausprägung

Es bietet sich an, die mechanische Ausprägung eines Moduls in der Standardgerätekategorie (siehe Kap. 4.3.1) so abzubilden, dass sie weitgehend parametrisierbar ist. Zum Beispiel sollten die Länge eines Rollenförderers oder die Anzahl und Positionen seiner Stau- bzw. Übergabeplätze über Variablenwerte einstellbar sein.

- Strategieregeln

Die Strategieregeln, welche das Verhalten eines Moduls implementieren, müssen einzeln über Parameter aktivierbar sein (siehe Kap. 4.3.2). Zudem sollten deren Kostenfunktionen so anzupassen sein, dass auf die Verhaltensweise eines Moduls Einfluss genommen werden kann. So kann beispielsweise die Anfahrhäufigkeit eines Querverschiebewagens zu bestimmten Zuführstrecken mittels Parametern an die zu erwartende Ankunftsrate auf den jeweiligen Strecken angepasst werden.

- Anpassung an unterschiedliche Steuerungshardware bzw. Emulation

Die Schnittstelle zwischen der strategischen und der maschinennahen Steuerung sollte so gestaltet sein, dass sie möglichst einfach an unterschiedlichste Geräte angepasst werden kann (siehe Kap. 4.3.1). Dazu wird eine Middlewareschicht vorgeschlagen, welche eine Übersetzung der Befehle an die Maschinensteuerungsebene vornimmt, und deren Rückmeldungen an die Gerätekategorie weiterleitet.

Die Middleware sollte so aufgebaut werden, dass der dabei entstehende Entwicklungsaufwand nur einmal für jeden Typ von Steuerungshardware zu investieren ist. Dies ist dadurch zu erreichen, dass sie unabhängig von der konkreten Steuerungsaufgabe entwickelt und lediglich mittels Parametrierung angepasst wird. So wird sich der Zugriff auf die internen Variablen beispielsweise einer Siemens-S7-Steuerung unabhängig von der Steuerungsaufgabe immer gleich darstellen. Auf diese Weise ist es ausreichend, in einer Konfigurationsdatei mitzuteilen, dass es sich um diesen Typ der Steuerungshardware handelt und auf welche Variablenzustände zugegriffen werden soll.

Die Middleware sollte zudem so gestaltet sein, dass sie nicht nur die Anbindung eines konkreten Geräts an die operative Steuerung regelt. Sie sollte es ermöglichen, die reale Steuerung gegen eine Emulation derselben auszutauschen, ohne dass dazu die Steuerungsprogramme geändert werden müssen. In diesem Fall kann die Konfigurationsdatei so angepasst werden, dass die Middleware ihre Nachrichten an einen Emulator weiterleitet und von diesem die Rückmeldungen entgegen nimmt (siehe Kap. 4.4.1). Die Kommunikation zwischen strategischer und operativer Steuerung kann beispielsweise über eine TCP/IP-Schnittstelle erfolgen, welche für aktuelle Steuerungshardware inzwischen Standard ist (siehe Kap. 2.3.1). Je nach Ausführung der Steuerungshardware besteht hier meist die Möglichkeit, über ein herstellerspezifisches Protokoll direkt auf Variablenzustände innerhalb der SPS zuzugreifen. Es ist aber auch denkbar, das Kommunikationsprotokoll auf Basis von TCP/IP selbst zu entwickeln, und in der operativen Steuerung einen entsprechenden Interpreter zu implementieren.

Online-Parametrierung

Unter Online-Parametrierung ist in diesem Zusammenhang die Anpassung eines Programms während seiner Abarbeitung zu verstehen, indem Parameter zur Laufzeit verändert werden, um auf die aktuelle Situation zu reagieren. Zu diesem Zweck bietet es sich an, für jedes Modul eine Web-Schnittstelle zu implementieren, welche eine komfortable Veränderung dieser Parameter gestattet.

Anwendungsgebiete für die Online-Parametrierung sind:

- Wechsel der Betriebsart und Strategieänderung
Der Anlagenbetreiber soll in der Lage sein, auch während des Betriebs in die Arbeitsweise eines Moduls einzugreifen. Dies ist dann nötig, wenn entweder die Betriebsart eines Moduls z.B. von manuellem auf automatischen Betrieb gewechselt, oder wenn bewusst ein bestimmtes Verhalten hervorgerufen werden soll. Neben der Offline-Anpassung der Strategieregeln vor dem ersten Anlagenanlauf kann es notwendig sein, diese Strategien auch Online zu ändern. Obwohl die Module genügend Intelligenz aufweisen müssen, um auf Staus, Störungen und eine veränderte Systemauslastung selbständig zu reagieren,

soll der Bediener auch in Eigenregie auf diese Vorfälle reagieren können. Ein Beispiel dafür ist die Initiierung einer speziellen Kommissionier- oder Sortierstrategie.

- **Umweltmodell**

Das Umweltmodell eines Moduls beschreibt unter anderem die Anlagentopologie. Auf Grundlage der Topologie wird berechnet, welche Wege einer TE im System zur Verfügung stehen. So kann beispielsweise durch eine Veränderung der Topologie dafür gesorgt werden, dass ein bestimmter Anlagenbereich nicht mehr angefahren wird, um Wartungsarbeiten durchzuführen. Dieser Eingriff wird mit Hilfe der Visualisierungsumgebung vorgenommen, welche in Kap. 4.4.2 beschrieben ist.

4.4 Funktionsabsicherung

Die Entwicklung von Fördertechnikmodulen für das Internet der Dinge ist, wie in Kap. 4.1 beschrieben ein iterativer Prozess, bei welchem die einzelnen Entwicklungsschritte regelmäßig auf ihre Funktionserfüllung überprüft werden müssen. Zur Funktionsabsicherung werden dabei verschiedene Verfahren eingesetzt. In den folgenden Kapiteln werden die Emulation, die Visualisierung und Methoden zu Fehlersuche und Fehlerbehebung beschrieben.

Die Emulation dient der Überprüfung der strategischen Steuerungsprogrammierung und ist ein Hilfsmittel zur Analyse des Anlagenverhaltens und des Anlagendurchsatzes, noch bevor die Anlage physisch existiert. Sowohl in der Planungsphase als auch beim späteren Betrieb der Anlage muss ein Visualisierungswerkzeug vorhanden sein, das den aktuellen Anlagenzustand aufzeigt und dem Benutzer die Möglichkeit gibt, das Geschehen zu analysieren, bzw. durch eigene Vorgaben einzugreifen. Diese Visualisierung wird in Kap. 4.4.2 beschrieben. In Kap. 4.4.3 werden Methoden aufgezeigt, die Fehlersuche und Fehlerbehebung in einem dezentralen System ermöglichen.

4.4.1 Emulation

Bei der Emulation handelt es sich um ein Teilgebiet der Simulation. Nach [VDI-3633] ist die Simulation von Logistiksystemen ein Verfahren, um ein dynamisches System

in einem experimentierbaren Modell nachzubilden. Die bei den Experimenten mit diesem Modell entstehenden Erkenntnisse sollen dabei anschließend auf die Wirklichkeit übertragbar sein.

Die Simulation wird bei der Realisierung eines Materialflusssystems laut [Dan-09] in verschiedenen Lebenszyklusphasen und zu unterschiedlichen Zwecken eingesetzt:

Planung

- Verbesserung vorhandener Anlagen (Redesign, Tuning)
- Überprüfung neugeplanter Anlagenkonzepte
- Entwurf von Steuerungsstrategien
- Bewertung von Alternativen

Realisierung

- Entwicklung und Test von Steuerungssoftware
- Ermittlung des Anlaufverhaltens der Anlage
- Mitarbeiterschulung

Betrieb

- Vergleichende Untersuchung von Strategien und Ablaufvarianten
- Reihenfolgeoptimierung von Prozessabläufen
- Strategien bei Störfällen
- Mitarbeiterschulung

Im Internet der Dinge soll von der Simulation möglichst umfangreich und so früh wie möglich Gebrauch gemacht werden, da auf diese Weise Fehler früher entdeckt und so die Fehlerbehebungskosten gesenkt werden können.

Bei einer Emulation handelt es sich um ein Simulationsmodell, das zum Teil mit realen Funktionskomponenten gekoppelt ist. Im Internet der Dinge sollen dazu der mechanische Aufbau und die operative Steuerung durch simulierte Komponenten ersetzt werden. Die Emulation ermöglicht es also, z.B. die reale Steuerungssoftware zu prüfen, ohne dass dazu ein Modul physisch vorhanden sein muss, und erlaubt daher eine Parallelisierung von Software- und Hardwareentwicklung bzw. -aufbau.

Die Anwendung der Emulation wird dadurch begünstigt, dass es innerhalb des Moduls eine klare Trennung zwischen maschinennaher und strategischer Steuerung

gibt. Diese beiden Ebenen werden durch eine Middleware verbunden, welche sowohl mit der operativen Steuerung als auch mit einem Emulator kommunizieren kann (siehe Kap. 4.2.1). Die maschinennahe Steuerungsebene und die mechanische Ebene werden dabei gegen eine Emulationssoftware ausgetauscht, welche deren Funktionen nachahmt (siehe Abb. 4.27).

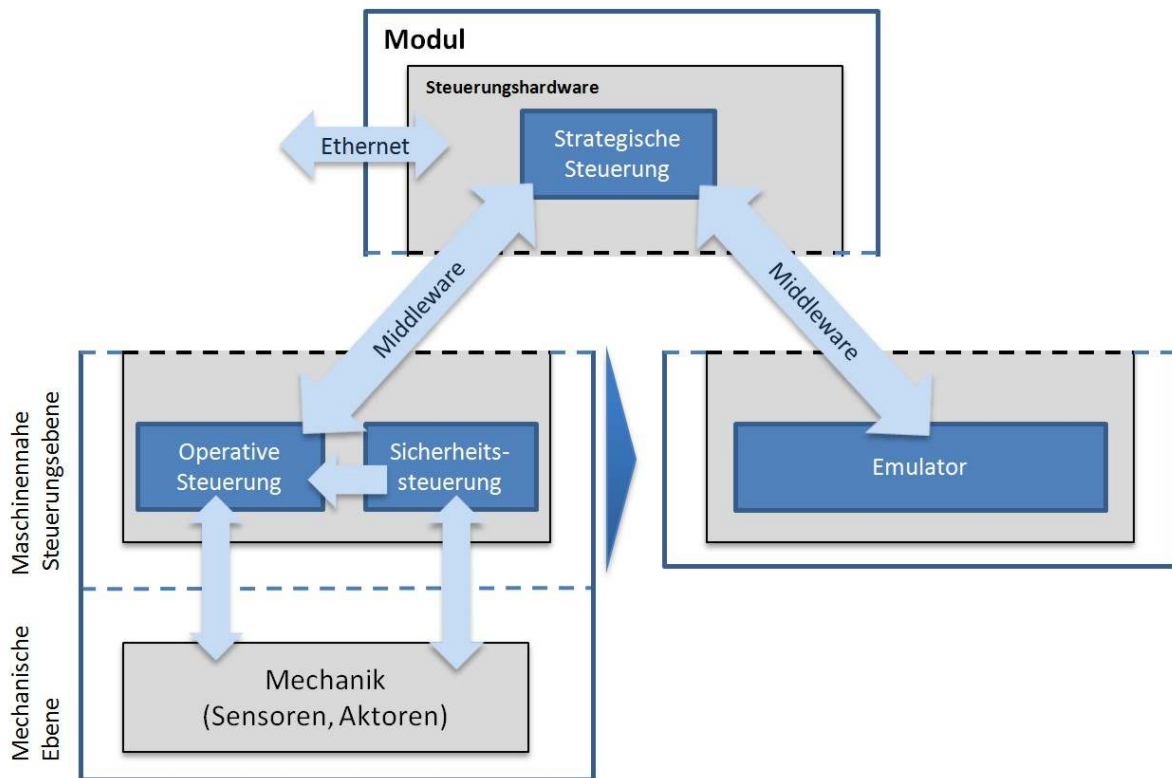


Abb. 4.27: Emulation der Modulhardware

Dabei spielt es aus Sicht der strategischen Steuerung keine Rolle, ob sie ihre Anweisungen an die real vorhandene Maschinensteuerung gibt und von dieser Rückmeldungen erhält, oder ob an deren Stelle eine Emulationssoftware eingesetzt wird.

Die strategische Steuerungsimplementierung kann daher sowohl zur Steuerung der realen Anlage als auch zur Simulation eingesetzt werden. Damit vereinfacht sich die herkömmliche Vorgehensweise, bei welcher erst die Steuerungslogik in einem Simulationsmodell abgebildet wird, um sie anschließend in der realen Steuerung zu implementieren. Es ist sogar möglich, einen Mischbetrieb aus virtuellen und realen Modulen zu realisieren, da alle Entitäten im System lediglich auf der Ebene der strategischen Steuerung kommunizieren und somit eine Unterscheidung eines realen von einem virtuellen Modul steuerungstechnisch nicht möglich ist. So können die Auswirkungen einer geplanten Erweiterung des Materialflusssystems und die dafür

notwendige Steuerungsprogrammierung geprüft werden, noch bevor der physische Umbau in Angriff genommen wird.

Je nachdem, welche Fragestellung eine Simulation beantworten soll, muss auch die Emulationssoftware so gestaltet werden, dass sie dazu entsprechende Antworten liefern kann. Hierbei sind zwei Fragestellungen von Bedeutung:

Korrektheit der Steuerungslogik /-implementierung

Die Emulation wird in diesem Fall dazu eingesetzt, um die Logik, Kommunikation und Fehlerfreiheit der strategischen Steuerung zu überprüfen. Dazu müssen in der Emulatorsoftware zum einen die Kommunikationsschnittstelle zur strategischen Steuerung und zum anderen ein einfaches Modell des aktuellen Zustands, in dem sich ein Modul befindet, nachgebildet werden. Letzteres umfasst beispielsweise eine virtuelle Position, an der sich ein Verschiebewagen auf der Schiene befindet bzw. die Tätigkeit, welche dieser aktuell in der Simulation ausführt. Eine derartig gestaltete Emulation eignet sich bereits, um Verhaltensstrategien beispielsweise auf Blockadefreiheit zu untersuchen (siehe Kap. 4.3.2) bzw. die Fehlerfreiheit des strategischen Steuerungsprogramms zu überprüfen. Die Nachbildung von Geschwindigkeiten oder Bearbeitungszeiten ist hierfür noch nicht notwendig.

Manche Module verlangen eine direkte Benutzerinteraktion. Verfügt beispielsweise ein Kommissionierarbeitsplatz über eine Benutzerschnittstelle, welche es dem Kommissionierer ermöglicht, Auftragsdaten abzurufen oder sich vor dem Verlassen des Arbeitsplatzes abzumelden, so kann die hierzu erforderliche Schnittstelle ebenfalls im Emulator nachgebildet werden. Auf diese Weise können z.B. erste Schulungen der Anlagenbediener durchgeführt werden.

Überprüfung der Modul- und Systemleistung

Eine wichtige Aufgabe einer Simulation ist die Bestimmung der im System erreichbaren Leistung bzw. die Erkennung von Engpässen. Sollen diese Fragen untersucht werden, so muss der Emulator auch das zeitliche Verhalten der Mechanik eines Moduls nachbilden. Gibt die strategische Steuerung also einen Arbeitsschritt wie z.B. die Zielfahrt eines Verschiebewagens zu einer bestimmten Förderstrecke in Auftrag, so berechnet der Emulator die Zeitspanne, welche der echte Verschiebewagen für diese Wegstrecke benötigen würde. Dabei wird in der Emulationssoftware festgelegt, ob diese Zeitspanne tatsächlich abgewartet werden soll, oder ob der Planer eine

beschleunigte Simulation durchführen will. Das Verhältnis von Realzeit zu Simulationszeit kann dabei skalierbar gestaltet werden, wobei die maximale Simulationssgeschwindigkeit von der Rechengeschwindigkeit der Ausführungsplattform abhängig ist. Je mehr Erfahrungswerte und Berechnungsverfahren für die Zeitabschätzung eines Arbeitsschrittes vorhanden sind, desto genauer kann z.B. eine Durchsatzanalyse durchgeführt werden. Daher ist es hilfreich, beim Betrieb realer Anlagen so viele Daten wie möglich zu sammeln, indem die operative Steuerung automatisch Durchschnittszeiten von erledigten Aufträgen speichert. Diese Informationen können dann dem Emulator über eine Konfigurationsdatei zur Verfügung gestellt werden und erhöhen so die Genauigkeit zukünftiger Simulationen.

Es bietet sich an, die Emulationssoftware so zu gestalten, dass die Berechnung der Zeitabschätzung deaktivierbar ist. Auf diese Weise kann bei Fragestellungen, bei welchen die zeitlichen Aspekte keine Rolle spielen oder keine verlässlichen Daten aus realen Anlagen vorhanden sind, auf die entsprechende Berechnung verzichtet werden.

4.4.2 Visualisierung

Eine Anlagenvisualisierung ist ein wichtiges Hilfsmittel zur Überwachung des ordnungsgemäßen Systemverhaltens und ist daher unverzichtbarer Bestandteil jedes Materialflusssystemes. Sie wird ausschließlich als Softwareprogramm implementiert und ist daher im Sinne des Internet der Dinge ein Dienst (siehe Kap. 3.2.3). Die Visualisierung wird sowohl von den Betreibern als auch von den Entwicklern genutzt und bietet folgende Funktionen:

Darstellung des aktuellen Anlagenzustands

In erster Linie dient die Visualisierung der übersichtlichen Aufbereitung des aktuellen Anlagenzustands. Hierzu wird die Anlagentopologie dargestellt und beispielsweise anhand von Farbkodierungen die Auslastung der verschiedenen Streckenabschnitte angezeigt. Damit lassen sich Staus bzw. eine ungleichmäßige Auslastung von Anlagenteilen schnell erkennen und, falls notwendig, Gegenmaßnahmen einleiten. Zudem können in einer Visualisierungsumgebung umfangreiche Zusatzinformationen abgerufen werden. Das sind z.B. Auftragslisten, Standorte der TE, Anlagendurchsätze oder Lagerfüllungsgrade.

In Bezug auf ein Fördertechnikmodul müssen zusätzlich folgende Informationen verfügbar gemacht werden:

- aktueller Auftrag und dessen Bearbeitungsstatus
- aktuelle Betriebsart
- eigene Belegung
- aktuell ausgeführte Materialflussstrategie
- Störungsmeldungen
- Nachschubmeldungen
- Status der maschinennahen und strategischen Steuerung
- Zustände wichtiger Ein- / Ausgänge bzw. Motoren

Darstellung / Aufzeichnung von Statistiken

Statistiken dienen der Beurteilung der Anlagenleistung über einen längeren Zeitraum. Sinnvoll ist es, z.B. folgende Daten zu erfassen:

- Auslastung von Anlagenteilen und Fördertechnikmodulen
- Ausfälle von Modulen bedingt durch Störungen und Wartung
- Liege- und Durchlaufzeiten der TE
- Durchschnittliche Auftragsbearbeitungszeit
- Zeitliche Verteilung der eingelasteten Aufträge und ihrer Ziele
- Lagerfüllungsgrade

Diese Daten müssen in einer Datenbank erfasst und dem Benutzer beispielsweise in Linien-, Balken- oder Tortendiagrammen übersichtlich dargestellt werden.

Eingriff in ablaufende Prozesse bzw. Strategieänderungen

Neben der Darstellung aufbereiteter Informationen soll die Visualisierung dem Benutzer auch gestatten, Einfluss auf das Verhalten der Materialflussanlage zu nehmen. Folgende Veränderungen sollten direkt in der Visualisierungsumgebung ausgelöst werden können:

- Änderung von Materialflussstrategien
- Umstellung des Betriebszustands der Module, z.B. von Handbetrieb auf automatisch und umgekehrt
- Sperrung von Streckenabschnitten bzw. Anlagenbereichen, um Wartungsarbeiten und Reparaturen auszuführen

- Umleitung von TE an einen neuen Ort, beispielsweise, um auf Staus oder unerwartete Systemlasten zu reagieren
- Einlasten von Transportaufträgen, um z.B. eine Lagerreorganisation vorzunehmen
- Absetzen von direkten Befehlen an bestimmte Fördertechnikmodule, um deren ordnungsgemäße Funktion der Module zu prüfen. Daher sollen z.B. Weichen geschaltet oder Regalbediengeräte in eine bestimmte Position verfahren werden können

Anforderungen

Auch die Visualisierungsumgebung sollte so gestaltet werden, dass die beschriebenen Funktionen bei möglichst vielen unterschiedlichen Materialflusssystemen realisiert werden können, ohne dass dazu eine Neuprogrammierung notwendig ist. Dazu bietet es sich an, die Darstellung der Topologie, die angezeigten Informationen und die Möglichkeiten zur Benutzerinteraktion generisch auszuführen. Die Topologiedarstellung sollte beispielsweise mit einem Editor aus vorgefertigten Baukastenelementen schnell zusammenstellbar und mittels einfacher Verhaltensregeln konfigurierbar sein. Diese Verhaltensregeln geben an, welche Informationen dargestellt und wie diese kodiert werden sollen. Dabei kann festgelegt werden, dass ein Streckenabschnitt ab einem bestimmten Durchsatz in rot dargestellt werden soll, oder eine EHB-Katze sich entlang einer bestimmten Route bewegen kann.

Auch die Aufzeichnung von Statistiken muss individuell an den jeweiligen Einsatzfall angepasst werden können. Daher sind Funktionen vorzusehen, um Diagramme zu erstellen und die darin dargestellten Informationen frei festzulegen.

Es bietet sich an, Informationen und Eingriffsmöglichkeiten, welche sich auf ein konkretes Modul beziehen, nicht innerhalb der Visualisierungsumgebung des gesamten Systems zu implementieren, sondern lediglich bei Bedarf einzublenden, wenn der Benutzer das entsprechende Modul auswählt. Dies kann beispielsweise über eine Webschnittstelle des Moduls geschehen, so dass jedes Modul seine eigene Visualisierung und spezielle Eingriffsmöglichkeiten bereit stellt. Damit wird diese Visualisierung dem Modul eindeutig zugeordnet und die Entwicklung dieser Schnittstelle ist zusammen mit der restlichen Modulentwicklung projektunabhängig durchführbar.

Insgesamt soll die Systemvisualisierung webbasiert ausgeführt werden, so dass die Experten des Anlagenerbauers im Fehlerfall einen schnellen Zugriff auf alle relevanten Informationen erhalten. Indem die zugrunde liegenden Informationen getrennt von der Visualisierungsumgebung, in einer Datenbank, vorgehalten werden, kann verhindert werden, dass ein Fehler in der Darstellungsprogrammierung Auswirkungen auf den Betrieb des Materialflusssystems hat.

4.4.3 Fehlersuche und -behebung

Die Fehlersuche und -behebung im Internet der Dinge lässt sich in drei Aufgabenbereiche unterteilen:

- Konzepte, um eine fehlerhafte Programmierung eines strategischen oder operativen Steuerungsprogramms zu behandeln.
- Nachvollziehen des Nachrichtenverkehrs, um z.B. die ordnungsgemäße Abarbeitung von verteilten Koordinationsstrategien zu verfolgen.
- Methoden, um die ordnungsgemäße Verkabelung und die einwandfreie Funktion der an die operative Steuerung angeschlossenen Sensoren und Aktoren zu prüfen.

Diese drei Aufgabenbereiche werden im Folgenden erläutert:

Debugging eines Steuerungsprogramms

Die Entwicklung der strategischen Steuerung kann parallel zu derjenigen der operativen Steuerung durchgeführt werden (s. Kap. 4.1.2). Zur Überprüfung der ordnungsgemäßen Funktion der strategischen Steuerung wird ein Emulator eingesetzt (s. Kap. 4.4.1). Auf diese Weise kann die strategische Steuerung unabhängig von der realen Anlage getestet werden.

Die operative Steuerung sollte über eine Benutzerschnittstelle verfügen, welche zum einen den Handbetrieb eines Moduls und zum anderen die Aktivierung von Funktionen gestattet, welche von der strategischen Steuerung angesprochen werden. So sollten beispielsweise Teile eines Arbeitsspiels getrennt von der strategischen Steuerung ausführbar sein, um die ordnungsgemäße Funktion des mechanischen Aufbaus eines Moduls und die Programmierung seiner operativen Steuerung zu prüfen.

Wie in Kap. 4.2.2 beschrieben, wird die strategische Steuerung in einer Hochsprache wie z.B. C++ oder C# implementiert, während die operative Steuerung in einer echtzeitfähigen Umgebung mittels Maschinensteuerungssprachen nach IEC-61131-3 umgesetzt wird. Für beide Steuerungsarten stehen Programmierumgebungen zur Verfügung, welche umfangreiche Hilfsmittel zum Debugging bereit stellen.

Die Programmierumgebungen von Hochsprachen bieten an, das Programm zur Fehlersuche auf eine spezielle Weise zu übersetzen, so dass auch zur Laufzeit Informationen über Variablennamen und -inhalte verfügbar gemacht werden können. Dies ist bei Maschinensteuerungsprogrammen sogar während des Betriebs und ohne eine spezielle Übersetzung möglich, so dass das Programm zur Fehlersuche nicht angehalten werden muss. Für beide Ausprägungen müssen dann folgende zwei Mechanismen zur Verfügung stehen, um ein Programm auf Fehler zu überprüfen, bzw. die Abarbeitung für den Programmierer nachvollziehbar zu machen:

- Steuerung des Programmablaufs

In vielen Programmierumgebungen kann der Programmablauf schrittweise abgearbeitet werden, indem entweder nur der nächste Befehl oder ein Unterprogramm bis zum Rücksprung ausgeführt wird. Zudem kann ein Programm an einem vorher definierten Haltepunkt oder bei Erfüllung einer bestimmten Bedingung angehalten werden. Beide Methoden dienen dazu, dem Programmierer eine schrittweise Abarbeitung der Programmlogik zu ermöglichen, denn normalerweise wird diese so schnell ausgeführt, dass ein Mensch nicht rechtzeitig auf einen Fehler reagieren kann.

Die einzelnen Steuerungsprogramme im Internet der Dinge lassen sich zwar besser überblicken als herkömmliche Programme, da die klaren Zuständigkeiten der einzelnen Module eine bessere Strukturierung erlauben. Trotzdem bedeutet die Verteilung der Steuerungsaufgabe auf viele Entitäten eine besondere Herausforderung, da nun viele Prozesse parallel abgearbeitet werden, welche früher sequenziell gestaltet waren. Aus diesem Grund sollte ein Programmierer dabei unterstützt werden, die zeitliche Abfolge von Entscheidungen, welche in unterschiedlichen Entitäten getroffen werden, nachzuvollziehen. An dieser Stelle stoßen herkömmliche Programmierumgebungen noch an ihre Grenzen, denn diese

erlauben lediglich die Untersuchung einer einzigen Entität, während die anderen Entitäten ihre Arbeit fortsetzen.

Eine Programmierumgebung für die strategische Steuerungsprogrammierung im Internet der Dinge sollte in Zukunft so gestaltet sein, dass bei der schrittweisen Abarbeitung des Programms nicht nur Sprünge von Unterprogramm zu Unterprogramm innerhalb einer Entität verfolgt werden können, sondern auch Entscheidungen quer über die verschiedenen Entitäten. So ist einerseits eine Funktion hilfreich, mit welcher das gesamte System angehalten und neu gestartet werden kann. Andererseits muss es dem Programmierer möglich sein, alle Entitäten gleichzeitig in einen Einzelschrittmodus zu versetzen, so dass das Gesamtsystemverhalten schrittweise analysiert werden kann. Zudem ist denkbar, ein von einer Entität ausgelöstes Ereignis mit einer eindeutigen Nummer zu versehen, und anhand dieser Nummer die Auswirkungen des Ereignisses über verschiedene Entitäten hinweg zu beobachten.

Die operative Steuerung verfügt nur über eine einfache Schnittstelle zu anderen Modulen, indem sie mit der Sicherheitssteuerung kommuniziert, welche wiederum an den Notauskreis angeschlossen ist (s. Kap. 4.2.1). Daher reicht es aus, auf die herkömmliche Programmierumgebung für Maschinensteuerungssprachen zurückzugreifen.

- Inspizieren und Modifizieren von Daten

Wie bereits erwähnt kann während der Ausführung eines Maschinensteuerungsprogramms auf die aktuellen Inhalte von Programmvariablen bzw. Speicherzustände zugegriffen werden. Bei den klassischen Hochsprachen muss dazu das Programm beispielsweise an einem Haltepunkt gestoppt werden. Der Programmierer kann dann auf aktuelle Inhalte zugreifen, indem er z.B. eine Liste von Variablen definiert, welche ihn speziell interessieren, und sich dann ihre aktuellen Werte anzeigen lassen.

In diesem Zusammenhang bieten die Maschinensteuerungssprachen zusätzlich die Funktion, Programmvariablen während der Laufzeit zu verändern. Auf diese Weise kann z.B. ein Gerät, das einen Endschalter überfahren hat und sich daher im Notaus befindet, dazu gezwungen werden, seinen Antrieb trotzdem zu aktivieren und den Bereich des Endschalters zu verlassen.

Überprüfung und Nachvollziehen der Kommunikation

Gerade in einem dezentral gesteuerten System spielt der Nachrichtenverkehr zwischen den Entitäten eine wichtige Rolle. Diese Kommunikation wird beispielsweise zur Auftragsdisposition und auch zur Koordination der Module untereinander eingesetzt. Ein Programmierer, welcher die ordnungsgemäße Ausführung einer Strategieimplementierung prüfen will, muss also auch den Nachrichtenverkehr genau analysieren. Daher sollte es spezielle Werkzeuge geben, welche sowohl die Nachrichteninhalte strukturiert darstellen als auch die einzelnen Nachrichten in ihrer zeitlichen Abfolge so aufbereiten können, dass sie leicht nachvollziehbar sind. Die Fülle der in einem dezentralen System ausgetauschten Nachrichten macht zudem Filtermechanismen notwendig, mit welchen nur bestimmte Kommunikationsteilnehmer speziell beobachtet werden, während andere vernachlässigt werden.

Dabei ist es nicht ausreichend, lediglich den aktuellen Nachrichtenverkehr zu betrachten. Häufig werden die Auswirkungen eines Fehlers erst einige Zeit nach dessen Auftreten sichtbar. Es muss also auch im Nachhinein nachvollzogen werden können, wie ein Fehler zustande kam. Zu diesem Zweck muss die für eine spätere Fehlersuche relevante Kommunikation aufgezeichnet und den jeweiligen Entscheidungen und Handlungen zugeordnet werden können. Dies kann sowohl dezentral in den einzelnen Modulen erfolgen als auch mittels eines eigenen Aufzeichnungsdiens-tes, der die Daten sammelt und in Logdateien ablegt. Auch hier muss, genauso wie bei anderen Ereignissen, sowohl eine zeitliche als auch eine logische Zuordnung der Nachrichten zu bestimmten Vorgängen möglich sein.

Überprüfung der Verkabelung bzw. Sensoren und Aktoren

Die Sensoren und Aktoren eines Moduls werden an Ein-/Ausgangsbausteine einer SPS angeschlossen (siehe Kap. 4.2.1). Der Zustand dieser Eingänge und Ausgänge kann meist direkt an den entsprechenden Baugruppen der SPS mittels Leuchtdioden überprüft werden. Zudem bietet eine Programmierumgebung für Maschinensteuerungssprachen auch Funktionen, um diesen Zustand zu überprüfen und zu verändern. Im Internet der Dinge wird die Verkabelung einfacher zu gestalten sein als bisher, da statt einer SPS, welche einen ganzen Anlagenbereich steuert, nun für jedes Modul eine SPS eingesetzt wird. Damit werden die Verkabelungswege kürzer und die Verkabelung der jeweiligen Steuerung wird übersichtlicher.

5 Realisierung am Lehrstuhl für Fördertechnik Materialfluss Logistik

Die in der vorliegenden Arbeit beschriebenen Konzepte wurden am Lehrstuhl für Fördertechnik Materialfluss Logistik fml der Technischen Universität München anhand eines Elektrohängebahnsystems mit angeschlossener Stetigfördertechnik umgesetzt. Diese Umsetzung wird im Folgenden beschrieben.

5.1 Fördertechnische Einrichtungen und Layout



Abb. 5.1: Versuchshalle des Lehrstuhls fml

Die Versuchshalle des Lehrstuhls fml (siehe Abb. 5.1) verfügt über verschiedene fördertechnische Einrichtungen, welche sich sehr gut dafür eignen, übliche Problemstellungen bei der Steuerung eines Materialflusssystems darzustellen und die Konzepte des Internet der Dinge in der realen Welt zu überprüfen.

Eine schematische Übersicht der fördertechnischen Einrichtungen in der Versuchshalle des Lehrstuhls fml sind in Abb. 5.2 zu sehen.

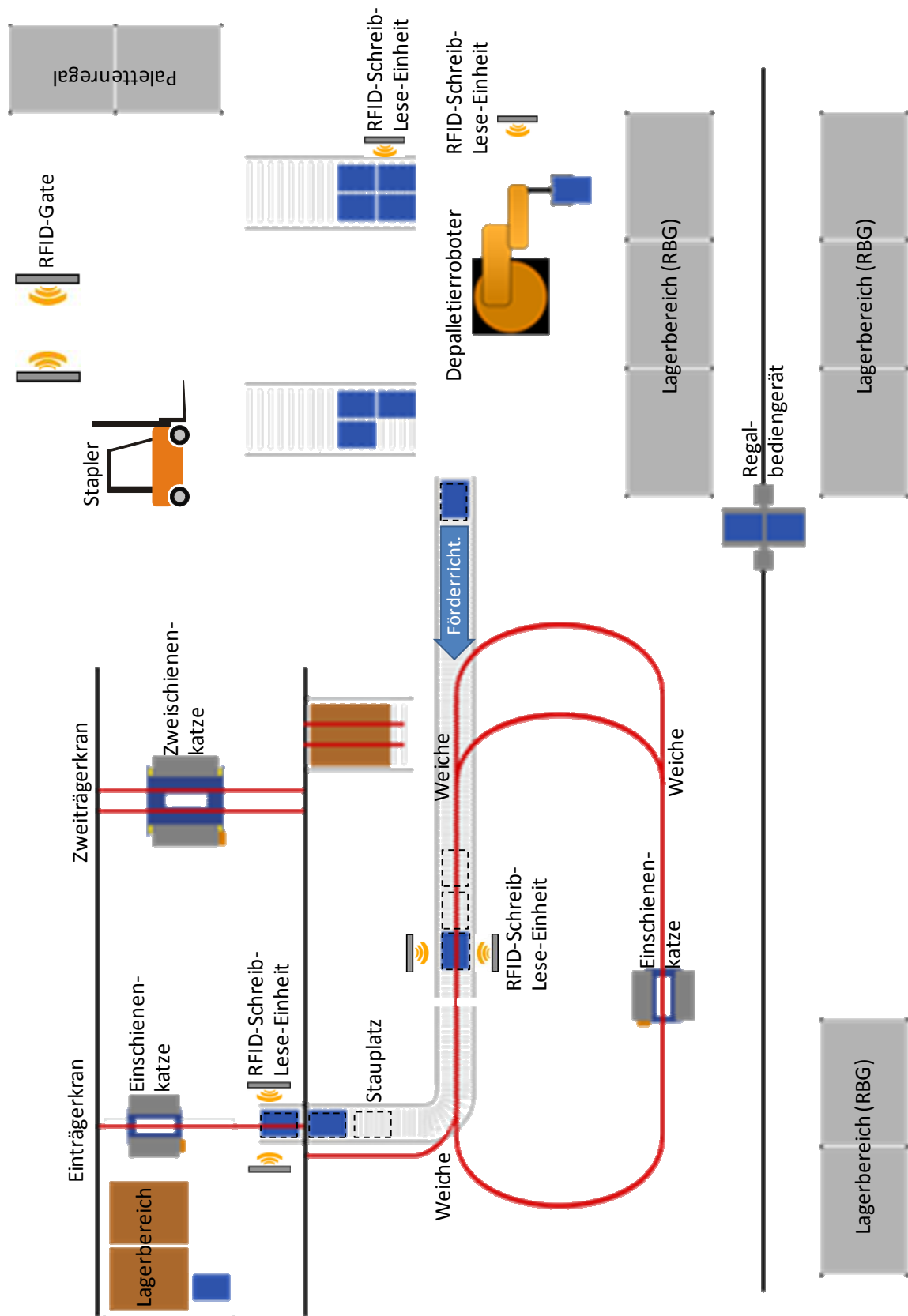


Abb. 5.2: Materialflusssystem am Lehrstuhl fml (schematisch)

Transporteinheiten

In der hier vorgestellten Versuchsanlage werden durchgängig zwei Typen von TE befördert. Das sind einerseits VDA-Kleinladungsträger mit den Abmessungen 400x600x280mm und andererseits Europaletten. Zusätzlich gibt es Fördereinrichtungen, welche Kartons verschiedener Größen (Roboter) oder Gitterboxen und (Großladungsträger-Katze, Stapler) aufnehmen können.

Sowohl die Kleinladungsträger als auch die Paletten sind jeweils mit einem RFID-Transponder ausgestattet, welcher neben einer eindeutigen ID auch Informationen über den Typ der Transporteinheit, deren Inhalt und das Ziel enthält. Bei den Transpondern handelt es sich durchgängig um UHF-RFID-Chips. Diese haben eine Kapazität von 96 Bit bzw. 12 Byte und sind auf die Speicherung eines Elektronischen Produktcodes (EPC) ausgelegt, welcher vom Massachusetts Institute of Technology entwickelt und u.a. von der GS1, einer internationalen Normungsorganisation, gepflegt wird.

RFID-Gate

Die Erfassung der Paletten am Wareneingang erfolgt durch ein RFID-Gate. Die RFID-Leseantennen sind dabei an einem Portal befestigt, so dass die Transponder beim Durchfahren des Gates automatisch erfasst und die dabei ausgelesenen Daten (wie Identnummer, Anzahl der Güter, etc.) verwendet werden können, um Softwareagenten für die jeweiligen TE zu generieren.

Elektrohängebahn

Die Elektrohängebahnanlage (EHB) lässt sich in zwei Bereiche unterteilen. Ein Bereich dient dem Transport von Kleinladungsträgern (KLT), der andere dem Befördern von Großladungsträgern (GLT) und Paletten.

Die KLT werden mit zwei EHB-Katzen transportiert, welche sich auf einer Kreisbahn, einem inneren Bogenstück, drei Weichen und einer beweglichen Schiene in einem Kranfeld bewegen können. Für das Greifen von VDA-Behältern mit einem Gewicht von bis zu 50kg wurde ein intelligentes Lastaufnahmemittel entwickelt und patentiert, das Positionierungsgenauigkeiten im Bereich mehrerer Zentimeter ausgleichen kann. Damit kann die Katze ihre Ladung sowohl an Arbeitsplätzen als auch an einer Rollenbahn absetzen bzw. abholen. Die Katzen verfügen darüber hinaus über jeweils einen eigenen IPC der Fa. Beckhoff, Laserdistanzscanner zur Kollisionsvermeidung,

WLAN-Anbindung, ein Lesegerät für RFID-Wegmarkierungen und ein absolutes Wegmesssystem.

Für den Transport der Großladungsträger steht eine weitere Katze mit ähnlicher Ausstattung bereit. Diese bewegt sich alleine auf einem eigenen, relativ kurzen Streckenabschnitt und benötigt daher weder RFID-Wegmarkierungen noch einen Laserdistanzscanner. Auch diese Katze kann auf ein bewegliches Schienenstück des Kranfelds übersetzen, welches gemeinsam mit den KLT-Katzen genutzt wird. Die GLT-Katze soll Ladungen von bis zu 1t transportieren können. Deshalb ist sie an zwei parallel laufenden Schienen aufgehängt.

Depalettierroboter

Ein Sechssachs-Knickarm-Industrieroboter der Firma Kuka wird dazu eingesetzt, KLT oder Kartons zu (de)palettieren, welche auf Europaletten auf zwei Rollenbahnen bereit gestellt werden können. Für diese Aufgabe verfügt der Roboter über ein Lastaufnahmemittel, das die Transporteinheit mit einer Saugvorrichtung fixieren und mit einer Gabel anheben kann. Da das Lastaufnahmemittel in mehreren Dimensionen einstellbar ist, können Lasten mit unterschiedlichen Abmessungen gegriffen werden.

Die Zuführstrecke ist mit einem RFID-Lesegerät ausgestattet, sodass der Inhalt und das Packmuster einer ankommenden Palette aus deren Transponder ausgelesen und dem Roboter übermittelt werden kann. Je nachdem, um welche Sorte von TE es sich handelt, wird das Lastaufnahmemittel an die jeweilige Behältergröße angepasst.

Zusätzlich befindet sich innerhalb der Roboterzelle ein weiteres Lesegerät. Der Roboter führt jeden unbekanntem Behälter an diesem Lesegerät vorbei und erhält damit Informationen über das Transportziel des Behälters. Dies kann entweder eine Leerpalette auf einer Abtransportstrecke, eine Position im Hochregallager oder der Übergabepplatz der Rollenbahn sein, welche die Roboterzelle mit der Elektrohängebahn verbindet.

Der Roboter verfügt über eine Steuereinheit der Fa. Kuka, welche via DeviceNet-Schnittstelle mit einem IPC kommuniziert. Letzterer verwaltet die Transportaufträge und übernimmt die Kommunikation mit der restlichen Anlage.

Regalbediengerät

Das Regalbediengerät ist in der Lage, die vom Roboter bereit gestellten KLT und Kartons aufzunehmen und umzulagern. Dazu ist es mit einem Lastaufnahmemittel ausgestattet, welches zweifach tief auf das Regal zugreifen kann. Die Einlastung der Aufträge bzw. manuelle Ansteuerung wird mittels eines mobilen IPC mit Touch-Screen durchgeführt.

Stapler

Der mit einem Bedienterminal ausgestattete Gegengewichtsstapler kann seine Position anhand im Boden eingelassener Transponder bestimmen und dient dem Transport von Europaletten.

5.2 Szenario

Wird das Materialflusssystem des Lehrstuhls fml nach den Konzepten der funktionsorientierten Modularisierung strukturiert, ergeben sich folgende Module:

- ein Depalettierroboter
- drei EHB-Katzen
- drei EHB-Weichen
- zwei EHB-Kräne
- ein Regalbediengerät
- ein RFID-Gate
- vier Rollenbahnen (GLT und KLT)
- ein Stapler mit Bedienterminal

Die möglichen Wege, Transportziele und Entscheidungspunkte, welche eine Transporteinheit passieren kann, sind in Abb. 5.3 dargestellt.

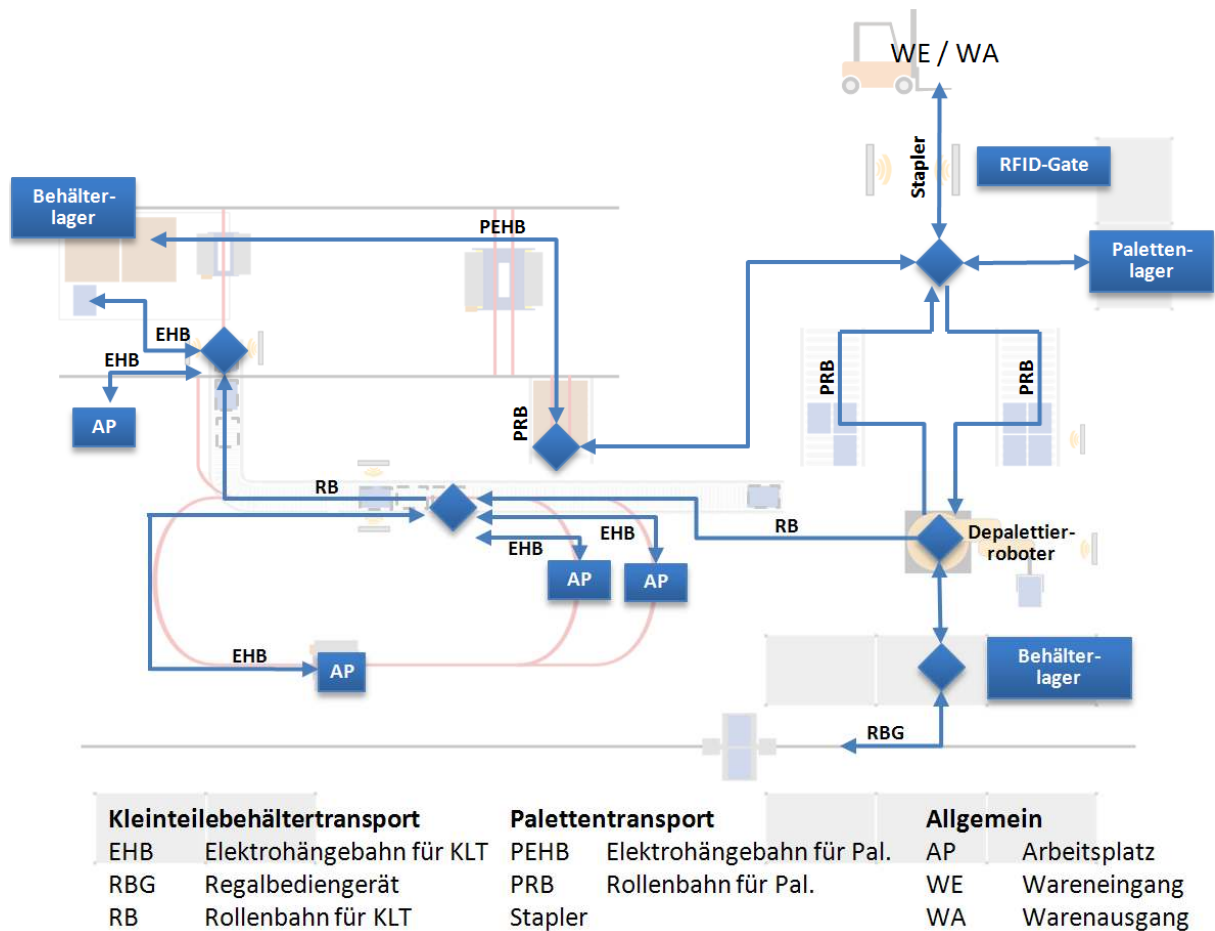


Abb. 5.3: Wege, Ziele und Entscheidungspunkte im Materialflusssystem des Lehrstuhls fml

Im Folgenden wird ein Szenario beschrieben, das diese Module in einen Zusammenhang setzt und die typischen Vorgänge in einem Materialflusssystem abbildet:

Eine Palette mit KLT wird am Wareneingang angeliefert. Der Stapler nimmt diese auf und fährt durch das RFID-Gate, um die Ladung anhand der an der Palette und den KLT angebrachten Transpondern zu identifizieren und eine Wareneingangskontrolle durchzuführen. Die Daten zur Anlieferung von Waren werden von einem Warenwirtschaftssystem verwaltet. Dieses bestimmt auch die Transportziele innerhalb der Materialflussanlage, überlässt aber die konkrete Ausführung der Transportaufträge den Modulen des Internet der Dinge. Der Softwareagent des Gates gleicht den Inhalt der Palette mit den Informationen aus dem Warenwirtschaftssystem ab und übermittelt das Ergebnis dieser Prüfung an den Stapler. Außerdem werden die auf dem Transponder der Palette vorhandenen Daten um den Zielort innerhalb der Versuchsanlage ergänzt und ein TE-Agent gestartet, welcher sich von nun an um die Belange der Palette „kümmern“ soll. Nun bekommt der Staplerfahrer über das Benutzerterminal mitgeteilt, ob es sich um eine korrekte Lieferung handelt.

Vom Wareneingang aus kann die Palette ausschließlich mit dem Stapler abtransportiert werden. Der Softwareagent des Staplers überwacht diesen Transport und teilt dem Staplerfahrer über das Bedienterminal die weiteren Schritte mit.

Für die Palette sind vier Ziele möglich:

- der Depalettierroboter
- das Palettenlager in der Nähe des Wareneingangs
- der Lagerbereich unterhalb der EHB-Anlage
- der Warenausgang

Weder der Depalettierroboter noch der Lagerbereich unterhalb der EHB-Anlage sind mit dem Stapler direkt erreichbar. Daher ist lediglich ein Transport zu einem vorgelagerten Modul möglich. Im ersten Fall ist das Transportziel des Staplers die Zuführstrecke des Depalettierroboters, im zweiten Fall die Rollenbahn unterhalb der GLT-Katze. Ein Transport mit der GLT-Katze zum Lagerbereich kann über den EHB-Kran erfolgen. Im Kranfeld müssen sich die zwei Kräne für die KLT- und die GLT-Katzen miteinander abstimmen, um Kollisionen zu vermeiden. Der Warenausgang und das Palettenlager können direkt mit dem Stapler angefahren werden. Verfügt das Ziel über einen eigenen Agenten, so fragt der Stapleragent bei seiner Ankunft die Bereitschaft des Ziels zur Aufnahme der Palette an und führt, sofern die Erlaubnis dazu erteilt wurde, einen Lastwechsel durch.

Im Folgenden wird davon ausgegangen, dass die Palette KLT enthält, welche zu einem Arbeitsplatz unterhalb der EHB-Anlage geliefert werden müssen. Daher befördert der Stapler die Palette zur Zuführstrecke des Depalettierroboters. Hier passiert sie ein RFID-Lesegerät, welches das Packmuster aus dem Transponder der Palette ausliest oder über das Warenwirtschaftssystem in Erfahrung bringt. Die Zuführstrecke informiert den Roboter, welcher nun mit dem Depalettieren beginnt. Der Roboter führt dabei jeden KLT der Palette an einem zweiten RFID-Lesegerät in der Roboterzelle vorbei und kann durch den daran angebrachten Transponder dessen nächstes Ziel bestimmen. Vom Palettierroboter aus können folgende Ziele erreicht werden:

- das Behälterlager, auf welches auch das RBG zugreifen kann
- eine Leerpalette auf der Abtransportstrecke
- die Rollenbahn, welche zur EHB-Anlage führt

Der Roboter setzt, nachdem auch hier die Bereitschaft zum Lastwechsel abgefragt wurde, den KLT auf die Rollenbahn, welche zur EHB-Anlage führt.

Diese Rollenbahn transportiert den Behälter bis zu einem Stauplatz, welcher von oben mit einer EHB-Katze erreichbar ist. An diesem Stauplatz befindet sich ein RFID-Lesegerät, welches den KLT-Transponder ausliest. Der Agent der Rollenbahn führt nun eine Wegplanung durch und kann entscheiden, ob der KLT am Stauplatz angehalten werden soll, um eine Übergabe an die EHB zu ermöglichen, oder ob er weitertransportiert wird. Da das Ziel im Bereich der EHB-Anlage liegt, wird der KLT am Stauplatz angehalten. Nun wird eine EHB-Katze gesucht, welche die Beförderung auf dem letzten Teilstück übernimmt. Eine von der Rollenbahn initiierte Auktion ermittelt diejenige Katze, welche sich am nächsten zum aktuellen Standplatz des KLT befindet. Diese plant ihren Weg und sperrt ihn entgegen ihrer eigenen Fahrtrichtung. Damit werden Kollisionen mit der anderen EHB-Katze verhindert. Auf ihrem Weg zum Ziel kann es notwendig sein, Weichen zu stellen. Diese Aufgabe wird von der Katze in einer direkten Abstimmung mit der jeweiligen EHB-Weiche erfüllt. Beim KLT angekommen, wartet die EHB-Katze auf die Erlaubnis der Rollenbahn, diesen aufzunehmen. Nach dem Lastwechsel bewegt sie sich schließlich zum Arbeitsplatz, welcher das endgültige Ziel des KLT ist, um den Behälter dort abzustellen.

5.3 Realisierte Module

Das im vorherigen Kapitel beschriebene Szenario bietet die Möglichkeit, die konkrete Ausprägung der logistischen Funktionen von Modulen in der Versuchshalle des Lehrstuhls fml zu bestimmen. Auf diese Weise kann eine Realisierung von Teilbereichen so gestaltet werden, dass diese später aufwandsarm zu einem funktionierenden Gesamtsystem zusammengefügt werden können.

Im Rahmen der vorliegenden Arbeit wurden die EHB-Anlage, ihre Katzen, Kräne und Weichen sowie die darunter liegenden Rollenbahnen als voll funktionsfähiges Transportsystem nach den Konzepten des Internet der Dinge realisiert. Anhand des Depalettierroboters und der davor liegenden Rollenbahnen konnte die Identifikation

mittels RFID-Lesegeräten, eine flexible Greiftechnologie und das automatische Depalettieren gezeigt werden. Diese bereits realisierten Module sind so gestaltet, dass eine Ergänzung um weitere Module jederzeit möglich ist, um das beschriebene Szenario zu erweitern.

Im Folgenden werden die am Lehrstuhl fml vorhandenen Module ihren Funktionsklassen und Funktionen zugeordnet, die Auftragsdisposition als zentrale Modulfunktion beschrieben und das der Anlagenrealisierung zugrunde liegende modulinterne und externe Kommunikationsmodell erläutert. Eine Beschreibung der verwendeten Steuerungshardware schließt das Kapitel ab.

5.3.1 Modulklassen und -funktionen

Die Module des Szenarios lassen sich wie folgt in die in Kap. 4.1.4 beschriebenen Funktionsklassen einordnen:

Unstetigförderer

- EHB-Katzen
- Regalbediengerät
- Stapler mit Bedienterminal

Stetigförderer und Schienen

- Rollenbahnen

Verzweigung / Zusammenführung

- EHB-Weichen
- EHB-Kräne

Arbeitsstation

- RFID-Gate
- Depalettierroboter

Im Rahmen der vorliegenden Arbeit wurde aus jeder Funktionsklasse mindestens ein Modul ausgewählt, an welchem spezielle Konzepte des Internet der Dinge überprüft wurden. Das Regalbediengerät, das RFID-Gate und der Stapler wurden noch nicht realisiert.

Mit den realisierten Modulen wurden alle in Kap. 4.1.4 beschriebenen Modulfunktionen durch eine Implementierung abgedeckt (siehe .Tab. 5.1).

	EHB-Katze	EHB-Weiche	EHB-Kran	Rollenbahn	Roboter
Eigene Fähigkeiten und Funktionen bekannt geben	x	x	x	x	
Verwaltung des eigenen Zustands	x	x	x	x	x
Verwaltung der eigenen Plätze	x	x	x	x	x
Ermittlung der Eigenschaften einer TE	x			x	x
Aufsetzpunkt / Übergabepplatz bereit stellen	x			x	x
Einlasten von TE in den Materialfluss				x	x
Lastwechselabstimmung	x			x	x
Auftragsdisposition	x			x	x
Anzeige von Auftragsinformationen für einen Bediener	x				x
Transportauftrag ausführen	x			x	x
Herstellen von Wegverbindungen		x	x		x
Koordinationsmechanismen	x				
Kollisionskontrolle	x	x	x		
Wegplanung	x			x	x

Tab. 5.1: Am Lehrstuhl fml realisierte Modulfunktionen

5.3.2 Auftragsdisposition

Wie aus dem im Kap. 5.2 beschriebenen Szenario hervorgeht, ist die Auftragsdisposition eine Modulfunktion, welche vor jedem Wechsel eines Transportmittels stattfindet, um den nächsten Transporteur für eine TE zu identifizieren. Der Ablauf ist dabei folgendermaßen (siehe Abb. 5.4):

- Die einzelnen Module melden sich nach ihrer Initialisierung bei einem Verzeichnisdienst an. Dort geben sie ihre Funktionen und Eigenschaften bekannt und legen fest, welche Typen von TE sie transportieren können.
- Soll eine TE transportiert werden, so stellt diese eine Anfrage an den Verzeichnisdienst und erhält als Antwort eine Liste aller Module, welche prinzipiell dafür geeignet sind, den jeweiligen Typ von TE zu befördern. Im Rahmen einer Auktion bittet die TE nun alle in dieser Liste verzeichneten Module um die Abgabe eines Gebots für den Transport.

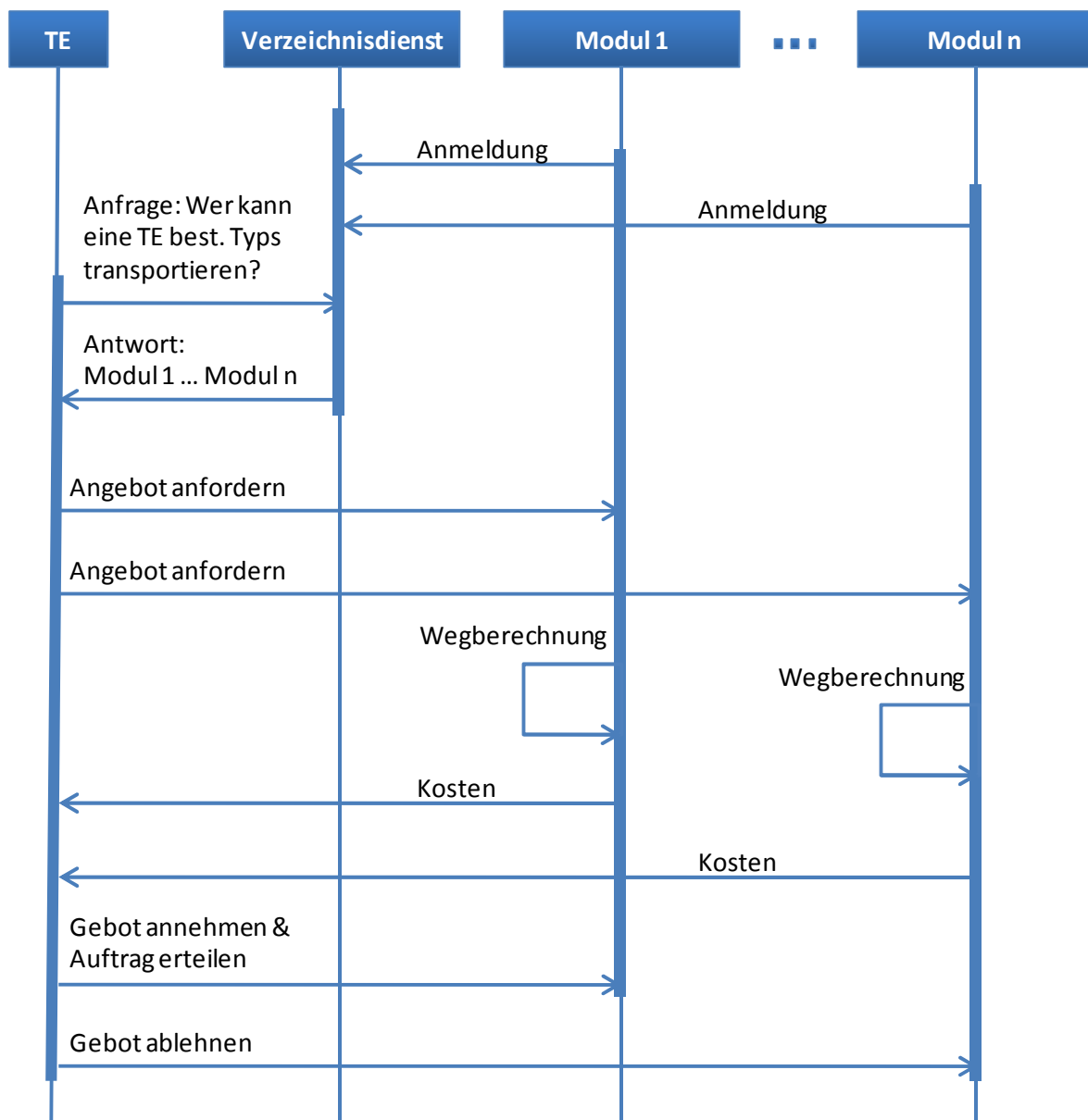


Abb. 5.4: Auftragsdisposition

- Alle Module, welche gerade mit einem anderen Auftrag beschäftigt sind, schicken kein Gebot, sondern senden eine negative Antwort an die TE. Die anderen Module berechnen anhand eines speziellen Algorithmus bzw. einer Kostenfunktion, wie gut sich das Modul für den angefragten Transport eignet. Dazu muss zuerst entschieden werden, ob das Modul das Ziel oder ein mögliches Zwischenziel darstellt. Die EHB-Katzen berechnen ihren Weg mittels des Dijkstra-Algorithmus [Dij-59], wobei auch die aktuelle Streckenbelegung und eventuelle Wegreservierungen anderer Katzen in die Berechnung mit einbezogen werden. Ist das Ziel nicht über die EHB erreichbar, so wird eine Ablehnung an die TE versendet. Bei Erreichbarkeit des Ziels liefert der Dijkstra-Algorithmus alle Wegpunkte, welche auf

dem Weg zum Ziel liegen und die dafür zurückzulegende Wegstrecke. Das Gebot einer EHB-Katze besteht dann in der Zeit, welche benötigt wird, um die TE anzufahren, und die sich aus der Fördergeschwindigkeit und der Entfernung bzw. eventuellen Schaltzeiten berechnet.

Die im Szenario vorhandenen Rollenbahnen und der Palettierroboter können nur wenige Wegalternativen bereit stellen. Daher wurde bei den Rollenbahnen und dem Roboter kein Wegplanungsalgorithmus implementiert, sondern lediglich eine Liste von Wegverbindungen und die dafür benötigten Zeiten hinterlegt.

- Nachdem die TE von allen angefragten Modulen ein Gebot erhalten hat, kann sie sich für das niedrigste entscheiden und erteilt dem entsprechenden Modul den Transportauftrag.

Das hier vorgestellte Verfahren zur Auftragsdisposition hat den Vorteil, dass es unabhängig von der Anzahl und der Ausprägung der Fördertechnikmodule einsetzbar ist. Wird bei einer eventuellen Anlagenerweiterung ein zusätzliches Modul bzw. eine neue Wegstrecke hinzugefügt, so sind diese ohne eine Programmanpassung sofort einsetzbar.

5.3.3 Steuerungshardware und -software

Das Internet der Dinge-Konzept stellt neue Anforderungen an die verwendete Steuerungstechnik und propagiert den verstärkten Einsatz von Hochsprachen auch auf der Ebene der Fördertechnik. Soll das Konzept ohne Einschränkungen verwirklicht werden, so müssen moderne Steuerungen eingesetzt werden, welche über dazu geeignete Schnittstellen und Betriebssysteme verfügen. Im Folgenden werden die am Lehrstuhl fml verwendete Steuerungshardware und die darauf implementierte Steuerungssoftware beschrieben.

5.3.3.1 Steuerungshardware

In Kap. 4.2.3 wurde ein IPC als vielversprechende Hardwarelösung für die Umsetzung des Internet der Dinge empfohlen. Die Module in der Versuchsanlage des Lehrstuhls fml sind daher mit einem IPC automatisiert.

Der IPC CX9010 der Firma Beckhoff (siehe Abb. 5.5) hat folgende technische Daten:

- Intel IXP-Prozessor XSCALE (533MHz)
- 32 MB interner Flash-Speicher
- 128 MB interner Arbeitsspeicher
- E/A-Schnittstelle EtherCAT
- Windows CE Betriebssystem
- SPS-Laufzeitumgebung
- 16 Digitale Eingänge
- 16 Digitale Ausgänge
- CAN-Bus Schnittstelle



Abb. 5.5: Beckhoff IPC CX9010

Die hier vorgestellte Hardwarelösung wurde gewählt, da auf dieser Plattform sowohl die strategische als auch die operative Steuerung implementiert und daher die Module nach den Vorgaben der funktionsorientierten Modularisierung gestaltet werden können. Dadurch, dass einerseits Windows CE und andererseits eine echtzeitfähige SPS-Laufzeitumgebung integriert sind, sind sowohl eine Hochsprachenprogrammierung als auch die Programmierung mit IEC 61131-3 Programmiersprachen möglich. Die Kommunikation zwischen diesen beiden Betriebssystemen kann mit der herstellereigenen TwinCAT ADS-Schnittstelle realisiert werden.

Der IPC ist modular und um vielfältige Ein-/Ausgangsbausteine erweiterbar und genügt den Anforderungen einer industriellen Umgebung. Die standardmäßig vorhandene Ethernet-Schnittstelle garantiert eine schnelle Einbindung des IPC in das Kommunikationsnetzwerk der Versuchshalle. Die beweglichen Fördermodule, wie EHB-Katzen oder Stapler können drahtlos über handelsübliche WLAN-Access-Points angebunden werden. Die modulinterne und echtzeitfähige Kommunikation wird mit einer CAN-Bus-Schnittstelle realisiert, welche mit Funktionsbausteinen des Maschinensteuerungsprogramms angesprochen wird.

Mit diesen Eigenschaften ist die hier vorgestellte Hardwarelösung sehr gut für den Aufbau eines Internet der Dinge-Materialflusssysteme geeignet.

5.3.3.2 Steuerungssoftware

Die Steuerungssoftware für Module wird auf der operativen Steuerungsebene mit Maschinensteuerungssprachen nach IEC 61131-3 umgesetzt. Dazu steht die Programmierumgebung TwinCAT zur Verfügung, welche es gestattet das Steuerungs-

programm auf einem herkömmlichen PC zu entwickeln und dann via Ethernet auf die Steuerung zu übertragen. TwinCAT baut auf die Entwicklungsumgebung CoDeSys (Controller Development System) auf. Über 250 namhafte Gerätehersteller aus unterschiedlichen Branchen setzen CoDeSys als Programmierschnittstelle für ihre Automatisierungsgeräte ein. Damit ist CoDeSys neben Step 7, der Entwicklungsumgebung von Siemens, eines der am meisten verbreiteten Entwicklungssysteme in Europa.

Die strategische Steuerung ist als Agentensystem ausgeführt und verwendet das Agentenframework JADE (Java Agent DEvelopment Framework). Dieses implementiert die Vorgaben der FIPA-Organisation [FIPA-05], welche die Standardisierung von Agentensystemen voran treibt. JADE ist in der Hochsprache Java implementiert und stellt dem Programmierer vielfältige Hilfsmittel zur Agentenprogrammierung bereit. Beispiele sind die „Lebensumgebungen“ für Softwareagenten in sogenannten Containern, eine Unterstützung für die Programmierung von nebenläufigen Prozessen und verschiedene Kommunikationsprotokolle. Für die Programmierung in Java kann auf unterschiedliche Entwicklungsumgebungen zurück gegriffen werden. Am Lehrstuhl fml wurde NetBeans eingesetzt, eine Open-Source-Software, welche sowohl über einen leistungsfähigen Editor als auch über die klassischen Funktionen zur Fehlersuche verfügt.

5.3.4 Schnittstellen und Kommunikation

In der Versuchsanlage des Lehrstuhls fml wurden verschiedene Schnittstellen und Kommunikationskonzepte für die modulinterne und modulexterne Kommunikation eingesetzt. Diese werden im Folgenden beschrieben.

5.3.4.1 Modulinterne Schnittstellen und Kommunikation

Wie in Kap. 4.2.1 beschrieben, verfügt ein Modul über maximal drei interne Schnittstellen. Das sind:

- die Verbindung der Sensoren und Aktoren mit der maschinennahen Steuerungsebene
- die Kommunikationsschnittstelle zwischen strategischer und operativer Steuerung
- die Schnittstelle zwischen der operativen Steuerung und der Sicherheitssteuerung, sofern diese beiden Steuerungsarten in getrennten Geräten implementiert sind

Die am Lehrstuhl fml realisierten Module vereinen die operative Steuerung und die Sicherheitssteuerung in einem gemeinsamen Steuerungsprogramm, daher entfällt diese Schnittstelle. Die übrigen Schnittstellen sind folgendermaßen realisiert:

Sensoren / Aktoren - maschinennahe Steuerungsebene

Die Sensoren und Aktoren sind entweder direkt mit den Ein-/Ausgangsbausteinen des IPC verbunden oder werden über eine CAN-Bus-Schnittstelle angesprochen.

Im ersten Fall werden Maschinensteuerungsprogramm die Ein- und Ausgänge mit Variablen verknüpft. Dabei bilden Eingangsvariablen den aktuellen Zustand der Sensoren ab, während die Aktoren durch eine Veränderung von Ausgangsvariablen geschaltet werden können.

Im zweiten Fall sind Baugruppen über den CAN-Bus an den IPC angebunden, welche über eine eigene Intelligenz verfügen. Sie teilen ihren Zustand selbst mit und können mit entsprechenden Nachrichten dazu veranlasst werden, diesen zu verändern. Ein Beispiel für eine solche Baugruppe ist ein Laserdistanzsensor, welcher regelmäßig den Abstand zu einem Hindernis meldet.

Strategische Steuerung - Operative Steuerung

Die Schnittstelle zwischen operativer Steuerung und Sicherheitssteuerung wird durch eine Middleware realisiert. Diese erlaubt sowohl den Zugriff auf die Programmvariablen des SPS-Programms der operativen Steuerung als auch die Kommunikation mit einem Emulatoragenten (siehe auch Kap. 4.4.1). Welche Kommunikationsform gerade aktiv ist, wird über eine Konfigurationsdatei geregelt und ist für die strategische Steuerung nicht sichtbar. Die Konfigurationsinformationen sind in der Beschreibungssprache XML (eXtensible Markup Language) formuliert. Diese Sprache zeichnet sich dadurch aus, dass neben den Daten selbst auch Informationen über die Bedeutung dieser Daten enthalten sind. Dadurch können neue Daten für bestimmte Module aufgenommen werden, ohne dass bei allen anderen Modulen in die Programmierung eingegriffen werden muss. Jeder Empfänger interpretiert nur die für ihn relevanten Daten und ignoriert die restlichen Informationen.

```
<envelope>
<geraetedaten>
  <virtuell>false</virtuell>
  <zugriffsart>Manuell</zugriffsart>
  <zugriffsart>Automatik</zugriffsart>
</geraetedaten>
<spsinterface>
  <ADSSOAPIP>192.168.10.20</ADSSOAPIP>
  <message id="Fahrauftrag_Fahrwerk_Position_mm" io="out" valueType="int">
    <transport type="wsadsgo" port="301" group="F030" offset="5A" netid="5.2.3.26.1.1"/>
  </message>
  <message id="Fahrauftrag_WP_Stopp" io="out" valueType="bool">
    <transport type="wsadsgo" port="301" group="F031" offset="298" netid="5.2.3.26.1.1"/>
  </message>
  <message id="Fahrauftrag_Fahrtrichtung" io="out" valueType="int">
    <transport type="wsadsgo" port="301" group="F030" offset="54" netid="5.2.3.26.1.1"/>
  </message>
  <message id="Fahrauftrag_WP_Global" io="out" valueType="int">
    <transport type="wsadsgo" port="301" group="F030" offset="55" netid="5.2.3.26.1.1"/>
  </message>
  <message id="Fahrauftrag_WP_Lokal" io="out" valueType="int">
    <transport type="wsadsgo" port="301" group="F030" offset="57" netid="5.2.3.26.1.1"/>
  </message>
  <message id="Fehlernummer" io="in" valueType="int">
    <transport type="wsadsgo" port="301" group="F020" offset="4B" netid="5.2.3.26.1.1"/>
  </message>
  <message id="Auftrag_beendet_Nr" io="in" valueType="int">
    <transport type="wsadsgo" port="301" group="F020" offset="4B" netid="5.2.3.26.1.1"/>
  </message>
</spsinterface>
</envelope>
```

Abb. 5.6: Konfigurationsdatei für eine reale EHB-Katze

In Abb. 5.6 ist ein Auszug aus einer Konfigurationsdatei für eine reale EHB-Katze abgebildet. Diese Datei gliedert sich in die zwei Abschnitte „Gerätedaten“ und „SPS-Interface“.

In den Gerätedaten können Festlegungen zur konkreten Ausprägung eines Förder-technikmoduls getroffen werden. Die hier beschriebene EHB-Katze verfügt, im Gegensatz zur zweiten Katze im System, über ein automatisches Lastaufnahmemittel, das selbstständig die Aufnahme bzw. Abgabe einer TE ausführen kann. In der Konfigurationsdatei ist über die Zugriffsart spezifiziert, dass diese Katze sowohl Aufträge übernehmen kann, welche manuell abgearbeitet werden können als auch solche, welche automatisch ausgeführt werden müssen.

Das strategische Steuerungsprogramm kann konfigurierbar gestaltet werden, so dass beispielsweise die Ansteuerung unterschiedlicher Rollenbahnen mit der gleichen Programmlogik realisierbar ist (siehe Kap. 4.3.3). Die konkrete Ausprägung wird dann in der hier vorgestellten Konfigurationsdatei festgelegt. Im Abschnitt „Gerätedaten“ können z.B. die Anzahl und die Positionen der Übergabeplätze definiert werden, ohne dass dazu das strategische Steuerungsprogramm geändert werden muss.

Der Abschnitt „SPS-Interface“ legt den Zugriff der strategischen Steuerung auf die Programmvariablen der operativen Steuerung fest. Hier werden einerseits die IP-Adresse der operativen Steuerung und andererseits für jede Variable folgende Informationen übergeben:

- Variablenname
- Variablentyp
- Übertragungsrichtung
- Übertragungsprotokoll

Das Übertragungsprotokoll gibt an, auf welchem Weg auf die Variable zugegriffen wird. Die am Lehrstuhl fml realisierte Middleware kann sowohl mit einem IPC des Steuerungstechnikherstellers Beckhoff (siehe Kap. 5.3.3) kommunizieren als auch mit einem Emulatoragenten.

Die herstellereinspezifische Schnittstelle TwinCAT ADS, welche zur Kommunikation mit dem IPC eingesetzt wird, benötigt spezielle Zusatzdaten, um eine Variable eindeutig

zu adressieren. Auf die Bedeutung dieser Informationen soll hier nicht näher eingegangen werden. Die Konfigurationsdatei für eine Emulation ist in Abb. 5.7 zu sehen.

```
<envelope>
<geraetedaten>
  <virtuell>true</virtuell>
  <zugriffsart>Automatik</zugriffsart>
  <zugriffsart>Manuell</zugriffsart>
</geraetedaten>
<spsinterface>
  <message id="Fahrauftrag_Fahrwerk_Position_mm" io="out" valueType="int"> <transport type="agent" /> </message>
  <message id="Fahrauftrag_WP_Stopp" io="out" valueType="bool"> <transport type="agent" /> </message>
  <message id="Fahrauftrag_Fahrtrichtung" io="out" valueType="int,> <transport type="agent" /> </message>
  <message id="Fahrauftrag_WP_Global" io="out" valueType="int,> <transport type="agent" /> </message>
  <message id="Fahrauftrag_WP_Lokal" io="out" valueType="int,> <transport type="agent" /> </message>
  <message id="Fehlernummer" io="in" valueType="int,> <transport type="agent" /> </message>
  <message id="Auftrag_beendet_Nr" io="in" valueType="int,> <transport type="agent" />
</message>
</spsinterface>
</envelope>
```

Abb. 5.7: Konfigurationsdatei für den Emulator einer EHB-Katze

Die in diesen Konfigurationsdateien spezifizierten Programmvariablen lassen sich wie folgt untergliedern:

- Statusmeldungen
 - Betriebszustand (z.B. Automatikmodus, manuelle Bedienung)
 - Störungsmeldungen
 - Aktueller Standort
- Auftragsdaten
 - Auftragsart (z.B. Leerfahrt, Transport)
 - Anzufahrende Position
 - An Startpunkt und Zielpunkt durchzuführende Aktion (z.B. Aufnahme einer TE) und die dazu notwendigen Informationen wie die Zugriffsrichtung
- Daten zur Auftragssteuerung
 - Auftragsfreigabe
 - Vollzugsmeldung
 - Fehlermeldung
 - Quittierung einer Fehlermeldung

Die Kommunikation zwischen der strategischen und der operativen Steuerung bei der Abarbeitung eines Auftrags ist in Abb. 5.8 dargestellt.

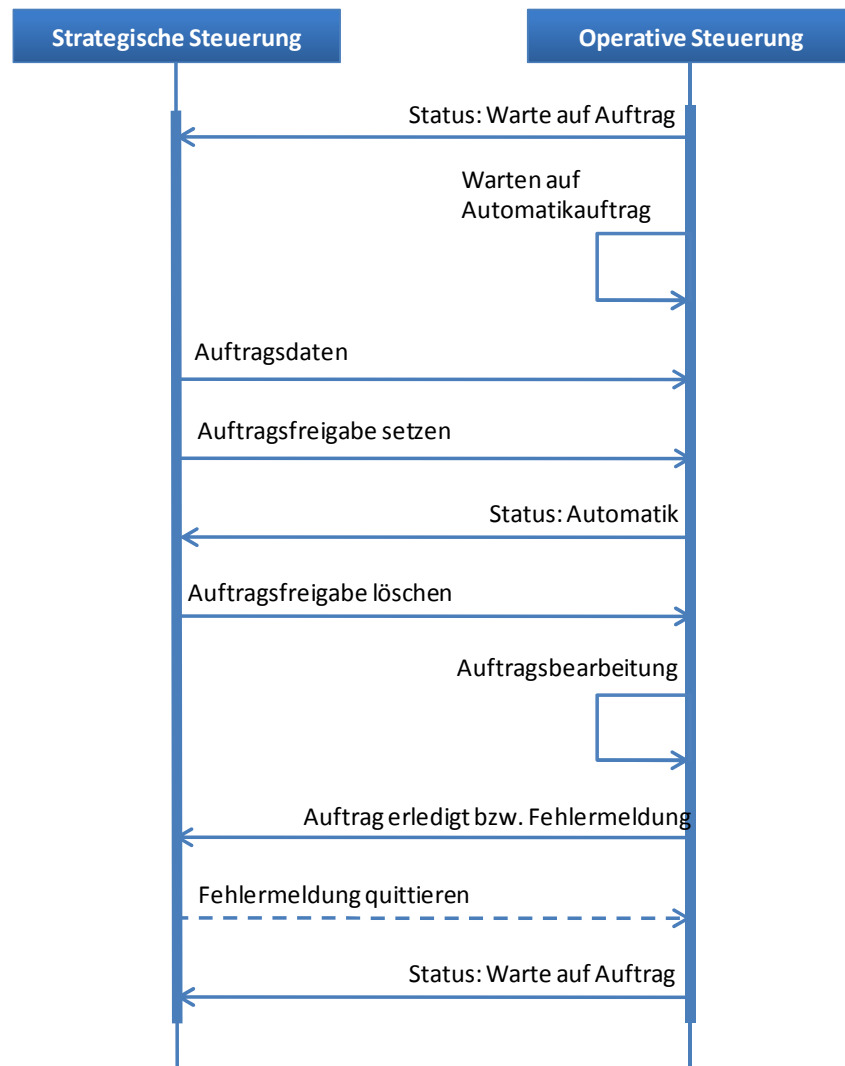


Abb. 5.8: Kommunikation zwischen strategischer und operativer Steuerung

5.3.4.2 Modulare Schnittstelle und Kommunikation

Jedes Modul verfügt über zwei externe Schnittstellen zu anderen Modulen bzw. Entitäten (siehe Kap. 4.2.1.1):

- ein Notauskreis verbindet die Sicherheitssteuerungen miteinander
- eine Ethernetschnittstelle dient der Übertragung von Daten zu den strategischen Steuerungen anderer Module oder zu anderen Entitäten

Der Notauskreis ist in der Versuchshalle des Lehrstuhls fml im Bereich des Depalettierroboters als Direktverbindung zwischen den operativen Steuerungen realisiert. Die EHB-Module überwachen sicherheitskritische Funktionen, beispiels-

weise mittels der Laserdistanzsensoren selbst, und müssen dazu nicht mit anderen Modulen kommunizieren.

Die zwischen den Entitäten übertragenen Nachrichten sind mit der Beschreibungssprache XML kodiert, welche eine gute Strukturierung komplexer Daten erlaubt, indem diese hierarchisch gegliedert werden.

Das in der Versuchsanlage des Lehrstuhls fml realisierte Kommunikationskonzept ist eine Mischung aus direkter Kommunikation zwischen den Entitäten und der Verwendung einer zentralen Datenaustauschplattform.

Die direkte Kommunikation eignet sich besser für die Übertragung von Informationen, welche nur für ein spezielles Modul bestimmt sind, während die Datenaustauschplattform für die Verteilung von allgemein interessanten Informationen verwendet wird.

So wird beispielsweise die Übermittlung eines Schaltauftrags an eine Weiche direkt kommuniziert, während die Topologieinformationen über die Datenaustauschplattform verteilt werden. Durch diese Aufteilung werden die Vorteile beider Kommunikationsprinzipien kombiniert.

6 Auswirkungen des Internet der Dinge-Konzepts

Die Konzepte der vorliegenden Arbeit, haben vielfältige Auswirkungen auf die zukünftige Planung, Realisierung und Inbetriebnahme von Materialflusssystemen. Es ist davon auszugehen, dass sich mit dem Internet der Dinge einige Tätigkeiten verkürzen werden, sich aber bei anderen der Aufwand erhöhen wird. Außerdem sorgt die verstärkt parallele Bearbeitung von Arbeitsinhalten dafür, dass bestimmte Tätigkeiten früher als bisher durchgeführt werden können. Im Rahmen des Forschungsprojekts Internet der Dinge wurden diese Veränderungen in Expertenrunden diskutiert und abgeschätzt. Im Folgenden werden die Ergebnisse dieser Gespräche über die Phasen im Lebenszyklus eines Materialflusssystems aufgezeigt.

6.1 Planungsphase

In der Planungsphase (siehe auch Kap. 2.2.1.1) ist zu erwarten, dass sich verschiedene Arbeitsinhalte vereinfachen werden und sich damit die Bearbeitungszeit verkürzt. Abb. 6.1 bietet einen Überblick über die Tätigkeiten in dieser Phase und zeigt die erwarteten Verbesserungen.

Ein zentraler Aspekt des Internet der Dinge Konzeptes ist die Wiederverwendbarkeit einmal realisierter Lösungen. Daher sollte auch in der Planungsphase auf vielfältige Informationen aus Vorgängerprojekten zurück gegriffen werden können (siehe Kap. 4.2.4). Bei der Machbarkeitsstudie sind evtl. bereits Simulationsmodelle für einzelne Module vorhanden und die

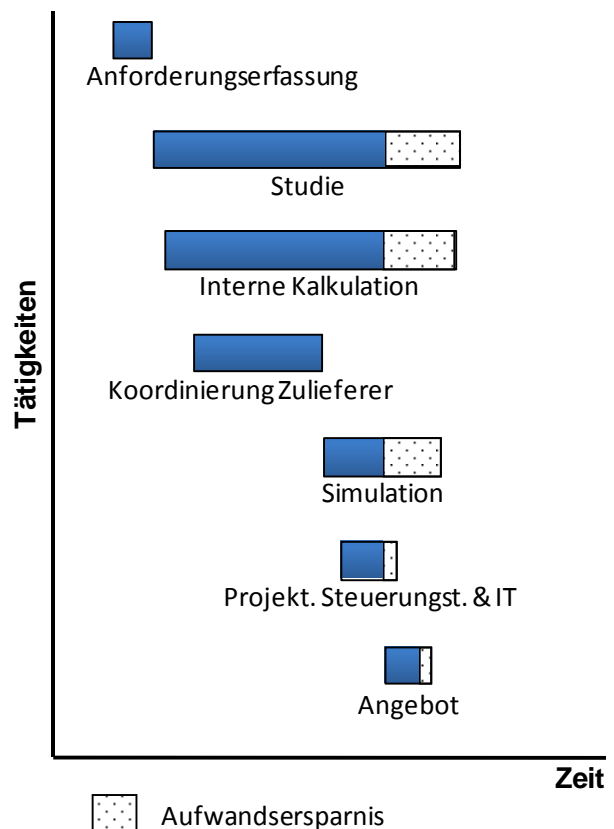


Abb. 6.1: Erwartete Veränderung des Zeitaufwands in der Planungsphase(qualitativ)

interne Kalkulation kann von Erfahrungswerten bzgl. Aufwand und Kosten der Modulsimulation und -realisierung profitieren. Da diese Informationen sich jeweils eindeutig den Modulen zuordnen lassen, kann auch „modular kalkuliert“ werden.

Die strategische Steuerungslogik eines Moduls wird sowohl für die reale Anlage als auch für eine Simulation eingesetzt (siehe Kap. 4.4.1). Für jedes einmal realisierte Modul steht damit, ohne zusätzlichen Aufwand, ein aktuelles Simulationsmodell zur Verfügung. Dieses dient als Baustein für die Erstellung der Simulation des Gesamtsystems. Zwar müssen die Module zusätzlich anhand der Planungsvorgaben miteinander verknüpft werden, jedoch erwarten sich die Planer durch einen Internet der Dinge-Modulbaukasten eine Halbierung des Aufwands bei der Erstellung von größeren Simulationsmodellen.

Die Module des Internet der Dinge sind mechatronische Systeme, welche mit eigener Steuerungshardware und -software ausgestattet sind. Die Projektierung der Steuerungstechnik und IT wird daher einfacher durchführbar sein als bisher, da die Module projektunabhängig entwickelt und im konkreten Projekt ohne großen Aufwand wiederverwendet werden können.

Dadurch, dass die technischen Details eines Moduls von Beginn an bekannt sind, wird sich genauso wie die Kalkulation auch die komplette Angebotserstellung erheblich vereinfachen.

6.2 Realisierungsphase

Das Internet der Dinge-Konzept wirkt sich auch in der Realisierungsphase (siehe Kap. 2.2.1.2) positiv aus. Abb. 6.2 zeigt diese Auswirkungen anhand der in dieser Phase ausgeführten Tätigkeiten und der erwarteten Verbesserungen.

In der Detailstudie werden die konkret eingesetzten Technologien und Materialflussstrategien spezifiziert und der Leistungsnachweis des Konzeptes anhand eines verfeinerten Simulationsmodells geführt. Hier kann davon ausgegangen werden, dass sich der Anteil der projektspezifischen Softwarebausteine verringert. Dies ist wegen der konsequenten Standardisierung, vor allem im Bereich der Schnittstellen, zu erwarten. Darüber hinaus können Strategieimplementierungen vielseitiger eingesetzt werden als bisher: Sie sind unabhängig von der Anzahl der beteiligten Fördertechnologien.

nikmodule oder der Topologie des Systems entwickelt worden. Ein strategisches Steuerungsprogramm wird direkt von der Simulation auf eine reale Anlage übertragen. Daher verkürzt sich die Programmierung der Steuerungsprogramme, denn wichtige Probleme sind bereits bei der Erstellung des Simulationsmodells gelöst worden.

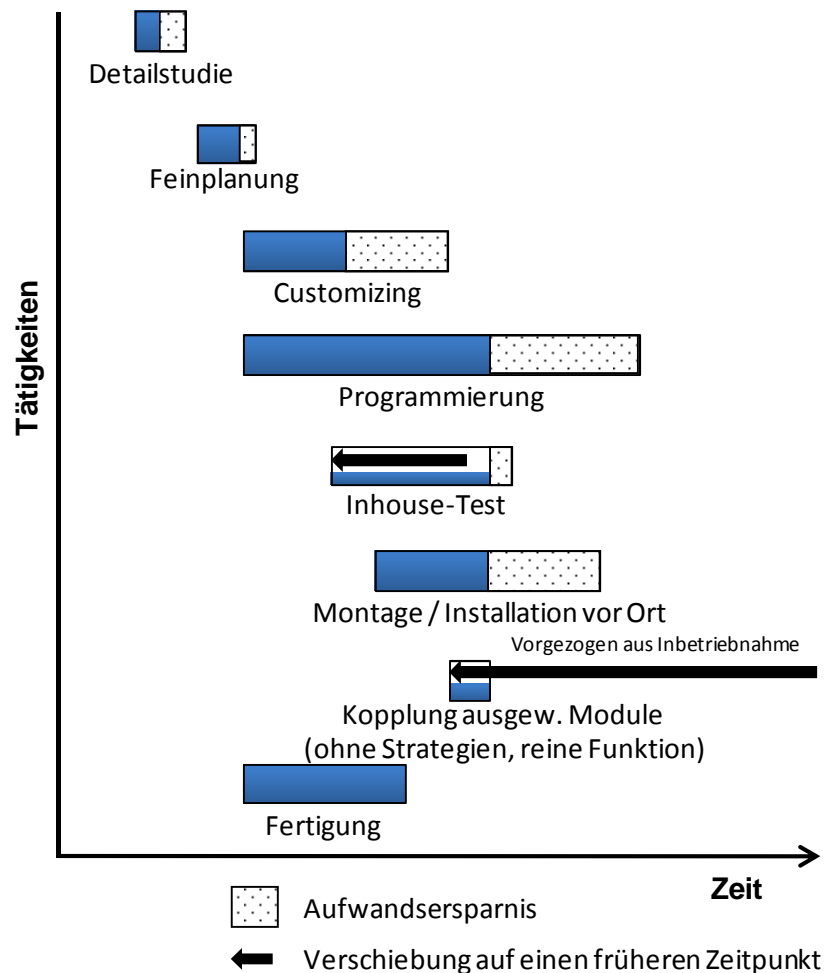


Abb. 6.2: Erwartete Veränderung des Zeitaufwands in der Realisierungsphase(qualitativ)

Spezielle Entitäten werden für bestimmte Sicherheits-, Kunden- und Werksanforderungen länderspezifisch vorbereitet und in einen Modulbaukasten integriert. Dabei sind neben Softwareanpassungen auch mechanische Varianten denkbar. Die begleitenden Informationen und Programme werden dann strukturiert mit einem Produktdatenverwaltungssystem gespeichert. Damit können der Auswahlprozess für konkrete Module rechnergestützt durchgeführt und neue Modulvarianten aus vorhandenen Lösungen zusammen gesetzt werden. In diesem Verwaltungssystem sind auch Abmessungen, Konstruktions- und Verkabelungspläne und andere für eine Feinplanung relevante Daten hinterlegt, was den Aufwand in dieser Phase merklich reduziert. Die

Anpassung von Soft- und Hardware an eine konkrete Kundenlösung (Customizing) wird daher seltener notwendig sein.

Indem Software häufiger wiederverwendet wird, kann diese von einer Anlagenrealisierung zur nächsten iterativ verbessert werden. Das Internet der Dinge ist ein verteiltes System, das schrittweise aufgebaut und getestet werden kann. Dadurch kann der Inhouse-Test, im Gegensatz zu früher, parallel zur Programmierung durchgeführt werden. Die beim Hersteller vormontierten Fördertechnikmodule sind mit Sensoren, Aktoren, Steuerungstechnik und einer eigenen webbasierten Visualisierung ausgestattet und in ihrer mechanischen und maschinennahen Steuerungsfunktion getestet. Damit kann die Materialflussanlage vor Ort schneller mit einem geringeren Fehlerisiko aufgebaut werden. Treten trotzdem noch Fehlfunktionen bei maschinennahen Funktionen auf, so sind diese leicht zu lokalisieren, denn die Auswirkungen bleiben auf das gestörte Modul beschränkt.

Insgesamt wird der Aufbau der Anlagenkomponenten nicht mehr sequentiell, sondern parallel durchgeführt. Somit können die Module in ersten Anlagenteilen miteinander gekoppelt und in ihrem Zusammenspiel getestet werden, während andere noch aufgebaut werden. Auch der Kopplungstest wird durch die hohe Anzahl verwendeter Standardkomponenten und damit Standardschnittstellen vereinfacht. Diese Arbeit wird bei klassischen Systemen erst in der Inbetriebnahme-/Hochlaufphase ausgeführt, kann im Internet der Dinge aber schon früher beginnen. Insgesamt wird die Mehrzahl der Arbeitsumfänge in der Realisierungsphase mit geringerem Aufwand und in kürzerer Zeit abgeschlossen werden können, als bei herkömmlichen Systemen.

6.3 Inbetriebnahme- /Hochlaufphase

Auch in der Inbetriebnahme- /Hochlaufphase (siehe Kap. 2.2.1.3) sind hauptsächlich positive Auswirkungen des Internet der Dinge-Konzepts zu erwarten, bestimmte Arbeitsumfänge werden jedoch mehr Aufwand verursachen, als bisher (siehe Abb. 6.3).

Die Einzelinbetriebnahme, die Offline-Abnahme und der Kopplungstest werden genauso wie der Anlagenaufbau parallel für die verschiedenen Anlagenbereiche durch-

geführt. Der Hersteller der mechatronischen Module nimmt diese vor der Auslieferung in Betrieb, so dass der Aufwand vor Ort sinkt.

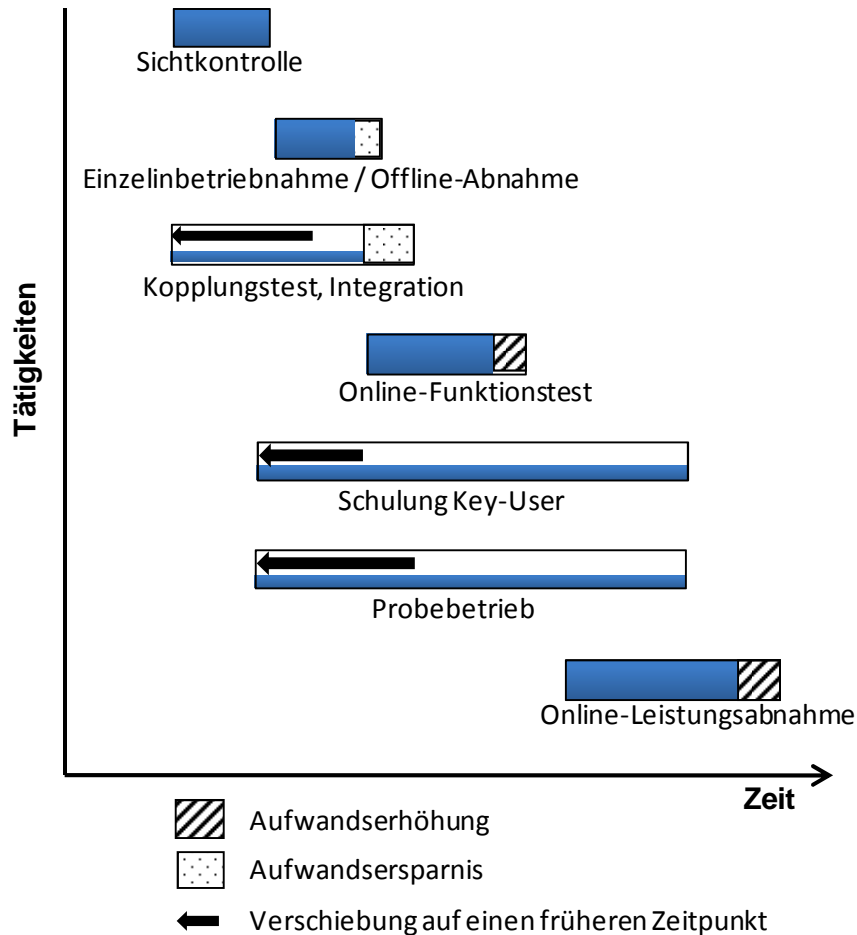


Abb. 6.3: Erwartete Veränderung des Zeitaufwands in der Inbetriebnahme-/Hochlaufphase (qualitativ)

Der beim Online-Funktionstest und der späteren Online-Leistungsabnahme entstehende Aufwand ist stark von der Verfügbarkeit geeigneter Hilfsmittel für Fehlersuche und Fehlerbehebung abhängig. Es kann jedoch davon ausgegangen werden, dass Fehler in der Strategieimplementierung und in der Kommunikation schwerer nachzuvollziehen sein werden, als in einem hierarchisch aufgebauten System. Dies liegt einerseits daran, dass in verteilten Systemen prinzipiell mehr kommuniziert wird und andererseits am hohen Anteil parallel getroffener Entscheidungen. Die negativen Auswirkungen können in Grenzen gehalten werden, indem die Anlagenfunktionen vor der Inbetriebnahme intensiv simulativ geprüft und Hilfsmittel zu Fehlersuche und

-behebung eingesetzt werden, welche speziell auf verteilte Systeme zugeschnitten sind (siehe Kap. 4.4.3).

Die Schulung der Key-User verschiebt sich im Vergleich zu herkömmlichen Systemen nach vorne. Dies wird durch zwei Umstände begünstigt: Erstens können die Steuerungsprogramme im Internet der Dinge auch in einer Simulation verwendet werden. Daher steht in der Regel ein detailliertes Simulationsmodell aus der Planungs- bzw. Realisierungsphase zur Verfügung. Und zweitens sind die Module schon bei Auslieferung mit ihrer eigenen Visualisierungsumgebung ausgestattet (siehe Kap. 4.4.2). Diese Visualisierung kann für eine Schulung verwendet werden.

Auch der Probetrieb der Anlage kann daher früher begonnen werden als bisher. Dieser wird erst für einzelne Anlagenteile durchgeführt und schrittweise auf Gesamtanlage ausgedehnt.

Zusammenfassend kann gesagt werden, dass sich die Inbetriebnahme- /Hochlaufphase ebenso wie die voran gegangenen Phasen verkürzen wird. Hersteller, welche sich für die Realisierung von Anlagen nach dem Internet der Dinge-Konzept entscheiden, sollten jedoch besonderes Augenmerk auf den frühen Einsatz der Simulation legen und sich im Vorfeld intensiv mit der Entwicklung bzw. Auswahl von Hilfsmitteln zur Fehlersuche und -behebung auseinandersetzen.

6.4 Betrieb

Im normalen Betrieb darf der Nutzer einer Internet der Dinge-Materialflussanlage keinen Unterschied zu herkömmlichen Systemen bemerken. Die Benutzerschnittstellen muss genau so zu bedienen sein, wie bisher. Seine Vorteile kann das neue Konzept ausspielen, wenn unerwartete Anforderungen auftreten oder Wartungsarbeiten anstehen. Die Materialflussmodule sind so gestaltet, dass sie auf neue Anforderungen selbstständig reagieren und beispielsweise bei der Überlastung oder Störung einer Wegverbindung automatisch neue Wege suchen, welche das gestörte Modul umgehen.

Die Auswirkungen von Fehlern und Ausfällen werden begrenzt und die Wahrscheinlichkeit eines Totalausfalls des Systems verringert sich, da im Gegensatz zu herkömmlichen Systemen keine Zentralkomponenten wie z.B. ein Materialflussrechner

oder eine Bereichs-SPS mehr vorhanden sind. Für Wartungsarbeiten können Module oder ganze Anlagenbereiche außer Betrieb genommen werden, ohne dass das restliche System abgeschaltet werden muss.

Das Wartungspersonal des Anlagenerstellers muss zwar höher qualifiziert sein als bisher, da sowohl Kenntnisse in agentenbasierter Hochsprachenprogrammierung als auch im Umgang mit Maschinensteuerungssprachen gefordert sind. Die Anzahl der hier noch auftretenden Fehler sollte jedoch niedriger sein als bisher, da die Module vor ihrem Einsatz intensiven Tests unterzogen wurden. Aus diesem Grund kann auch hier eine positive Bilanz gezogen werden.

6.5 Erweiterung / Modernisierung

Für die Erweiterung oder Modernisierung (siehe Kap. 2.2.1.5) eines Internet der Dinge-Materialflusssystemes kann aufwandsarm ein aktuelles Simulationsmodell erstellt werden, da die Steuerungslogik der bestehenden Module als Grundlage dienen können. Die Steuerungslogik neuer Anlagenteile kann dann in dieses Modell ergänzt und ausführlich getestet werden. Die Emulation erlaubt, die Steuerung des neuen Anlagenteils in die bestehende Anlage zu integrieren, ohne dass dieser Teil vorhanden ist (siehe Kap. 4.4.1).

Da alle Steuerungsstrategien unabhängig von der Anzahl der beteiligten Module entwickelt werden (siehe Kap. 4.3.2), müssen an der bestehenden Anlagenprogrammierung keine Änderungen vorgenommen werden. Der neue Anlagenteil wird aufgebaut, ohne dass der Betrieb der bestehenden Anlage unterbrochen werden muss. Auch eine von der restlichen Anlage getrennte Inbetriebnahme ist möglich. Der Eingriff in das komplexe Steuerungsprogramm eines Materialflussrechners entfällt.

Prinzipiell ist auch denkbar, eine herkömmliche Anlage, welche nach einem hierarchischen Ebenenmodell, z.B. nach VDMA 15276 (siehe Kap. 2.1.1), strukturiert wurde, zusammen mit Modulen des Internet der Dinge zu betreiben bzw. bestehende Fördertechnik aufzurüsten. Hierzu muss eine Schnittstelle geschaffen werden, welche die Kommunikation des alten Systems für die neue Steuerung übersetzt. Dies kann ein speziell dafür zu entwickelnder Softwaredienst übernehmen, welcher eine Entität im Sinne des Internet der Dinge darstellt. Ist eine solche Schnittstelle geschaf-

fen, so können beispielsweise Teile der Anlage oder Erweiterungen derselben nach dem Internet der Dinge-Konzept betrieben werden, während die restliche Anlage noch herkömmlich gesteuert wird.

Die Entscheidung für eine solche Mischform sollte jedoch gründlich überlegt sein. Die verteilte Steuerung wird ihr Potenzial hier nicht komplett ausspielen können. Außerdem verkompliziert die Kombination beider Konzepte die Anlagensteuerung und benötigt für Aufbau und Wartung Experten, welche sich in beiden Denkweisen zu Hause fühlen.

7 Zusammenfassung und Ausblick

7.1 Zusammenfassung der Ergebnisse

Die Anforderungen an produzierende Unternehmen haben sich in den letzten Jahren stark verändert. Heute verlangt der Kunde immer noch nach qualitativ hochwertigen, aber verstärkt auch individuellen Produkten, welche innerhalb kürzester Zeit und zu einem niedrigen Preis geliefert werden sollen.

Deutsche Unternehmen konkurrieren dabei auf einem weltweiten Markt mit Schwellenländern wie China oder Indien. Diese dringen mittlerweile auch in das Segment der qualitativ hochwertigen Erzeugnisse vor, welches bisher den westlichen Industrienationen vorbehalten war. Die im weltweiten Vergleich hohen Löhne zwingen die deutschen Unternehmen dazu, ihre Waren in hochautomatisierten Produktionsstätten herzustellen, jedoch werden heutige Automatisierungslösungen den neuen Anforderungen nicht mehr gerecht.

Denn die geforderte Individualität führt zu einer höheren Anzahl an Produktvarianten, welche in kleinen Losgrößen produziert werden. Dies vervielfacht die Anzahl der innerbetrieblichen Transporte und erhöht die Komplexität der dazu notwendigen Materialflusssteuerung. Zudem müssen Veränderungen im Produktspektrum oder Anlagenerweiterungen schneller als bisher umsetzbar sein, um rechtzeitig auf eine neue Marktentwicklung reagieren zu können.

Dabei reicht Flexibilität alleine nicht mehr aus, da diese nur vorgedachte Szenarien berücksichtigt. Zukünftige Systeme müssen auch auf ungeplante Ereignisse reagieren können, also wandelbar sein.

Aktuelle Materialflusssysteme haben jedoch schon jetzt einen Komplexitätsgrad erreicht, welcher nur noch schwer handhabbar ist. Sie sind daher auf eine schnelle Umstellung ihrer Prozesse nicht vorbereitet. Dies liegt unter anderem daran, dass diese Materialflusssysteme in einer hierarchischen Ebenenstruktur organisiert sind. An deren Spitze steht ein Materialflussrechner. Er steuert die Abläufe des Systems. Veränderungen an einem Teilsystem führen meist auch zu einem Eingriff in die Programmierung dieses Rechners. Dies kann sehr aufwändig werden, da seine Soft-

ware in der Regel sehr umfangreich ist und viele Wechselwirkungen mit den unterlagerten Steuerungsebenen existieren.

Diese hohe Komplexität wirkt sich auf alle Tätigkeiten bei der Realisierung einer Materialflussanlage aus. Gerade in späten Phasen lässt sich ein fehlerhaftes Zusammenspiel der Materialflusselemente nur noch mit großem Aufwand beheben. Die projektspezifische Auslegung der Steuerungsarchitektur hat zudem zur Folge, dass einzelne Arbeitsumfänge wie Steuerungsprogramme oder Simulationsmodelle nur schwer wiederverwendbar sind. Viele Arbeitsumfänge müssen daher für jede Anlagenrealisierung wieder von Neuem abgearbeitet werden.

Mittlerweile stehen jedoch Technologien zur Verfügung, welche zur Basis einer neuen Architektur für die innerbetriebliche Logistik werden können:

Die Steuerungskomponenten in der Automatisierungstechnik sind in den letzten Jahren kleiner, leistungsfähiger und preisgünstiger geworden. So kann immer mehr Steuerungslogik vom Materialflussrechner auf die unterlagerten Steuerungsebenen verlagert werden. Auf diese Weise lässt sich die Steuerungsaufgabe dezentralisieren. Indem sie in überschaubare Teile gegliedert wird, reduziert sich die Komplexität des gesamten Systems. Begünstigt wird diese Entwicklung durch die Technologie der IPC, welche die Ausführung von Maschinensteuerungs- und Hochsprachenprogrammen auf einem Gerät ermöglichen.

Die Radio Frequenz Identifikation, eine Identtechnologie zum drahtlosen Lesen und Schreiben von kleinen, elektronischen Datenträgern (Transpondern), kann dazu eingesetzt werden, Daten direkt am Produkt zu speichern. Auf diese Weise kann ein produktbegleitender Informationsfluss ohne zentrale Datenbank realisiert werden.

Mit der Internet-Technologie wurden zahlreiche Übertragungsprotokolle und preiswerte Geräte zur Vernetzung verteilter Einheiten entwickelt, die auch zur dezentralen Organisation eines Materialflusssysteme eingesetzt werden können. Zudem wurden speziell für die Automatisierungstechnik auf den TCP/IP-Standard des Internet aufbauende echtzeitfähige Kommunikationsprotokolle entwickelt, mit welchen nun auch steuerungstechnische Komponenten vernetzt werden können.

Multiagentensysteme, die ihren Ursprung in der Forschung zur verteilten, künstlichen Intelligenz haben, werden inzwischen auch zur Steuerung technischer Prozesse

eingesetzt. Mit dieser Technologie kann den Einheiten eines dezentral gesteuerten Materialflusssystem die notwendige Intelligenz gegeben werden.

Diese Entwicklungen sind die Grundlage für die Realisierung eines Internet der Dinge, in welchem autonome und intelligente Materialflussmodule die Transporteinheiten auf Basis dezentraler Entscheidungen zu ihrem Ziel transportieren - ähnlich Routern und Datenleitungen, die eine E-Mail weiterleiten. Diese Transporteinheiten tragen zu diesem Zweck einen RFID-Transponder, auf welchem, wie im Header einer E-Mail, das Ziel gespeichert ist.

Ein Internet der Dinge-Materialflusssystem kann aus drei Typen von autonomen Einheiten (Entitäten) aufgebaut werden. Das sind Fördertechnikmodule, Transporteinheiten und Dienste, welche jeweils über eine eigene Intelligenz in Form eines Softwareagenten verfügen. Dienste sind dabei reine Softwareprogramme. Sie übernehmen Aufgaben, die nicht einem einzelnen Modul oder einer Transporteinheit zugeordnet werden können.

Die vorliegende Arbeit setzt ihren Schwerpunkt auf die Konzeption und Entwicklung der Fördertechnikmodule. Diese Module werden nach den Vorgaben der funktionsorientierten Modularisierung mechatronisch gestaltet. Sie lassen sich in vier Klassen einordnen, welche jeweils die gleichen logistischen Grundfunktionen zu erfüllen haben. Das sind erstens Unstetigförderer, zweitens Stetigförderer und Schienen, drittens Verzweigungen bzw. Zusammenführungen und viertens Arbeitsstationen.

Alle Module verfügen über eine steuerungstechnische und eine mechanische Ebene. Die mechanische Ebene umfasst die mechanische Konstruktion sowie alle Sensoren und Aktoren. Auf der steuerungstechnischen Ebene befinden sich die Sicherheitssteuerung, die strategische und die operative Steuerung.

Die strategische Steuerung übernimmt dabei alle dispositiven Aufgaben und kommuniziert über eine Ethernet-Schnittstelle mit anderen Modulen. Zu diesem Zweck wird sie in einer Hochsprache wie C# oder Java implementiert.

Die operative Steuerung übernimmt alle maschinennahen Aufgaben, wie z.B. die Regelung von Antrieben, während die Sicherheitssteuerung alle sicherheitskritischen Vorgänge überwacht. Für diese beiden eignet sich am besten eine Programmierung in IEC 61131-3 Maschinensteuerungssprachen.

Die gesamte Steuerungslogik kann auf einem IPC ausgeführt werden, welcher sowohl über ein Betriebssystem für die Hochsprachenprogrammierung als auch über ein Laufzeitsystem für Maschinensteuerungssprachen, eine sogenannte Soft-SPS, verfügt.

Die strategische Steuerungsprogrammierung wird durch ein Vererbungsmodell und Möglichkeiten zur Konfigurierung bzw. Parametrierung ergänzt. So erhöht sich die Wiederverwendbarkeit der einzelnen Programmteile. Bei der Entwicklung von Materialflusstategien wird einheitlich vorgegangen, indem iterativ neue Verhaltensregeln implementiert und simulativ überprüft werden.

Ein dezentral gesteuertes System muss über spezielle Methoden zur Funktionsabsicherung verfügen. Werden die operativen Steuerungsfunktionen mit einem Softwareagenten emuliert, so kann die strategische Steuerung unabhängig von der realen Anlage getestet werden. Vorhandene Systeme können virtuell um neue Anlagenteile erweitert werden.

Eine Visualisierungsumgebung bietet den Entwicklern und Betreibern jederzeit einen Überblick über das aktuelle Anlagenverhalten und implementiert spezielle Hilfsmittel zur Fehlersuche und -behebung, die sie beim Auffinden von Fehlerursachen unterstützen.

Das Konzept des Internet der Dinge wurde am Lehrstuhl für Fördertechnik Materialfluss Logistik anhand einer Elektrohängebahnanlage, eines Depalettierroboters und eines Stetigförderers realisiert. Dabei konnte der grundsätzliche Funktionsnachweis für dieses Konzept erbracht werden.

Der hohe Wiederverwendungsgrad einmal gefundener Lösungen, die automatische Reaktion auf unvorhergesehene Ereignisse und die leichte Erweiterbarkeit werden den Zeitaufwand für die Realisierung von Materialflussanlagen erheblich reduzieren. Das Konzept des Internet der Dinge beschreibt einen Paradigmenwechsel in der Intralogistik: Zukunftsfähige Materialflussanlagen werden zu niedrigeren Preisen möglich.

7.2 Ausblick

Das in dieser Arbeit vorgestellte Konzept beschreibt den Aufbau und die wichtigsten Vorgehensweisen zur Gestaltung von Fördertechnikmodulen für das Internet der Dinge. Nun sind die Hersteller von Materialflusssystemen gefragt, diese Vorgaben in industrietaugliche Produkte zu überführen. Damit das Internet der Dinge seinen vollen Nutzen entfalten kann, müssen einheitliche und herstellerübergreifende Standards geschaffen werden, welche eine Kombination von Fördertechnik verschiedener Hersteller in einem Materialflusssystem erlauben. Nur so wird sich beispielsweise die Koordination von Zulieferern vereinfachen und eine Vergleichbarkeit von Preisen erzielt. Diese Zulieferer werden sich dann von reinen Mechaniklieferanten zu Herstellern mechatronischer Module entwickeln und die Entwicklung der Modulsteuerung selbst mit übernehmen können.

Derzeit verfügbare Programmierumgebungen unterstützen die Entwicklung dezentral gesteuerter Systeme nicht ausreichend. Daher wären spezielle Internet der Dinge-Entwicklungssysteme wünschenswert, um in diesen verteilten Systemen die Fehlersuche und -behebung zu vereinfachen.

Ein Produktdatenmanagementsystem, das die Verwaltung aller zu einem Modul gehörenden Daten und die automatische Erzeugung von Produktvarianten unterstützt, könnte die Zusammenstellung neuer Systeme aus vorhandenen Komponenten erleichtern. Dazu sollte es Schnittstellen zur Entwicklungs-, Simulations- und Visualisierungsumgebung bereitstellen.

Die Fähigkeit zur schnellen Anpassung an neue Anforderungen ist ein zentraler Aspekt des Internet der Dinge-Konzepts. Schnelle Zukunftsprognosen über die Auswirkungen einer Veränderung der Prozesse sind dabei eine wichtige Hilfe für den Betreiber, um rechtzeitig geeignete Maßnahmen zu ergreifen. Da die Fördertechnikmodule ihre Verhaltensweisen nach einfachen Regeln koordinieren, ist eine schnelle Simulationsrechnung noch während des Betriebs denkbar. Auf diese Weise kann eine bessere Entscheidungsgrundlage für den Betreiber geschaffen werden.

Es ist sogar denkbar, dass die Fördertechnikmodule selbstständig und im regulären Betrieb ihre nächsten Schritte in Voraus simulieren. Auf diese Weise könnten sie die Auswirkungen von Handlungsalternativen früher abschätzen und eine bessere Leistung erzielen. Dann gilt: Die Technik denkt mit.

8 Literaturverzeichnis

- [Bal-94] Balzert, H.:
Die Entwicklung von Software-Systemen: Prinzipien, Methoden, Sprachen, Werkzeuge. Mannheim, 1994.
- [Ber-08] Bemeleit, B., Baalsrud Hauge, J., Thoben, K.-D.:
Risikomanagement für selbststeuernde logistische Transportprozesse; 4. Wissenschaftssymposium der BVL, 11. - 12. Juni 2008, München, S. 323 – 346
- [Bie-00] Biethahn, J.; Mucksch, H.; Ruf, W.:
Ganzheitliches Informationsmanagement: Entwicklungsmanagement, München, 2000.
- [Bir-07] Birkhahn, C.:
Smart Production Systems - Intelligente Konzepte zur Gestaltung von Produktionssystemen, Diss., Technische Universität Kaiserslautern, Kaiserslautern, 2007
- [Ble-99] Blessing, S.:
Gestaltung der Materialflusssteuerung in dynamischen Produktionsstrukturen. Herbert Utz Verlag Wissenschaft, München, 1999 (iwb Forschungsberichte 134)
- [Cof-71] Coffman, E.G., Elphick, M.J., Shoshani, A.:
System Deadlocks, ACM Computing Surveys, 3(2), 67-78, 1971
- [Dae-94] Daenzer, W.F.; Huber, F.:
Systems Engineering – Methoden und Praxis, Verlag Industrielle Organisation, Zürich, 1994
- [Dan-09] Daniluk D., Chisu R.:
Simulation und Emulation im Internet der Dinge
In: Internet der Dinge in der Intralogistik, Springer, Berlin, 2009

- [Dem-05] Dembski, N.; Timm, I. J.:
Contradictions between Strategic Management and Operational Decision-Making - Impacts of Autonomous Processes to Decision-Making in Logistics. In: Palwar, K. S. et al. (eds.): Innovations in Global Supply Chain Networks. Proceedings of the 10th International Symposium on Logistics. University of Nottingham Business School, 2005, pp. 650-655.
- [Die-03] Diekmann, T.; Hagenhoff S.:
Ubiquitous Computing: State of the Art, Arbeitsbericht Nr. 24/2003, Institut für Wirtschaftsinformatik, Georg-August-Universität, Göttingen, 2003
- [Dij-59] Dijkstra, E. W.:
A note on two problems in connexion with graphs, Numer. Math., S.269-271, 1959
- [DIN-15201] Deutsches Institut für Normung:
DIN 15201-1 Stetigförderer: Benennungen, Beuth-Verlag, Berlin, 1994
- [Dör-08] Dörner, D.:
Die Logik des Misslingens: strategisches Denken in komplexen Situationen. Rowohlt, 2008
- [Dum-00] Dumke, R.; Koepe, R.; Wille, C.:
Software Agent Measurement and Self-Measuring Agent-Based Systems. Preprint Nr. 11, Fakultät für Informatik, Otto-von-Guericke-Universität Magdeburg, 2000
- [Eis-01] Eisenschmid, M.:
Supply-Chain-Collaboration bei Daimler-Chrysler, in: Innovative Logistik in der Automobilindustrie, VDI-Bericht 1628. Düsseldorf: VDI Verlag, 2001

- [End-03] Enderlein, H.; Hildebrand, T.; Müller, E.:
Plug+Produce – Die Fabrik mit Zukunft aus dem Baukasten, wt
Werkstattstechnik 93 (2003) H.4 S. 282-286.
- [Etz-95] Etzioni, O.; Weld, D. S.:
Intelligent Agents on the Internet: Fact, Fiction, and Forecast; IEEE
Expert – Intelligent Systems & their Applications, August 1995, S. 44-
47, 1995
- [Eym-00] Eymann T.:
AVALANCHE - Ein agentenbasierter dezentraler Koordinations-
mechanismus für elektronische Märkte, Dissertation, Albert-Ludwigs-
Universität Freiburg im Breisgau, 2000
- [Fer-99] Ferber, J.:
Multi-Agent Systems. Addison-Wesley, Harlow, England, 1999
- [FIPA-05] FIPA Standards Committee (FIPA SC), Policies and Procedures;
Institute of Electrical and Electronics Engineers (IEEE), 2005
- [Fin-06] Finkenzeller K.:
RFID Handbuch, Hanser Verlag, München, ISBN: 3-446-40398-1,
2006, S.439
- [Fis-06] Fischer, H.:
Ein systemorientierter Ansatz zur Modularisierung von Planspielen
mit dem Ziel der Komplexitätssteuerung und Integration in Standard-
software, Diss., Universität Göttingen, 2006
- [FML-09] Logistikkompodium des Lehrstuhls für Fördertechnik Materialfluss
Logistik fml, Technische Universität München:
Unstetigförderer, www.fml.mw.tum.de/fml/index.php?Set_ID=319,
Abgerufen am 17.09.2009
- [Fou-94] Fournier, G.:
Informationstechnologien in Wirtschaft und Gesellschaft, Heidelberg:
Springer 1994, ISBN 342808098, S. 59

- [Fre-04] Freitag, M.; Herzog, O.; Scholz-Reiter, B.:
Selbststeuerung logistischer Prozesse – Ein Paradigmenwechsel und seine Grenzen. In: *Industrie Management*, 20(2004)1, S. 23-27
- [Ger-04] Gerber A.:
Flexible Kooperation zwischen Autonomen Agenten in Dynamischen Umgebungen, Dissertation, Universität des Saarlandes, 2004
- [Gol-99] Gollos, M.:
Produktionssysteme und Materialfluss schnell und kostengünstig an kurze Produktlebenszyklen anpassen. In: *Logistik im Unternehmen* 13 (1999), Nr. 10, S. 52-53.
- [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
- [Gud-99] Gudehus, T.:
Logistik – Grundlagen, Strategien, Anwendungen. Berlin: Springer Verlag, 1999
- [Gün-98] Günthner, W. A.; Bambynek, A.:
Integrierbarkeit eines flurfreien Leichtfördersystems: Schnittstellen-Baukasten. In: *Hebezeuge und Fördermittel* 38 (1998) Nr. 12, S. 593-595.
- [Gün-00a] Günthner, W. A.; Reinhart, G. (Hrsg.):
Abschlussbericht MATVAR – Materialflusssysteme für variable Fertigungssegmente im dynamischen Produktionsumfeld, Herbert Utz Verlag Wissenschaft, München 2000.
- [Gün-00b] Günthner, W. A.:
Verbundforschungsprojekt MATVAR – Wege zum wandelbaren Materialflussnetz. In: *Hebezeuge und Fördermittel*. Berlin 40 (2000) 5, S. 267-270.

- [Gün-02] Günthner, W. A.; Heinecker, M.; Wilke, M.:
Materialflusssysteme für wandelbare Fabrikstrukturen, Industrie Management 18 (2002) 5, GITO mbH Verlag für Industrielle Informationstechnik und Organisation, Berlin, 2002
- [Gün-03] Günthner, W. A.; Wilke, M.:
Mass Customization requires changeable material flow systems, Interdisciplinary World Congress on Mass Customization and Personalization (MCPC2003), Oct. 2003
- [Gün-05a] Günthner W. A., Boppert J., Schedlbauer M.:
RFID - Es funkt gewaltig, Zukunft im Brennpunkt; Ausgabe 4-2005, S. 25-30
- [Gün-05b] Günthner, W. A.; Wilke, M.:
Mit neuen Lösungen zu wandelbaren Materialflusssystemen Schweizer Jahrbuch für Logistik 2005
- [Gün-08] Günthner, W. A.; Chisu, R.; Kuzmany, F.:
Internet der Dinge - Intelligent verteilt. In: F+H Fördern und Heben, 7 (2008), S. 422-425
- [Gün-08a] Günthner, W. A.; Chisu, R.; Kuzmany, F.:
Internet der Dinge - Zukunftstechnologie mit Kostenvorteil, F+H Fördern und Heben, Ausgabe Oktober 2008, S. 556-558
- [Gün-08b] Günthner, W. A.:
Vorlesung „Materialfluss und Logistik“, Sommersemester 2008, TU München, 2008
- [Han-01] Handrich W.:
Flexible, flurfreie Materialflusstechnik für dynamische Produktionsstrukturen (Dissertation), Herbert Utz Verlag, München, 2001
- [Han-06] Handley, M.:
Why the Internet only just works, BT Technology Journal, Vol 24 No 3, Springer Netherlands, Dordrecht, 2006

- [Hei-06] Heinecker, M.:
Methodik zur Gestaltung und Bewertung wandelbarer Materialflusssysteme, Diss., TU München; Herbert Utz Verlag München, ISBN 3-8316-0620-X, 2006
- [Her-03] Hertzberg J., Ritter A., Bredenfeld A., Mayer J., Schöll P., Schönherr F., Staab H., Worst R.:
Agententechnologie zur Integration heterogener Systeme am Beispiel eines Prototypen für ein Kommissioniersystem
in: SPS/IPC/DRIVES 2003 : Elektrische Automatisierung, Systeme und Komponenten, Fachmesse & Kongress 25.-27.11.2003 Nürnberg, MESAGO Messe Frankfurt GmbH, Stuttgart, VDE Verlag, Berlin, 2003
- [Hil-05] Hildebrand, T.; Mäding, K.; Günther U.:
Plug+Produce – Gestaltungsregeln für die wandlungsfähige Fabrik. Institut für Print- und Medientechnik. TU Chemnitz, 2005
- [Hom-08] ten Hompel, M.; Follert G.:
Internet der Dinge – Autonomie im serviceorientierten Materialfluss, Wissenschaft und Praxis im Dialog – Robuste und sichere Logistiksysteme, Deutscher Verkehrs-Verlag DVV, Hamburg, 2008, ISBN 9783871543817
- [Huh-99] Huhns, M. N.; Stephens, L. M.:
Multiagent Systems and Societies of Agents
In: Weiß, G., editor (1999b). Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence, The MIT Press, Cambridge, Massachusetts
- [IEC 61131-3] IEC 61131-3 Programmable controllers - Part 3:
Programming languages. VDE Verlag GmbH, Berlin, 1993
- [IEEE 802.3] Institute of Electrical and Electronics Engineers:
802.3-1998 IEEE Std 802.3, 1998 Edition

- [IEEE 802.11] Institute of Electrical and Electronics Engineers:
802.11k-2008 IEEE Standard for Information Technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, New York, 2008
- [IEEE-1588] Institute of Electrical and Electronics Engineers:
IEEE Std 1588-2002: IEEE Standard for a Precision Clock Synchronisation Protocol for Networked Measurement and Control Systems, New York, 2002
- [Ise-99] Isermann, R.:
Mechatronische Systeme – Grundlagen, Springer-Verlag, Berlin, 1999
- [ISO-7498] International Organization for Standardization (ISO):
ISO/IEC 7498-1:1994 Information technology -- Open Systems Interconnection -- Basic Reference Model: The Basic Model, 1994
- [ITU-05] ITU-Internet Reports 2005 – Internet of Things Executive Summary:
International Telecommunication Union, Genf, 2005
<http://www.itu.int/osg/spu/publications/internetofthings/>, Abgerufen am 27.08.09
- [Jad-09] Java Agent Development Framework:
<http://jade.tilab.com/>, Abgerufen am 07.09.2009
- [Jed-06] Jedermann R., Gehrke J.; Lorenz M.; Herzog O., Lang W.:
Realisierung lokaler Selbststeuerung in Echtzeit - Der Übergang zum intelligenten Container; Dortmund, 3. Wissenschaftssymposium Logistik der BVL, 30.-31.05.2006, Schriftenreihe Wirtschaft & Logistik, Deutscher Verkehrs-Verlag, Hamburg, 2006, ISBN 3871543403

- [Kar-98] Karl, A.:
Wandlungsfähige Materialflusstechnik. In: F+H Fördern und Heben 48 Nr. 4, S. 270-273, 1989
- [Kim-97] Kim C.O., Kim, S.S.:
An efficient real-time deadlock-free control algorithm for automated manufacturing systems, International Journal of Production Research, Issue 6 Volume 35, London, 1997
- [Kle-96] Kleve, H.:
Soziale Arbeit als wissenschaftliche Praxis und als praktische Wissenschaft. Systemtheoretische Ansätze einer Praxistheorie Sozialer Arbeit.
In: Neue Praxis, 26. Jg. S. 245 – 252., Verlag neue praxis GmbH, Lahnstein, 1996
- [Koh-07] Kohn N.:
Kommunikations-Infrastruktur für hochdynamische Parallelroboter, Dissertation, Technische Universität Carolo-Wilhelmina, Braunschweig, 2007
- [Lan-99] Langer, G.:
HoMuCS – A methodology and architecture for Holonic Multi-cell Control Systems, Technical University of Denmark, Dissertation 1999
- [Liu-01] Liu, F-H.; Hung, P-C.:
Real-time deadlock-free control strategy for single multi-load automated guided vehicle on a job shop manufacturing system, International Journal of Production Research, Volume 39 Issue 7, London, 2001
- [Mal-06] Malik, F.:
Strategie des Managements komplexer Systeme: ein Beitrag zur Management-Kybernetik evolutionärer Systeme. Haupt, Bern; Stuttgart, Wien, 9., unveränd. Aufl.

- [May-09] Mayer S.:
Development of a completely decentralized control system for modular continuous conveyors, Dissertation, Universität Karlsruhe, 2009
- [Mes-90] Mesina M.; Bartz, W.; Wippler, E.:
CIM-Einführung: Rationalisierungschancen durch die Anschaffung und Integration von CA-Komponenten In: Das Handbuch für Ingenieure, 3. Ausgabe, Ehningen, expert Verlag, 1990
- [Moo-65] Moore G.:
Cramming more components onto integrated circuits, Electronics 38, 1965 S. 114-117
In: B. Batinic, Internet für Psychologen, Hogrefe, Göttingen, 1997
- [Mus-01] Muscholl K.:
Interaktion und Koordination in Multiagentensystemen, Dissertation, Universität Stuttgart, 2001
- [Mus-97] Musch J.:
Die Geschichte des Netzes: ein historischer Abriß
- [Nie-09] Nieke C., Henk M.:
Die Erstellung eines Baukastens für das Internet der Dinge
In: Internet der Dinge in der Intralogistik, Springer, Berlin, 2009
- [Noa-90] Noack, M.; Wegner, K.; Gluch, D.; Dienhart, U.:
CIM-Integration und Vernetzung: Chancen und Risiken einer Innovationsstrategie, Berlin u.a.: Springer 1990
- [Ode-00] Odell, J.; Van Dyke, H.; Bauer, B.; Bauer, P.:
Extending UML for Agents; AOIS Workshop at AAAI 2000; URL: <http://aot.ce.unipr.it/auml/working/Odell-W001.pdf>

- [Ost-93] Ost, S.:
Entwicklung eines Verfahrens zur differentiellen Flexibilitätsanalyse und –bewertung. TU Hamburg-Harburg, Arbeitsbereich Werkzeugmaschinen und Automatisierungstechnik an der TUHH, 1993
- [Par-72] Parnas, D.:
On the Criteria To Be Used in Decomposing Systems into Modules. In: Communications of the ACM 15, Nr. 12, S.1053-1058., 1972
- [Rao-91] Rao, A. S.; Georgeff, M. P.:
Modeling rational agents within a BDI-Architecture. In: Technical Report 14, Australian AI Institute, Carlton, 1991
- [Rit-03] Ritter, A.:
Ein Multi-Agenten-System für mobile Einrichtungen in Produktionssystemen, Diss., Universität Stuttgart, Jost Jetter Verlag Heimsheim, ISBN 3-936947-07-4, 2003
- [Rei-06] Reichwald, R.; Piller F.:
Interaktive Wertschöpfung, Wiesbaden: Gabler 2006, ISBN 3-8349-0106-7, S. 23, 2006
- [Rei-99] Reinhard, G.: Wandel in der Montage – Flexibilität ist gut, Reaktionsfähigkeit noch besser, in: wt, Werkstattstechnik 89. Berlin: Springer Verlag, 1999
- [Sch-08a] Schiller, F.:
Vorlesung Automatisierungstechnik, Skript, Technische Universität München, Lehrstuhl für Informationstechnik im Maschinenwesen, München, 2008
- [Sch-08b] Schmitt, R.:
Vorlesung Qualitätsmanagement, Skript, RWTH Aachen, Lehrstuhl f. Fertigungsmesstechnik und Qualitätsmanagement, Aachen, 2008

- [Sch-08c] Schröder-Preikschat W.:
Vorlesung Betriebssysteme, Skript, Friedrich Alexander Universität
Erlangen-Nürnberg, Erlangen, 2008
- [Sch-93] Schwill, A.:
Objektorientierte Programmierung. Eine Rechtfertigung aus kogniti-
onspsychologischer Sicht. In: LOGIN, 13. Jg. (1993), H. 4, S. 44-45,
LOG IN Verlag GmbH, Berlin, 1993
- [Sch-98] Schuh, G.; Millarg, K.; Göransson, A.:
Virtuelle Fabrik. Neue Marktchancen durch dynamische Netzwerke.
Hanser Wiss., München, 1998
- [Sei-08] Seitz M.:
Speicherprogrammierbare Steuerungen, Hanser Fachbuchverlag,
München, 2008
- [SFB-467] [http://www.ipvs.uni-
stuttgart.de/abteilungen/vs/forschung/projekte/SFB_467/de](http://www.ipvs.uni-stuttgart.de/abteilungen/vs/forschung/projekte/SFB_467/de), Abgeru-
fen am 26.08.2009
- [Sta-07] Pressemitteilung des Statistischen Bundesamts Nr.486 vom
30.11.2007, Wiesbaden
- [Sua-96] Suarez F.; Cusumano M.; Fine G.:
Wie flexibel produziert Ihre Fabrik? Harvard Business Manager 2, S.
36-44, 1996
- [Tim-04] Timm I.:
Autonomie und dynamische Koordination in kooperativen Systemen
In: Florian, M.; Hillebrandt, F. (Hrsg.): Adaption und Lernen von und
in Organisationen, S. 103-127, VS Verlag für Sozialwissenschaften,
Wiesbaden, 2004
- [VDI-2206] Verein Deutscher Ingenieure (VDI) (Hrsg.):
VDI-Richtlinie 2206: Entwicklungsmethodik für mechatronische Sys-
teme, Beuth-Verlag, Berlin 2004

- [VDI-2221] Verein Deutscher Ingenieure (VDI) (Hrsg.):
VDI 2221: Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte, VDI Verlag Düsseldorf 1993
- [VDI-2516] Verein Deutscher Ingenieure (VDI) (Hrsg.):
Flurförderzeuge für die Regalbedienung - Spielzeitermittlung in Schmalgängen, Beuth-Verlag, Berlin 2003
- [VDI-3633] Verein Deutscher Ingenieure (VDI) (Hrsg.):
VDI 3633 Blatt 3 - Simulation von Logistik-, Materialfluss- und Produktionssystemen, Beuth-Verlag, Berlin 1997
- [VDI-3647] Verein Deutscher Ingenieure (VDI) (Hrsg.):
VDI 3647 Kleinbehälter-Fördersysteme Fahrwerksysteme, Beuth-Verlag, Berlin 1987
- [VDI-3961] Verein Deutscher Ingenieure (VDI) (Hrsg.):
VDI 3961: Planung der Materialflusssteuerung in Fertigungsbetrieben, Beuth-Verlag, Berlin 1989
- [VDI-5100] Verband deutscher Ingenieure (VDI), Verband deutscher Maschinen- und Anlagenbau e.V. (VDMA)
VDI/VDMA-Richtlinienentwurf 5100: Systemarchitektur für die Intralogistik (SAIL), Beuth-Verlag, Berlin 2008
- [VDMA-15276] Verband Deutscher Maschinen- und Anlagenbau(VDMA):
VDMA-Einheitsblatt 15276: Datenschnittstellen in Materialflusssystemen, Beuth-Verlag, Berlin 1994
- [Vol-96] Volberda, H. W.:
Towards the flexible firm: How to remain vital in hypercompetitive environments. *Org. Sc.*, Vol. 7 (1996) Nr. 4, S. 359–374
- [War-93] Warnecke, H. J.:
Revolution in der Unternehmenskultur: Das Fraktale Unternehmen. Berlin: Springer Verlag, 1993

- [Weh-04] Wehking, K.-H.:
Das materielle Internet - der Treiber für neue Logistikstrukturen und –
komponenten; 2. Wissenschaftssymposium Logistik der BVL, Berlin,
2004.
- [Weh-08] Wehking, K.-H.; Vorwerk C.:
Technische Basiskomponenten der Intralogistik im Wandel, Zukünf-
tig: Stückgutförderung mit Fahrzeugschwarm, Hebezeuge und För-
dermittel 05/2008, S.242 , Berlin: HUSS-MEDIEN GmbH
- [Wei-91] Mark Weiser. Mark Weiser's homepage. Available at
<http://www.ubiq.com/hypertext/weiser/> , 1991. Abgerufen am
20.08.2009
- [Wel-03] Wellner J.:
Selbstorganisierende Kommunikationssysteme für Multiagentensys-
teme, Dissertation, Universität Dortmund, 2003
- [Wes-00] Westkämper, E.; Zahn, E.; Balve, P.; Tilebein, M.:
Ansätze zur Wandlungsfähigkeit von Produktionsunternehmen, wt
Werkstatttechnik online Jahrgang 90 (2000) H. 1/2;S. 22-26
- [Wes-97] Westkämper, E.:
Wandlungsfähige Unternehmensstrukturen - Teil I, in: Logistik-
Spektrum Nr. 3. Mainz: Vereinigte Fachverlage, 1997
- [Wie- 98] Wiendahl, H.-P.; Hartmann, M.:
Produktion 2000plus. Visionen und Forschungsfelder für die Produk-
tion in Deutschland. Untersuchungsbericht zur Definition neuer For-
schungsfelder für die Produktion nach dem Jahr 2000, RWTH Aa-
chen, Aachen, 1998
- [Wil-06] Wilke, M.:
Wandelbare automatisierte Materialflusssysteme für dynamische
Produktionsstrukturen, Diss., TU München; Herbert Utz Verlag Mün-
chen, ISBN 3-8316-0591-2, 2006

- [Wom-92] Womack, J.P.; Jones, D.T.; Roos, D.:
Die zweite Revolution in der Autoindustrie. Frankfurt: Campus-Verlag, 1992
- [Woo-95] Woolridge, M.; Jennings, N. R. Intelligent Agents:
Theory and Practice; Knowledge Engineering Review, Vol. 10 (2);
Cambridge University Press, S. 115-152