



TECHNISCHE UNIVERSITÄT WIEN

DISSERTATION

ARBEITSZENARIOEN UND OPTIMEIRUNG VON ABLÄUFEN UND STEUERUNG
VON SELBSTORGANISIERENDEN BIONIC ASSEMBLY SYSTEM
IN CIM UMGEBUNG

Ausgeführt zum Zwecke der Erlangung des akademischen Grades
eines Doktors der technischen Wissenschaften unter der Leitung von

Univ. Prof. Dipl. -Ing. Dr. techn. Dr. mult. h.c. Branko Katalinic
E311
Institut für Fertigungstechnik

eingereicht an der Technischen Universität Wien
Fakultät für Maschinenbau

von

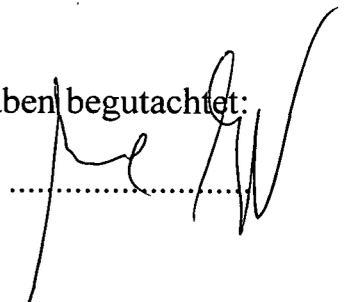
Dipl. -Ing. Vedran Kordic
Matr. Nr. 9927389
Gumpendorferstraße 39, 1060 Wien

Kordic V.

Wien, im September 2004

Diese Dissertation haben begutachtet:


.....


.....

Abstract

Further development of assembly systems in the context of Next Generation Production Systems initiative goes after the example of nature in the direction of self organization, autonomy and learning. Due to and in contexts of this trend the original concept of Bionic Assembly System is proposed. Horizontal and vertical integration lead to distributed and decentralized system solutions. These are based on distributed architectures and on the increase of the autonomy of single units. Idea behind the autonomising consists in limiting of complexity of the control and in creating prerequisites for a modular self-organizing assembly system. First part of this work is focused on the question: How the autonomy can be reached and enlarged as well as how to realize autonomous agents for use in the self-organizing assembly system. The result of this assessment is agent control architecture based on reactivity paradigm extended with task oriented requirements of the assembly system. Second part of this work represents concept, structure and function of the biologically motivating self-organizing assembly system. Assembly system is an integration of agent based control, autonomous agents of different types and functionalities and human personal. These components form a modular self-organising assembly system which is furthermore a part of the existing CIM system. Furthermore reactive scheduling is worked out and compared with conventional methods in the simulation. Advantage of this method is that the time and content of the negotiation can be changed so that the assembly system can autonomously reach optimal conditions.

Kurzfassung

Weiterentwicklung von Montagesystemen im Rahmen von Next Generation Production Systems initiative geht in die Richtung von Selbstorganisation, Autonomie und Lernen nach dem Vorbild der Natur. Aufgrund und in Rahmen dieses Trends wurde das originelle Konzept von Bionic Assembly System vorgeschlagen. Horizontale und vertikale Integration führen zu verteilten und dezentralen Systemlösungen. Diese basieren auf verteilten (Steuerungs-)Architekturen und auf die Zunahme der Autonomie einzelner Einheiten. Idee hinter der Autonomisierung besteht darin, Komplexität der Steuerung einzuschränken und Voraussetzungen für ein selbstorganisierendes Montagessystem zu schaffen. Erster Teil dieser Arbeit beschäftigt sich mit der Frage, wie Autonomie erreicht und erweitert werden kann sowie wie darauf aufbauend autonome Agenten für den Einsatz im selbstorganisierenden Montagesystem zu realisieren sind. Das Ergebnis dieser Untersuchung ist eine reaktiv basierende Agentensteuerungsarchitektur, die auf Anforderungen des Bionic Assembly System abgestimmt ist. Zweiter Teil dieser Arbeit stellt Konzept, Struktur und Funktion des biologisch motivierenden selbstorganisierenden Montagesystems dar. Dem Bionic Assembly System ist eine Integration von agentenbasierender Steuerung, autonomen Agenten unterschiedlicher Arten und Funktionalitäten und menschliche Operateure zugrunde gelegt. Diese Komponenten bilden ein modulares selbstorganisierendes Montagesystem, das weiterhin ein Teil der vorhandenen CIM Steuerung ist. Des Weiteren ist reaktive Ablaufplanung ausgearbeitet und in einer Simulation mit herkömmlichen Methoden verglichen. Vorteil dieser Methode ist es, dass der Zeitpunkt und damit der Inhalt der Verhandlungen verändert werden kann, sodass das Montagesystem autonom optimale Bedingungen erreichen kann.

Danksagung

Hiermit möchte ich mich bei Herrn Univ. Prof. Dipl. -Ing. Dr. techn. Dr. mult. h.c. Branko Katalinic für die geleistete Betreuung bedanken. Ihm ist herzlich gedankt für die Bereitstellung dieses interessanten Themas und den großen Freiraum bei dessen Bearbeitung.

Dieser Dank gilt auch für meinen Koreferent Herrn Univ. Prof. Dipl. -Ing. Dr. techn. Uwe Egly für die Begutachtung dieser Arbeit.

Von meinen Kollegen möchte ich einen herausgreifen, denen ich in besonderer Weise zu Dank verpflichtet bin Dr. techn. Dipl. -Ing. Kemajl Stuja.

Auch der freundlichen Korrekturleserin meiner Arbeit, Claudia Mijajlovic, sei hier ein großes Lob gespendet.

Dieser Dank gilt für meine Eltern ohne deren Verständnis die vorliegende Arbeit nicht hätte entstehen können.

Inhaltverzeichnis

1. Problem Stellung	1
1.1 Kapitel Struktur	3
2. Definitionen und Grundkonzepte	4
2.1 Agent	5
2.2 Autonomie	7
2.3 Intelligenz	8
2.4 Adaptivität	10
2.5 Verbindungen zwischen den Begriffen	10
2.7 Intentionaler Standpunkt	12
2.7.1 Handlung, Verhalten, Aktion	14
3. Verhalten in der realen Umgebung	17
3.1 Der Aufbau der Regelkreise	18
3.2 Umgebung-Aufgabe Kontext	22
3.3 Muster	26
3.4 Design von Agenten	28
3.4.1 Form	29
3.4.2 Antrieb	29
3.4.3 Sensorik	30
3.4.4 Energieversorgung	31
3.5 Zeit	32
4. Die Kontrolle des Agenten	35
4.1 Kognitive Kontrollparadigma	36
4.2 Reaktive Kontrollparadigma	38
4.3 Hybride Kontrollparadigma	39
4.4 Richtige Kontrolle für Agenten in der realen Umgebung	42
5. Reaktive Architekturen	45
5.1 Kompetitive Ansätze	46
5.2 Kooperative Ansätze	48
5.3 Die Felder	50
5.3.1 Von Skalar- bis zum Vektorfeld	51
5.3.2 Attraktoren	52

5.3.3 Magnitude Profil	54
5.3.4 Erzeugung des Potentialfeldes	59
5.3.5 Das Lokale Minima Problem	59
5.4 Agenten Architektur	60
5.4.1 Aufbau von Modulen	62
5.4.2 Welche Neuronale Netz Struktur eignet sich für die Bewegungskontrolle den Agenten	64
5.5 Simulation	68
6. Von klassischer bis agentbasierender Steuerung	72
6.1 Klassische Produktionssteuerung	74
6.2 Agentenbasierende Produktionssteuerung	75
6.3 Vorteile und Grenzen der Agentenbasierenden Steuerung	77
6.4 Klassische versus Agentenbasierenden Steuerung	78
6.5 Passende Domäne für agentenbasierenden Steuerung	79
7. Von klassischer bis selbstorganisierender Montage	81
7.1 Besonderheiten des Montagesystems	81
7.2 Die Montageaufgabe	82
7.3 Auswahl der Montageorganisation	83
7.3.1 Fließbandmontage	86
7.4 Montagesystem der IMS Group der TU Wien	86
7.4.1 Genauer Ablauf in den Montagestationen	91
7.5 Grenzen des Fließbandmontagekonzepts	93
7.6 Motivation für die selbstorganisierende Montage	94
7.7 Integration der agentenbasierenden Montage in CIM Struktur	101
7.8 Kommunikationskonzept	105
7.8.1 Zielansteuerung	108
7.8.2 Informationsaustausch innerhalb der Montagestation	114
7.8.3 Technische Realisation der Vertikalen Kommunikation und der Horizontalen Kommunikation innerhalb der angefahrenen Station	114
7.9 Transportagent Datenset	116
7.10 Ablauf Szenario der selbstorganisierende Montage	116
7.11 Ablaufszenario für die Agentenklassen	121
7.12 Koordination zwischen Agenten	126
8. Optimierung von Abläufen	134
8.1 Verhandlung als Scheduling in realer Zeit	135
8.2 Verhandlungsmethode	136
8.3 Vorteile der verwendeten Verhandlungsmethode	141
8.4 Simulationsversuche	143
8.4.1 Parameter für die Evaluierung des Montagesystems	144
8.4.2 Das Simulationsmodell	145
8.4.3 Simulationsannahmen	146
8.4.4 Ergebnisse	147
9. Rekonfiguration des Systems	151

9.1 Mathematisches Model der Rekonfiguration	155
9.1.1 Technologische Montagezeiten.....	155
9.1.2 Montagestation Auftrag	157
10. Schlussfolgerung	159
11. Abbildungsverzeichnis	162
12. Tabellenverzeichnis	164
13. Literaturverzeichnis	165
14. Lebenslauf	173

Problem Stellung

Nichts ist so beständig wie der Wandel. Für kaum einen anderen Bereich ist diese Aussage zutreffender als für unsere Industriegesellschaft. Dabei geht es weniger um die Erweiterung vorhandener Automatisierungs- und Produktionsverfahren, sondern viel mehr um innovative Produktionskonzepte.

Der Drang zu diesen neuen Produktionskonzepten ist allerdings weder Selbstzweck noch ausschließlich die logische Folge der technologischen Weiterentwicklung.

Die weltweit wachsenden Überkapazitäten, die Erschließung neuer Märkte in Rahmen der Globalisierung, rasche Entwicklung in der Produktion und Informationstechnologie und die steigenden Entwicklungskosten verschärfen den Wettbewerb zwischen den Produktionsunternehmen.

Aus diesen Rahmenbedingungen ergeben sich neue Herausforderungen. Zu diesen Herausforderungen zählen kürzere Einführungszeiten für neue Produkte und Technologien, eine steigende Variantenzahl, eine flexiblere Nutzung und Skalierung von Kapazitäten, die Reduzierung der Produktionskosten durch Wiederverwendung von Anlagen und die Erhöhung der Anlagenproduktivität durch Reduzierung von Störungen.

Um neue Produkte hervorzubringen bzw. erforderliche Änderungen bei vorhandenen durchzuführen, müssen neue Funktionen dem jetzigen Produktionssystem hinzugefügt werden. Einführung des Computergestützten Entwurfs hat drastisch Produktentwicklungszeiten reduziert. Jedoch existiert solche Methodologie für die Entwicklung von Produktionssystemen noch nicht. Produktionseinführungszeit ist aus diesem Grund Engpass geworden.

Moderne Produktionssysteme sind massgeschneiderte Systeme mit sehr begrenzten Anpassungsmöglichkeiten. Daher ist Inbetriebnahme eines modernen Produktionssystems eine der schwierigsten Phasen eines Arbeitszenarios.

Die Produktionssysteme für das 21. Jahrhundert müssen in der Lage sein, die Kapazität und Funktionalität auf Marktumstände schnell anpassen zu können. Das heißt, dass das

Produktionssystem nicht nur beim Produzieren einer Vielfalt von Produkten sondern auch beim Ändern des Systems selbst flexibel ist.

Diese Herausforderungen sind mit den vorhandenen Produktionskonzepten nicht mehr zu meistern. Deswegen sind neue Produktionskonzepte und deren Umsetzung zum Überlebensrezept der Unternehmer geworden.

Neulich hat eine Studie des US nationalen Forschungsrates rekonfigurierbare Produktionssysteme als die höchste Priorität für zukünftige Forschung in der Produktionstechnik und eine der sechs Schlüsselproduktionsherausforderungen für das Jahr 2020 gekennzeichnet. Der Zweck der Rekonfiguration ist, ein Fertigungssystem von seiner gegenwärtigen Konfiguration zu einer anderen Konfiguration schnell und kosteneffektiv ändern zu lassen, ohne die Leistung des Fertigungssystems zu drosseln, wenn Produkt oder Produktion sich ändert, oder Ausfälle auftreten.

Wesentliche Fortschritte bei der Bewältigung künftiger Produktionsaufgaben lassen sich erzielen, wenn das Produktionssystem die einzelnen Arbeitsschritte, im Idealfall lückenlos vom Rohteil bis zum Fertigteil, autonom und störungstolerant ausführt. Die Aufgabe eines Zentralsteuerungssystem wird in diesem Fall auf das Definieren von Aufträgen, Ausführungszeiten und Auswahl der Fertigungsanlagen reduziert. Wie Aufträge ausgeführt werden, wird den autonomen Fertigungseinheiten überlassen. Die Idee hinter der *Autonomisierung* besteht darin, Komplexität der Steuerung einzuschränken und einen Rahmen für ein selbstorganisierendes Produktionssystem zu schaffen.

Voraussetzung dafür ist die Entwicklung der Fertigungseinheiten, die in der Lage sind, ohne menschlichen Eingriff zu laufen, in Echtzeit mit dynamischen Umgebungen und freundlich mit Menschen zu interagieren und dies während des intelligenten Ausführens der ihnen zugeteilten Aufgaben. Dies erfordert den Bau von Fertigungseinheiten, die ein adaptives, autonomes und zu gleich zielgerichtetes Verhalten aufweisen.

Für ein solches Vorhaben gab es jedoch weder Erfahrungen aus der laufenden Produktion noch aus anderen Industrieprojekten, die einen reibungslosen Betrieb unter industriellen Bedingungen sichern konnten.

Ziele der vorliegenden Arbeit sind:

- Entwurf von modularen, selbstorganisierenden Montagesystem.
- Ausarbeitung von Arbeitszenarien und Optimierung von Abläufen von selbstorganisierenden Montagesystem in CIM Umgebung.
- Harmonisierung konträrer Ansätze, Subordination und Selbstorganisation, im Rahmen von Arbeitszenarien von selbstorganisierendem Montagesystem.

- Wie Autonomie erreicht und erweitert werden kann sowie wie darauf aufbauend autonome Agenten für die Einsatz im selbstorganisierenden Montagesystem zu realisieren sind.

1.1 Kapitel Struktur

Erster Teil dieser Arbeit beschäftigt sich mit der Frage, wie Autonomie erreicht und erweitert werden kann sowie wie darauf aufbauend autonome Agenten für die Einsatz im selbstorganisierendem Montagesystem zu realisieren sind.

Im zweiten Kapitel werden zuerst Definitionen und Grundkonzepte der autonomen mobilen Systeme erläutert. Kapitel drei bearbeitet biologisch motivierende Leitpfade für den Entwurf autonomer Agenten. Kapitel vier analysiert bisherige Ansätze zur Steuerung von Agenten. Darüber hinaus wird analysiert welche Ansätze eignen sich für die Anwendung in der industriellen Umgebung. Daraus resultierende Architekturen werden sodann im Kapitel 5 näher untersucht.

Im Kapitel 6 geht es um Probleme der klassischen Produktionsteuerung und wie diese durch die Anwendung agentenbasierenden Steuerung gelöst bzw. umgangen werden können. Kern der Kapitel 7 bildet der Entwurf eines modularen selbstorganisierenden Montagesystems. Das Montagesystem besteht aus speziell aufgelegten autonomen Agentenklassen. Weiters werden Probleme der Integration von Selbstorganisation in die CIM-Struktur und das daraus resultierende Kommunikationskonzept vorgestellt und in detail analysiert. Abschließend werden für jede Agentenklasse eigene Abläufe definiert und in Algorithmen umgewandelt.

Um Abläufe innerhalb des Montagesystems zu optimieren, wurde im Kapitel 8 das reaktive scheduling basierend auf Vertragmethodologie entwickelt. Abschließend ist ein Vergleich heuristischer Methoden, die üblicherweise in der Produktionsoptimierung benutzt werden, mit reaktiven scheduling dargestellt. Im Kapitel neun werden Regeln für Rekonfiguration von Warteschlangen ausarbeitet.

Definitionen und Grundkonzepte

Ernsthafte Versuche, denkende handelnde Maschinen zu bauen, setzten nach dem Zweiten Weltkrieg ein. Eine Forschungsrichtung, Kybernetik genannt, verwendete elektronische Schaltkreise, die das Nervensystem imitierten, um Maschinen zu konstruieren, die einfache Muster zu erkennen lernten.

Ein anderer Ansatz, künstliche Intelligenz genannt, verwendete die arithmetische Leistung der Nachkriegscomputer zum abstrakten Schlussfolgern und stellte während der 60er Jahre Computer her, die logische und geometrische Theoreme bewiesen, Rechenprobleme lösten und gute Schachspiele ausführen konnten. Es schien dass der Weg zum Bau intelligenten Maschinen gefunden sei.

Ende der 60er Jahre verbunden Forscherteams am MIT und in Stanford ihre Computer mit Fernsehkameras und Roboterarmen, wodurch Programme beginnen konnten, Informationen direkt von der äußeren Welt aufzunehmen. Während die ausschließlich schlussfolgernden Programme ihre Arbeit befriedigend ausführten, brauchten die besten Programme zum Steuern von Robotern Stunden, um einen Baustein auf einem Tisch zu finden und zu ergreifen. Oft scheiterten sie daran vollständig und zeigten eine Leistung, die viel schlechter als die eines sechs Monate alten Kindes war. Mittlerweile ist es gelungen Computer zu bauen die mit den Großmeistern Schach spielen können, wenn deren Züge mit einer Tastatur eingegeben werden, aber kein künstliches System kann die Figuren auf einem Schachbrett so wie ein durchschnittlicher Mensch sehen und ergreifen.

Warum ist es leichter, menschliches Denken als Wahrnehmen und Handeln nachzuahmen. Die Antwort von Hans Moravec lautet: *„Über hunderte Millionen von Jahren überlebten unsere Vorfahren, indem sie besser sahen und sich bewegten als ihre Konkurrenz. Dadurch wurden sie phantastisch effizient. Wir können unsere gewaltigen Fähigkeiten kaum würdigen, weil sie uns einfach gegeben sind und von allen Menschen und den meisten Tieren beherrscht werden. Auf der anderen Seite ist rationales Denken, wie man es für Schachspiel braucht, eine neu erlangte Fähigkeit, die es vielleicht nicht länger als 100.000 Jahre gibt. Die dazu benutzten Hirnbereiche sind nicht gut organisiert, weswegen wir das*

nur sehr schlecht können. Wir haben bis vor kurzem nicht bemerkt, wie schlecht wir dies können, weil wir keine Konkurrenz hatten, die es uns vor Augen geführt hat“ (Moravec, 2000).

Warum sollte man bei der Entwicklung künstlicher Systeme von den seit jüngster Zeit vorhandenen rationalen Fähigkeiten bestimmter Lebewesen ausgehen? In der Evolutionspyramide betrachtet hieße dies von oben nach unten.

Rationales Denken ist eine höhere Ausdrucksform der Intelligenz aber nicht der Zweck der Entwicklung der Fähigkeit selbst. Intelligenz hat sich in Laufe der Zeit als Folge der Bedürfnisse in realer Umgebung zu recht zu kommen entwickelt.

Die Diskrepanz zwischen dem Ausgangspunkt der klassischen künstlichen Intelligenz sowie Richtung des Programms und den realen Bedürfnissen eines künstlichen Systems für den Einsatz in der realen Umgebung ist sehr groß. Die klassische künstliche Intelligenz ist geprägt durch die Annahme, dass sich die Intelligenz durch rein mathematisches – logisches Schlussfolgern auszeichnet. Dem zu Folge ist die Welt der klassischen künstlichen Intelligenz voraussagbar und statisch. Die Zeit, die Masse und die Form des Körpers, der „Intelligente“ Programme trägt, sind außer Acht gelassen. Das heißt, die klassische künstliche Intelligenz unterscheidet geometrische und physische Welten nicht. Das ist der Grund warum autonome künstliche Systeme, die der Basis der klassischen künstlichen Intelligenz aufgebaut sind, in der physischen Welt bescheidende Resultate hervorbringen. Mobiler Roboter Shakey ist das berühmteste Beispiel dafür. Bescheidende Resultate haben aber dazu beigetragen einen neuen Untersuchungsraum zu schaffen. Für diesem Zweck sind folgende Fragen zu stellen: Was benötigen Systeme, um in einer realen, teilweise unbekanntem, dynamischen und ungenauen Umgebung fähig zu sein, sich selbständig so Verhalten dass gegebene Aufgaben erledigt sein können?

Im diesem Kapitel sollen nun die Antworten auf diesem Fragen erarbeitet werden.

2.1 Agent

Autonome Agenten sind Systeme, die dynamische und unvorhersehbare Umgebungen bewohnen. Sie interpretieren selbst Sensordaten, die Ereignisse in der Umwelt widerspiegeln, und führen Aktoren, die auf die Umwelt wirken und sie verändern. Was diese Systeme von anderen unterscheidet, ist ihre Fähigkeit, in komplexen Umgebungen effektiv zu agieren ohne ständig von Menschen geführt und gesteuert zu werden.

Es gibt mehrere Arten von Umgebungen, in welchen Agenten eingesetzt werden können. Die physische reale Welt ist derart komplex, dass der Entwurf von Agenten, die in ihr erfolgreich arbeiten können, die größte Herausforderung darstellt. Es gibt auch virtuelle Welten verschiedener Arten. Das Internet, zum Beispiel, wird heutzutage immer komplexer. Hier

werden andere Formen von Agenten gebraucht, die nützliche Aufgaben in solcher Umgebung durchführen können.

AGENT: *Agent ist ein System mit der Fähigkeit innerhalb seiner Umgebung wahrzunehmen, zu berechnen und zu agieren. Multi Agent System besteht sich aus zwei oder mehreren Agenten (Mataric, 1994).*

Fortschritte auf den Gebieten der Computergraphik und der Simulationstechnologie haben ebenfalls dazu geführt, dass komplexe simulierte Welten und Agenten, die ein realistisches Verhalten aufweisen, konstruiert werden können.

Agenten können, abhängig von der Art der Umgebung, mehrere Formen annehmen (Abbildung 2.1). Agenten, die die physische Welt bewohnen, nehmen normalerweise die Verkörperung der mobilen Roboter an. Agenten, die mit der Software (z.B. Betriebssystem eines PCs, Internet, usw.) interagieren, werden oft Softbots oder Shopping Agenten genannt. Software-Agenten, die Menschen beim Treffen von komplexen Entscheidungen helfen oder bei der Erledigung von anderen Wissensverarbeitungsaufgaben unterstützen, werden Expert-Assistenten genannt. Diese Agenten werden oft mit anderer konventioneller Software (wie z.B. Datenbanksysteme) oder mit herkömmlichen Kontrollsystemen gekoppelt. Agenten in simulierten Umgebungen, wie Computeranimierte Umgebungen oder Videospiele, werden Virtuelle Agenten genannt.

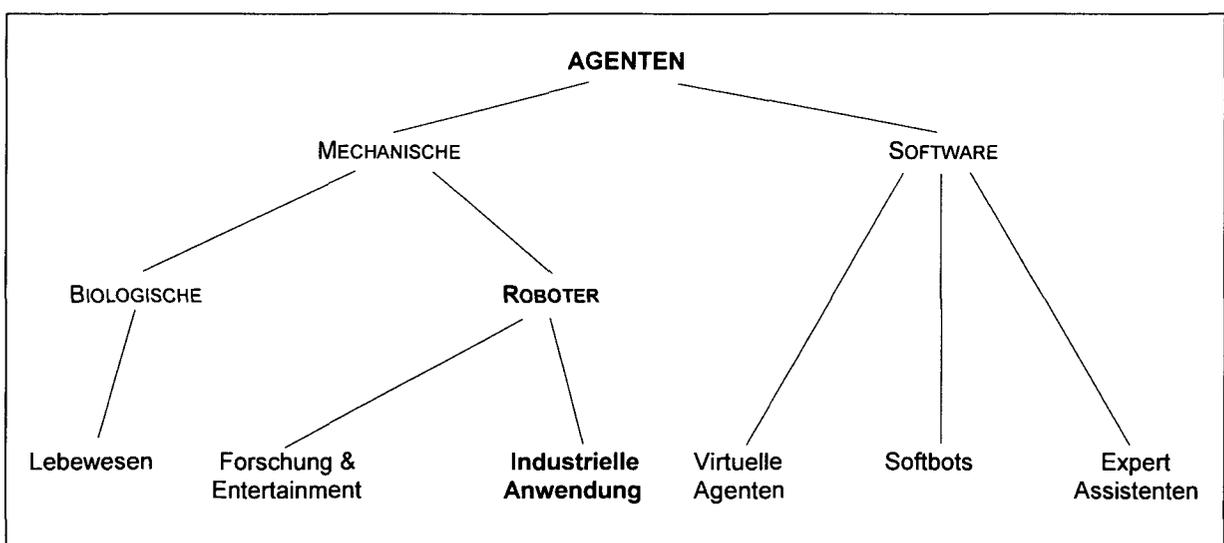


Abb. 2.1. Klassifikation von Agenten. Fett gedruckt bezeichnet Form und Art den Agenten, die in der Dissertation untersucht werden.

Agenten können auch verschiedene Rollen spielen. In der physischen Umgebung reicht das Spektrum von Spielzeugroboter, autonomen Haushaltgeräten sowie Rollstühle bis hin zu Agenten, die in der Automatisierungstechnik und Fertigung eingesetzt werden können und nicht zuallerletzt Marsrobotern und selbst steuernden Raumsonden. Softbots können als persönliche Assistenten, die auf dem eigenen Arbeitsplatzrechner des einzelnen Benutzers laufen, oder als Informationsvermittler im Netzwerk dienen. Expert-Assistenten können in der medizinischen Überwachung, in der Computerunterstütztenfertigung oder in der Verwaltung des Luftverkehrs eingesetzt werden. Virtuelle Agenten, die eher Charakter-Eigenschaften wie Glaubwürdigkeit und Persönlichkeit als tiefe Intelligenz hervorheben, können in interaktiven Systemen für Unterhaltung, Kunst, Bildung oder Verbraucher-Software eine bedeutende Rolle spielen. Der Bedarf an solchen autonomen Systemen wurde in den letzten Jahren erkannt, und es wurden viele Anstrengungen unternommen, solche Systeme für eine Vielzahl von Aufgaben zu konstruieren.

2.2 Autonomie

Die zentrale Idee in dem Konzept von Autonomie ist durch die Etymologie des Wortes *autos* (selbst) und *nomos* (Regel oder Gesetz) gegeben. Griechische Städte, deren Bürger ihre eigenen Gesetze gemacht haben und nicht nach den Gesetzen einer externen Regierungsmacht leben wollten, wurden als autonom bezeichnet.

Es ist nützlich, Autonomie mit dem Konzept der Automatik zu vergleichen. Die Bedeutung des Wortes „Automatik“ kommt von der Etymologie des Wortes *Kybernetik*, das von dem griechischen Wort für Selbst-Regulierung abgeleitet ist. Mit anderen Worten, automatische Systeme sind selbstregelnd, machen aber nicht die Gesetze, die von ihren Regelungsaktivitäten erfüllt werden müssen. Diese Gesetze werden ihnen zur Verfügung gestellt oder in sie eingebaut. Automatische Systeme steuern sich selbst entlang eines gegebenen Pfades. Während ihrer Arbeit korrigieren sie und gleichen die Auswirkungen von externen Störungen aus.

Autonome Systeme andererseits sind Systeme, die für sich selbst die Gesetze und Strategien entwickeln, nach denen sie ihre Verhaltensweisen regulieren: Sie sind in der Lage, sowohl die Pfade zu bestimmen, denen sie folgen, als auch sich entlang dieser Pfade zu bewegen. Damit ein System autonom sein kann, muss es zuerst automatisch sein. Dies bedeutet, dass das System in der Lage sein muss, in seiner Umgebung zu agieren, diese Umgebung wahrzunehmen und sie so zu beeinflussen dass Aufgaben erledigt werden können. Autonomie aber geht über die Automatik hinaus, weil dabei zusätzlich angenommen wird, dass die Basis für Selbst-Steuerung aus der eigenen Fähigkeit des Systems entsteht, seine Verhaltensprinzipien zu bilden und anzupassen.

AUTONOMIE: *Unter Autonomie versteht man "Handlungsfreiheit" in dem Sinne, dass ein System sich nicht nur automatisch, das heißt selbst - regulierend, verhält, sondern auch selbst -steuernd (Christaller, 2000).*

Ferner, ist es so, dass die Prozesse des Aufbaus und der Anpassung der eigenen Kompetenz eine Sache ist, die während des Agierens des Systems in seiner Umgebung stattfindet. Das System hat nicht die Zeit, eine große Menge von Beispielen zu studieren oder komplexe Bearbeitungsprozesse anzustellen, wie es unvorhersehbare Situationen bewältigen kann. Stattdessen muss es kontinuierlich agieren und reagieren, um zu überleben.

Eine andere Art, Autonomie zu charakterisieren, wäre es, die Sicht des Betrachters zu berücksichtigen. Der Verhaltensforscher David McFarland (McFarland, 1993) hebt hervor, dass ein automatisches System ein System ist, für das man das Verhalten vollständig vorhersagen kann, sobald man seine interne Basis für die Entscheidungsfindung kennt. Ein autonomes System dagegen ist ein System, das sich selbst entscheiden kann. Selbst für den ursprünglichen Designer eines Systems ist es nicht klar, wie es in der Zukunft reagieren wird, weil es so aufgebaut war, dass Reaktionen sich entwickeln und ändern, um neue Situationen zu bewältigen.

Ein autonomer Agent ist somit ein selbstverwaltendes, selbstregulierendes System, das sich selbstständig in seiner Umgebung zurechtfinden kann. Dies bedeutet, dass ein autonomes System, adaptiv und langfristig selbst – genügend sein muss. Beide Merkmale würden intuitiv als wesentliche Teile der Intelligenz bezeichnet werden. Offenbar wird die Autonomie einer Agent, einer Maschine oder eines Lebewesens durch dessen Intelligenz bestimmt.

Absolute Autonomie existiert nicht. Jeder Agent hängt von den äußeren Faktoren; anderen Agenten und der Umgebung ab.

2.3 Intelligenz

Der Versuch, Intelligenz zu definieren, wird umso schwieriger, je genauer man versucht, den Begriff einzukreisen. Eine relativ strenge Definition des Begriffes ist möglicherweise zurzeit sogar schädlich, weil man noch gar nicht weiß, aus welchen Bestandteilen sich das Phänomen Intelligenz zusammensetzt. Eher soll eine Art Sammlung von Eigenschaften genannt werden, die wesentliche Elemente intelligenter Systeme darstellen.

Die klassische künstliche Intelligenz hat den Begriff Intelligenz meist als etwas vorgestellt, das mit logischer Vernunft und Symbolbearbeitung zu tun hat. Man glaubte das Wesen von Intelligenz im planvollen, strukturierten, abstrakten, logischen Denken erkannt zu haben. Dies ist um so mehr, als man, entsprechend der wörtlichen Bedeutung von *intelligentia*, das

mit Verstand, Vorstellung übersetzt wird, unter Intelligenz die Fähigkeit zu problemlösendem, einsichtigem Verhalten verstehen kann. Diese Einsicht setzt die Fähigkeit voraus das jeweilige Problem vor einem inneren Auge sehen zu können. Konsequenterweise beschäftigte sich die klassische Künstliche Intelligenz mit Programmen zum Lösen abstrakter, gut formalisierter Probleme. Typische Beispiele solcher Probleme sind rein mathematische-logische Berechnung wie das Schachspielen, das Führen eines mathematischen Beweises oder das logische Folgern aus einer Menge von Axiomen. Diese Vorstellung der Intelligenz ist durch den Turing-Test, entwickelt von Allan Turing erfasst. Turing Test hatte entscheidenden Einfluss auf die Formulierung der Basisannahmen und die Entwicklung der künstlichen Intelligenz Forschung. Der Test ist wie folgt; Eine Versuchsperson kommuniziert über eine Tastatur mit zwei Partnern. Der eine ist ein Rechner und der andere ist ein Mensch. Wenn nach einer vorher bestimmten Zeit die Versuchsperson nicht entscheiden kann, welcher der beiden der Rechner ist, dann ist die Maschine intelligent. *„Dieser Test kann auch heute noch dazu dienen, das Gebiet der klassischen künstlichen Intelligenz zu charakterisieren: Es geht darum, Rechnersysteme zu konstruieren, die den Turing-Test bestehen“ (Christaller, 1998).*

In den letzten Jahren wurde die klassische Auffassung von Intelligenz oft bestritten. In Vergleich zum einzig sicheren intelligenten System, nämlich das der Menschen trat ein Paradoxon auf; Je mehr Informationen vorhanden sind, desto langsamer findet das Künstliche System eine passende Lösung. Genau umgekehrt ist es bei Menschen und bei anderen Systemen, die man zu mindest intuitiv als intelligent bezeichnen könnten. Je mehr Informationen vorhanden sind desto mehrere Kombinationsmöglichkeiten gibt es.

Das künstliche System tut etwas, weißt aber nicht, dass er es tut und wie er es tut. Weil wir wissen, dass wir etwas tun, können wir innehalten und einfach aufhören. Und weil wir wissen, wie wir es tun, können wir beschließen, uns in Zukunft anders zu verhalten als bisher. Selbst mit Hochleistungscomputern ist es bisher keinem künstlichen System gelungen, Schuhbänder zu einer Schleife zu knüpfen, was ein Kind lernt, lange bevor es in der Lage ist, über mathematische Beweise nachzudenken.

Mehrere Forscher meinen (McFarland 1993, Brooks 1989, 1991, Steels 1994) dass die künstliche Intelligenz Forschung nur auf festen Grundlagen stehen kann, wenn sie mit der Biologie stark integriert wird. Einer der Ansprüche dieses neuen Forschungsstils ist, dass man ein großes Stück in Richtung des Verstehens der biologischen Grundlagen intelligenter Aktivität zurückgelegt wird, wenn man den Basismechanismus studieren und verstehen kann, mit dem wir und andere Tiere die Grundbedürfnisse Ernährung oder Fortpflanzung befriedigen. Es stellt sich heraus, dass viele der Annahmen der klassischen KI überprüft werden müssen, wenn ein strenger biologischer Ansatz angewendet wird. Aus der Sicht der Biologie ergibt sich die folgende Betrachtungsweise; Die Evolution belohnt nicht die

Entwicklung allgemeiner oder theoretischer Fähigkeiten, sondern sie belohnt die Entwicklung der Fähigkeit, in einer konkreten Umgebung zurechtzukommen.

INTELLIGENZ: *Ein System ist Intelligent, wenn es in einer gegebenen und einer sich ändernden Umwelt die Chancen seiner Selbsterhaltung im Vergleich zu seinem aktuellen Zustand verbessern kann (Steels, 1994).*

Der Intelligenz ist die Fähigkeit des Systems, eigenes Verhalten in einer gegebenen und einer sich ändernden Umwelt ständig anzupassen, sodass die Aufgaben ausgeführt werden können (Fogel, 2000).

2.4 Adaptivität

Offensichtlich ist Adaptivität wesentlicher Teil intelligenter Systeme. Grundlegende Eigenschaft besteht darin, dass das System sich an wechselnde Umweltbedingungen anpassen kann. Anpassung beim Verhalten bedeutet natürlich nicht, dass sich das System wie eine wächserne Masse einer vorgegebenen Form anpasst. Anpassen bedeutet eher die Fähigkeit, nur auf gewisse, für das System wichtige Eigenschaften der Umwelt, die ohne Konsequenzen für das System sind oder sich sogar störend auf wichtige Funktionen auswirken könnten, absehen zu können. Solche Anpassungen findet man auf verschiedenen Ebenen, von einfachen senso-aktorischen Mechanismen bis hin zu komplexen Verhaltensweisen.

Die Adaptivität besteht aus zwei Komponenten; zustimmen mit den existierenden Gesetzen und die Erzeugung neuen Verhaltens. Nur dann, wenn beide Komponenten vorhanden sind, spricht man von Adaptivität.

ADAPTIVITÄT: *Adaptivität ist die Fähigkeit des Systems, mit den Veränderungen in der Umwelt fertig zu werden (Mataric, 1994).*

2.5 Verbindung zwischen den Begriffen

Schon die ersten Einzeller, hatten Bedürfnisse (in Agenten Wörterverzeichnis Aufgaben) und entwickelten relativ einfache Mittel, um sie zu befriedigen. Um Nahrung zu sammeln benützten sie einen Greifarm am vorderen Ende des Körpers. Ihre Suche nach Nahrung war eine zufällige Irrfahrt. Stießen sie gegen die richtigen Dinge, so griffen sie danach. Die hatten keinen Plan, keine Karte der Umgebung, keine Repräsentation der gesuchten Gegenstände,

die über die Form des Greifarms hinausgegangen wäre. Es war das Prinzip von Schlüssel und Schloss, mehr nicht (Dennett, 1996). Nach der Auffassung der klassischen künstlichen Intelligenz über Intelligenz, sind Einzeller keineswegs intelligent. Trotzdem haben sie mit realen Bedürfnissen der Agenten viele Gemeinsamkeiten. Existierende Agenten können das Verhalten des Einzellers nachahmen. Nächste Stufe: Insekten.

Sie können sich mit einer Geschwindigkeit von einem Meter und mehr pro Sekunde bewegen, dabei Hindernisse und Räubern ausweichen sowie Futter und Partner für die Paarung finden. Insekten sind in der Lage je nach dem aktuellen Zustand ihrer Umwelt passende Aktionen auszuwählen und unter Einhaltung der zeitlichen Bedingungen sie auch auszuführen.

Soziale Insekten sind in der Lage ohne eine Form der zentralen Steuerung allein durch lokale Interaktionen erstaunliche globale Verhaltensmuster und Strukturen zu erzeugen.

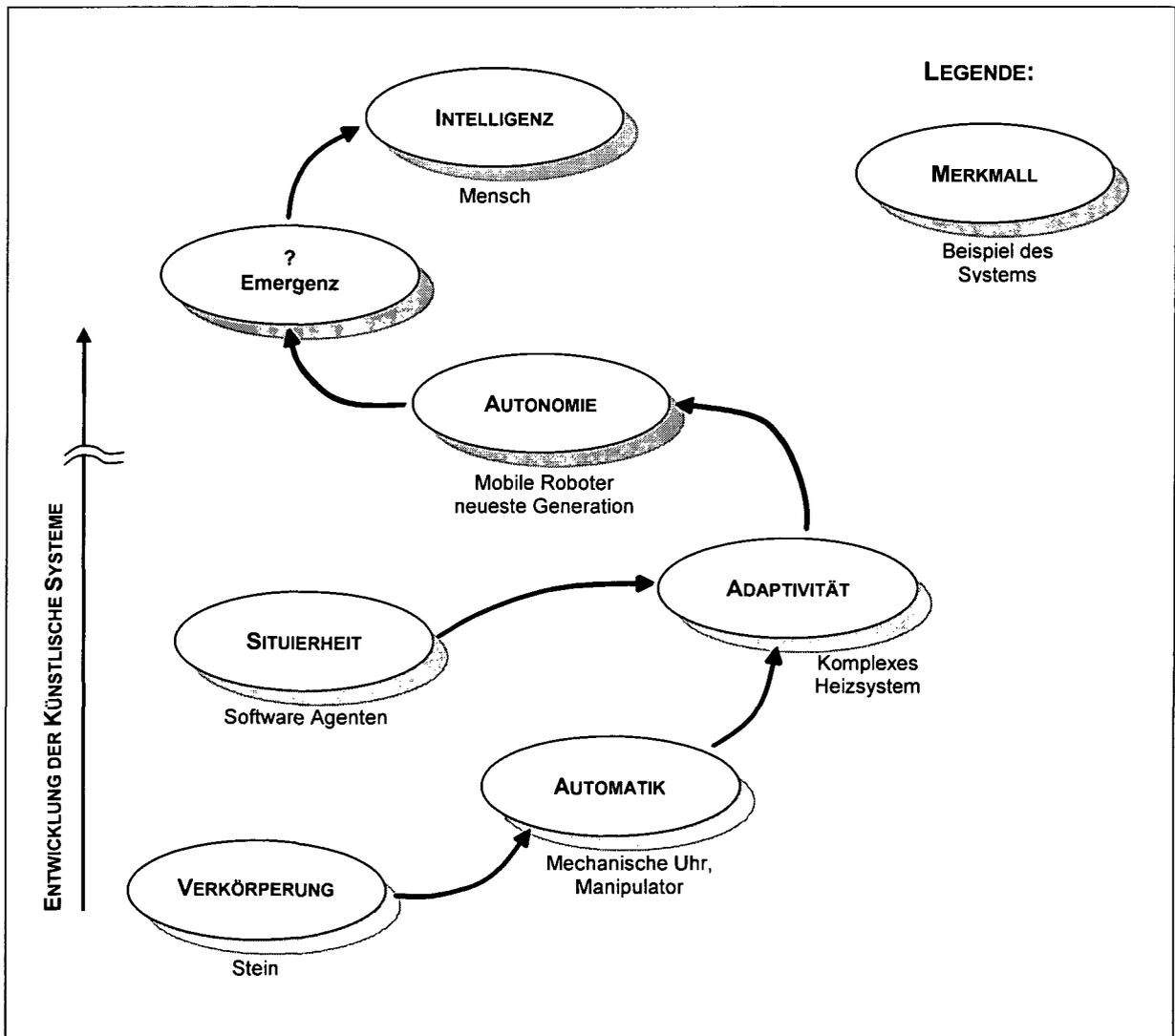


Abb. 2.2. Komponenten, die einen Agent in der realen Umgebung besitzen soll.

Insekten können wesentlich mehr als jeder existierende autonome mobile Roboter. Was ist das Besondere an der Organisation und am Bauplan von Einzellern und Insekten, das es ihnen erlaubt, so erhebliche Leistungen mit geringer Rechnerleistung zu erbringen? Ein Einzeller, Insekt, Mensch oder Roboter sowie alle andere Dinge auf diesem Planeten sind physische Objekte und unterliegen damit physikalischen Gesetzen. Ein lebender Organismus ist eine organisatorisch geschlossene Einheit. Die Akzeptanz einer derartigen Aussage hat erhebliche Konsequenzen für die Beschreibung autonomer Systeme, denn sie verlangt die organisatorische Geschlossenheit autonomer Systeme. Sie verlangt „komplette“ Agenten (Pfeifer & Scheier, 1999). Außerdem kann die Intelligenz in der Natur nie ohne einen Körper existieren. Folge dessen sind zwei fundamentale Prinzipien; *Verkörperung* und *Situiertheit*.

Verkörpert ist ein Wesen oder Agent, das oder der einen Körper hat und die Welt zumindest teilweise direkt durch deren Einflüsse auf seinen Körper erfährt. Nur wenn ein System direkt mit seiner Umwelt in Verbindung steht, also über einen Körper verfügt, kann es sich wirklich situativ Verhalten. Dies ist eine radikale Abkehr von der klassischen künstlichen Intelligenz mit ihren geometrischen, statischen, zeitlosen Welten. Systeme wie zum Beispiel Schach oder theorembeweisende Programme, sowie Mobileroboter wie Shakey haben keine kontinuierliche Existenz, die in dem Fluss der Zeit eingebettet ist. Sie sind nicht situiert.

Situiert ist ein Wesen oder Agent, das oder der in die Welt eingebettet ist und nicht mit abstrakten Beschreibungen arbeitet, sondern durch seine Sensoren mit dem Hier und Jetzt der Umgebung, auf das Verhalten dieses Wesens wirkt. Ein Manipulator, das gleiche Muster wiederholt, ist verkörpert aber nicht situiert.

Im Gegenteil zu unbelebten ein lebendes System regelt seine Regelung selbst. Insekten sowie andere Lebewesen sind Systeme, die in ungewissen Umgebungen operieren können, wobei ihre Fähigkeiten zur Selbstkontrolle aus der Möglichkeit resultieren, eigene Verhaltensweisen zu verändern und anzupassen. Sie sind Autonome Systeme.

2.6 Intentionaler Standpunkt

Zur Beschreibung eines System, biologisches oder künstliches, kann man zwei elementare Standpunkten oder Voraussagenstrategien benutzen: physikalischen Standpunkt und Gestaltungsstandpunkt.

Der *physikalische Standpunkt* ist die Methode, bei der man alle seine Kenntnisse über die Gesetze der Physik und den physischen Zustand des in Frage stehenden Systems einsetzt, um schließlich eine Voraussage machen zu können. Wenn man voraussagt, dass ein Stein, der aus der Hand losgelassen wird, zum Boden fallen wird, bedient man sich des physikalischen Standpunktes. Jedes physisches System, unterliegt den Gesetzen der Physik und verhält sich deshalb auf eine Art, die man mit Hilfe des physikalischen Standpunktes erklären und

voraussagen kann. Aber wenn es um die Beschreibung und die Voraussagen der Funktionsfähigkeit von komplexeren Systemen geht, zum Beispiel Taschenrechner, ist es mühsam mittels physikalischen Standpunktes Voraussagen zu machen. Einfacher und viel schneller ist es zu unterstellen, dass er eine bestimmte Konstruktion besitzt und dass er dieser Konstruktion entsprechend funktionieren wird. Aufgrund dieser Voraussage ist man bereit große Risiken einzugehen, nämlich die falsche Berechnung zu machen. Das ist *Gestaltungsstandpunkt*.

Voraussagen aufgrund des Gestaltungsstandpunktes sind riskanter als solche nach dem physikalischen Standpunkt, weil man zusätzliche Annahmen einbeziehen muss, nämlich, dass das untersuchte System tatsächlich so konstruiert ist, wie man es ihm unterstellt und dass keine Fehlerfunktion eintritt. Bei gut gestalteten Systemen funktioniert die Voraussage aufgrund des Gestaltungsstandpunktes ausgezeichnet. Der Gestaltungsstandpunkt ist eine hervorragende Abkürzung des physikalischen Standpunktes, den man im täglichen Leben ständig verwendet. Eine weitere Abkürzung des Gestaltungsstandpunktes ist der intentionale Standpunkt.

Alle Systeme, die von Signalen gelenkt werden und nach Zielen streben kann man als *intentionale Systeme* bezeichnen. Darunter alle Lebewesen sowie Agenten.

Die grundlegende Strategie des *intentionalen Standpunktes* besteht darin, dass man das fragliche System als Akteur behandelt, um seine Handlungen vorauszusagen und sie damit in einem gewissen Sinn zu erklären. Beispiel dafür ist die anfangs von McFarlads beschriebene Betrachtungsweise der Autonomie.

Annahmen des intentionalen Standpunktes sind: untersuchtes System ist ein vernünftig handelnder Akteur, für welches man Zweck und Ziele des Verhaltens erkannt hatte. Dabei ist die Konstruktion des beobachteten Systems außer acht gelassen. Man ist, wie etwa in Psychologie und der Verhaltensbiologie, auf das beobachtete Verhalten des Lebewesens begrenzt. Ausgangsfrage des intentionalen Standpunktes lautet: Was würde ich tun, wenn ich in der gleichen Lage wie das betrachtete System wäre? Gefahr dabei besteht darin, dass man alle intentionale Systeme so behandelt, als wären sie wie wir, was natürlich nicht stimmt. Außerdem ist die Bedeutung der Information für das betrachtete System nicht notwendigerweise identisch mit der Bedeutung der Information die "im Kopf des Betrachters errechnet" wird. Das kann zu Überinterpretationen führen.

Intentionaler Standpunkt ist notwendig wenn es um Analyse von komplexen Systemen geht. Unbedacht angewandt, kann der intentionale Standpunkt den Beobachter gewaltig in die Irre führen, aber richtig verstanden, ist er auf dem Gebiet der Agententechnologie eine vernünftige Sichtweise, welche die Aktionen der Agenten deutlich macht und unsere Aufmerksamkeit auf die entscheidenden Experimente lenkt. So ein derartig bahnbrechendes Experiment ist von Neurowissenschaftler Valentino Breitenberg durchgeführt. In seinem

Buch „*Vehicles: Experiments in Synthetic Psychology*“ beschreibt Breitenberg eine Ansammlung von Gedankenexperimenten in dem steigend komplexere Vehicles aus einfachen mechanischen und elektronischen Komponenten aufgebaut sind. Vehicle bei Breitenberg ist gleichbedeutend mit dem Begriff Agent in dieser Arbeit. Insgesamt gibt es 14 Agenten. Jeder dieser Agenten, vom intentionalen Standpunkt aus betrachtet, ahmt intelligentes Verhalten nach. Je nach dem welches Verhalten der Agent aufweist, hat er eigenen Namen. So gibt es „Getting Around“, „Fear“, „Aggression“, „Love“, „Values“, „Logic“ usw. Breitenbergs Experimente zeigten, dass man aufgrund intentionalen Standpunktes bereit ist, den Agenten viel mehr Fähigkeiten zuzuschreiben als es tatsächlich der Fall ist. Deswegen tritt er für eine wechselhafte Kombination von Analyse und Synthese ein. Er nennt diesen Vorgang „*law of uphill analysis and downhill invention*“. Das heißt, es ist schwieriger allein aufgrund von Verhaltensbeobachtung des Agenten Vermutungen über seine innere Struktur anzustellen als einen Agenten zu konstruieren, der das Verhalten nachahmt.

Warum ist es so? Ein Agent ist ein System, der aus mehreren Komponenten besteht. Diese Komponenten können aber ohne expliziten äußeren Zugriff miteinander interagieren. Damit entstehen neue Strukturen und Eigenschaften, die aus der Systemebene sichtbar aber aus der Sicht der einzelnen Komponenten nicht nachvollziehbar sind. Dieses Phänomen nennt man Emergenz. Kinnebrock definiert Emergenz folgendermaßen: „*Die Eigenschaft eines Systems ist emergent, wenn man bei der Beschreibung aller Eigenschaften der Einzelteile des Systems auf diese Eigenschaft verzichten kann*“ (Kinnebrock, 1996). Mit Emergenz ist also gemeint, von Aristoteles schon gekennzeichnete Tatsache, dass das Ganze mehr als die Summe seiner Teile ist.

Meinung, die hier vertreten wird, ist dass der Begriff Emergenz nur als ein Übergangsbegriff dient, die unsere mangelhaften Kenntnisse über Zusammenwirken der Komponenten eines Systems deckt. Tatsächliches Verständnis des Zusammenwirkens der Komponenten des Systems wird den Begriff Emergenz überflüssig machen.

2.6.1 Handlung, Verhalten, Aktion

Direkte Folge des intentionalen Standpunktes sind die Begriffe Handlung, Verhalten und Aktion, die zur Beschreibung der Tätigkeiten des Agenten benutzt werden. Der Agent selbst verfügt weder über ein Bewusstsein noch Sinn oder Zielsetzung. Handlungen, Verhalten und Aktionen existieren nur im Kopf des Betrachters. Die Begriffe sind aber weder fest definiert noch voneinander eindeutig abgegrenzt. Deren Verwendung in der Psychologie und in der künstliche Intelligenz Forschung weicht voneinander ab. In der Psychologie wird ein Verhalten als Handlung bezeichnet, wenn es aus Sicht des Handelnden zielgerichtet ist, also einen Sinn hat (Heckhausen, 1989). Also ein Beobachter wird das Verhalten als Handlung

bezeichnen, wenn er diesen Sinn erkennen und verstehen kann. Als Verhalten werden alle beobachtbaren Reaktionen eines Agents auf einen Reiz bezeichnet. Im Unterschied zu Handlungen kann man Verhalten zwar Ursachen, aber keinen Sinnzusammenhang zuweisen (Heckhausen, 1989). So können demselben Verhalten unterschiedliche Handlungen zugrunde liegen. Zum Beispiel kann das Verhalten, in Richtung Tür zu gehen Teil der Handlung sein, jemanden zu treffen. Es kann aber auch mit der Intention geschehen, die Tür zu blockieren, damit niemand durchkommt.

In der früheren künstlichen Intelligenz wurden alle Interaktionen eines Agents mit der Umgebung als Aktion bezeichnet. Für zielgerichtetes Handeln werden Aktionen in Handlungsplänen zusammengefasst. Mitte der achtziger Jahre wurde von Brooks der Begriff des Verhaltens in der Robotik Forschung eingeführt. Das war die Geburt der „behavior-based“ Robotik. Den Begriff des Verhaltens benützt man um fest kodierte Algorithmen zu bezeichnen, die die Interaktion des Agenten mit der Umgebung definieren. *„A behavior is a control law that clusters a set of constraints in order to achieve and maintain a goal“* (Mataric, 1990). Damit ist das Verhalten auf Algorithmus reduziert. Diese Reduktion ist direkte Folge eines unbedachten Gebrauchs des intentionalen Standpunktes. Bedauerlicherweise ist diese Verwendung des Begriffes in der behavioristischen Robotik ausgebreitet.

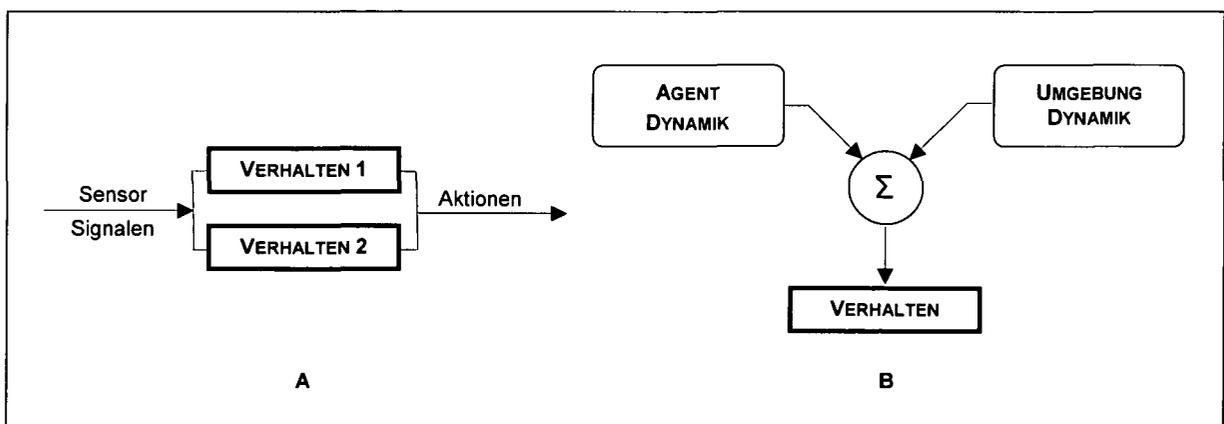


Abb. 2.3. (A) Bedeutung des Begriffes Verhalten im behavioristischen Robotik und (B) in der Verhaltensbiologie.

Im Gegenteil zur üblichen Verwendung des Begriffes in der behavioristischen Robotik wird hier das Verhalten als Eigenschaft der Agent-Umgebung Interaktion bezeichnet. Die Umgebung ist als Grundlage für die Entstehung sowie Erhaltung des Verhaltens zu sehen. Das Verhalten ist immer mit der Umgebung verbunden und kann nicht abstrakt betrachtet werden. Zwei gleich aufgebaute Agenten weisen in unterschiedlichen Umgebungen

unterschiedliches Verhalten auf. Das Verhalten ist die Eigenschaft der Agent-Umgebung Interaktion und kann nicht weder zum Agent noch zur Umgebung allein zuschreiben werden. Noch weniger vernünftig ist das Verhalten nur mit einem inneren Prozess des Agenten zu begrenzen.

AKTION, VERHALTEN, HANDLUNG: *Aktion ist die von einem Agenten verursachte Überführung eines Zustands in einen Folgezustand. Verhalten ist zeitlich ausgedehnte, situationsgesteuerte Aktionssequenz. Handlung ist zielgerichtetes, aber nicht notwendigerweise bewusstes Verhalten.*

Aktionen stehen also für einzelne diskrete Interaktionen mit der Umgebung, die die kleinste Einheit möglicher Interaktion mit der Umgebung bilden und sind von kurzer zeitlicher Dauer. Verhalten werden durch die Ausführung einzelner Aktionen realisiert.

Was in behavior-based Robotik das Verhalten bedeutet ist hier als der auslösende Mechanismus bezeichnet. Damit ist das Wesen des Prozesses erfasst. Auslösender Mechanismus besteht aus einem Algorithmus, der aufgrund von Sensorensignalen und deren Höhe, eine vordefinierte Reaktion auslöst. Was für ein Verhalten diese Reaktion auslöst hängt von Agent-Umgebung Gegebenheiten ab.

Verhalten in der realen Umgebung

„Wir selbst sind ein Produkt der Evolution. Wir erleben diesen Evolutionsprozess unmittelbar in unserem eigenen Leben. Die Welt um uns herum wandelt sich so rapide wie nie zuvor. Vor den Veränderungen, die wir selbst miterlebt haben, vollzog sich die Evolution der modernen Zivilisation, die selbst wiederum in früheren Zivilisationen und primitiveren Gesellschaftsformen wurzelt. Dahinter liegt die lange Hominidenentwicklung, und dann stößt man auf Menschenaffen. Und weiter zurück geht die Linie über primitive Säugetiere, Reptilien, Fische, die ersten Wirbeltiere bis hin zu Einzellern, Mikroben und zu den ersten lebendigen Zellen. Noch weiter zurück gelangt man in das Reich der Moleküle und Kristalle und zuletzt zu den Atomen und Elementarteilchen. Das ist unser Stammbaum. Wenn wir die Evolution als den Mechanismus akzeptieren, der uns hervorbracht hat, verstehen wir, dass wir nicht mehr sind als eine hochgradig organisierte Ansammlung von Biomolekülen“ (Brooks, 2002).

Auch moderne Biologie liefert Beweise dass Lebewesen komplexe physikalisch-chemische Systeme sind (Churchland, 2001). Vom Biologen John Maynard Smith und dem Evolutionsbiologen George Price wurde 1973 „Theorie der Evolutionsstabile Strategie“ aufgestellt. Evolutionsstabile Strategie bezeichnet Verhaltensmuster, das sich unter dem Einfluss der natürlichen Selektion gegen jedes andere Muster durchsetzt hat. In der betreffenden Population ist es allgemein verbreitet. Man kann annehmen, dass sich das Verhalten von Lebewesen auf programmierte Verhaltensregeln, die Lebewesen in bestimmten Situationen befolgen, zurückführen lässt. Denn die natürliche Selektion führt zu solchen Strategien und auch dazu, dass sie beibehalten werden.

Eines der wichtigsten Experimente, die die Gehirnforscher heute zu derartigen Ansagen ermutigen, fand bereits Mitte der achtziger Jahre an der Universität Berkeley statt. Wissenschaftler griffen dabei, damals noch eher zufällig, in das Programm der Nervenzellen ein. Die Neurochirurgen reizten das Gehirn einer jungen Frau bei geöffneter Schädeldecke mit elektrischen Impulsen. Weil das Gehirn selbst keine Schmerzen empfindet, konnte die

Frau unnarkotisiert über ihre Empfindungen sprechen, während an ihren Gehirnzellen manipuliert wurde. Als die Hirnchirurgen ein bestimmtes Areal im Vorderhirn mit Strom traktieren, begann die Frau plötzlich laut zu lachen. Nachdem hat sie auch gute Gründe dafür genannt. Die neuesten derartigen neuropsychologischen Experimente auf verschiedenen Tierarten liefern immer dichtere Indizien dafür, dass das Gehirn vordefinierte Algorithmen und Reaktionsschemen hat. Ein Nervensystem ist ein Kontrollsystem, das für den grundlegenden Zusammenhalt des Lebewesens sorgt (Dennett, 1996). Neurowissenschaftler Damasio vertrat dieselbe These; „*Bald werden wir wissen wie wir Glück und Trauer, Lust und Schmerz erfahren und selbst die Mechanismen des Menschlichen Bewusstseins werden uns keine Rätsel mehr aufgeben*“ (Damasio, 1999).

Die aufgestellten Thesen vermitteln einen Eindruck, dass biologische Systeme nachgeahmt werden können. Obwohl dies mit heutiger Technologie nicht möglich ist, treibt diese Idee Forschung in eine neue Richtung. Der Schwerpunkt der Forschung liegt auf der Konstruktion solcher Systeme, die fähig sind intelligentes Verhalten in der realen Umgebung auszuweisen. Die Untersuchung, wie biologische Systeme und deren Mechanismen in der Umgebungsaufgabe Kontext sich entwickelt haben, ist die zentrale Fragestellung. Genau dies ist für die Entwicklung funktionsfähiger autonomer Agenten entscheidend.

3.1 Der Aufbau der Regelkreise

Eine von den zentralen Aufgaben jedes lebendigen Wesens ist es bestimmte Stoffe und Temperaturen im Körper innerhalb eines Sollwertbereiches zu halten oder rechtzeitig in diesen zurückzuführen, wenn sie ihn verlassen haben.

Die Lebensprozesse mancher Organismen sind an eine bestimmte Körpertemperatur gebunden. Innerhalb dieser entwickelte sich der passende Mechanismus, der die Körpertemperatur in der für den Organismus akzeptablen Grenzen halt. Die Lebensprozesse verbrauchen Energie, die daher ständig neu bereitgestellt werden muss. Auch da gelten gewisse Grenzwerte, die erhalten werden mussten. Zum Beispiel der Anteil an verschiedenen Mineralien und Flüssigkeitsmenge innerhalb einzelner Organe oder Sauerstoffgehalt im Blut usw. Die beschriebene Einhaltung bzw. Wiederherstellung gewünschter Systemzustände kann mittels einfacher Regler in künstlichen Systemen nachgeahmt werden. Abbildung 3.1 zeigt einen dafür geeigneten Regelkreis. Dies muss jedoch nicht immer der Fall sein. Bei einer größeren Störung des Steuerwerts ist der Regelkreis nicht ohne weiteres in der Lage, den Steuerwert genügend rasch nachzuregeln. Wenn $\Delta \geq \Delta_{\max}$ der Regelkreis ist nicht mehr stabil noch kann er sich selbst in den Stabilbereich zurückstellen. Um sich erneuert synchronisieren zu können muss der Regler zurückgestellt sein.

Der beschriebene Regelkreis kann nur in einer Weise auf eine Sollwertabweichung reagieren.

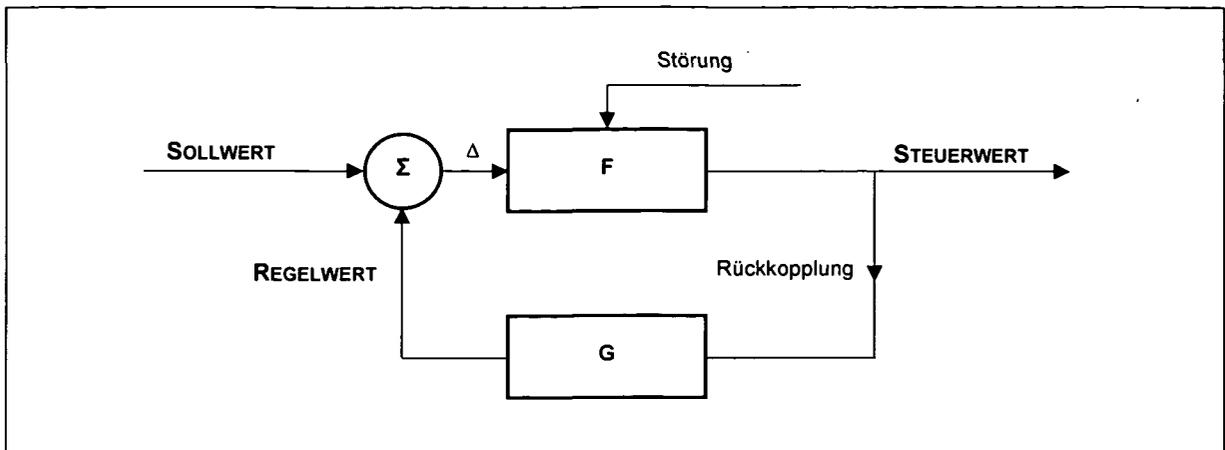


Abb. 3.1. Blockdiagramm des Regelkreises.

Es gibt aber auch Bedarf nach Systemen, die auf verschiedene Sollwerte reagieren sollen oder bei denen der höhere Stabilitätsgrad erforderlich ist. Wenn etwa die Körpertemperatur allzu sehr ansteigt, wird zunächst durch Erweiterung der Hautgefäße dafür gesorgt, dass mehr Blut in die Körperperipherie strömt, auf der Abbildung 3.2 als Mechanismus 1 bezeichnet. Dies führt zu einer stärkeren Wärmeabstrahlung und dadurch zu einer Temperatursenkung, ein Effekt, der hinreichen und die Körperwärme wieder in den Bereich des Sollwerts bringen kann. Bleibt diese Maßnahme jedoch erfolglos, schaltet sich als nächste Stufe die Schweißabsonderung ein (Mechanismus 2). Reicht das aber nicht, so könnte als dritte Stufe die Frequenz des Herzschlags erhöht werden (Mechanismus 3), was dazu führt, dass noch mehr Blut in die Körperperipherie strömt und somit die Wärmeabstrahlung noch einmal erhöht wird.

Um die Stabilität des Reglers zu steigern, hat die Evolution Redundanz angewendet. Redundanz findet man auf den verschiedenen Ebenen. Zum Beispiel die Visual- und Hörsysteme sind teilweise überlappend. Man konnte nur mittels eines Hörsystems die relative Orientierung und Entfernung aufgrund von Geräuschen abschätzen. Das Prinzip der Redundanz wird in biologischen Systemen häufig benützt. Auf diesem wichtigen Aspekt des biologischen Designs sollte bei der Entwicklung künstlicher Systeme Rücksicht genommen werden. Ein funktionsvergleichbarer Regelkreis, der um ein Gleichgewicht zu sichern verschiedene Steuervariablen nacheinander einschaltet, nennt man Kaskadenregelung. Die Aktivierung der jeweils nächsten Regelstufe kann bei einer Kaskadenregelung in verschiedener Weise geschehen. Sie kann einfach eine Zuschaltung sein. Neben der ersten Regelstufe wird jetzt noch die zweite aktiv und so folgend. Das System könnte aber auch die jeweils untere Regelung bei Einschaltung der jeweils höheren abschalten. Jeweilige Umschaltung können auch zeitlich beschränkt/kombiniert werden. Für die Regelung des Innenzustandes, wie etwa die Erhaltung des Energiebedarfs des Agenten, sind Regelkreise

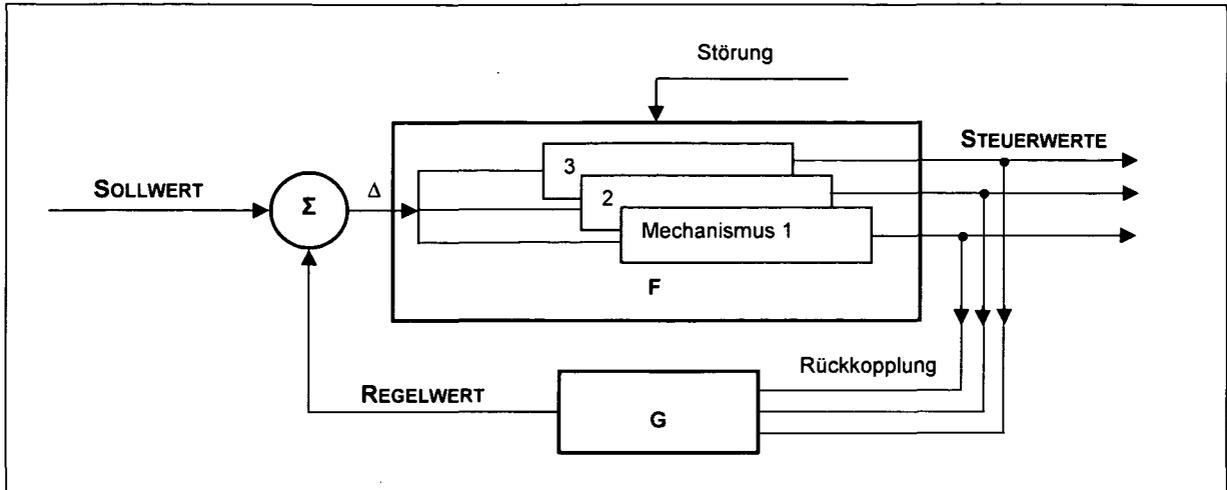


Abb. 3.2. Blockdiagramm der Kaskadenregelung.

auf Abbildung 3.1 und 3.2 sind ausreichend. Was ist aber wenn ein äußerer Zustand angesteuert muss, zum Beispiel eine Aktionssequenz der Industrieroboter?

Industrieroboter stützen sich hauptsächlich auf deterministische weg-abhängige Steuerungsprozesse. Die Sollwerte zu jeder Position sind im Programmspeicher eines Steuerrechners enthalten. Durch Anordnung mehrerer Wegbezugsebenen können verschiedene Freiheitsgrade festgelegt werden. Als Wegaufnehmer dienen Tachogeber, Encoder oder Strichcode-Lineale, die von optischen Sensoren abgetastet werden und Wegstrecken-Zählimpulse liefern. Die Aktionen der Industrieroboter sind ebenfalls ohne Rücksicht auf Umgebungszustand ausgeführt. Wesentlicher Nachteil dabei ist, dass jede Änderung des Umgebungszustandes Fehlerfunktion verursachen kann und im extremen Fall, Zerstörung des Roboters. Um solche Fälle zu vermeiden, muss die Umgebung im Regelungskreis repräsentiert sein. Das ist durch die Umwandlung eines geschlossenen in einem offenen Regelungskreis eingeräumt.

Abbildung 3.3 zeigt Blockdiagramm eines offenen Systems. Der Regelkreis ist zwischen Steuergröße und Regelgröße gewissermaßen aufgeschnitten. Im Schnittpunkt liegt die Umgebung auf die die Steuergröße wirken kann. So ein System haben etwa Protophyten (pflanzliche Einzeller). Protophyten sind Organismen, die mit der Umgebung blind interagieren.

Auf die besonderen Bedingungen der Umgebung können sie sich nicht einstellen, da sie nicht auf das Identifizieren ausgerichtet sind. Die Überlebenschancen der Organismen, die die Umgebung besser ausschöpfen können, steigen. Damit war deren weitere Vermehrung und Entwicklung begünstigt.

Alle bis jetzt beschriebenen Systeme sind automatische Systeme. Schließt man dagegen die Umgebung in der Regelkreis ein, gewinnt man die Wurzel für situierte, adaptive und damit autonome Systeme.

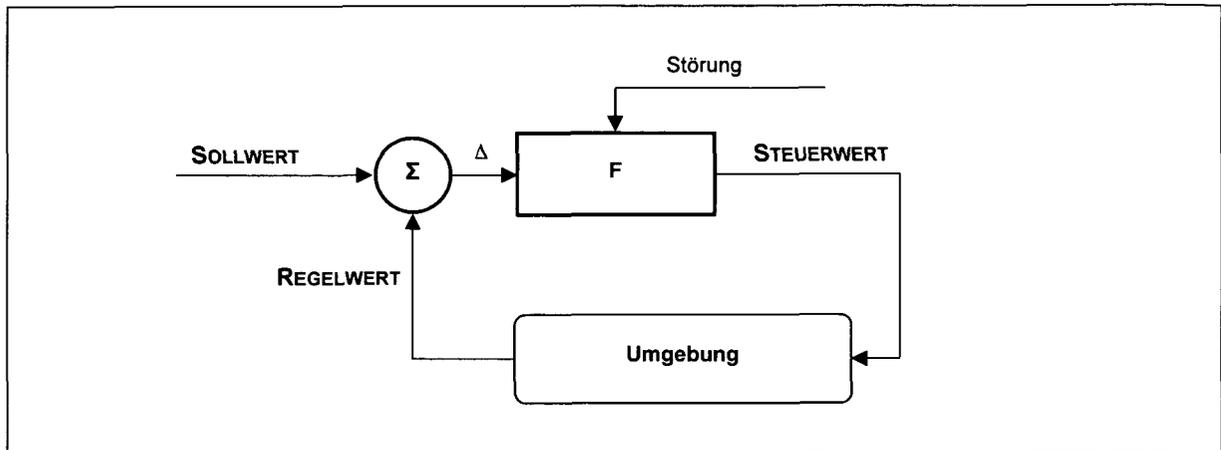


Abb. 3.3. Blockdiagramm der offenen Regelung.

Was das für den oben angeführten Industrieroboter bedeutet? Zusätzliche Regelungsprozesse, die auf Sensorsignale aus der Umgebung basieren, dienen zur Korrektur und Modifikation des Ist-Zustandes. Diese Art adaptiver Regelung dient beispielsweise dazu um Abweichungen vom festgelegten Bewegungsverlauf auszugleichen, der durch unterschiedliche Lastzustände entsteht. Beim Einsatz von festgebundenen Lackier- oder Schweißroboter kann man auf die Erweiterung auf offenen Regelkreis verzichten, denn die Genauigkeit der Roboter durch die starre Aufteilung der Aufgaben und Produktvielfalt kann mit mechanischen Aufbau der Roboter erreicht und die Umgebung passend organisiert werden. Wenn aber ein schienengebundenes Fahrzeug (AGV) oder ein frei beweglicher Roboter in Einsatz kommen soll, dann erweist sich offener Regelkreis als Notwendigkeit. Es ist unmöglich, alle denkbaren Situationen im Bordrechner vorzuprogrammieren. Dazu kommen noch Ungenauigkeiten aller Art, zum Beispiel, Sensoren und Aktoren Ungenauigkeit, Rauschen, Unebenheiten, Radschlupf oder Radabnutzung usw. Erweiterung des offenen Regelkreises mit Sensoren und Aktoren führte zur aktiven Interaktion zwischen Regelkreis und Umgebung in realer Zeit. Der Regelkreis braucht die Veränderungen in der Umgebung nicht mehr nur passiv durchzustehen, sondern kann die Ursachen der Veränderungen direkt beeinflussen. Eine bestimmte Aktion hat unter bestimmten Umständen bestimmte Folgen. Der Regelkreis kann Wirkungen erzeugen, die von Sensorsignalen gelenkt werden und nach Zielen streben. Damit ist zielgerichtetes Verhalten, das den jeweiligen Bedingungen angepasst werden kann, ermöglicht.

Die Verhaltensfähigkeit der Protozoen (tierischer Einzeller) sieht so aus. Für das, was Protozoen tun, gibt es zwar Gründe, aber dieser Gründe sind sich sie nicht bewusst. Dennoch ist ihre Art der Verhaltensfähigkeit der einzig mögliche Nährboden, auf dem Same unserer eigenen Verhaltensfähigkeit wachsen kann (Dennett, 2001). Somit bilden sie den Grundbaustein in der Entwicklung von Adaptiven, Autonomen und Intelligenten Systemen.

3.2 Umgebung – Aufgabe Kontext

Eine Ameise, die sich auf einem Sandstrand auf ihr Nest bewegt, muss vielen Sandkörnern und anderen Hindernissen ausweichen. Von außen betrachtet führt es zu einem sehr komplex gebogenen Pfad. Dieses komplexe Verhalten kommt aber keineswegs dadurch zustande, dass die Ameise intern über komplexe Regeln zur Erzeugung des Pfades verfügt. Vielmehr weicht sie nur immer den gerade vorhandenen Hindernissen aus. Die Komplexität des Verhaltens kommt aufgrund der Interaktion der Ameise mit ihrer Umgebung. Die Interaktion erfolgt automatisch in dem Versuch die Aufgabe, suche nach Nest, zu erledigen.

Um die Evolution nachzuahmen baute man Programme, die sich innerhalb eines Computers reproduzieren können. Erste Programme waren so etwas wie ein Computervirus, allerdings von der ungefährlichen Sorte.

Das Programm hat zwei absichtliche Fehlerquellen. Es gab Kopierfehler, die beim Schreiben eines Wortes im Speicher zufällig ein Bit veränderten und es gab einen Mechanismus, der von Zeit zu Zeit zufällig ein Bit im Speicher umdreht. Beide Fehler produzieren die Vielfalt. Während sich der Speicher des Computers mit Kopien des ursprünglichen Programms füllte, tauchten auf diese Weise Mutationen auf. Einige diese Programme funktionieren nicht mehr und wurden entfernt. Andere begannen sich zu optimieren. Das ursprüngliche Programm bestand aus 80 Bytes Programmcode, aber sehr schnell entwickelten sich daraus funktionsfähige Programme mit weniger als 80 Byte und Parasiten mit nur 22 Byte Codelänge. Parasiten waren nicht in der Lage, sich zu kopieren, aber sie könnten ein größeres Programm so täuschen, dass es sie statt sich selbst kopierte. Unerwartete Resultate brachten neue Idee, eine virtuelle künstliche Evolution zu schaffen und interessante komplexe Wesen würden sich von selbst entwickeln. Damit kann man viel Zeit und Geld sparen, sowie bessere Lösungen erfinden als sonst. Trotz vieler weiterer Jahre Forschung erzielten kaum Interessanteres als die ersten Experimente. Den Hauptgrund dafür ist, dass die Welt, in der Programme lebten, einfach nicht komplex genug war, damit etwas Interessantes passieren konnte (Ray, 1991).

Ein System, das die Möglichkeit zur Interaktion mit der Umgebung besitzt, selbst bei recht einfacher Sensorik, kann ein unerwartet komplexes Verhalten zeigen.

Die Komplexität der Umgebung und die der Aufgaben sind die Voraussetzungen für komplexes Verhalten. Je komplexer Umgebung und Aufgaben sind, desto raffinierter ist die Motorik, Sensorik und die anschließende Verarbeitung der Information. Das heißt, die Komplexität der Umgebung und Aufgaben einerseits und die Komplexität des Agenten andererseits, sollen im Gleichgewicht sein. Das Wesen des Darwinismus ist, dass die Form und Funktion des Systems zusammenwachsen sind.

ERSTES PRINZIP: *Komplexität der Umgebung und Aufgaben verlangt höhere die Komplexität des Agenten.*

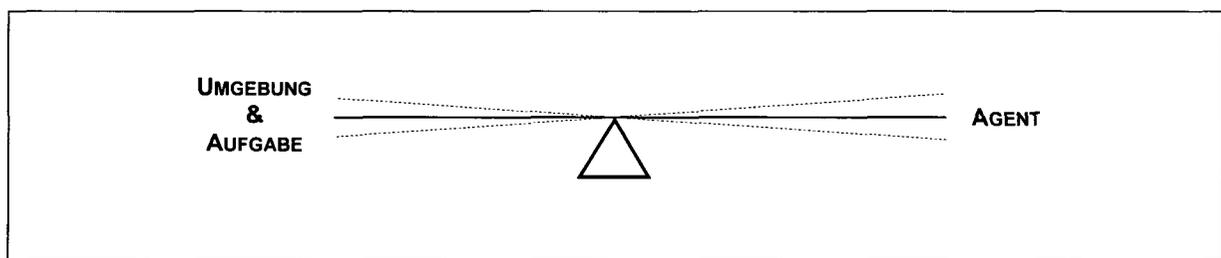


Abb. 3.4. Gleichgewicht zwischen drei Elementen.

Also es ist notwendig sowohl System, als auch Umgebung zu verstehen wenn man die resultierende Leistung des Systems verstehen will. Da taucht ein übersehener Aspekt der Intelligenz auf, nämlich dass die Leistung eines Systems sich nicht unabhängig vom Kontext beurteilen lässt. Genau sowie Verhalten des Agenten. Und dies ist deshalb weil absolute Adaptivität sowie absolute Autonome nicht möglich sind.

Klassische künstliche Intelligenz dagegen propagiert Universalität. Turing Maschine ist universal. Dies ist auch der tiefere Grund für den beschränkten Erfolg der Implementierung klassischer künstlicher Intelligenz Konzepten in der mobilen Roboter Forschung.

Aus dieser Sicht muss noch eine Annahme in Frage gestellt werden. Nämlich der so genannte Descartes Dualismus. Eine Form des Descartes Dualismus ist die Idee dass jedes intelligente System aus zwei separaten und unabhängigen Teilen besteht; geistigen und physischen (Harvey, 2000). Diese Idee prägte klassische künstliche Intelligenz durch strikte Trennung zwischen Steuerungs- und Mechanischenteil. Diese Trennung ist in dem Agentendesign für Einsatz in den realen Umgebungen höchst unerwünscht wenn überhaupt möglich. Das zeigt folgendes Beispiel.

Agenten auf Abbildung 3.5 besitzen zwei Hinterräder, von denen jedes über einen eigenen Motor aktiv betrieben werden kann. An der Vorderseite besitzen sie zwei Sensoren, die die Intensität eines Reizsignals, sei dieses ein Lichtsignal, ein Ton oder ein Duftstoff, registrieren kann. Je stärker der Sensor gereizt ist, desto stärker wird der entsprechende Motor aktiviert. Einziger Unterschied zwischen Agenten besteht in unterschiedlicher Weise wie Sensoren und Motoren verbunden sind. Im ersten Fall das Signal des linken Sensors steuert auf den rechten Motor, das des rechten Sensors den linken Motor. Im zweiten Fall ist es genau umgekehrt.

Ist die Reizquelle, nimmt man an eine Glühbirne, abgeschaltet, so bewegen sich den Agenten nicht. Wird sie eingeschaltet und befindet sie sich auf der rechten Seite des Agenten, wird der rechte Lichtsensor stärker aktiviert, was bedeutet, dass sich der linke Agent genau zur Reizquelle hin bewegt. Und das um so schneller, je näher er der Reizquelle kommt. Am Ende wird der Agent diese Glühbirne durch die Kollision vielleicht zerstören.

Der Agent auf der rechten Seite wird sich im Gegensatz zum linken genau von Reizquelle bewegt weil linken Sensor stärker aktiviert ist, was stärkere Aktivierung des linken Motors nach sich zieht. Er kommt zur Stillstand, wenn sich die Reizquelle genau hinter ihm und in großer Entfernung befindet.

Dieses Beispiel zeigt dass die Lösungen nicht immer explizit berechnet werden müssen, sondern dass sie bereits durch die physikalischen Eigenschaften vorgegeben sein könnten. Das heißt, dass im Design der Agenten Steuerungs- und Mechanischenteil parallel entwickelt sein müssen (Sehe: 5.3 Die Felder). Die Symbiose der Komponenten führen zur Vereinfachung der Steuerung.

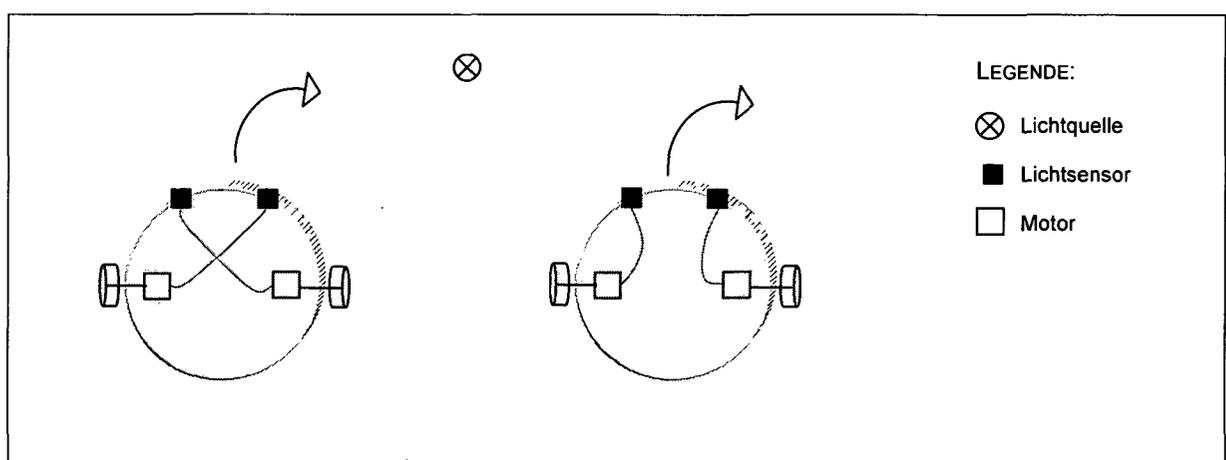


Abb. 3.5. Zwei Agenten, nach Breitenbergs Klassifikation handelt sich um Agent Typ 2 „Aggression“ (linken) und Typ 1 „Fear“ (rechten).

Lebewesen waren weder von Anfang an voll ausgebildet noch blieben sie über Hunderte oder Tausende von Generationen unverändert. Das billigste, am wenigsten gestaltete System wird von Evolution als erstes „entdeckt“ und kurzfristig ausgewählt.

Da in der Welt der Lebewesen zwangsläufig Konkurrenz um Rohmaterial herrscht, entwickelten sich einfache Lebewesen nach und nach. Die Aufgabe, der alle Lebewesen gegenüberstehen, kann man als eine Version des kindischen Versteckspiels betrachten. Man sucht, was er braucht, und versteckt es vor denen, die brauchen, was man hat. Diejenigen, die dies nicht oder auch nur weniger gut konnten, sind ausgestorben. Zufällig erschaffene Organismen die überlebten entwickelten sich weiter. Dabei bringt der evolutionäre Prozess nicht die beste Lösung, sondern diejenige Verbesserung, die nahe liegt und einen Vorteil gegenüber Konkurrenten oder Feinden verschafft. Dadurch bildet sie aber nur ein kurzfristiger Vorteil, denn dies treibt Weiterentwicklung anderer Lebewesen in derselben Umgebung. Dadurch entsteht eine Vielzahl verschiedener Fähigkeiten, die in die Umgebung passen, in welcher ein Lebewesen lebt. Das heißt dass, alle Lebewesen speziell auf ihre gewöhnliche Umgebung „konstruiert“ sind. Bienen, Wespen, Fliegen die erstaunliche Leistungen in komplexen Umgebungen wie Wiesen oder Wälder zeigen können, sind gegenüber einer Fensterscheibe machtlos.

Die Natur hat alle diese Wesen mit einer Reihe von fixen Verhaltensregeln ausgestattet, die es ihnen gewöhnlich erlauben, in ihrer Welt zu überleben. Das heißt, Universelle Agenten sind nicht reale Anforderung.

Der entscheidende Schritt für eine erfolgreiche Einführung von Agenten in realen Anwendungen findet sich in der Anpassung von Agenten an die Umgebung sowie in der Individualisierung gegenüber den zukünftigen Aufgaben.

ZWEITES PRINZIP: *Man soll spezialisierte Agenten mit Aufgaben - Umgebung angepasstem Bau anvisieren.*

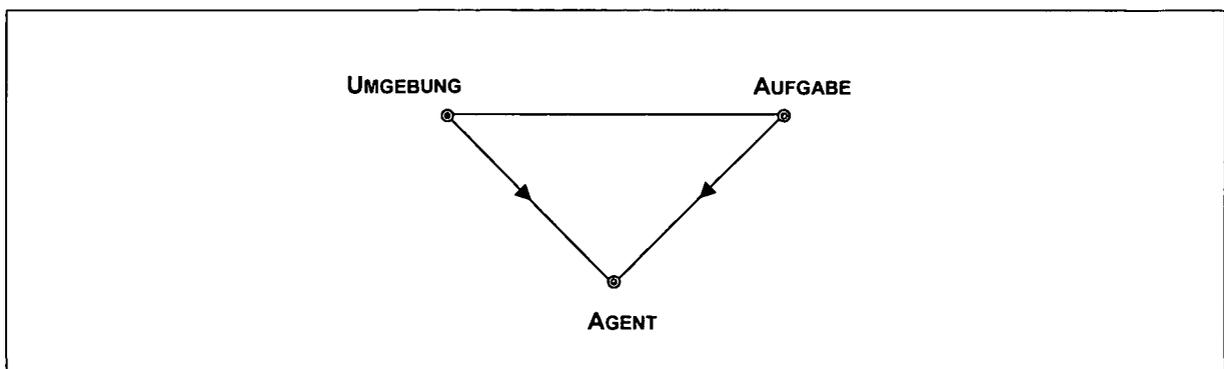


Abb. 3.6. Abhängigkeit der Agenten Umgebung und Aufgabe.

3.3 Muster

Die Evolution ist eine Irrfahrt, durch die erstaunliche Menge an Konstruktionen, die durch die Zeit getestet und herauskristallisiert wurden. Evolution selbst hat weder Zweck noch Ziel. Eben die Resultate der Evolution sind hervorragend gestaltete Systeme. Jedes Lebewesen ist ein hervorragend ausbalanciertes System interagierender Regelkreise von Sensorik und Aktorik und verhält sich sehr aktiv gegenüber seiner Umgebung. Unser Körper, Sensoren, Aktoren und Neuronnetzwerke sowie das Gehirn sind adäquate ob nicht ideale Ausstattung (Wilson, 2000). Für das Design von Agenten ist deshalb Vorteilhaft zu untersuchen, ob es bei Entwicklung der Komplexität des Lebewesens ein Muster gibt. Also in welchem Zusammenhang haben sich Körper, Sensorik, Aktorik, und Steuerungssystem entwickelt.

Alle Lebewesen haben einen Körper, der eine sich selbst steuernde und selbst schützende Organisation erfordert, die unter unterschiedlichen Bedingungen unterschiedlich aktiviert werden kann. Diese Organisation wurde auf der untersten Ebene aus einer Riesenzahl passiver Schalter, die je nach den ebenso passiven Bedingungen, auf welche die Lebewesen bei ihrer Wanderung trafen, ein- oder ausgeschaltet werden (Dennett, 2001). Diese Organisation besteht aus dem Stoffwechselsystem, dem Immunsystem und den komplexen Selbstreparatur- und Gesunderhaltungssystemen. Die Kommunikationskanäle dieser ersten Systeme waren nicht die Nerven, sondern die Blutgefäße. Lange bevor es in den Organismen das Nervensystem gab, benutzen die Lebewesen den Flüssigkeitskreislauf im Körper, der zwar langsam, aber zuverlässig Informationen dahin transportierte, wo sie zur Steuerung und Selbsterhaltung gebraucht wurden. Die Weiterentwicklung dieses uralten Kommunikationskanals findet man heute sowohl bei Tieren als auch bei Pflanzen. Die Flüssigkeitsströme in den Pflanzen sind ein Mittel, um Signale zu transportieren. Entwicklung von Bewegungsapparat bei Tieren dagegen setzte größere Maßstäbe auf Menge und Geschwindigkeit des Informationsaustausches innerhalb des Tiers. Das brachte eine wichtige Neuentwicklung: die Evolution einfacher Nervensysteme, Vorläufer des heutigen Nervensystemen, die Informationen schneller und wirksamer übertragen konnten.

Um den Evolutionsvorgang und dessen Folgen auf Entwicklung des ausbalancierten ausgeglichenen Systems weiter zu analysieren, sei als Ausgang der Entwicklung der Menschenhand angenommen.

Der erste anatomisch zweifüßige Vorfahr des Menschen, der entdeckt wurde, lebte vor etwa 3,2 Millionen Jahren in Ostafrika. Australopithecien besaß eine affenunähnliche Hand und hatte ein schimpansengroßes Gehirn.

Neue Lebensweise auf dem Erdboden begünstigte die Weiterentwicklung der aufrechten Gangweise. Dies löste Veränderungen der oberen Gliedermaßen aus, die das Repertoire der Handbewegungen so veränderten, dass der Werkzeuggebrauch begünstigt wurde. Auch neulich eroberte Lebensräume in der Savanne gab es genug Gründe, die Hand zu erhalten

und Weiterentwickeln. „Dank der Dreh- und Gleitbewegungen des Schulterblatts waren Australopithecien in der Lage, Gegenstände, die sie in der Hand hielten, mit einer über dem Kopf ausgeführten, beschleunigten Bewegung zu schleudern. Um dieses ballistische Potential, in der Savanne herzlich Willkommen, völlig auszuschöpfen, mussten noch einige Verbesserungen des visuo-aktorischen Steuersystems stattfinden. Zweifellos hat die Evolution die erste Fähigkeit genutzt, um die zweite auszubilden“ (Wilson, 2000).

Das Gehirn und der Bewegungsapparat entwickeln sich als Organe genau wie Organismen selbst, nämlich durch eine Modifizierung von Bau und Funktion im Laufe der Zeit.

Infolgedessen manifestieren sich im Verhalten jedes lebenden Mitglieds einer Art zu jeder Zeit die funktionellen Merkmale der einzelnen Teile des Körpers im allgemeinen, und des Gehirns und Bewegungsapparat im besonderen (Washburn, 1960).

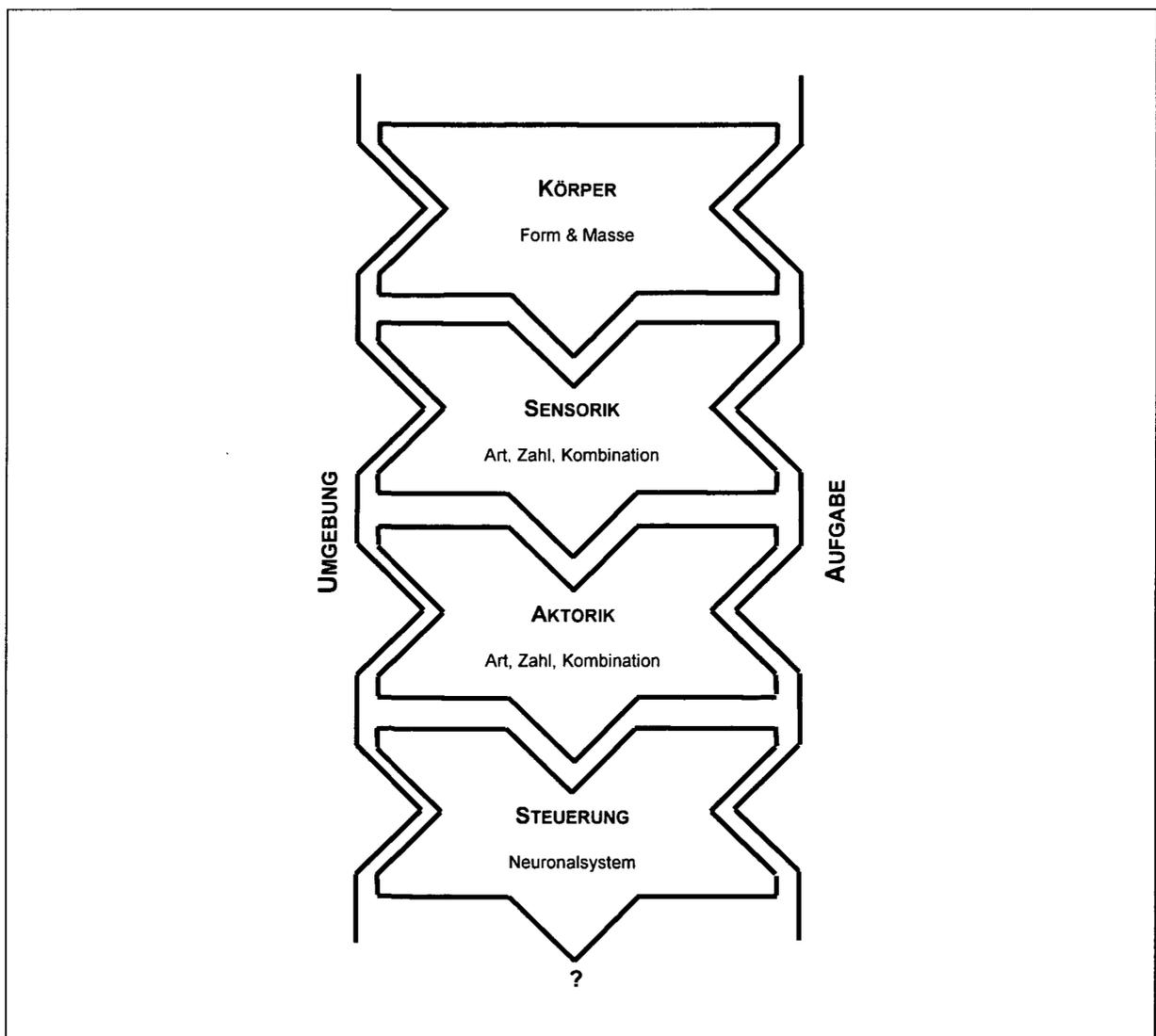


Abb. 3.7. Schaffung des ausbalancierten ausgeglichenen Systems.

Der Körperbau des Lebewesens ermöglicht sein Verhalten. Alle diese und ausschließlich menschliche Verhaltenseigenschaften müssen in dem langen Prozess der Gehirnvergrößerung entstanden sein (Wilson, 2000). Dieser Prozess begann mit der Ausweitung des neuartigen, innovativen Werkzeuggebrauchs und mit den unzähligen neuen Erfahrungen und Umwelten, die sich aus der erfolgreichen Anwendung solchen Verhaltens zwangsläufig ergeben. Das Gehirn habe an dieser Entwicklung erst als letztes Organ teilgehabt (Washburn, 1960). Anatomische Veränderungen lösten Steuerung und damit Verhaltens Veränderungen aus.

Aufgrund dieser Betrachtung machen wir ein Gedankenexperiment, indem wir uns überlegen, was passieren würde, wenn man dem Agentn auf Abbildung 3.5 ein mächtiges Gehirn einbauen würde. Was könnte man damit erreichen? Es wird rasch klar, dass es in diesem System gar keine Aufgabe für ein solches Gehirn gibt. Bei einem einfachen System von zwei Sensoren und zwei Motoren bringt ein leistungsfähiges Gehirn nur etwas, wenn auch die Sensoren und Motoren angepasst werden. In der klassischen künstlichen Intelligenzforschung wurde das Schwergewicht sehr stark auf das Gehirn gelegt. Dieses Beispiel illustriert aber, wieso auch ein leistungsfähiges Gehirn nichts nützt, wenn es nicht in der Lage ist, mit seiner Umwelt zu interagieren. Aus der heutigen Sicht der künstlichen Intelligenz ist es deshalb sinnvoll, von einfachen Systemen von Sensoren und Motoren mit nur wenig interner Verarbeitung auszugehen.

3.4 Design von Agenten

Die erste Stufe beim Design eines Agenten ist eine sorgfältige Analyse der Umgebung und der Aufgabe. Dadurch kann man situationsbezogene und zielgerichtete Verhaltensrepertoire des Agenten bestimmen.

Die größte Herausforderung dabei liegt bei sehr enger Interaktion zwischen dem Agenten und der Umgebung und setzt somit tiefes Verständnis der Beziehung zwischen Agent und Umgebung voraus. Nur wenn Agentenverhalten auf beide Anforderungen gleichzeitig angepasst ist, kann ein Beobachter das Verhalten als intelligentes Verhalten bezeichnen.

Wenn man in der realen Umgebung handlungsfähigen Agenten bauen will, muss man ihn erstens, der Umgebung und Aufgaben entsprechenden, Körper schaffen. Um formale Modelle körperlicher und physikalisch situierter Agenten zu aufzustellen, bieten sich systemdynamische Techniken automatisch an. Was den Aspekt der Körperlichkeit betrifft, so sind Agenten- sowie Tierkörper durch den Physischen Gesetze geprägte, dynamische Systeme.

Die verkörperten Agenten sollten, für entsprechende Interaktion mit der Umgebung, passend ausgestattet sein. Aufgrund der Agentenkörperform bildet man weiter passendes

Sensorsystem ein. Da muss der Typ der Sensoren sowie deren räumlichen Verteilung auf dem entsprechenden Agentenkörper berücksichtigt sein. Davon hängt die notwendige Komplexität des Informationsbearbeitungsteils des Agenten.

Wenn alle, auf der Abbildung 3.7 genannte Komponenten, im Gleichgewicht sind, dann geschehen zielgerichtete, situationsbezogene Interaktionen des Agenten in der Umgebung automatisch.

Eine zuverlässige, zielgerichtete Bewegung ist eine unumgängliche Voraussetzung zur Entwicklung eines autonomen Agenten. Pfad des Agenten ist einerseits durch Zielposition, dynamische Hindernisse und der Begrenzung des Kernsubsystems des selbstorganisierenden Montagesystems und andererseits durch Agenten Form, Sensorik-, Steuerungs- und Aktoriksystem determiniert.

3.4.1 Form

Die Form eines Agenten hat einen großen Einfluss auf seine Leistungsfähigkeit. Ein nicht zylindrischer Agent läuft eher Gefahr als zylindrischer, bei einer ungünstigen Anordnung der Hindernisse, zu engen Passagen oder überfüllten Produktionsflächen steckenzubleiben.

3.4.2 Antrieb

Das Antriebssystem soll so konzipiert sein, dass der Agent keine komplizierten Rangier- und Wendemanöver auf der flachen Oberfläche durchführen muss. Deshalb kommt eine Anordnung von zwei unabhängig angetriebenen Rädern und einem freilaufenden Schlepprad zum Einsatz. Dabei muss das geometrische Zentrum des Agenten mit dem kinematischen Zentrum zusammenfallen. Das kinematische Zentrum befindet sich in der Mitte zwischen zwei angetriebenen Rädern. Der Agent ist dadurch in der Lage, sich auf der Stelle zu drehen ohne weiteren Raum einzunehmen. Diese Anforderung an die Konstruktion ist sehr wichtig, denn der Agent dreht sich wenn Hindernis detektiert wird, daher in der Nähe des Hindernisses. Zum Beispiel die Experimentierplattform MOBOT-V ist mit einem Differentialantrieb ausgestattet, wobei das am hinteren Ende zwischen den Antriebsrädern liegende kinematische Zentrum nicht mit dem geometrischen Zentrum des Chassis zusammenfällt. Folge dessen ist das diese Plattform sehr oft in der Abweichungsmanöver noch näher an die Hindernisse gerät (Ferner, 1996).

Differentialangetriebene Agenten sind nicht-holonomen Agenten, denn sie besitzen zwei Freiheitsgrade; Translation parallel auf die Räderachsen und Rotation. Dritter Freiheitsgrad, nämlich sich seitwärts, senkrecht auf Räderachse zu bewegen haben sie nicht. Für die Bewegung auf den flachen Oberflächen ist dies auch nicht notwendig. Damit ist auch

Bewegungssteuerung des Agenten einfacher zu realisieren, denn die Bewegung des Agenten durch Angabe der Linear- und Winkelgeschwindigkeit ist determiniert.

3.4.3 Sensorik

Damit ein Agent autonom in einer Umgebung sich bewegen kann, muss er mit Sensoren, die seine Umgebung wahrnehmen, ausgestattet sein. Agenten sind inhärent unsicher über den Zustand ihrer Umgebung. Ungewissheit ergibt sich aus Sensorenbeschränkungen und der Tatsache, dass die dynamischen Umgebungen unvorhersehbar sind. Im Bereich der Agentennavigation verwendet man fünf Sensorentypen: Odometrie, Vision, Sonar, Laserscanners und Infrarotsensoren.

Die Odometrie besteht aus Sensoren, welche die Drehbewegung der Räder messen können. Auf diese Weise wird bestimmt, wie lang der zurückgelegte Weg des jeweiligen Rades ist. Im Fall des differentialen Antriebs kann die Richtungsänderung des Agenten aus der Differenz der beiden gemessenen Wegstrecken ermittelt werden. Die Odometrie ist auf kurzen Strecken sehr genau, sie hat aber den Nachteil, dass die Messung des zurückgelegten Wegs mit zunehmender Strecke immer ungenauer wird. Somit wird die Positionsschätzung zunehmend schlechter. Zur Verbesserung der Positionsbestimmung wird zusätzlich ein Kreiselkompass eingesetzt. Mit diesem Sensor kann recht genau die Orientierung des Agenten bestimmt werden. Nachteil dieses Sensors ist, dass die Orientierung mit der Zeit ungenau wird (Drift), selbst wenn sich der Agent im Stillstand befindet.

Videosysteme liefern eine zu große Datenmenge und die Extraktion von Information ist schwierig und zeitaufwendig. Darüber hinaus sind Videosysteme anfällig gegen verschiedene Beleuchtungsverhältnisse. Die Entfernungen kann man nur indirekt durch den Vergleich von mindestens zwei aufgenommenen Bildern berechnen.

Das Datenvolumen von Sonar, Laserscanners und Infrarotsensoren ist dagegen gering. Deswegen sind die Signale die diese Sensoren liefern einer Verarbeitung in Echtzeit zugänglicher.

Sonar- oder Ultraschallsensoren werden zur Entfernungsbestimmung benützt. Sonar sendet dabei ein Signal im Ultraschallbereich aus und wartet auf ein rückkehrendes Echo. Das Signal breitet sich kegelförmig in die Abstrahlungsrichtung des Sensors aus. Der Öffnungswinkel liegt im Bereich von 5° bis 30° . Das Signal wird an der Oberfläche eines in diesem Bereich befindlichen Objekts reflektiert und kehrt zum Sensor zurück. Das Sonar misst die Zeit zwischen Aussenden und Empfangen des Signals und bestimmt so die Entfernung. Der Vorteil von Sonar ist, dass sie preiswert sind und wegen ihrer handlichen Größe nahezu überall montiert werden können. Nachteilig ist dass man wegen des großen Öffnungswinkels nicht die genaue Richtung kennt, in die eine Messung durchgeführt wurde,

da das erste Objekt in dem Öffnungsbereich die Entfernung bestimmt. Ein weiterer Nachteil ist dass das Sonar an glatten Oberflächen abgelenkt wird und kein Echo an den Sensor zurückgeht. Ein ähnliches Problem tritt beim gleichzeitigen Einsatz mehrerer Sonare auf. Hier kann es passieren, dass ein Sensor die ausgestrahlten Signale eines anderen Sensors empfängt und als eigenen interpretiert.

Laserscanner haben ein ähnliches Messprinzip wie Sonarsensoren. Auch hier wird ein Signal in eine Richtung ausgesandt und auf die Rückkehr dieses Signals gewartet. Die ausgesandten Signale bewegen sich jedoch mit Lichtgeschwindigkeit, was zu einer deutlich kürzeren Signallaufzeit führt, aber auch schnellere Zeitmessungen erfordert. Das Signal wird wieder kegelförmig ausgestrahlt, der Kegel hat nun aber einen sehr kleinen Öffnungswinkel, so dass man näherungsweise von einem Strahl ausgehen kann. Durch Rotation des Sende- und Empfangskopfes oder Ablenken des Lasersignals an einem Spiegel kann man Entfernungsmessungen in mehrere Richtungen durchführen. Laserscanner können sehr genaue Abstandsmessungen liefern. Genauigkeit ist kleiner als 1 cm.

Bei Infrarotsensoren nachteilig ist, dass der Agent nur in die Richtungen Hindernis abtasten kann, denn Öffnungswinkel sehr klein ist. Daraus resultiert die Frage wie viel und wo soll man Infrarotsensoren einbauen um zufrieden stellende Resultate zu bekommen.

Für die Anwendung in der industriellen Umgebung und für die Aufgabe Vermeidung von Kollisionen mit Hindernissen und den Begrenzungen des Kernsubsystems scheint Laserscanner der ideale Kompromiss. Dabei beschränken sich die Daten des Scanners auf zweidimensionale Abstandsprofile, die in der Horizontalen aufgenommen wurden. Beim Entwurf von Agenten Umgebung muss aber berücksichtigt sein, dass die Umgebung keine stark absorbierte, Glas oder Spiegel Oberflächen beinhaltet.

3.4.4 Energieversorgung

Jeder Agent sowie jedes Lebewesen hat eine gewisse Energiemenge zu Verfügung, kann also nur eine gewisse Gesamtleistung erzielen. Daher hat Energieeinsparung im Prozess des Agentenentwurfs absolute Priorität.

Die Technik setzt traditionellerweise Konstruktionen aus Elementen Additiv zusammen. Jedes Element dabei hat eine Funktion und es wird auf die Maximierung dieser Funktion hin ausgelegt. Im Gegenteil wird hier strikt auf eine Maximierung verzichtet. Vielmehr steht die Optimierung des gesamten Agenten im Vordergrund. Integrierte, parallele Entwicklung von Körperbau, Sensorik und Aktorik führt zu dem multifunktionalen Design der Elemente. Dadurch kann man die gesamte Masse der Agenten und damit auch Energieverbrauch verringern.

3.5 Zeit

Im Gegensatz zur Informatik, die sich mit der Transformation von Informationen vom Input zum Output befasst, geht es in der autonomen Agenten um die Transformation der physikalischen Welt von einem Inputzustand zu einem Outputzustand.

Im ersten Fall ist die genauere Bestimmung der Zeit irrelevant. Bei Nutzung einer Software mindestens Geschwindigkeit der Bearbeitung unser Befehl ist durch unser Gefühl für die Langweile bestimmt. Es ist vorteilhaft dass das System so schnell wie möglich reagiert. Die Verwendung von rechenleistungsfähigerer Hardware und besseren Software kann sicherlich dieses Problem weniger kritisch machen.

Beim Verhalten spielt die Zeit wesentliche Rolle, denn jedes Verhalten hat eine zeitliche Ausdehnung. Tätigkeiten wie Greifen, Gehen oder Sprechen bestehen aus einer genau aufeinander abgestimmten Folge einzelner Körperbewegungen. Das Gehirn darf den Muskeln nicht nur ein einziges Mal einen einzigen aktorischen Aktivitätsvektor liefern, sondern es muss ihnen zur Steuerung eine kontinuierliche Abfolge motorischer Vektoren übermitteln, die dann gleichmäßige Bewegungen erzeugen. Die Geschwindigkeit der Transformation von Input- zu Outputzustand muss in der zeitlichen Abstimmung mit der Dynamik der Umgebung sein. Dies ist gerade in physischen Umgebungen notwendig, in denen sich der Zustand der Umgebung auch ohne das Zutun des Agenten verändern kann. Damit der Agent in einer physischen Umgebung überleben kann, sind rechtzeitige, und nicht nur schnelle, Aktionen notwendig. Genauer gesagt die Geschwindigkeit von den relevanten Veränderungen in der Umgebung bestimmt die optimale Antwortzeit. Der Begriff die optimale Antwortzeit von Aktionen ist allerdings verschwommen und lässt sich im Allgemeinen nicht exakt definieren. Im Einzelfall aber, kann ein Beobachter entscheiden, ob ein Agent diese Bedingung erfüllt oder nicht.

Unsere Denkgeschwindigkeit ist mit den Abläufen der physischen Gesetze bestimmt. Mindestdenkgeschwindigkeit erlaubt uns im normalen Zustand reibungslos zu funktionieren. Im Fall dass ein Gebäude auf uns stürzt, schalten wir auf instinktive Reaktionen, die schnellere Antwort auf Situation erlauben. Ebenfalls ausgelöste Reaktion kann sich auch als nicht intelligente erwiesen.

Wenn Aktion eines Agenten zu langsam ist, und die Veränderung, die durch die Aktion ausgelöst wird spät eintrifft, dann verliert sie ihre Gültigkeit. Dieses Problem haben alle klassischen künstlichen Intelligenz Systeme (Shakey & Co.) sowie herkömmliche Produktionssysteme.

Echtzeitberechnung ist nicht darüber, "schnelle" Systeme zu bauen; sie ist darüber, Systeme in gut angegebene Weisen einzubauen, die voraussagbar "schnell genug" ihre Umgebungen zu bearbeiten sind (Musliner, 1995).

DRITTES PRINZIP: *Zeitabstimmung zwischen der Reaktion des Agenten und der Dynamik der Umgebung ist Voraussetzung für die reibungslose Funktion des Agenten.*

Steigerung der Leistungsfähigkeit der in den Agenten eingebauten Hardware (hier ist nicht nur die Steuerungshardware, sondern auch alle Komponenten des Agenten gemeint, d.h. Körperbau, Sensoren, Aktoren usw.) und Software ist nicht unbedingt der richtige Vorgang. Die tatsächliche Lösung liegt eher in der Verbindung zwischen abhängigen Faktoren, und in deren zeitlichen Abstimmung.

Es zeigt noch einen leicht übersehbaren Aspekt der Intelligenz auf, der Begriff der Intelligenz hat keinen Sinn wenn die Zeit für die Aktion (erweitert auf Verhalten und Handlung genau so) nicht vorgegeben ist.

Die Kontrolle des Agenten

Die Art wie Signale von Sensoren in Aktionen transformiert werden nennt man die Kontrolle des Agenten. Wie diese Transformation abläuft wird mit Architektur des Kontrollsystems definiert.

Die Kontrollparadigmen haben sich auf Uneinigkeiten über die Natur der Intelligenz entwickelt. Es gibt zwei Kontrollparadigmas, die Extrempositionen bilden; Reaktive und Kognitive. Der Raum dazwischen bildet den Rand für Hybride Kontrolle, die aus der Kombination von Reaktiven und Kognitiven Kontrollparadigma besteht.

Diese übliche Darstellung resultiert aus der Annahme von Gegensätzen der gegebenen Kontrollparadigmen. In der Tat handelt es sich hierbei nicht um Gegensätze sondern um verschiedene Entwicklungsstufen. Ein biologisches reaktives System, das anfangs noch kein Gedächtnis besaß, wurde im Laufe der Evolution mit neuronalen Strukturen ausgestattet die so dann als Basis für ein Gedächtnis dienen konnten. Die reaktiven und kognitiven Systeme stellen sich nicht als logisch ausschließende Alternativen dar, sondern die letzteren repräsentieren eine entwickelte Form reaktiver Systeme. Dies legt die hier vertretene These nahe, dass man zum Verständnis kognitiver Systeme zunächst Systeme untersuchen muss, die rein reaktiv sind. Sensor-aktorische Systeme sind also vermutlich näher an der Kognition als die in Rahmen der klassischen künstlichen Intelligenz untersuchten, sensorisch dominierten Wahrnehmungsprozesse.

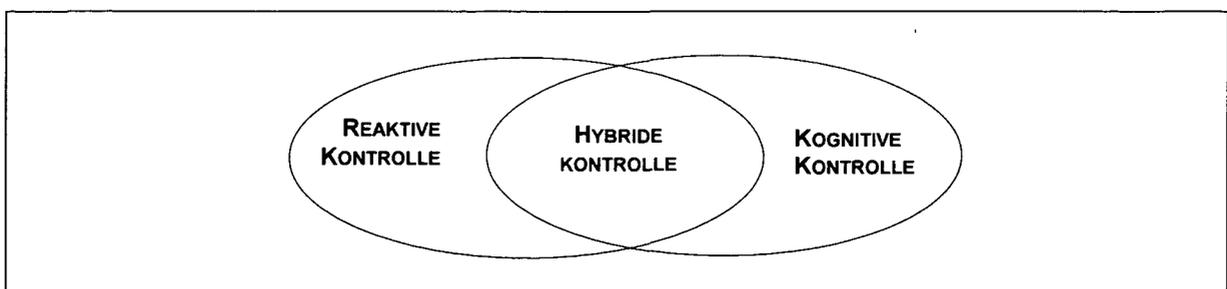


Abb. 4.1. Kontrollparadigmen.

4.1 Kognitive Kontrollparadigma

Das meiste an früheren Arbeiten an Kontrollsystemen für den Einsatz in Robotik wuchs aus dem was heute oft als klassische künstliche Intelligenz bezeichnet wird. Diese ist im Originalvorschlag von Marvin Minsky determiniert: *“An intelligent machine would tend to build up within itself an abstract model of the environment in which it is placed. If it were given a problem it could first explore solutions within the internal abstract model of the environment and then attempt external experiments.”* (Minsky, 1973).

Der Vorgang der Kognition wird 1979 von Popper kennzeichnend beschrieben. Seine Eimertheorie des Denkens schließt an die Ansicht dass wir nicht über etwas denken können, das uns nicht zuvor durch unsere Sinne erreicht hat. Diese Analogie vergleicht Kognition mit einem Eimer, der sukzessiv mit Wissen durch unsere Sinnesorgane gefüllt wird. Dieses Wissen wird in symbolbasierten Weltmodellen repräsentiert. Das Weltmodell ist explizite formale Darstellung der Umgebung samt ihren Ereignissen. Das Verhalten des Agenten ist demzufolge durch einen Zyklus bestehend aus Wahrnehmung, Modellierung, Klassifikation, Planung und Aktion bestimmt. Die Bearbeitung der einzelnen Schritte erfolgt in einer festen zeitlichen Abfolge, an deren Ende die Erzeugung der Steuerungsgrößen, die an die Aktoren des Agenten übergeben werden, steht. Aufgrund der unveränderbaren zeitlichen Abfolge der Bearbeitungsschritte, können solche Steuerungsprogramme durch das Adjektiv „sequentiell“ charakterisiert werden. Die Funktionsmodule sind hierarchisch und das Gesamtsystem ist in mehreren Schichten eingeteilt, die aus den Hierarchieebenen der einzelnen Funktionsmodule bestehen. Eine direkte Verbindung ist nur zwischen zwei benachbarten Schichten möglich.

Der Ablauf in einer solchen Architektur ist üblicherweise wie folgt: Zuerst werden die Signale aus allen Sensoren gesammelt. Dann wird versucht, Rauschen und Konflikte aufzulösen, so dass ein konsistentes Weltmodell modelliert werden kann. Das Weltmodell muss die geometrischen Details allen zu dem fehlerfreien Betrieb des Agenten notwendigen Objekts in der Agent Umgebung sowie ihre Positionen und Orientierungen enthalten. Nur dann kann kognitiver Agent die eigene Umgebung klassifizieren und somit erkennen. Danach wird für das gegebene Ziel eine Sequenz von Aktionen geplant, die zu diesem Ziel führt. Schließlich wird der formulierte Plan ausgeführt, indem passende Befehle zu den Aktoren geschickt werden.

Um in der Lage zu sein, sich mit verlässlichen Plänen zu befassen, muss die Weltmodellierung vollständig und sehr genau sein. Bei dem Versuch, ein solches Modell zu konstruieren, treten aber viele Probleme auf. Erstens, die reale Welt ist voller Details, und Methoden für ihre Repräsentation brauchen viel Speicher. Zweitens, Sensordaten sind unvermeidlich verrauscht und Sensoren sind systematischen Fehlern unterworfen. Dies führt dazu, dass Sensorfusion, die Kombination von Signalen aus mehreren Sensoren zur Erzeugung einer Datenstruktur benutzt wird. Dies ist in der Regel sehr rechnerintensiv.

Je mehr Zeit das Programm für die Verfeinerung des Weltmodells und Optimierung des Plans braucht, desto länger wird die Verzögerung zwischen Signalempfang und Aktion. Diese Verzögerung erhöht die Wahrscheinlichkeit, dass eine signifikante Änderung in der Umgebung stattfindet, und dass der Plan somit ungültig wird. Die Umsetzung des Planes in eine Aktionssequenz führt dann zu einem sinnlosen Verhalten des Agenten.

Daher versucht man den Speicher und die Manipulation des Weltmodells praktikabel zu machen, indem die Umwelt vereinfacht wird. Es gibt aber kein Mittel, das die Art und Weise der Beziehung zwischen einer bestimmten Eigenschaft des Objekts und einer Agentenaktion definiert. Da es diesbezüglich kein geeignetes Mittel gibt, resultiert aus zwei ungelösten Problemen; Das Frame und Symbol Grounding-Problem.

Das Frame-Problem, das die Angabe eines wirkungsvollen Formalismus verlangt, der beschreibt, was sich in der Umwelt eines Wesens ändert und was konstant bleibt. Fehlt der Formalismus, so führt die Beschreibung von Handlungen und deren Konsequenzen zu kombinatorischen Explosion. Viele Autoren, insbesondere Kritiker der künstlichen Intelligenz argumentieren, dass das Frame-Problem eine Konsequenz des Logik-orientierten Zugangs zu künstliche Intelligenz und Artificial Life ist. Benutzt man für die Beschreibung kognitiver Prozesse den Sprachrahmen, der durch die klassische (zweiwertige) Logik aufgespannt wird, dann führt die Beschreibung kognitiver Prozesse zu Systemen bzw. Modellen, die geschlossen sind (Goldammer & Paul, 1996). Dagegen sind alle biologischen kognitiven Systeme offene Systeme. Damit stellt sich die Frage ob ist es überhaupt nützlich, derartige Systeme formal zu beschreiben, um sie anschließend technisch zu realisieren? Tiere (wie auch Menschen) haben weitaus begrenzter Berechnungskapazitäten was das Ziehen von Schlüssen betrifft, dennoch sind sie in der Lage, mit komplizierten Situationen fertig zu werden.

Das Symbol Grounding-Problem (Harnad, 1990), das sich auf die Frage nach der Bedeutung einer "Wissenssprache" bezieht: Wie kann die Bedeutung von definitionsgemäß bedeutungsfreien Symbolen, die allein auf Grund ihren formalen Eigenschaften verarbeitet werden, in etwas anderem als in anderen bedeutungsfreien Symbolen verankert sein? Wie schaffen es natürliche Lebewesen diese Probleme zu überwinden und können wir dieses Prinzip für Agenten einsetzen? Auf den ersten Blick scheint der Implikationen-Berechnende Agent einer Informationsüberflutung gegenüber zu stehen. Dieser Begriff sagt, dass das betreffende Wesen mit einem Ausmaß an Information konfrontiert ist, das seine kognitiven Fähigkeiten überschreitet. Wie kann man unter solchen Bedingungen jemals zu einer Lösung gelangen? Die Antwort muss daher lauten: Kognition besteht nicht darin, Information oder Fakten abzuarbeiten, sondern darin, diese in geeigneter Weise zu ignorieren.

Ein weiterer Nachteil der kognitiven Kontrolle ist die mangelnde Robustheit. Wegen der engen gegenseitigen Abhängigkeit der Module (Komponenten), führt der Ausfall einer

Komponente (z.B. wenn der Planer keinen Plan konstruieren kann) zum Ausfall des ganzen Systems.

4.2 Reaktive Kontrollparadigma

Wegen der Erkenntnis, dass eine symbolische Repräsentation der Umgebung die Problemquelle ist, haben einige Forscher versucht, Alternativlösungen zu finden. Mitte der achtziger Jahre entstand der reaktive verhaltensbasierte Ansatz.

Während kognitive Agenten auf der Existenz und Manipulation eines globalen Weltmodells basieren, vertreten reaktive verhaltensbasierte Systeme die andere extreme Ansicht, dass ein solches internes Modell unnötig ist. In der Literatur findet man oft die Aussage von Rodney Brooks: „*Die reale Welt dient als ihr bestes Modell.*“ (Brooks, 1991). Das Verhalten des Agenten entsteht nicht durch die Umsetzung eines Planes, sondern durch eine direkte Koppelung zwischen den Sensoren und den Aktoren des Agenten.

Bereits einfache Organismen reagieren auf Reize aus der Umgebung. So ziehen Amöben ihre Pseudopodien zurück, wenn man sie mit einer Nadel sticht. Auch bei den Menschen, existieren neben den kognitiven Prozessen auch ausschließlich reizgesteuerte Mechanismen.

Bekannt ist der Patellarsehnenreflex, der bei einem Schlag gegen die Patellarsehne unterhalb der Kniescheibe eine Muskelkontraktion im Oberschenkel auslöst. Aber auch die meisten Körperfunktionen des Menschen sind datengesteuert und kognitiv nicht penetrabel, daher können sie weder beeinflusst noch unterdrückt werden (Strube, 1998). Tritt der auslösende Reiz wird unwillkürlich der entsprechende Reflex ausgeführt.

Obwohl dies auf den ersten Blick einfach erscheint, ist die erreichte Leistung der reaktiven Agenten beachtenswert.

Reaktive Kontrolle besteht also darin, aufgrund von Sensorsignalen, die momentane Situation in der Umgebung widerspiegeln, eine passende Reaktion auszuführen. Auswahl der Reaktionen des Systems auf bestimmte Umgebungssituationen sind mit auslösenden Mechanismen definiert (siehe: Handlung, Verhalten, Aktion). Auslösender Mechanismus kann zum Beispiel für die Vermeidung von Kollisionen zwischen einem Agenten und Objekten in seiner Umgebung oder für die Haltung des Gleichgewichts des Agenten zuständig sein. Die Leistung der reaktiven Agenten wächst, wenn mehrere auslösende Mechanismen miteinander verbunden sind. Die Interaktionen zwischen solchen Mechanismen können zu Ergebnissen führen, die einem Betrachter als durchdachte Tätigkeit oder sogar als vernünftiges Verhalten erscheinen.

Der Erfolg des reaktiven Ansatzes hat gezeigt, dass komplexes, intelligent scheinendes Verhalten erzeugt werden kann, ohne von Speicher und Modell der Umgebung Gebrauch zu

machen. Die Basiskomponente eines reaktiven Systems ist ein Prozess, der Kognition und Aktion direkt verbindet.

Der Vorteil der reaktiven Kontrolle ist, dass sie sehr schnell ausgeführt werden kann, da Sensorik und Aktorik direkt miteinander verbunden sind. Nachteilig ist, dass eine solche direkte Kopplung sehr starr ist. Weil ein einzelner auslösender Mechanismus, der ein Sensorsignal direkt in passende Aktionen umsetzt, nur ein Teilproblem und nicht die ganze Aufgabe lösen kann, ist es nötig, so viele auslösende Mechanismen wie es Teilprobleme gibt zu entwickeln und diese dann zu verbinden. Daher soll für jede wichtige Umgebungssituation passende Reaktion vorgesehen sein. Oft führt das zu Aktionen des Agenten, die nicht optimal sind.

Auch behauptet werden muss, dass Informationen über die Umgebung immer unvollständig und oft unkorrekt sind. Physisch ist es nicht realisierbar dass, ein Agent den gesamten momentanen Zustand der Umgebung wahrnimmt. Diese Behauptung gilt für alle Agentenarten sowie alle Lebewesen. Im Fall von reaktiven Agenten ist es aber die einzige Informationsquelle, von der die Handlung des Agenten abhängt. Rein reaktive Agenten haben weder Apparate, die die Aufgaben und Umgebung repräsentieren können noch ein Gedächtnis.

Die Merkmale reaktiver Kontrolle sind nicht Optimalität und Vollständigkeit, sondern Schnelligkeit und Robustheit. Dabei ist es Vorteilhaft dass die ausgelöste Aktion zeitlich mit den Umgebungsereignissen abgestimmt ist. Das heißt, die Agentendynamik soll auf Umgebungsdynamik angepasst sein. Robuste Kontrolle bedeutet dass beim Auftreten von nicht-deterministischen Effekten das System nicht zusammenbricht, sondern allenfalls mit zunehmenden Problemen graduell in der Qualität abnimmt (Dorer, 1999).

Reaktive Kontrolle kann mit dem Lernprozess weiter aufgebaut werden. Manche Wissenschaftler behaupten, dass mehr als 95% unserer Aktivitäten durch Lernprozess und ständige Wiederholung der Aktionen so geprägt sind, dass sie rein reaktive Handlungen geworden sind (Brooks 2002, Dörner, 2001, Murphy, 2000). Beispiel dafür sind spazieren, laufen oder Rad fahren.

4.3 Hybride Kontrollparadigma

Der Versuch reaktive Merkmale mit der aktiven Repräsentation der Aufgabe zu integrieren führt zu hybriden Systemen. Reaktive und kognitive Komponente müssen so verknüpft werden, dass sie sich gegenseitig ergänzen, um auf diese Weise deren Vorzüge miteinander zu kombinieren. Deren Integration ist jedoch nicht trivial, weil sie auf unterschiedlichen und zum Teil inkompatiblen Prinzipien beruhen. Die Frage ist, von welcher Seite man beginnen soll, um geeignete Charakteristika der hybriden Architektur zu bekommen.

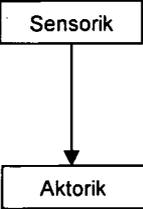
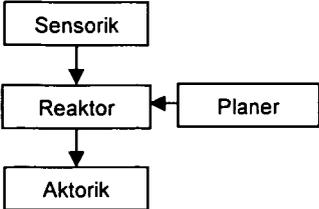
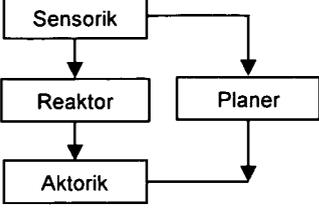
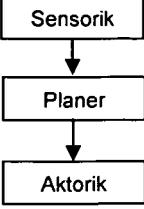
Die meisten vorhandenen Lösungsansätze fangen von der kognitiven Seite an. Es wird versucht, das klassische Planen so zu erweitern, dass es mit der Dynamik und Unsicherheit der realen Welt fertig werden kann. Diese Systeme sind meistens so aufgebaut, dass beim Auftreten eines Fehlers, d.h. der Plan ist ungültig geworden, ein neuer Planungsvorgang gestartet wird oder es wird versucht, alle potentiellen Fehler vorherzusagen, und sie vom Anfang an bei der Erstellung des Plans zu berücksichtigen (Tabelle 4.1 Spalte 4). Im Zentrum der Untersuchung befindet sich verschiedene Planer Architekturen.

Das Problem bei Planern besteht darin, beschreiben zu können, welche Eigenschaften sich ändern, und welche unverändert bleiben, wenn Aktionen ausgeführt werden. Also das Frame-Problem. Für einen Agenten, der in einer dynamischen Umgebung agiert, ist dies von großer Wichtigkeit, weil er dadurch sein Weltmodell und sein Wissen im Allgemeinen mit der realen Welt synchron halten kann. Diese theoretischen Ergebnisse geben zu erkennen, dass in jedem System, in dem Zeiteinschränkungen existieren, selbst sehr verfeinerte Techniken sich letztendlich als unbrauchbar zeigen werden. Diese Vorgehensweise scheint also die Aufgabe nur komplexer zu machen und löst die grundsätzlichen Probleme nicht.

Eine andere Möglichkeit für die Kombination von Reaktivität und Zielorientierung besteht darin, Planer und Reaktor gleichzeitig laufen zu lassen (Tabelle 4.1 Spalte 3). Dazu muss geklärt werden, wie diese beiden Komponenten zusammenarbeiten. Dieses Kohärenzproblem wird auf verschiedene Arten gelöst. Ein Vorschlag ist, je nach Situation zwischen Reaktor und Planer hin und her zu schalten (Jens und Veloso, 1998). Solange nichts Unvorhergesehenes passiert, wird das Verhalten des Agenten vom Planer gesteuert. Tritt eine unerwartete Situation in Form einer günstigen Gelegenheit oder eines Ausführungsfehlers ein, wird auf Reaktor umgeschaltet, bis der Planer einen neuen, der Situation angepassten Plan fertig gestellt hat. Die Entscheidung, wann eine solche Situation vorliegt, ist aber keinesfalls trivial. Diese Ansätze leiden aber an dem so genannten „Horizonteffekt“. Sie liefern mit der reaktiven Komponente bessere Gesamtleistung, haben aber einfach die Einschränkungen der reinen Planer-Systeme nur etwas weiter in die Zukunft verschoben.

Wenn die Umwelt dynamisch genug ist, ist es fraglich, ob der Planer überhaupt zum Einsatz kommt. Einige hybride Architekturen verwenden für die Lösung des Kohärenzproblems ein zwischen Reaktor und Planer liegendes vermittelndes Modul. Zum Beispiel in RAP heißt es Executor (Firby, 1989), in ATLANTIS (Gat et al., 1992, 1997) und T3 (Bonasso et al., 1996) Sequenzer. Der RAP Executor ist dafür verantwortlich, dass die von der Planungsebene gelieferten Pläne in kontinuierliche Handlungen umgesetzt werden.

In ATLANTIS und T3 sorgt der Sequenzer selbst für die Aktionsinitiierung. Er übergibt direkt ausführbare Aufgaben an den Reaktor und delegiert Planungsaufgaben an den Planer. Damit verwaltet er die Rechnerzeit, die der Reaktor und Planer zugeteilt wird.

BASIS STRUKTUR			
			
Rein Reaktivität	Modifizierte Reaktivität	Unterordnete Reaktivität	Rein Planen
KONTROLLPARADIGMA			
Reaktiv	Hybrid	Hybrid	Kognitiv
IDEE HINTER KONTROLLPARADIGMA			
Denkt nicht, (re)agiert	Denkt und (re)agiert gleichzeitig und unabhängig	Denkt und (re)agiert gleichzeitig und abhängig	Hart denkt, dann agiert
REPRÄSENTATION DER UMGEBUNG			
Sensor Daten	Sensor Daten	Weltmodell Sensor Daten	Weltmodell
REPRÄSENTATION DER AUFGABE			
Keine Aufgabe Suchend	Plan Aufgabe Explizit Definiert	Plan Aufgabe Implizit Definiert	Plan Aufgabe Implizit Definiert
AUSFÜHRUNG DER AKTION			
Schnell	⇨		Langsam
STABILITÄT DES SYSTEMS (Antwort auf Störung)			
Robust	⇨		Schwach
BEISPIEL DES AGENTEN (Konstrukteur, Jahrgang)			
Elsie und Elmer (Walter, 1945)	Architektur	RAP (Firby, 1989)	Shakey (SRI, 1970)
Genghis, Kismet (Brooks, seit 1986)	Verwendet im Montagsystem	ATLANTIS (Gat, 1992)	SOAR (Rosenbloom, 1993)
Harv (Arkin, 1989)		T3 (Bonasso et al., 1996)	

Tab. 4.1. Kontrollparadigmen und ihre Hauptmerkmale.

Allen drei Architekturen ist gemeinsam, dass sie keine motivationalen Aspekte beinhalten. Der Agent hat weder die Möglichkeit, die unterschiedliche Wichtigkeit von Zielen zu berücksichtigen, noch Information darüber, welche von verschiedenen Alternativen zur Erreichung eines Ziels am meisten erfolgversprechend sind.

Eine weitere Untergruppe der Hybriden besteht in der Erweiterung eines reaktiven Systems um Zielorientierung (Tabelle 4.1, Spalte 2). Zielorientierung setzt eine intelligente Verhaltensauswahlstrategie voraus. Im diesem Fall wird der Agent von Reaktor geleitet. Der Planer kommt nur dann zum Einsatz wenn Reaktor zu viel von dem Ziel abweicht. Ebenfalls der Grenzwert, den der Planer einschaltet, muss für jede Situation definiert sein.

Der Vorteil von hybriden Architekturen ist, dass der modulare Aufbau den Entwurf und die Erstellung des Agenten vereinfacht. Die Module können Idealerweise getrennt voneinander erstellt und zum Teil auch getestet werden.

4.4 Richtige Kontrolle für Agenten in der realen Umgebung

Anzahl existierender Agentenarchitekturen ist groß und rasch steigend. Jede Agentenarchitektur ist unter gewissen Annahmen konstruiert. Nur wenn die gegebenen Voraussetzungen getroffen sind, kann der Agent tatsächlich funktionieren. Aufgrund dessen konnte man solche Architekturen auswählen, die dem eigenen Bedarf entsprechen. Aspekte die Agentenarchitektur beeinflussen, sind eben explizit gegeben, und dann vorwiegend in der ad-hoc Art. Klare systematische Verbindung zwischen Agentenarchitektur und auf der Konstruktion beeinflussende Merkmale existiert nicht.

Bevor es weiter geht, soll die Antwort auf eine fundamentale Frage gefunden werden, nämlich wie soll die Kontrolle des Agenten ausschauen, die zur für die industrielle Anwendung brauchbare Agentenarchitektur führen?

Entscheidender Einfluss auf die Leistung des Agenten sowie dessen erforderlicher Komplexität des Kontrollteiles, hat der Art und Weise auf welche verwendete Kontrollparadigma die Umgebung repräsentiert.

Aus der Sicht der Kontrolle kann die Umgebung durch drei Merkmale beschreiben werden. Das sind:

- *Voraussagbarkeit* - Eine Umgebung ist deterministisch, wenn sich aus dem aktuellen Zustand des Agenten der zukünftige Zustand des Agenten berechnen lässt. Insbesondere bedeutet das, dass alle Aktionen eines Agenten in einer deterministischen Umgebung immer die erwarteten Effekte nach sich ziehen. Dadurch sind solche Umgebungen einfacher als nichtdeterministische, in denen ein Agent lediglich statistische Aussagen über das Eintreffen von Effekten machen kann.

- *Komplexität der Umgebung* – Hier geht es in erster Linie um physische Anforderungen der Umgebung. Also um die Struktur der Bewegungsebene, Anzahl und räumliche Distribution der Ressourcen sowie Hindernisse.
- *Dichtheit der Entscheidungspunkte* - Es gibt Teile der Umgebung, die mehrere für die Ausführung der Aufgabe wichtige Markmale haben. Zum Beispiel kann einen Agent auf den Punkt kommen, ab welchem er in mehrere Richtungen weiter gehen kann, die aus der Sicht der Ausführung der Aufgabe gleichwertig sind. Diesbezüglich soll er die Entscheidung autonom treffen. Dichtheit der Entscheidungspunkte ist spezielle Form der Komplexität der Umgebung. Weil sie stark die Geschwindigkeit des Agenten in der realen Umgebung beeinflusst, ist sie als getrennte Punkt zu behandeln.

In dynamischen Umgebungen erweist sich die Planungsweise klassischer künstlicher Intelligenz oft als nicht ausreichend oder sinnvoll. Durch die raschen und unvorhergesehenen Änderungen der Situation verlieren Pläne schnell ihre Gültigkeit. Je weniger voraussagbar eine Umgebung ist, desto weniger lohnt sich der Aufwand, langfristig zu planen. Aus diesem Grund sind kognitive Agentenarchitekturen, die nach klassischer künstlicher Intelligenz aufgebaut sind, für Umgebungen geeignet, die höheren Voraussagbarkeitsgrad haben, in der Dichtheit der Entscheidungspunkte niedrig ist, sodass das Weltmodell leicht aufgebaut werden kann.

Andererseits funktioniert der reaktive Ansatz besonders gut in Umgebungen, die teilweise unbekannt sind und in denen die Sensordaten verrauscht und unverlässlich sind und in denen das Operieren in Realzeit erfolgen soll und unerwartete Ereignisse berücksichtigt werden müssen. Da kommt Schnelligkeit und Robustheit der reaktiven Kontrolle im vollen Gang. Höhere Dichtheit der Entscheidungspunkte ist da auch unerwünscht. Dies kann im extremen Fall den Agenten zum Stillstand bringen.

Typische industrielle Umgebung (auf Abbildung 4.2 mit 1 bezeichnet) hat mittleren Komplexitätsgrad, Voraussagen sind relativ zuverlässig und Dichtheit der Entscheidungspunkte niedrig. Die Ideale Agent Architektur soll den Vorteil von Voraussagbarkeit nutzen und gleichzeitig soll sie die Möglichkeit haben schnelle Veränderungen des aktiven Planes durchzuführen, wenn der Plan nicht mehr dem Umgebungszustand entspricht. Da der Agent die Möglichkeit hat die Pläne dynamisch zu aktualisieren ist rechtfertigen. In der Tabelle 4.1 sind dafür geeignete Lösungen in zwei Linien geteilt; Modifizierte Reaktivität und Untergeordnete Reaktivität.

Diese Analysen zeigen, dass solche Agentenarchitektur erwünscht ist, bei welcher die Repräsentation der Aufgabe am wenigsten Merkmale der reaktiven Kontrolle behindert. Auf

diesem Aspekt ist bei der Entwicklung von Aufgaberepräsentation der Agenten für den Einsatz im selbstorganisierenden Montagesystem Rücksicht zu nehmen. Damit wird gewährleistet, dass eine rein reaktive Kontrolle ausreichend ist.

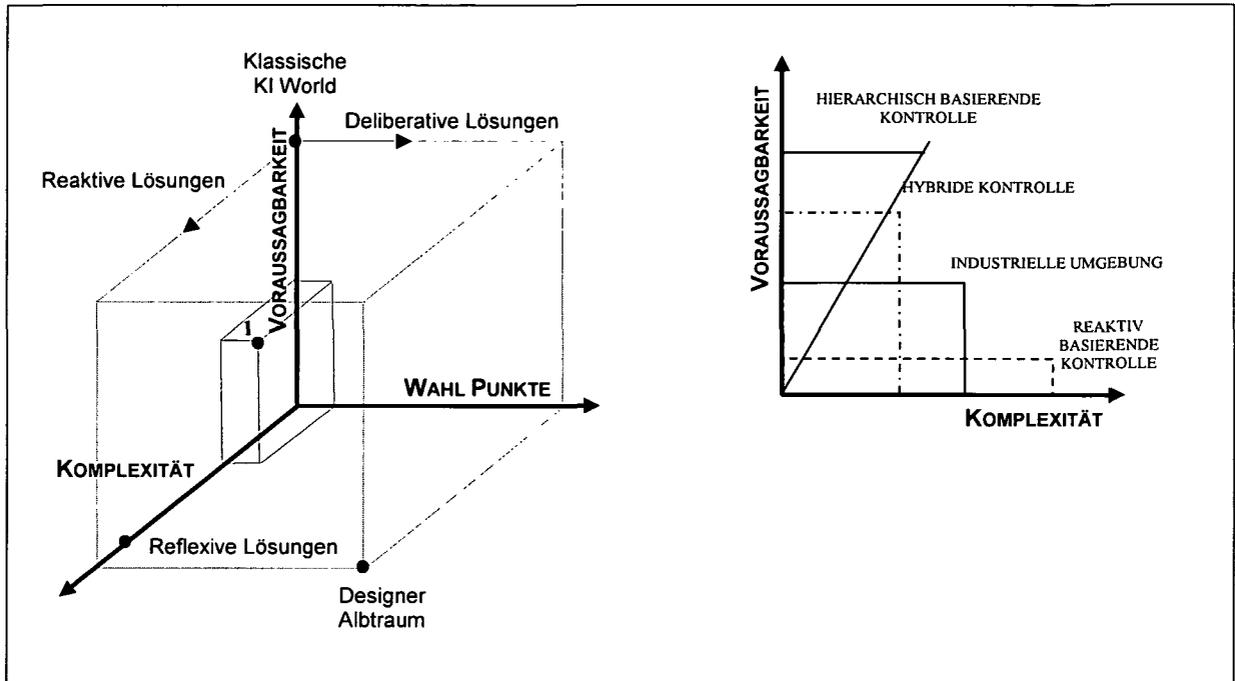


Abb. 4.2. Wegweiser für Agentenkontrollparadigma Auswahl.

Bemerkung: Oben eingeführte Pro und Kontra kognitiver Architekturen können auf voreilige Schlussfolgerung führen, dass kognitive Architekturen nicht geeignet sind für den Einsatz in der realen Umgebung. Wir selbst sind so betrachtet kognitive Agenten. Also stimmt obige Aussage nach dem heutigen Stand des Wissens und der Technologie, die noch leider nicht ausreichend reif sind um effiziente kognitive Agenten für den Einsatz zu bauen. Dagegen sind reaktiv basierende Agenten realisierbar und liefern Resultate, die von heutigen kognitiv basierenden Agenten nicht zu übertreffen sind. Gut dabei ist dass reaktiv basierende Architekturen den Voraussetzungen genügen, die aus in der Arbeit definierter Anwendung, resultieren.

Reaktive Architekturen

Die Aktion reaktiver Agenten ist unmittelbare Antwort auf Umgebungsgegebenheiten. Die Aktionsauswahl besteht darin, eine vorkodierte Reaktion auszuwählen, die in der jeweiligen Situation angemessen ist. Zwei Aspekte sind von Schlüsselwichtigkeit für den Erfolg reaktiven Ansatzes. Das sind (1) der Satz des Grundverhaltens und (2) die gegenseitigen Interaktionen.

Das Verhalten des Agenten entsteht aus der Verkettung der Aktionen und deren Interaktion mit der Umgebung. Jede Aktion dauert und gilt für den Bruchteil der Gesamtzeit. Das bedeutet, dass das Steuerungssystem die Aktionswahl treffen und diese dann auch so koordinieren muss, dass ein kohärentes Verhalten des Agenten entsteht.

Maes formuliert das Problem der Koordination als Aktion Selektion Problem; „*How can an agent select „the most appropriate“ or „the most relevant“ next action to take at a particular moment, when facing a particular situation?*“ (Maes, 1989). Nach der Art und Weise, wie diese Koordination generiert wird, zerfallen reaktive Architekturen in zwei Gruppen; Kompetitive und Kooperative.

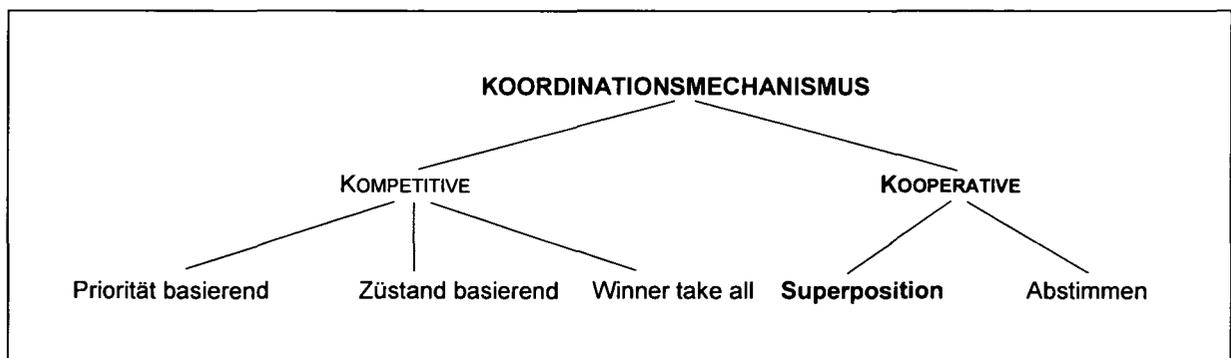


Abb. 5.1. Arten von Koordinationsmechanismen. Fett gedruckt bezeichnet Typ des Koordinationsmechanismus, der angewendet wird.

5.1 Kompetitive Ansätze

Kompetitive Ansätze bilden eine Gruppe von Architekturen, in der zu jedem Zeitpunkt immer nur die Vorgaben eines einzelnen Moduls an die Aktoren des Agenten weiter geleitet werden. Die Module konkurrieren um den Zugriff auf die Aktoren, so dass diese Ansätze mit „kompetitiv“ beschrieben werden können. Gleichzeitig kann nur die Ausgabe eines Moduls bis zu den Aktoren vordringen.

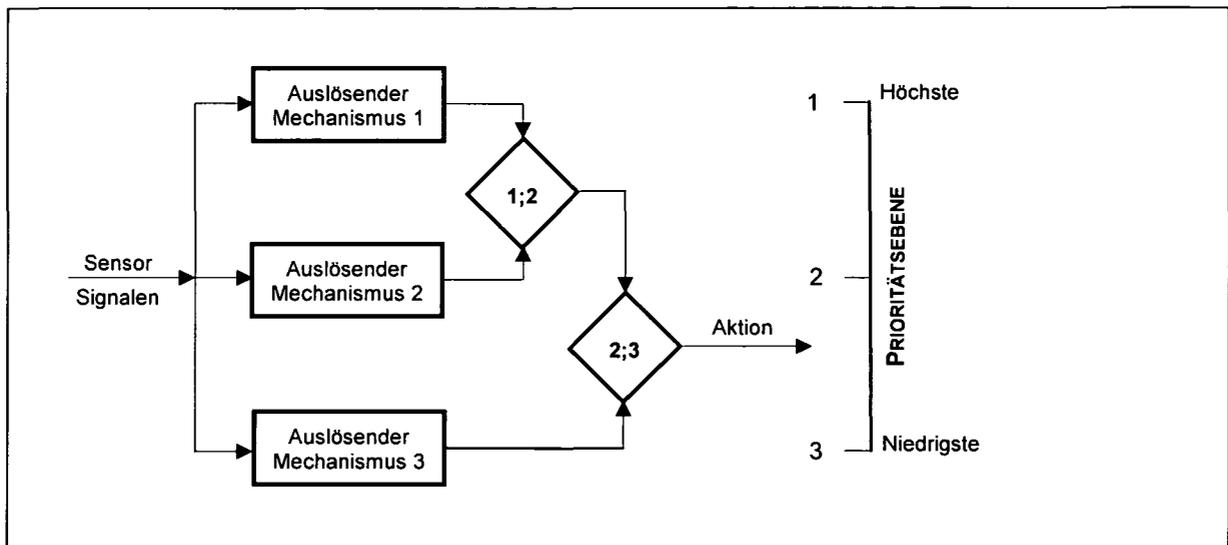


Abb. 5.2. Subsumption Architektur. Priorität basierend Aktion Selektionssystem.

Die bekannteste kompetitive reaktive Architektur ist die von Brooks entwickelte Subsumption Architektur (Brooks, 1986, 1989, 1991). Subsumption Architektur ist durch eine Reihe, parallel verbundener auslösender Mechanismen (im Original: Behaviors, siehe Kapitel 2) charakterisiert. Es wird vorausgesetzt, dass diese Reihe für die Erledigung einer Klasse von Aufgaben ausreicht. Jeder auslösender Mechanismus ist für eine Teilaufgabe zuständig zum Beispiel Hindernisse ausweichen, Ziel Ansteuerung, fortfahren, usw. Aufgrund der Kombination von aktivierten Sensoren und deren Signalamplituden wird ein auslösender Mechanismus aktiviert. Die Signalamplituden werden nach der Shannon Quantisierungsmethode (Sampling) gewonnen.

Um die Priorität der Teilaufgaben zu bewahren, sind diese in Prioritätsebenen gegliedert. Das resultierende Steuerungssystem ist fest verdrahtet und der Agent ist nicht programmierbar (für verschiedene Aufgabenstellungen). Seine Fähigkeiten beschränken sich auf die Aktionsmuster, die der Designer implementiert hat. Ein solches Prioritätssystem ist auf der Abbildung 5.2 dargestellt. Arbeitsszenario der Subsumption Architektur wird durch folgenden deskriptiven Algorithmus beschrieben:

Start

```

{
  wenn {auslösender Mechanismus 1 ist aktiv}           /Aktion 1 wird ausgeführt/
  dann {alle andere unterdrücken}
  sonst {auslösender Mechanismus 2 ist aktiv}         /Aktion 2 wird ausgeführt/
    wenn {auslösender Mechanismus 2 ist aktiv}
    dann {unterdrückt alle auf niedrigeren Kompetenzebenen}
    sonst {auslösender Mechanismus 3 ist aktiv}       /Aktion 3 wird ausgeführt/
  .
  .
  .
}
```

Ende

In Subsumption Architektur sind die Module einander fix untergeordnet. In Maes Action Selection Network kommt der Wettbewerb der einzelnen Module zum Ausdruck (Maes, 1989, 1991). Hier wird in einem „Winner takes all“ Verfahren das Modul, welches Zugriff auf die Aktoren des Agenten erlangt, bestimmt. In Koseckas und Bajcsys Diskret Event Systems wird aufgrund des aktuellen Zustandes des Agenten entschieden, welches Modul Kontrolle über den Agenten erhält (Koseckas, 1993).

Der Vorteil des kompetitiven Ansatzes, nur einem Modul Zugriff auf die Aktoren zu gewähren, liegt in der Konzentration der Kontrolle über den Agenten in einem einzigen auslösenden Mechanismus in einem gewissen Zeitraum. Die Suche nach Fehlern kann sich auf den Mechanismus beschränken, welcher zu dem Zeitpunkt, zu welchem der Fehler auftritt, aktiv ist.

Nachteile ergeben sich in sehr dynamischen Umgebungen oder bei schnellen Bewegungen des Agenten. In beiden Fällen ändert sich die Situation, in welcher sich der Agent befindet, schnell, so dass häufig zwischen den der jeweiligen Situation angepassten auslösenden Mechanismen umgeschaltet werden muss. Dadurch entstehen Diskontinuitäten im Fluss der an die Aktoren des Agenten gesendeten Steuerungsgrößen. Dadurch kann es zu ruckartigen unnatürlichen Bewegungsabläufen kommen.

Ein weiterer Nachteil ist, dass die Aktionsvielfalt auf vorprogrammierte Fälle beschränkt ist. Im oben gegebenen Beispiel handelt es sich um nur drei mögliche Reaktionen. In den meisten Situationen ist der Entscheidungsraum recht groß. Das heißt es kann leicht passieren dass situationsgemäße Aktion nicht vorhanden ist. Diesen Nachteil will man mit der Entwicklung mehrerer auslösender Mechanismen vermeiden. Dieser Vorgang ist eher limitiert, weil er zur kombinatorischen Explosion führt.

Ein Beispiel ist Genghis von Brooks Arbeitsgruppe entwickelter mobiler Roboter, der sechsbeiniges Laufen in sehr unübersichtlichen Situationen ermöglicht. Genghis kann sich,

aufgrund 51 auslösender Mechanismen, an wechselnde Umweltbedingungen anpassen. Er kann über verschiedene geformte Hindernisse klettern, kann trotz der Amputation eines Beines weiterlaufen und kann sich nach einem Sturz von selbst wieder aufrichten.

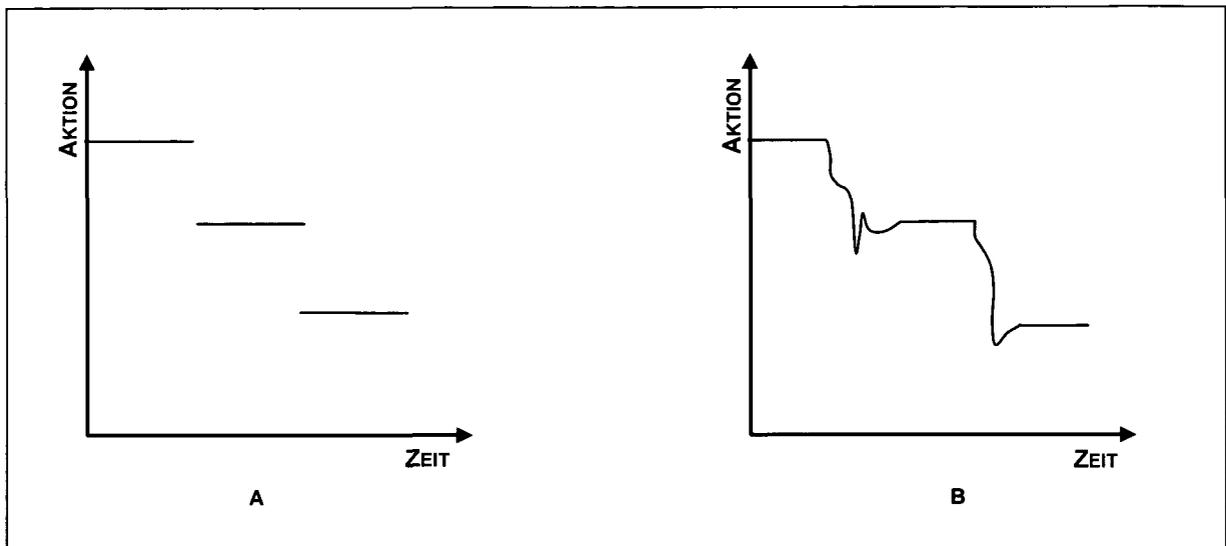


Abb. 5.3. Aktionschema des Agenten mit drei kompetitive Kompetenzebenen. In jedem Moment kann nur ein Kompetenzebene aktiv sein; (A) Theoretisch, (B) Sprünge durch diskrete Übergänge.

5.2 Kooperative Ansätze

Den durch das Umschalten zwischen einzelnen Modulen entstehenden Problemen begegnen Ansätze, in welchen die Steuerungsgrößen durch eine Überlagerung der Ausgaben aller Module entstehen. Da sich die Bewegung des Agenten aus dem Zusammenspiel der einzelnen Module innerhalb des Steuerungsprogramms ergibt, werden solche Ansätze auch als „kooperativ“ bezeichnet.

Beispiele für eine parallele kooperative Architektur sind Arkins Motor Schemas (Arkin, 1992,1998), Konoliges und Myers Saphira Architektur (Konolige, 1996, 1997), Steinhage und Schöner Dynamical Systems (Steinhage, 1997) und Rosenblatts Distributed Architecture for Mobile robot Navigation (Rosenblatt, 1997).

Das Hauptproblem kooperativer Ansätze entsteht durch die unterschiedlichen Ziele der separaten Module. Der gleichzeitige Zugriff auf die Aktoren des Agenten kann in bestimmten Fällen dazu führen, dass ein unerwünschtes Gesamt-Verhalten des Agenten entsteht. Diese Situationen müssen von dem Designer erkannt und durch wechselseitige Abhängigkeiten der Module gelöst werden. Ebenfalls die Überlagerung der Vorgaben mehrerer Module durch den Überlagerungsmechanismus kann in den erfahrungsgemäßen Fällen sinnvolle

Steuerungsgrößen hervorbringen, die zum stabilen Verhalten des Agenten führen. Die Konflikte zwischen den Vorgaben werden auf einer von der Aktion abstrahierten Ebene gelöst (Umgebungs-Agent Interaktion). Module können daher zu einem gewissen Grad vollständig autonom agieren und dennoch mit anderen Modulen allein aufgrund der Art der erzeugten Ausgaben kooperieren. Die Möglichkeit Ausgaben vieler Module zu überlagern, ergibt sich in allen vier Ansätzen jedoch aus der Festlegung einer einzigen Schnittstelle zwischen den Modulen und dem Überlagerungsmechanismus. Der Überlagerungsmechanismus erzeugt in allen genannten kooperativen Architekturen einen Steuerungswert in Form einer Orientierung zur Steuerung des rotatorischen Freiheitsgrades des Agenten. Der translatorische Freiheitsgrad wird nur von Saphira und DAMN durch die Erzeugung eines Steuerungswert in Form einer Geschwindigkeit gesteuert. Neben dem Missetand, dass Motor Schemas und Dynamical Systems dem Entwickler keine Kontrolle des translatorischen Freiheitsgrades ermöglichen, fällt auf, dass die Kontrolle über den rotatorischen Freiheitsgrad nur über die Vorgabe einer Orientierung möglich ist. Damit ist es in allen vier Ansätzen nur auf Umwegen und bei genauer Kenntnis des Reglers der Aktoren möglich, die Winkelgeschwindigkeit des Agenten zu kontrollieren.

Dieser Umstand macht es insbesondere schwierig, existierende, auf die Steuerung der Geschwindigkeit und der Winkelgeschwindigkeit angewiesene Algorithmen, in parallelen kooperativen Architekturen zu verwenden.

Überlagerungsmechanismus der kooperativen Ansätze wurzelt auf der Idee von Potentialfeldern.

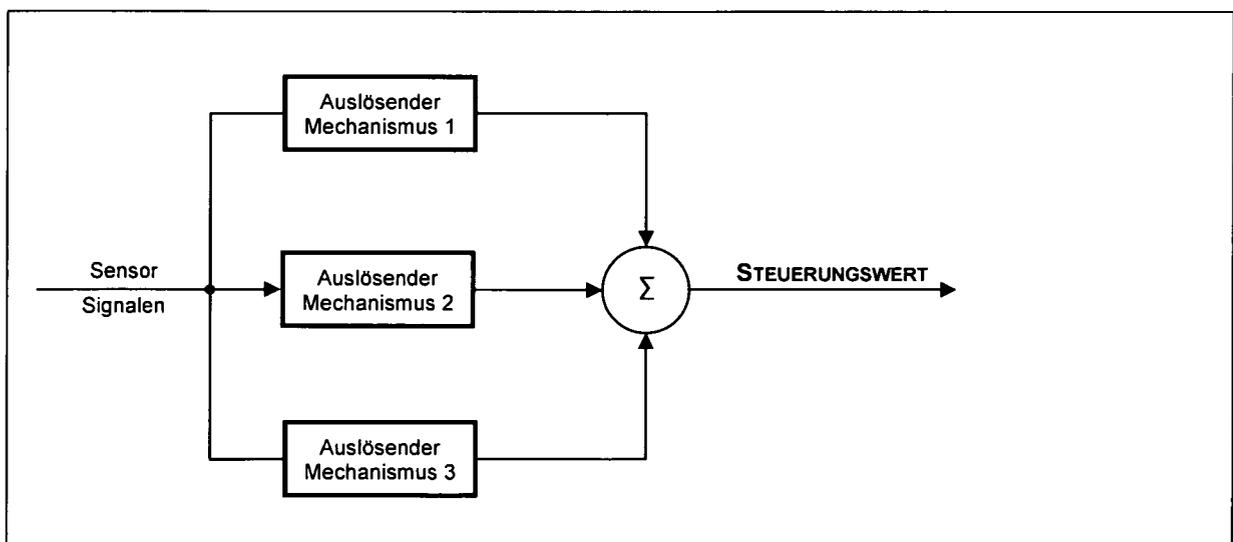


Abb. 5.4. Kooperative reaktive Steuerung. Steuerungswerte, die alle Algorithmen erzeugen, überlagern sich im Überlagerungsmechanismus Σ und werden als eine Resultante zum Aktoren übergeben.

5.3 Die Felder

Felder sind nichtmaterielle Einflusszonen physikalischer Größen (Sheldrake, 1998). Das Gravitationsfeld der Erde umgibt uns überall. Man kann es nicht sehen, dass es real ist, erkennt man jedoch durch dessen Einfluss.

Felder sind das Medium von „Fernwirkungen“. Über Felder können Dinge aufeinander einwirken, ohne in direkten materiellen Kontakt miteinander zu stehen. Wir sehen die Dinge um uns herum, weil wir mit ihnen durch das elektromagnetische Feld verbunden sind, in dem die Schwingungsenergie des Lichts sich ausbreitet. Die Existenz elektromagnetischer Felder hat die Entwicklung der Sensoren allen Lebewesen geformt. Stets und überall sind wir von zahllosen Schwingungsmustern dieses Feldes umgeben, die wir nicht unmittelbar mit den Sinnen wahrnehmen können, aber doch mit Hilfe entsprechender Geräte, so etwa einem Radio oder Fernsehapparat.

Also es gibt einerseits Schwingungen des elektromagnetischen Felds und andererseits Instrumente (Sensoren), die diese Schwingungen registrieren können. Diese Schwingungen stehen in der direkten Verbindung mit der momentanen Situation in der Umgebung. Das ist der erste Gedanke bei der Nützung der Felder in der Entwicklung von autonomen Agenten.

Aus der Physik der Felder stammt eine nützliche Eigenschaft. Ein Stein in Gravitationsfeld, oder ein Teilchen im Elektromagnetischen Feld wird immer versuchen den Punkt mit dem lokalen minimalen Potential zu erreichen. Befindet sich das Objekt in diesem Punkt bleibt er dort bis das Potentialfeld sich verändert oder bis man zusätzliche Energie in das Objekt reinschiebt. Wenn das Ziel des Agenten sich genau in dem Punkt mit dem lokalen minimalen Potential befindet den der Agent kann erreichen ohne es zu „wissen“. Dass heißt, das Feld, das die Bewegung des Agenten beeinflussen kann, kann ihm allein durch den Raum führen. Das ist der zweite Gedanke bei der Nützung der Felder in der Entwicklung von autonomen Agenten.

Das elektromagnetische Feld kann den Agenten durch den Raum führen. Diese Führungsmöglichkeit ist durch das Gravitationsfeld limitiert. Folge dessen ist dass, das erste Feld Sensorik und zweites Mechanik des Agenten beeinflusst. Diese Annahme ist aber nicht ganz korrekt. Das elektromagnetische- und Gravitationsfeld wirken nicht seriell sondern parallel. Aus der Sicht der Entwicklung der biologischen Systeme bedeutet es dass die Wirkungen der Felder in den entstandenen Lösungen in einander verschachtelt sind. Aber wie, kann man es leider noch nicht begreifen. Deswegen betrachtet man isolierte Prozesse, die allen auch seriell beschrieben werden können. Und deswegen sind solche Approximationen zurzeit der einzig mögliche Weg. Man muss sich darüber im Klaren sein, dass isoliert betrachtet solche Modelle ziemlich gut sind, dass sie auf der Ebene des Agenten leicht in Irre führen kann. Oftmals deckt man angenehme Überraschungen solcher Art mit dem Begriff Emergenz.

Der dritte Gedanke bei der Nützung der Felder in der Entwicklung von autonomen Agenten sollte sein, dass die Felder Information über Beziehung zwischen Form und Funktion des Agents beinhalten.

5.3.1 Von Skalar- bis zum Vektorfeld

Sensoren tasten Energieschwankungen innerhalb deren Breitweite ab. Damit entstehen Signale, deren Wert nur von der Größe der abgetasteten Schwankungen abhängt. Die Umgebung des Agenten ist also in Form von Skalaren kodiert. Aufgrund Skalargroßen lässt sich ein Skalarfeld darstellen.

Der Pfad des Agenten lässt sich dagegen nur im Vektorfeld darstellen. Dass heißt, dass das Skalar- im Vektorfeld transformiert werden muss. Die Transformation muss passend und fein abgestimmt werden weil sie entscheidend auf das Verhalten des Agents wirkt.

Betrachtet man das Potentialfeld, schreitet man nun im Raum in verschiedenen Richtungen immer um den gleichen Betrag $\Delta r =$ konstant fort, so zeigt sich, dass es eine Abhängigkeit der Potential Änderung von der Richtung des Fortschreitens gibt. Beim Fortschreiten entlang einer Linie gleichen Potentials (in Temperaturfeld heißt sie Isotherme) ändert sich das Potential nicht. Beim Fortschreiten senkrecht zur Linien gleichen Potentials ändert sich das Potential am stärksten. Für die Potential Änderung im skalaren Potentialfeld muss es also eine Abhängigkeit von der Richtung, also eine Verknüpfung mit dem Vektor \vec{r} geben:

$$\begin{aligned}\Delta P &\approx \Delta \vec{r} \\ \Delta \vec{r} &= \Delta x \vec{i} + \Delta y \vec{j} + \Delta z \vec{k}\end{aligned}\tag{5.1}$$

Eine skalare Größe wie ΔP lässt sich aber aus der vektoriellen Größe $\Delta \vec{r}$ nur als Ergebnis eines Skalarprodukts erhalten. Für die Produktbildung benötigt man einen zweiten Vektor. Deswegen definiert man einen Hilfsvektor \vec{a} . Da es entlang der Linien konstanter Potential keine Änderung gibt, senkrecht dazu aber eine maximale Potentialänderung auftreten soll, macht das Skalarprodukt nur dann Sinn, wenn \vec{a} ein Vektor ist, der senkrecht zur Linien konstanter Potential steht. Zur Ermittlung des Betrags benutzt man die Komponentendarstellung:

$$\begin{aligned}\Delta \vec{r} &= \Delta x \vec{i} + \Delta y \vec{j} + \Delta z \vec{k} \\ \vec{a} &= a_x \vec{i} + a_y \vec{j} + a_z \vec{k} \\ \Delta P &= \vec{a} \cdot \Delta \vec{r} = a_x \Delta x + a_y \Delta y + a_z \Delta z\end{aligned}\tag{5.2}$$

mittels Taylorsche Reihe der Funktion $P(x,y,z)$ lässt ΔP darstellen:

$$\Delta P = \frac{\partial P}{\partial x} \Delta x + \frac{\partial P}{\partial y} \Delta y + \frac{\partial P}{\partial z} \Delta z \quad (5.3)$$

Aus Komponentenvergleich ergibt sich:

$$a_x = \frac{\partial P}{\partial x} \quad a_y = \frac{\partial P}{\partial y} \quad a_z = \frac{\partial P}{\partial z} \quad (5.4)$$

Und damit Hilfsvektor:

$$\vec{a} = \frac{\partial P}{\partial x} \vec{i} + \frac{\partial P}{\partial y} \vec{j} + \frac{\partial P}{\partial z} \vec{k} \quad (5.5)$$

Damit ist der Hilfsvektor bezüglich Richtung und Betrag eindeutig festgelegt. Man nennt diesen Hilfsvektor, der durch räumliche Differentiation aus der skalaren Größe Potential hervorgeht, den Gradientenvektor des Skalarfeldes. Der Gradient ist also der in der Richtung $d\vec{r}$ genommene Differentialquotient vom Skalar P.

$$\text{grad}P = \frac{\partial P}{\partial x} \vec{i} + \frac{\partial P}{\partial y} \vec{j} + \frac{\partial P}{\partial z} \vec{k} \quad (5.6)$$

Der Gradient gibt die Richtung des größten Potentialgefälles an und hat den Betrag:

$$|\text{grad}P| = \sqrt{\left(\frac{\partial P}{\partial x}\right)^2 + \left(\frac{\partial P}{\partial y}\right)^2 + \left(\frac{\partial P}{\partial z}\right)^2} \quad (5.7)$$

Das größte Potentialgefälle ist in der Richtung des Punktes mit dem lokalen minimalen Potential. Der Punkt benimmt sich als Attraktor. Also wenn sich das Ziel des Agenten in dem Punkt befindet, dann kann man mittels der Berechnung des Gradienten den Pfad des Agenten feststellen. Ebenfalls ist es offensichtlich dass ernsthafter Fehler bei der Pfadberechnung durch die Entstehung falscher Attraktoren auftreten kann. Dieses Problem lässt sich mit einer eindeutig markierter Position lösen.

Die auf Potentialfeldern basierende Pfadberechnung ist ein biologisch und mathematisch motiviertes Verfahren, um eine reaktive situationsgebundene Steuerung zu erzeugen.

5.3.2 Attraktoren

Attraktoren sind ein zentrales Phänomen in Potentialfeld basierenden Methoden. Intuitiv ist ein Attraktor ein Gebiet im Phasenraum, das benachbarte Trajektorien "anzieht", indem sie sich diesem Gebiet asymptotisch nähern. Der einfachste Fall sind punktförmige Attraktoren,

so genannte Fixpunktattraktoren. Abbildung 5.5 A zeigt ein Phasenportrait mit zwei Fixpunktattraktoren. Trajektorien, die im "Anziehungsbereich" des Fixpunktattraktors liegen, entwickeln sich auf diesen zu und nähern sich ihm asymptotisch beliebig nahe an. Wenn das System zufällig gerade in einem solchen Punkt gestartet wird, verharrt es einfach darin. Der Fixpunktattraktor selbst ist eine punktförmige Trajektorie.

Fixpunkt-Attraktoren sind in empirischen Systemen sehr häufig, sie entsprechen intuitiv gesehen Ruhezuständen.

Das klassische Beispiel ist ein durch Reibung gedämpftes Pendel. Ruhezustände sind stabil. Das bedeutet, dass kleine Störungen des Systems nach kurzer Zeit wieder ausgeglichen werden, indem die Trajektorie zum Fixpunktattraktor zurückkehrt.

Bei andauernden kleinen Störungen bedeutet die Stabilität, dass sich das System mit seinen Zuständen in einer kleinen Umgebung des Ruhezustandes aufhält. In Abbildung 5.5 B findet man einen zyklischen Attraktor. Der Attraktor selbst ist eine geschlossene, zyklische Trajektorie. Man sieht, dass Trajektorien aus der Umgebung des Attraktors sich bei $t \rightarrow \infty$ beliebig nahe spiralförmig diesem nähern. Zyklische Attraktoren sind, intuitiv gesehen, "stabile Schwingungen". Das klassische Beispiel liefert das angeregte Pendel, z.B. das Pendel einer Uhr. Auch hier heißt Stabilität wieder, dass bei kleinen Störungen der Attraktor von der Systemtrajektorie asymptotisch wieder eingenommen wird. Zyklische Attraktoren sind ein sehr häufiges Phänomen in dynamischen Systemen. Man muss sich beim Erfinden von Systemgleichungen nicht besonders anstrengen, um einzelne Gebiete im Phasenraum mit stabilen Schwingungen auszustatten. Diese Ubiquität von stabilen periodischen Aktivitätsmustern kann in ihrer Bedeutung für die Welt, wie wir sie kennen (und die uns hervorgebracht hat), nicht überschätzt werden.

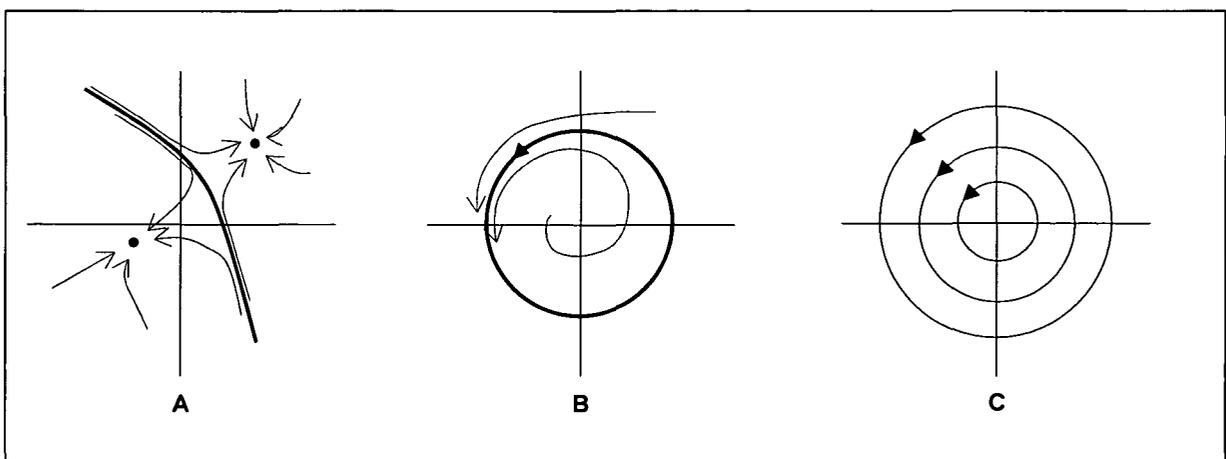


Abb. 5.5. Die beiden einfachsten Attraktor-Typen. (A) Fixpunkt-Attraktor (zwei Stück; die schattierte Line markiert die Grenze zwischen den beiden zugehörigen Attraktionsbecken), (B) zyklischer Attraktor, (C) kein Attraktor in diesem Phasenportrait.

Gäbe es nämlich gar keine Attraktoren, so gäbe es, grob gesprochen, überhaupt keine stabilen Phänomene in der physikalischen Realität. Gäbe es jedoch nur Fixpunktattraktoren, so würde die Realität in einem gigantischen, finalen Ruhezustand ersterben. Erst zyklische Attraktoren eröffnen die Möglichkeit, dass einerseits sich etwas tut, andererseits aber stabile, wieder erkennbare, klassifizierbare Phänomene in die Welt kommen. Es gibt mit zyklischen Attraktor verwandte Erscheinungen, die etwas komplizierter aussehen als dieser. Sie entstehen, wenn sich mehrere solcher Attraktoren überlagern. Zum Kontrast zeigt Abbildung 5.5 C ein Phasenportrait, in dem die Trajektorien parallel nebeneinander herlaufen. Nichts nähert sich irgendetwas anderem. Hier gibt es keinen Attraktor.

Was für eines System der Attraktor darstellt, hängt von zwei Aspekten: Typ und Sensibilität der Sensoren. In einführenden Beispiel sind Attraktoren chemische Spüren. In Fall von Transportagenten sind Attraktoren Zielposition und Hindernisse. Ebenfalls sind die Attraktoren, die Hindernisse repräsentieren, durch eine minus Vorgabe im Algorithmus in den Repulsoren umgewandelt (Sehe Gleichung 5.8).

5.3.3 Magnitude Profil

Ein wichtiger Aspekt der Attraktoren ist dessen Magnitude Profil. Magnitude Profil ist eine graphische Repräsentation von Einflussfunktion innerhalb Einflusszone eines Attraktors oder Repulsors. Magnitude Profil hat entscheidender Einfluss auf das Verhalten des Agenten in der Einflusszone.

Auf der Abbildung 5.6 ist einen 2D Potentialfeld mit dem linearen Magnitude Profil gezeigt. Die Längen der Pfeile innerhalb der Einflusszone des Repulsors D sind konstant. In Polarkoordinatensystem mit dem Ausgang im Zentrum des Hindernis kann man das Potentialfeld Mathematisch repräsentieren als:

$$\begin{aligned}
 v_{H \text{ direction}} &= -\Phi \\
 v_{H \text{ magnitude}} &= \begin{cases} k & \text{für } d \leq D \\ 0 & \text{für } d > D \end{cases} \quad (5.8)
 \end{aligned}$$

Der Agent bewegt sich mit der konstanten Geschwindigkeit \bar{v}_0 gerade aus. Im Moment t_1 nimmt der Agent das Hindernis wahr und biegt in die Richtung $\Phi=180^\circ$ ab. Eben die Geschwindigkeit vom Abweichungsmanöver ist konstant und hängt nicht von der momentanen Distanz des Agenten vom Hindernis soweit der Agent innerhalb der Einflusszone des Repulsors ist.

Wenn der Agent sich außerhalb der Einflusszone befindet, nimmt er wieder Richtung und Geschwindigkeit von v_0 und gerät wieder ins repulsiven Feld. Theoretisch befindet sich der Agent wieder in demselben Punkt wie beim ersten Eintritt im repulsiven Feld. In der Praxis führt es aber zur ruckartigen Bewegung am Rande der Einflusszone. Das lineare repulsive Feld benimmt sich binär. Es hat zwei Zustände, 0 und 1. Das Problem des linearen Profils liegt im scharfen Übergang auf dem Rand der Einflusszone.

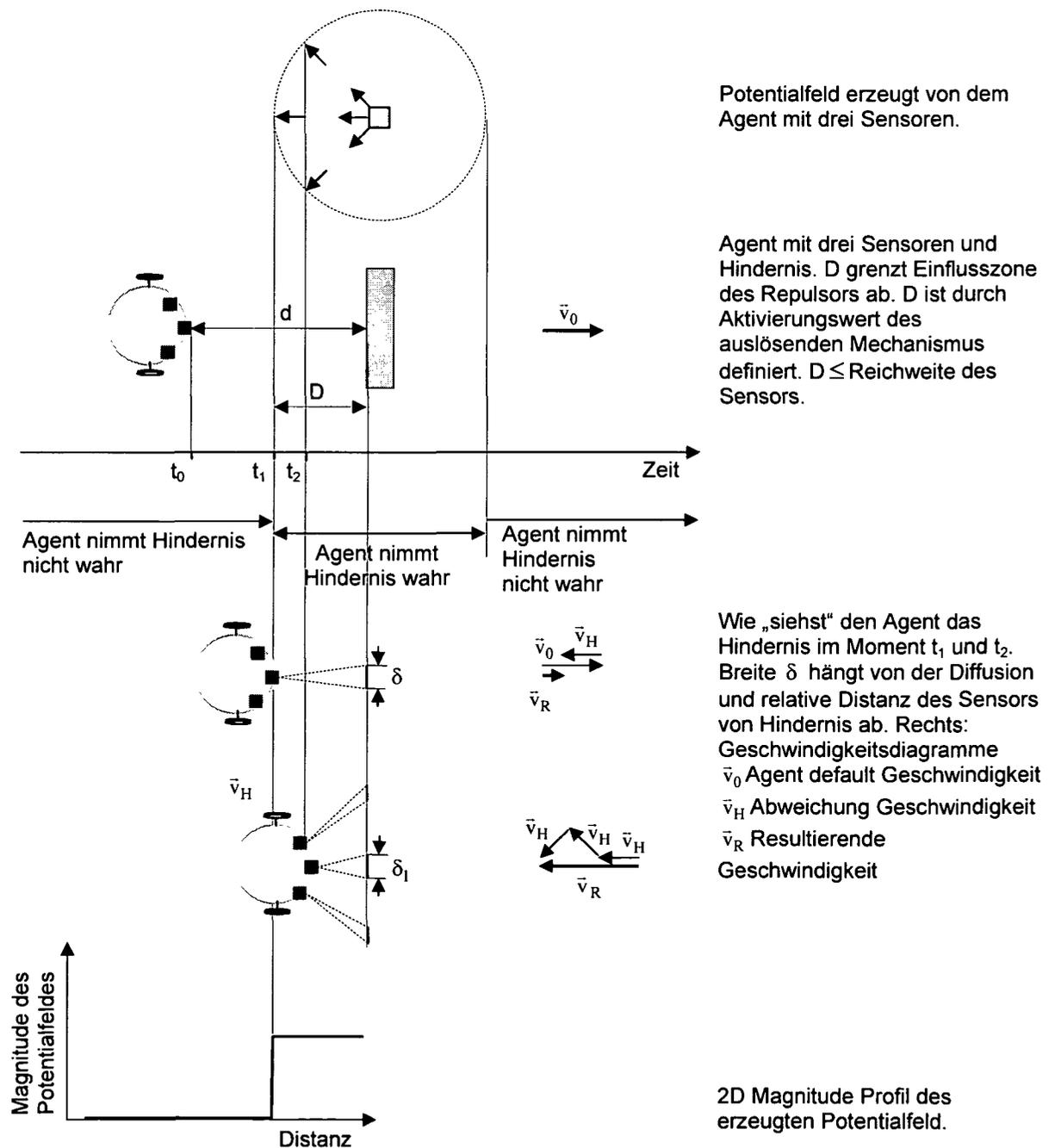


Abb. 5.6. Beispiel vom linearen Potentialfeld.

Außerdem enthält lineares Profil die Information über momentane Distanz des Agents zum Hindernis innerhalb Einflusszone nicht. Eine linearen fallende Funktion beseitigt diesen Nachteil. Ein scharfer Übergang am Rande der Einflusszone bleibt eben. Um dieses Problem zu beseitigen wird exponentiale Funktion nach Veerbeek (Veerbeck, 2001) benutzt:

$$g(f) = e^{-\left(\frac{D}{40}\right)^2} \quad (5.9)$$

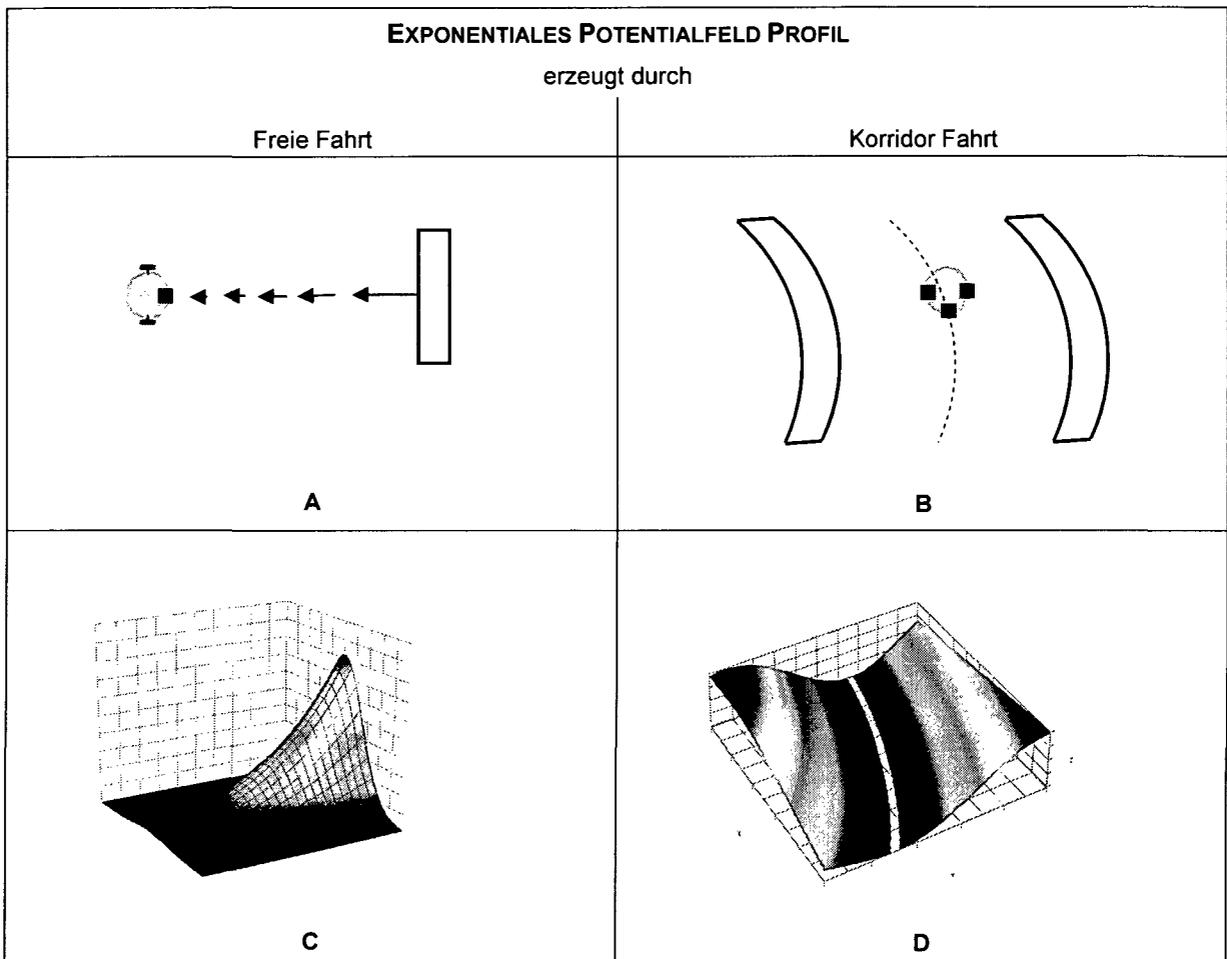


Abb. 5.7. Exponentiales Potentialfeld Magnitude Profil. (A) 2D quadratisches Potentialfeld, das der Agent mit einem Sensor detektieren kann. (B) Beispiel der Erzeugung des Potentialfelds im Korridor. Da sind mindestens drei Sensoren erforderlich (Vorne, links, rechts). Zwei repulsive Felder gliedern sich in dem Zentrum des Korridors aus und bilden damit den Pfad des Agenten. (C) Teil der 3D Repräsentation des repulsiven Potentialfeldes eines runden Hindernis (D) Weise Spur ist der tatsächliche Pfad des Agenten für den Fall B. Wegen Ungenauigkeiten der Sensoren und Mechanik des Agenten liegt der Pfad ungefähr im Zentrum zwischen zwei Wänden. Der Agent bewegt sich ungefähr geradlinig.

Das resultierende repulsive Feld hat exponentiales Magnitude Profil. Das Exponentiale Magnitude Profil beinhaltet Information über die Distanz zum Hindernis D und der scharfe Übergang am Rande der Einflusszone wird umgangen. Auf der Abbildung 5.7 sind erzeugte exponentiale Profile für freien und Korridor Fahrt gezeigt. Auf der Abbildung 5.7 B und C repräsentierte 3D Potentialfelder dienen nur dazu, dass die Magnitude Profile besser repräsentiert werden können. Technisch sind solche Felder zwar realisierbar, aber in Anbetracht der Ebene der Aufgaberealisierung der Agenten innerhalb eines selbstorganisierenden Montagesystems sind sie nicht vertretbar. Dies ist in erster Linie wegen hohen Kosten der Sensoren, die eingebaut werden müssten (Vision Systems) und damit verbunden unnötig vieler erzeugter Signale. In der Realität reicht die Repräsentation der Repulsors durch einen Bruchteil des zweidimensionalen Raums, denn die Größe der Hindernisse bekannt ist und die Agenten sind in zweidimensionalen nicht-holonomen Raum gebunden.

Attraktives Potentialfeld innerhalb Kernsubsystems des Montagesystems entsteht durch die elektromagnetische Strahlung von Hochfrequenz Radiosender, der auf der Zielposition montiert ist. Detailliert wird auf das Hochfrequenz Radiosystem im Kapitel 7 eingegangen. Vom Interesse ist aber das Magnitude Profil des Attraktors. Magnitude Profil ist durch die Funkfelddämpfung kennzeichnet.

Vom Senderausgang wird an die Sendeantenne die Leistung P_t abgegeben, die gleichmäßig in alle Richtungen ausgestrahlt wird. In einem Abstand d vom Senderantenne, wird die ausgestrahlte Energie gleichmäßig über einen Bereich von $4 \cdot \pi \cdot d^2$ verteilt (die Fläche von einem Bereich von Radius d). Die Energie Flussdichte ist:

$$s = \frac{P_t}{4 \cdot \pi \cdot d^2} \quad (5.10)$$

Eine Empfangsantenne erhält in einem ebenen homogenen Wellenfeld der Strahlungsdichte s maximale Wirkleistung P_r :

$$P_r = s \cdot A_r \quad (5.11)$$

Die Wirkfläche einer Antenne ist im Empfangsfall die zur Ausbreitungsrichtung senkrechte Fläche, durch die bei einer einfallenden Welle die maximale Empfangsleistung durch die Antenne hindurchtreten würde. Die Wirkfläche A_r ist für die gleichmäßige kugelförmige (isotrope) Empfangsantenne:

$$A_r = \frac{\lambda^2}{4 \cdot \pi} \quad (5.12)$$

Aus (5.10) und (5.11) folgt:

$$P_r = P_t \left(\frac{\lambda}{4 \cdot \pi \cdot d} \right)^2 \quad (5.13)$$

Aus obiger Gleichung leitet man:

$$d^2 = \frac{\lambda^2}{16 \cdot \pi^2} \cdot \frac{P_t}{P_r} \quad (5.14)$$

oder nach Substitution $\lambda = \frac{c}{f}$

$$d^2 = \frac{c^2}{16 \cdot \pi^2 \cdot f^2} \cdot \frac{P_t}{P_r} = k \cdot \frac{P_t}{P_r} \quad (5.15)$$

Gleichung (5.15) zeigt quadratische Abhängigkeit des Signalniveaus vom Abstand. Magnitude Profil von Attraktoren ist also durch quadratische Funktion gegeben. Außerdem kann der Agent aufgrund bekannter Ausstrahl- und Empfangleistung sowie Senderfrequenz die Entfernung zur Zielposition berechnen. Abbildung 5.8 zeigt das Richtdiagramm und Richtcharakteristik der benützten Empfängerantenne.

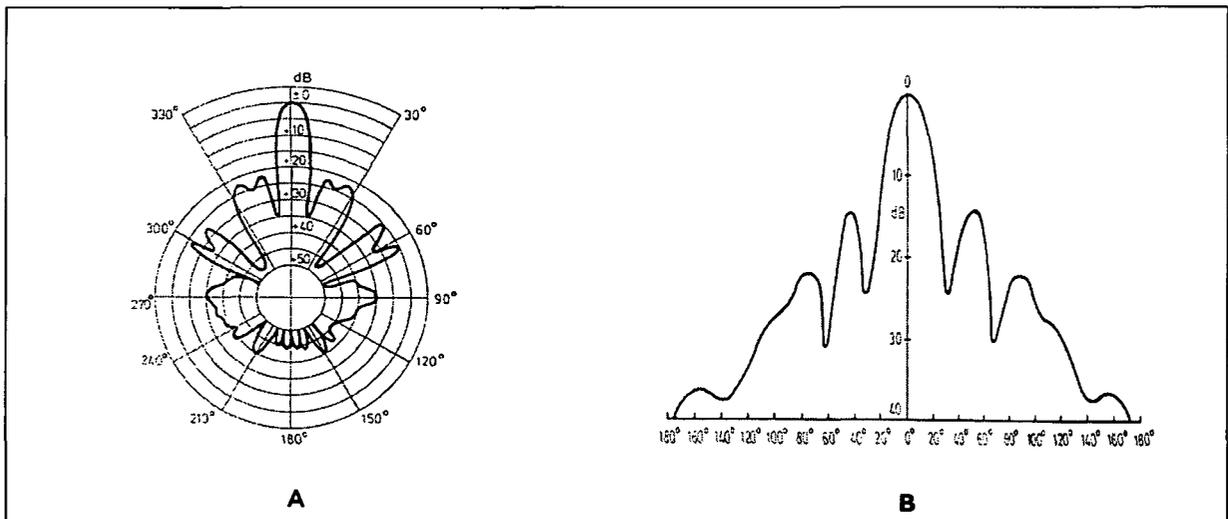


Abb. 5.8. (A) Die Verteilung abgestrahlter Energie im Raum. (B) Das Richtdiagramm der Empfängerantenne. Keine Richtstrahlantenne besitzt eine ideale Richtwirkung. Ein Teil der Energie wird immer seitlich der Hauptkeule und nach rückwärts abgestrahlt, d.h. es treten unerwünschte Nebenkeulen oder Nebenzipfel auf, die durch Nullstellen untereinander von der Hauptkeule abgegrenzt sind.

Abgeleitete Gleichung nimmt Absorption im Raum in der Berechnung nicht und gilt nur für isotrope Antenne. Bei dem Empfängersystem ist es vorteilhaft Richtstrahlantennen zu nutzen. Die Strahlungsleistung wird von Richtstrahlantennen nicht isotrop in den Raum abgegeben. Es gibt bevorzugte Richtungen, die Haupt- und Nebenkeulen und solche, in denen wenig Energie abgestrahlt wird, die so genannte Nullstellen (Abbildung 5.8 a). Dadurch ist es möglich aufgrund maximal empfangener Energie relative Richtung zur Zielposition zu finden.

5.3.4 Erzeugung des Potentialfeldes

Das Sensorsystem, das auf die Zielposition Z reagiert, erzeugt ein attraktives Potential mit seinem Zentrum bei Z . Dadurch wirkt eine Kraft, die den Agenten an die Zielposition zieht. Hingegen eines der N auf die Hindernisse reagierenden Sensoren erzeugen ein repulsives Potentialfeld mit dem Zentrum beim abgetasteten Hindernis H_i . Innerhalb der Reichweite der Hindernissensoren erfährt der Agent daher eine Kraft, die ihn von den Hindernissen wegtreibt. Die Hindernisse sind alle Gegenstände außerhalb momentan suchender Zielposition. Die Superposition der $N+1$ Potentialfelder bzw. die Addition der $N+1$ Kraftvektoren ergibt die auf den Agent wirkende Gesamtkraft.

5.3.5 Das Lokale Minima Problem

Das Potential eines Punktes entsteht als Superposition absoluten Werts von Repulsor und Attraktor Kraftvektor. Je nach der Hinderniskonfiguration gibt es Punkte wo diese zwei Werte gleich sind. Diese Punkte bilden lokale Minima.

Abbildung 5.9 zeigt diese Situation in MobotSim Simulator für zwei Hinderniskonfigurationen. Schwarzer Agent hat die Aufgabe Punkt hinter dem Hindernis zu erreichen. Drei weiße Agenten bewegen sich ziellos. In beiden Fällen, ist Agent in den lokalen Minima gefangen und könnte das Ziel nicht erreichen.

Wenn ein Agent in den lokalen Minima gefangen wird kann er nicht mehr sinnvolle Bewegungen ausüben. Charakteristisch für differential angetriebene zylindrische Agenten ist dass sie sich dann auf der Stelle oder in einem ungefähr zylindrischen Pfad bewegen. Das bedeutet, dass die Differenz der Geschwindigkeitsvorgaben für linken und rechten Motor über längere Zeitraum gleich null oder sehr klein bleibt. Diese Eigenschaft wird benützt um das lokale Minima Problem zu umgehen. Wenn die Differenz der Geschwindigkeitsvorgaben mehr als 2 s innerhalb des Bereichs $0 \leq \Delta \leq 0.1$ liegt, schaltet sich das Zielersteuerungsmodul aus sodass das resultierende Potentialfeld nur aus Repulsiven Kraftvektoren aufgebaut ist. Danach schaltet sich Zielersteuerung wieder ein. Diese

Strategie bringt den Agenten aus dem lokalen Minima. Zeitliche Verzögerung von 2 s dient dazu eindeutig lokale Minima zu detektieren denn es gibt Bewegungsmanöver bei denen scharfe Umdrehungen nötig sind. Eben, diese sind von kürzerer Dauer. Wie lang die Zielansteuerung ausgeschaltet sein soll, hängt von der Hinderniskonfiguration ab und wird in Simulationsverfahren nachgeprüft.

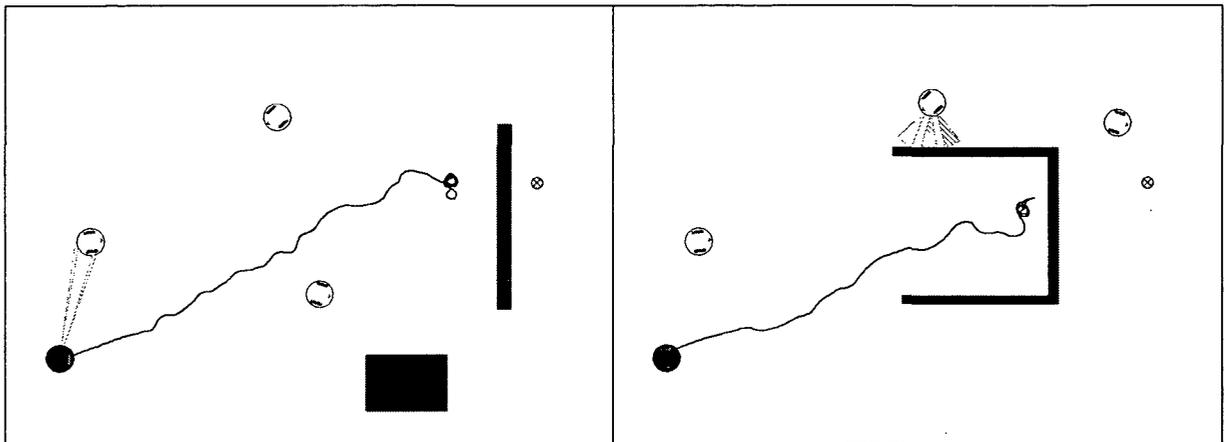


Abb. 5.9. Zwei typische Hinderniskonfigurationen, die das lokale Minima erzeugen.

5.4 Agenten Architektur

Die Aufgabe des Agenten besteht darin markierte Position zu erreichen bei gleichzeitiger Kollisionsvermeidung mit anderen Agenten und statischen Hindernissen und in der Energieerhaltung. Jede Subaufgabe ist von einem Modul kontrolliert.

Zielansteuerung hat die Aufgabe Zielposition zu orten und zu ihr hinzusteuern. Da das Zielansteuerungssystem auf die Aufgabe und Applikationsdomäne maßgeschneidert ist, wird es erst in der Kapitel 7 (S.108) entworfen. Hier ist wichtig dass jede Zielposition eindeutig markiert ist und dass die Zielposition durch eine relative Orientierung Φ und Entfernung d wahrgenommen wird. Damit bietet sich die Erzeugung einer vom Abstand zur Zielposition abhängigen Lineargeschwindigkeit an. Ist die vorgegebene Orientierung gleich Φ , so dreht sich der Agent in die Richtung. Die Lineargeschwindigkeit muss positiv gewählt werden, damit sich der Agent auf die Zielposition zubewegt. Die Erzeugten Vorgaben haben die Form einer Orientierung und eines Geschwindigkeit mit:

$$\begin{aligned}
 v_{Z \text{ direction}} &= \Phi \\
 v_{Z \text{ magnitude}} &= \bar{v}_0 \cdot \tanh\left(\frac{d}{50}\right)
 \end{aligned}
 \tag{5.16}$$

Für große Abstände d bewegt sich der Agent mit einer Lineargeschwindigkeit von $\approx \bar{v}_0$. Verringert sich der Abstand zum Zielpunkt, so geht die Lineargeschwindigkeit gegen $u = 0$. Dies gilt aber nur wenn der Agent geradlinig fährt. Bevor der Agent sinnvolle komplexere Bewegung erzeugen kann, muss die relative Orientierung in die Winkelgeschwindigkeit transformiert werden. Zur Ansteuerung soll eine $(\bar{v}, \bar{\omega})$ - Schnittstelle verwendet werden, die es ermöglicht, die Bewegung des Agents durch Angabe der Lineargeschwindigkeit (\bar{v}) und der Winkelgeschwindigkeit ($\bar{\omega}$) zu steuern. Wie diese Transformation erfolgt, hängt von dem Antriebssystem des Agenten an.

Bei einer differentialen Antriebanordnung ergibt sich die Kurvenfahrt aus den unterschiedlichen Geschwindigkeiten der einzelnen Räder.

Der Zusammenhang zwischen $(\bar{v}, \bar{\omega})$ und den einzelnen Radgeschwindigkeiten ergibt sich aus Gleichungen:

$$v = \frac{v_L + v_R}{2} \left[\frac{\text{cm}}{\text{s}} \right] \quad (5.17)$$

$$\omega = \frac{v_L - v_R}{D} = \frac{d\Phi}{dt} \left[\frac{\text{Grad}}{\text{s}} \right] \quad (5.18)$$

Energieversorgung besteht darin die Ladestation anzusteuern sobald Batterie Spannung 60% erreicht. Energieversorgungs- und Ziellansteuerungsmodul bilden eine Art Subsumption.

Wenn Energieversorgungsmodul nicht aktiv ist verfolgt Ziellansteuerungsmodul Aufgabe bezogenes Ziel. Wenn Spannung auf 60% fällt schaltet Energieversorgungs-Ziellansteuerungsmodul auf Ladestation um.

Hindernisabweichungsmodul sorgt für die Kollisionsvermeidung. Das Modul verfügt über Infrarot und Ultraschallsensoren, die an der Außenseite des Agenten angebracht sind. Wird innerhalb der maximalen Distanz kein Objekt detektiert, so liefert der Sensor den Wert -1. Ansonsten liefert der Sensor die Distanz zu dem wahrgenommenen Objekt in Zentimetern mit einem maximalen Fehler von 2,5 cm für Infrarot beziehungsweise 10 cm für Ultraschall.

Ziellansteuerungs- und Hindernisabweichungsmodul bilden kooperativen Ansatz nach Potentialfeld Methode. Abbildung 5.10 zeigt innere Struktur der Bewegungssteuerung.

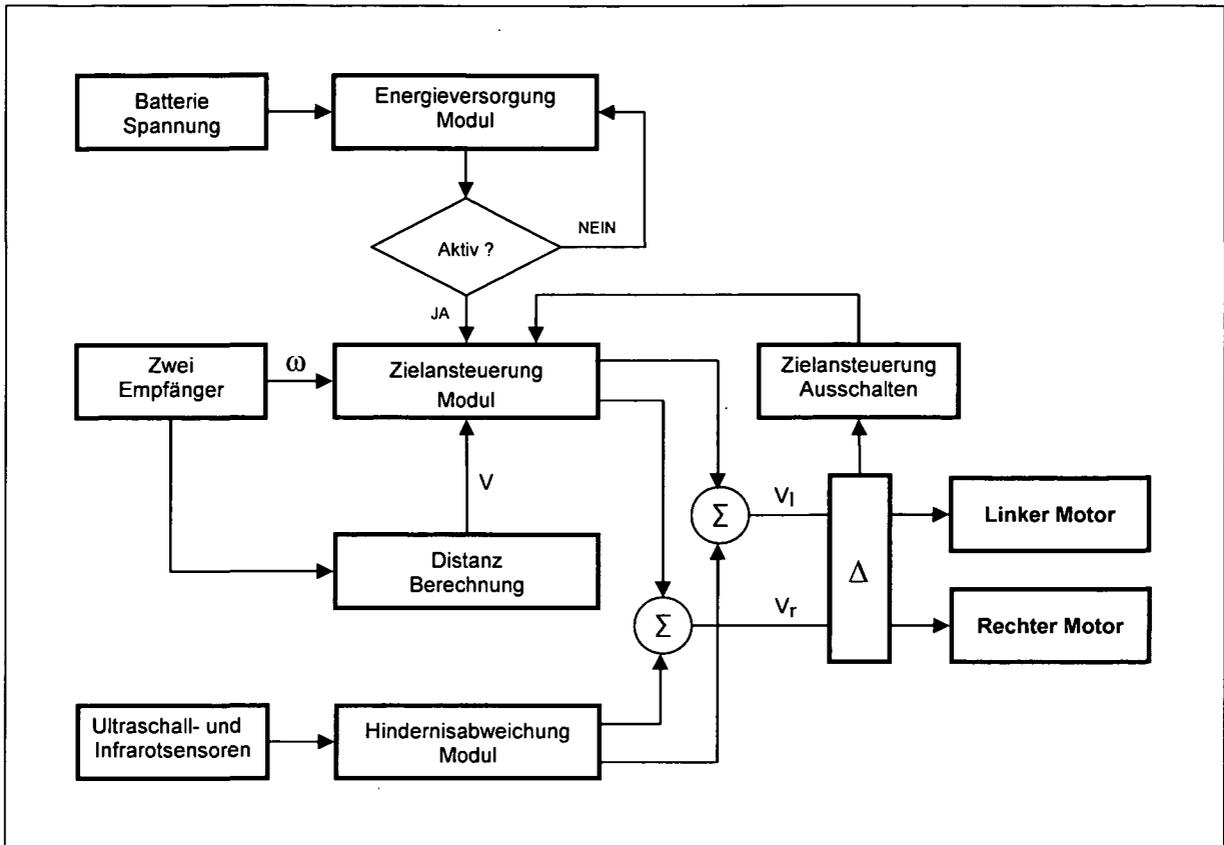


Abb. 5.10. Bewegungssteuerung des Agenten.

5.4.1 Aufbau von Modulen

Module lassen sich sowohl mit den herkömmlichen Regelalgorithmen wie auch mit den Neuronalen Netzwerken, Fuzzy Reglern oder alle Kombinationen davon realisieren. Welcher Methode oder Kombinationen davon der Vorzug gegeben wird, ist ausschließlich eine Frage der Zweckmäßigkeit.

Energieversorgungsmodul hat zwei Zustände, nämlich Aktiv oder Passiv. Zielsteuerung verfolgt gegebene Ziele immer auf dieselbe Art und Weise. Deswegen lassen sich beide Module mit herkömmlichen Regelalgorithmen leicht realisieren.

Nachteilig ist dass sie aus der Designer Sicht aufgebaut sind und daher dem Agenten ganz klar ein gewisses Vorwissen eingebaut wird. Das Verhalten des Agenten bleibt Plan orientiert denn das Verhalten ist vorprogrammiert. Obwohl so aufgebaute Agenten in der Laufzeit autonom interagieren, bleiben Agenten für nicht definierte Situationen blind. Daher können Agenten eigenes Verhalten nicht anpassen.

Da Agenten in einer dynamischen Umgebung eingesetzt werden, ist eine vorprogrammierte Kollisionsvermeidung nicht vertretbar. Deswegen ist Hindernisabweichungsmodul mittels eines Neuronale Netz aufgebaut.

Folgende Gründe motivieren Benutzung Neuronaler Netze für Steuerung autonomer Agenten in der realen Umgebung:

- Ihre Robustheit - sie sind tolerant gegenüber Fehlern und Rauschen. Da ihr Verhalten auf einer Summe mehrerer bewerteter Signale basiert, beeinflussen Oszillationen in den einzelnen Werten dieser Signale das Verhalten des Agenten nicht drastisch. Deswegen sind sie fähig unvollständige Daten zu einem höheren Grad zu behandeln als konventionelle Algorithmen. Dies ist eine nützliche Eigenschaft für den physisch eingebetteten Agenten, der durch geräuschanfällige Sensoren aus gerauschten Umgebungen Informationen liefern müssen.
- Neuronale Netze bieten einen relativ glatten Suchraum an. Allmähliche Änderungen an den Parametern die ein neuronales Netz definieren (Verbindungsstärken, Zeitkonstanten, Strukturen), verursachen dem entsprechend Änderungen seines Verhaltens.
- Sie sind intrinsische Lernsysteme und sie sind in der Lage zu verallgemeinern. Sie erlauben Agenten aus der Interaktion mit ihrer Umgebung zu lernen. Auf diese Art reduziert sich die Aufgabe des Designers auf die Auswahl einer passender Neuronal Netz Struktur.
- Ein weiterer Grund ist damit verknüpft, dass sie dem natürlichen Gehirn nachempfunden sind, wodurch sich Ideen aus der Neurobiologie relativ einfach verwirklichen lassen. Parallelität selbst ist eine solche Idee.
- Neuronale Netze bieten die Möglichkeit einer natürlichen Einbettung in Agenten. Man kann sie direkt zwischen Sensoren und Aktoren einbauen.
- Wenigstens kleinere Neuronale Netze sind sehr effizient und die Optimierung des Netzes kann von konventionellen Rechner berechnet werden kann (Churchland, 2001).
- Aufbau von Neuronalen Netz basierende Steuerung vom Intentionaler Standpunkt ist nur bei einfachen Neuronalen Netzen möglich. Bei komplexeren Neuronalen Netzstrukturen kann man nicht mehr aus Aktivierungssignalen und synaptischen Verbindungsstärken eindeutig auf Verhalten des Agenten zurück schließen. Das zwingt zur Konzentration auf die Leistung des Agenten, die er ausweisen soll. Also Umgebungs-Agent Interaktion kommt automatisch zum Vorschein.

5.4.2 Welche Neuronale Netz Struktur eignet sich für die Bewegungskontrolle den Agenten

Zahlreiche Autoren (Beer, 1990, Biro, 1998, Meeden, 1996, Nolfi, 1999, Tani, 1996, 1998, Sharky, 1996,1999, Ulbricht, 1996, Ziemke 1998, 1999) haben umfassende experimentelle Vergleiche von verschiedenen Neuronalen Netzstrukturen für Bewegungskontrolle des Agenten ausgeführt. Die experimentellen Ergebnisse haben eindeutig gezeigt, dass rückgekoppelte Netze konsistent bessere Resultate als Feed Forward Neuronale Netze geliefert haben. Unterschied in der Leistung zwischen Feed Forward und rückgekoppelten Netzen war durchschnittlich 30 % für Kollisionsvermeidung in einer einfacher Umgebungen mit statischen Hindernissen und ist auf 55% gestiegen für komplexere Umgebungen mit dynamischen Hindernissen.

Was fehlt Feed Forward Netzwerken? In dem Feed Forward Netzwerk sind die einzelnen Neuronschichten durch Verbindungen miteinander verknüpft, die nur in eine Richtung verlaufen, nämlich aufwärts. Ein Feed Forward System arbeitet wie eine Pipeline. Je weiter vom Beginn der Pipeline man eine Probe des Signalflusses zieht, desto mehr Zeit ist seit der Einspeisung dieses Signals in die Pipeline vergangen und desto älter ist auch das Ergebnis, das das Signal beschreibt. Dies bedeutet, dass Feed Forward gesteuerten Agenten keine Vorstellung von den Zeit Aspekten seines Verhalten hat, und dass solche Agenten immer auf seinen Sensorenzustand in derselben, vom gegenwärtigen Kontext, unabhängigen Art reagieren. Ein Feed Forward Netz ist nicht in dem Zeitkontinuum eingebettet.

Um kognitives Verhalten beeinflussen zu können braucht man eine Art Kurzzeitgedächtnis. Die Feed Forward Netzwerke besitzen bereits eine Form von Gedächtnis: das Wissen oder die Fähigkeiten, die im Gesamtmuster seiner synaptischen Verbindungsstärken verschlüsselt liegt. Diese Form von Gedächtnis ist jedoch blind für die Details bestimmter Ereignisse der Vergangenheit. Churchland (Churchland, 2001) gibt dafür folgendes Beispiel; Ein Stein, den stetige Wassertropfen ausgehöhlt haben, bezeugt natürlich das Werk, das unzählige Tropfen über die Jahre hinweg geleistet haben, aber Aussehen, Größe, Temperatur, pH-Wert und Zeitpunkt jedes einzelnen Wassertropfens sind Informationen, die trotz der Form des Steines für immer verloren sind. Ganz ähnlich verhält es sich mit dem Muster der Verbindungsstärken in einem neuronalen Netzwerk. Das Endresultat dieses Prozesses enthält weder Informationen über die einzelnen Eingangs- und Ausgangssignale noch über die unzähligen Veränderungen der Synapsengewichte, die es in seine gegenwärtige Form gebraucht haben. Wenn das Netzwerk explizit bestimmte Ereignisse in der nahen Vergangenheit erfassen will, benötigt es einen zusätzlichen Mechanismus.

Wenn ein Axion an einer Nervenzelle weiter hinten in der Pipeline entspringt und zurück nach vorne läuft, liefert es Signale über vergangene Aktivitäten des Netzwerks nach vorne. Mit diesen zusätzlichen absteigenden Bahnen ist das Netzwerk nicht mehr nur in der

hauchdünnen Schicht Gegenwart gefangen, sondern sein kognitiver Raum erstreckt sich nun wenigstens Bruchteile von Sekunden in die Vergangenheit.

Diese Signale stehen dann für die aktuellen Verrechnungen in der Neuronensicht zur Verfügung. Das sind so genannte rückgekoppelte Verbindungen. Rückgekoppelte Verbindungen erlauben einen ständigen Zugriff auf die unmittelbare kognitive Vergangenheit eines Agenten, die mit den aktuellen sensorischen Signalen der Gegenwart verrechnet werden kann.

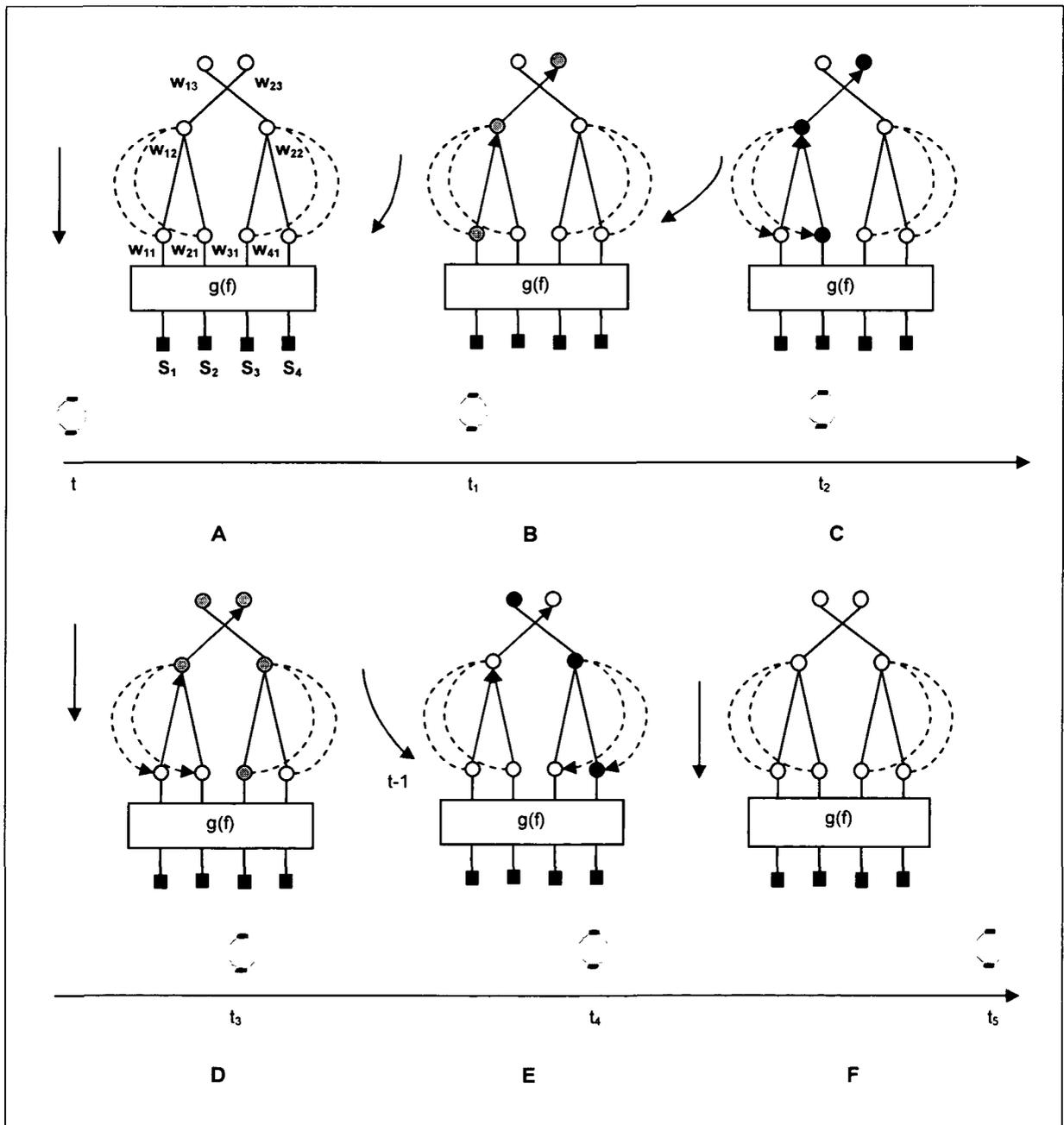


Abb. 5.11. Kollisionsvermeidung mittels eines einfachen rückgekoppelten Neuronalen Netzwerks.

Dadurch kann ein Agent bei seiner Reaktion auf die derzeitige Situation auch unmittelbar vorhergehende Ereignisse berücksichtigen. Das heißt, rückgekoppelte Neuronale Netze nutzen ihren internen Zustand um gegenwärtige Sensorensignale in der nächsten Aktion auszuführen. Der Agent wird nicht mehr auf dieselbe Art und Weise auf Sensorsignale reagieren. Damit ist Verhaltensrepertoire des Agenten erweitert.

Auf der Abbildung 5.11 ist ein rückgekoppeltes Neuronales Netz mit vier Infrarotsensoren gezeigt. Wenn sich Hindernis innerhalb der Reichweite der Sensor befindet, liefert der Sensor die Distanz zum Hindernis. Bevor Sensorsignale Eingangsschicht des Netzes erreichen, werden sie durch das Magnitude Profil Funktion $g(f)$ und Relevanzfaktoren geändert.

Die Erzeugung der entsprechenden Abweichungsbewegung geschieht durch die Vorgabe von Orientierungen der aktivierten Sensoren. Die Überlagerung der Orientierungen führt daher zu einer Winkelgeschwindigkeit, welche eine Drehung des Agenten in den freien Raum veranlasst. Da Differentialbetriebene Agenten nur durch den Unterschied der Lineargeschwindigkeiten des linken und rechten Motors Umdrehungen machen können, müssen jeweilige Orientierungen der Sensoren direkt in der Lineargeschwindigkeiten repräsentiert werden. Dazu ist Relevanzfaktor eingeführt. Durch den Einfluss der Relevanzfaktoren auf das Gewicht der Eingangsschicht des Netzes ist implizit Orientierung der Sensoren kodiert. Der Sensor, der in der Fahrriechtung tätig ist, hat die größte Relevanz. Je größer der Winkel zwischen Sensorposition und Fahrriechtung ist, desto ist die Relevanz des denjenigen Sensorsignal weniger.

Experimental bestimmt man die Relevanzfaktoren wie im Folgenden beschreiben wird. Man legt die Relevanzfaktoren so, dass sie gemeinsam das gewünschte Verhalten vollziehen. Wenn man aber nicht weiß, wie die Relevanzfaktoren verteilt sein müssen, setzt man alle Relevanzfaktoren auf zufällige Werte. Die Werte sollen sowohl positive als auch negative sein, die zwischen -1 und 1 liegen.

Da die Relevanzfaktoren zufällig gesetzt sind, wird das Ergebnis höchstwahrscheinlich nicht dem gewünschten Verhalten entsprechen. Diese zufällige Verteilung der Relevanzfaktoren zeigt wie weit das Verhalten des Agenten danebengegangen ist. Die gemessenen Abstände r_1 ; r_8 können als ein Maß für den Erfolg des Hindernis Vermeidungsmodul angesehen werden. Sind alle Entfernungen groß (bzw. gleich -1, wenn kein Objekt wahrgenommen wird), dann erfüllt Hindernis Vermeidungsmodul seine Aufgabe gut, da keine Gefahr für den Agenten durch andere Objekte besteht. Sind die gemessenen Abstände dagegen klein, so hat Hindernis Vermeidungsmodul durch Bewegung des Agenten dafür Sorge zu tragen, dass der Abstand zu den wahrgenommenen Objekten vergrößert wird. Die Aufgabe von Hindernis Vermeidungsmodul kann daher auch als Maximierung (unter Berücksichtigung, dass ein maximaler Abstand durch den Wert -1 gegeben ist) aller gemessenen Abstände interpretiert

werden. Tabelle 5.1 gibt die Gewichte für acht Neuronen in der Eingangsschicht die im Simulationsverfahren optimiert sind. Im obigen Beispiel werden nur erste vier benutzt.

SENSOR	ORIENTIERUNG	GEWICHT Relevanzfaktor x Magnitude Profil Funktion
1.	$\varphi_1=0^\circ$	$w_{1,1} = -g(f)$
2.	$\varphi_2=330^\circ$	$w_{2,1} = 0.95g(f)$
3.	$\varphi_3=300^\circ$	$w_{3,1} = 0.9g(f)$
4.	$\varphi_4=225^\circ$	$w_{4,1} = 0.5g(f)$
5.	$\varphi_5=180^\circ$	$w_{5,1} = g(f)$
6.	$\varphi_6=135^\circ$	$w_{6,1} = 0.5g(f)$
7.	$\varphi_7=60^\circ$	$w_{7,1} = 0.9g(f)$
8.	$\varphi_8=30^\circ$	$w_{8,1} = 0.95g(f)$

Tab. 5.1. Gewichte für acht Neuronen in der Eingangsschicht. $g(f) = e^{-\left(\frac{D}{40}\right)^2}$

Signal aus dem Sensor wird die Änderung der Gewichte in der Eingangs- und weiter in der Zweiten- und Ausgangsschicht verursachen. Je größer das Gewicht in der Ausgangsschicht ist, desto mehr Strom fließt in den entsprechenden Motor. Zweite Schicht ist mit der Eingangsschicht noch durch Rückkopplung verbunden. Also Signal dass die zweite Schicht in der Zeit t_1 erreicht hat, wird noch mal in der Zeit t_2 Eingangsschicht erreichen und dort mit dem Sensorsignal in der Zeit t_2 summieren. Rückgekoppeltes Signal beträgt $0.38 \times w_{2i}$. Das bedeutet dass das Gewicht w_{2i} (Abbildung 5.11 B) und entsprechend auch w_{3i} in der Zeit t_2 noch größer wird. Also Agent wird noch schärfer nach rechts abbiegen. Wenn es aber in der Zeit t_2 kein zusätzliches Sensorensignal gibt, verringert sich das Gewicht $w_{2i}(t_2) = 2 \times 0.38 \times w_{2i}(t_1)$. Der Agent wird allmählich auf dem früheren Pfad zurückkommen. Im Fall dass ein Feed Forward Netzwerk benützt werden würde, würden die Änderungen in den Gewichten zwischen t_1 und t_2 wesentlich größer.

SCHICHT	GEWICHT
ZWEITE	$w_{1,2} = \frac{w_{1,1} + w_{2,1}}{2}$
	$w_{2,2} = \frac{w_{3,1} + w_{4,1}}{2}$
	$w_{1,2R} = 0.38 w_{1,2}$
	$w_{2,2R} = 0.38 w_{2,2}$
AUSGANGSSCHICHT	$w_{1,3} = w_{2,2}$
	$w_{3,3} = w_{1,2}$

Tab. 5.2. Gewichte für zweite und Ausgangsschicht.

Das heißt dass sich der Strombedarf schlagartig ändern. Dies führt zur oszillatorischen Bewegung und zugleich zu größerem Stromverbrauch. Rückgekoppelte Netze ermöglichen glattere Bahn des Agenten und Strömersparnis.

5.5 Simulation

Der Zweck der Simulation ist das attraktive und repulsive Potentialfeld so abzustimmen, dass der Agent eine zuverlässige, zielgerichtete Bewegung erzeugen kann und das Verhalten der Agenten mit einer großen Anzahl von dynamischen Hindernissen zu prüfen.

Unter der Voraussetzung das die Morphologie des Agenten, Attraktives Potentialfeld, Magnitude Profil des Potentialfeldes und Relevanzfaktoren festgestellt sind kann das repulsive Feld anhand von der Reichweite, Einordnung und Zahl der Infrarotsensoren erzeugt werden.

Simulation ist in MobotSim Simulator durchgeführt. MobotSim ist Simulationssoftware für 2 D Simulation von differential betriebenen Agenten in konfigurierbarer Umgebung. Vorteile der Simulation: Unbegrenzte Anzahl von Agenten und Hindernissen verschiedenen Größen und Formen, Flexible Konfiguration von Agenten (Durchmesser, Entfernung zwischen Rädern, Geschwindigkeit, Anzahl, Typ und Einordnung von Sensoren), Flexible Konfiguration von Sensoren (Einordnung, Strahlungswinkel, Reichweite und mitteln Basic Editor Magnitude Profil), Vollständig visuelle Grundlage für Bewegungen, mittels Visual Basic für Applikationen kann Steuerung durch Fuzzy Logic, Neuronale Netze oder herkömmlichen Algorithmen aufgebaut sein.

Nachteilig ist das bei der Konzeption von Antrieb Kinematik die Dynamik des Systems nicht in Betracht gezogen ist.

Es ist in einer Vielzahl von Arbeiten, z.B. (Nolfi, 1997) gezeigt worden, dass die Simulation von Infrarotsensoren und Differential basierenden Aktorik genug realistisch ist um die mit der Simulation entwickelte Lösungen direkten auf realen Agenten zu transferieren.

Simulierter Agent hat 100 cm in Durchmesser. Seine maximale Lineargeschwindigkeit beträgt 0.8 m/s, Winkelgeschwindigkeit 128 Grad/s.

Sensoren sind ringförmig auf Durchmesser von 1 m und 0,6 m über dem Boden platziert. In MobotSim sind diese Angaben über die Geometrie des Agenten und Sensoren Konfiguration definiert. Abbildung 5.12 zeigt Agenten Charakteristika die in Simulator definiert sind.

Agenten Umgebung ist 20 x 25 m flache Oberfläche mit unterschiedlicher Anzahl von statischen und dynamischen Hindernissen. Statische Hindernisse dienen dazu das Problem of Local Minima zu erzeugen. Bei beweglichen Hindernissen ist dieses Problem schwer zu untersuchen, denn die Situation ändert sich schnell. Dynamische Hindernisse sind andere Agenten.

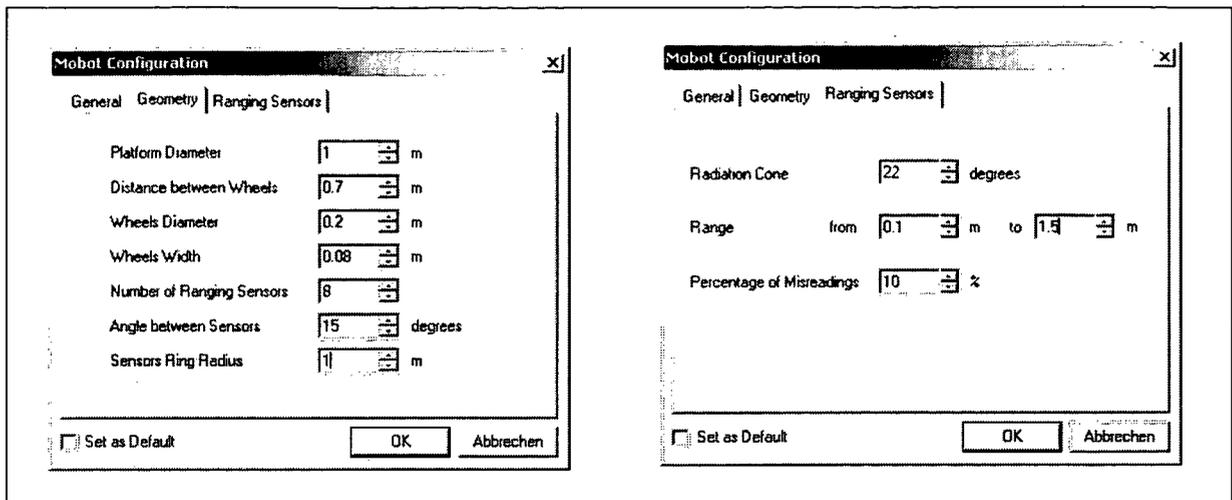


Abb. 5.12. Repräsentation von Agenten Geometrie und Sensoren Konfiguration in MobotSim Simulator.

Da sich die Agenten auf zufälligen Pfaden bewegen und keinem Versuch gemacht werden, ihre zukünftigen Bewegungen vorherzusagen, kann ein einfaches Sensormodell ohne größere Behinderungen eingesetzt werden (Gini, 1996). Genaueres Sensormodell würde nur Berechnungszeit ausdehnen aber nicht dazu beitragen bessere Hindernisabweichung Strategien zu machen (Gini, 1996).

Die Simulation ist unter folgenden Annahmen realisiert:

- Der Agent verfolgt gleichzeitig ein Ziel und das Ziel ist als eine Position repräsentiert.
- Die Umgebung enthält bewegliche und statische Hindernisse.
- Bewegung von Hindernissen ist unvorhersehbar.
- Die Hindernisse sind bekannt. Das sind anderen Agenten, Operatoren und physische Begrenzung der Agenten Umgebung.
- Die Hindernisse werden von den Agenten eigenen Sensoren wahrgenommen.
- Die Hindernisse machen keinen Versuch Kollision mit dem Agenten zu vermeiden. Der Agent muss allein die Kollision vermeiden.
- Energieversorgungsmodul ist immer ausgeschaltet.

Das Ziel erster Untersuchung ist die Optimierung der Einordnung und Zahl der Sensoren bei konstanter Sensorenreichweite und Agenten Geschwindigkeit. Die Zahl der Infrarotsensoren

variiert zwischen 8 und 18 und deren Winkel zwischen 10 und 20 Grad. Für jede Kombination sind 20 Versuche durchgeführt. Zusammenfassung der Ergebnisse ist in der Tabelle 5.3 gezeigt.

	ANZAHL DER SENSOREN	SENSOREN ABSTAND in Grad	SIMULATION ERGEBNIS		
			Ziel erreicht	Zeit in s	Anzahl von Kollisionen
Reichweite: 150 cm Geschwindigkeit: 0.8m/s Öffnungswinkel 2°	18	20	115	24.32	18
	12		114	26.20	28
	8		108	28.55	35
	18	15	114	27.80	25
	12		117	26.75	16
	8		113	26.50	21
	18	10	118	29.80	11
	12		118	27.60	12
	8		114	28.10	24

Tab. 5.3. Ergebnisse erster Simulation.

Ergebnisse haben gezeigt dass gemäßige Einordnung der Sensoren mehr bringt als mehrere Sensoren. Z.B. 12/10 hat bessere Resultate gezeigt als 18/15. Abbildung 5.13 zeigt Agenten Pfad für zwei beste Kombinationen 12/10 und 18/10. Verkleinerung des Abstandes zwischen Sensoren erhöht Dichte in betroffenen Gebiet.

Die Ausdehnung der Sensorenbreite verbessert die Situation nicht immer, denn der Agent wird auf entfernte Hindernisse reagieren. Dies führt zu einem unnötig langen Pfad und oft, dazu dass der Agent überhaupt die Zielposition nicht erreichen kann.

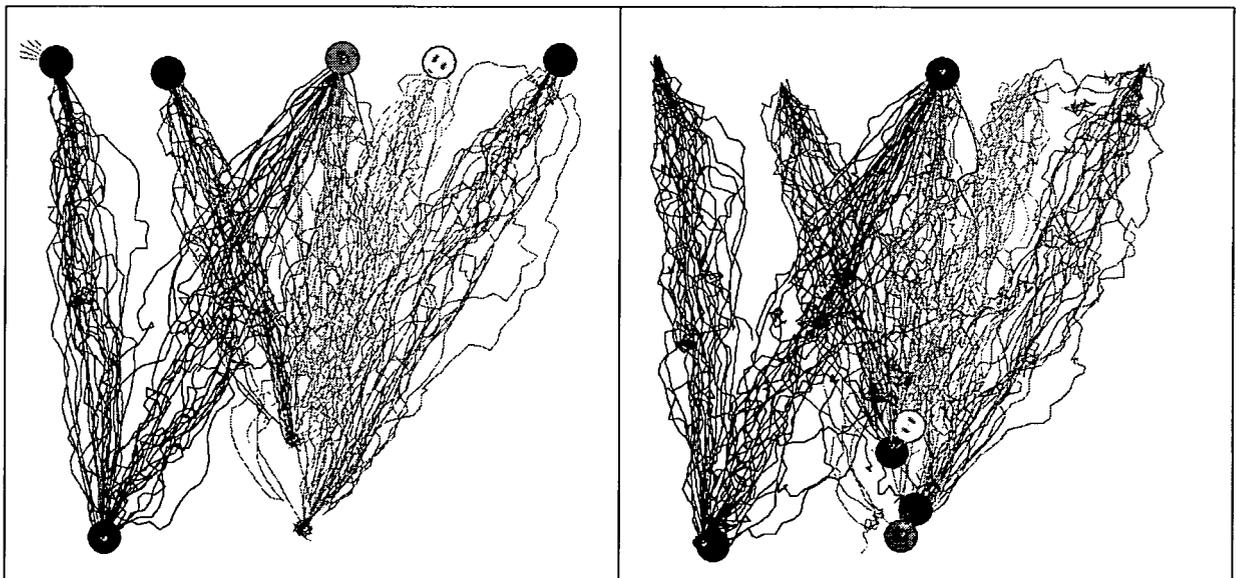


Abb. 5.13. Agenten Pfad für 12/10 und 18/10 Konfiguration. (A) Agenten in der Anfangs- und (b) Endposition.

Infrarot: Anzahl/Abstand: 12/10 Reichweite: 150 cm Öffnungswinkel 2° Anzahl/Abstand: 4/10 Reichweite: 300 cm Öffnungswinkel 15° Agentengeschwindigkeit: 0.8m/s	SIMULATION ERGEBNIS		
	Ziel erreicht	Zeit in s	Anzahl von Kollisionen
	120	24.10	6

Tab. 5.4. Ergebnis Optimierte Sensoren Zahl und Einordnung.

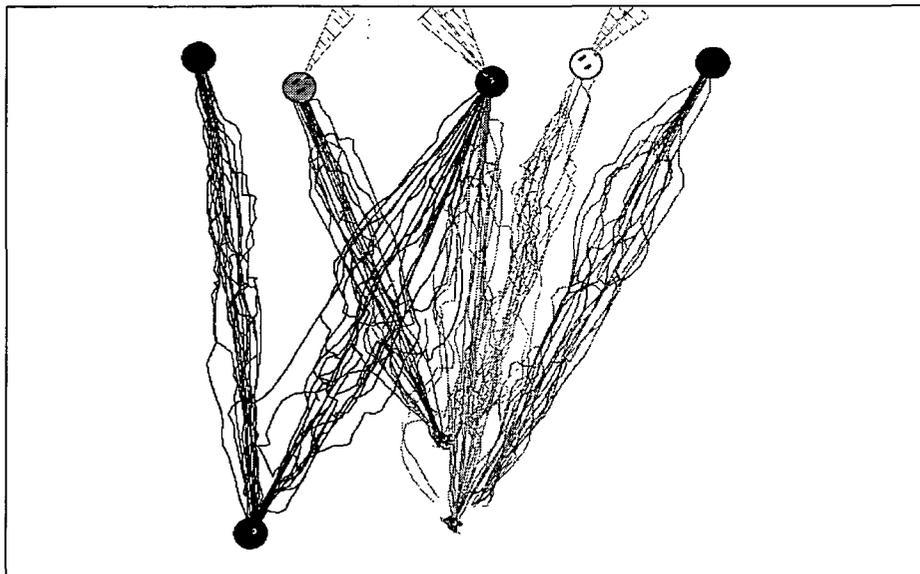


Abb. 5.14. Agenten Pfad mit 4 Ultraschall und 12 Infrarotsensoren.

Agenten Geschwindigkeit	Agenten Anzahl	Durchschnittliche Zeit (s)	Zeit Deviation	Ziel erreicht	Anzahl von Kollisionen
10	10	79.07	13.40	99	1
10	20	93.92	20.93	95	5
10	30	110.19	27.35	98	2
10	40	126.23	34.25	92	8
10	50	135.06	41.06	82	18
30	10	80.75	11.98	99	0
30	20	96.17	23.43	95	4
30	30	110.95	28.18	89	4
30	40	116.94	28.91	80	9
30	50	125.46	32.30	72	8
50	10	85.10	18.56	92	1
50	20	97.56	19.37	75	4
50	30	111.48	23.09	63	4
50	40	123.48	29.11	37	12
50	50	127.72	29.49	32	9

Tab. 5.5. Ergebnisse zweiter Simulation.

Von klassischer bis agentenbasierender Steuerung

Um produzieren zu können, müssen der Information-, Energie-, Material- und Arbeitsmittelfluss innerhalb des Fertigungssystems so organisiert sein, dass am Ende der vorausdefinierten Zeit genauer Bestand und die Qualität der gewünschten Produkte unter Berücksichtigung der angemessenen Kosten zum Vorschein kommen.

Erster Schritt bei der Gestaltung dieser Organisation ist Produktionsplanung. Die Produktionsplanung besteht aus der Auswahl und Sequenzierung von Produktionsprozessen. Damit können einzelne oder mehrere Ziele erfüllt werden und somit ein Satz von Domänenbeschränkungen eingehalten werden.

Der Auswahlprozess zwischen alternativen Plänen heißt Scheduling. Dabei geht es um das Zuteilen von Produktionsressourcen und Zeiten zu Satz von ausgewählten Produktionsprozessen im Plan.

Scheduling ist ein Optimierungsprozess, an dem beschränkte Produktionsressourcen über Zeit an parallele und sequentielle Aktivitäten vergeben werden. Scheduling gibt die Antwort auf folgenden Fragen:

WAS	Es gibt ein Set von Aufgaben, das erledigt werden muß. Aufteilung der globalen Aufgabe auf Basis Aufgaben beantworten auch die Frage Wie produziert werden muss.
WANN	Es gibt Zeitperiode und Zeitintervalle (Start-Stop) wann eine Aufgabe bearbeiten werden muss.
WO	Es gibt ein Set von Arbeitsmitteln, das benutzt werden muss um die Aufgaben erledigen zu können. Unter diesen sind nicht nur die Maschinen sonder auch Werkzeuge und Material definiert.

Tab. 6.1. Die Fragen, auf die das Scheduling antwortet.

Das klassische Ablaufplanungsproblem besteht darin, n Aufträge auf m Maschinen unter verschiedenen Beschränkungen und Auftragsprioritäten aufzuteilen. Theoretisch gibt es für

nur eine Maschine $n!$ mögliche Permutationen. Folglich für m parallelen Maschinen gibt es $n!m^n$ Untersuchungsraum. Das ist aufgrund vom rechnerischen Aufwand und benötigten Zeit in industrieller Umgebung nicht vertretbar. Um Untersuchungsraum zu verkleinern, greift man zu Statistik oder neulich zu KI Heuristik zurück. In beiden Fällen versucht man Produktionsparameter wie Maschinenstillstände zu minimieren oder Produktionskapazität zu maximieren.

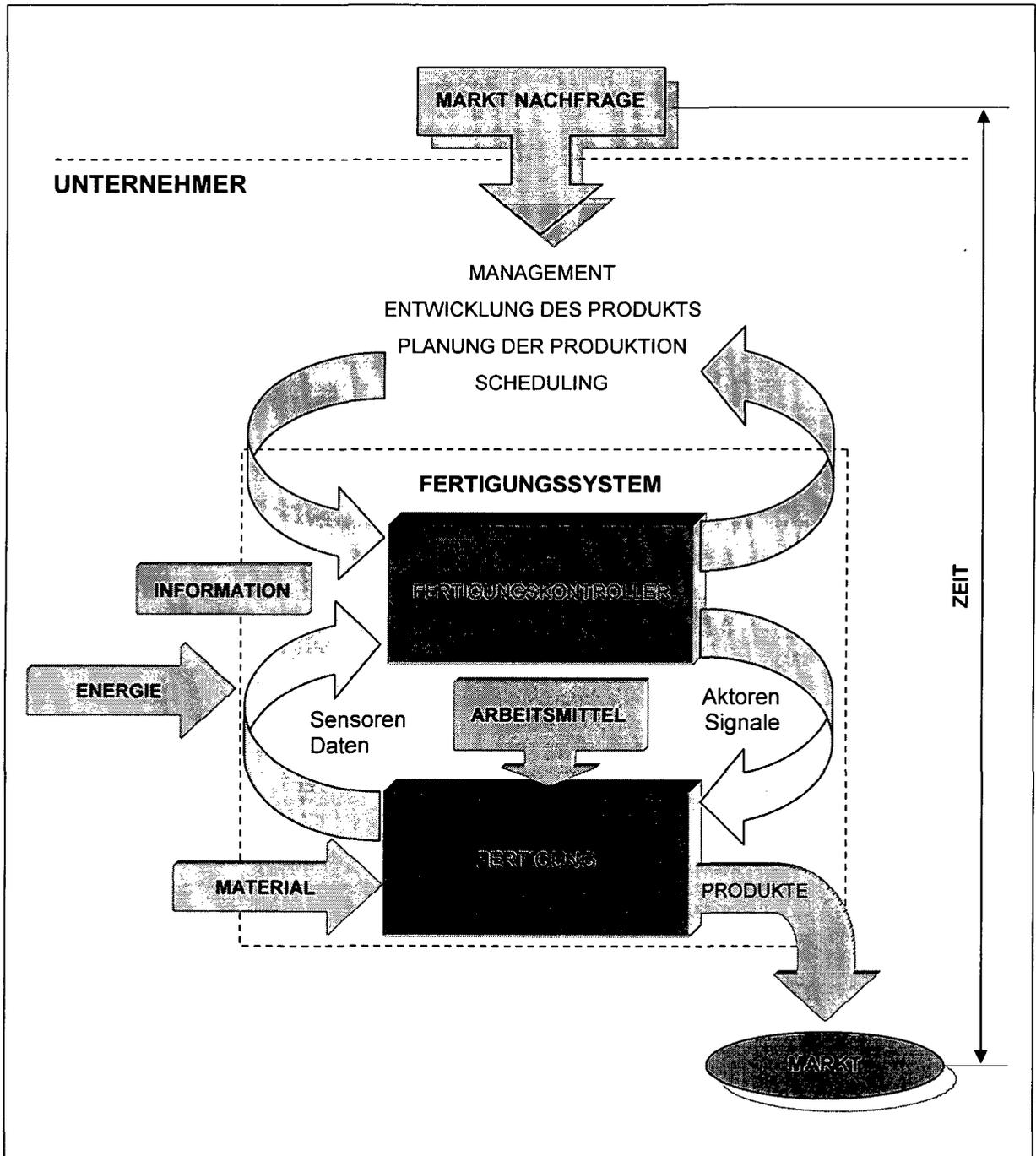


Abb. 6.1. Fertigungssystem und dessen globale Kontrolle.

Da der verwendeter Ansatz auf Just in Time beruht wird er in folgenden Kapiteln für den Einsatz in Montagesystem maßgeschneidert.

Die Nervenstränge eines Produktionsablaufes sind seine Informationsflüsse. Als Information bezeichnet man zweckbezogenes Wissen über Zustände und Ereignisse. Fehlt diese Zweckeignung, so spricht man eher von Daten.

Die Art und Weise wie ein Informationsfluss organisiert ist bestimmt in großen Ausmaß die Steuerungs- und die Konfigurationsart eines Produktionssystems. Es gibt zwei extreme Formen den Informationsfluss zu strukturieren. In dem einen Extremfall besteht der Informationsfluss aus aufeinander folgenden Informationen. Folge dessen ist eine Steuerungs- und Konfigurationsart des Produktionssystems, die auf aufeinander folgenden Operationen aufgebaut ist.

Das sind sequentielle Systeme. Eine Uhr, die durch einen komplizierten Bewegungsablauf die Zeit misst, ist der Archetyp solcher Systeme. Die meisten existierenden Produktionssysteme folgen diesem Prinzip.

Auf der entgegengesetzten Seite finden sich Systeme, die auf parallelen Operationen aufgebaut sind. Beispiele solcher Systeme sind das neuronale Netzwerk eines Gehirns oder einer Ameisenkolonie. Organisatorisch handelt es sich jeweils um eine Ansammlung vieler autonomer Agenten. Diese autonomen Agenten sind mit allen anderen, jedoch nicht mit einer Zentralinstanz hochgradig verschaltet. Sie bilden somit ein Netzwerk von gleichrangigen Einheiten. Diese Feststellung ist zugleich der Leitgedanke bei der Entwicklung des biologischen Montagekonzepts innerhalb dieser Dissertation.

6.1 Klassische Produktionssteuerung

Klassische Produktionssysteme sind deterministisch basierende Systeme, die auf a priori Information und auf die Voraussagen der Aktivitäten innerhalb eines geschlossenen Produktionssystems angewiesen sind (Peklenik, 1997).

Die Steuerung vorhandener Produktionssysteme ist planorientiert. Zuerst wird der Produktionsprozess im Detail geplant, dann werden die einzelnen Schritte ausgeführt. Die strikte Trennung von Planung und Ausführung führt jedoch zu einer ständigen Diskrepanz zwischen Soll und Ist Zustand. Da Änderungen der Pläne zeitaufwendig sind, finden bei Störungen Umplanungen oftmals situativ statt und führen zu einem praktikablen, nicht aber zu einem optimalen Planungsergebnis.

Die Planorientierung findet ihre Entsprechung in einer zentralisierten hierarchischen Steuerungsorganisation. Ausgehend vom Gesamtplan werden die geplanten Arbeitsschritte über die einzelnen Ebenen der Hierarchie an die ausführenden Produktionsbereiche verteilt. Eine Anpassung des Plans erfolgt nur im Rahmen der lokalen Sicht einer Hierarchie-Einheit.

Der Planorientierte Steuerungsansatz ist damit in starren und störanfälligen Strukturen festgelegt.

Um Konsistenz der Daten durch das ganze Unternehmen zu sichern ist eine zentrale Database aufgebaut. Folge dessen ist eine sukzessive Steigerung der Datenmenge durch die Hierarchieebenen. Gerade bei der Herstellung fertigungs-technologisch anspruchsvoller Produkte in kleinen und mittleren Stückzahlen und einem größeren Mix der Produkte stoßen die bestehenden Informationssysteme an ihre Grenzen (Tönshoff et al, 2000). Außerdem lässt die zunehmende Komplexität von Fertigungsprozessen und die Integration verschiedener Bearbeitungsverfahren in einzelnen Maschinen, das zur Planung notwendige Wissen, das durch den meist guten Ausbildungsstand der Maschinenbediener (Facharbeiter) lokal, nicht aber zentral, vorhanden ist, deutlich ansteigen.

Zentral gerichtete Steuerung des Systems zieht bei dem Ausfall des Zentralrechners den Ausfall des gesamten Systems nach sich.

Steigernde Komplexität der Produktionssysteme verringert weiter deren Störungstoleranz und Wiederverwendung. Das macht vorhandene Systeme noch starrer und unzuverlässiger.

6.2 Agentenbasierende Produktionssteuerung

Agentenorientierte Produktionssysteme hingegen bieten einen neuartigen Steuerungsansatz, der zu einer flexiblen und robusten Produktionssteuerung führt. Grundsätzlich bleibt eine gewisse Vorplanung dort erhalten, wo sie für die Vorbereitung des Produktionsprozesses notwendig ist, z.B. bei der Bedarfs- und Ressourcenplanung. Die eigentliche Prozessplanung wird jedoch dezentralisiert und mit der Ausführung und der Prozessüberwachung in einzelnen autonomen Einheiten zusammengefasst. Die Hierarchie bleibt ebenfalls als Ordnungsstruktur erhalten, dient jetzt aber nur noch der Zieldekomposition und der übergreifenden Koordination des Produktionsprozesses. Das Ergebnis dieses agentenbasierenden Steuerungsansatzes ist ein autonomes und kooperatives Steuerungssystem. Die agentenbasierende Steuerung basiert auf folgenden Prinzipien:

- Die Steuerung soll *so dezentral wie möglich und nur so zentral wie nötig ausgeführt*.
- Die Ausführung der Aufgaben soll *so spät wie möglich und nur so früh wie nötig geplant werden*.

Der erste Schritt bei der Konzeption einer agentenorientierten Produktionssteuerung ist die Zuordnung eines Agenten zu jeder physischen Produktionseinheit. Dieser Basisagent verfügt über das lokale Wissen der Einheit und steuert die Produktionsabläufe in dieser Einheit. In Katalinic et al, 2002 wurde ein Elektromotoren Montagesystem mit Hilfe von

agentenorientierten Techniken modelliert. Jeder Montageeinheit wurde ein Agent zugeordnet, der den eigenen Aufgabenbereich steuert, überwacht und optimiert. Die Basisagenten werden nach Aufgabentyp geteilt. Im konkreten Fall sind dies Basisagenten; Transport- Montage-, Qualitätsicherung-, Versorgungs-, Energie- und Reparaturagent. Basisagenten desselben Typs werden zu Gruppen zusammengefasst und einem Bereichsagenten zugeordnet.

Dieser Bereichsagent koordiniert die Zielvorgabe und die Interaktionen mit anderen Bereichen, während die Basisagenten weiterhin über die Produktionsmittel in ihrem Einzugsbereich verfügen. Die Basisagenten verlieren also aufgrund der Aggregation nicht ihre Autonomie, zeigen aber eine hohe Bereitschaft, gemeinsam die Ziele des Bereichs zu erreichen. In diesem Sinne sind die Basisagenten semi-autonom. Die Aggregation wird fortgesetzt, bis das gesamte Montagesystem in einer Hierarchie von Agenten abgebildet ist.

Das Ergebnis des ersten Konzeptionsschrittes ist also der strukturelle Aufbau des agentenbasierenden Steuerungssystems.

Der Aufbau beinhaltet die Identifikation der notwendigen Agenten und ihrer strukturellen Beziehungen untereinander.

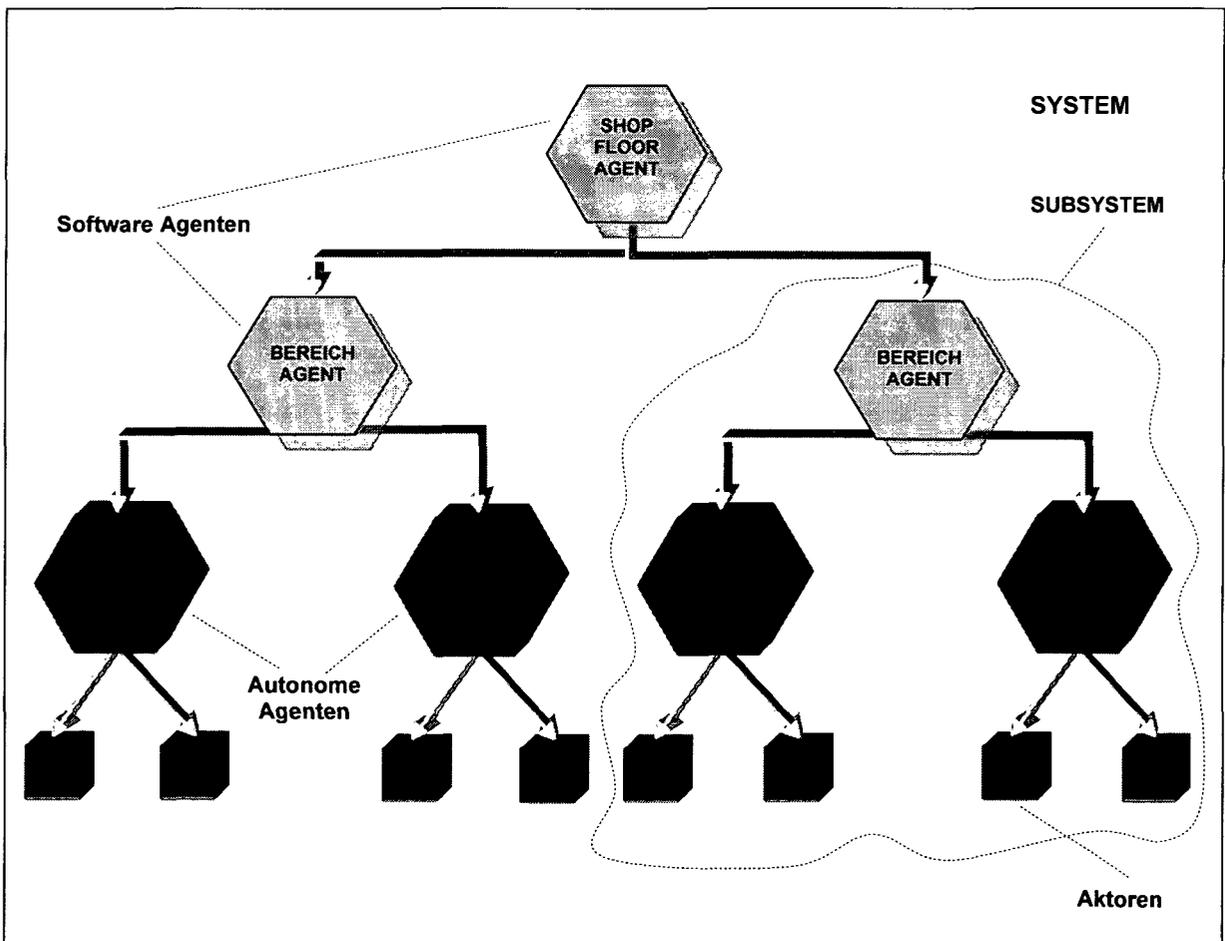


Abb. 6.2. Hierarchie der Agenten.

Der zweite Schritt bei der Konzeption einer Agent basierenden Produktionssteuerung ist die Festlegung der Steuerungsverfahren.

Wichtige Voraussetzung dafür ist deren Kompatibilität mit den vorhandenen Steuerungsverfahren. Dies ist aufgrund des Kostenaufwands für die Umwandlung sowie der traditionellen Produktionsdenkweise erforderlich. Für die Verteilung und Feststellung der Ausführungsreihenfolge sollen klassische, für die jeweilige Aufgabenstellung entwickelte Steuerungs- und Optimierungsverfahren verwendet werden. Während der Ausführung wird der Ablauf zentral überwacht wobei die Steuerungsstrategie bei sich verändernden Prozessbedingungen entsprechend von Basisagenten autonom angepasst wird.

Die agentenbasierenden Steuerungsverfahren müssen drei grundsätzliche Aufgaben erfüllen:

- 1. Die Produktionsziele bzw. die sich aus den Zielen ergebenden Aufgaben werden auf die Basisagenten verteilt.*
- 2. Die Basisagenten steuern ihre Einheit entsprechend der Ziele bzw. Aufgaben.*
- 3. Die Agenten koordinieren die Ausführung der Aufgaben und passen die Vorgaben entsprechend der aktuellen Situation in der Umgebung.*

6.3 Vorteile und Grenzen der Agent basierenden Steuerung

Die Vorteile des agentenorientierten Steuerungsansatzes liegen zum einen in der modularen und strukturtreuen Abbildung des Produktionsprozesses und zum anderen in der Zusammenfassung von Planung und Ausführung begründet. Die strukturtreue Abbildung der Produktionseinheiten durch Basisagenten erhöht die Korrespondenz von Steuerungs- und Produktionssystem und fördert sowohl die Transparenz des Steuerungsprozesses als auch die Übereinstimmung von Planung und realen Gegebenheiten. Gleichermaßen wird die Komplexität des Steuerungsprozesses durch die Agenten und die zum Produktionssystem korrespondierende Steuerungsstruktur beherrschbar gemacht. Die Zusammenfassung von Planung und Ausführung in autonomen Einheiten sichert die Flexibilität und Robustheit des Produktionssystems. Die Agenten entscheiden erst zum letztmöglichen Zeitpunkt über genauere Ausführung des nächsten Schrittes und können so die aktuelle Situation im Produktionsprozess in die Entscheidung einbeziehen. Des Weiteren werden aufgrund der Autonomie die Abhängigkeiten zwischen den Agenten auf ein Minimum beschränkt und eine Abstimmung erfolgt nur dort, wo sie unbedingt notwendig ist. Veränderungen und Störungen verlieren somit an Einfluss, da sie zum alltäglichen Ablauf gehören.

Dem agentenorientierten Steuerungsansatz sind jedoch auch Grenzen gesetzt. Flexibilität setzt Entscheidungsalternativen voraus. Alternativen bestehen jedoch nur dort, wo entweder Ressourcen mit überlappenden Fähigkeiten oder parallele Transportwege vorhanden sind.

Autonomie der Agenten bringt auch neue Anforderungen an sie. Nämlich die Agenten müssen anpassungsfähig sein. Bei der Ausführung von Aufgaben stützt sich das Steuerungssystem eigentlich auf Anpassungsfähigkeit der beteiligten Agenten. Die Anpassungsfähigkeit kann auf reflexartige Aktionen begrenzt sein, wie etwa bei den Ameisen bis zu komplexeren Anpassungen die Intelligenz der Agenten benötigt, wie bei Menschen.

Des Weiteren ist Wandelbarkeit von Steuerungen zwar erforderlich, aber nur dann nutzbar, wenn der physische Agent in gleicher Weise wandelbar ist. Eine Maschine ist zum Beispiel nur dann wirklich wandelbar, wenn die Spannvorrichtung bei gleich bleibender Genauigkeit an unterschiedliche Werkstücke angepasst werden kann.

Grenzen des agentenorientierten Steuerungsansatzes sind durch den mechanischen Aufbau der Agenten und den physischen Aufbau des Produktionssystems gesetzt.

6.4 Klassische versus agentenbasierende Steuerung

Agentenbasiertes Steuerungskonzept bietet einen Ausweg aus der starren, zentralisierten, planbasierten und sequentiell ausgeführten Steuerung. Das agentenbasierte Steuerungskonzept ist dezentralisiert und zusammenwirkend statt sequentiell.

AUFLEGUNG	AGENTENBASIERENDES KONZEPT	KLASSISCHES KONZEPT
Model	Biologie, Ethologie, Ökonomie	Militärisch
Vorteile der klassischen Produktionssteuerung		
Theoretische Optimum	Nein	Ja
Vorhersehbarkeit	Anhäufung	Individuell
Stabilität der Kontrolle	Niedrig	Hoch
Vorteile der Agentenbasierender Produktionssteuerung		
Einstimmung zur Realität	Hoch	Niedrig
Zentrales Datensystem	Nein	Ja
Reaktion auf Störung	Robust	Schwach
Natur der Software	Kurz, einfach	Lang, komplex
Re-Konfiguration des Steuerungssystem	Einfach	Schwer
Scheduling Zeit	Reale Zeit	Zeit Verzögerung

Tab. 6.2. Gegenüberstellung von Agent basierende und Klassische Produktionssteuerung.

Agentenbasierende Steuerung ersetzt zentrale Datenbank und Leitreechner mit einem Netzwerk von Agenten, die mit dem Bild des aktuellen Zustandes der eigenen Umgebung ausgestattet sind und beauftragt autonom auf Veränderungen zu reagieren.

Agentenbasierende Steuerung ist nicht auf Kreislauf zwischen Planung und Exekution angewiesen sondern Ablaufplan bildet sich durch autonome Entscheidungen jedes Agenten. Damit ist Zeitverzögerung aufgrund Planung aufgehoben.

Agentenbasierende Steuerung ist von biologischen und ökonomischen Ansätzen inspiriert. Im Gegenteil zu konventionellen Konzepten sind hierarchische Strukturen überflüssig.

Agentenbasierende Konzepte könnten weder ein theoretisches Optimum garantieren noch die Leistung der Agentensysteme definieren lassen. Das Auftreten von instabilen nichtlinearen Zuständen ist hier die Regel und lineare einfache Phänomene die Ausnahme. Mittels traditionellen mathematischen Instrumentariums können instabile, komplexe, dynamische Systeme nicht befriedigend beschrieben werden. Deswegen findet man diese Punkte als größten Nachteil der Agentenbasierende Produktion. Gleichzeitig ist aber das theoretische Optimum der konventionellen Produktionskonzepte meistens auch nicht in der Praxis erreichbar. Trotzdem entwickelt man aufgrund von Berechnungsmöglichkeiten sowie langjährigen Praxis ein gewisses Vertrauen in solche Voraussagen.

6.5 Passende Domäne für agentenbasierende Produktion

Ranze und Müller motivieren drei notwendige Kriterien, die ein Problembereich aufweisen sollte, damit der Einsatz von agentenbasierender Produktion adäquat ist (Ranze und Müller, 1997). Diese sind durch folgende Charakteristika gegeben:

- Natürliche Verteiltheit, also das Wiederfinden der verteilten Struktur in der Anwendungsdomäne und nicht eine künstlich erzeugte Verteilung einer zentralen Struktur.
- Dynamische Umgebung, also eine Domäne, in der strukturelle Änderungen berücksichtigt werden müssen und in der flexibel auf Änderungen reagiert werden muss,
- Komplexe Interaktion, eine komplexe Koordination zwischen den einzelnen Agenten.

Der Hauptvorteil agentenbasierende Produktion ist Selbstorganisation.

SELBSTORGANISATION: *Selbstorganisation beschreibt den Vorgang, durch den ein System eine räumliche, zeitliche, funktionale oder organisatorische Struktur aufbaut, ohne dass von außen in die inneren Systemabläufe eingegriffen wird.*

Die Analyse der Selbstorganisation der biologischen Systemen wie soziale Insekten Kolonien zeigte, dass physische Systeme zur Selbstorganisation fähig sind, nur wenn alle der folgenden Voraussetzungen erfüllen sind:

1. Das Fehlen einer aufgezwungenen Zentralsteuerung.
2. Die autonome Natur der Agenten.
3. Die hochgradige Vernetzung der Agenten.
4. Natürliche parallele Distribution der Abläufe.

Selbstorganisierende Prozesse können sich nur in offenen Systemen entfalten, d.h. von außen wird dem System ständig Material, Energie und Information zugeführt, um den Prozess aufrecht zu erhalten. Der Einfluss der Umgebung auf das System wird durch Kontrollparameter beschrieben. Sie legen die Rahmenbedingungen fest, unter denen der Prozess abläuft.

Diese Analyse zeigt dass Selbstorganisation nicht nur mittels einer Umwandlung des Steuerungssystems erreicht werden kann. Aus der Sicht der Selbstorganisation erhöht die Umwandlung des Steuerungssystems den Dezentralisierungsgrad. Damit ist Raum für Autonomie und Vernetzung der Agenten geschaffen.

Weiterer Aspekt offener Systeme ist die Fähigkeit sich zu rekonfigurieren, da Änderungen jeglicher Art zu normalen Ablauf gehören. Dabei muss man die Kompatibilität der Schnittstellen berücksichtigen. Schnittstelle eines Agenten ist eigenes Kommunikationssystem. Durch Kommunikationssystem vermittelt jeder Agent eigene Kompetenzen und Fähigkeiten und wird nach Bedarf in der Struktur ein- oder ausgeschlossen. Abbildung 6.3 zeigt Aspekte der Rekonfiguration.

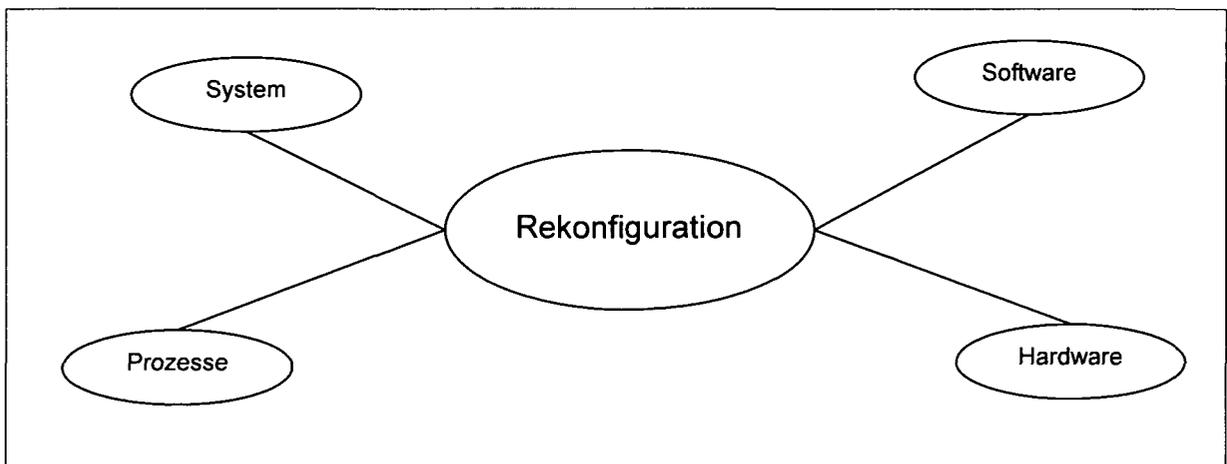


Abb. 6.3. Aspekte der Rekonfiguration.

Von klassischer bis selbstorganisierender Montage

Im diesem Kapitel wird die Transformation von klassischer Fliessband- zu selbstorganisierende Montage dargestellt. Zur Erläuterung dieses Vorgangs ein Beispiel: eines der modernsten vorhandenen Montagesysteme, das Montagesystem von IMS Group der TU Wien.

7.1 Besonderheiten des Montagesystems

Die Teilanfertigung eines Produkts ist vom Endprodukt unabhängig (Katalinic, 1999). Das heißt dass auf einer Maschine die Teile für wesentlich verschiedene Endprodukte gefertigt werden können. In dieser Art und Weise sind Fertigungswerkzeuge „universal“. Das ist der Hauptgrund, weshalb die Fertigung der Teile einen hohen Stand bei der Automatisierung der Prozesse erreicht hat.

Die Montage dagegen hängt vom Endprodukt ab. Es ist nicht möglich eine Montageanlage für grundsätzlich verschiedene Produkte zu benutzen (Katalinic, 1999). Aufgrund dessen und des Zusammenwirken vieler und unterschiedlicher Aufgaben innerhalb des Montageprozesses selbst, kann ein weitaus niedrigerer Automatisierungs- und Flexibilitätsgrad erreicht werden. Der hohe Automatisierungsgrad der Teilanfertigung hat die Montage in vielen Bereichen der Produktion zum teuersten Prozess werden lassen. Aus wirtschaftlichen Gründen wird die Montage bei den einzelnen oder niedrigeren Losgrößen sowie sehr komplexen Produkten häufig manuell durchgeführt.

Im Maschinenbau beträgt der Montagezeitanteil bis 25%, im Fahrzeugbau bis zu 50% und in der Elektro- und Feinwerktechnik bis zu 70% (Lotter, 1992).

Der Automatisierungsgrad der Montage im Maschinenbau ist wegen der kleinen Losgrößen sehr gering. Im Fahrzeugbau ist er im Baugruppen- und Aggregatbereich ein hoher und in der Endmontage aufgrund Vielfalt des Typs der Produkte und der uvorhersagbarer Marktnachfrage gering. Der höchste Automatisierungsgrad ist in der Komponentenmontage

der Elektrogeräte zu finden. Die geringe Komplexität und die große Produktionsstückzahl dieser Komponenten sind der Grund hierfür. Die Endmontage der Elektrogeräte wird jedoch wegen der großen Variantenzahl, im großen Umfang noch manuell oder teilweise automatisiert.

Dieser Zustand zeigt deutlich, dass in der Montage erhebliche Rationalisierungspotentiale vorhanden sind.

Außerdem, betreffen die, in der Einführung des Kapitels, genannten Anforderungen an Produktionssysteme, in besonderem Maße die Montage, die als letztes Glied in der Produktionskette die vorhandenen Turbulenzen weitestgehend ungefiltert bewältigen muss (Westkämper et al, 2001). Genau da kommen die Vorteile von Agententechnologien voll zum Tragen.

7.2 Die Montageaufgabe

Montageaufgabe besteht darin den Komponentenfluss so aufeinander abzustimmen, dass er durch die Montageschritte im Produkt zusammenwachsen kann. Die richtigen Teile, im richtigen Zustand, in der richtigen Menge, zur richtigen Zeit und am richtigen Ort bereitzustellen, ist die Grundvoraussetzung, für einen erfolgreichen Montageabschluss. Nach jedem Montageschritt entsteht eine neue Baugruppe, die als Basiskomponente für die nächsten Montageschritte dient. Ganzer Vorgang kann mittels einer Produkthierarchie beschreiben werden. Auf der Spitze der Produkthierarchie befindet sich das Produkt und auf dem Fuß die Basiskomponenten (Abbildung 7.1). Erzeugung der Produkthierarchie ist äquivalent zur Planung in der Fertigung.

Die Baugruppebezeichnungen zeigen aus welchen Basis Komponenten sie aufgebaut sind. Die Baugruppe B_{67} ist aus Basis Komponente B_6 und B_7 aufgebaut. Während Kreise in der Produkthierarchie Basis Komponenten oder darauf aufgebaute Baugruppen bezeichnen, bezeichnen Rechtecke Montageschritte. Basis Komponente B_5 und Baugruppe B_{67} sind durch Montageschritt S_4 zusammengebracht. Resultat des Montageschrittes S_4 ist Baugruppe B_{567} . Die Produkthierarchie definiert die zeitliche Reihenfolge der Montage. Zum Beispiel, Abbildung 7.1 zeigt dass Montageschritte S_1 , S_3 und S_5 genau auf eine Weise durchgeführt werden müssen. Dagegen gibt es keine direkte Verbindung zwischen Montageschritt S_1 und S_2 .

Produkthierarchie hängt von zwei Aspekten ab; von der Natur des Produkts und der Montageausrüstung. Ein extremer Fall besteht wenn alle Montageschritte auf einer Montagestation durchgeführt werden und wenn dabei die Montagereihenfolge beliebig definiert werden kann. Ein Anderer extremer Fall besteht wenn es nur eine Montagereihenfolge gibt, die zum Produkt führt.

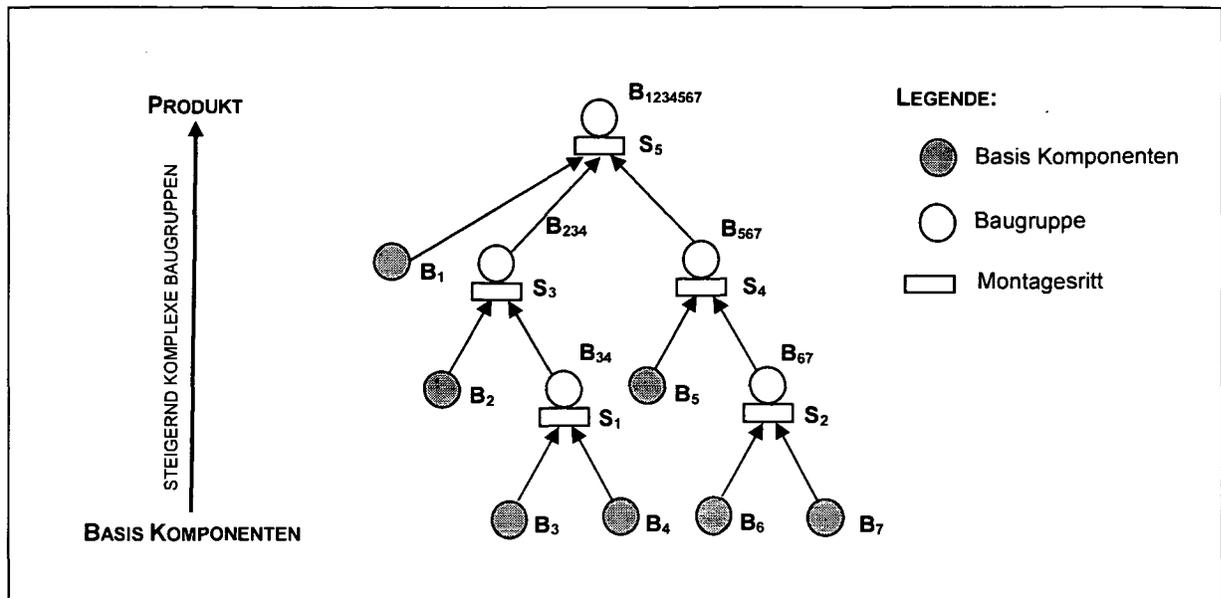


Abb. 7.1. Produkt Hierarchie für den generellen Fall der Montage.

7.3 Auswahl der Montageorganisation

Der Preis und die Qualität sind nach wie vor die Hauptkriterien für den Kauf eines Produktes. Daher ist eine kostengünstigere Montage auf hohem Qualitätsniveau die Grundvoraussetzung für ein marktfähiges Produkt. Für die optimale Erfüllung der Montageaufgaben ist die Wahl der Montageorganisation von entscheidender Bedeutung. Es handelt sich um die Alternativen in der Art der organisatorischen Charakterisierung der Montage nach der räumlichen und zeitlichen Anordnung sowie der Montagedurchführung. Es ist genau zu überlegen, welche Kombination von Mensch, Maschine und deren räumlich - zeitlichen Anordnung die Anforderungen des Marktes am besten erfüllt. Die Auswahl der Montageorganisation richtet sich im Wesentlichen nach folgenden Kriterien:

- Produktionsvolumen und Lösgrößen
- Variantenzahl
- Komplexität des Produkts
- Schwierigkeitsgrad der Montage
- Größe des Produktes

Montagedurchführungsmöglichkeiten sind von Boothroyd systematisch untersucht (Redford & Lo, 1992). In einer Studie betrachtete Boothroyd die Kosten der Einzel- und Mehrstationenmontage, der manuellen Montage und Montage mit den speziellen Maschinen für die unterschiedliche Anzahl von Teilen im Produkt und verschiedenen Gesamt-

Produktionsvolumen, die innerhalb eines Jahres bei Zweischichtarbeit hergestellt sein sollten. Die Ergebnisse dieser Studie sind in Abbildung 7.2 gezeigt.

Bei der Findung der geeigneten Montageorganisation hinsichtlich der räumlich-zeitlichen Anordnung kann Tabelle 7.1 als Richtlinie dienen. Tabelle 7.1 erfasst grundsätzliche Montageorganisationen, die weiter auf zahlreiche spezielle technische Subvarianten geteilt werden können. Sie sind aber für den weiteren Ablauf der Dissertation wenig von Bedeutung und werden deshalb ausgeschlossen.

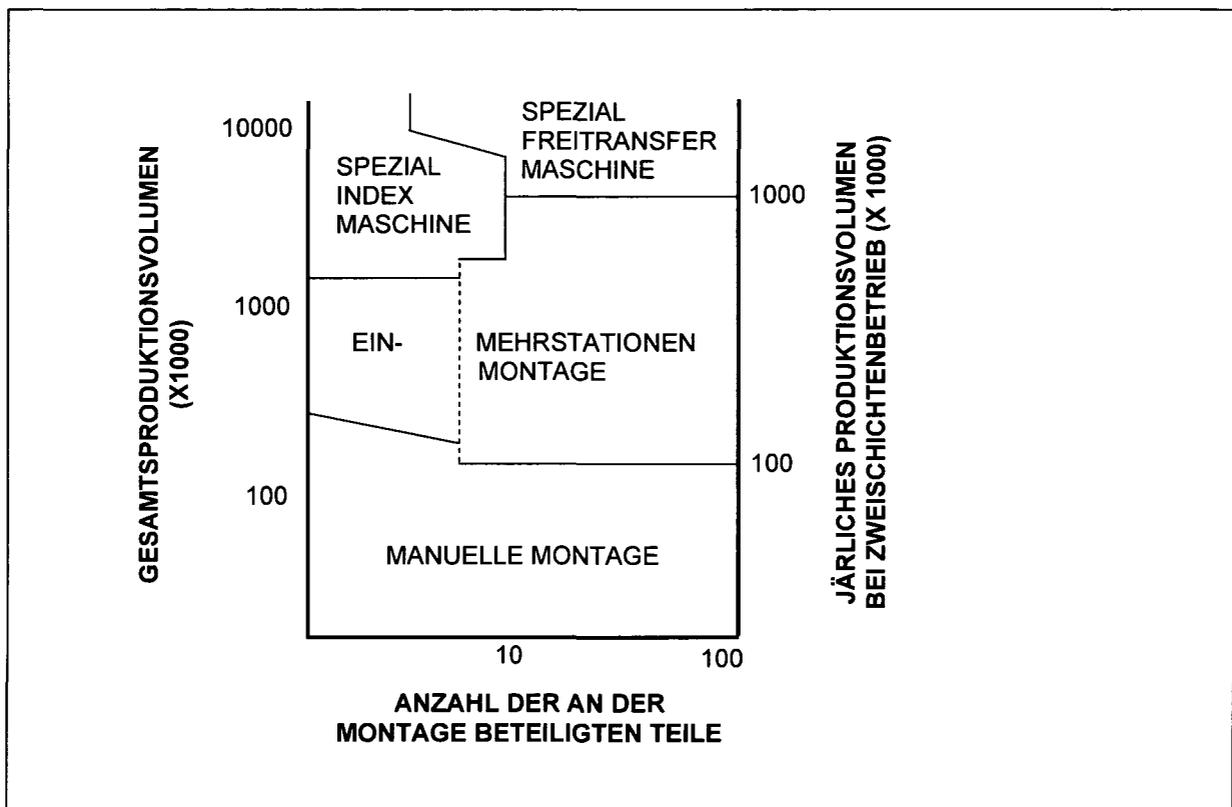


Abb. 7.2. Bedingungen für die ökonomische Anwendung der verschiedenen Montagesysteme (Boothroyd, 1984).

MONTAGE ORGANISATION	ANWENDUNG	MERKMALE
Einzelplatzmontage	Ortsfester Montageplatz, kleine Erzeugnisse	<ul style="list-style-type: none"> • Eine oder mehrere Montagepersonen montieren eine Einheit • Alle Einzelteile und Werkzeuge werden an dem Platz bereitgestellt • Bei größeren Stückzahlen werden sehr spezialisierte Einzelmontageplätze benötigt
Gruppenmontage	Mittel- und Grosserien	<ul style="list-style-type: none"> • Die Montageeinhalte sind in fach- oder ablaufspezifische Teilabschnitte geteilt, die von Arbeitsgruppen durchgeführt werden • Nacheinander werden verschiedene Erzeugnistypen montiert
Reihenmontage	Klein- und Mittelserie	<ul style="list-style-type: none"> • Materialfluss wird nicht zeitlich festgelegt • Keine Vorgabe einer Taktzeit • Spezialisierung von Personal und Montageplatz
Gleitende Reihenmontage	Objekt wird zwischen mehreren Stationen in Reihenfolge des Montagefortschritts transportiert	<ul style="list-style-type: none"> • Personal hat spezielle Montagemittel, die mitgeführt werden • Montageplatz ist für einen gegebenen Teilprozess speziell ausgerüstet • Personal kann mit dem Objekt gleiten
Stationäre Reihenmontage	Mehrere gleiche Montageplätze, an denen die Montageeinheit komplett zusammengebaut wird	<ul style="list-style-type: none"> • Jede Person führt an allen Montageeinheiten den gleichen Teilprozess durch • Montageplätze sind universell ausgerüstet • Die benötigten Bauteile werden an jedem Montageplatz bereitgestellt
Fliessmontage	Mittel- und Großserien	
Kontinuierliche Fliessmontage	Montageeinheit wird während des Montagevorgangs ununterbrochen bewegt	<ul style="list-style-type: none"> • Zeitlich festgelegter Materialfluss • Personal oder Montagemittel bewegen sich synchron zur Montageeinheit • Operationen müssen innerhalb einer vorgegebenen Zeitspanne ausgeführt werden
Stationäre Fliessmontage	Montageeinheiten wird für die Zeit des Montagevorgangs vom Fördermittel getrennt	<ul style="list-style-type: none"> • Erforderlich bei höheren Genauigkeitsanforderungen • Bei schweren Werkzeugen oder unbeweglichen Montagemitteln
Taktstrassenmontage	Montageeinheit während des Vorgangs stationär Weitertransport im gesamten System in starren, zeitgleichen Schritten	<ul style="list-style-type: none"> • Bei Störungen kommt die gesamte Linie zum Stillstand, sofern keine Maßnahmen zur Pufferung vorgesehen sind

Tab. 7.1. Montageorganisationen, Anwendung und Hauptmerkmale (Schulte, 2000).

7.3.1 Fließmontagekonzept

Die Fließmontage ist für die Produktion großer Mengen einer Produktart über einen längeren Zeitraum konzipiert worden. Eine solche Anlage besteht aus einer linearen Anordnung von Montagestationen, die jeweils einen Montageschritt an eine Baugruppe vornehmen können. Wichtiges Kriterium der Fließmontage ist die Orientierung am Arbeitsgegenstand. Die Betriebsmittel werden in der durch den Arbeitsablauf vorgegebenen Reihenfolge aufgestellt. Arbeitsmittel und Arbeitskräfte sind ortsgebunden. Hingegen „fließt“ der Arbeitsgegenstand gemäß der Arbeitsablauffolge von Arbeitsplatz zu Arbeitsplatz. Der zeitliche Ablauf ist kontinuierlich (Fließband) oder in Zeitintervallen (Getaktete Fließband). Aufgrund der straffen Organisation der Montage sind meist keine Zwischenpuffer vorhanden. Trotzdem können sie besonders bei Montage komplexerer Produkte und wegen im getaktetem Fall Zwischenpuffern erheblich die Zuverlässigkeit und Funktionalität des Montagesystem erhöhen.

Der Vorteil der Fließbandmontage besteht im übersichtlichen Materialfluss, leicht erkennbaren Montagenverlauf, kurzen Durchlaufzeiten, keine beziehungsweise geringen Beständen, kurze Transportwegen, einfacher Steuerung und modularen Automatisierung der Montageschritte. Deswegen ist das Fließmontagekonzept sehr beliebt und wird häufig eingesetzt. Der Ausgangspunkt für die Weiterentwicklung der Montageanlagen ist damit vorausgesetzt.

7.4 Montagesystem der IMS Group der TU Wien

Ein von den Hauptrichtungen in der Forschung und Entwicklung von Montagesystemen ist der Erzeugung von flexibleren und universaleren Montagesystemen gewidmet. Unter Flexibilität des Montagesystems versteht man die Fähigkeit einer Montageanlage, auf äußere Änderungen und innere Störungen ohne wesentliche Änderung der Kostenrelationen und / oder Stückkosten zu reagieren. Unter äußeren Änderungen versteht man die Einführung neuer Produkte, Kapazitäts- und Typenmix- und Lösgrößenschwankungen. Innere Störungen beziehen sich auf alle nicht deterministische Prozesse, die auf die Montage Einfluss haben, d.h. Montagestationen und Transportsystem Ausfälle, Montage von fehlerhaften Elementen usw.

Das Projekt, aus dem diese Dissertation hervorgeht, entstand aus dem realen industriellen Bedarf nach einer signifikanten Reduktion der Kosten für Elektromotoren in der Massenproduktion und steigernden Vielfalt der Elektromotorbauarten zu senken (Katalinic, 1999 a).

Um Komplexität des Problems zu bewältigen wird dieses Ziel auf zwei Ebenen verfolgt (Katalinic, 1999 b):

1. Montagesystem Niveau
2. Niveau von Montagesystem Einheiten, eben immer in Hinblick auf das Ganze

Ziel des ersten Schrittes des Projektes war die Entwicklung und Umsetzung eines Montagesystems das folgenden Anforderungen genügt:

- Qualitätsverbesserung des Produktes.
- Hohe Verfügbarkeit und Produktivität der Gesamtanlage sowie der einzelnen Komponenten
- Verbesserung der Montagesteuerung
- Schnellere Reaktionsmöglichkeiten auf Bedarfsschwankungen, hohe Produktzahl- und Varianten Vielfalt
- Verbesserung der Arbeitsgestaltung (Ergonomie)
- Automatische Produktzahlerfassung
- Modularer Aufbau der Montageanlage, die ein stufenartiger Aufbau erlaubt
- Just-in-time Organisation der Produktion.

Haupt Merkmale der existierenden Elektromotorproduktion sind:

- Variantenzahl: 21 Bauarten und Varianten.
- Losgrößen: von 50 bis par tausend Elektromotoren pro Tag
- Tägliche Produktion: 7000 im zwei Schichten
- Montagestationen: Zahl und Typ der Montagestationen sowie deren Position ist nicht Fix. Umstrukturierung wird Manual ausgeführt.
- Um den Preis der Montageteile, die auf dem Markt gekauft werden, zu senken, musste auf 100% Qualitätsgarantie der Teile verzichten werden.
- Durchschnittliche Produktlebenszeit: zwei Jahre mit weiter sinkender Tendenz.

Abbildung 7.3 repräsentiert das industriell eingesetzte Montagesystem (Katalinic, 1999, 2001). Es handelt sich um ein kontinuierliches Fliessmontagekonzept. Das Transportsystem besteht aus dem Fliessband, das die Palette durch das Montagesystem transportiert. Die Palette dient als Transportmittel, Montageplatz sowie als Montagebezogener Datenträger. Jede Palette hat eine Integrierte Schaltung, die als Datenspeicher für Daten über einen Montagezyklus dient. Diese Daten kann jede Station lesen und entsprechend modifizieren.

Diese stellen den Schlüssel für die Realisierung einer flexibleren Montage dar. Diese Daten sind:

Palettentyp: Um die Montageteile mit Paletten transportieren zu können und die Montage sehr unterschiedlicher Motorenbaugruppen zu erleichtern, müssen diese mit unterschiedlichen Zentrierelementen bestückt sein. Dieser Rüstzustand wird als Palettentyp bezeichnet. Dadurch ist es festgelegt wo genau das hergebrachte Teil auf der Palette platziert werden soll.

Motortyp: Diese Information ermöglicht jeder Montagestation die Erkennung der spezifisch benötigten Montageteile. Diese werden sodann auf die Palette gebracht.

Montagestatus: Diese Information beinhaltet alle Montageschritte, die nach der Montagereihenfolge geordnet sind. Montagestatus verändert sich entsprechend von einer leeren Palette bis zu einer mit gefertigtem Motor. Aufgrund dieser Information kann jede Montagestation erkennen ob sie für diese Palette verantwortlich ist.

Qualitätsstatus: Nach jedem Montageschritt wird die Kontrolle des ausgeführten Montageschrittes durchgeführt. Dies wird durch Qualitätsstaus repräsentiert. Wenn die Qualität des erfolgten Montageschrittes gut ist, ist er positiv. Montagestatus in dem Palettenspeicher wird um eins erhöht. Wenn aber aktuellen Montageschritt nicht erfolgreich war, ist der Qualitätsstatus negativ. Montagestaus wird auf Reparaturstatus gewechselt.

Jede Station hat eigene Ablaufsteuerung, die aufgrund des Montagestatus, Palette-, und Elektromotortyps die richtige Aktion ausübt.

Neuer Montagezyklus fängt in der Entlade Station an. Dort werden auf die gegebenenfalls leere Palette die Schlüsseldaten für den nächsten Montagezyklus in den Palettenspeicher eingegeben und gespeichert. Wenn Palette Entlade Station betritt, passiert wie im folgenden deskriptiven Algorithmus beschrieben ist:

Start

```

{
lesen   {Montagestatus}
wenn    {Montagestatus gleich letzten Montagestaus ist}
dann    {Nimmt gefertigten Motor, schreib neues Datenset in Palettenspeicher}
sonst   {lass Palette weiter}
}
```

Ende

Der Ausgang jeder Station ist mit einer mechanischen Palettesperre versehen. Diese wird aufgehoben wenn Station für die Palette nicht zuständig ist oder sobald Station neuen Montagestatus auf dem Palettespeicher geschrieben hat.

Wann Palette Arbeitsbereich der Montagestation betritt, wird automatisch Montagealgorithmus ausgeführt. Dies ist im folgenden deskriptiven Algorithmus beschrieben:

Start

```

{
  lesen  {Montagestatus, Palettetyp, Motortyp, }
  wenn   {Montagestatus entspricht Montageschritt, das gerade da ausgeführt sein muss}
  dann   {Montageschritt ausführen, Qualitätstest des Schrittes ausführen, neue Montagestatus auf dem Palettespeicher schreiben, lass Palette weiter}
  sonst  {lass Palette weiter}
}

```

Ende

Die Montagestationen wurden so geplant und gebaut, dass sie ständig die Teile für zwei verschiedene Elektromotorenbauarten behalten können. Was für welche dies sind hängt von der Planung ab, die auf Stunden, Schicht oder täglicher Basis ausgemacht ist. Um Umplanungen zu unterstützen, ist das dritte Komponentenlager vorhanden, dessen Kapazität nur für niedrigere Losgrößen genügt. Bei den größeren Losgrößen müssen die Teile von größeren Lagern herangezogen werden. In der Übergangszeit werden drei Elektromotorenbauarten gefertigt. Dadurch ist es gewährleistet, dass mit der Rüstzeit 0 gefahren und die Losgröße 1 erreicht werden kann. Damit erfolgt diese Typenumstellung gleitend.

Wenn ausgeführter Montageschritt nicht erfolgreich war, ist Qualitätsstatus negativ. Dies bedeutet, dass der aktuelle Montagestatus auf Reparaturstatus gewechselt wird. Damit sind alle weiteren geplanten Montageschritte gesperrt und die Palette steuert die Reparaturstation an. Dort wartet die Palette bis der Montagestatus gewechselt wird. Sobald ein neuer Montagestatus existiert wird die Palette in das System wieder integriert und Montage geht weiter je nach aktuellem Montagestatus. In Fall dass es um gröbere Fehler handelt, die nicht repariert werden, wird der fehlerhafte Motor manuell entfernt und die leere Palette wird zurückgesetzt. Sie wird zwar wieder in den Umlaufkreis eingeschleust jedoch wird auf dieser nichts mehr montiert. Erst in der Entladestation bekommt sie neues Datenset und der neue Montagezyklus erfolgt von vorne.

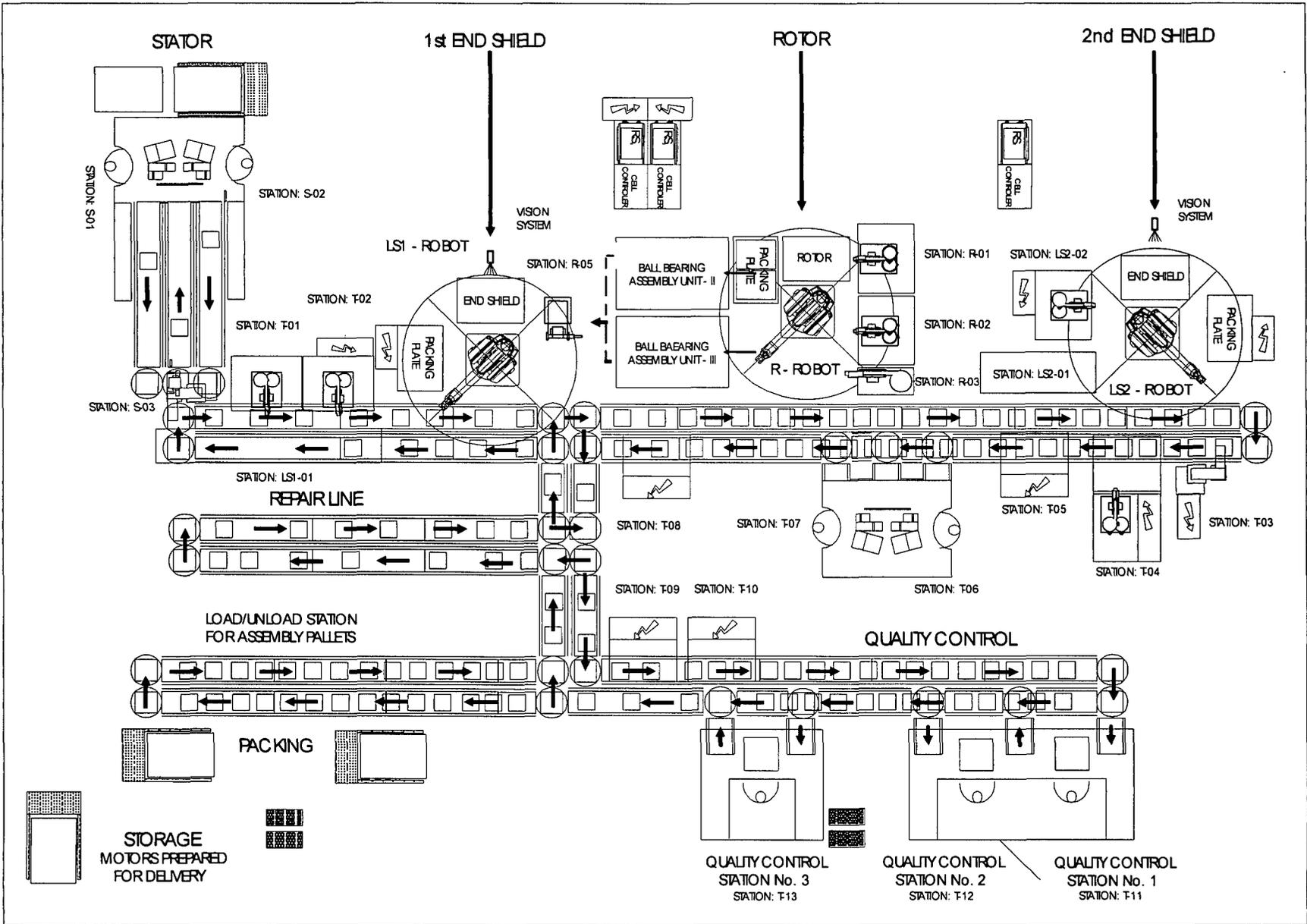


Abb. 7.3. Grundriss des flexiblen Fließbandmontagesystems (Katalinic, 1999).

Arbeitsszenario der Reparaturlinie ist durch folgendem Algorithmus beschrieben:

Start

```

{
  lesen  {Montagestatus}
  wenn   {Montagestatus gleich Reparaturstatus ist}
  dann   {schieb die Palette in der Reparaturlinie und wartet auf Reparaturstation Bediener}
         {Montagestatus gewechselt ist}
         dann   {schreib neuen Montagestatus auf dem Palettespeicher}
         sonst  {warte}
}

```

Ende

7.4.1 Genauer Ablauf in den Montagestationen

Montagestation 1 (LS1): Lagerschild (4) wird auf die Palette gebracht.

Wenn die Palette zum Eingang der Montagestation kommt, liest Scanner der Montagestation folgende Informationen ab: Qualitätsstatus, Montagestatus, Palettentyp und Typ des Motors. Qualitätsstatus kann positiv (zuständig) oder negativ (nicht zuständig) sein. Montagestatus beinhaltet die Information ob diese Station für diese Palette zuständig ist oder nicht. Palettentyp definiert wo auf der Palette das gesuchte Montageelement platziert sein muss. Typ des Motors sagt was für einen Element montiert wird. Lagerschild Behälter hat in jeder Moment zwei Lagerschildtypen. Für jede Kombination existiert eine typdefinierte Agentarm-Bahn, die aufgrund dieser Informationen eindeutig gewählt werden kann. Das Lagerschild wird von dem Agentarm genommen und unter die Kamera gebracht. Vision System der Montagestation kontrolliert Qualität und die Lage des Lagerschildes, das danach auf die Palette gebracht wird. Am Ausgang der Montagestation wird auf dem Palettenspeicher der aktuelle Zustand geschrieben. Die Palette steuert nächste Montagestation an. Dieser Vorgang wiederholt sich für alle voll automatisierte Montagestationen.

Montagestation 2 (S 01 und S 02): Der Stator wird händisch der Palette zugeführt.

Diese Montagestation ist bemannt weil auf ihr zusätzliche Montageschritte gemacht werden müssen, die für heutige Automatisierungssysteme nicht geeignet sind. Alle Kabelenden müssen abisoliert werden, durch die Statoröffnung ins Flachstecker (3a, 3b, 3c, 3d) gebracht werden und dort auf den Stecker geschraubt werden. Außerdem müssen alle Kabel auf Lackreste überprüft werden.

Montagestation 3 (T 01): Ausgleichsscheibe (7) wird montiert.

Montagestation 4 (T 02): Kugellager (8) wird montiert.

Montagestation 5 (R, RO1, RO2 und RO3): Diese Montagestation besteht aus vier Teilen. Stationen RO1, RO2 und RO3 montieren Kugellager (9), Passscheibe (10) und Sicherungsring (11) auf dem Rotor. Agentarm entnimmt typdefiniert gefertigten Rotor und setzt ihn in den Stator.

Montagestation 6 (LS 2): Lagerschild (6) wird auf die Palette montiert.

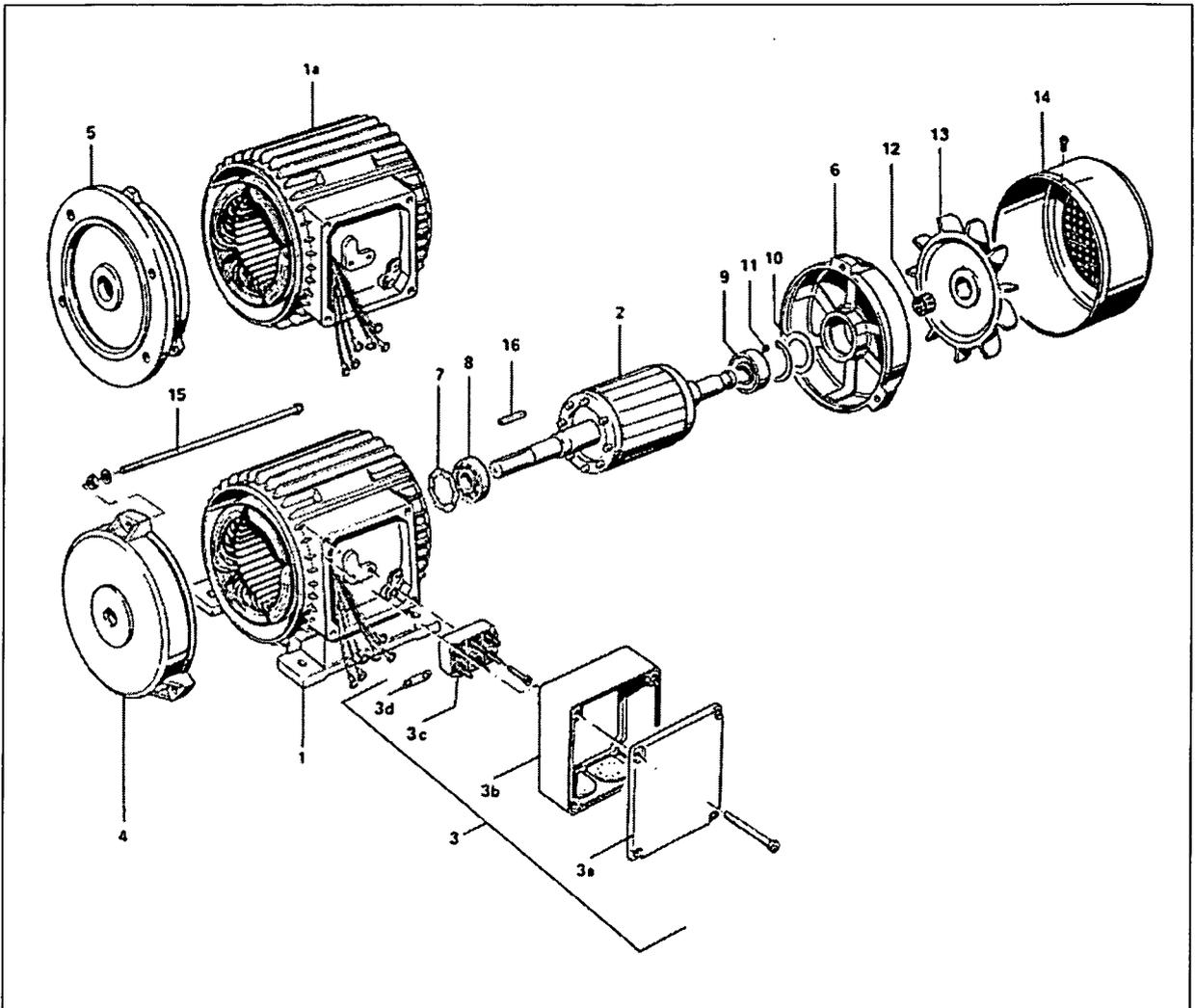


Abb. 7.4. Explosion Darstellung der Elektromotor und Montagereihfolge (ATB Katalog, 2000). Komponenten: 1 Fuß-Stator mit Blechpaket und Wicklung, 1a Flansch-Stator mit Blechpaket und Wicklung, 2 Rotor, 3 Klemmenanordnung, komplett, 3a Deckel, 3b Unterteil, 3c Klemmenplatte, komplett, 3d Verbindungsbrücke, 4 Lagerschild AS – IM B3, 5 Flansch IM B5 oder IM B14, 6 Lagerschild BS, 7 Kugellager-Ausgleichsscheibe, 8 Kugellager AS, 9 Kugellager BS, 10 Passscheibe, 11 Sicherungsring, 12 Toleranzring BS, 13 Ventilator, 14 Schutzhaube, 15 Befestigungsbolzen, 16 Passfeder.

Montagestation 7 (T-03): Die Zuganker werden sortiert und vereinzelt. Zwei getrennte Zuführungen mit einem Handlingsgerät vor dem Einsetzen erfolgt das Benetzen mit Schraubsicherung.

Montagestation 9 (T-08): Ventilatorflügel wird aufgepresst.

Montagestationen 10 (T-09 bis T-13): Mechanische und Elektrische Prüfstationen.

Montagestation 8 (T-04): Motor wird planparallel oder mit beweglicher Druckplatte vorgespannt. Die Zuganker werden verschraubt.

Montagestation 11 (T-05 und T-06): Beide Montagestationen sind bemannt. Dort wird typenabhängiger Kondensator und Ventilatorflügel (13) angesetzt.

7.5 Grenzen des Fließbandmontagekonzepts

Gerade für die Fertigung von Großserien ist dieses Montage- und Materialflusssystem vorteilhaft, besonders im Hinblick auf die einfache Steuerung, zumindest solange ein und derselbe Produkttyp montiert wird und konstante Ausbringung erwartet wird. Würden alle Montagestationen hundertprozentig arbeiten, gäbe es in der Tat kaum Bedarf, von diesem Prinzip abzuweichen. Obwohl die technische Verfügbarkeit der einzelnen Montagestationen sehr hoch ist (deutlich über 90%), führt die lineare Anordnung zusammen mit dem getakteten Betrieb im Fehlerfall zu einer deutlich geringeren Verfügbarkeit der gesamten Anlage. Sobald nur eine Montagestation eines Fließbands ausfällt, steht sofort die gesamte Montageanlage still. Im Fall, dass Puffern vorhanden sind, läuft einerseits der Puffer vor dieser Montagestation voll, so dass die vorgelagerte Montagestation nicht weiterarbeiten kann. Andererseits werden nachfolgende Montagestationen nicht mehr versorgt, so dass auch sie nicht mehr weiter montieren können, wenn ihre Puffer leer gelaufen sind.

Empirisch werden diese Produktionseinbrüche sichtbar in einem so genannten k-Faktor. Dieser würde im fehlerfreien Fall per Definition 1 sein, liegt aber bei realen Anlagen in der Größenordnung von 0,65 mit deutlichen Schwankungen um diesen Wert (Sundermeyer et al, 2001). Da die technische Verfügbarkeit der Montagestationen nahezu ausgeschöpft ist, könnte man zur Verbesserung dieser Situation an Fließbandanlagen allenfalls die Größe der Puffer nach jeder Montagestation eingesetzt und dessen Kapazität erhöhen. Theoretische Analysen zeigen jedoch, dass extrem große Puffer nötig wären, um den k-Faktor signifikant zu heben (Bussmann, 2000). Dies ist wegen der höheren Kosten des nötigen Puffer Platzes und des im System gebundenen Materials nicht vertretbar.

Ein weiterer Nachteil der klassischen starr verketteten Fließbandmontageanlagen liegt darin, dass sie umprogrammiert werden müssen, sobald eine neue Produktvariante auf der

Anlage montiert werden soll. Außerdem gibt es keine Montageanlage, die ohne erhebliche mechanische Eingriffe und Produktionsstillstand auf Aufträge reagieren kann, die nicht von Anfang an vorgesehen sind. Weil eine Umstellung oft nur schwer möglich ist, bleibt das Produktionsprogramm relativ starr, so dass nur ein seltener oder überhaupt kein Wechsel der Erzeugnisse stattfinden kann.

Bei der gleichzeitigen Montage verschiedener Produkte muss der Takt des Fließbandes so eingerichtet werden, dass das komplizierteste Produkt montiert werden kann. Optimierung der Montage für solchen Fall benötigt Kompromisse, die Leistung des Fließbandes sehr beschränken können.

In Fall der Parallelen Montage mehrere Motorentypen beziehungsweise fehlerhaft montierten Baugruppen entsteht die Forderung Paletten umzuleiten. Jedoch eine Palette umzuleiten, ohne in der Lage zu sein, Montagestationen zu umgehen, ist nicht möglich; vor allem mit einer Anzahl von anderen Paletten in der Warteschlange vor der Montagestationen. In dem vorliegenden Montagesystem ist dieses Problem durch die Einführung zusätzlicher Fließbändern auf den kritischen Stellen umgegangen. Eine rückwärts gerichtete Schleife sollte auf diese Art die Paletten mit fehlerhaftmontierten Baugruppen schneller aus dem Hauptfließbandsystem ausschleusen. Dennoch ist das Wachstum der Komplexität des Systems, wegen dadurch entstandenen Ausnahmezuständen, nichts zu vermeiden.

Aus den oben genannten Gründen ist einsichtig, dass zur Überwindung der Schwächen klassischer Fließbandmontagekonzepte und beim gleichzeitigen Bedarf die Flexibilität, Robustheit und Wiederverwendbarkeit des Montagesystems erheblich zu erweitern ein grundsätzlich andersartiges Layout der Produktionsanlage konzipiert werden muss.

7.6 Motivation für die selbstorganisierende Montage

Aus der Sicht der Flexibilität und Robustheit der Fließbandmontage entstehen die meisten Probleme genau angesichts dieses Fließbandkonzepts von selbst. Daraus entstand die Idee Fließband abzulösen und Transport sowie Montageplatz mit fahrerlosen frei fahrenden Transportern zu realisieren. Dieses Konzept ist auf der Abbildung 7.5 repräsentiert.

Es besteht aus ort gebundenen Montage- und Qualitätsstationen, Reparatur-, Lade- und Entladestation die einen geschlossenen Kreis abbilden. Der Kreis bildet eine Physische- sowie eine Systemgrenze. Innerhalb des Kreises befinden sich fahrerlose frei fahrende Transporter. Frei fahrende Transporter müssen in der Lage sein, entsprechende Montagestation zu finden und zu ihr hinzusteuern sowie Änderungen in ihrer Umgebung zu erfassen und angepasst darauf zu reagieren. Es erfordert einen zielgerechtes Verhalten und zugleich einen hohen Grad an Adaptivität und Autonomie des Transporters.

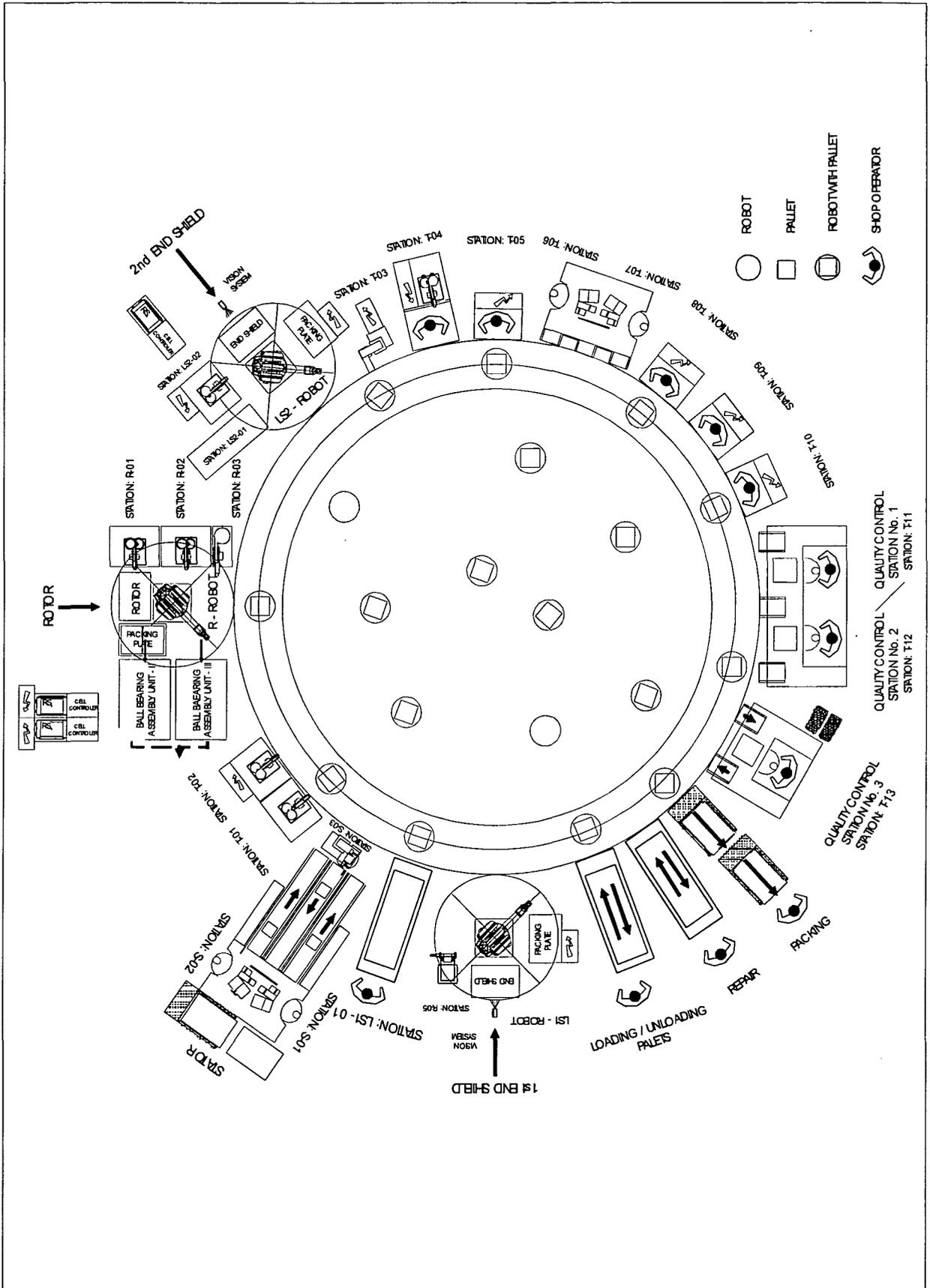


Abb. 7.5. Die erste Stufe der Entwicklung des selbstorganisierenden Montagekonzepts (Katalinic, 1999).

Obwohl man versucht vorhandene Fertigungstransporteinheiten störungstoleranter zu machen, gehen genannte Anforderungen an Transportern weit hinaus über diese Versuche. Sie müssen frei beweglich sein und nicht wie zum Beispiel beim AGV schinengebunden. Störungen müssen als Bestandteil und nicht als spezieller Zustand der Abläufe der frei fahrenden Transportern sein. Es ist eine Anforderung an autonome Agenten. Autonome Agenten die genannten Anforderungen genügen existieren nicht. Daher ist für die Verwirklichung der vorgeschlagenen Umwandlung Entwurf und Bau von autonomen Agenten, eine notwendige Voraussetzung. Solche autonomen Agenten werden im weiteren Transportagenten genannt. Die Motivation zum Bau von Transportagenten ergibt sich aus Vorteilen, die diese Umwandlung bringt.

Hauptvorteile sind:

- *Bahn Optimierung:* Abhängig von der Montageaufgabe und Status der Montagestationen, kann jeder Transportagent eigene Bahn arrangieren. Montageteile, Montagegruppe und Werkzeuge können vollkommen frei von einem zum anderen Punkt des Montagesystems gebracht werden. Im Falle eines Montagestationsausfalles liefert diese Art der Redundanz dem System die Flexibilität, einer Transportagenten zu einer anderen Montagestation umzuleiten.
- *Glatte Einführung neuer Produkte:* Solange Montagestationen erforderliche Montageschritte ausführen können, ist physischer Umbau der Montagestationen nicht erforderlich. Man kann neue Montagestationen auf die Systemgrenze einführen. Freie Transportagenten könnten online mit neuen Montagereihenfolgen nachgerüstet werden, ohne die Leistung der gegenwärtigen Produktion zu benachteiligen.
- *Gleichzeitige Montage verschiedener Produkte:* Das Problem der Fließbandbalancierung ist aufgehoben. Parallelmontage unterschiedlicher Motoren hat geringen Einfluss auf die Kapazität des Montagesystems.
- *Robustheit:* Das System ist wesentlich robuster, da durch Ausfall eines Transportagenten nicht die ganze Montage in Frage gestellt wird. Beim Ausfall einer Montagestation kann auf einem anderen Platz auf die Systemgrenze andere Montagestation mit gleichen Eigenschaften eingeführt werden.

Wenn aber Montagestationen auch frei beweglich sind, kann sich das Montagesystem bedarfsgemäß selbst organisieren. Das Hauptmerkmal von selbstorganisierendem Montagesystem ist seine Fähigkeit, sich selbständig in vielen Weisen einzustellen, z.B. hinzufügen und/oder entfernen von mechanischen, elektronischen, und Software Modulen,

Werkzeugen usw. Das Montagesystem kann ohne mechanische Eingriffe auf neue Montageaufgaben umgestellt werden. Damit ist das Montagesystem in der Lage, reibungslos einen neuen Mix von Produktvarianten zu verarbeiten. Außerdem ist es möglich die Anlage bedarfsgerecht abzuändern, ohne dass an der Software und Hardware Änderungen vorgenommen werden müssen. Auf diese Weise sind die Voraussetzungen für die Modularitäts-, Leistungs- sowie Flexibilitätssteigerung geschaffen.

Das war das Leitmotiv bei der Weiterentwicklung des Montagekonzepts. Entstandenes Montagesystem ist ein modulares System, das aus eine Menge frei beweglicher autonomer Agenten ausgebildet ist. Abbildung 7.6 verdeutlichter das Konzept.

Das Montagesystem besteht aus zwei Subsystemen: Kernsystem (Core System) und Ergänzendes Subsystem (Supplementary System). Diese sind durch die Systemgrenze geteilt. Kernsystem ist Zentralteil des selbstorganisierenden Montagesystems. Aktivitäten im Kernsystem sind: Montieren, Qualitätskontrolle, Montageteileversorgung, Reparatur und Energieerhaltung des Agenten. Gemäß dieser Aktivitätsverteilung sind spezielle Agenten vorgesehen. Es gibt also Transport-, Montage-, Qualitäts-, Versorgungs-, und Energieagenten und Reparatur- Prüfungs- und Entlade Station. Alle Agenten eines Typs bilden eine Agentenklasse. Tabelle 7.2 fasst die Funktion und Auflegung den Agentenklassen zusammen. Aufgrund der Spezialisierung der Agenten beschränkt sich die Aufgabe der Agenten wie folgt:

Transportagent: (1) Anfahrt zu gesuchtem Agent bei gleichzeitiger Vermeidung von Kollisionen mit den Hindernissen und den Systemgrenze des Kernsystems, (2) Docking innerhalb Arbeitsbereiches des angefahrenen Agenten und (3) in der Energieerhaltung.

Montageagent: (1) Montage: Herstellung räumlicher Relation zwischen Montageobjekt auf dem Transportagenten und dem typbezogenen Montageteil im Montageagenten Lager, Transport von Montageteil, (2) Qualitätskontrolle des Montageschrittes, (3) Montagestation Lager auf laufenden Montageprozessen immer vorbereitet zu halten und (4) in der Energieerhaltung der Montageagenten selbst.

Qualitätsagent: (1) Überwachung von Montageschritt, (2) Änderung des Montagestatus, (3) Überwachung von der Lieferung der Montageteile.

Versorgungsagent: (1) Anfrage an den Montageagenten, laden gefragten Zahl und Typ der Teile, (2) steuert zu ihm an bei gleichzeitiger Vermeidung von Kollisionen mit den Hindernissen und den Systemgrenze des Kernsystems, (3) Lieferung von Teilen und (4) in der Energieerhaltung.

Energieagent: Jeder Agent ist Batteriebetrieben. Das heißt, dass jeder Agent nach gewisser Zeit eigene Batterien wieder aufladen muss. Es gibt zwei Typen von Energieagenten. Ein Typ

ist frei beweglich und steuert nach der Meldung den Montageagent an und wechselt dessen Batterie auf der aktuellen Position von Montageagenten. Zweiter Typ ist Orts gebunden. Er befindet sich in den Transportagenten Pool. Freie Transportagenten steuern beim Bedarf Ladestation an und stecken sich auf das Ladegerät.

Reparaturstation: (1) Reparatur von fehlerhaft montierten Baugruppen, (2) Änderung des Montagestatus.

Prüfungsstation: (2) Mechanische und Elektrische Kontrolle von gefertigten Elektromotor (2) Änderung des Montagestatus.

Entlade Station: (1) Manipulation von gefertigten Motor.

AGENTENKLASSE	FUNKTION	AUFLEGUNG
Transport	Montageplatz, Transportmittel für entstehende Produkte, Montage bezogenen Informationsträger.	Autonomer mobiler Agent mit konfigurierbare Palette. Batterie betrieben.
Montage	Montage von zugeordneten Montageschritt, Lagerung von entsprechenden Typ und Zahl von Montageteilen.	Mobile Agent mit Manipulator und Lager für drei verschiedene Montageteilen. Betrieben mitteln zwei austauschbaren Batterien.
Qualität	Qualitätssicherung von Montageschritt.	Technisches Vision System, das auf jedem Montagestation Manipulator montiert ist.
Versorgung	Lieferung von Montageteilen zur aktiven Montageagenten.	Autonomer mobiler Agent. Batterie betrieben.
Energie	Austausch von Montageagenten Batterie auf dem aktuellen Montageagenten Position.	Autonome mobile Agent angepasst für Transport- und Austausch von Montageagenten Batterien.
Reparatur	Reparatur oder Entfernung von fehlerhaften Baugruppen/ Produkten.	Fixe bemannte Station. Befindet sich auf der Systemgrenze.
Prüfung	Elektrische und Mechanische Kontrolle.	Fixe bemannte Station. Befindet sich auf der Systemgrenze vor Entlade Station.
Entlade	Übernahme von gefertigten Produkten mit guter Qualität.	Fix montierte Industrie Agent auf der Systemgrenze.

Tab. 7.2. Zerlegung von Agentenklassen, deren Funktion und Auflegung.

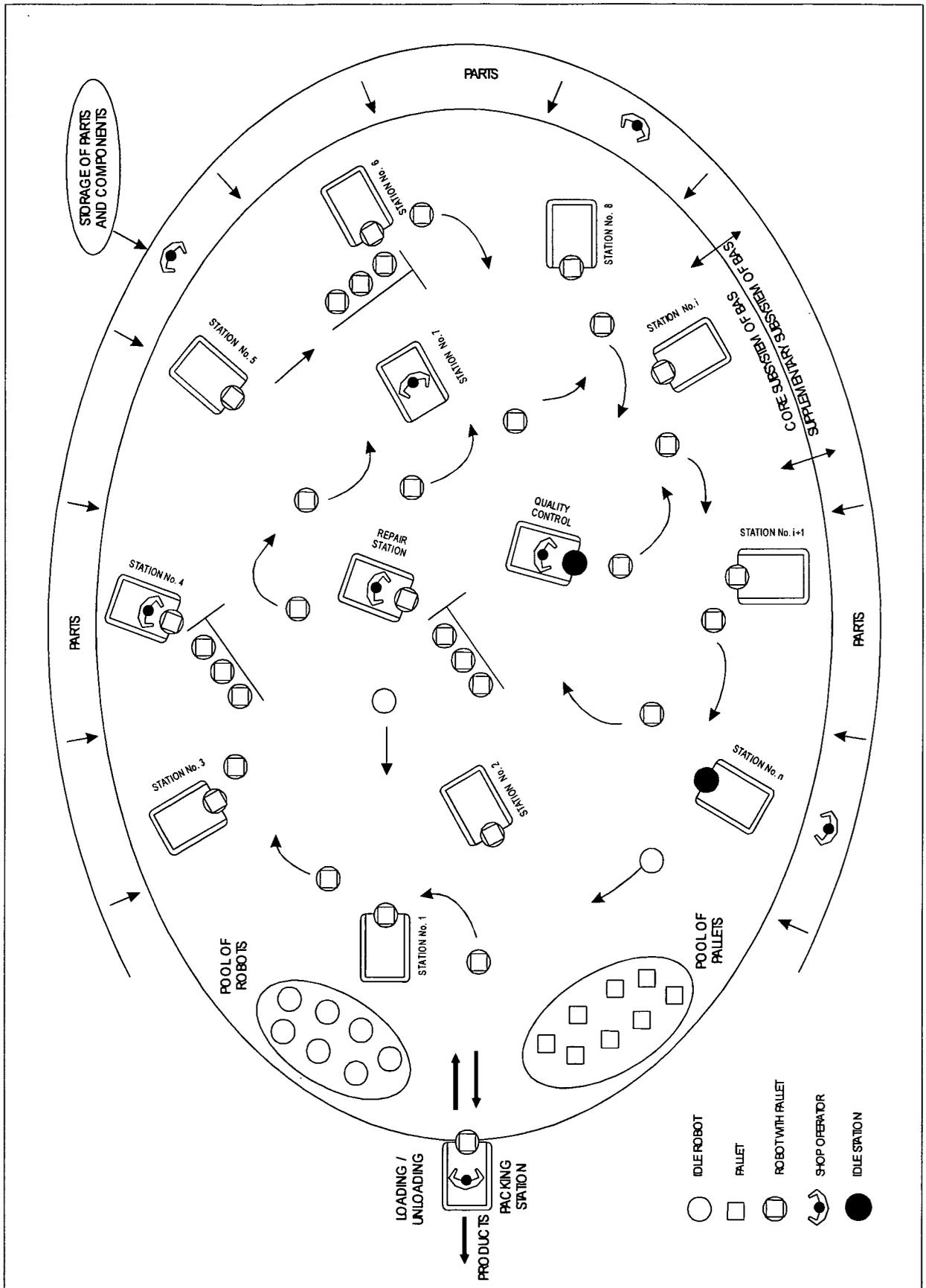


Abb. 7.6. Die zweite Stufe der Entwicklung des „Lineless“ Montagekonzepts (Katalinic, 2000).

Ergänzendes Subsystem umkreist das Kernsystem. Dessen Hauptaktivitäten sind Montageteile- und fehlerhafte Produktlagerung sowie richtig montierte Produkte nach Bestellungen zu sortieren und zu verpacken. Hauptaktivität an der Systemgrenze ist die Besorgung der Montageteile in eine Richtung und in der anderen, Lieferung der fehlerhaften Produkte oder Baugruppen und Lieferung von richtig montierenden Produkten. Weil Montageteile auf einer beliebigen Stelle auf der Systemgrenze eingeführt werden können, können nicht reparierte Produkte nur aus Reparatur- und Prüfungsstation und richtig montierte Produkte nur durch Entlade Station geliefert werden.

Jede Agentenklasse ist von einem Bereichagenten koordiniert. Bereichagenten sind softwaremäßige Agenten, die Bestandteil der vorhandenen Systemsteuerung, wie im Unterkapitel 6.2 erwähnt ist, sind. Sie bilden eine Schnittstelle zwischen selbst organisierende Montage und übergeordneten Steuerungsstrukturen. Bereichagenten ermitteln zum jeden Agent des entsprechenden Bereichs die Ziele, die er verfolgen muss. Was aber dieses System von klassischen Systemen unterscheidet ist, dass die Bereichagenten den Lösungsweg nicht beschreiben, sondern die Agenten so stimulieren, dass sie das Ziel selbst anstreben. Innerhalb des Kernsystems existiert keine zentrale Instanz.

KLASSISCHE SYSTEME	SELBSTORGANISIERENDE SYSTEME
SYSTEM	
Analytisch definierte, zentralistisch organisierte, geschlossene Systeme, im statistischen Gleichgewicht	Rekonfigurabel: polyzentrisch organisierte, offene Systeme mit selbsterzeugter Hierarchie, im dynamischen Gleichgewicht
STRUKTUR DES SYSTEMS	
Fix determinierte Struktur	Struktur entsteht und verändert sich durch die Interaktion der Agenten
UMSTRUKTURIERUNG	
Ohne extremen Aufwand Unveränderbar, große Zeitverluste	"glatte" Umstrukturierung in realer Zeit
STEUERUNG	
extern	intern
KAUSALITÄT	
Linear, Sequenziell	Nicht Linear, Parallel

Tab. 7.3. Vergleich von Hauptmerkmalen der klassischen und selbstorganisierenden Systeme.

Das Montagesystem organisiert sich selbst und zwar entsprechend dem einlaufenden Zustand der Umgebung, dem Auslastungsgrad und Fehlerzustand der Agenten. Dabei formen die Agenten implizit ein globales Verhaltensmuster, das auf der Ebene eines externen Beobachters sichtbar wird. Das globale Verhaltensmuster wirkt auf die lokalen Regeln des einzelnen Agenten zurück, indem es die Bedingungen mitbestimmt, die auf die Regeln wirken können. Diese interne Struktur, die durch die Kooperation der einzelnen Agenten entsteht bestimmt das Funktionieren des Gesamtsystems in selbst organisierender Weise. Der Vergleich zwischen klassischen und selbstorganisierten Systemen ist in Tabelle 7.3 dargestellt.

7.7 Integration der agentenbasierenden Montage in CIM-Struktur

Bei der Konzeption des selbstorganisierenden Montagesystems wurden die vorhandenen CIM-Strukturen und die Voraussetzungen für die Entstehung der Selbstorganisation berücksichtigt. Eine von Voraussetzungen für die Selbstorganisation ist Dezentralisation der gesamten Steuerung. Der Abbau der Zentralisierung erfolgt mittels der Verlagerung des Entscheidungspunktes von dem Leitrechner an die einzelnen Agenten. Das heißt, bestimmte Routineprozesse werden auf Agenten ausgelagert (Sehe: Transportagent Datenset). Diese Agenten übernehmen damit bestimmte Steuerungsprozesse autonom, ohne direkte Kontrolle durch höhere Steuerungszentren.

Dezentralisierung ist ein Hauptprinzip im agentenorientierten Ansatz. Konsequente Dezentralisierung verhindert das Auftreten von Engpässen in Berechnung oder Kommunikation, wenn das System sich ändern. Agenten müssen das System von sich aus erkunden und andere Agenten als Informationsquelle oder Dienstleistungs Anbieter betrachten.

Folgende Schritte der Verarbeitung sind je nach Grad der Zentralisierung zwischen den Agenten und dem Leitrechner verteilt:

AUFTRAGSANNAHME	ein Auftrag wird entgegengenommen
HANDLUNGSPLANUNG	der Auftrag wird in eine Folge von auszuführenden Aktionen zerlegt
ROUTENPLANUNG	für die notwendigen Fahraktionen werden Routen zwischen Start- und Zielpunkten ermittelt
BAHNPLANUNG	zur Feinplanung der Routen werden Bahnen zwischen Teilzielpunkten ermittelt
STEUERUNG	die Bahnen werden in Steueranweisungen für die Antriebe und Lenkungen gewandelt

Tab. 7.4. Die Verarbeitungsschritte.

Von Interesse ist hier ihre Verteilung je nach Grad der Dezentralisierung. Es lassen sich dabei die beiden Extreme, der zentralen Organisation und der vollen Autonomie ausmachen. Bei der streng zentralen Organisation ist das System ein Top Down organisiertes System.

ZENTRALISIERTE ORGANISATION	LEITRECHNER	Auftragsannahme
		Handlungsplanung
		Routenplanung für jeden Agent in zentralem Umweltmodell
		Bahnplanung für jeden Agent in zentralem Umweltmodell
	AGENT	Steuerung nach Bahnen vom Leitreehner

Tab. 7.5. Die Aufteilung der Verarbeitungsschritte in der zentralisieren Organisation.

Der Agent kann nur als „Sklave“ des Leitrechners betrachtet werden, dem die volle Verantwortung für das ganze System zukommt.

Bei voller Autonomie ist dagegen der Leitreehner obsolet, beziehungsweise ist der „Sklave“ eines Bottom Up selbstorganisierten Systems.

VOLLE AUTONOMIE	AGENT	Auftragsannahme
		Handlungsplanung
		Routenplanung
		Bahnplanung
		Steuerung nach Bahnen

Tab. 7.6. Die Aufteilung von Verarbeitungsschritten in der völlig autonomen Organisation.

In der Darstellung des selbstorganisierten Montagesystems sind die Kompetenzen wie in der Tabelle 7.7 bezeichnet, geteilt.

VORHANDENE CIM STRUKTUR	LEITRECHNER	Auftragsannahme
		Handlungsplanung
SELBSTORGANISATION	AGENT	Routenplanung
		Bahnplanung
		Steuerung nach Bahnen

Tab. 7.7. Die Aufteilung von Verarbeitungsschritten im Montagesystem.

Die Aufträge, entgegengenommene Handlungsplanung, Optimierung und Scheduling erfolgt über das Produktionsplanungssystem. Die Produktionsplanung auf der Betriebebene gibt Antworten auf folgenden Fragen: was für einen Montagevorgang erfolgen soll, welche Montageanlage dafür beauftragt wird, wie viele Aufträge gibt es, ab wann und wie lange ist die dafür beauftragte Anlage besetzt.

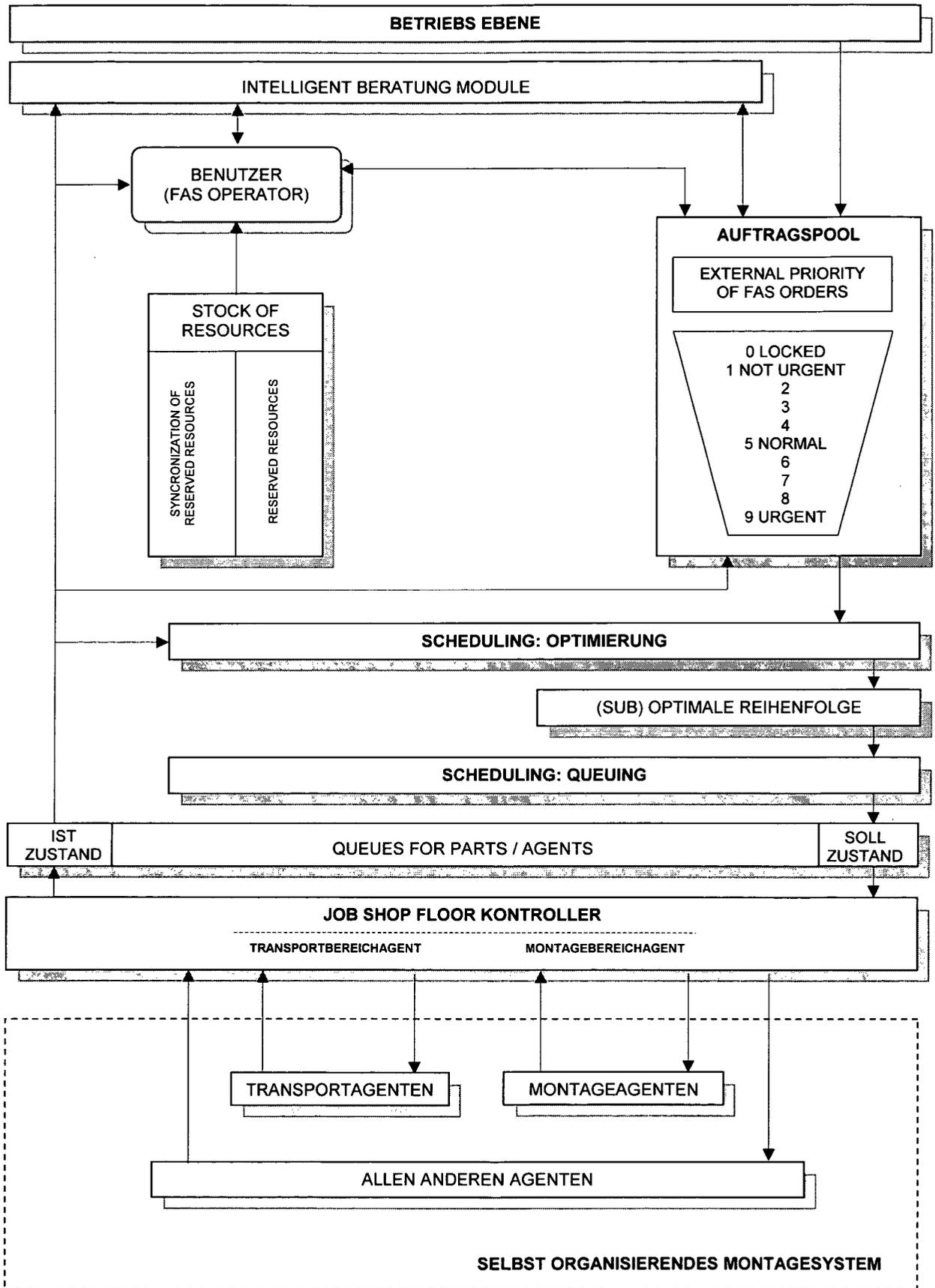


Abb. 7.7. Interaktion zwischen Hierarchical Scheduler und Job Shop Floor Kontroller.

Schnittstelle zwischen Betriebsplanung und Montageanlage erfolgt durch Auftragspool. In einem Auftragspool gibt es Aufträge mit verschiedenen Wichtigkeitsstufen.

Das übergeordnete Planungssystem vergibt zu jedem einzelnen Auftrag eine Priorität, die dem Eiligkeitsgrad dieses Auftrags entspricht. Dieses Prioritätensystem hat zehn Stufen, von 0 bis 9. Die höchste Prioritätsstufe ist 9 und bezeichnet sehr eilige Aufträge. Stufe 5 bezeichnet normale Aufträge. Stufe 1 bezeichnet nicht eilige Aufträge und 0 bezeichnet Aufträge, die gesperrt sind (Katalinic, 1990). Dieses Prioritätensystem drückt die Anforderungen des Marktes aus. Dadurch wird das Just-in-Time Produktionskonzept in die Basis des Systems eingebettet. Optimierungsprozess muss gemäß dem Auftragspool das passende Ausführungsszenario innerhalb der folgenden Beschränkungen finden: Ziel Szenario, Planungskriterien, eigentlicher Zustand der Montageanlage, freie und reservierte Produktionsmittel innerhalb der geplanten Zeit.

Resultat der Optimierung ist (sub)-optimale Reihenfolge. Das ist der Ausgangspunkt für Queing. Queing bestimmt Montagereihenfolge und Materialfluss. Abbildung 7.7 zeigt das Konzept (Katalinic, 1990, Katalinic, 1999), das von Professor Branko Katalinic im Zeitraum von 1990 bis 1999 entwickelt worden und in (Katalinic, 1997, Stuja, 2002) detailliert beschreiben ist.

Eingang von dem Scheduling Kontroller besteht aus Typ und Zahl den Elektromotoren, die in der festgelegten Abfolge montiert werden sollen. Diese Information wird von den Transport- und Montagebereichagent, die Teil von Scheduling Kontroller sind, übernommen und in konkrete Aufträge für die Agentenklasse umgewandelt. Aufgaben von Montagebereichagenten sind:

Auswahl von Montageagenten: entsprechend den Montageaufträgen werden passende Montageagenten aktiviert.

Übermittlung von Motortyp: Jeder aktiven Montageagent bekommt Liste mit Elektromotorentypen, für welche er zuständig ist. Aufgrund dieser Information kann jeder Montageagent kontrollieren, ob er für angefahrenen Transportagenten zuständig ist.

Übermittlung von Rüstungsplan zu Shop Floor Operator: Rüstungsplan besteht aus der Liste allen neuen aktivierten Montageagenten mit der Zahl und Typ der Teile, deren Lagern sie übernehmen müssen. Bevor aktivierter Montageagent in der Montagestruktur eingeschlossen wird, muss dessen Lager manuell ausgerüstet sein. Dies wird von Shop Floor Operator gemacht. Nachrüstung aktiver Montageagenten wird von Versorgungsagenten übernommen.

Übermittlung von Rüstungsplan zu Montageagenten: Jede Montageagent bekommt Zahl und Typ der Teile, vor der Montage im eigenen Lager vorhanden sein müssen.

Überwachung von Montageagenten: Zahl und Typ der aktiven, passiven und ausgefallenen Montageagenten.

Wartung: Sobald Montageagent ausgefallen ist, aktiviert Montagebereichagent alternativen Montageagent, berechnet Eiligkeitsgrad der Reparatur und benachrichtigt darüber Shop Floor Operator. Eiligkeitsgrad der Reparatur wird aufgrund von Montageaufträgen und aktuellen Auslastungsgrad der Montageagenten zugeteilt. Es gibt drei Eiligkeitsgrade; 0 – nicht eilig, 1 – kann bis 30 Minuten warten, 3 – sofort erledigen. Eiligkeitsgrad 3 wird zugeteilt wenn kein alternativer Montageagent frei ist und wenn Montageagenten mit Eiligkeitsgrad 0 und 1 den Eiligkeitsgrad 3 erreicht haben.

Aufgaben von Transportbereichagent sind:

Vorbereitung von Transportaufträgen: Eine Liste von Motorentyp und Fälligkeitstermin der Montage wird erstellt. Abbildug xy zeigt Auftrag Pool des Transportbereichagenten.

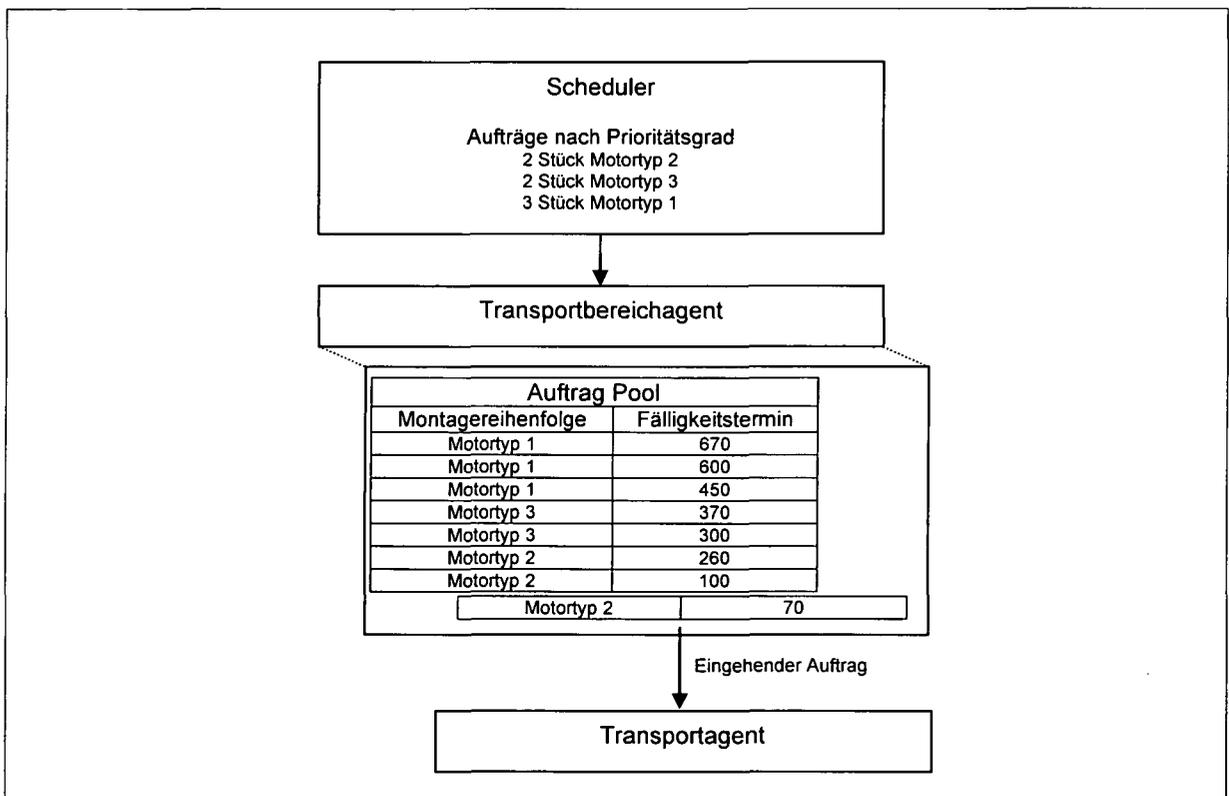


Abb. 7.8. Auftrag Pool des Transportbereichagenten.

Aktivieren von Transportagenten: Transportagent wird aktiviert sobald er Motortyp, der auf ihm montiert wird, von Transportbereichagent bekommt.

Überwachung von Transportagenten: Zahl der aktiven, passiven und ausgefallenen Transportagenten.

Wartung: Sobald Transportagent ausgefallen ist, wird neuer Transportagent aktiviert und Wartungsauftrag zum Shop Floor Operator geschickt. Da es innerhalb Kernsubsystems immer genügend Transportagenten gibt, wird vom Transportbereichagent der Eiligkeitsgrad der Reparatur nicht berechnet.

7.8 Kommunikationskonzept

Der Informationsaustausch zwischen dem übergeordneten Computer Integrierte Produktionsstruktur und dem selbstorganisierenden Montagesystem sowie zwischen Agenten innerhalb Montagesystems selbst, hat entscheidenden Einfluss auf die Funktionalität des Montagesystems. Die Aktionen des Agenten sollen im Kontext der Erfüllung eines Montageauftrags erfolgen.

Entsprechend der Aufgabeverteilung gibt es aber keinen Agent oder Agentenklasse die allein einen Montageauftrag erledigen könnte. Dagegen ist jeder Agent einer Klasse fähig die Aufgabe der Klasse selbständig auszuführen.

Der Montagaauftrag entsteht aus der Kooperation der verschiedenen Agentenklassen. Kooperation wird hier als ein Prozess verstanden, bei dem die Agenten ihre Aktivitäten aufeinander abstimmen. Aktivitäts Abstimmung zwischen Agenten ist jedoch ohne Kommunikation undenkbar, denn die Agenten erreichen durch Informationsaustausch

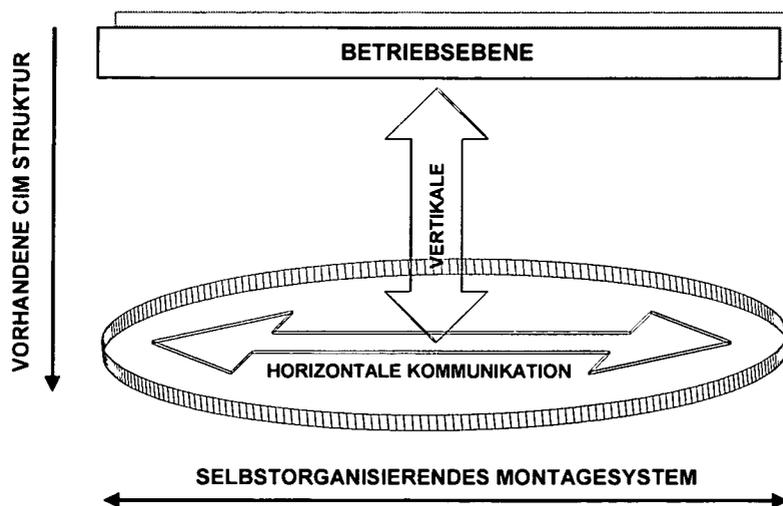


Abb. 7.9. Integration des selbst organisierenden Montagesystems in CIM Struktur aus der Sicht des Informationsaustausches. Vertikale und horizontale Kommunikation bilden den Kern des Konzepts.

notwendige Einstellungen anderer Agenten. Gemäß diesem Entwurf sind zwei globale Kommunikationskanäle erforderlich, im Weiteren als Vertikale und Horizontale Kommunikation bezeichnet. Dieses Kommunikationskonzept ist auf der Abbildung 7.9 gezeigt.

Vertikale Kommunikation bezeichnet funktionelle Kopplung zwischen übergeordnetem Computer Integrierte Produktionsstruktur und selbstorganisierenden Montagesystem. Vertikale Kommunikation erfolgt zwischen Transport- und Montageberiechagent und entsprechenden Agentenklassen und zwischen allen anderen Agentenklassen und Scheduler. Nach der Aufgabeverteilung unter Agentenklassen, hängen nur Transport- und Montageagentenklasse direkt von Montageaufträgen ab. Alle anderen Agentenklassen hängen von aktiven Transport- und Montageagenten ab.

Vertikaler Informationsaustausch beinhaltet Meldungen von Reparaturagenten und Prüf Station über Typ und Zahl von ausgeschiedenen sowie die Information von Entladestation über Typ und Zahl von gefertigten Elektromotoren. Dadurch weiß Scheduler was für einen Motorentyp nicht gefertigt wird, erhöht die Zahl der entsprechenden Motorentypen, die noch montiert werden müssen und macht re-scheduling. Aus demselben Grund meldet ausgefallener Transportagent Motorentyp der ihn trägt. Darüber hinaus meldet jeder Agent eigenen Status. Aufgrund dieser Meldungen wird Scheduler immer auf dem Laufenden gehalten. Montageagenten können vier Status haben, alle anderen Agentenklassen nur erste drei, die in der Tabelle 7.8 zusammengefasst sind.

STATUS	BESCHREIBUNG
Aktiv	Agent beteiligt sich in der Montage. Er ist Teil der aktuellen Montagestruktur.
Funktionsstörung	Agent ist nicht mehr funktionsfähig. Wird automatisch aus der Montagestruktur abgezogen.
Passiv	Agent ist nicht aktiv und wartet auf Aktivierungssignal von Scheduler.
Queue überlastet	Warteschlange vor dem Agent ist voll. Dieser Status gilt nur für Montagestation. Montagestation ist funktionsfähig.

Tab. 7.8. Agentenstatus.

Abbildung 7.10 fasst vertikale Kommunikation Struktur und vertikale Informationsaustausch zusammen.

Horizontale Kommunikation dagegen ermöglicht und strukturiert Kooperation in der selbst organisierenden Weise. Dies ist durch gezielten Informationsaustausch zwischen Agentenklassen realisiert.

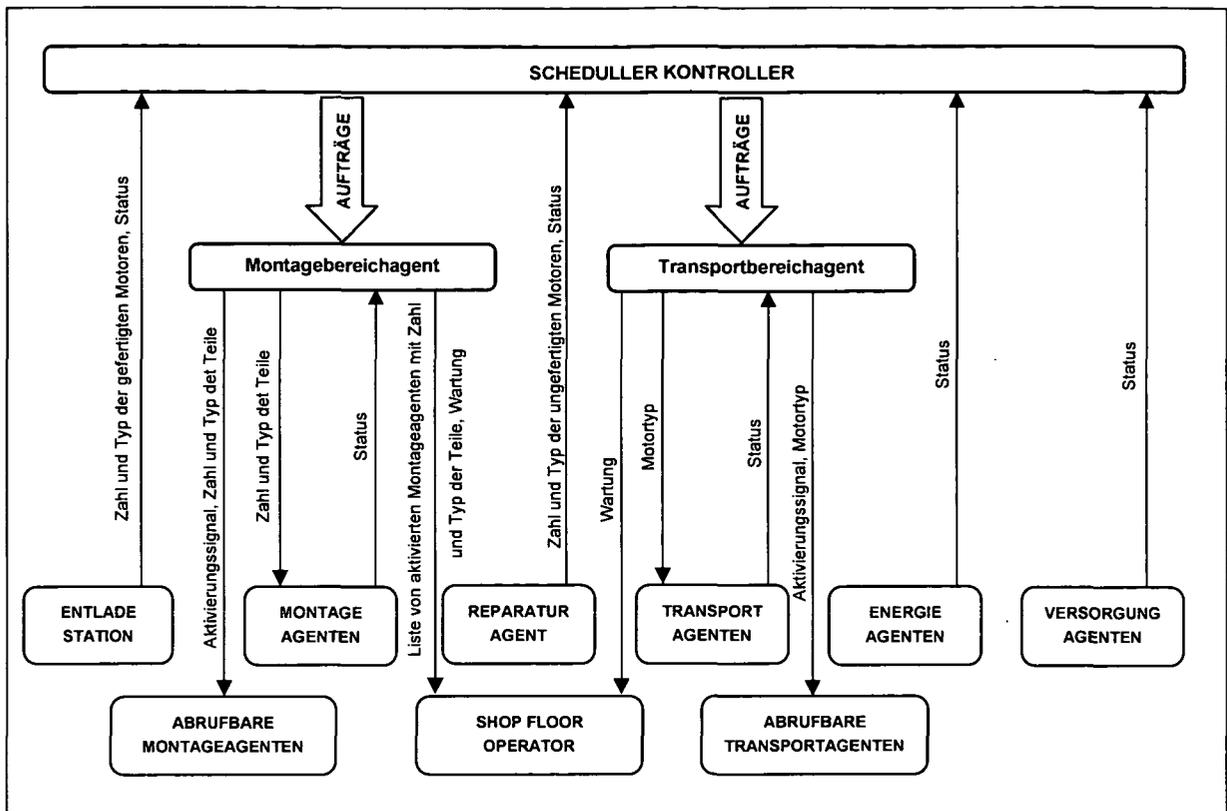


Abb. 7.10. Vertikale Kommunikationsstruktur und Informationsaustausch.

Horizontale Kommunikation ist ein synergetisches Glied, das ein Intelligentes Gesamtverhalten des Montagesystems bildet. Agenten haben selbst keine Intelligenz per se. Fasst man jedoch die Aktivitäten von Agenten zusammen, so entsteht ein intelligentes Gesamtverhalten.

Horizontale Kommunikation besteht aus drei Subsystemen:

- Auswahl der geeigneten Station für nächsten Schritt,
- Zielansteuerung,
- Informationsaustausch innerhalb der angefahrenen Station.

Auswahl der geeigneten Station für nächsten Schritt ist Teil der Horizontalen Kommunikation zwischen den Transportagenten und Montagestationen. Die Art und Weise, wie und wann diese Kommunikation stattfindet, hat entscheidenden Einfluss auf das Entstehen, die Stabilität und Kapazität des Montagesystems. Außerdem entsteht dadurch ein neuer Scheduling-Ansatz. Diesem Teil der Horizontalen Kommunikation ist nächstes Kapitel gewidmet.

7.8.1 Zielansteuerung

Transport-, Montage-, Versorgungs- und Energieagentenklasse sind frei bewegliche Agenten. Dass heißt, jeder Agent muss zusätzlich zum vertikalen und horizontalen Informationsaustausch die Fähigkeit besitzen, eigene Zielposition zu orten und zu ihr hinzusteuern bei gleichzeitiger Vermeidung von Kollisionen mit den Hindernissen und den Begrenzungen des Kernsystems des selbstorganisierenden Montagesystems. Erst dann können Agenten sinnvolle Tätigkeiten ausüben.

Ähnliche Aufgabe haben zahlreiche autonome Agenten die in Büros als Müllsammler arbeiten, oder die Fußball auf einem mit Markierungen versehenen Spielfeld spielen. Bei nahe allen ist die Zielansteuerung zur Navigation Domäne zugeschrieben. Leonard und Durrant-Whyte (Leonard & Durrant-Whyte, 1991) fassen das allgemeine Problem der Agentennavigation in die drei Fragen zusammen; Wo bin ich? Wohin gehe ich? und Wie gelange ich dorthin? Anders formuliert müssen für ein vollständiges Navigationssystem die Probleme Selbstlokalisierung, Kartenerstellung und Pfadplanung gelöst werden (Gutmann, 2000).

Bei der Selbstlokalisierung geht es darum, herauszufinden, wo sich der Agent in seiner Umgebung befindet. Hierfür können die Messungen der auf dem Agenten befindlichen Sensoren eingesetzt werden. Bei der Kartenerstellung möchte man mit den Sensoren des Agenten eine Karte erstellen, die der Agent z.B. für Selbstlokalisierung und Pfadplanung verwenden kann. Schließlich wird bei der Pfadplanung versucht, kollisionsfreie Wege von einer Start- zu einer Zielposition zu finden, die der Agent in der realen Welt abfahren soll. Betrachten wir ein anderes Beispiel. Agenten bewegen sich plan- und ziellos auf einer Oberfläche. Allerdings generiert jeder Agent bei jedem Schritt Information, indem er lokal eine Markierung setzt. Die Markierung codiert also Information auf materieller Grundlage und kann deshalb als strukturelle Information bezeichnet werden. Da die Markierungen auf der Oberfläche gespeichert sind, ist die strukturelle Information auf diese Weise unabhängig von den Agenten. Die Markierungen selbst haben eine Eigendynamik, sie können verblassen und damit langsam wieder verschwinden, wenn sie nicht ständig erneuert werden. Wenn andererseits ein Platz von einem oder verschiedenen Agenten mehrmals aufgesucht wird, nimmt die Stärke der Markierung wieder zu. Die Information kann also lokal akkumuliert werden. Außerdem kann die Information sich eigenständig ausbreiten (z.B. durch Diffusion der Markierungen).

Die Oberfläche ist damit charakterisiert durch eine Informationsdichte $h(r, t)$, die angibt wie stark die Markierung an einem bestimmten Ort r zu einer gegebenen Zeit t ist.

In diesem Modell nimmt man an, dass die Agenten gedächtnislos sind, das heißt, sie können intern keine Information akkumulieren, aber sie verfügen über funktionelle Information, die

die Aufgabe hat, die strukturelle Information im Hinblick auf den Rezipienten zu interpretieren.

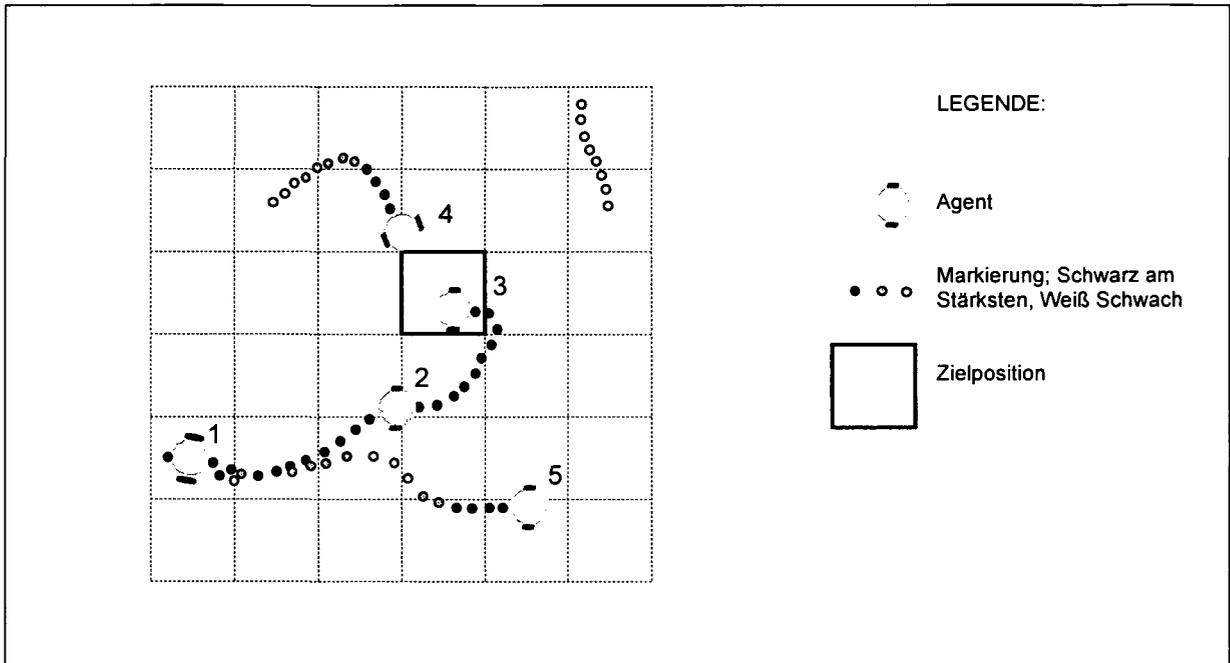


Abb. 7.11. Strukturelle Kommunikation rein reaktiver Agenten. Agenten 1 und 2 folgen Agent 3. Agenten 3, 4 und 5 bewegen sich ziellos. Eben Agenten 3 und 4 stoßen zufällig auf Zielposition.

Im konkreten Fall ist die funktionale Information ein Algorithmus, der die Agenten befähigt, die strukturelle Information, also die Markierungen zu lesen, wenn sie sich im direkten Umkreis ihres Platzes befinden.

Bei einer Realisierung auf einem Gitter bedeutet dies, dass der Agent genau die nächsten Nachbarplätze erkennen kann, nicht aber Gitterplätze, die mehr als einen Schritt entfernt sind. Abbildung 7.11 zeigt diese Situation. Werden Markierungen in der unmittelbaren Umgebung entdeckt, dann können sie die Bewegungsrichtung des Agenten beeinflussen. Der Agent wird mit einer gewissen Wahrscheinlichkeit der stärksten Markierung folgen.

Die funktionale Information existiert im vorliegenden Modell als ein Satz von einfachen Regeln, die sind in folgenden deskriptiven Algorithmus beschreiben.

Start

{der Agent prüft lokal, ob sich in seiner unmittelbaren Umgebung Markierungen befinden

wenn {ja}

dann {setzt an seinen jetzigen Platz eine Markierung und nimmt die Richtung des nächsten Schrittes in Abhängigkeit von der Stärke der lokalen Markierung}

```

    sonst {fährt ziellos weiter}
  }
Ende

```

Mit dem Algorithmus ist vorgegeben, was der Agent an Information aus der vorhandenen strukturellen Information herauslesen kann. Dieser Algorithmus kann in der Tat von sehr simplen Agenten abgearbeitet werden, da es keinerlei interner Informationsspeicherung bedarf. Physikalisch gesehen, bewegen sich die Agenten fortlaufend in die Richtung des größten lokalen Gradienten eines Potentials, das sie selbst verändern können. Über das Feld $h(r, t)$ sind also die Agenten des Systems miteinander verkoppelt. Diese Art von Wechselwirkung kann als indirekte Kommunikation der Agenten beschrieben werden, wobei das Feld als Kommunikationsmedium wirkt, das dem jeweiligen Agenten etwas über den gegenwärtigen Zustand des Systems und die Aktivitäten anderer Agenten mitteilt. Wichtig dabei ist, dass die Agenten durch das Feld nur eine lokale Information erhalten; sie kennen nicht den Zustand des Systems (den Wert von $h(r, t)$) an jedem Ort r , sondern nur in der Umgebung ihres eigenen Standortes. Natürlich kann sich im vorliegenden Modell die mit $h(r, t)$ verbundene Information durch Diffusionsprozesse ausbreiten. Der Agent soll aber jeweils nur über diejenige Information verfügen, die durch den Wert von $h(r, t)$ an seinem gegenwärtigen Standort gegeben ist, unabhängig davon, wie diese Information zu ihm gelangt ist.

Das Agenten-Modell simuliert einen Aggregationsprozess auf der Grundlage chemotaktischer Kommunikation, wie er für viele biologische Wesen, insbesondere für Insekten, typisch ist. Agenten gelingt es von Ausgang bis zum Zielpunkt zu steuern ohne Selbstlokalisierung, Kartenerstellung und Pfadplanung. Das Agentensystem optimiert sich von selbst. Auf dem kürzesten Weg verstärkt sich die Markierung schneller, so dass immer mehr Agenten diesem Weg folgen.

Wesentlicher Nachteil des gegebenen Modells ist dass es nur dann funktioniert wenn die Zahl der Agenten groß ist. Experimente in den Ameisen Kolonien zeigten dass die Effizienz der Kolonie linear mit dem exponentialen Zuwachs der Population steigt. Unter einer gewissen Zahl von Ameisen stirbt die Population aus.

Um Effizienz der Zielerreichung unabhängig von der Zahl der Agenten zu machen, muss jeder Agent die Fähigkeit besitzen, aus jedem Ort innerhalb Kernsubsystems selbständig Zielposition zu detektieren.

Die Montage kann man als eine Reihe von Übereinstimmungen zwischen Montage und Transportagenten betrachten. Das heißt, Transportagent sucht geeigneten Montageagenten, der geeigneten Transportagenten sucht. Demzufolge muss jeder beteiligte Agent eindeutige

Signatur, die im ganzen Kernsubsystem lesbar ist, haben. Dies ist mittels Radio RF System realisiert.

Als Funkfrequenz wird das globale Industrial, Scientific und Medical Band verwendet. Das Band deckt den Frequenzbereich zwischen 2,400 GHz und 2,4835 GHz. Dieser Frequenzbereich steht nahezu weltweit zur Verfügung und kann für unlicenzierte Übertragungen gebührenfrei genutzt werden sodass der schnellen Einführung von Kommunikationssystem nichts im Weg steht.

Das Radio RF System ist auf Agentenklassen angepasst, da die Agentenklassen unterschiedlichen Anforderungen genügen müssen. Montageagenten richten eigene Position schrittweise nach der Richtung von angefahrenen Transportagenten unter der Bedingung dass Stillstand zwischen zwei Montageschritten minimiert wird. Transportagenten dagegen müssen sich in der Richtung der geeigneten Montageagenten bewegen und eigene Geschwindigkeit abhängig vom Abstand zum Montageagent, anpassen. Genau so Versorgungs- und Energieagenten. Reparatur-, Prüfungs-, Entlade Agent und Transportagenten Pool müssen nur eigene Position markieren.

Alle Agentenklassen, außerhalb Versorgungs- und Energieagentenklasse, sind je mit einem Sender ausgestattet. Jeder Sender sendet auf vordefinierte Frequenz. Damit ist jeder Agent, der oben genannten Klassen eindeutig gekennzeichnet.

Um relative Orientierung eines Senders zu orten genügen ein Empfänger und Richtstrahlantenne, die sich rotieren kann. Der Empfänger muss die Sender Frequenz empfangen und ausgestrahlte elektromagnetische Energie mäßigen können. Der Sender befindet sich in der Richtung, in der maximale Elektromagnetischeenergie abgetastet wird. Dadurch ist relative Orientierung des gesuchten Senders bekannt und Agent kann zu ihm steuern. Außerdem es ist möglich relative Distanz zwischen dem Agenten und Sender zu berechnen. Dies ist für Anfahrt wichtig, denn Agent kann eigene Geschwindigkeit abhängig von Distanz zu Zielposition einstellen und damit Zusammenstöße vermeiden.

Um den Abstand zum Sender berechnen zu können, muss Sender Leistung und Wirkleistung von Senderantenne bekannt sein (Gleichung 5.xy). In stahlarmierten Betongebäuden ist jedoch die Berechnung nach dieser Methode wegen der Reflexionen und zahlreichen Sendern nicht genau einstellbar. Deswegen sind Transport-, Versorgungs- und Energieagenten mit zwei Empfängern und Richtstrahlantennen ausgerüstet. Die Richtstrahlantennen befinden sich von linker und rechter Seite des Agenten und sind nach vorne eingerichtet. Beide Richtstrahlantennen rotieren sich parallel zu einander bis abgetastete Energie in beiden Empfängern gleich ist. Danach rotieren sich Richtstrahlantennen zu einander solange in beiden Empfänger maximale Elektromagnetischeenergie abgetastet wird. Berechnung der relativen Distanz zum Ziel für die Einordnung der Größen wie auf der Abbildung 7.12 gezeigt ist wie folgt:

$$D = D_x + D_y \quad (7.1)$$

$$\tan \beta = \frac{d}{D_y} \Rightarrow D_y = \frac{d}{\tan \beta} \quad (7.2)$$

$$\tan \alpha = \frac{d}{D_x} \Rightarrow D_x = \frac{d}{\tan \alpha} \quad (7.3)$$

$$D = \frac{d}{\tan \alpha} + \frac{d}{\tan \beta} \Rightarrow d = \frac{\tan \alpha \cdot \tan \beta}{\tan \alpha + \tan \beta} \cdot D \quad (7.4)$$

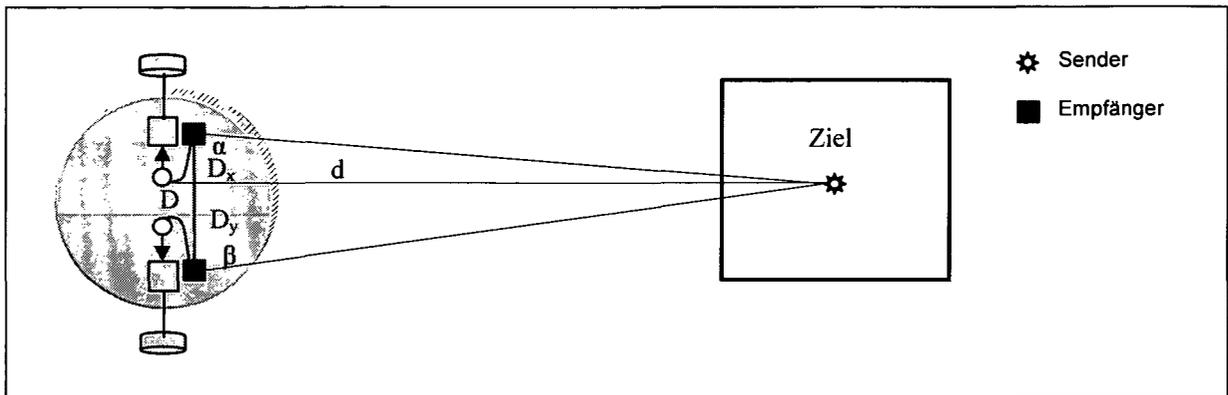


Abb. 7.12. Zielsteuerung nach Trigonometrische Methode.

Empfänger auf den Transportagenten können gleichzeitig nur eine Frequenz empfangen. Was für eine Frequenz empfangen wird hängt von dem aktuellen Montagestatus sowie Auslastungsgrad und Verfügbarkeit der Montageagenten.

Versorgungs- und Ladeagenten scannen ganzen Senderfrequenzbereich und warten auf das Aktivierungssignal von den Montageagenten. Aktivierungssignal trägt Senderfrequenz, auf die Empfänger danach eingestellt werden.

Dieser Teil des Radio Systems dient dazu Montageagenten, Reparatur-, Prüf- und Entlade Agent sowie Transportagenten Pool zu lokalisieren und zu denen Transport, Versorgungs- und Energieagenten anzusteuern.

Darüber hinaus soll jeder Montageagent eigene Stellung innerhalb der entstehenden Montagestruktur optimieren. Optimierung geschieht nach dem minimalen Stillstand Gesetz. Das heißt, jeder Montageagent versucht durch schrittweise Nachregelung der aktuellen Position und Orientierung die Zeit zwischen zwei Montagen zu minimieren. Die Nachregelung kann aber beschleunigt werden wenn Montageagent die Richtung von angefahrenen Agenten abtasten kann. Deswegen ist jeder Montageagent mit einem Empfänger und Richtantenne ausgestattet. Diese Empfänger empfängt nur eine Frequenz. Was für eine, hängt von Montageschritt, mit welchem der jeweilige Montageagent beauftragt

ist. Jeder Transportagent ist dagegen mit einem Sender ausgestattet. Dieser Sender sendet auf für jeweiligen Montageschritt vordefinierter Frequenz.

Tabelle 7.9 zeigt Auslegung von Radio RF System nach Agentenklassen und deren Funktion.

AGENTENKLASSE	RF RADIO AUSSTATTUNG	FUNKTION
Transport	2 Empfänger, 2 Richtantennen	Zielposition Ansteuern
	Sender	Eigene Markierung
Montage	Empfänger, Richtantenne,	Position und Orientierung Optimieren
	Sender	Eigene Markierung, Aktivierungssignal
Versorgung und Energie	2 Empfänger, 2 Richtantennen	Zielposition Ansteuern
Prüfung, Reparatur und Entlade Transportagenten Pool	Sender	Eigene Markierung

Tab. 7.9. Auslegung und Funktion von dem Radio RF System nach Agentenklassen.

Das Radio RF System dient als Markierung der Zielpositionen. Da keine beziehungsweise bei den Montageagenten nur paar Bit Informationen ausgestrahlt werden, spielen Störquellen geringere Rolle. Daher kann Band auf mehreren Kanälen aufgeteilt werden. Im Bereich von 2,400 GHz bis 2,4835 GHz werden 83 Kanäle im Abstand von 1 MHz erreicht. Für reibungslose Funktion des Systems beträgt minimaler Kanalabstand 0,2 MHz. Wenn der Kanalabstand kleiner wird, kann es passieren, dass Empfänger eines Agenten, ausgestrahlte Signale von mehreren Kanälen gleichzeitig empfängt oder dass er einen anderen Sender als sein eigenes Ziel interpretiert (cross-talk). In den beiden Fällen verliert der Agent die Orientierung und kann keine sinnvollen Tätigkeiten mehr ausführen. Eben die maximale Zahl der beteiligten Agenten ist durch physikalische Größe des Kernsubsystems determiniert.

7.8.2 Informationsaustausch innerhalb der angefahrenen Stationen

Dieser Teil des horizontalen Informationsaustauschs erfolgt, wenn sich ein Agent innerhalb des Arbeitsbereichs eines anderen Agenten befindet. Abbildung 7.13 fasst den Informationsaustausch innerhalb der angefahrenen Station zusammen.

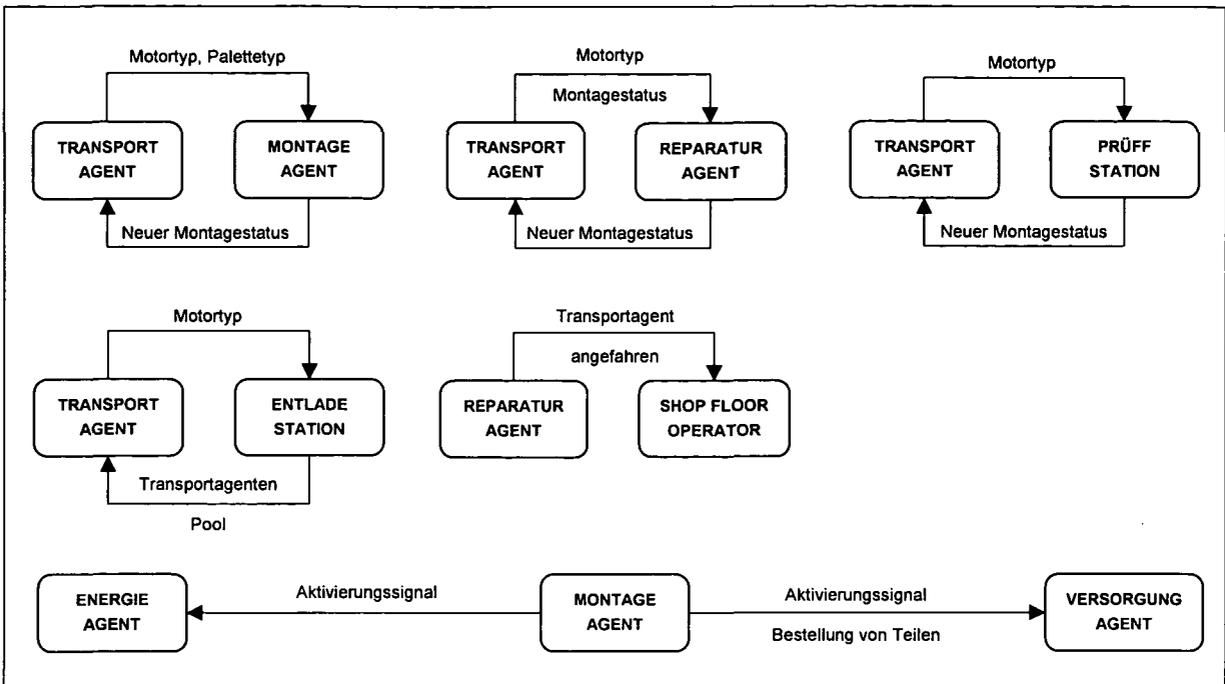


Abb. 7.13. Horizontale Kommunikation Struktur und Informationsaustausch.

7.8.3 Technische Realisation der Vertikalen Kommunikation und der Horizontalen Kommunikation innerhalb der angefahrenen Station

Daher kann horizontale Kommunikation mittels eines kurzrangigen drahtlosen Kommunikationssystem aufgebaut werden.

Vertikale Kommunikation muss dagegen ganzes Kernsubsystem des selbstorganisierenden Montagesystems decken. Die Möglichkeit von jedem Ort innerhalb Kernsubsystem vertikale Kommunikation zu realisieren ist entscheidend.

Vertikale und horizontale Kommunikation erfolgt mit Hilfe Wireless LAN und Infrarot Kommunikationssystem. Tabelle 7.10 fasst Anforderungen und Auflegung von Kommunikationssystem zusammen.

Die Reichweite eines Wireless LAN Systems ist von einer ganzen Reihe von Faktoren abhängig wie z.B. Hindernisse, Reflexionen, Störquellen. Kommerzielle Wireless LAN Systeme erreichen in Räumen ohne größere Hindernisse und Störeinflüsse bis zu 400 m.

KOMMUNIKATION	DATENTRANSFER	REICHWEITE	AUFLEGUNG
Vertikale	Niedrig	Kernsubsystem	Wireless LAN, Infrarot
Horizontale	Niedrig	< 2 m	Infrarot

Tab. 7.10. Anforderungen und Auflegung von Kommunikationssystem.

Darüber hinaus können Richtfunkantennen für dedizierte Strecken auch mehrere Kilometer überwinden. Maximale Datenraten betragen 11Mbit/s. Durch die Wireless LAN ist vertikaler Informationsaustausch außer Transportaufträgen Übermittlung aufgebaut.

Infrarot Kommunikation deckt die Arbeitsbereiche der Agenten und dient dazu ortsspezifischen Informationsaustausch zu ermöglichen. Da gehört horizontale Kommunikation zwischen Agentenklassen und die vertikale Kommunikation zwischen Transportbereichagent und Transportagenten in der Transportagenten Pool. Durch Infrarotkommunikation ist es gesichert dass nur ein Transportagent gleichzeitig einen Transportauftrag übernehmen kann und dass diese Übertragung nur in dem Transportagenten Pool Ausgang passiert.

Infrarotkommunikation bietet Datenraten bis 9,6 kbit/s. Der Vorteil der Infrarotkommunikation ist, dass die Ausbreitung von Infrarot den Gesetzen der Optik gehorcht. Daher ist die Reichweite ziemlich exakt festlegbar, was wiederum erlaubt, mehrere Infrarot Schnittstelle parallel zu betreiben. Innerhalb eines bestimmten Ortsgebiets darf jedoch kein Kanal doppelt belegt werden. Derselbe Kanal funktioniert an einem anderen Ort jedoch ebenfalls kollisionsfrei. Sichere Datenübertragung ist im jeden Fall auf paar Meter leicht realisierbar. Ein zusätzlicher Vorteil ist der, dass Infrarot eine unsichtbare Form des Lichts ist, die zudem anders als Laser keinerlei Gefahrenmoment für Menschen darstellt.

Alle am Markt erhältlichen integrierten Infrarotkommunikationssysteme enthalten sowohl eine Sende- und eine Empfangseinheit. Mit einem derartigen Infrarotkommunikationssystem ist allerdings nur eine Halbduplex-Verbindung möglich. Es kann also zu einem Zeitpunkt nur gesendet oder empfangen werden. Die einzelnen Infrarot-Geräte stehen im Stand-by Modus, solange keine Verbindung angefordert wird. Jeder Agent ist mit einem Infrarot-Gerät ausgestattet.

Da die Datentransferraten sehr niedrig sind, sind beide Systeme überdimensioniert. Damit spielen die Störungen geringere Rolle. Infrarot sowie Wireless LAN Systeme sind kommerzielle Produkte die direkt eingebaut und benutzt werden können. Ziellansteuerungssystem dagegen muss erst konzipiert und aufgebaut werden.

7.9 Transportagent Datenset

Die einzelnen Transportagenten sind mit individuellen Steuerungen und Speicher ausgestattet sodass die Dateien, die für Transportabläufe wichtig sind, auf einzelne Transportagenten verlagert sind. Das sind: Motorentypen, dem Palettentypen und Liste aller Empfänger- und Senderfrequenzen nach Montageschritten. Datenset für einen Motortyp ist in der Tabelle 7.11 erfasst. Jeder Transportagent hat für jeden Motortyp solches Datenset.

Verlagerung der Information auf den Transportagenten entlastet vertikalen Kommunikationskanal erheblich, denn Transportauftrag besteht nur aus dem Motortyp, der auf jeweiligen Transportagent montiert wird. Nur wenn ein neuer Motorentyp montiert wird, ladet Transportbereichagent erst neues Datenset zu jeden Transportagentenspeicher herunter. Außerdem ist dynamische Optimierung der Montage ermöglicht.

Motortyp	Auswahl der Palettentyp	Montageschritt	Montageagenten	Empfangsfrequenzen	Senderfrequenz
		1	2;5;7	2452;2455;2457 MHz	2505
2	1;4	2451; 2454 MHz	2506		
3	3;6	2453; 2456 MHz	2507		
...		
Reparatur	Reparaturagent	2400 MHz			
Prüfung	Prüf Station	2500 MHz	-		
Entladen	Entlade Station	2501 MHz	-		
Transportagenten Pool		-	2502 MHz	-	

Tab. 7.11. Dateiset für einen Motortyp.

7.10 Ablauf Szenario der selbstorganisierende Montage ohne Optimierung des Verhandlungsprozesses

Das Ablaufszenario des selbstorganisierenden Montagesystem wird auf dem Beispiel von drei typischen Situationen innerhalb Kernsubsystems im folgendem beschreiben. Dieses Beispiel ist auf der Abbildung 7.14 graphisch präsentiert. Montagezyklus startet im Ausgang von Transportagenten Pool. Dort befindet sich Infrarot Kommunikationsschnittstelle.

Damit übermittelt Transportbereichagent den Motorentyp zum Transportagenten. Auf der Abbildung 7.14 ist das der Transportagent TA₂. Aufgrund des Motorentyps wählt der Transportagent passenden Palettentyp aus und aktiviert typbezogenen Montagezyklus. Transportagent TA₂ stellt eigene Empfänger auf erste Senderfrequenz, daher auf 2452 MHz und steuert Montageagent M₂ an.

Wenn er sich im Arbeitsbereich des Montageagenten befindet, startet er Dockingsequenz. Docking wird so lange ausgeführt bis Transportagent richtige Stellung einnimmt. Dann stoppt er, übermittelt Motor- und Palettentyp und wartet solange bis der Montagestatus geändert wird. Informationsaustausch zwischen Transport- und Montageagent passiert mittels Infrarot Schnittstelle. Aufgrund der ermittelten Informationen greift Montageagent typbezogenes Teil aus dem eigenen Lager und legt ihn auf die Palette hin.

Danach wird Qualitätskontrolle des Montageschrittes ausgeführt. Qualitätskontrolle wird mit Vision System, das auf dem Montageagenten Greifer eingebaut ist, gemacht.

Nach dem erfolgreichen Abschluss des ersten Montageschrittes, übermittelt Montageagent M_2 zum Transportagent TA_2 Montagestatus erhöht um eins.

Dies wird die Empfangsfrequenz der Transportagentenempfängern auf den nächsten Montageschritt umschalten, daher auf 2451 MHz. Transportagent steuert neuen Montageagent an. Dabei betrachtet er Montageagent M_2 als Hindernis.

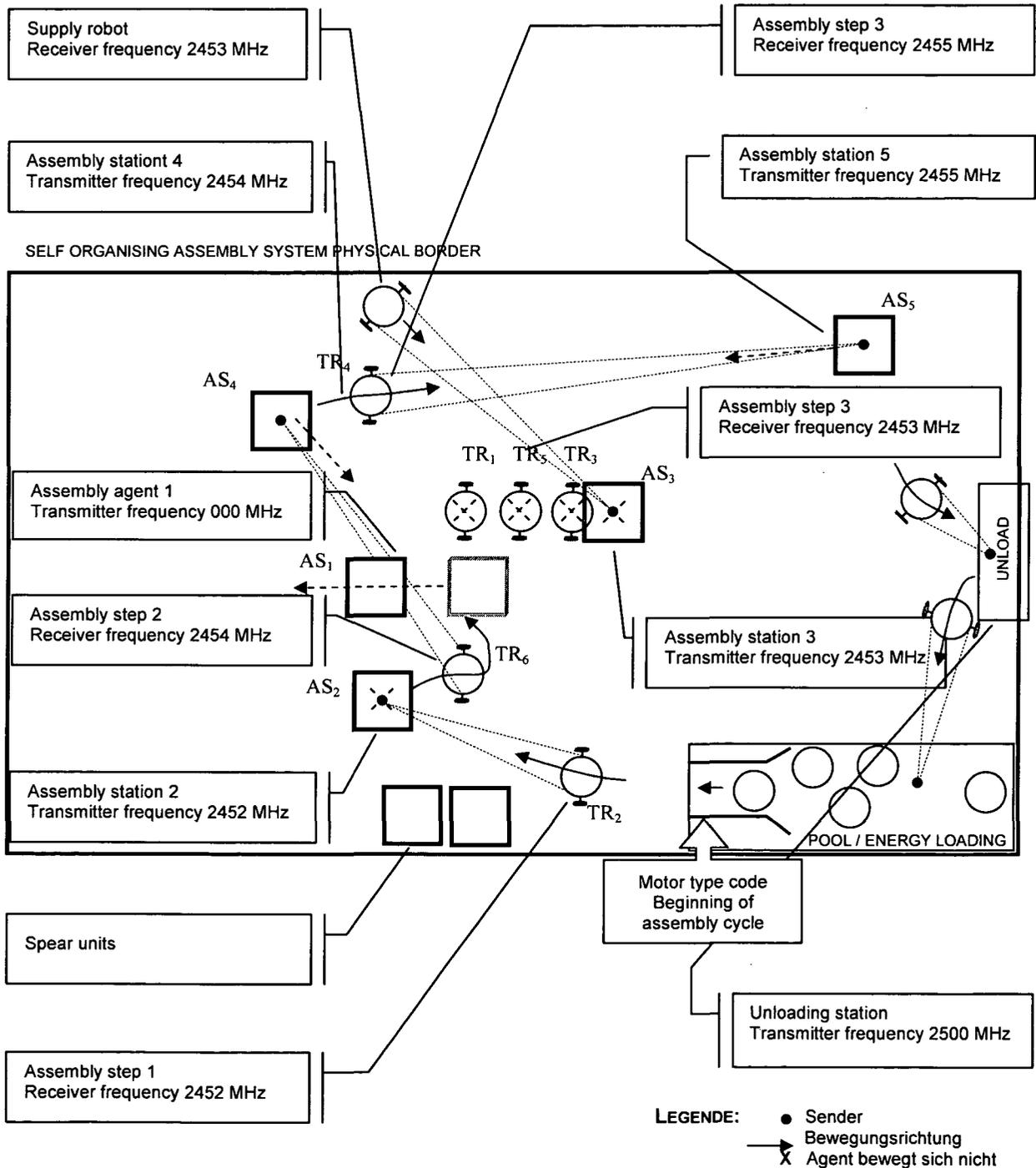


Abb. 7.14. Example of the biologic self-organising assembly system.

Wenn Montageschritt nicht erfolgreich beendet ist, erhöht Montageagent Montagestatus auf Reparatur. In dem Fall das die gesuchte Senderfrequenz nicht gefunden werden kann, schaltet der Transportagent automatisch auf alternative Senderfrequenz, die für entsprechenden Montageschritt gegeben ist. Für zweiten Montageschritt gibt es zwei passenden Frequenzen, 2451 und 2454 MHz. Das bedeutet innerhalb Kernsubsystems befinden sich insgesamt zwei Montageagenten, die diesen Montageschritt ausführen können. Das sind Montageagenten M_1 und M_4 . Wenn erste Frequenz nicht gefunden werden kann, schaltet Transportagent eigene Empfänger auf 2454 MHz.

Für jeden Montageschritt muss mindestens ein Montageagent aktiv sein um Montagesystem aufrecht zu erhalten. Dafür sorgt Montagebereichagent. Es gibt drei Gründe warum ein Montageagentensender nicht sendet. Das sind:

1. Montageagent ist ausgelastet: Wenn mehreren Transportagenten in der Warteschlange des Montageagenten sich befinden, schaltet er eigenen Sender aus und meldet Auslastungsgrad zu Montagebereichagent. Montagebereichagent aktiviert alternativen Montageagent. Auf der Abbildung 7.14 ist das der Fall von Montageagent M_3 . Wenn aber kein weiterer passender Montageagent zu Verfügung steht, muss Schedule Kontroller die Verteilung neuer Aufträge, die weiter betroffenen Montageagenten belasten, vorerst stoppen. Diese Maßnahme vermeidet Zerlegung der Montagestruktur. Wenn Transportagenten im Bereich eines Montageagenten schwärmen, wird Montagestruktur, wegen ständiger Umpositionierung den Montageagenten steigernd unstabiler werden. Dies muss unbedingt vermieden werden.
2. Montageagent ist ausgefallen: Auf der Abbildung 7.14 ist das Montageagent M_1 .
3. Montageagent ist absichtlich ausgeschaltet: Bei geringer Kapazität schaltet Montagebereichagent nicht mehr erforderliche Montageagenten aus. Ausgeschaltete Montageagenten werden sich automatisch aus der Montagestruktur zurückziehen.

Montageagent M_1 sendet kein Signal, denn er ist ausgefallen. Montageagent M_1 muss aber Montagebereichagent darüber sofort informieren. Weil der Montageagent M_1 , einzig aktiver Montageagenten, fähig war Montageschritt zwei durchzuführen, schaltet der Montagebereichagent alternativen Montageagent M_4 ein. Transportagent TA_2 richtet sich nach der Signalquelle und weicht Montageagent M_1 , aus der jetzt als Hindernis betrachtet wird. Ausgefallener Montageagent M_1 zieht sich aus der Montagestruktur automatisch zurück.

Neuer Montageagent M_4 sucht eigenen Platz innerhalb des Montagezyklus. Dafür schaltet er eigenen Empfänger ein. Empfänger empfängt nur die Frequenz, die für jeweiligen

Montageschritt vordefiniert ist, daher 2506 MHz. Jeder Transportagent, der vierten Montageschritt gerade ausführen muss sendet auf diese Frequenz solange er außerhalb des Arbeitsbereichs des gesuchten Montageagenten ist. Montageagent M_4 berechnet resultierende relative Orientierung zu der Transportagenten, entsprechend regelt er eigene Orientierung nach und bewegt sich schrittweise in die Richtung. Wenn erster Transportagent Montageagentenarbeitsbereich erreicht hat, stoppt er und schaltet auf Positionsoptimierung. Optimaler Platz wird nach dem minimalen Stillstand Gesetz gesucht. Minimaler Stillstand hängt aber von Transportagent Geschwindigkeit, minimal erlaubten Abstand zwischen Montageagenten und aktuellen Produktionskapazität ab. Minimaler Abstand hängt von der Größe der Montageagenten und Reichweite des eingebauten Manipulators ab. Reichweite vom Manipulator definiert Arbeitsbereich des Montageagenten. Um Kollision zweier Manipulatoren zu vermeiden muss minimal erlaubter Abstand größer als Arbeitsbereich der Montageagenten sein. Minimaler Abstand ist durch die Reichweite der Hindernissensoren auf dem Montageagent festgelegt. In dem Testfall beträgt sie 3 m. Fortbewegungs Geschwindigkeit des Transportagenten hängt von der gesamten Anzahl der aktiven Agenten und Stabilität der Montagestruktur. Maximale Geschwindigkeit ist durch die Konstruktion der Transportagenten bedingt und beträgt 1,8 m/s. Am Anfang des Montageprozesses ist Montagestruktur instabil, sodass die Transportagenten öfter Abweichungsmanöver in Kauf nehmen müssen. Nach dem die Montagestruktur formiert ist, steigt Transportagenten Geschwindigkeit und wird gleichmäßiger. Die Transportagenten bewegen sich eben mit Geschwindigkeiten, die nicht exakt berechnet werden können.

Beim jeweiligen inneren (Ausfälle, zu viel fehlerhaften Montageteile oder montierte Motoren) und/oder äußeren (wechselhafte Produktionskapazität) Störungen die auf aktuelle Produktionskapazität Einfluss haben, entstehen neue Engpässe oder Überkapazitäten. Die Engpässe werden durch das Einschließen neuer Montageagenten vermieden. Überkapazitäten dagegen durch Ausschaltung überschüssiger Montageagenten. Nach dem minimalen Stillstand Gesetz versuchen neue/gebliebene Montageagenten eigene optimale Plätze innerhalb bestehender Montagestruktur zu finden. Damit wirken sie auf alle anderen Montageagenten, die sich ebenso auf die neue Situation anpassen müssen.

Diese dynamische Restrukturierung der Montagestruktur findet beim jeder Veränderung der Anzahl der beteiligten Montageagenten automatisch statt und endet wenn alle Montageagenten neuen optimalen Platz erreicht haben. Damit bilden sie eine flexible Montagestruktur.

Wenn es für alle Montageschritte mindestens zwei Montageagenten gibt, entfesseln sich das Montagesystem auf zwei parallelen Montagestrukturen. Innerhalb Kernsubsystems koexistieren zwei Montagesysteme.

7.11 Ablaufszenario für die Agentenklassen

Der oben beschriebene Verlauf bildet eine Vorstellung von Abläufen innerhalb Kernsubsystems. Im weiteren sind Abläufe für jede Agentenklasse in Flussdiagrammen strukturiert. Somit bilden sie eine Basis für Entwicklung von Steuerungsalgorithmen.

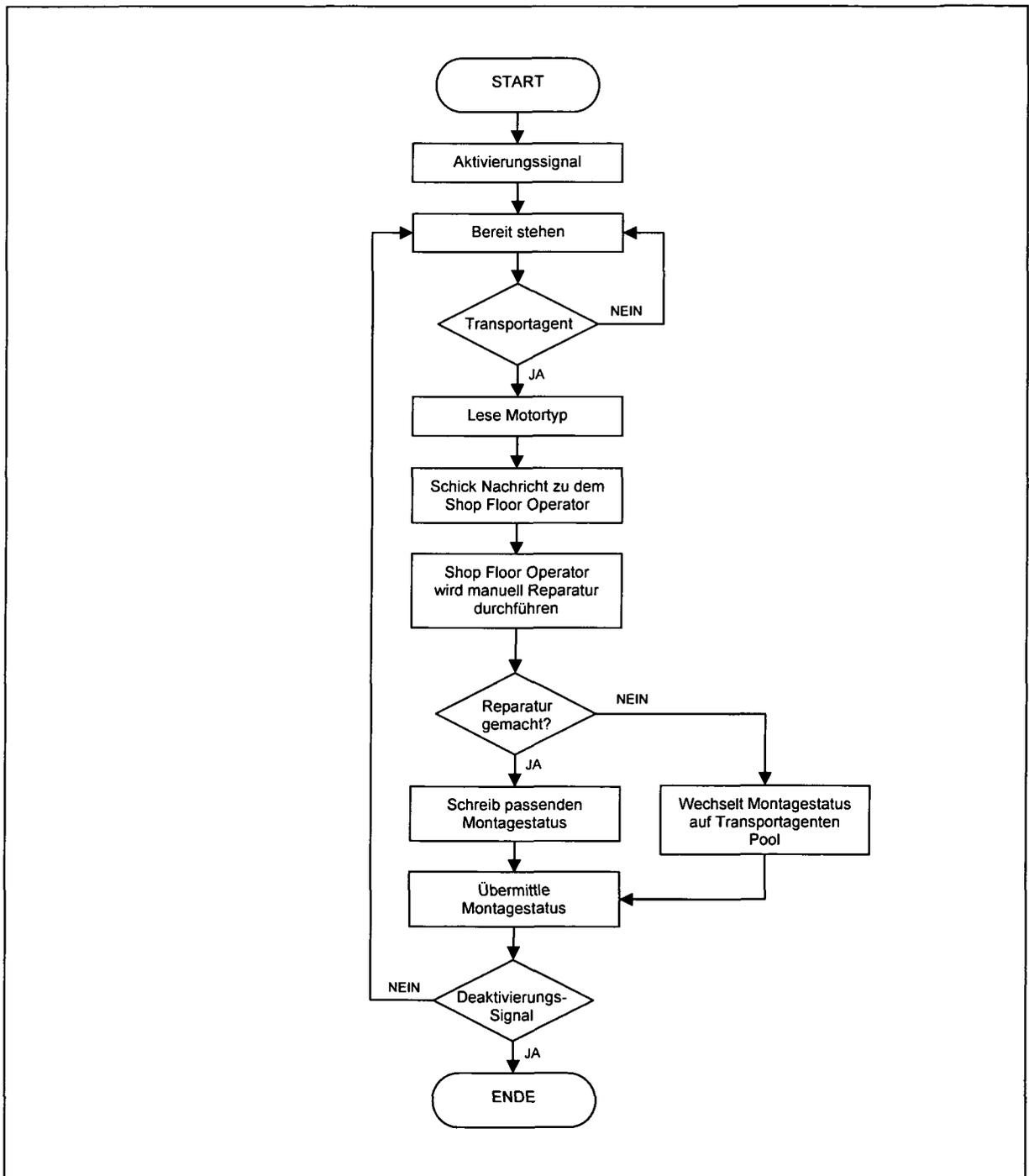


Abb. 7.15. Flussdiagramm von Reparaturagent Ablauf.

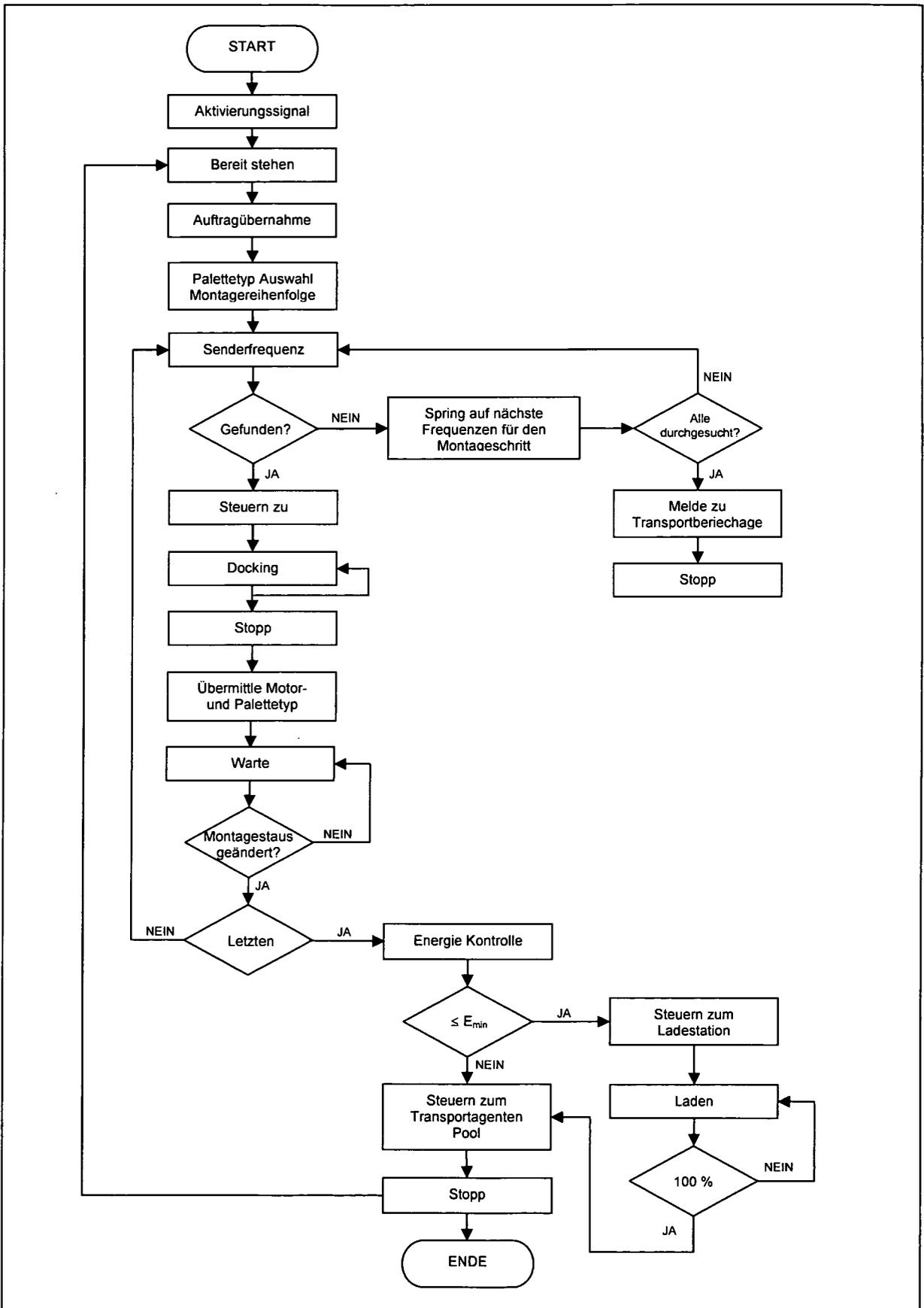


Abb. 7.16. Flussdiagramm von Transportagent Ablauf.

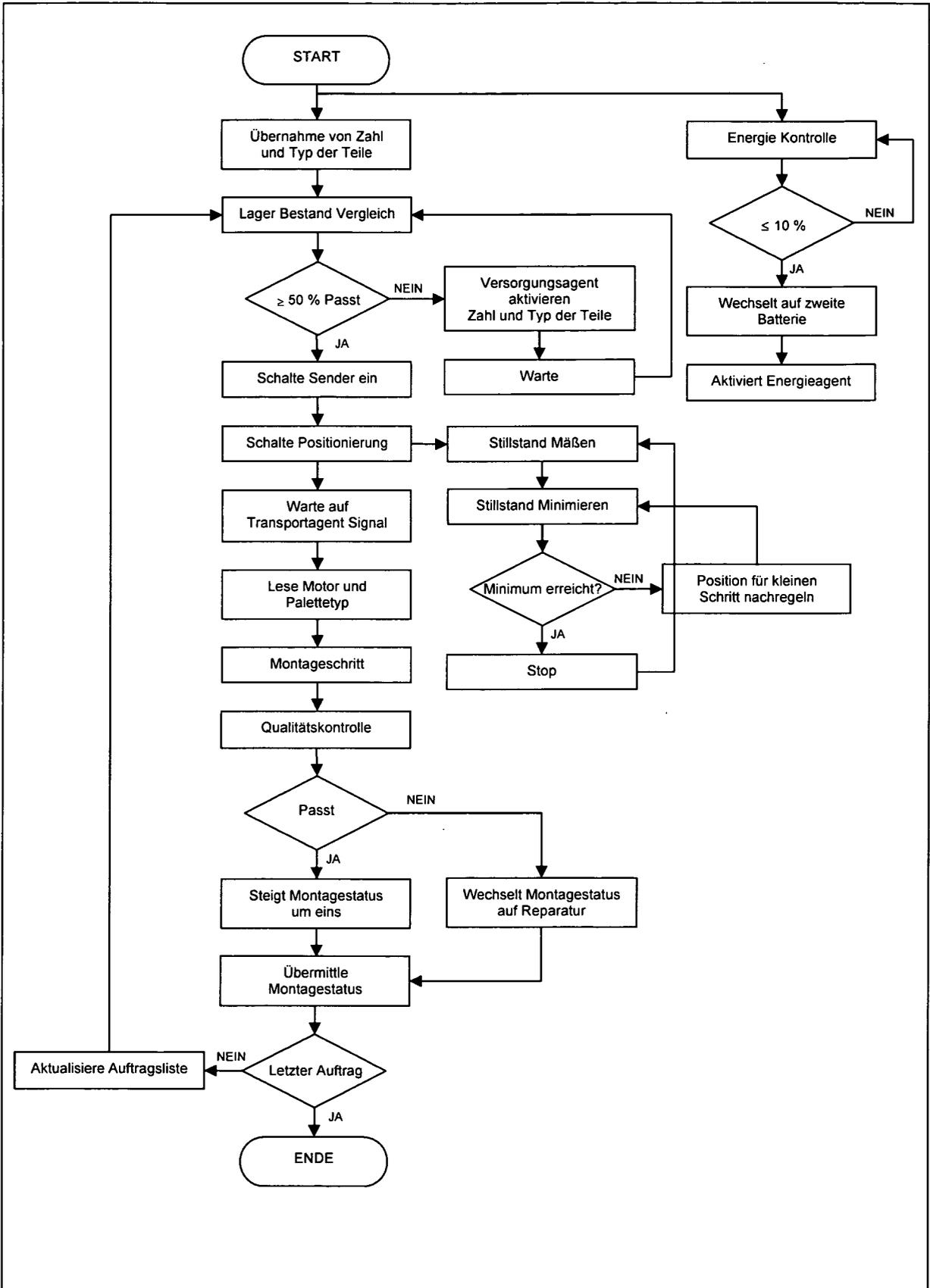


Abb. 7.17. Flussdiagramm von Montageagent Ablauf.

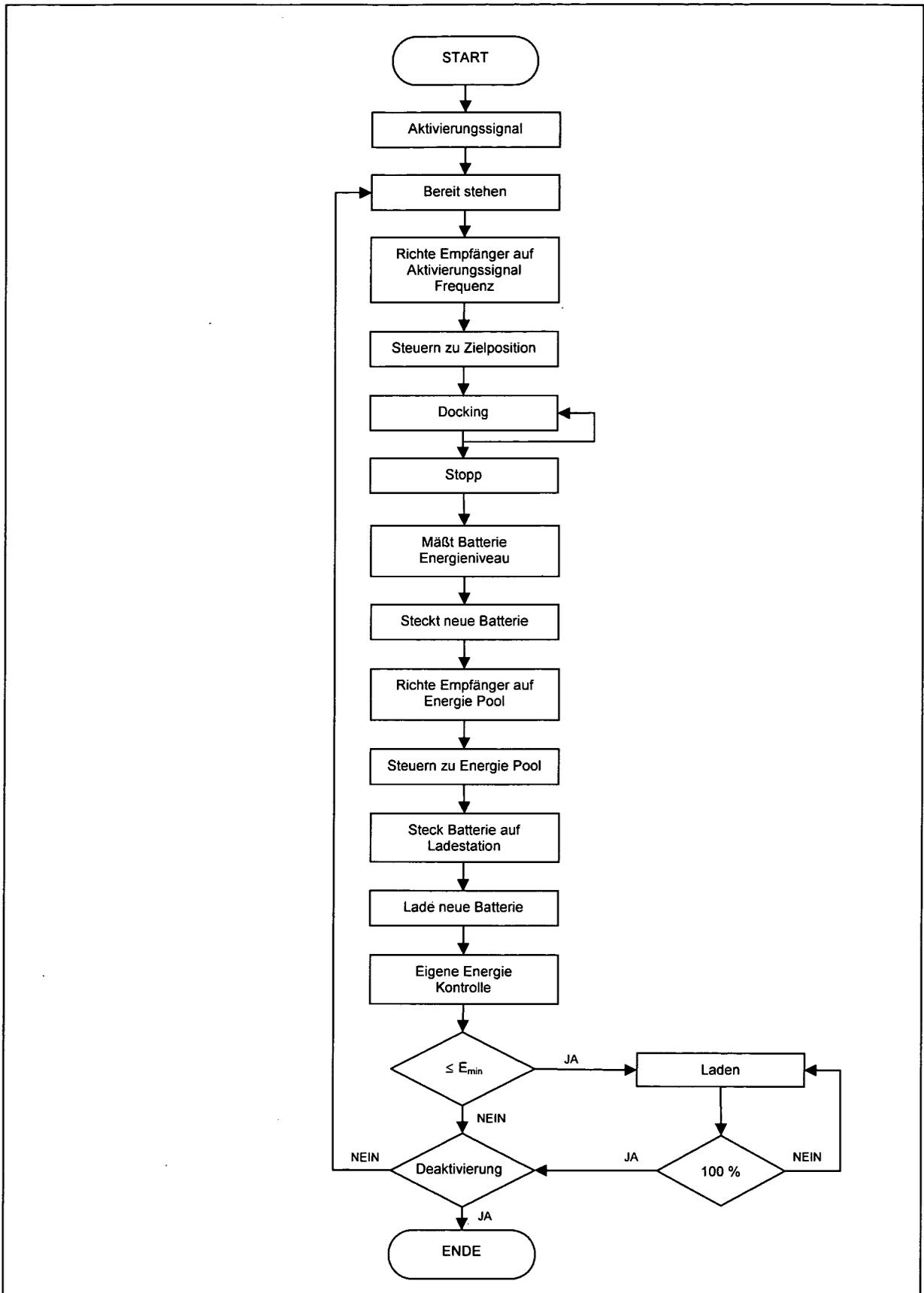


Abb. 7.18. Flussdiagramm von Energieagent Ablauf.

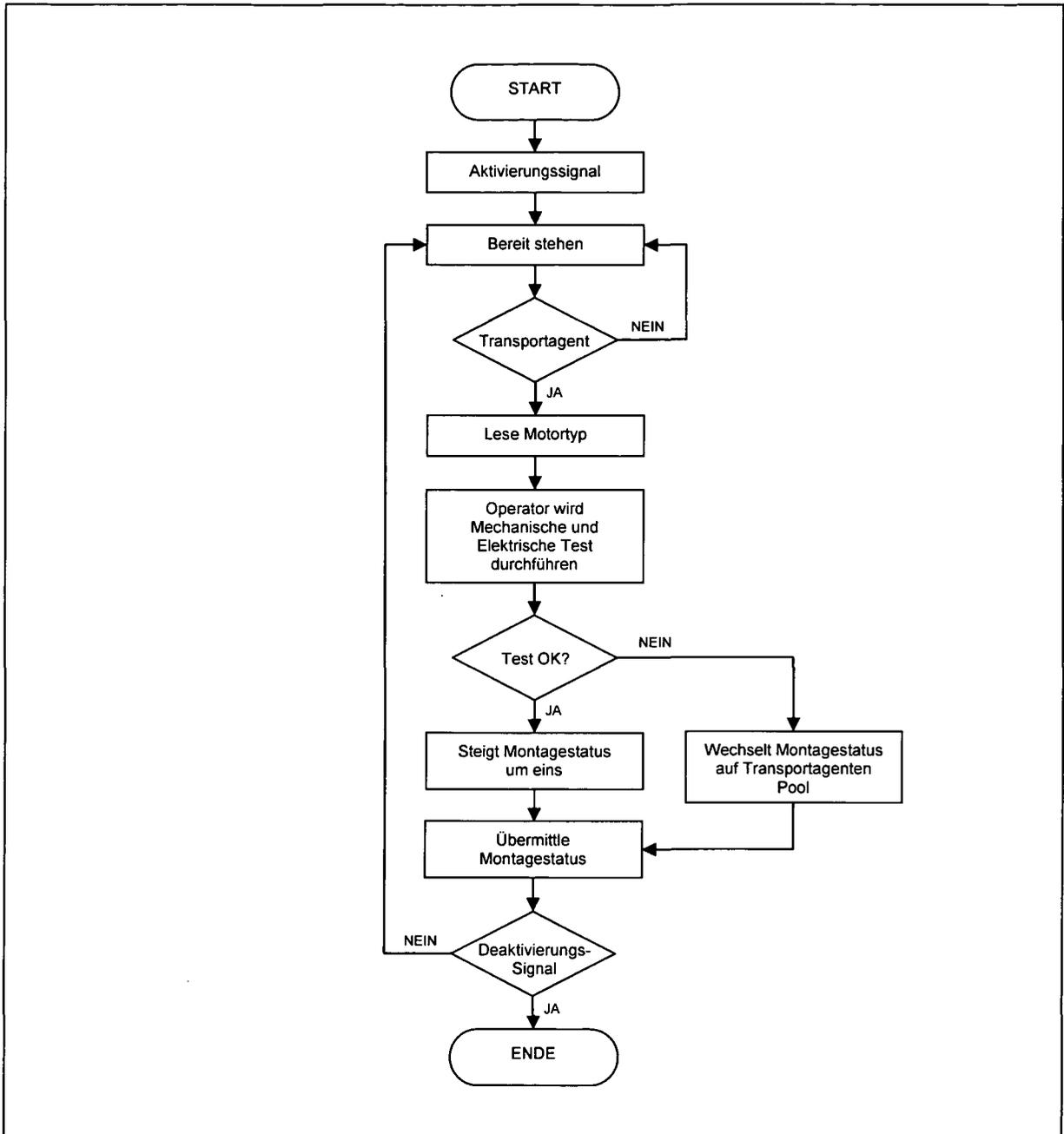


Abb. 7.19. Flussdiagramm von Prüfstation Ablauf.

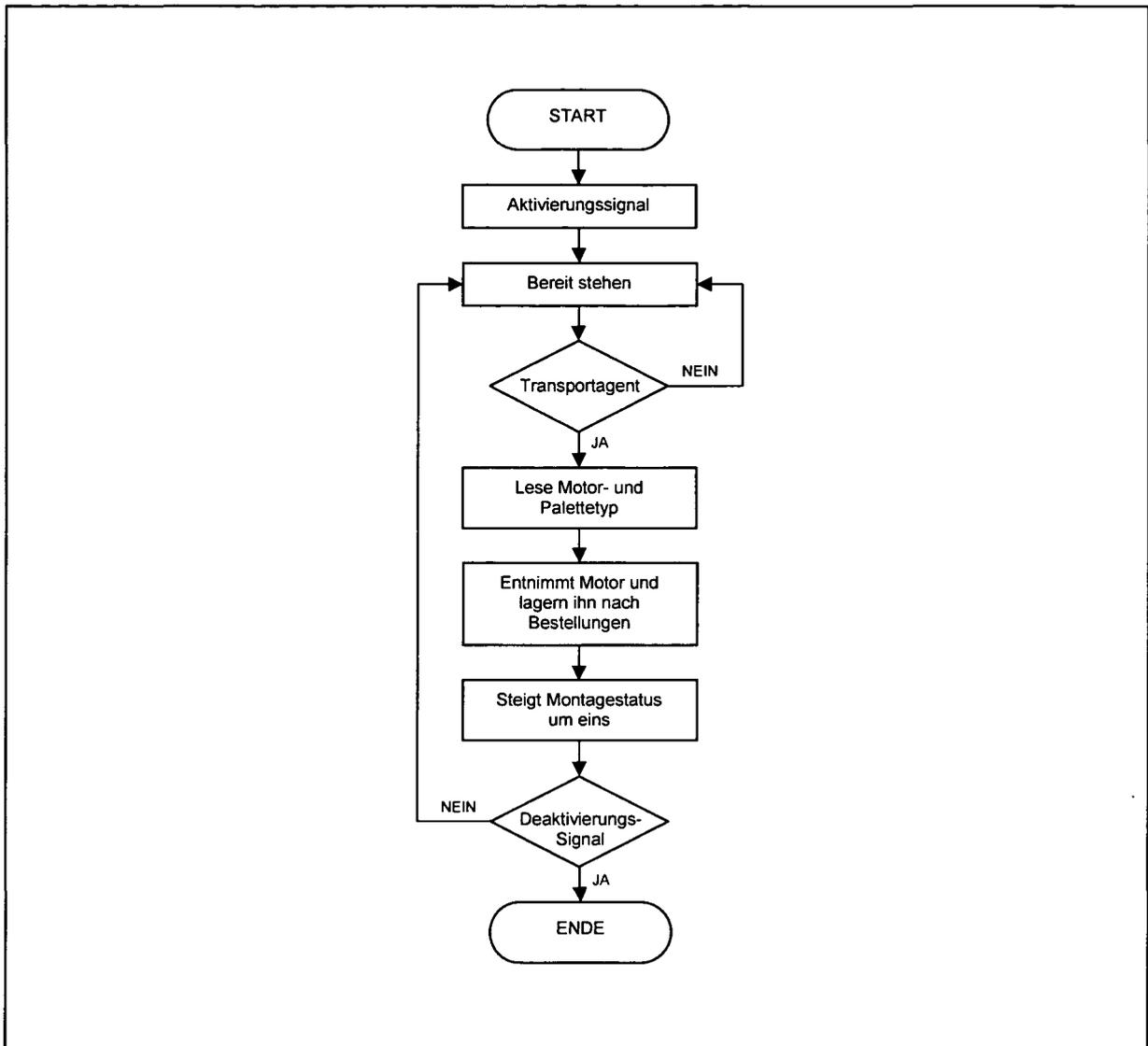


Abb. 7.20. Flussdiagramm von Entlade Station Ablauf.

7.12 Koordination zwischen Agenten

Flussdiagramme strukturieren Abläufe für jede Agentenklasse. Diese Abläufe stellen eine Kombination von indirekter (Steuern zu Zielposition, Funktionskontrolle, Position Optimierung) und direkter Kooperation zwischen Agentenklassen dar (Docking, horizontaler Informationsaustausch, Abfahrt von Arbeitsbereich eines anderen Agenten, Energieerhaltung). Aus diesen Handlungen entstehen Konfliktsituationen. Diese sind Teil der normalen Abläufe von Transport-, Montage-, Versorgungs- und Energieagentenklassen. Deswegen müssen sie genau geregelt sein. In folgenden werden Kooperation und Konfliktsituationen in der Form von deskriptiven Algorithmen definiert.

Transportagentenklasse

- *Auftragübernahme*

Start

```

{
  wenn {Transportagent innerhalb Transportagenten Pool ist}
  dann {Schalte Infrarot Kommunikation Schnittstelle}
  wenn {Motortyp existiert}
  dann {Schalte Suchmodus}
  sonst {Warte}
}

```

Ende

- *Suchmodus*

Start

```

{
  Suchen      {Suche Senderfrequenz nach vordefinierte Reihenfolge für entsprechende
               Montagestatus}
              wenn {Senderfrequenz gefunden ist}
              dann {Schalte auf Zielpositionierung um}
              sonst {Wiederholt Suchvorgang}
              wenn {Keine Senderfrequenz nach 10 mahl gefunden ist}
              dann {Stopp, melde es zum Montagebereichagent}
}

```

Ende

- *Zielpositionierung* (Gleich für Energie- und Versorgungsagentenklasse)

Start

```

{
  Einstellen  {Stelle Eigene Richtantenne auf Anfangsposition}
  Rotieren    {Rotiert Richtantennen parallel zu einander}
  Maßen       {Elektromagnetische Energie}
              wenn {Sie gleich ist}
              dann {Stopp}
  Rotieren    {Rotiert Richtantennen auf einander}
  Maßen       {Elektromagnetische Energie}
              wenn {Elektromagnetische Energie Maximum erreicht hat}
              dann {Stopp, schalte Zielansteuerung}
}

```

}
Ende

- *Zielansteuerung*

Start

```

{
  Lesen      {Elektromagnetische Energie}
  Maßen     {Winkel}
  Berechnung {Winkelgeschwindigkeit, Distanz zum Ziel nach Trigonometrische und
              Elektromagnetische Methode, Lineargeschwindigkeit} /Gleichungen 3.xy, 5.xy, 7.xy/
              wenn {Winkelgeschwindigkeit < Maximum}
              dann {Fahre mit berechnete Winkelgeschwindigkeit}
              sonst {Fahre mit Maximale Winkelgeschwindigkeit}
              wenn {Lineargeschwindigkeit < Maximum}
              dann {Fahre mit berechnete Lineargeschwindigkeit}
              sonst {Fahre mit Maximale Lineargeschwindigkeit}
  Abtasten  {Hindernissensorwerte wird in gleichmäßigen Zeitspannen gelesen}
              wenn {Sensorwerte in der linke Seite erhöht sind}
              dann {Bieg nach recht ab, schalte Zielpositionierung}
              wenn {Sensorwerte in der rechte Seite erhöht sind}
              dann {Bieg nach links ab, schalte Zielpositionierung}
              wenn {Sensorwerte in der vordere Seite erhöht sind}
              dann {Macht Vergleich mit dem Zielorientierung}
              wenn {Das Ziel befindet sich gerade aus, schalte auf Dockingmodus um}
              sonst {Bieg nach recht ab, schalte Zielpositionierung }
              wenn {Senderfrequenz nicht mehr gefolgt werden kann}
              dann {Stopp, Schalte auf Suchmodus ein}
}
Ende

```

- *Docking*

Start

```

{
  Umschalten {Schalte Hindernissensoren auf 50 cm Reichweite}
              wenn {Sensorwerte in der vordere Seite erhöht sind}
              dann {Stopp, Macht Vergleich mit der Distanz zum Ziel}
              wenn {Beide Werte gleich sind}
              dann {Rotiert sich auf der Stelle}
              wenn {linke Sensorenwerte erhöht sind}
              dann {Stopp}
              wenn {Vordere Sensor meldet keine Werte}
}

```

```

        dann {Fahrt auf dem Radius nach recht}
            wenn {Infrarot Sensor erregt ist}
                dann {Stopp, schalte Montagemodus}
            sonst {Stopp, wiederhol Schleife}
    sonst {Fahre nach Links auf dem gemessenen Radius}
    wenn {Rechte Sensoren keine erhöhte werte melden}
    dann {Stopp, dreht sich so lange bis Elektromagnetische Energie in beiden
        Empfänger gleich ist, starte Docking}
    sonst {fortfahren, nach 1 Minute schalte Suchmodus}
}
Ende

```

- *Montagemodus*

```

Start
{
    Sende      {Motortyp, Palettetyp, Montagestatus}
    wenn      {Montagestatus geändert ist}
    dann      {Schalte Suchmodus}
    sonst     {Warte}
}
Ende

```

- *Energieversorgung* (Gleich für Energie- und Versorgungsagentenklasse)

```

Start
{
    wenn      {Energie Niveau  $E_{min}$  erreicht}
    dann      {Die aufgenommene Auftrag zum Ende führen, start die Ladestationssuchfolge}
    sonst     {fortfahren}
}
Ende

```

Der Grenzwert E_{min} ist so definiert, dass jeder Agent noch einen ganzen Montagezyklus beenden kann und dass er noch rechtzeitig Ladestation erreichen kann. Dazu gab es noch 6% Überschuss wegen Batteriekapazitätsabweichungen.

Montageagentenklasse

• *Auftragübernahme*

Start

```

{
  Lesen  {Zahl und Typ der Teilen, Motortypen und Montagestatus vorhanden sind}
  wenn  {Montageagentenstatus von Passiv auf Aktiv gewechselt ist}
  dann  {Warte}
        wenn  {Entsprechende Zahl und Typ der Teilen in Lager ist}
        dann  {Schalte auf Positionierung um}
  sonst {Aktualisiere bestehende Liste}
        wenn  {Zahl der Teilen unter 50 % von aktuellen Aufträgen liegt}
        dann  {Sende Aktivierungssignal, Zahl und Typ der Teilen}
}

```

Ende

• *Positionierung*

Start

```

{
  Einschalten  {Schalte Empfänger ein}
  Maßen        {Elektromagnetische Energie}
  Rotieren     {Rotiert Richtantenne}
              wenn  {Elektromagnetische Energie Maximum erreicht hat}
              dann  {Stopp}
  Rotieren     {Agent dreht sich}
              wenn  {Unterschied zwischen Agent und Antenne Winkel 0 ist}
              dann  {Stopp}
  Bewegung     {Bewegt sich schrittweise gerade aus}
  Abtasten     {Hindernissensorwerte wird in gleichmäßigen
               Zeitspannen gelesen}
              wenn  {Sensorwerte erhöht sind}
              dann  {Stopp, schalte Positionierung}
              sonst {fortfahren}
  Messen       {Zeit zwischen zwei Montagen}
              wenn  {Zeit kleiner wird}
              dann  {Wiederholt Bewegung}
              sonst {Schritt zurück, wiederholt Positionierung}
}

```

Ende

- *Montageschritt*

Start

```

{
  Lese  {Motortyp, Palettetyp, Montagestatus}
        wenn {Motortyp und Montagestatus passt}
        dann {Nimmt passenden Teil aus dem Lager und platziert ihm auf der Palette}
            Qualitätskontrolle    {Vision System Checkt Montageschritt}
                                   wenn {Montageschritt gut war}
                                   dann {Erhöhe Montagestaus um eins, schicke es zu
                                       Transportagent}
                                   sonst {wechsle Montagestatus auf Reparatur und übermittle es
                                       zu Transportagent}
        sonst {wechsle Montagestatus auf Reparatur und übermittle es zu Transportagent}
}

```

Ende

- *Energieversorgung*

Start

```

{
  wenn  {Energie Niveau von laufende Batterie 10% erreicht}
  dann  {Wechsle auf zweite Batterie, sende Energie Aktivierungssignal}
  sonst  {fortfahren}
}

```

Ende

Energieagentenklasse

- *Auftragübernahme*

Start

```

{
  Lese  {Aktivierungssignal}
        wenn {Aktivierungssignal existiert}
        dann {
            Einschalten    {Schalte Empfänger auf entsprechende Frequenz ein}
            wenn {
                dann {Wechsle auf zweite Batterie, Aktiviere Energieagent}
            }
        }
  sonst {fortfahren}
}

```

Ende

- *Zielansteuerung* (Gleich für Energie- und Versorgungsagentenklasse)

Start

```

{
  Lesen          {Elektromagnetische Energie}
  Maßen         {Winkel}
  Berechnung     {Winkelgeschwindigkeit, Distanz zum Ziel nach Trigonometrische und
                  Elektromagnetische Methode, Lineargeschwindigkeit} /Gleichungen 3.xy, 5.xy, 7.xy/
                wenn {Winkelgeschwindigkeit < Maximum}
                dann {Fahre mit berechnete Winkelgeschwindigkeit}
                sonst {Fahre mit Maximale Winkelgeschwindigkeit}
                wenn {Lineargeschwindigkeit < Maximum}
                dann {Fahre mit berechnete Lineargeschwindigkeit}
                sonst {Fahre mit Maximaler Lineargeschwindigkeit}
  Abtasten      {Hindernissensorenwerte wird in gleichmäßigen Zeitspannen gelesen}
                wenn {Sensorwerte in der linke Seite erhöht sind}
                dann {Bieg nach recht ab, schalte Zielpositionierung}
                wenn {Sensorwerte in der rechte Seite erhöht sind}
                dann {Bieg nach links ab, schalte Zielpositionierung}
                wenn {Sensorwerte in der vordere Seite erhöht sind}
                dann {Macht Vergleich mit dem Zielorientierung}
                wenn {Das Ziel befindet sich gerade aus, schalte auf Dockingmodus um}
                sonst {Bieg nach recht ab, schalte Zielpositionierung}
                wenn {Senderfrequenz nicht mehr gefolgt werden kann}
                dann {Stopp, schalte Zielansteuerung}
}

```

Ende

- *Docking* (Gleich für Energie- und Versorgungsagentenklasse)

Start

```

{
  Umschalten    {Schalte Hindernissensoren auf 50 cm Reichweite}
                wenn {Sensorwerte in der vordere Seite erhöht sind}
                dann {Stopp, Mach Vergleich mit der Distanz zum Ziel}
                wenn {Beide Werte gleich sind}
                dann {Rotier auf der Stelle}
                wenn {linke Sensorenwerte erhöht sind}
                dann {Stopp}
                wenn {Vordere Sensor meldet keine Werte}
                dann {Fahrt auf dem Radius nach rechst}
                wenn {Infrarot Sensor erregt ist}
}

```

```

        dann {Stopp, schalte Montagemodus}
      sonst {Stopp, wiederhol Schleife}
    sonst {Fahre nach Links auf dem gemessenen Radius}
  wenn {Rechte Sensoren keine erhöhte werte melden}
  dann {Stopp, dreht sich so lange bis Elektromagnetische Energie in beiden
        Empfänger gleich ist, starte Docking}
  sonst {fortfahren}
}
Ende

```

Versorgungsagentenklasse

- *Auftragübernahme*

```

Start
{
  Lese {Senderfrequenz, Zahl und Typ der Teile}
  Senden {
    Einschalten {Schalte Empfänger auf entsprechende Frequenz ein}
    wenn {
      dann {Wechsle auf zweite Batterie, Aktiviere Energieagent}
    }
  }
  sonst {fortfahren}
}
Ende

```

Optimierung von Abläufen

Die im Kapitel 5 entwickelte Agentenarchitektur ist schließlich die Basis für das im vorhergehenden Kapitel beschriebene selbstorganisierende Montagesystem.

Die Integration von zwei Ebenen, der der Agenten und des Systems gewährleistet die Funktionsfähigkeit des Montagesystems.

Im Montagesystem ist die Auswahl der nächsten Montagestation vom aktuellen Montageschritt sowie von der Suchreihenfolge innerhalb des Pooles geeigneter Montagestationen für nächsten Montageschritt abhängig. Dabei ist der Auslastungsgrad der Stationen unbekannt. Nur wenn eine Station voll ausgelastet ist, schaltet sich der Sender der Station aus. In der Folge schalten, die von der Station entfernt befindlichen Transportagenten, erneuert den Suchvorgang ein und verfolgen ein anderes Signal. Die Agenten, die in der Warteschlange der Station sind, verfügen nicht über die eigene Wartezeit. Genau so wenig über die Möglichkeit einer früheren Ausführung des Montageschrittes auf alternativen Montagestationen. Dieser Zustand hat folgende Nachteile:

- Montagestationen montieren nach dem Prinzip First in First out. Dabei ist es nicht möglich nach dem Prioritätsgrad Warteschlangen umzustrukturieren.
- Da der Auslastungsgrad nicht bekannt ist, kommt es dazu dass einige Stationen öfter Montageteile bestellen müssen, ungeachtet dessen dass alternative Stationen noch ausreichend besorgt sind.
- Unregelmäßige Auslastung der Montagestationen gefährdet Kapazität und Stabilität der Montagestruktur weiter.
- Zentralsystem hat Information über ausstehende Dauer der Montage nicht.
- Transportagenten haben beschränkte Möglichkeit eigenen Ablauf zu optimieren.

8.1 Verhandlung als Scheduling in realer Zeit

Oben angeführte Nachteile lassen sich durch Erweiterung des Verhandlungsmechanismus innerhalb der horizontalen Kommunikation beheben. Der Verhandlungsmechanismus dient dazu in der echten Zeit Montagepläne zu generieren.

Der Verhandlungsmechanismus wird aus zwei Perspektiven aufgebaut;

- Montageteile zentrierte Verhandlung – wird von Lagerbestand der Montagestation bestimmt. Findet zwischen Montagestationen und Versorgungsagenten statt. Ziel der Verhandlung ist es, dass das Lager der Montagestation richtige Mengen und Kombination von Montageteilen hat. Diese Verhandlungen ziehen Montageteile durch das Montagesystem.
- Montagestation zentrierte Verhandlung - Findet zwischen Transportagenten und Montagestationen statt. Erfolgt jedes Mal sobald der Montagestatus des Transportagenten geändert wird. Ziel der Verhandlung ist es, diejenige Montagestation vorzumerken, die beste Konditionen angeboten hat. Diese Verhandlung schiebt Transportagent (Produkt) durch das Montagesystem.

Vorteil dieser Methode ist es, dass der Zeitpunkt und damit der Inhalt der Verhandlungen verändert werden kann, sodass das Montagesystem optimale Bedienungen erreichen kann.

Je Verhandlungen früher anfangen, desto ist das optimale Schedule schwieriger zu finden und die tatsächliche Realisierung der ausarbeiteten Schedule unwahrscheinlicher. Da der Entscheidungshorizont groß ist, wird mehr Informationen benötigt was wiederum Robustheit der Entscheidungen gefährdet.

Beim Feststellen von Verhandlungszeitpunkt ist von der im Kapitel 6 aufgestellten Regel auszugehen: Die Ausführung der Aufgaben soll so spät wie möglich und nur so früh wie nötig geplant werden.

Aus der Sicht des Verhandlungszeitpunktes gibt es prinzipiell zwei unterschiedliche Zeitebenen:

- Verhandlung fängt vor der Montage an – Diese Methode ist üblich für die meisten vorhandenen Produktionssysteme. Hierbei geht es um das Festlegen des Ablaufs auf der Ebene einer Schicht oder für bestimmte Lößgrößen. Bei der Auftragsübernahme bekommt der Transportagent eine Liste aller Montagestationen die er ansteuern muss. Montagestationen dagegen bekommen Aufträge für die Montage für vordefinierte Lößgrößen.
- Verhandlung wird initiiert bei Übernahme eines Auftrags – Transportagent übernimmt Motortyp und Lieferungszeit von Scheduler. Damit ist Transportagent beauftragt und er

verhandelt mit geeigneten Montagestationen über den Montageablauf. Transportagent verhandelt für bestimmten Anzahl der Montageschritte voraus. Dieser Ansatz wird im weiteren Montageschritt Reservierung genannt. Montagestationen bekommen Aufträge für die Montage für vordefinierte Lößgrößen.

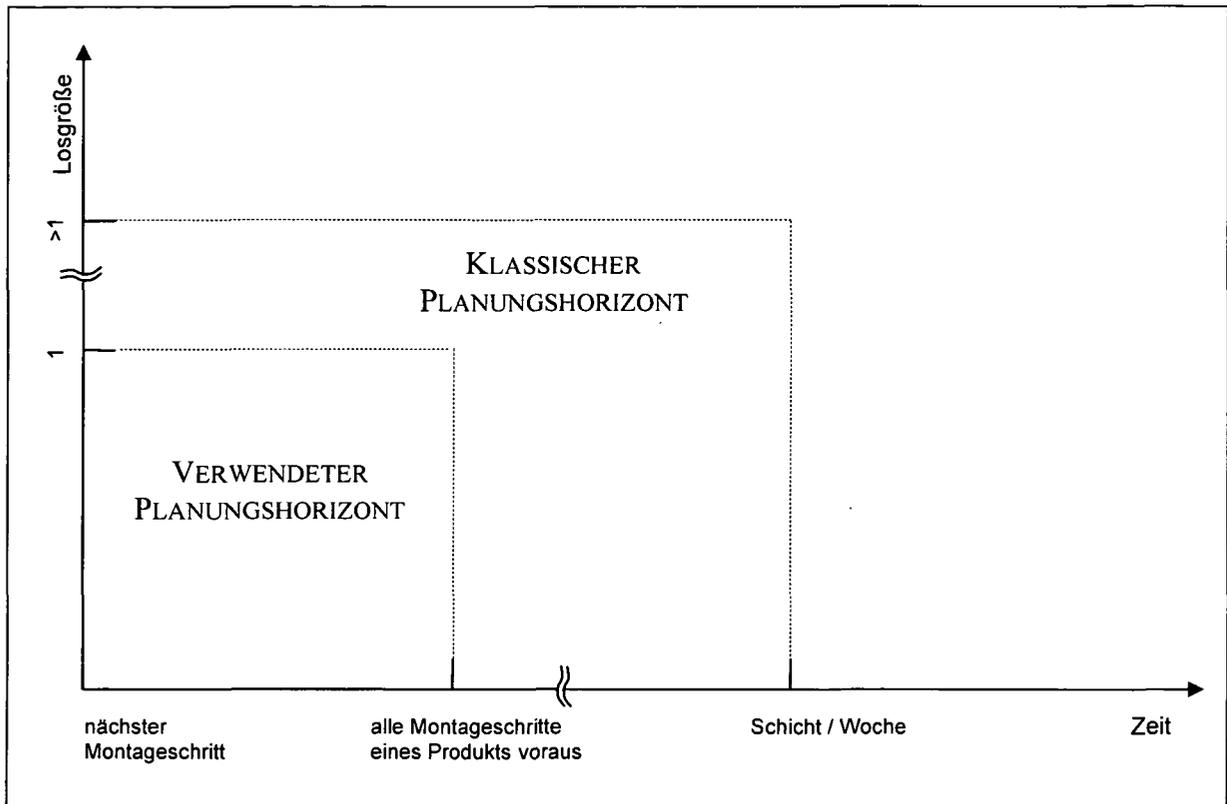


Abb. 8.1. Planungshorizont.

8.2 Verhandlungsmethode

Eine Verhandlung ist ein auf gegenseitigem Informationsaustausch beruhender Prozess der Entscheidungsfindung zwischen zwei oder mehreren Agenten, die eine Einigung zum Ziel hat.

Der Prozess besteht aus einer Abgabe von Anfragen und Angeboten so lange, bis eine Einigung erzielt oder von mindestens einem der beteiligten Agenten abgebrochen wurde. Da die Verhandlung zielorientiert und das Ziel im voraus bekannt ist, erfolgt es autonom. Der vom Montagesystem verwendete Verhandlungsansatz basiert auf der Vertragmethodologie (Smith, 1980; Kraus, 1997).

Transportagent meldet sich bei jenen Montagestationen an, die das Potential haben, nächsten Montageschritt durchzuführen. Angefragte Montagestationen werden ein Angebot generieren und ihn zurück an den Transportagenten schicken. Sobald Transportagent

Angebote ausgewertet hat, wird er diejenige Montagestation, die beste Konditionen angeboten hat, reservieren. Das ist der Kern des Vertragsprotokolls. Das Vertragsprotokoll besteht aus drei Phasen; Anfrage, Angebot und Reservierung, wie auf der Abbildung 8.2 abgebildet ist.

Montagezyklus fängt im Transportagenten Pool. Dort bekommt Transportagent von Scheduler den Motortyp und den Fälligkeitstermin der Montage.

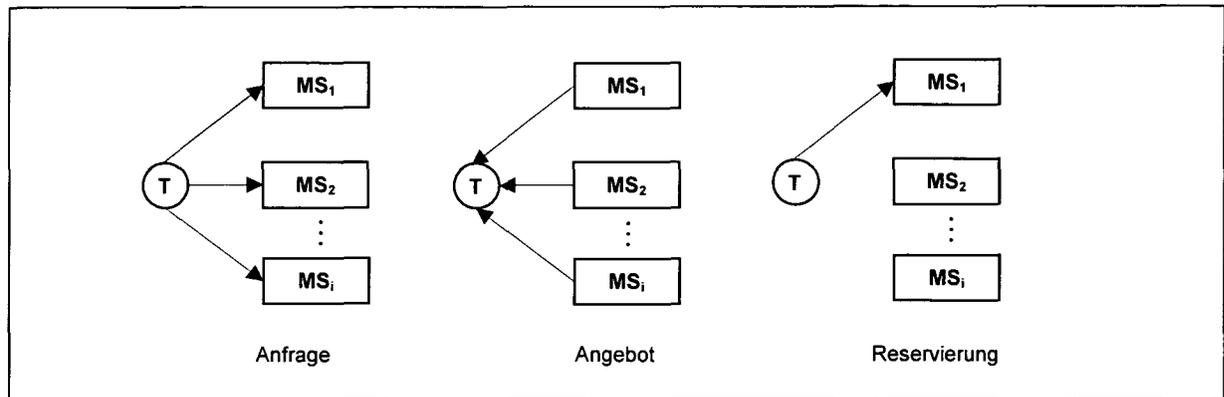


Abb. 8.2. Drei Phasen des Verhandlungsprotokolls.

Aufgrund der Motortypen und des aktuellen Montageschrittes wählt der Transportagent passende Montagestationen, wie im Kapitel 7 dargestellt. Fälligkeitstermin der Montage entspricht dem Lieferungstermin in der Entladestation. Alle Zeitangaben sind in der so genannten Systemzeit gegeben.

TRANSPORTAGENT	
ANFRAGE	ZWECK DER INFORMATION
Motortyp Code	Kontrollwert
Montageschritt	Aufgrund der Information weiß die angefragte Montagestation ob sie tatsächlich für die angekommene Anfrage zuständig ist. Montagestation kontrolliert ob sie benötigte Montageteile lagernd hat und berechnet die Dauer der Montage.
Fälligkeitstermin der Montage	Entspricht dem Lieferungstermin in der Entladestation. Der Termin entspricht dem Prioritätsgrad der Montage. Je kleiner der Fälligkeitstermin der Montage ist, desto größer ist der Prioritätsgrad.
Fälligkeitstermin des Angebots	Ein begrenztes Zeitintervall vom Beginn der Verhandlung zum letzten Antworttermin, erlaubt jeder Montagestation, die Angebote anderer Transportagenten in eine Reihenfolge zu sortieren. Dadurch werden überlappende Anfragen minimiert.

Tab. 8.1. Komponenten der Anfrage nach Reihenfolge und ihr Zweck.

„Die Systemzeit wird im astronomischen Kalender auf einen Zeitpunkt justiert und dieser Zeitpunkt wird als Nullpunkt definiert. Ab diesem Nullpunkt werden alle Zeiten in Sekunden gerechnet und als Integervariable betrachtet. Diese Methode der Zeitrechnung hat im Vergleich zur Astronomischen Zeitrechnung folgende Vorteile: Einfache Berechnung, wenig Speicherplatz beim Speichern wird benötigt und einfache Datenmanipulation.“ (Katalinic, 1990).

Die Anfrage des Transportagenten besteht aus: Montageschritt, Motortyp, Fälligkeitstermin der Montage, Fälligkeitstermin des Angebots. In der Tabelle 8.1 sind die Komponenten der Anfrage und deren Zweck erläutert.

Der Fälligkeitstermin des Angebots (FT_i) entspricht einem fixen Prozentsatz der erwarteten Zeit, die benötigt wird, um die Montage auf der gegenwärtigen Montagestation aufzustellen und zu verarbeiten. Dadurch ist gesichert, dass der neuste Stand erfasst wird.

Angebot der Montagestationen besteht aus: Geschätzter Endtermin des Montageschrittes und Fälligkeitstermin für die Reservierung und ist in der Tabelle 8.2 erfasst.

MONTAGESTATION	
ANGEBOT	ZWECK DER INFORMATION
Geschätzter Endtermin des Montageschrittes	Transportagent reserviert die Montagestation, die kürzesten Endtermin angeboten hat.
Fälligkeitstermin für die Reservierung	Falls Transportagent sich bis zum definierten Termin nicht meldet, löscht Montagestation das Angebot und korrigiert weitere Angebote.

Tab. 8.2. Komponenten des Angebots nach Reihenfolge und dessen Zweck.

Geschätzter Endtermin des Montageschrittes ist die Basis für die Wahl der nächsten Montagestation. Geschätzter Endtermin des Montageschrittes der Montagestation MS_i ergibt sich aus der Summe der:

$$\text{gem MS}_i = \text{Systemzeit} + \text{Zeit um aktuelle Montage zu fertigen} + \sum \text{Aufträgen in der Warteschlange} + \sum \text{Reservierte Aufträge die noch nicht in der Warteschlange sind} + \text{Zeit um die angefragte Montage zu machen}$$

Fälligkeitstermin für die Reservierung muss kleiner als der nächste Fälligkeitstermin des Angebots sein:

$$FR_{MS_i}^{T_i} < \frac{T_s + FT_{i+1}}{2}$$

Dies ist notwendig, da die Montagestation MS_i weitere Angebote schicken muss und das nächste Angebot von Annahme oder Ablehnung des aktuellen Angebots abhängt.

Falls ein nächster Fälligkeitstermin nicht existiert:

$$FR_{MS_i}^{T_i} = T_s + 10$$

Angebot der Montagestation MS_i ergibt sich aus $\left(gemMS_i^{TA_i}, FR_{MS_i}^{TA_i} \right)$.

Transportagent wählt diejenige Montagestation, die am kürzesten $gemMS_i$ Zeit angeboten hat. Um Angebot zu bestätigen muss sich Transportagent zurückmelden bevor Fälligkeitstermin für die Reservierung $FR_{MS_i}^{TA_i}$ erreicht ist. Wenn Transportagent bis $FR_{MS_i}^{TA_i}$ keine Reservierung bei Montagestation MS_i gemacht hat wird er automatisch aus der Liste der Montagestation MS_i gestrichen werden. Falls der Transportagent nächsten Montageschritt nicht gesichert hat muss er erneuert Anfragen versenden. Verhandlungsprozess wird mit einem Beispiel verdeutlicht. In dem Beispiel beträgt die Systemzeit 150. Montagestation MS_1 bekommt Anfragen von drei Transportagenten; TA_1 , TA_2 , TA_3 .

Angekommene Anfragen werden in Montagestation wie auf die Abbildung 8.4 bearbeitet. Als erstes wird Montagebezogene Informationen überprüft.

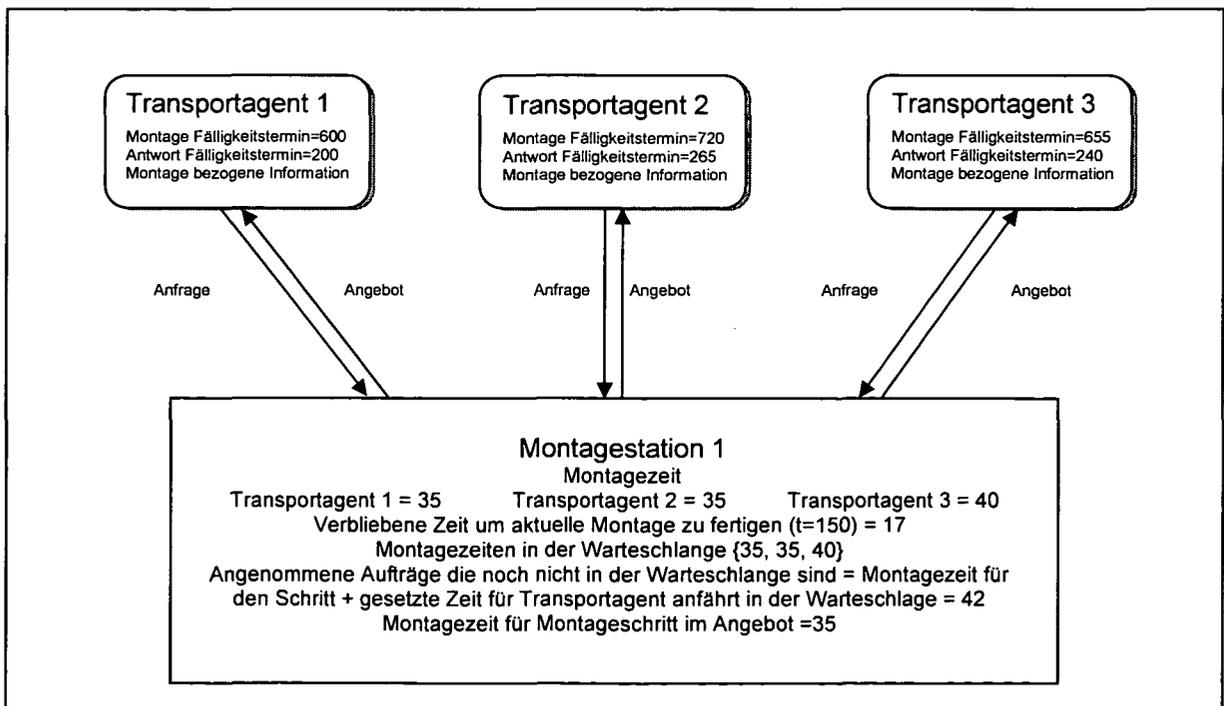


Abb. 8.3. Beispiel der Verhandlung zwischen drei Transportagenten und Montagestation.

Dadurch erfährt die Montagestation ob sie für den Montageschritt zuständig ist und ob sie zurzeit diesen Auftrag ausführen kann. Es kann passieren dass die Montagestation zur dieser keine passenden Teile lagernd hat.

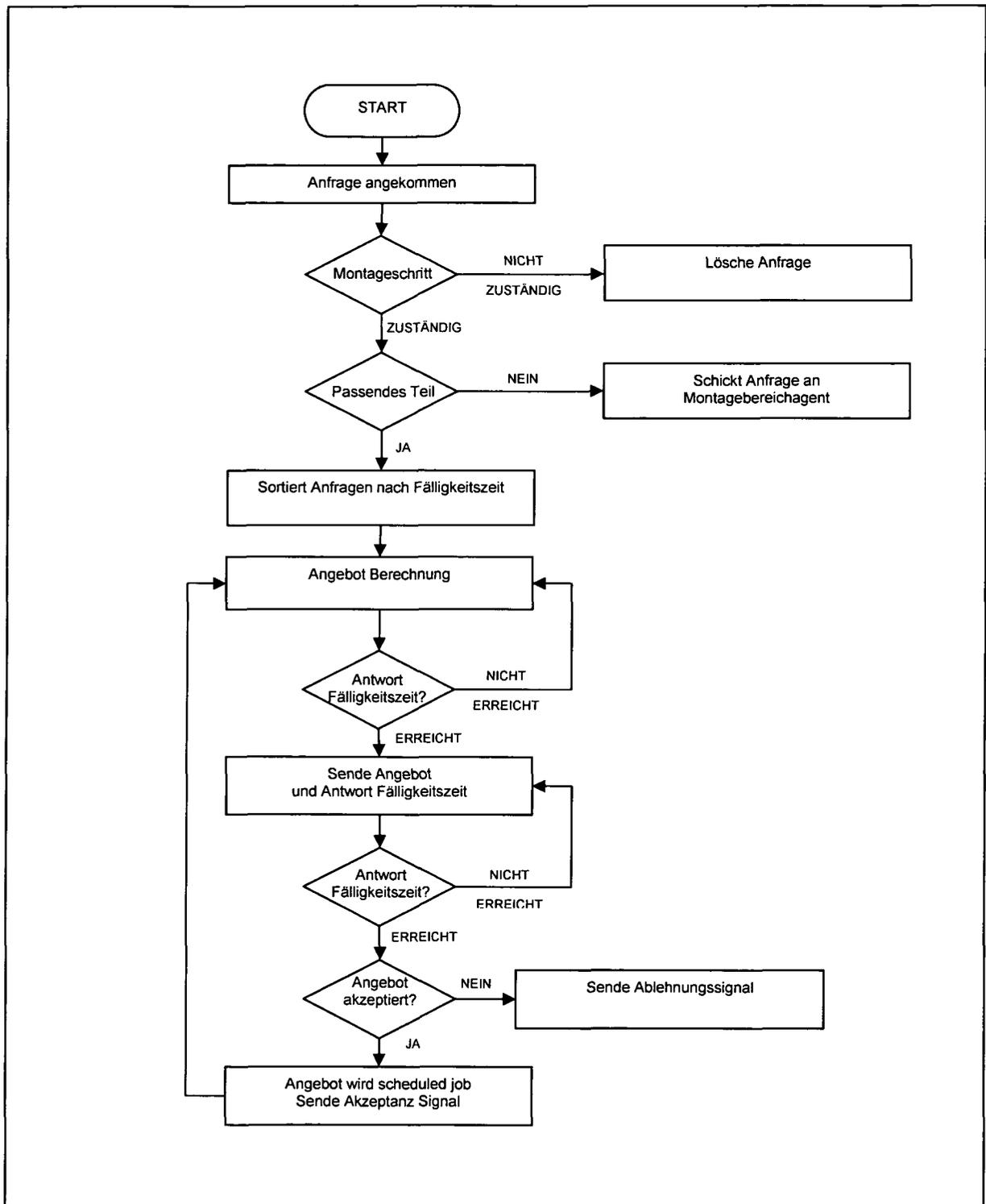


Abb. 8.4. Teil der Verhandlung aus dem Sicht der Montagestationen.

Ursachen dieses Zustandes sind:

- Fehler in der Montagestationsteuerung,
- passende Montageeile sind bestellt aber noch nicht angekommen,
- geplante Montage kann mit einer anderen Montagestation zeitig beendet werden und weitere Bestellungen der Montageeile dieses Typs sind unterdrückt.

Anfragen werden weiter nach Montage Fälligkeitstermin strukturiert und bearbeitet. Antwort Fälligkeitstermine sind 200, 265 und 240. Das heißt, wenn Systemzeit 200 erreicht ist, verschickt Montagestation MA₁ das Angebot zu dem Transportagent TA₁.

$$gemMS_1^{TA_1} = 200 + 17 + 35 + 35 + 40 + 42 + 35 = 404$$

In dem Fall ist es 404. Da die nächste Antwort Fälligkeitstermin 265 ist, ergibt sich Fälligkeitstermin für die Reservierung für Transportagent TA₁ wie folgt

$$FR_{MS_1}^{TA_1} = \frac{200 + 240}{2} = 220$$

$$\text{Angebot } A_{MS_1}^{TA_1} = \left(gemMS_1^{TA_1}, FR_{MS_1}^{TA_1} \right) = (404, 220)$$

In der Systemzeit 240 Montagestation MA₁ schickt Angebot zu Transportagent TA₃:

$$\text{Angebot } A_{MS_1}^{TA_3} = (404, 252.5)$$

Und in der Systemzeit 265 zu Transportagent TA₂

$$A_{MS_1}^{TA_2} = (479, 275)$$

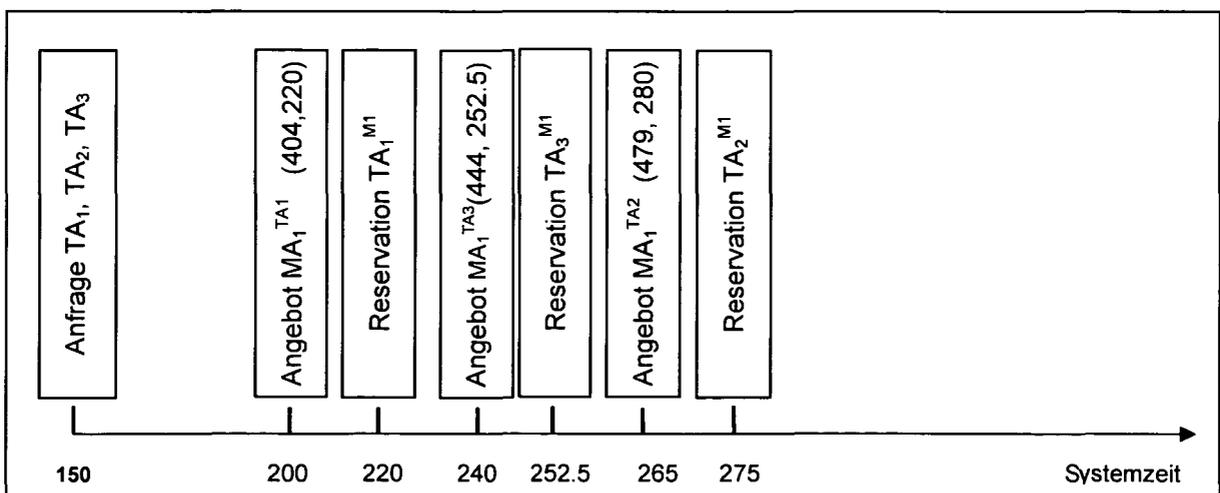


Abb. 8.5. Ablauf der Kommunikation in Systemzeit. Alle drei Transportagenten reservieren Montagestation 1 für nächsten Montageschritt.

Falls Transportagent TA_1 bis Systemzeit 220 keine Reservierung der Montagestation MA_1 gemacht hat korrigiert MA_1 weitere Angebote:

Angebot gesendet in der Systemzeit 240

$$\text{gem } MA_1^{TA_3} = 444 - 35 = 409$$

$$A_{MS_1}^{TA_3} = (444, 252.5)$$

Angebot gesendet in der Systemzeit 265

$$\text{gem } MA_1^{TA_2} = 409 + 35 = 444$$

$$A_{MS_1}^{TA_2} = (444, 275)$$

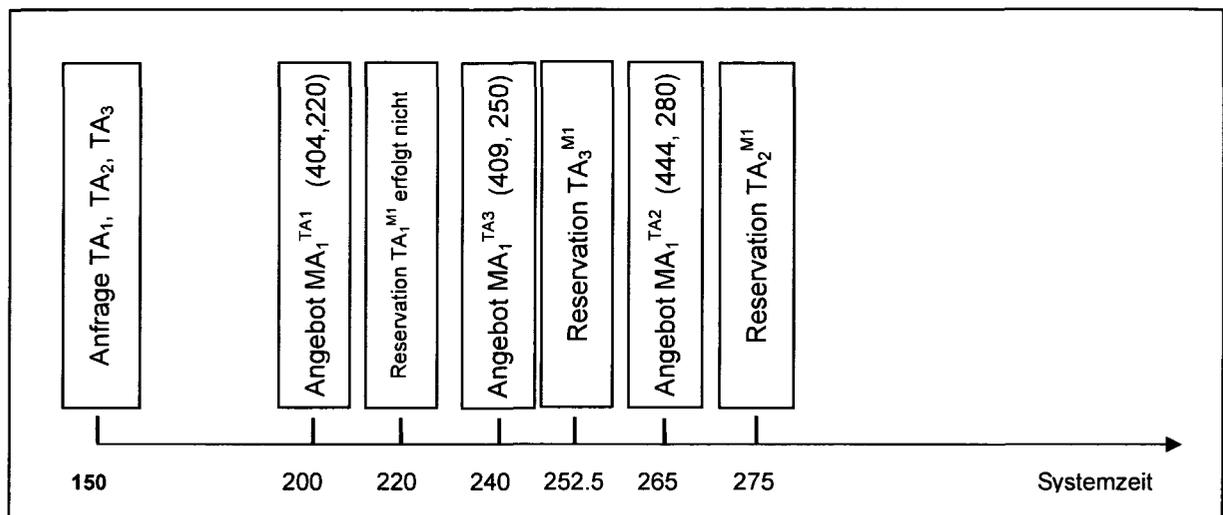


Abb. 8.6. Ablauf der Kommunikation in Systemzeit. TA_1 meldet sich bis Systemzeit 220 nicht. MA_1 korrigiert weitere Angebote.

Abbildung 8.7. zeigt Verhandlungsablauf bei Transportagenten.

8.3 Vorteile der verwendeten Verhandlungsmethode

Typische Länge eines Verhandlungsprozesses innerhalb des Multiagentensystems ist durch die Summe der Zeit für die Zusammensetzung der Anfrage, Sendung, Zusammensetzung des Angebots und dessen Empfang definiert. Also die Informations Bearbeitung einer Montagestation basiert auf der First-In-First-Out Methode. Dabei können weder weitere Angebote wegen abgelehnten Aufträgen korrigiert noch Prioritätsstufen eingeführt werden. Außerdem sind überlappende Anfragen beziehungsweise Angebote unvermeidbar. Einführung der Fälligkeitstermins von Angebot und Reservierung ermöglicht beides.

Diese Verhandlungsmethode ermöglicht Ressourcenservierung eines Montageschrittes im voraus. Durch den Fälligkeitstermin von Angebot und Reservierung ist Verhandlungszeit ausgedehnt. Dies kann auch genutzt werden um die gleiche Anfrage oder das gleiche Angebot mehrmals verschicken zu können, um Störungen innerhalb Kommunikationssystem zu

minimieren. Jedoch ist dies innerhalb dieser Studie und in Simulationsversuchen nicht betrachtet. Also es wird angenommen, dass die Information fehlerfrei durch das System geleitet wird.

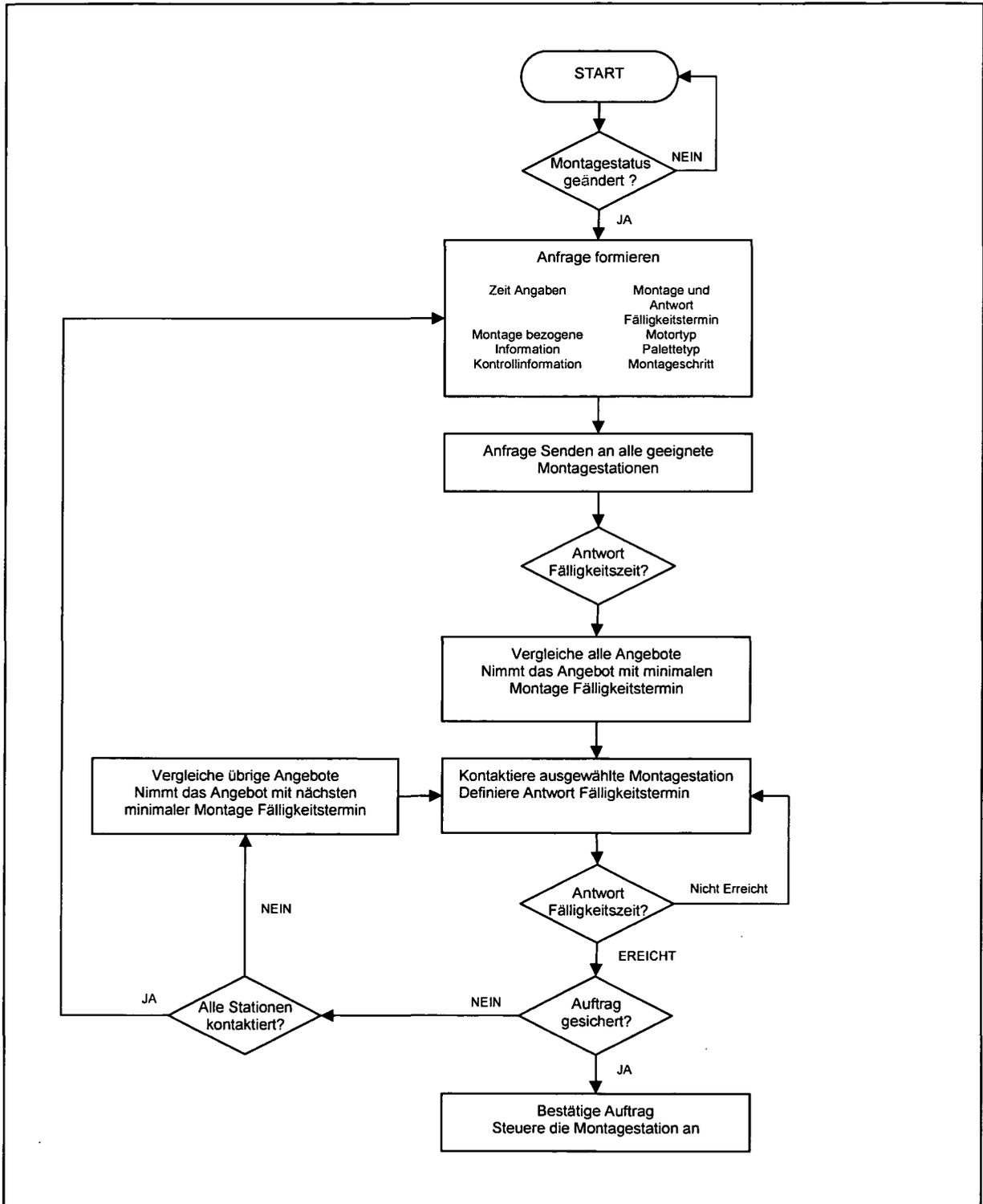


Abb. 8.7. Teil der Verhandlung aus dem Sicht der Transportagenten.

8.4 Simulationsversuche

Zweck der Simulation ist es Verhalten des Montagesystems unter Produktionsreservierungssystem zu untersuchen und mit traditionellen Kriterien für die Strukturierung von Warteschlangen zu vergleichen. Simulationsversuche sind mit vier Kriterien für die Strukturierung von Warteschlangen durchgeführt. Das sind:

- First In First Out (FIFO) – Transportagenten in den Puffern der Montagestationen werden nach Eintrittstermin im Puffer bearbeitet.
- Earliest Due Date (EDD) – Transportagenten werden nach dem Fälligkeitstermin der Montage bearbeitet. Der mit kürzestem Fälligkeitstermin wird als erster bearbeitet.
- Least Slack Time (LST) – Transportagenten werden nach der Differenz zwischen Fälligkeitstermin der Montage und Systemzeit bearbeitet (slack). Da dieser Unterschied dynamischer Art, ist verursacht diese Methode weniger tardy jobs als Earliest Due Date Kriterium.
- Shortest Processing Time (SPT) – Transportagenten werden nach der Montagedauer bearbeitet, angefangen mit kürzester Bearbeitungszeit.

Ein weiterer Satz von Simulationen wurde ausgeführt, um den Einfluss von Planungshorizont auf die Produktionsparameter zu untersuchen. Dabei wurde Planungshorizont auf 2, 3, 4, 5, 6 und 9 Montageschritte ausgedehnt.

Der verwendete Simulator ist ARENA 7.0 von Rockwell Automation. ARENA Simulationssoftware ist speziell zum Modellieren komplexer Kontrollelogiken entwickelt worden.

8.4.1 Parameter für die Evaluierung des Montagesystems

Oberstes Prinzip, dem zukünftige Produktionssysteme verpflichtet sein müssen ist Just in Time Produktlieferung in kleinen bis mittleren Lößgroßen. Also erstes Kriterium für die Evaluierung des Montagesystems ist Prozentsatz von Aufträgen, die nach (Tardiness) und die vor (Eliness) der Lieferungszeit die Entladestation erreicht haben.

Zweiter Parameter ist Work in Progress Parameter. Work in Progress Parameter gibt maximale Anzahl der Transportagenten, die sich in den Warteschlangen aktiver Montagestationen befinden. Entstandene Montagekonfiguration (Anzahl und Stellung von allen aktiven Montagestationen) wird sich aufgrund höherer Anzahl von Transportagenten im Montagesystem früher rekonfigurieren als notwendig wäre. Jedes Mal wenn mehr als 8 Transportagenten in der Warteschlange sich befinden, schaltet sich alternative

Montagestation ein. Damit fängt die Rekonfiguration der Montagestruktur an. Um Rekonfiguration des Montagesystems zu untersuchen gibt es zurzeit Simulationssoftware die für die Simulation von Verhalten sozialer Insekten und für Ethologische Untersuchungen gemacht ist. Resultate solcher Simulationen sind wegen hoher Abstraktion für den Einsatz im Produktionsbereich nicht anwendbar. Durch der, Work in Progress Parameter kann man zwar nicht die Rekonfiguration eines Montagesystems untersuchen, der Parameter muss jedoch niedrig sein um unnötige Rekonfigurationen zu vermeiden.

Montagestationbelegungszeit gibt die tatsächliche Zeit, die Montagestationen benötigen um einen Montageschritt auszuführen.

Für alle eingeführten Parameter gilt der Grundsatz: je weniger desto besser.

8.4.2 Das Simulationsmodell

Das Montagesystem besteht aus Transportagenten Pool mit 30 Transportagenten, 9 Montagestationen, Reparatur- und Entladestation. Jede Montagestation ist für einen Montageschritt zuständig wie in Tabelle xy gezeigt ist.

Da sich die Montage von Lagerschild, Ausgleichscheibe und Kugellager zwei Mal wiederholt, gibt es für diese Montageschritte alternative Montagestationen; Lagerschild (M1, M8), Ausgleichscheibe (M2, M7) und Kugellager (M3, M6).

Solche Kombination von Montagestationen ist nicht wirtschaftlich begründet aber auf der Basis dieser Konfiguration kann man die in der Kapiteln 7 und 8 aufgestellte Agenten- und Systemlogik beziehungsweise Verhandlungsmethode untersuchen. Aus den alternativen Montagestationen ergeben sich neue Kombinationen, die von Transportagenten genutzt werden können. Diese Kombinationen sind in der Tabelle xy erfasst.

MONTAGESCHRITT	MONTAGESTATION	MONTIERT WIRD
1	M ₁	Lagerschild
2	M ₂	Ausgleichscheibe
3	M ₃	Kugellager
4	M ₄	Stator
5	M ₅	Rotor
6	M ₆	Kugellager
7	M ₇	Ausgleichscheibe
8	M ₈	Lagerschild
9	M ₉	Befestigungsbolzen
10	Entladestation	Entladen
11	Reparatur	Reparatur

Tab. 8.3. Aufteilung der Montageaufgaben.

Abbildung 8.8 zeigt das Model des Montagesystems in Simulationssoftware ARENA. Das Model besteht aus folgenden Submodellen: Transportagenten Pool, 9 Montagestationen, Entladestation sowie Reparaturstation.

MONTAGESCHRITT	VON	ZU
1 Lagerschild	Montageagenten Pool	M ₁ M ₈
2 Ausgleichscheibe	M ₁ M ₈	M ₂ M ₇
3 Kugellager	M ₂ M ₇	M ₃ M ₆
4 Stator	M ₃ M ₆	M ₄
5 Rotor	M ₄	M ₅
6 Kugellager	M ₅	M ₃ M ₆
7 Ausgleichscheibe	M ₃ M ₆	M ₂ M ₇
8 Lagerschild	M ₂ M ₇	M ₁ M ₈
9 Befestigungsbolzen	M ₁ M ₈	M ₉
10 Entladestation	M ₉	M ₁₀
11 Reparatur	M ₂ – M ₉	Reparaturstation

Tab. 8.4. Transportwege.

	Name	Type	Action	Priority	Resources	Delay Type	Units	Allocation
1	1st Lageschild	Standard	Seize Delay Release	Medium(2)	1 row	Triangular	Seconds	Value Added
2	1st Ausgleichscheibe	Standard	Seize Delay Release	Medium(2)	1 rows	Triangular	Seconds	Value Added
3	1st Kugellager	Standard	Seize Delay Release	Medium(2)	1 rows	Triangular	Seconds	Value Added
4	2nd Kugellager	Standard	Seize Delay Release	Medium(2)	1 rows	Triangular	Seconds	Value Added
5	2nd Ausgleichscheibe	Standard	Seize Delay Release	Medium(2)	1 rows	Triangular	Seconds	Value Added
6	2nd Lageschild	Standard	Seize Delay Release	Medium(2)	1 rows	Triangular	Seconds	Value Added

Abb. 8.8. Bionic Assembly System Model in ARENA 7.0.

8.4.3 Simulationsannahmen

Jede Simulation ist 20 000 Zeiteinheiten lang und wird 5 Mal wiederholt. Jeder Versuch startet mit zufälligem Mix von Motortypen in Lößgroßen 10-70. Insgesamt gibt es 10 Motortypen die in 9 Montageschritten montiert werden.

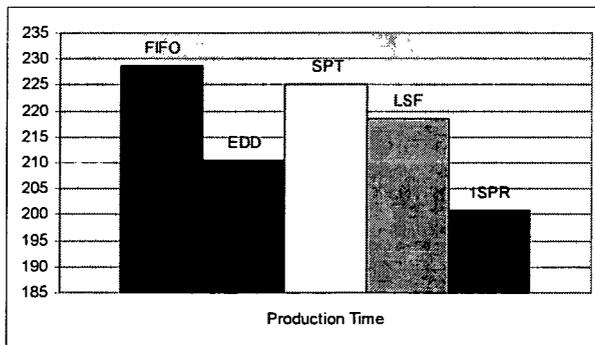
Montagezeit und Transportzeit zwischen Montagestationen wurde mittel Triangular Funktion simuliert. Ein Beispiel: Für die Lagerschildmontage benötigt Montagestation mindestens 5, durchschnittlich 6 und maximal 8 Zeiteinheiten.

Systemzeit, in der Transportagent der Auftrag übernimmt ist mindestens vier Mal kleiner als durchschnittliche Montagezeit für den entsprechenden Motortyp.

8.4.4 Ergebnisanalyse

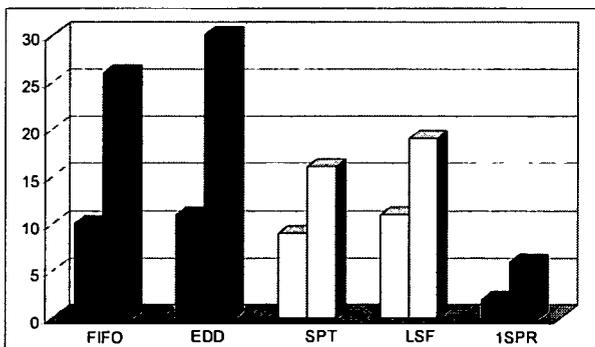
Abbildung 8.9 bis 8.12 zeigen die Resultate des Vergleichs vier lokaler heuristischer Methoden und Produktionsreservierung mit Planungshorizont einen Montageschritt voraus unter konstanter Produktionskapazität (300 MTBA) sowie Produktkomplexität (9 Montageschritte).

Flow Time, Production Time, maximaler Work in Progress und Tardiness sind deutlich besser bei Produktionsreservierung als bei traditionellen heuristischen Kriterien für die Strukturierung von Warteschlangen.



Durchschnittliche Produktionszeit eines Produktes = \sum Transportzeiten + \sum Wartezeiten in Warteschlangen + \sum Montage

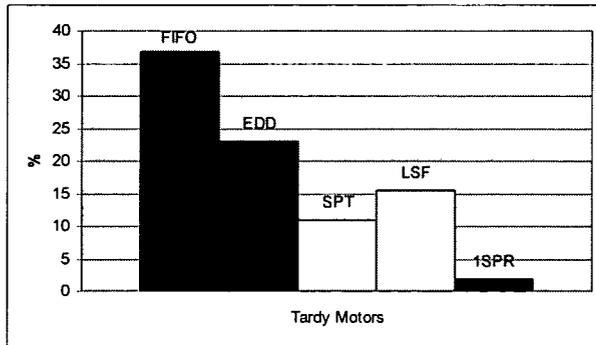
Abb. 8.9. Production Time.



Minimale und Maximale Anzahl der Transportagenten in den Warteschlangen.

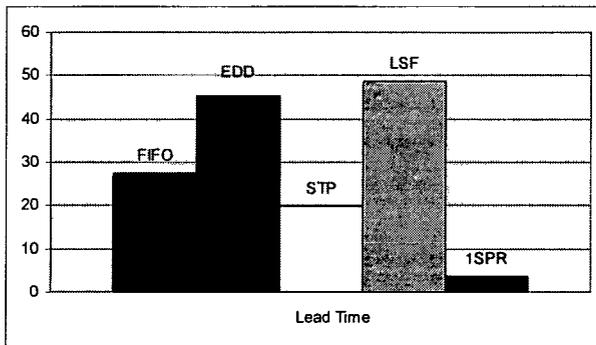
Abb. 8.10. WIP per assembly station.

Deutlich ist die Verbesserung im Vergleich zu FIFO Methode ausgefallen. Das Montagesystem ohne Verhandlungsphase verhält sich nach dem FIFO Prinzip. Resultate zeigen, dass die Steigerung der Steuerungskomplexität des Agenten im jeden Fall vertretbar ist.



Nach dem Lieferungstermin gelieferte Produkte in Prozenten angegeben.

Abb. 8.11. Tardy motors.



Wie Lange dauert um die Montage eines neuen Produkts zu starten. Da das Montagesystem kein eingangs Puffer hat, reduziert sich die Zeit auf Transportzeit zwischen Agenten Pool und erste Montagestation. Deswegen ist der direkte Vergleich zur Lead Time nicht möglich.

Abb. 8.12. Lead Time.

Tabelle 8.5 zeigt Resultate der Untersuchung. Verbesserung der Produktionsreservierung in Vergleich zu anderen Kriterien ist in Prozenten angegeben.

In zweiten Simulationsversuch ist untersucht, wie der Planungshorizont Produktionsparameter beeinflusst. Deswegen ist der Planungshorizont auf 2 (2SPR), 3 (3SPR), 4 (4SPR), 5 (5SPR) und 6 (6SPR) Montageschritte ausgedehnt. Verbesserung eines Parameters hat dazu geführt alle anderen deutlich zu verschlechtern.

	PRODUCTION TIME	% TARDY	TOTAL WIP
FIFO vs PR	18,40%	34,24%	22,51%
EDD vs PR	3,48%	20,20%	26,82%
SPT vs PR	12,65%	8,14%	12,14%
LSF vs PR	10,92%	14,21%	20,86%

Tab. 8.5. Prozentsatz der Verbesserung im Vergleich zu vier traditionellen Kriterien.

Diagramme 8.13 bis 8.15 fassen Resultate der Simulationen zusammen.

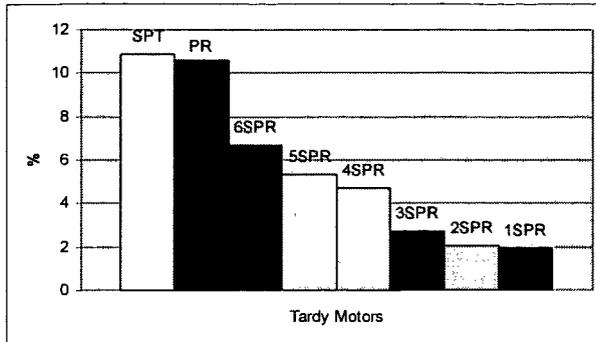


Abb. 8.13. Tardy motors.

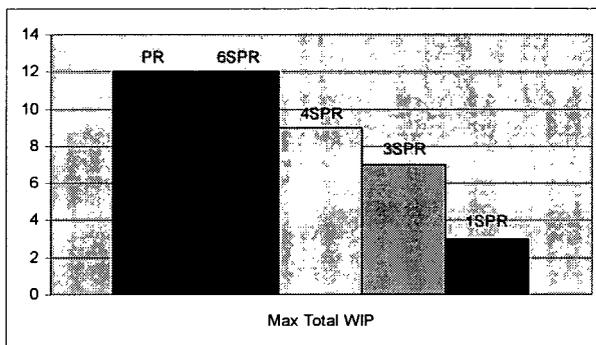


Abb. 8.14. Work in Progress.

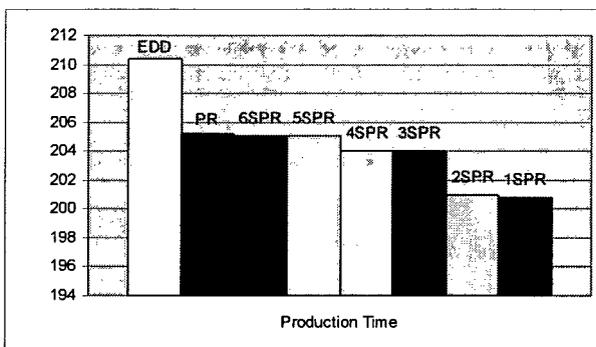


Abb. 8.15. Production time.

Im einen anderen Satz von Simulationen ist untersucht wie sich das Montagesystem bei Kapazitätsschwankungen und Schwankungen in der Komplexität des Produkts verhält.

Produktionskapazität ist durch Mean time between Arivals (MTBA) simuliert. Schwankung der Produktkomplexität ist durch Anzahl von Montageschritten simuliert. In der Simulation sind drei Komplexitätsebenen untrsucht: Produkte mit 3 bis 7, 4 bis 9 und 5 bis 11 Montageschritten. Abbildung 8.14 zeigt Leistungsvergleich für den Fall der 5-11

Montageschritte und MTBA zwischen 100 und 350, Abbildung 8.15 Leistungsunterschied als Funktion der Kapazitäts- und Produktkomplexitätsschwankung.

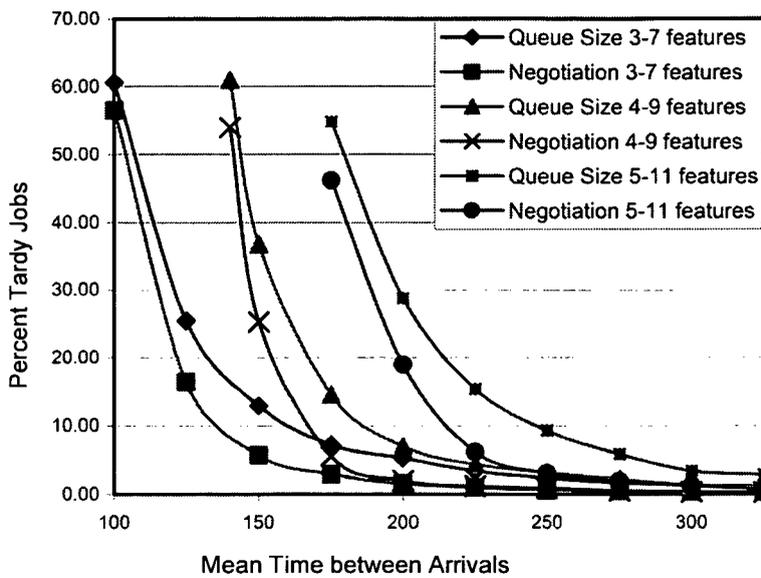


Abb. 8.16. Leistungsvergleich für den Fall der 5-11 Montageschritte und MTBA zwischen 100 und 350.

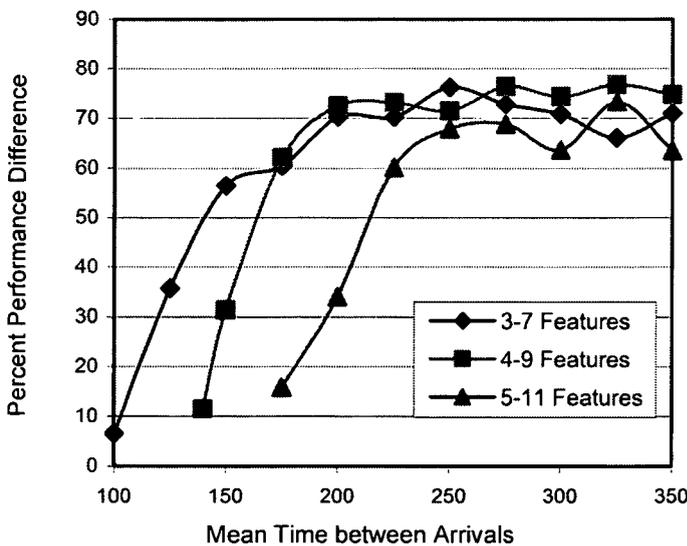


Abb. 8.17. Leistungsunterschied als Funktion der Kapazitäts- und Produktkomplexitätsschwankung.

Bei der Montage von einem größeren Mix komplexerer Produkte ist Überlegenheit der Produktionsreservierung geringer. Der Grund liegt in der schwachen Kopplung zwischen Transport- und Versorgungsagenten. Reservierung der Montagestationen und Bestellung der Teile von Montagestationen sind zwar zwei verkoppelte Prozesse, jedoch kommt es vor dass Lieferungen der Teile in der Montagestation nicht mit der Ankunft der Transportagenten

abgestimmt sind. Bei der Verteilung von Aufträgen von Scheduler bekommt jede Montagestation Zahl und Mix von Montageteilen. Weil das Lager der Montagestation begrenzt ist und die Aufteilung der Montageteile auf Statistik berührt, geschieht es, dass manche Montagestationen öfter bestimmte Montageteile bestellen müssen. Diese Bestellungen erfolgen sobald ein Montageteiltyp gewisse Zahl erreicht. Genau beim größeren Mix von Produkten passiert es, dass Bestellungen dynamischen Gegebenheiten nicht folgen können und deshalb bestellte Montageteile nicht mehr gebraucht werden.

Rekonfiguration des Systems

Der größte Vorteil des selbst-organisierenden Systems ist die Möglichkeit sich selbst im Laufe der Zeit zu rekonfigurieren und sich damit auf neue Umstände anzupassen. Ziel der autonomen dynamischen Rekonfiguration ist es, ausgehend von einer gegebenen Konfiguration diese zur Laufzeit zu überprüfen und gegebenenfalls dynamisch zu verändern. Durch dynamische Rekonfiguration wird die Zuteilung von Aufgaben und Fähigkeiten, in einer sich verändernden Umgebung so angepasst, dass eine möglichst optimale Endkonfiguration erreicht wird. Die Entscheidung zur Rekonfiguration wird dabei dezentral von jedem einzelnen Agenten getroffen und durchgeführt.

Es gibt drei Zustände des Systems, die Rekonfiguration auflösen:

1. Produkte – (a) Es gibt Produkte die sofort montiert werden mussten. Damit wird die Rekonfiguration der Warteschlangen initiiert. (b) Es gibt Produkte die zusätzliche Montagestationen benötigen. Rekonfiguration des Montagestruktur wird initiiert.
2. Montagestation Ausfall
3. Schwankungen in der Lieferungszeit – Falls die Produkte mit steigender Tendenz früher in die Entladestation ankommen, schaltet Montagestationsbereichagent gewisse alternative Montagestationen aus. Falls gelieferte Produkte mit steigender Verspätungstendenz ankommen, schaltet Montagestationsbereichagent gewisse alternative Montagestationen ein. Welche Montagestationen ein oder ausgeschaltet werden hängt von aktueller Montagestruktur ab; (1) Montagestationsbereichagent kann nur diejenigen Montagestationen ausschalten für welche es Alternative gibt. (2) Zwischen Alternativen Montagestationen wird diejenige ausgeschaltet die kleineren aktuellen Auslastungsgrad hat. Aktueller Auslastungsgrad jeder Montagestation ist durch Wartezeit in den Warteschlangen bekannt. Da Montagezeiten für alle Produkte kleine Abweichungen haben, ist es damit gesichert, dass dadurch ausgelöste Rekonfiguration weniger Transportagenten treffen will. Also Rekonfiguration wird in weniger Schritten ausgeführt.

Der normale Zustand des Montagesystem ist in Balance. Theoretisch heißt es, die Montage jeder Prioritätsgruppe an alternativen Montagestationen wird gleichzeitig beendet. In der Praxis heißt es, dass Endmontagezeiten für jede Prioritätsgruppe sich aber innerhalb Sensibilität S bewegen. Auf der Abbildung 9.1 ist ein ausbalancierter Zustand gezeigt.

$$\begin{aligned}
 T^{P_1} &= \frac{T_{MS_i}^{P_1} + T_{MS_{i+1}}^{P_1} + \dots + T_{MS_n}^{P_1}}{n} & T^{P_1} - T_{MS_i}^{P_1} &\leq S; & T^{P_1} - T_{MS_{i+1}}^{P_1} &\leq S; & T^{P_1} - T_{MS_n}^{P_1} &\leq S \\
 T^{P_2} &= \frac{T_{MS_i}^{P_2} + T_{MS_{i+1}}^{P_2} + \dots + T_{MS_n}^{P_2}}{n} & T^{P_2} - T_{MS_i}^{P_2} &\leq S; & T^{P_2} - T_{MS_{i+1}}^{P_2} &\leq S; & T^{P_2} - T_{MS_n}^{P_2} &\leq S \\
 T^{P_3} &= \frac{T_{MS_i}^{P_3} + T_{MS_{i+1}}^{P_3} + \dots + T_{MS_n}^{P_3}}{n} & T^{P_3} - T_{MS_i}^{P_3} &\leq S; & T^{P_3} - T_{MS_{i+1}}^{P_3} &\leq S; & T^{P_3} - T_{MS_n}^{P_3} &\leq S
 \end{aligned} \quad (9.1)$$

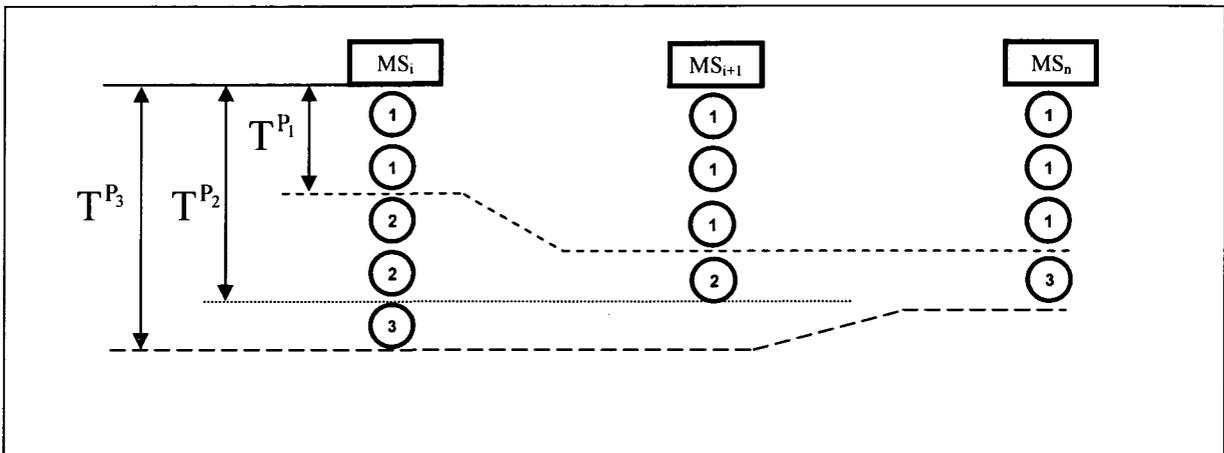


Abb. 9.1. Ausbalancierter Zustand.

Sobald Differenz zwischen Endmontagezeiten für eine Prioritätsgruppe die Sensibilität überschreitet, wird die Rekonfiguration ausgelöst.

REGEL FÜR REKONFIGURATION DER WARTESCHLANGEN:

1. Die Rekonfiguration fängt bei aktiver alternativer Montagestation, die am größten Montagezeitvorrat in der höchsten Priorität hat.
2. In der Station fängt die Rekonfiguration beim letzten Tarnsportagenten der oberste Prioritätsgruppe an und geht bis zur ersten in der Gruppe.
3. Prozess wiederholt sich für alle alternativen Montagestationen in derselben Prioritätsgruppe.
4. Wenn Rekonfiguration erste Montagestation wieder erreicht hat, wiederholt sich ganzer Prozess für die Gruppe mit einem Prioritätsgrad weniger.
5. Schritte 3 und 4 wiederholen sich bis alle Montagezeitvorräte ausgeglichen sind. Dann endet Rekonfiguration automatisch.

Abbildung 9.2 zeigt Algorithmus der Rekonfiguration im Form des Flussdiagramms. Rekonfiguration ist verdeutlicht auf der Abbildung 9.3 Transportagent mit dem Produkt höherer Priorität ist in Anfahrt zu Montagestation MS_i .

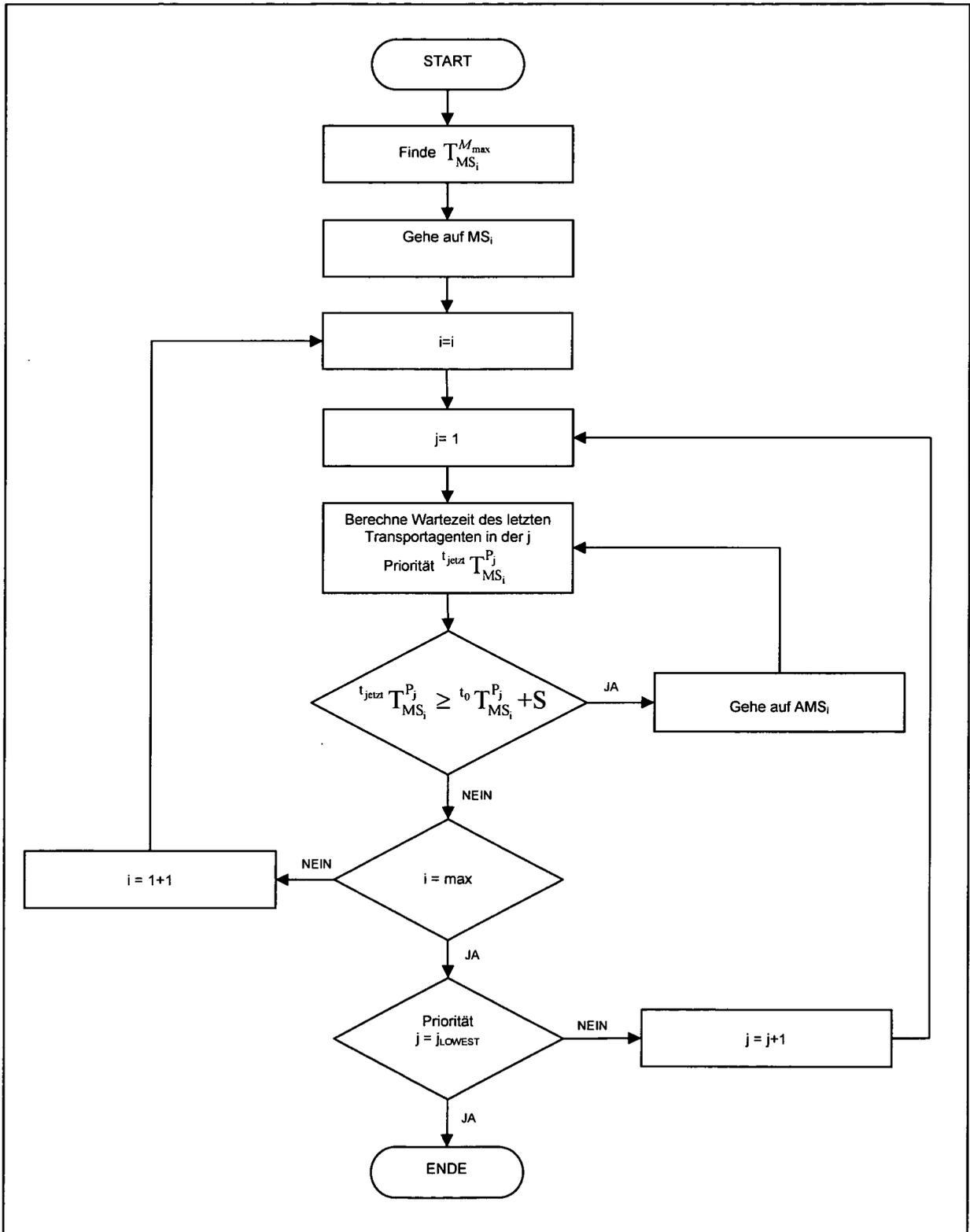


Abb. 9.2. Ablauf der Rekonfiguration.

Im Kernsystem gibt es zwei aktive alternative Montagestationen, MS_i und MS_{i+1} .
Abbildung 9.3 zeigt die Situation in der Zeit t_0 .

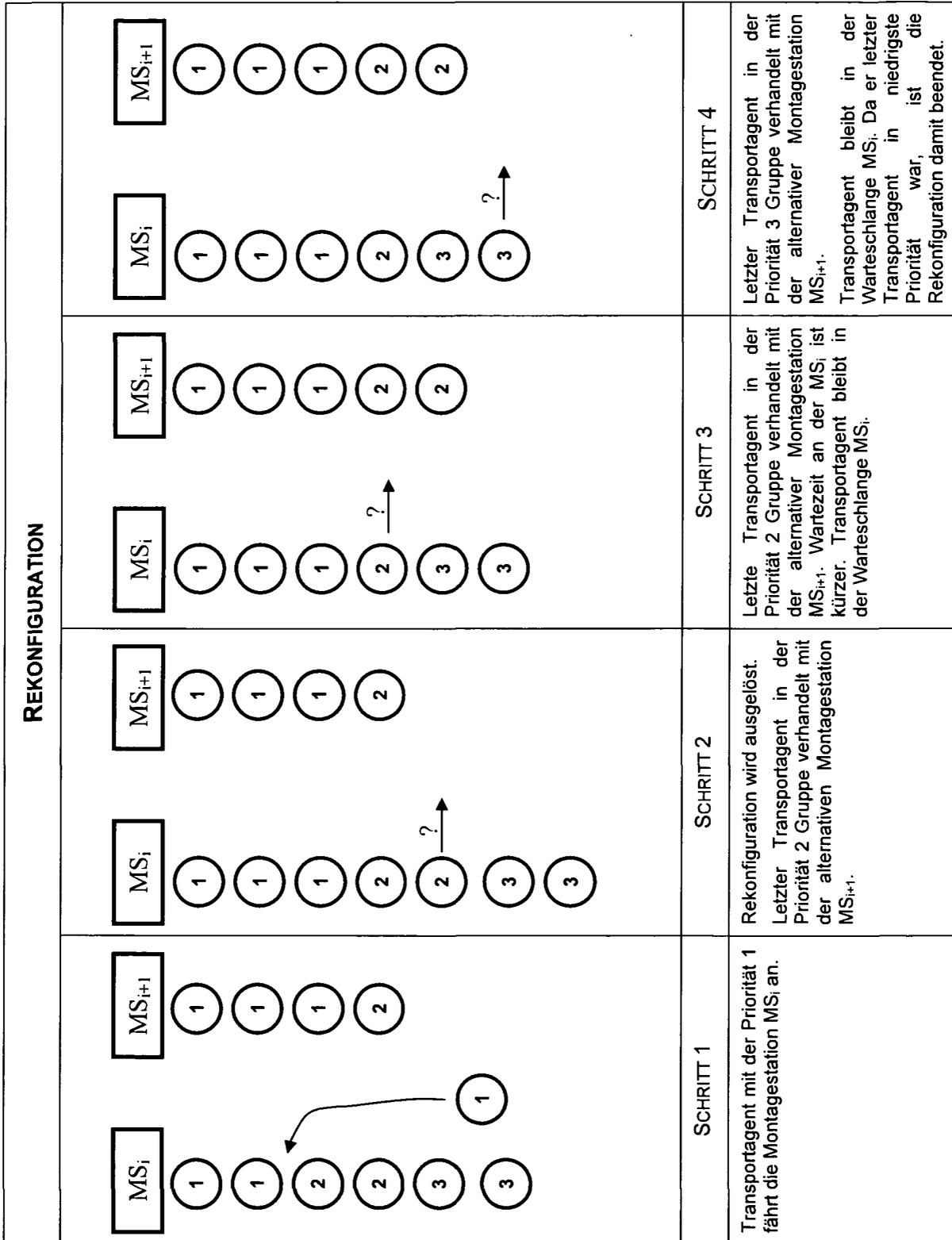


Abb. 9.3. Rekonfiguration.

Abbildung 9.4 zeigt Rekonfigurationsprozess auf dem Beispiel des Ausfalls einer Montagestation.

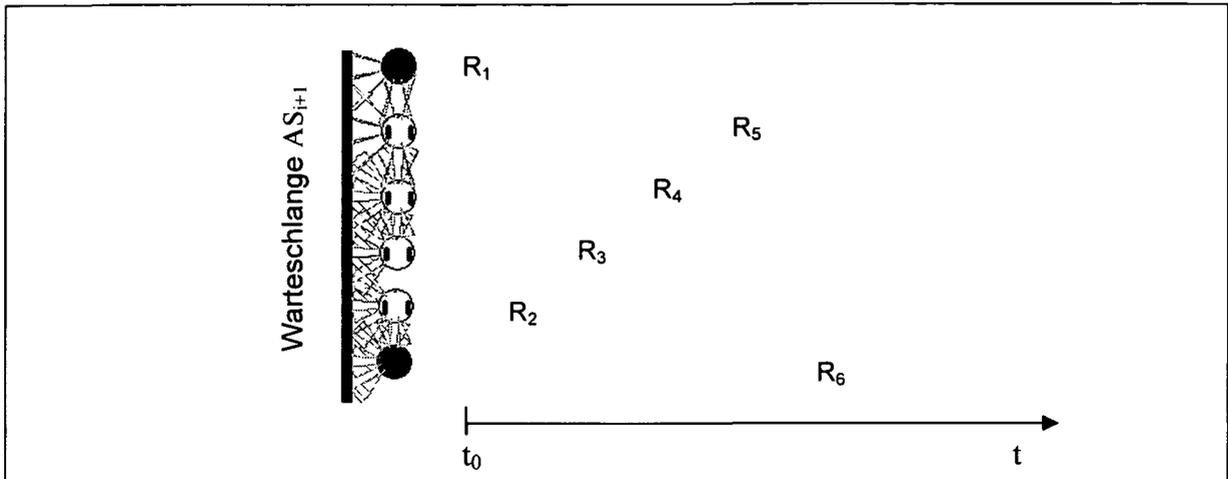


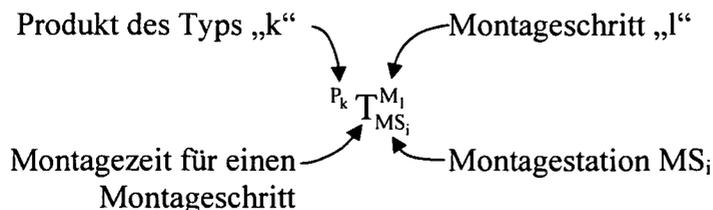
Abb. 9.4. Auflösen der Warteschlange AS_{i+1} nach Reihenfolge R_n .

9.1 Mathematisches Modell der Rekonfiguration

9.1.1 Technologische Montagezeiten

Matrix den technologischen Montagezeiten zeigt wie lange dauert jeder Montageschritt auf jeder Montagestation. Diejenigen Montagestationen, die gewisse Montageschritte nicht ausführen können, schreiben unendliche Montagezeiten. In der Praxis geht es um begrenzte Zeit, zum Beispiel von zwei Tagen. Da jeder der Transportagenten diejenige Montagestation sucht, die am schnellsten montieren kann, kommen die mit so großen Wartezeiten nie in Betracht. Damit wird der Matrix ermöglicht Montageszenarien zu generieren. Dies ist für Montagebereichagent wichtig, da für jedes Produkt, das gerade montiert wird, mindestens eine Montagestation pro Montageschritt und Produkt aktiv sein muss. Dadurch kann der Montagebereichagent die Anzahl der Montagestationen optimieren.

Technologische Montagezeit eines Montageschrittes ist wie folgt bezeichnet



Produkt	Montageschritt	MS ₁	MS ₂	...	MS _i	MS _{i+1}	...	MS _n
P₁	M ₁₁	P ₁ T _{MS₁} ^{M₁}	P ₁ T _{MS₂} ^{M₁}	...	∞	P ₁ T _{MS_{i+1}} ^{M₁}	...	∞
	M ₁₂	∞	P ₁ T _{MS₂} ^{M₂}	...	P ₁ T _{MS_i} ^{M₂}	P ₁ T _{MS_{i+1}} ^{M₂}	...	∞

	M _{1l}	P ₁ T _{MS₁} ^{M₁}	∞	...	P ₁ T _{MS_i} ^{M₁}	∞	...	P ₁ T _{MS_n} ^{M₁}
	M _{1l+1}	∞	∞	...	P ₁ T _{MS_i} ^{M_{l+1}}	P ₁ T _{MS_{i+1}} ^{M_{l+1}}	...	P ₁ T _{MS_n} ^{M_{l+1}}
	M _{1m}	P ₁ T _{MS₁} ^{M_m}	P ₁ T _{MS₂} ^{M_m}	...	P ₁ T _{MS_i} ^{M_m}	∞	...	P ₁ T _{MS_n} ^{M_m}
...
P_k	M _{k1}	P _k T _{MS₁} ^{M₁}	P _k T _{MS₂} ^{M₁}	...	P _k T _{MS_i} ^{M₁}	P _k T _{MS_{i+1}} ^{M₁}	...	∞
	M _{k2}	P _k T _{MS₁} ^{M₂}	∞	...	P _k T _{MS_i} ^{M₂}	P _k T _{MS_{i+1}} ^{M₂}	...	P _k T _{MS_n} ^{M₂}

	M _{k_l}	P _k T _{MS₁} ^{M₁}	∞	...	P _k T _{MS_i} ^{M₁}	∞	...	P _k T _{MS_n} ^{M₁}
	M _{k_l+1}	∞	P _k T _{MS₂} ^{M_{l+1}}	...	P _k T _{MS_i} ^{M_{l+1}}	P _k T _{MS_{i+1}} ^{M_{l+1}}	...	P _k T _{MS_n} ^{M_{l+1}}
	M _{km}	∞	P _k T _{MS₂} ^{M_m}	...	P _k T _{MS_i} ^{M_m}	∞	...	P _k T _{MS_n} ^{M_m}
...
P_p	M _{p1}	P _p T _{MS₁} ^{M₁}	∞	...	P _p T _{MS_i} ^{M₁}	∞	...	P _p T _{MS_n} ^{M₁}
	M _{p2}	P _p T _{MS₁} ^{M₂}	∞	...	P _p T _{MS_i} ^{M₂}	P _p T _{MS_{i+1}} ^{M₂}	...	P _p T _{MS_n} ^{M₂}

	M _{p_l}	P _p T _{MS₁} ^{M₁}	P _p T _{MS₂} ^{M₁}	...	P _p T _{MS_i} ^{M₁}	∞	...	∞
	M _{p_l+1}	∞	P _p T _{MS₂} ^{M_{l+1}}	...	P _p T _{MS_i} ^{M_{l+1}}	P _p T _{MS_{i+1}} ^{M_{l+1}}	...	∞
	M _{p_m}	P _p T _{MS₁} ^{M_m}	P _p T _{MS₂} ^{M_m}	...	∞	P _p T _{MS_{i+1}} ^{M_m}	...	P _p T _{MS_n} ^{M_m}

Tab. 9.1. Technologische Montagezeiten.

Anzahl der verschiedenen Produkte im Kernsystem:

$$P = \sum_{k=1}^p k \tag{9.2}$$

Anzahl der Montageschritte pro Produkt

$$M_{Pl} = \sum_{l=1}^m l \tag{9.3}$$

Technologische Montagezeit des Produktes k :

$$P_k T^M = \sum_{l=1}^m \sum_{i=1}^{last} l \cdot P_k T_{MS_i}^{M_l} \tag{9.4}$$

9.1.2 Montagestation Auftrag (AMSi)

Reservierte Ressourcen einer Montagestation bilden den Montagestation Auftrag. Tabelle 9.2 zeigt Montageauftrag für Montagestation MS_i .

Priorität	Montageschritt	Anzahl	Montagezeit	Σ Montagezeit
j = 1	M_{12}	2	$P_1 T_{MS_i}^{M_2}$	$2 \cdot P_1 T_{MS_i}^{M_2}$
	M_{1l+1}	1	$P_1 T_{MS_i}^{M_{l+1}}$	$P_1 T_{MS_i}^{M_{l+1}}$
	M_{kl}	1	$P_k T_{MS_i}^{M_l}$	$P_k T_{MS_i}^{M_l}$
	M_{km}	1	$P_k T_{MS_i}^{M_m}$	$l \cdot P_k T_{MS_i}^{M_m}$
	M_{p2}	$l+1$	$P_p T_{MS_i}^{M_2}$	$(l+1) \cdot P_k T_{MS_i}^{M_m}$
	M_{1l}	m	$P_1 T_{MS_i}^{M_l}$	$m \cdot P_1 T_{MS_i}^{M_l}$
...
j = j+1	M_{p2}	2	$P_p T_{MS_i}^{M_2}$	$2 \cdot P_k T_{MS_i}^{M_m}$
	M_{1l+1}	1	$P_1 T_{MS_i}^{M_{l+1}}$	$l \cdot P_1 T_{MS_i}^{M_{l+1}}$

	M_{pl}	m	$P_p T_{MS_i}^{M_l}$	$m \cdot P_p T_{MS_i}^{M_l}$
...
j = min	M_{p1}	1	$P_p T_{MS_i}^{M_1}$	$l \cdot P_p T_{MS_i}^{M_1}$
	M_{12}	$l+1$	$P_1 T_{MS_i}^{M_2}$	$(l+1) \cdot P_1 T_{MS_i}^{M_2}$

	M_{1m}	m	$P_1 T_{MS_i}^{M_m}$	$m \cdot P_1 T_{MS_i}^{M_m}$

Tab. 9.2. Auftrag für Montagestation MS_i .

Montageaufträge sind nach dem Prioritätsgrad strukturiert. Es gibt 5 Prioritäten:

1	Urgent
2	--
3	Normal
4	--
5	Nicht Eilig

Tab. 9.3. Prioritäten im BAS.

Gesamte Montagezeit der j-en Priorität an der Montagestation MS_i :

$$T_{MS_i}^{P_j} = \sum_{b=a_{j1}}^{a_{jLAST}} a_{j2} \cdot a_{j3} \quad (9.5)$$

Gesamter Montagezeitvorrat an der Montagestation MS_i :

$$T_{MS_i}^M = \sum_{j=1}^{\min} T_{MS_i}^{P_j} \quad (9.6)$$

Gesamte Montagezeit im Kernsystem:

$$T_{BAS}^M = \sum_{i=erste}^{letzte} \left(\sum_{j=höchste}^{niedrigste} T_{MS_i}^{P_j} \right) [s] \quad (9.7)$$

Wartezeit an der Montagestation MS_i :

$$T_{WP_{jk}}^{MS_i} = \sum_{l_i=höchste}^{j_i+1} k_{l_i} T_{l_i} + \sum_{k_{ji}=1}^{n_{ji}} k_{j_i} T_{j_i} [s] \quad (9.8)$$

Arithmetische Summe der Wartezeiten pro Priorität:

$$T_A^{P_j} = \frac{T_{WP_{jk}}^{MS_i} + T_{WP_{jk}}^{MS_{i+1}} + T_{WP_{jk}}^{MS_n}}{n} [s] \quad (9.9)$$

Sensibilität ist arithmetische Summe der Transportzeiten zwischen alternativen Montagestationen:

$$S = \frac{T_{T_1} + T_{T_2} + \dots + T_{T_n}}{n} [s] \quad (9.10)$$

Rekonfiguration wird ausgelöst, sobald sich Wartezeit in einer Priorität um mehr als Sensibilität von arithmetischer Summe der Wartezeiten für diejenige Priorität vergrößert.

$$\Delta T_{MS_i}^{P_j} \geq T_A^{P_j} + S \quad (9.11)$$

Schlussfolgerung und Ausblick

Sorgfältige Analyse der Umgebung und der Aufgabe hat entscheidenden Einfluss auf Design eines Agenten. Die größte Herausforderung dabei liegt bei sehr enger Interaktion zwischen dem Agenten und der Umgebung und setzt somit tiefes Verständnis der Beziehung zwischen Agent und Umgebung voraus. Nur wenn Agentenverhalten auf beide Anforderungen gleichzeitig angepasst ist, ist situationsbezogene und zielgerichtete Verhaltensrepertoire des Agenten realisierbar.

Bestimmenden Einfluss auf die Leistung des Agenten sowie dessen erforderlicher Komplexität des Steuerungsteiles, hat der Art und Weise auf welche verwendete Steuerungsparadigma die Umgebung repräsentiert. Durchgeführte Analysen zeigen, dass für die Anwendung innerhalb Montagesystems solche Agentenarchitektur erwünscht ist, bei welcher die Repräsentation der Aufgabe am wenigsten Merkmale der reaktiven Steuerung behindert. Daher ist eine Agentensteuerung vorgeschlagen, die Aufgabevorstellung als Teil der reaktiven Steuerungsschicht hat. Einbeziehung von Aufgaberepräsentation in einzelne Agenten hat positive Auswirkungen auf Vereinfachung der Steuerung auf Montagesystemebene gebracht. Außerdem ist durch solche Informationsverteilung die Kooperation zwischen Agentenklassen sowie Anforderungen auf Kommunikationssystem bestimmt.

Dem vorgeschlagene Bionic Assembly System (BAS) ist eine Integration von agentenbasierenden Steuerung, autonomen Agenten unterschiedlicher Arten und Funktionalitäten und menschliche Operateure zugrunde gelegt. Diese Komponenten bilden ein modulares selbsorganisierendes Montagesystem.

BAS Montagesystem befindet sich im CIM Umgebung als ein untergeordnetes Produktionssystem. Den Aktivitäten von BAS Montagesystems wird durch CIM Subordination ein Rahmen gesetzt. Die Selbstorganisation von BAS Montagesystem kann sich nur innerhalb dieser Rahmen verwirklichen. Die zwei Ansätze sind widersprüchlich und grundsätzlich nicht kompatibel. Wesentlicher Beitrag dieser Arbeit ist die

Harmonisierung dieser konträren Ansätze im Rahmen von Arbeitszenarien von BAS Montagesystem.

Das Kommunikationssystem ist aus zwei globalen Kommunikationskanälen aufgebaut. Vertikale Kommunikation bezeichnet funktionelle Kopplung zwischen übergeordnetem Computer Integrierte Produktionsstruktur und Bionic Assembly System (BAS) Montagesystem. Horizontale Kommunikation dagegen ermöglicht und strukturiert Kooperation in der selbst organisierenden Weise. Jeder Agent trifft aufgrund ausgetauschter Informationen die Entscheidung autonom welche Montagestation als nächste er ansteuern will. Da dieser Informationsaustausch erst nach aktuellem Montageschritt durchgeführt wird, ist Planungshorizont auf einen Montageschritt voraus reduziert. Diese Reduzierung hat dazu geführt dass Störungen jeglicher Art automatisch in den Plan des Agenten einbezogen werden. Die nächste Aktion des Agenten ist daher automatisch optimiert.

Konzept von BAS hat folgende Vorteile:

Modularität. Modularer Aufbau der Montageanlage erlaubt einen stufenartigen Aufbau sowie Plug & Produce Funktion. Damit wird die Einführung neuer Technologie erleichtert. Solange Kommunikationsschnittstelle kompatibel ist und der mechanische Aufbau den Anforderungen der Montage genügt kann man neue Einheiten in das existierende System integrieren. Dies ist insbesondere wichtig, da sich mobile Roboter Technologie rasant entwickelt.

Flexibilität in Bezug auf Kapazität und Funktionalität. Das Montagesystem kann ohne mechanische Eingriffe auf neue Montageaufgaben umgestellt werden. Damit ist das Montagesystem in der Lage, reibungslos einen neuen Mix von Produktvarianten zu verarbeiten. Außerdem ist es möglich die Anlage bedarfsgerecht abzuändern, ohne dass an der Software und Hardware Änderungen vorgenommen werden müssen. Solange Montagestationen erforderliche Montageschritte ausführen können, ist physischer Umbau der Montagestationen nicht erforderlich. Man kann neue Montagestationen auf die Systemgrenze einführen. Freie Transportagenten könnten online mit neuen Montagereihenfolgen nachgerüstet werden, ohne die Leistung der gegenwärtigen Produktion zu benachteiligen.

Selbstoptimierung. Abhängig von der Montageaufgabe und Status der Montagestationen, kann jeder Transportagent eigenen Ablauf arrangieren. Montageteile, Montagegruppe und Werkzeuge können vollkommen frei von einem zum anderen Punkt des Montagesystems gebracht werden. Im Falle eines Montagestationsausfalles liefert diese Art der Redundanz dem System die Flexibilität, einer Transportagenten zu einer anderen Montagestation umzuleiten. BAS Montagesysteme können durch Selbstorganisation diese Probleme vielfach

besser meistern. Typische Konfliktsituationen der Arbeitszenarien von BAS wurden im Rahmen dieser Arbeit detailliert untersucht und einer Lösung herangeführt.

Robustheit. Das Montagessystem ist wesentlich robuster, da durch den Ausfall eines Transportagenten nicht die ganze Montage in Frage gestellt wird. Beim Ausfall einer Montagestation wird auf einem anderen Platz innerhalb Montagessystems alternative Montagestation mit gleichen Eigenschaften eingeführt.

Inbetriebnahme. Diese Phase wird Schritt für Schritt nach der vorgegebenen Reihenfolge und Prozedur durchgeführt. Selbstorganisierendes BAS Montagesystem meistert diese Phase sehr flexibel, die Prozedur ist sehr einfach und Reihenfolge ist nahezu frei wählbar.

Die Entwicklung selbstorganisierender Montagessystemen befindet sich erst am Anfang. So ein Montagessystem existiert zurzeit nicht. Entwickeltes Konzept und Simulationsergebnisse innerhalb dieser Arbeit sowie neueste Veröffentlichungen auf diesem Gebiet (Arai, 2004) zeigen, dass das auf autonome mobile Roboter basierende Montagessysteme haben ein sehr großes Potential das mit anderen Produktionskonzepten nicht erreichbar ist. Jedoch bleiben die technische Realisierung der prioritätsbasierenden Anordnung und der Rekonfiguration von Transportagenten in der Warteschlange der Montagestationen ein schwieriges Problem. Dies ist ein logischer weiterer Schritt in der Entwicklung des BAS Montagesystems. Es ist angemessen, zu behaupten, dass die Lösung dieses Problems ein sehr bedeutsamer Schritt für die Behebung des Rekonfigurationsproblems der Montagestationen ist. Genau so wird damit die autonome Entstehung eines bedarfsgerechten Montagessystems vorangetrieben.

Abbildungsverzeichnis

Abb. 2.1.	Klassifikation von Agenten.....	6
Abb. 2.2.	Komponenten, die einen Agent in der realen Umgebung besitzen soll.....	11
Abb. 2.3.	Bedeutung des Begriffes Verhalten im behavioristischen Robotik und in der Verhaltensbiologie.....	15
Abb. 3.1.	Blockdiagramm des Regelkreises.....	19
Abb. 3.2.	Blockdiagramm der Kaskadenregelung.....	20
Abb. 3.3.	Blockdiagramm der offenen Regelung.....	21
Abb. 3.4.	Gleichgewicht zwischen drei Elementen.....	23
Abb. 3.5.	Zwei Agenten. Nach Breitenbergs Klassifikation handelt sich um Agent Typ 2 „Aggression“ (linken) und Typ 1 „Fear“ (rechten).....	24
Abb. 3.6.	Abhängigkeit der Agenten Umgebung und Aufgabe.....	25
Abb. 3.7.	Schaffung des ausbalancierten ausgeglichenen Systems.....	27
Abb. 4.1.	Kontrollparadigmen.....	34
Abb. 4.2.	Wegweiser für Agentenkontrollparadigma Auswahl.....	43
Abb. 5.1.	Arten von Koordinationsmechanismen.....	44
Abb. 5.2.	Subsumption Architektur. Priorität basierend Aktion Selektionssystem.....	45
Abb. 5.3.	Aktionschema des Agenten mit drei kompetitive Kompetenzebenen.....	47
Abb. 5.4.	Kooperative reaktive Steuerung.....	48
Abb. 5.5.	Die beiden einfachsten Attraktor-Typen.....	52
Abb. 5.6.	Beispiel vom linearen Potentialfeld.....	54
Abb. 5.7.	Exponentiales Potentialfeld Magnitude Profil.....	55
Abb. 5.8.	Die Verteilung abgestrahlten Energie im Raum.....	57
Abb. 5.9.	Zwei typische Hinderniskonfigurationen, die das lokale Minima erzeugen.....	59
Abb. 5.10.	Bewegungssteuerung des Agenten.....	61
Abb. 5.11.	Kollisionsvermeidung mitteln einfache rückgekoppelte Neuronale Netzwerk.....	64
Abb. 5.12.	Repräsentation von Agenten Geometrie und Sensoren Konfiguration in MobotSim Simulator.....	68
Abb. 5.13.	Agenten Pfad für 12/10 und 18/10 Konfiguration.....	69
Abb. 5.14.	Agenten Pfad mit 4 Ultraschall und 12 Infrarotsensoren.....	70
Abb. 6.1.	Fertigungssystem und dessen globale Kontrolle.....	72
Abb. 6.2.	Hierarchie der Agenten.....	75
Abb. 6.3.	Aspekte der Rekonfiguration.....	79
Abb. 7.1.	Produkt Hierarchie für den generellen Fall der Montage.....	82
Abb. 7.2.	Konditionen für die ökonomische Anwendung der verschiedenen Montagesysteme (Boothroyd, 1984).....	83
Abb. 7.3.	Grundriss des flexiblen Fließbandmontagesystems (Katalinic, 1999).....	89
Abb. 7.4.	Explosion Darstellung der Elektromotor und Montagereihfolge (ATB Katalog, 2000).....	91
Abb. 7.5.	Die erste Stufe der Entwicklung des selbstorganisierenden Montagekonzepts (Katalinic, 1999).....	94
Abb. 7.6.	Die zweite Stufe der Entwicklung des „Lineless“ Montagekonzepts (Katalinic, 2000).....	98

Abb. 7.7.	Interaktion zwischen Hierarchical Scheduler und Job Shop Floor Kontroller	102
Abb. 7.8.	Auftrag Pool des Transportbereichagenten	104
Abb. 7.9.	Integration des selbst organisierenden Montagesystems in CIM Struktur aus der Sicht des Informationsaustausches	105
Abb. 7.10.	Vertikale Kommunikation Struktur und Informationsaustausch	107
Abb. 7.11.	Strukturelle Kommunikation rein reaktive Agenten	109
Abb. 7.12.	Zielansteuerung nach Trigonometrische Methode.....	112
Abb. 7.13.	Horizontale Kommunikation Struktur und Informationsaustausch.....	114
Abb. 7.14.	Example of the biologic self-organising assembly system.....	117
Abb. 7.15.	Flussdiagramm von Reparaturagent Ablauf	120
Abb. 7.16.	Flussdiagramm von Transportagent Ablauf.....	121
Abb. 7.17.	Flussdiagramm von Montageagent Ablauf	122
Abb. 7.18.	Flussdiagramm von Energieagent Ablauf.....	123
Abb. 7.19.	Flussdiagramm von Prüfstation Ablauf	124
Abb. 7.20.	Flussdiagramm von Entlade Station Ablauf.....	125
Abb. 8.1.	Planungshorizont	135
Abb. 8.2.	Drei Phasen des Verhandlungsprotokolls.....	136
Abb. 8.3.	Beispiel der Verhandlung zwischen drei Transportagenten und Montagestation.....	138
Abb. 8.4.	Teil der Verhandlung aus dem Sicht der Montagestationen.....	139
Abb. 8.5.	Ablauf der Kommunikation in Systemzeit	140
Abb. 8.6.	Ablauf der Kommunikation in Systemzeit	141
Abb. 8.7.	Teil der Verhandlung aus dem Sicht der Transportagenten	142
Abb. 8.8.	Bionic Assembly System Model in ARENA 7.0.....	145
Abb. 8.9.	Production time	146
Abb. 8.10.	WIP per assembly station	146
Abb. 8.11.	Tardy motors	147
Abb. 8.12.	Lead time.....	147
Abb. 8.13.	Tardy motors	148
Abb. 8.14.	WIP.....	148
Abb. 8.15.	Production time	148
Abb. 8.16.	Leistungsvergleich für den Fall der 5-11 Montageschritte und MTBA zwischen 100 und 350	149
Abb. 8.17.	Leistungsunterschied als Funktion der Kapazitäts- und Produktkomplexitätsschwankung.....	149
Abb. 9.1.	Ausbalanciertes Zustand.....	151
Abb. 9.2.	Ablauf der Rekonfiguration.....	152
Abb. 9.3.	Rekonfiguration	153
Abb. 9.4.	Auflösen der Warteschlange ASi+1 nach Reihenfolge Rn	154

Tabellenverzeichnis

Tab. 4.1.	Kontrollparadigmen und ihre Hauptmerkmale	40
Tab. 5.1.	Gewichte für acht Neuronen in der Eingangsschicht.....	66
Tab. 5.3.	Ergebnisse erster Simulation	69
Tab. 5.4.	Ergebnis Optimierte Sensoren Zahl und Einordnung	70
Tab. 5.5.	Ergebnisse zweiter Simulation.....	70
Tab. 6.1.	Die Fragen, auf der das Scheduling antwortet.....	71
Tab. 6.2.	Vergleichung der Agent basierende und Klassische Produktionssteuerung	77
Tab. 7.1.	Montageorganisationen, Anwendung und Hauptmerkmale (Schulte, 2000)	84
Tab. 7.2.	Zerlegung von Agentenklassen, deren Funktion und Auflegung	97
Tab. 7.3.	Vergleich von Hauptmerkmalen der klassischen und selbstorganisierenden Systeme.	99
Tab. 7.4.	Die Verarbeitungsschritte.....	100
Tab. 7.5.	Die Aufteilung der Verarbeitungsschritte in der zentralisieren Organisation	101
Tab. 7.6.	Die Aufteilung von Verarbeitungsschritten in der völlig autonomen Organisation.....	101
Tab. 7.7.	Die Aufteilung von Verarbeitungsschritten im Montagesystem	101
Tab. 7.8.	Agentenstatus	106
Tab. 7.9.	Auslegung und Funktion von dem Radio RF System nach Agentenklassen	113
Tab. 7.10.	Anforderungen und Auflegung von Kommunikationssystem	114
Tab. 7.11.	Dateistet für einen Motortyp.....	116
Tab. 8.1.	Komponenten der Anfrage nach Reihenfolge und dessen Zweck.....	136
Tab. 8.2.	Komponenten des Angebots nach Reihenfolge und dessen Zweck	137
Tab. 8.3.	Aufteilung der Montageaufgaben.....	144
Tab. 8.4.	Transportwege	145
Tab. 8.5.	Prozentsatz der Verbesserung im Vergleich zu vier traditionellen Kriterien.....	147
Tab. 9.1.	Technologische Montagezeiten	156
Tab. 9.2.	Auftrag für Montagestation MSi.....	157
Tab. 9.3.	Prioritäten im BAS	158

Literaturverzeichnis

- Adams, B.; Breazeal, C.; Brooks, R. A. & Scassellati, B. (2000). Humanoid Robots: A New Kind of Tool, *IEEE Intelligent Systems and Their Applications: Special Issue on Humanoid Robotics*, Vol. 15, No. 4, July/August 2000, pp. 25-31.
- Ahrns, I.; Bruske, J. & Sommer, G. (1995). On-line learning with dynamic cell structures. In: *Proceedings of the International Conference on Artificial Neural Networks (ICANN 95)*, October 1995, pp. 141-146.
- Altman, J. S. & Kien, J. (1989). New models for motor control. *Neural Computation*, Vol. 1. pp. 173-183.
- Arai, T.; Sugi, M.; Maeda, Y.; Aiyama, Y. & Harada, T. (2004). A Holonic Architecture for Easy Reconfiguration of Robotic Assembly Systems, *IEEE Transactions on Robotica and Automation*, Vol.20, No.3, June 2004
- Arimoto, S. (2000). Robotics research toward explication of everyday physics. *International Journal of Robotics Research*, Vol. 18, No.11, pp. 1056 – 1063.
- Arkin, R. C. & Balch, T. (1997). AuRA: Principles and Practice in Review, *Journal of Experimental and Theoretical Artificial Intelligence*, Vol. 9, No.2, pp. 175-189
- Arkin, R. C. (1998). *Behavior-Based Robotics*, MIT Press, Massachusetts Institute of Technology
- Arrowsmith, D. K. (1994). *Dynamische Systeme : mathematische Grundlagen*; Spektrum Akademische Verlag, ISBN 3-86025-308-5, Heidelberg, Deutschland
- Bäck, T. & Schwefel, H. (1993). An overview of evolutionary algorithms for parameter optimization, *Evolutionary Computation* 1(1), pp. 1-23.
- Ballard, D.; Hayhoe, M. & Pelz, J. (1995). Memory representations in natural tasks, *Journal of Cognitive Neuroscience*, pp. 66-80.
- Ballard, D. (1991). Animate vision. *Artificial Intelligence*, Vol. 48, pp. 57-86.
- Baratoff, G. & Neumann, H. (1998). Robot navigation by combining central and peripheral optical flow detection on a space-variant map, *Proceedings of Fourteenth International Conference on Pattern Recognition*, Los Alamitos, CA, IEEE Computer Society, pp. 1804-1807.
- Bauer, H. U. & Pawlezick, K. R. (1995). Quantifying the neighbourhood preservation of self-organizing feature maps, *IEEE Transactions on Neural Networks*, 3, pp. 570-578.
- Beer, R. D. & Gallagher, J. C. (1992). Evolving dynamical neural networks for adaptive behaviour, *Adaptive Behavior*, pp. 1-91
- Beer, R. D. (1995). A Dynamical Systems Perspective on Agent-Environment Interaction, *Artificial Intelligence*, Vol. 72 (1/2), pp.173-216
- Beer, R. D. (1995). A dynamical systems perspective on agent-environment interaction. *Artificial Intelligence* 72 pp. 173-215.
- Beer, R. D. (in press). The dynamics of adaptive behavior: a research program. To appear in: *Robotics and Autonomous Systems, Special Issue on Practice and Future of Autonomous Agents*, R. Pfeifer, and R. Brooks (Ed.)
- Beer, R. D. (1995). On the dynamics of small continuous-time recurrent neural networks. *Adaptive Behavior* 3(4), pp. 469-509.
- Bennett, C. H. (1991). The Thermodynamics of Computation: a Review. *International Journal of Theoretical Physics*, Vol. 21 (12), pp. 905-940
- Best, R. (1987). *Theorie und Anwendungen des Phase-locked Loops*, AT-Verlag, ISBN 3-85502-132-5, Aarau, Schweiz
- Blythe, J. & Veloso, M. (1997). Analogical Replay for Efficient Conditional Planning, *Proceedings of American Association of Artificial Intelligence (AAAI- 97)*, pp. 668-673.

- Bonabeau, E.; Dorigo, M. & Theraulaz, G. (1999) *Swarm Intelligence: From Natural to Artificial Systems*, New York, Oxford University Press
- Borenstein, J.; Everett, H. R. & L. Feng. (1996). *Where am I? Sensors and Methods for Mobile Robot Positioning*, Technical Report, The University of Michigan
- Boutilier, C. & Brafman, R. I. (1997). *Planning with Concurrent Interacting Actions*, Proceedings of American Association of Artificial Intelligence (AAAI- 97), pp. 720-726.
- Brafman, R. I. (1997). *A Heuristic Variable Grid Solution Method for POMDPs*, Proceedings of American Association of Artificial Intelligence (AAAI-97), pp. 727-733.
- Braitenberg, V. (1984). *Vehicles: experiments in synthetic psychology*, MIT Press, Cambridge, Massachusetts
- Brooks, R. A. & Stein, L. A. (1994). *Building brains for bodies*, *Autonomous Robots*, Vol.1, pp. 7-25.
- Brooks, R. A. (1990). *Elephants Don't Play Chess*, *Robotics and Autonomous Systems* (6), pp. 3-15.
- Brooks, R. A. (1991). *Intelligence without Reason*, Proceedings of the 1991 International Joint Conference on Artificial Intelligence, pp. 569-595.
- Brooks, R. A. (1991). *Intelligence without representation*, *Artificial Intelligence*, Vol. 47, pp. 139-159.
- Brooks, R. A. (1994). *Coherent behavior from many adaptive processes*, In: D. Cliff, P. Husbands, J.-A. Meyer, and S.W. Wilson (Ed.). *From animals to animats 3*. Proceeding SAB'94, pp. 22-29.
- Brooks, R. A. (1996). *Autonomy as Tolerance to Perturbation*, Proceedings of the International Workshop: Towards Real Autonomy, Osaka, Japan. November 1996.
- Brooks, R. A. (1999). *Cambrian intelligence*. Cambridge, Massachusetts, MIT Press.
- Brooks, R. A. (2001). *Artificial Intelligence and Robot Development*, Proceedings of the Heinz Nixdorf Museum Paderborn Podium, Ferdinand Shoningh Press, October, 2001, pp. 15-37.
- Brooks, R. A.; Breazeal, C.; Irie, R.; Kemp, C.; Marjanovic, M.; Scassellati, B. & Williamson, M. (1999). *The Cog Project: Building a Humanoid Robot*, *Computation for Metaphors, Analogy, and Agents*. C. Nehaniv (Ed.), *Lecture Notes in Artificial Intelligence* 1562, pp. 52-87, Springer, New York
- Brückner, S. & Parunak, H. V. D. (2000). *Multiple Pheromones for Improved Guidance*, Proceedings of Symposium on Advanced Enterprise Control
- Brückner, S. (2000). *Return from the Ant: Synthetic Ecosystems for Manufacturing Control*, Thesis at Humboldt University Berlin, Department of Computer Science
- Bushnell, E. M. & Boudreau, J. P. (1993). *Motor development in the mind: The potential role of motor abilities as a determinant of aspects of perceptual development*. *Child Development*, Vol. 64, pp. 1005-1021
- Bussmann, S. & Schild, K. (2001). *An Agent-based Approach to the Control of Flexible Production Systems*, Proceedings of the 8th IEEE International Conference on Emergent Technologies and Factory Automation (ETFA 2001), Vol.2, pp.481-488, Antibes Juan-les-pins, France
- Bussmann, S. & Sieverding, J. (2001). *Holonic Control of an Engine Assembly Plant - An Industrial Evaluation*, Proceedings of the 2001 IEEE Systems, Man, and Cybernetics Conference, pp.169-174. Springer-Verlag: Berlin, Germany
- Churchland, P.; Ramachandran, V. & Sejnowski, T. (1994). *A Critique of Pure Vision*, in C. Koch & J. Davis, (Ed.), *Large-Scale Neuronal Theories of the Brain*, MIT Press.
- Clancey, W. J. (1991). *The frame of reference problem in the design of intelligent machines*. In K. van Lehn (Ed.). *Architectures for intelligence*, Hillsdale, N.J., Erlbaum
- Cruse, H. (1990). *Coordination of leg movement in walking animals*. In: J.-A. Meyer, and S.W. Wilson (Ed.), *From animals to animat 1*. Proceeding SAB'90, pp. 105-119
- Cruse, H.; Bartling, Ch.; Cymbalyuk, G.; Dean, J. & Dreifert, M. (1994). *A Modular Artificial Neural Net for Controlling a Six-Legged Walking System*, Proceedings der Konferenz Perational Intelligence in Robotics: from Sensorimotor Intelligence to Collective Behavior, Report Nr. 10 der Forschungsgruppe Perational Intelligence, pp. 17-36, Zentrum für interdisziplinäre Forschung (ZIF), Universität Bielefeld
- Cruse, H.; Dean, J. & Ritter, H. (1998). *Die Entdeckung der Intelligenz oder Können Ameisen denken?*, Deutscher Taschenbuch Verlag, ISBN: 3-406-44073-8, München, Deutschland

- D'Inverno, M. & Luck, M. (2001). *Understanding Agent Systems*, Sprinter Verlag, Springer series on agent technology, ISBN 3-540-41975-6, Heidelberg, Germany
- Damasio, A. R. (1994). *Descartes' Error*, G.P. Putnam's Sons.
- Dellaert, F. & Beer, R. D. (1996). A developmental model for the evolution of complete autonomous agents, *From Animals to Animats*. Proc. of the 4th Int. Conf. on Simulation of Adaptive Behavior, P. Maes, M.J. Mataric, J.-A. Meyer, J. Pollack, and S. W. Wilson (Ed.), Cambridge, Massachusetts, MIT Press
- Deneubourg, J. L.; Goss, S.; Franks, N. & Chretien, L. (1991). The Dynamics of Collective Sorting: Robot-Like Ants and Ant-Like Robots, *From Animals to Animats 1*. Proceedings of First International Conference on the Simulation of Adaptive Behavior, Meyer, J.A., Wilson, S. (Ed.), pp. 356-365, MIT Press
- Dennett, D. C. (1999). *Cognitive Wheels: The Frame Problem of AI, Minds, Machines, and Evolution: Philosophical Studies*, C. Hookway (Ed.), London, Cambridge University Press
- Dreyfus, H. L. & Dreyfus, S. E. (1990). *Making a Mind versus Modelling the Brain: Artificial Intelligence Back at a Branch-Point*, The Philosophy of Artificial Intelligence, Boden, M. (Ed.), Oxford University Press
- Drogoul, A. (1995). When Ants Play Chess Or Can Strategies Emerge from Tactical Behaviors?, *Proceedings of Fifth European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, pp. 13-27, Springer
- Eggenberger, P. (1997). Evolving morphologies of simulated 3D organisms based on differential gene expression, *Proceedings of the 4th European Conference on Artificial Life*, P. Husbands, and I. Harvey (Ed.), Cambridge, Massachusetts, MIT Press.
- Eggenberger, P. (1999). Evolution of three-dimensional, artificial organisms: simulations of developmental processes, Unpublished PhD thesis, Faculty of Medicine, University of Zurich
- Elman, J. L. (1990). Finding Structure in Time. *Cognitive Science* 14 Nr. 2, 179-211
- Ferber, J. (1999). *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*, Addison Wesley Longman Limited, ISBN 0-201-36048-9, Singapore
- Ferrari, F.; Nielsen, P. Q. J. & Sandini, G. (1995). Space variant imaging. *Sensor Review*, 15, 17-20.
- Ferrell, C. & Kemp, C. (1996). An Ontogenetic Perspective to Scaling Sensorimotor Intelligence, *Embodied Cognition and Action*, Papers from the 1996 AAAI Fall Symposium, AAAI Press
- Floreano, D. & Urzelai, J. (2000). Evolutionary robots with online self-organization and behavioral fitness, *Neural Networks*, Vol. 13, pp. 431-443
- Floreano, D.; Nolfi, S. & Mondada, F. (2001). Co-evolution and ontogenetic change in competing robots, *Robotics and Autonomous Systems*,
- Fogel, B. D. (2000). *Evolutionary Computation: toward a new philosophy of machine intelligence*, IEEE Press, ISBN 0-7803-5379-X, New York; USA
- Franceschini, N.; Pichon, J. M. & Blanes, C. (1992). From insect vision to robot vision. *Phil. Trans. R. Soc. Lond. B*, 337, 283-294.
- French, R. (1999). When Coffee Cups Are Like Old Elephants, or Why Representation Modules Don't Make Sense. In: Riegler, A., Peschl, M. & Stein, A. von (eds.) *Understanding Representation in the Cognitive Sciences*. New York: Kluwer Academic / Plenum Publishers, pp. 93-100.
- Fritzke, B. (1995). Growing cell structures -- a self-organizing network for unsupervised and supervised learning. *Neural Networks*, 7, 1441-1460.
- Fuss, M. A. & Waverman, L. (1992). *Costs and productivity in automobile production : the challenge of Japanese efficiency*, Cambridge University Press, ISBN 0-521-34141-8, Cambridge
- Gat, E. (1991). *Reliable Goal-Directed Reactive Control of Autonomous MobileRobots*. PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, Mass.
- Haken, H. & Wunderlich, A. (1986). Synergetik: Prozesse der Selbstorganisation in der belebten und unbelebten Natur. In Dress et al. 1986, 35-60
- Haken, H. (1983). *Advanced Synergetics - Instability Hierarchies of Self-Organizing Systems and Devices*. Springer, Berlin/Heidelberg (Springer Series in Synergetics Vol. 20)

- Han, J. & Kamber, M. (2001). *Data Mining: Concepts and Techniques*, Morgan Kufmann Publishers, ISBN: 1-55860-489-8, San Francisco, USA
- Hannebauer, M. (2001). *Balancing reactivity and social deliberation in multi-agent systems: from RoboCup to real-world applications*, Springer Verlag Lecture notes in computer science Series, ISBN 3-540-42327-3, Berlin, Deutschland
- Harnad, S. (1990). The symbol grounding problem. *Physica D*, 42(1-3), 335-346.
- Harvey, I. (2000). *Robotics: Philosophy of Mind using a Screwdriver*. *Evolutionary Robotics: From Intelligent Robots to Artificial Life*, Vol. III, T. Gomi (Ed), AAI Books, Ontario, Canada, pp. 207-230, ISBN 0-9698872-3-X.
- Harvey, I.; Husbands, P. & Cliff, D. (1994). Seeing the Light: Artificial Evolution, Real Vision. In: Cliff, D. et al. (Hrsg.) (1994): *From Animals to Animats III: Proc. der Third Int. Conf. on Simulation of Adaptive Behavior*. Bradford/MIT Press, 392-401
- Harvey, I.; Husbands, P.; Cliff, D.; Thompson, A. & Jakobi, N. (1996). Evolutionary robotics: the Sussex approach. *Robotics and Autonomous Systems*, Special Issue on Practice and Future of Autonomous Agents, R. Pfeifer, and R. Brooks (eds.).
- Haugeland, J. (1995). Mind embodied and embedded. In Yu-Houng & Jih-Ching Ho (Eds.), *Mind and Cognition*, pp. 3-38, Taipei, Taiwan: Academia Sinica.
- Hauskrecht, M. (1997). Incremental Methods for computing bounds in partially observable Markov decision processes, in 'Proceedings of American Association of Artificial Intelligence (AAAI-97)', pp. 734-739.
- Helbing, D.; Schweitzer, F.; Keltsch, J. & Molnar, P. (1998). Active Walker Model for the Formation of Human and Animal Trail Systems. Institute of Theoretical Physics, Stuttgart, Germany, 1998. URL <http://xxx.lanl.gov/ps/cond-mat/9806097>.
- Holland, J. H. & Reitmann, J. S. (1978). *Cognitive Systems Based on Adaptive Algorithms*. In: Waterman, D.A. Hayes-Roth, F. (Hrsg.) (1978): *Pattern Directed Inference Systems*. Academic Press, New York, 313-329
- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, 1975
- Holland, J. H. (1986). Escaping Brittleness: The Possibilities of General-Purpose Learning Algorithms Applied to Parallel Rule-Based Systems. In: Anderson, J.R., Michalski, R.S., Carbonell, J. G. (Hrsg.) (1986): *Machine Learning*. Kaufmann, Los Altos, 593-623
- Holmstedt, P. (1997). A Vision of the Future Manufacturing System, in an Agile Perspective, *Proceedings of Proceedings of 29th CIRP International Seminar on Manufacturing Systems*, Iwata, K. & Ueda, K. (Ed.), pp. 28-32, Osaka, Japan
- Horsewill, I. (1992). A simple, cheap, and robust visual navigation system. In: J.-A. Meyer, H.L. Roitblat, and S.W. Wilson (eds.). *From animals to animats 2*. Proc. SAB'92, 129-137.
- Hudson, R. & Schamp E. W. (1995). *Towards a new map of automobile manufacturing in Europe? : new production concepts and spatial restructuring*, Springer Verlag, ISBN 3-540-58812-4, Berlin, Deutschland
- Jaeger, H. & Christaller, T. (1997). Dual dynamics: Designing behaviour systems for autonomous robots. In S. Fujimura and M. Sugisaka, editors, *Proceedings of the International Symposium on Artificial Life and Robotics (AROB '97)*, Beppu, Japan
- Jaeger, H. (1994). *Dynamic Symbol Systems*. Dissertation, Technische Fakultät der Universität Bielefeld
- Jaeger, H. (1994). *Gestaltbildung durch Informationsmaximierung: eine formale und algorithmische Rekonstruktion*. In: Opwis, K. (Hrsg.) (1994): *Proceedings Erste Fachtagung der deutschen kognitionswissenschaftlichen Gesellschaft*, Psychologisches Institut, Universität Freiburg 1994, 32-34
- Jaeger, H. (1995). An Introduction to Dynamic Symbol Systems. In: Hallam, J. (Hrsg.): *Hybrid Problems, Hybrid Solutions*. *Proceedings van de AISB-95*. IOS Press/Ohmsha, Amsterdam 1995, pp. 109-120
- Jaeger, H. (1996). *Brains on wheels: Mobile robots for brain research*. Technical report, GMD - Forschungszentrum Informationstechnik GmbH

- Jennings, N. R.; Wooldridge M. & Bussmann, S. (2001). On the Identification of Agents in the Design of Production Control Systems, in: P. Ciancarini, M.J. Wooldridge (eds.), *Agent-Oriented Software Engineering*, LNCS 1957, pp.141-162. Springer-Verlag: Berlin, Germany
- Jones, J. L. & Flynn, A. M. (1996). *Mobile Roboters*, Addison – Wesley, ISBN: 3-89319-855-5, Bonn, Deutschland
- Kaelbling, L. & Rosenheim, S. (1991). Action and planning in embedded agents. In P. Maes, editor, *Designing Autonomous Agents*, pages 35-48. MIT Press, Cambridge, MA.
- Kaelbling, L. P. (1991). *Foundations of Learning in Autonomous Agents*. *Robotics and Autonomous Systems* 8 (1-2), 1991. Nachdruck in: Van der Velde, W. (Hrsg.) (1993): *Toward Learning Robots*. Bradford/MIT Press
- Katalinic, B. & Kordic, V. (2004). Bionic Assembly System, 29th Scientific-Expert Conference Proceedings HIPNEF 2004, M Stojiljkovic et al. (Ed.), Vrnjacka Banja, Mechanical Faculty, University of Nis, ISBN: 86-80587-31-1; S. 35 - 42.
- Katalinic, B. & Kordic, V. (2003). Control System of Transport Class Assembly Units of Biologic Assembly System, *Annals of DAAAM for 2003 & Proceedings of the 14th DAAAM International Symposium*, B. Katalinic (Ed.), DAAAM International Vienna, Vienna, ISBN 3-901509-34-8; S. 213 - 214.
- Katalinic, B. & Kordic, V. (2002). Bionic Assembly System: Design and Scheduling of Next Generation of Self Organising Complex Flexible Assembly System in CIM Environment, *Proceeding of 35th CIRP International Seminar on Manufacturing Systems - Manufacturing Technology in the Information Age*, Seoul, Korea; 13.05.2002 - 15.05.2002; K. Lee et al. (Ed.), Seoul National University, S. 155 - 162.
- Katalinic, B. & Kordic, V. (2004). Reconfigurable assembly by reactive autonomous robots, *Proceedings of IEEE International Symposium on Robotics and Automation - ISRA 2004*, Queretaro, Mexico; 25.08.2004 - 27.08.2004
- Katalinic, B. & Kordic, V. (2004). Simple reactive robots in assembly domain, *Proceedings of 8th International Conference on Mechatronics Technology*, Hanoi, Vietnam; 08.11.2004 -12.11.2004
- Katalinic, B. & Kordic, V. Stopper, M. (2003). Interaction of Subordination and Self-Organisation in Working Scenarios of Bionic Assembly System, *Proceedings of the 3rd Asian Conference on Automation and Robotics - ASIAR 03*, Bangkok, Thailand, 08.05.2003 - 09.05.2003, ISBN: 974-229-467-4; S. 136 - 141.
- Katalinic, B. & Kordic, V.(2002). Bionic Assembly Systems: Autonomous Agents in Self Organising Complex Flexible Assembly System in CIM Environment, *Proceedings of 4th International Workshop on Emergent Synthesis*, K. Ueda (Ed.); Kobe University, 09.05.2002 - 10.05.2002; (2002), S. 3 - 12.
- Katalinic, B. & Kordic, V.(2003). Integration of Subordination and Self Organisation in Working Scenarios of Bionic Assembly System, *DAAAM International Scientific Book 2003*, B. Katalinic (Ed.), DAAAM International Vienna, Wien, ISBN: 3-901-509-36-4, S. 319 - 330.
- Katalinic, B. & Kordic, V. (2004). Bionic Assembly System: Concept, Structure and Function, *Proceedings of 5th International Conference on Integrated Design and Manufacturing in Mechanical Engineering*, Bath, United Kingdom, 05.04.2004 - 07.04.2004.
- Katalinic, B. (1999). Design of Sheduling Strategies for complex flexible Assembly System for the Mass Production of Electrical Motors, *Proceedings of International Workshop on Emergent Synthesis – IWES 99*, Ueda, K. (Ed.), Kobe, Japan
- Kawai, N. & Hara, F. (1998). Formation of morphology and morpho-function in a linear cluster robotic system. In Pfeifer, R., Blumberg, B., Meyer, J.-A., and Wilson, S.S. (eds.). *From animals to animats 5*. Proc. of the 5th International Conference on Simulation of Adaptive Behavior. Cambridge, Mass.: MIT Press, 459 – 464.
- Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal for Robotics Research*, Vol. 5(1), pp. 90-98
- Kikuchi, K. & Hara, F. (1998). Evolutionary design of mophology and intelligence in robotic system using genetic programing, *Formation of morphology and morpho-function in a linear cluster robotic system*. In Pfeifer, R., Blumberg, B., Meyer, J.-A., and Wilson, S.S. (eds.). *From animals*

- to animats 5. Proc. of the 5th International Conference on Simulation of Adaptive Behavior. Cambridge, Mass.: MIT Press, 540-545.
- Konolige, K. & Myers, K. (1996). The saphira architecture for autonomous mobile robots. Technical report, Artificial Intelligence Center SRI International, Menlo Park, CA
- Koren, Y. (1985). Robotics for engineers, McGraw-Hill, ISBN 0-07-035399-9, New York, USA
- Kosecka, J. & Bajcsy, R. (1993). Discrete event systems for autonomous mobile agents. In Proceedings Intelligent Robotics Systems 1993, pages 21-31, Zakopane, Slovak Republic.
- Krause, U. & Neumann, T. (1999). Differenzgleichungen und diskrete dynamische Systeme Eine Einführung in Theorie und Anwendungen, Teubner, ISBN 3-519-02639-2, Stuttgart, Deutschland
- Kruschke, J. K. (1992). ALCOVE: An exemplar-based connectionist model of category learning, *Psychological Review*, 97, 225-252.
- Lambrinos, D. & Scheier, C. (1995). Extended Braitenberg Architectures. AILab Technical Report, University of Zurich, No. 95-10.
- Lorenz, K. Z. & Tinbergen, N. (1939). Taxis und Instinkthandlung in der Eirollbewegung der Graugans. *Zeitschrift für Tierpsychologie* 2(1). Nachgedruckt in: Lorenz, K. (1984) Über tierisches und menschliches Verhalten. München: Piper.
- Maes, P. (1991). A bottom-up mechanism for behavior selection in an artificial creature. In: J.-A. Meyer, and S.W. Wilson (eds.). From animals to animats 1. Proc. SAB'90, 238-246.
- Mataric, M. (1994). Interaction and Intelligent Behavior, Ph. D. Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Massachusetts; USA
- Mataric, M. (1999). Behavior-Based Robotics, In: MIT Encyclopedia of Cognitive Sciences, Robert A. Wilson & Frank C. Keil, (Ed.), 74-77, MIT Press, Available from: <http://cognet.mit.edu/MITECS/>, Accessed: 2000-10-12
- Maturana, H. R. (1982). Erkennen: Die Organisation und Verkörperung von Wirklichkeit. Braunschweig: Vieweg
- Maturana, H. R. & Varela, F. J. (1984). Der Baum der Erkenntnis: die biologischen Wurzeln des menschlichen Erkennens. Scherz, Bern/München 1987; Goldmann Verlag
- McFarland, D. & Bösser, M. (1993). Intelligent behavior in animals and robots. MIT Press.
- McFarland, D. (1994). Towards Robot Cooperation. In: Cliff, D. et al. (eds.) (1994): From Animals to Animats III: Proceedings of the Third Int. Conf. on Simulation of Adaptive Behavior. MIT Press/Bradford Books, Cambridge, Mass., 440-444
- McFarlane, D. C. & Bussmann, S. (2000). Developments in Holonic Production Planning and Control, in: *Int. Journal of Production Planning and Control*, Vol. 11, No. 6, 2000, pp. 522 - 536.
- Murphy, R. R. (2000). Introduction to AI Robotics, A Bradford Book / MIT Press, ISBN: 0-262-13383-0, Cambridge, Massachusetts, USA
- Nehmzow, U. (2000). Mobile Robotics: A Practical Introduction, Springer Verlag, ISBN 1-85233-173-9, London, United Kingdom
- Newell, A. & Simon, H. A. (1976). Computer Science as Empirical Enquiry: Symbols and Search. *Communications of the association for computing machinery* 19: 113-126.
- Newell, A. (1990). Unified theories of cognition. Cambridge, Mass. & Harvard University Press.
- Parunak H. V. D. (1998). Industrial and Practical Applications of DAI." In G.Weiss, editor, Introduction to Distributed Artificial Intelligence. MIT Press.
- Parunak, D. V. H. (1998). What can Agents do in Industry, and Why? An Overview of Industrially-Oriented R&D at CEC, Proceedings of CIA 98, USA
- Parunak, H. V. D. & Brueckner, S. (2000). Ant-Like Missionaries and Cannibals: Synthetic Pheromones for Distributed Motion Control. In Proceedings of Fourth International Conference on Autonomous Agents (Agents 2000), pages 467-474.
- Parunak, H. V. D. & Brueckner, S. (2001). Entropy and Self-Organization in Multi-Agent Systems, Proceedings of the Fifth International Conference on Autonomous Agents, Müller, J. P. at al. (Ed.), pp. 124-130, ISBN 1-58113-326-X, New York, USA
- Parunak, H. V. D. & Brueckner, S. (2001). Entropy and Self-Organization in Multi-Agent Systems. In Proceedings of International Conference on Autonomous Agents

- Parunak, H. V. D. (1997). 'Go to the Ant': Engineering Principles from Natural Agent Systems. *Annals of Operations Research*, 75:69-101
- Parunak, H. V. D. (2000). Adaptive control of Distributed Agents through Pheromone Techniques and Interactive Visualization. Web page, www.erim.org/cec/projects/adaptiv/.
- Parunak, H. V. D.; Brueckner, S.; Sauter, J. & Matthews, R. (2000). Distinguishing Environmental and Agent Dynamics: A Case Study in Abstraction and Alternative Modeling Technologies. In A. Omicini, R. Tolksdorf, and F. Zambonelli, Editors, *Engineering Societies in the Agents' World (ESAW'00)*, vol. LNAI 1972, Lecture Notes in Artificial Intelligence, pages 19-33. Springer, Berlin.
- Parunak, H. V. D.; Baker, A. D. & Clark, S. J. (1997). The AARIA Agent Architecture: An Example of Requirements-Driven Agent-Based System Design." *Proceedings of the First International Conference on Autonomous Agents (ICAA-97)*.
- Peklenik, J. (1997). Structural and operational complexity of future manufacturing systems, *Proceedings of Proceedings of 29th CIRP International Seminar on Manufacturing Systems*, Iwata, K. & Ueda, K. (Ed.), pp. 1-8, Osaka, Japan
- Peschl, M. & Riegler, A. (1999). Does Representation Need Reality? In: Riegler, A., Peschl, M. & Stein, A. von (eds.) *Understanding Representation in the Cognitive Sciences*. New York: Kluwer Academic / Plenum Publishers, pp. 9-17.
- Peters, J. (1997). Manufacturing, a Vector of Worldwide Culture, *Proceedings of Proceedings of 29th CIRP International Seminar on Manufacturing Systems*, Iwata, K. & Ueda, K. (Ed.), pp. 9-13, Osaka, Japan
- Pfeifer, R. & Schaad R. (1995). Die Neue Künstliche Intelligenz, Available from: <http://www.ifi.unizh.ch/ailab/publications/1995.html> Accessed: 2001-02-19
- Pfeifer, R. & Verschure, P. (1992). Distributed adaptive control: A paradigm for designing autonomous agents. In F. J. Varela and P. Bourguin, editors, *Toward a practice of autonomous systems: Proceedings of the First European Conference on Artificial Life*, Cambridge, MA: MIT Press.
- Pfeifer, R. (1995). Cognition--perspectives from autonomous agents. *Robotics and Autonomous Systems*, 15, 47-70.
- Pfeifer, R. (1996). Building Fungus Eaters: Design principles of autonomous agents. *Proc. SAB'96*.
- Pfeifer, R. (1996). Symbols, patterns, and behavior: beyond the information-processing metaphor. In A. Kent and J.G. Williams (eds.). *Encyclopedia of Microcomputers*, Vol. 17, 253-275. New York: Marcel Dekker, Inc.
- Pfeifer, R. (1997). Teaching powerful ideas with autonomous mobile robots. *Journal of Computer Science Education*, 7, 161-186.
- Pfeifer, R. (1999). Dynamics, morphology, and materials in the emergence of cognition. In W. Burgard, T. Christaller, and A.B. Cremers (eds.). *KI-99: Advances in Artificial Intelligence. Lecture Notes in Artificial Intelligence 1701*. Berlin: Springer, 27-44.
- Pfeifer, R. (2000). On the role of Morphology and Materials in Adaptive Behavior, Available from: <http://www.ifi.unizh.ch/groups/ailab/publications/2000.html> Accessed: 2001-01-20
- Pfeifer, R. & Scheier, C. (1999). *Understanding Intelligence*, MIT Press, ISBN 0-262-16181-8, Hong Kong
- Pfeifer, R., & Scheier, C. (in press). Sensory-motor coordination: the metaphor and beyond. To appear in: *Robotics and Autonomous Systems, Special Issue on Practice and Future of Autonomous Agents*, R. Pfeifer, and R. Brooks (Ed.).
- Pirjanian, P. (1999). Behavior coordination mechanisms - state of the art. Technical report, USC Robotics Research Laboratory, University of Southern California, Los Angeles, CA
- Popper, K. R. (1979). *The Bucket and the Searchlight: Two Theories of Knowledge*. In: *Objective Knowledge: An Evolutionary Approach* (rev. ed.). Oxford: Clarendon Press.
- Rechenberg, I. (1994). *Evolutionsstrategie '94*. Stuttgart-Bad Cannstatt: Frommann-Holzboog.
- Pfeifer, R. (1999). Dynamics, morphology, and materials in the emergence of cognition. *Proc. KI-99, Lecture Notes in Computer Science*. Berlin: Springer, 27-44.

- Regenass, D. (1996). The Didabot Development System. Student Thesis, University of Zurich, Computer Science Department, AI Lab
- Rutkowska, J. C. (1994). Scaling up sensorimotor systems: Constraints from human infancy. *Adaptive Behavior* 2(4):349-373.
- Scheier, C. & Lambrinos, D. (1996). Categorization in a real-world agent using haptic exploration and active perception. In: Proc. SAB96.
- Scheier, C. & Lambrinos, D. (1996). Categorization in a real-world agent using haptic exploration and active vision. Proc. SAB'96.
- Scheier, C. (1996). Incremental category learning in a real world artifact using growing dynamical cell structures. In: Proceeding of the European Symposium on Artificial Neural Networks (ESANN '96), 117-123.
- Seabrook, J. (1997). *Odyssee im Cyberspace*, Metropolitan Verlag, ISBN. 3-89623-102-2; Düsseldorf-Regensburg, Germany
- Sims, K. (1994). Evolving virtual creatures. *Computer Graphics*, 28, 15-34. Sims, K. (1994b). Evolving 3D morphology and behavior by competition. In R. Brooks and P. Maes (Eds.), *Artificial Life IV Proceedings*. Cambridge, MA: MIT Press, 28-39.
- Steels, L. (1993). Building Agents out of Autonomous Behavior Systems. In: Steels, L., Brooks, R. (Hrsg.) (1993): *The Artificial Life Route to Artificial Intelligence: Building Situated Embodied Agents*. Lawrence Erlbaum, New Haven 1993
- Steels, L. (1994). A Case Study in the Behavior-Oriented Design of Autonomous Agents. In: Cliff, D. et al. (Hrsg.) (1994): *From Animals to Animats III. Proceedings der dritten internationalen Konferenz On the Simulation of Adaptive Behavior*, Bradford/MIT Press, 445-452
- Steels, L. (1994). Emergent Functionality in Robotic Agents through On-Line Evolution. In Brooks, R., Maes, P. (Hrsg.) (1994): *Artificial Life IV. Proceedings des Fourth Int. WS on the Synthesis and Simulation of Living Systems*. Bradford/MIT Press, 8-14
- Steinhage, A. & Schöner, G. (in press). Self-calibration based on invariant view recognition: Dynamic approach to navigation. To appear in: *Robotics and Autonomous Systems, Special Issue on Practice and Future of Autonomous Agents*, R. Pfeifer, and R. Brooks (Ed.).
- Sundermeyer, K. & Bussmann, S. (2001). Einführung der Agententechnologie in einem produzierenden Unternehmen - Ein Erfahrungsbericht (in German), in *Wirtschaftsinformatik*, Vol. 43, No. 2, pp.135-142
- Thelen, E. & Smith, L. (1994). *A dynamic systems approach to the development of cognition and action*. Cambridge, Mass., MIT Press, Bradford Books.
- Thorpe, S.J. & Imbert, M. (1989). Biological constraints on connectionist modelling. In R. Pfeifer, Z. Schreier, F. Fogelman-Soulié, and L. Steels (eds.). *Connectionism in Perspective*. Amsterdam: North-Holland, 63-92.
- Ueda, K. & Vaario, J. (1997). The biological manufacturing system: Adaptation to growing complexity and dynamics in manufacturing environment, *Proceedings of 29th CIRP International Seminar on Manufacturing Systems*, Iwata, K. & Ueda, K. (Ed.), pp. 39-44, Osaka, Japan
- Verschure, P. F. M. J. (1992). Taking connectionism seriously: the vague promise of subsymbolism and an alternative. *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*.
- von Foerster, H. (1984). Erkenntnistheorien und Selbstorganisation. *DELFIN* 1984 (4), 6-19. Nachdruck in Schmidt, S.J. 1987a, 133-158
- von Foerster, H. (1984). Principles of Self-Organization in a Socio-Managerial Context. In: Ulrich, H., Probst, G.J.B. 1984, 2-24
- von Foerster, H. (1990). Kausalität, Unordnung, Selbstorganisation. In: Kratky, K. W. & Wallner, F. (eds.) *Grundprinzipien der Selbstorganisation*. Darmstadt: Wiss. Buchgesellschaft.
- Wörn, H., Dillman, R. & Henrich, D. (1998). *Autonome Mobile Systeme*, Springer Verlag, ISBN 3-540-65192-6, Darmstadt, Deutschland
- Yamauchi, B. & Beer, R. D. (1994). Sequential behaviour and learning in evolved dynamical neural networks. *Adaptive Behaviour* 2(3), pp. 219-246.

Lebenslauf

1.0 ALLGEMEINE DATEN

Name: Vedran
Vorname: KORDIC
Geschlecht: männlich
Akademischer Grad: Dipl.-Ing. (Maschinenbau)
Büro: Technische Universität Wien, Karlsplatz 13/311, A-1040 Wien, Österreich
Privat Adresse: Gumpendorferstraße 39, A-1060 Wien, Österreich
Telefon: 0043 699 126 75 412
Geburtsort: Rijeka, Kroatien
Geburtsdatum: 10. Dezember 1972
Eltern: Kapitän Sime, Mag. Petrica
Familienstand: ledig
Sprachkenntnisse: Deutsch, Englisch, Kroatisch, weitere Südslawische Sprachen
Allgemeinkenntnisse: PC-Kenntnisse (MS Office, Web Design: Dreamweaver, Flash, Programmieren: Basic; Simulation Softwarepakete: Arena, Pro Model)
Forschungsschwerpunkte: Produktionstechnik / Flexible Automation / Intelligent Manufacturing Systems; Holonic Manufacturing, Biologic Manufacturing, Adaptive und Intelligente Produktionssysteme / Künstliche Intelligenz / Multiagentensysteme / autonome Mobilroboter / Robotik /
Wissenschaftliche und Professionelle Vereine: Mitglied in der Danube Adria Association for Automation & Manufacturing
Mitglied des Österreichischen Ingenieur- und Architekten Vereins
Industrielle Projekte: Optimierung der Abläufe des Skoda Part Zentrums (SPC), Skoda a. s., Mlada Boleslav, CZ
Prüfabläufe in der Montage, AUDI AG, Ingolstadt, D
Europäische Projekte: CEEPUS A-104/2000 (Slowakei), A-104/2001(Ungarn), A-104/2002 (Slowenien, Polen), A-104/2003(Tschechien)
Editorials: Editor für Intelligente Produktionssysteme in International Journal of Advanced Robotic Systems, ISSN 1729-8806
Spezialkenntnisse: Hochfrequenztechnik, Audioelektronik
Hobby: Tischtennis, Elektronik/Rundfunk

2.0 SCHULBILDUNG

2001- Dissertation, Institut für Fertigungstechnik, TU Wien.
Forschungsgebiet: Anwendung autonomer mobilen Roboter in der flexible rekonfigurierbare Montage

2000-2001	Deutschlehrgang an der Universität Wien
1998-1999	Bundesheerdienst
1991-1998	Technische Fakultät Rijeka, Richtung: Maschinenbau, Universität Rijeka, Kroatien Diplomarbeit Thema: Technical Vision Systems in Industrial Robotics
1987-1991	Mittlere Gesamtschule für Personal in Elektrotechnik
1979-1987	Hauptschule, Rijeka, Kroatien