

4.5 Selbststeuerung logistischer Prozesse mit Agentensystemen

Von Bernd Scholz-Reiter und Hartmut Höhns

4.5.1 Überblick

Unternehmen sind heutzutage einem immer dynamischer werdenden Umfeld ausgesetzt. Die sich ständig verändernden Rahmenbedingungen haben dabei erhebliche Auswirkungen auf die logistischen Prozesse innerhalb des Unternehmens und unternehmensübergreifend.

Nicht zuletzt auf Grund des durch die Globalisierung immer stärker werdenden Wettbewerbsdrucks wird es für die Unternehmen hierbei immer wichtiger, sich weitestgehend alle potenziellen Wettbewerbsvorteile durch eine bessere Beherrschung ihrer Prozesse zu erschließen. Die sprunghafte Entwicklung neuer Informations- und Kommunikationssysteme eröffnet dabei weiterreichende Möglichkeiten für Prozessinnovationen. Einige relevante Veränderungstreiber sind beispielsweise der Aufbau und Betrieb virtueller Unternehmen zur Verbesserung der Prozessabläufe entlang der Wertschöpfungskette, die verstärkte Kooperation in globalen logistischen Verbänden und Allianzen und eine damit einhergehende Zunahme von komplexen internen und unternehmensübergreifenden logistischen Prozessen. Weiterhin ist auch ein fortschreitender und sich verstärkender Wandel hin zu einem Käufermarkt mit der daraus zwingend resultierenden Wichtigkeit der Kundenorientierung als einem entscheidenden Wettbewerbsfaktor zu beobachten, welches unter anderem auch zu Veränderungen im Zielsystem der logistischen Prozesse führen wird. Zusätzlich sind als wichtige Rahmenbedingungen zu verzeichnen, dass die Transportvolumina drastisch ansteigen, wobei gleichzeitig ein Trend zur Atomisierung von Ladungseinheiten und ein Anstieg von Transportfrequenzen im Bereich der Transportlogistik erkennbar wird. In diesem Sinne ist dann letztlich auch eine relative Knappheit von logistischen Infrastrukturen eine sich abzeichnende Konsequenz.

Durch dieses breite Spektrum an Veränderungstreibern und Randbedingungen ergeben sich eine Vielzahl von höchst anspruchsvollen und sogar teilweise widersprüchlichen Anforderungen an die Identifikation, Gestaltung und Implementierung selbststeuernder Prozesse in der Logistik als einer neuen Möglichkeit der Dynamik zu begegnen. Diesbezüglich ermög-

licht und erfordert der Einsatz neuer Informations- und Kommunikationstechnologien, wie beispielsweise RFID-Technologien, neue Steuerungsstrategien, die hier unter dem Oberbegriff Selbststeuerung subsumiert werden. Dabei geht die Idee der Selbststeuerung logistischer Prozesse davon aus, dass die neuen Technologien wie Tags und Label im Bereich der RFID-Technologie und der heutigen BDE (Betriebsdatenerfassung), zukünftig über eine Rechenleistung und -kapazität von heute verfügbaren mobilen Rechnern (Laptops) oder PCs verfügen. In sofern stellt diese Idee eine Projektion in die Zukunft von etwa 10 bis 15 Jahren dar, die derzeit nicht mit konventionellen Technologien, Konzepten und Methoden „von der Stange“ realisierbar sind.

Ziel dieses Kapitels ist es, die Selbststeuerung logistischer Prozesse auf Basis der Verwendung von Agententechnologie beziehungsweise Softwareagenten zu skizzieren. Hierzu wird zunächst der Begriff der Selbststeuerung im Rahmen der Logistik und dessen Ursprünge aus anderen Wissenschaftsdisziplinen näher beleuchtet. Anschließend wird ein Überblick über die Softwareagenten gegeben, der sich von deren Herkunft, über deren Merkmale bis hin zur Technologie erstrecken wird. In einem nächsten Schritt wird in den Entwurf und die Konzeption der selbststeuernder logistischer Prozesse eingeführt, bevor abschließend Ansatzpunkte und Potenziale der Selbststeuerung logistischer Prozess in den Bereichen Produktions- und Transportlogistik, sowie dem Supply Chain Management skizziert werden.

4.5.2 Selbststeuerung im Zusammenhang mit Produktionsplanung und -steuerung

Ziel dieses Abschnitts ist es, die Grundidee und einige Grundkonzepte der Selbststeuerung logistischer Prozesse sowie deren Ursprünge aus anderen Wissenschaftsbereichen zu erläutern. Hierbei ist die Grundannahme, dass die Selbststeuerung logistischer Prozesse eine neue Möglichkeit darstellt, der weiter oben skizzierten erhöhten Dynamik zu begegnen. Diesbezüglich soll die Selbststeuerung logistischer Prozesse trotz hoher Komplexität der Umwelt die Entwicklung und Implementierung neuer, robusterer Prozesse ermöglichen.

4.5.2.1 Ursprünge von Selbststeuerungskonzepten

Die Grundidee der Selbststeuerung logistischer Prozesse, diese wird im nächsten Abschnitt (4.5.2.2) näher beschrieben und definiert, hat verschiedene Wurzeln, überwiegend in den Naturwissenschaften, die beispielswei-

se mit dem Oberbegriff Selbstorganisation in Verbindung gebracht werden können. „Quellwissenschaften“ sind in diesem Zusammenhang unter anderem die Physik und die Biologie. Diese Konzepte sind für die jüngeren Wissenschaftsbereiche wie die Informatik (z.B. Künstliche Intelligenz, Rechnernetze) und die Ingenieurwissenschaften (z.B. Regelungstechnik beziehungsweise -theorie) durch die sprunghafte Entwicklung der verfügbaren Technologien oder tieferen Durchdringung technologischer Probleme nun besonders interessant geworden. Dieses Phänomen ist jedoch nicht ganz neu und konnte beispielsweise auch schon in den Ideen und Konzepten zum Fraktalen Unternehmen (Warnecke 1993) identifiziert werden. Unter anderem können etwa folgende Stichworte als „Ideegeber“ für die Entwicklungen auf dem neuen Gebiet Selbststeuerung logistischer Prozesse identifiziert werden:

- Die Kybernetik und Systemtheorie,
- die Synergetik,
- das Konzept der Autopoiese,
- Komplexe Systeme in der Physik
- und die Chaostheorie.

An dieser Stelle kann nicht auf alle Bereiche erschöpfend eingegangen werden und doch sollen einige, im Kontext der Selbststeuerung logistischer Prozesse potenziell interessante, „Ideegeber“ kurz erläutert werden.

Unter den Stichworten Kybernetik und Systemtheorie sei hier gemäß der Definition der Deutschen Gesellschaft für Kybernetik von 1999 folgendes verstanden (Gfk 2002):

Der Begriff Kybernetik wird nicht auf die Theorie und Technik der Regelung beschränkt verstanden, sondern als Beschäftigung mit der Übertragung und Verarbeitung von Information unter Verwendung analytischer, modellierender, messender und kalkülisierender Methoden zum Zwecke von Prognosen und Objektivationen.

Dabei kann Verarbeitung und raumzeitliche Übertragung von Information (A) in und zwischen Subjekten (Anthropokybernetik) oder auf der (B) biologischen Ebene (Biokybernetik) oder auch (C) in Maschinen (Konstruktokybernetik) erfolgen, aber auch (D) als vom Seinsbereich unabhängige Struktur betrachtet werden (allgemeine Kybernetik). In allen diesen vier Bereichen führt die Analyse auf vier aufeinander aufbauende Gegenstandsstrukturen:

- Messung, Codierung und Übertragung von Information,
- Algorithmen und Systeme der Informationsverarbeitung,
- zielgerichtete Umweltlenkung (speziell: Regelung),

- Zielverfolgung im Einflussbereich anderer Subjekte (speziell: mathematische Spieltheorie).

Der umfassende Kybernetikbegriff schließt unter anderem die folgenden Disziplinen ein (Gfk 2002):

- Mathematische Informationstheorie,
- Informatik,
- Regelungstheorie,
- allgemeine Systemtheorie,
- Wirtschaftskybernetik (mathematische Wirtschaftsforschung),
- Spieltheorie,
- Organisationskybernetik,
- Theorie künstlicher Intelligenz,
- Bildungstechnologie.

Im Hinblick auf die Selbststeuerung logistischer Prozesse sind es insbesondere Bereich (A) (zwischen Subjekten) und (C) (in und zwischen Maschinen) sowie alle Gegenstandsstrukturen, die die Kybernetik für diese neuen Konzepte und Verfahren interessant machen.

Die Synergetik ist nach Haken (Haken 1990; Göbel 1998) die Lehre vom Zusammenwirken, nicht nur in der belebten sondern auch in der unbelebten Materie. Dabei geht es im Wesentlichen um die Bildung wohlgeordneter Muster (Ordnung), die die Einzelteile eines Systems selbst hervorbringen und durch sinnvolles Zusammenwirken, durch Zufuhr oder Abgabe von „Energie“, selbstständig aufrecht erhalten können. Dabei ist durchaus Konkurrenz zwischen den Einzelelementen möglich. Haken bezog diese Theorie aus der Untersuchung und Beobachtung von Laserlicht.

Auch hier existieren einige Aspekte, die die Synergetik als „Ideengeber“ für die Selbststeuerung logistischer Prozesse interessant erscheinen lassen, und zwar insbesondere die Entstehung von Ordnung im System durch das Zusammenspiel der einzelnen Prozesse und Objekte.

Die Theorie der Autopoiese wurzelt in der Biologie. Im Kern der Untersuchungen von Maturana und Varela (Göbel 1998) stand dabei die Frage nach dem ursächlichen Mechanismus der andauernden Selbsterzeugung bei Lebewesen. Eine wichtige Beobachtung war für sie die kontinuierliche „Randbildung“ (Membran) bei Zellen, die einen Zellstoffwechsel überhaupt erst ermöglicht. Daraus leiteten Maturana und Varela ab, dass durch die Autopoiese die Lebewesen als autonom gekennzeichnet sind, das heißt sie sind fähig, das ihnen Eigene zu spezifizieren. Welche Strukturen sich etablieren können, hängt von der Interaktion zwischen den Lebewesen und der Umgebung ab (Göbel 1998). Erscheinen die ausgelösten Strukturänderungen nach außen hin destruktiv oder konstruktiv, so wird dieses als eine

Art Selektion durch die Umwelt beziehungsweise durch das Lebewesen interpretiert.

Bezüglich der Theorie der Autopiese erscheinen insbesondere Aspekte der Interaktion, Selektion und Strukturbildung (Rand beziehungsweise Membran) als „Ideengeber“ für die Selbststeuerung logistischer Prozesse sehr interessant.

4.5.2.2 Selbststeuerung logistischer Prozesse – Eine Definition

Ansätze, Methoden und Konzepte der Selbststeuerung logistischer Prozesse werden im Sonderforschungsbereich (vgl. Literaturverzeichnis SFB 637) untersucht und entwickelt. In diesem Rahmen wurde folgende, zunächst von der Logistik losgelöste Definition für den Begriff Selbststeuerung erarbeitet:

Definition Selbststeuerung:

Die Selbststeuerung wird als ein Bündel von Prozessen dezentraler Entscheidungsfindung in heterarchischen Strukturen verstanden. Sie setzt die Fähigkeit und Möglichkeit interagierender Systemelemente zum autonomen Treffen von zielgerichteten Entscheidungen voraus.

Ziel des Einsatzes von Selbststeuerung ist das Erreichen einer höheren Systemrobustheit sowie eine Vereinfachung der Prozesse durch die verteilte Bewältigung von Dynamik und Komplexität in Form von höherer Flexibilität und Autonomie der Entscheidungsfindung.

Dies bedeutet, dass beispielsweise in den Bereichen der Produktions- und Transportlogistik zukünftig die logistischen Objekte, als wesentliche Systemelemente, autonom Entscheidungen treffen sollen.

Hinsichtlich der Ausprägungen von Selbststeuerung sind verschiedene Szenarien möglich, danach gestaffelt, welche logistischen Objekte Entscheidungen treffen. Hinsichtlich der Art der entscheidungstreffenden, mithin potentiell „intelligenten“ logistischen Objekte, erscheint eine Übertragung der Steuerungsentscheidungen auf Güter, Maschinen, Lager und Fördermittel sinnvoll. Neben Szenarien in denen nur eine der Gruppen logistischer Objekte selbststeuernd Entscheidungen treffen, sind auch beliebige Kombinationen denkbar, je nachdem, ob die Objekte der entsprechenden Gruppe tendenziell eher fremd- (durch den Menschen) oder selbstgesteuert agieren.

Den einzelnen Objektgruppen lassen sich logistische Zielgrößen zuordnen. So lässt sich das Ziel einer hohen Auslastung am ehesten einer Maschine zuordnen, während das Ziel der Termineinhaltung am sinnvollsten

mit dem Gut verknüpft ist. Konkrete Zielerreichungen ergeben sich jedoch erst durch die Interaktion vieler logistischer Objekte. Durch Verhandlungen der Objekte in einer Entscheidungssituation muss hier ein Ausgleich zwischen den oft konfliktären Zielen herbeigeführt werden. Hierbei ist von einem erhöhten Koordinations- und Kommunikationsbedarf gegenüber hierarchischen Formen der Entscheidungsfindung auszugehen. Je mehr Objekte und Objekttypen an einer derartigen Kommunikation beteiligt sind und autonom Entscheidungen treffen können, umso stärker wiegt dieser Punkt.

Während im intraorganisationalen Fall (hier Produktionslogistik) potenziell noch von kooperativer Entscheidungsfindung ausgegangen werden kann ist, erscheint diese Annahme spätestens beim Übergang zu interorganisationalen (hier Supply Chain Management und Transportlogistik) Verhandlungen eher unrealistisch. Verhandlungen in unternehmensübergreifenden Prozessen werden daher derzeit in der Regel unter der Prämisse der Nicht-Kooperativität betrachtet. Erschwerend kommt hinzu, dass bedingt durch die heterarchische Organisation keine übergeordnete Instanz vorhanden ist, die im Fall von Konflikten diese auflösen kann. Alle diese Punkte müssen beim Entwurf einer Selbststeuerungsstrategie sowie bei der Modellierung selbststeuernder logistischer Prozesse berücksichtigt werden.

4.5.3 Grundlagen der Softwareagenten

Ziel dieses Abschnitts ist es, einen Überblick über den Stand der Technik bezüglich der Agententechnologie beziehungsweise bei den Softwareagenten zu geben. Diesbezüglich wird ein Bogen von der Herkunft der Agententechnologie zu den Merkmalen und Anforderungen an die Gestaltung eines Softwareagenten hin zu aktuellen Anwendungen der Agententechnologie gespannt.

4.5.3.1 Herkunft der Agententechnologie

Die Agententechnologie hat verschiedenste Wurzeln. Besonders auf dem Gebiet der Informatik ist die Agententechnologie in den zirka letzten 10 Jahren ein sehr beliebtes Forschungsfeld geworden. Dabei sollte man auf diesem Gebiet unterscheiden zwischen der Verwendung der Begriffe im Bereich Software, Robotik und Automatisierungstechnik. Auch die Robotik und Automatisierungstechnik verwendet den Begriff Agenten und spielt dabei häufig auf eine oder mehrere physische Entitäten (technische Multiagentensysteme beziehungsweise Hardwareagenten) an. Beide Bereiche verwenden den Agentenbegriff aber auch unter dem Verständnis „in-

telligenter Programme“. Da es auf diesem Gebiet sehr vielschichtige Einflüsse aus verschiedenen Wissenschaften und Nachbargebieten der Informatik gibt, ist derzeit auch immer noch keine allgemein gültige Definition verfügbar. Besonderen Einfluss auf die Entwicklungen auf dem Gebiet der Agententechnologie und Softwareagenten haben, wie die Abb. 4.5-1 zeigt, die Künstliche Intelligenz, die Psychologie sowie die Kognitionswissenschaft, die verteilte Künstliche Intelligenz, Netzwerke und Kommunikation und schließlich die Entscheidungstheorie.

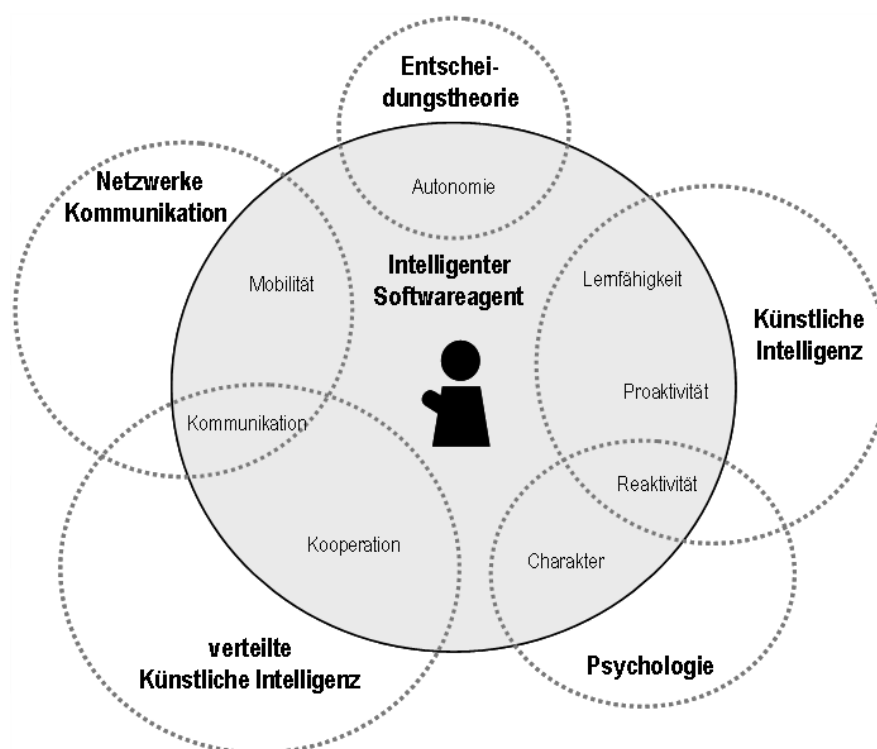


Abb. 4.5-1 Einflussgebiete auf die Softwareagenten (Brenner et. al. 1997)

Aus all diesen Blickwinkeln heraus werden Agentensysteme diskutiert, konzipiert und entwickelt, mit den unterschiedlichsten semi-formalen (z.B. textuell) bis hin zu formalen Methoden (z.B. mehrwertige Logiken), die unter Umständen maschinen- beziehungsweise rechnerverarbeitbar sind (z.B. Prolog-Programmierung, formale Ontologiesprachen). Die Motivation der Erforschung und Entwicklungen auf diesem Gebiet ergeben sich aus neuen Erwartungen an Softwareprogramme, die über kooperierende Programme, asynchrone und dezentrale Steuerung bis hin zu Assistenzsys-

temen oder komfortableren Mensch-Maschine-Schnittstellen reichen. Wenn im weiteren Verlauf der Begriff Agent oder auch intelligenter Agent verwendet wird, so wird darunter immer ein Softwareprogramm verstanden.

4.5.3.2 Definition und Merkmale von Softwareagenten

Obwohl, wie bereits erwähnt, noch keine allgemein akzeptierte Definition darüber existiert, was ein Agent (Softwareagent) ist, so kann doch quasi eine Art Minimaldefinition aufgestellt werden, die einen Eindruck darüber vermittelt, womit man es zu tun hat. Diese könnte gemäß (Burkhard 2000) etwa wie folgt lauten:

Definition Softwareagent:

Ein Softwareagent ist ein längerfristig arbeitendes Programm, dessen Arbeit als eigenständiges Erledigen von Aufträgen oder Verfolgen von Zielen in Interaktion mit einer Umwelt beschrieben werden kann.

Dementsprechend kann man gemäß (Burkhard 2000) für Multiagentensysteme etwa folgende Definition folgern:

Definition Multi-Agentensysteme (Mehragentensysteme):

Allgemein bezeichnet man eine Menge von interagierenden Agenten als ein Multi-Agentensystem. Bei Softwareagenten beruht die Interaktion insbesondere auf dem Austausch von Nachrichten, während in der Robotik auch die gemeinsame physische Arbeit in Betracht kommt.

Im zweiten Teil dieser Definition von Multi-Agentensystemen, im Rahmen der Robotik oder auch der Automatisierungstechnik, wird nochmals der bereits weiter oben beschriebene Unterschied zwischen Softwareagenten und technischen beziehungsweise Hardwareagenten deutlich.

Zu den wesentlichen Merkmalen oder auch „Dimensionen“ bei der Betrachtung von Softwareagenten zählen in Anlehnung (Burkhard 2000) die folgenden, die nicht gänzlich frei von inhaltlichen Überschneidungen sind. Hierfür wird bereits eine für die Selbststeuerung logistischer Prozesse relevante Auswahl getroffen und durch Anmerkungen an verschiedenen Stellen ergänzt:

Rahmenbedingungen & Umwelt:

- *Andauernde Verfügbarkeit / Aktivität:* Agenten sind über einen längeren Zeitraum hinweg ansprechbar (z.B. Planungsagenten) und neh-

men „Aufträge“ von Nutzern oder anderen Agenten entgegen. Sie können eigenständig aktiv werden (z.B. Aktualisierung beim Planungsagenten).

- *Interaktion mit einer Umwelt*: Agenten nehmen Informationen aus ihrer Umwelt auf (z.B. Aufträge, Erfassen von Situationen, Kontrolle ihrer Aktionen). Sie agieren in ihrer Umwelt, um sie (auftragsgemäß) zu beeinflussen.
- *Situiertheit*: Auch hier geht es um die Einbettung von Agenten in eine Umwelt. Es wird der Aspekt betont, dass komplexes Agentenverhalten auch als Resultat von direkten Reaktionen auf Umwelteinflüsse erzeugbar ist („emergentes“ Verhalten), was zu einfacheren Architekturen führen kann. Auch gerade dieser Punkt ist im Hinblick auf die Selbststeuerung logistischer Prozesse besonders interessant, aber „emergentes“ Verhalten ist nicht zwingend.

Agenteneigenschaften & -verhalten:

- *Eigenständigkeit (Autonomie) im Handeln*: Agenten unterliegen keiner (unmittelbaren) Steuerung und Kontrolle durch den Nutzer. Sie handeln eigenständig, aber „im Sinne ihrer Auftraggeber“. Die Auswahl beziehungsweise Planung ihrer Handlungen kann (aber muss nicht) nach sehr komplexen Methoden erfolgen. Dieser Aspekt findet sich bereits in ähnlicher Form in der Definition zur Selbststeuerung wieder.
- *Reaktivität*: Im engeren Sinne wird darunter das unmittelbare Reagieren auf Umweltereignisse (z.B. Stimulus-Response-Modell) verstanden. Im Allgemeinen betrifft es die Interaktion mit der Umwelt insgesamt. Besonders die allgemeine Interpretation ist im Hinblick auf die Selbststeuerung logistischer Prozesse besonders interessant.
- *Zielgerichtetheit*: Agenten verfolgen Ziele (hier logistische Ziele, z.B. Termintreue) beziehungsweise Aufträge, die angepasstes Handeln über lange Zeiträume hinweg erfordern. Gerade diese Aspekte werden im Rahmen der Definition der Selbststeuerung explizit adressiert.
- *Pro-Aktivität*: Der Begriff ist verwandt mit Zielgerichtetheit (und wird teilweise synonym benutzt). Eine spezielle Betonung liegt dabei auf der „Eigeninitiative“ des Agenten.
- *Deliberatives Verhalten*: „Deliberation“ bezeichnet die explizit modellierte Auswahl von Zielen beziehungsweise Absichten. Auch gerade dieser Punkt ist im Hinblick auf die Selbststeuerung logistischer Prozesse besonders interessant.

- *Rationalität*: Agenten treffen sinnvoll erscheinende Entscheidungen in angemessener Zeit auch unter dem Aspekt beschränkt verfügbarer Ressourcen.
- *Lernfähigkeit*: Agenten können ihr Verhalten an die Umwelt anpassen, indem sie zum Beispiel Fähigkeiten oder Entscheidungsprozesse geeignet variieren. Dieser Punkt ist im Hinblick auf die Selbststeuerung logistischer Prozesse sehr interessant, aber nicht zwingend.
- *Kooperation*: Agenten arbeiten zusammen mit Menschen und anderen Agenten. Dieser Punkt ist im Hinblick auf die Selbststeuerung logistischer Prozesse besonders wichtig.

Aspekte der technischen Implementierung:

- *Mobilität*: Agenten können eigenständig auf andere Plattformen (Rechner) migrieren. Dabei wird das Agenten-Programm mit seinen aktuellen Daten übertragen und setzt seine Arbeit dort fort mit der Möglichkeit des Zugriffs auf lokale Ressourcen. Dieser Punkt ist im Hinblick auf eine technische Realisierung der Selbststeuerung logistischer Prozesse besonders mit Softwareagenten sehr interessant.

Im Rahmen der Entwicklung von Softwareagenten wird häufig auch von Agentenorientierten Techniken (Krallmann u. Albayrak 2002; Burkhard 2000) oder der Agentenorientierten Softwareentwicklung (Weiß u. Jakob 2005) gesprochen, die mit speziellen Methoden, Programmier-techniken und Werkzeugen, den besonderen Merkmalen und Anforderungen der Softwareagenten Rechnung tragen soll. Unter den bekannteren Methoden sind hier sicherlich die Gaia von Wooldrigde, Jennings und Kinny (Wooldrigde et al. 2000) sowie die MASSIVE nach Lind (Lind 2001) zu nennen, die teilweise die Identifikation der Softwareagenten im System, sowie deren Konzeption und Modellierung unterstützen sollen. Zu den bekannten Tools und Werkzeugen gehören beispielsweise das nichtkommerzielle FIPA-OS Toolkit (FIPA 2003), welches mit seiner Systemarchitektur („informeller“ Technologie-Standard) maßgeblich ist, sowie die ebenfalls nicht kommerzielle JADE Plattform (Bellifemine et al. 2003), die sehr verbreitet ist. Zu den kommerziellen Werkzeugen zählt beispielsweise das AGENTBUILDER Toolkit (Acronymics 2004).

Ein weiterer interessanter Bereich zur Abbildung und Untersuchung von Selbststeuerungsstrategien ist die Multiagentensimulation. Mit der Multiagentensimulation wird versucht, die Defizite der herkömmlichen Simulationstechniken zu heilen, indem sie Agenten als aktive Bestandteile eines Modells einsetzen, die sich nicht nur selbst verändern, sondern auch auf die Umwelt einwirken (Klügl 2001). Dabei geschehen Antworten nicht nur als passive Antwort auf Umwelteinflüsse, sondern in Relation zur Umwelt,

die durch einen beschränkten Wahrnehmungs- und Aktionsradius (lokales Verhalten) hervorgebracht werden. Hierzu verfügt jeder Agent über ein nicht-triviales Verhaltensrepertoire, welches formal beschreibbar ist (Klügl 2001). Zur Gestaltung des Verhaltensrepertoires werden verschiedene Kategorien von Wissen benötigt, welches beispielsweise in Regeln formalisiert und durch Lernen manipuliert (Klügl 2001) werden kann. Zu den bisherigen Anwendungen von Multiagentensimulationen (Klügl 2001) zählen unter anderem Computerspiele, Zelluläre Automaten, mikroskopische Verkehrssimulationen, Insektenmodelle, Krankenhausszenarien oder auch ein Hochregalszenario aus dem Bereich der Lagerlogistik, welches mit dem Multiagentensimulationswerkzeug SeSAM (Shell für simulierte Agentensysteme) ab der Version 1.8 (SeSAM 2004) erstellt wurde. In SeSAM werden die einzelnen Agenten überwiegend mit einer grafischen Notation modelliert, die dann sofort in der Shell ausführbar sind. Darüber hinaus stehen Plug-Ins zur Verfügung, die beispielsweise den Import von Ontologien (formalisierte Wissensbasis) oder die Ausführung von Kommunikation mit Protokollen (konform zum FIPA-Standard) zwischen den modellierten Agenten erlaubt (SeSAM 2004). Es steht auch ein Vorgehensmodell (Oechslein 2004) inklusive einer Spezifikations- und Implementierungssprache für Multiagentensysteme zur Verfügung, die an SeSAM anknüpft. Für überschaubar große Szenarien aus den Bereichen Produktions- und Transportlogistik sowie dem Supply Chain Management ist das Konzept der Multiagentensimulation zur Entwicklung und zum Testen von Selbststeuerungsstrategien durchaus interessant. Auf dem Gebiet der Multiagentensimulation gehören Swarm (benutzt Swarm Intelligence), PECS oder AgentSheets (Klügl 2001) zu den bekanntesten Werkzeugen, mit jedoch jeweils sehr unterschiedlichem Focus und Hintergrund (Anwendungsgebieten).

Grundsätzlich durchläuft ein einzelner Agenten etwa einen **3-Phasen-Zyklus** (Burkhard 2000) der in die Informationsaufnahme, die Wissensverarbeitung und Entscheidung sowie die Aktionsausführung untergliedert ist. Dabei können die einzelnen Phasen in Anlehnung an (Burkhard 2000) wie folgt charakterisiert werden:

- *Informationsaufnahme*: Der Agent nimmt Informationen aus der Umwelt auf. Er nimmt Nachrichten und Aufträge entgegen. Er beobachtet die Wirkungen seiner Handlungen.
- *Wissensverarbeitung und Entscheidung*: Der Agent aktualisiert sein Wissen unter Zuhilfenahme der eingegangenen Informationen. Er analysiert und bewertet die aktuelle Situation, die neuen Aufträge und den Fortschritt bereits begonnener Handlungen. Er trifft Entscheidungen über seine unmittelbaren und zukünftigen Aktionen.

- *Aktionsausführung*: Die anstehenden Aktionen (z.B. Versenden von Nachrichten, Berechnungen, Planungsschritte) werden ausgeführt.

Auf der **Einzel-Agenten-Ebene** können etwa folgende Agententypen unterschieden werden, die in der Regel auch verschiedene softwaretechnische Architekturen aufweisen (Burkhard 2000):

- Reaktive Agenten (Stimulus-Response-Architektur),
- zielorientierte beziehungsweise deliberative Agenten,
- BDI-Agenten (**B**elief-**D**esire-**I**ntention Architektur),
- verschiedene Schichtenarchitekturen,
- situierte Agenten (Subsumtions-Architektur),
- und hybride Agenten beziehungsweise Architekturen als Mixtur aus den oben genannten (z.B. InteRRaP nach Müller 1996).

Auf der **Mehr-** beziehungsweise **Multi-Agenten-Ebene** können auch verschiedene Ansätze und Konzepte unterschieden werden. Diese, im weiteren Verlauf vorgestellten Verfahren und Konzepte, sind momentan die gängigsten Standardverfahren.

Zum einen gibt es das verteilte Problemlösen, bei dem in der Regel mindesten die folgenden Schritte durchlaufen werden (Burkhard 2000):

- Zerlegung des Problems in separat lösbare Teilprobleme,
- Zuordnung der Teilprobleme zu einzelnen Agenten,
- Lösung der Teilproblem durch die Agenten,
- Zusammenführen der Resultate.

Die beiden ersten Schritte werden alternativ im kooperativen Problemlösen, einem neueren Ansatz auf der Basis des verteilten Problemlösens, zusammengefasst. Hierbei werden zuerst die Teilaufgaben erstellt und verteilt, unter Berücksichtigung der jeweiligen Fähigkeiten der Agenten. Anschließend werden die Teilaufgaben bearbeitet, um die Teilergebnisse in einem letzten Schritt einer Synthese zuzuführen. Sollte das Gesamtproblem dann aufgrund von nicht kompatiblen Teillösungen insgesamt nicht gelöst worden sein, wird dieser Ablauf erneut gestartet (Burkhard 2000).

Weiterhin sind das Kontrakt-Netz-Protokoll (Contract Net Protocol nach Smith 1980) sowie die Black-Board-Systeme als Konzepte und Verfahren auf der Multi-Agenten-Ebene zu nennen. Bei dem **Kontrakt-Netz-Protokoll** werden in der Regel vier Schritte durchlaufen (Burkhard 2000):

- Ausschreibung (task announcement) durch einen *Manager-Agenten*,
- Bewerbung (bids) durch die *Arbeiter-Agenten*,
- Verhandlung (negotiation) zwischen *Manager* und *Arbeiter*,
- Zuweisung (contract) an die *Arbeiter*.

Besondere Eigenschaften bei dem Kontrakt-Netz-Protokoll sind darin zu sehen, dass der Verbund aus Agenten keine zentralen Module besitzt. Kein Agent hat Kenntnisse über das Wissen und die Fähigkeiten, die im Verbund vorhanden sind, auch nicht über das möglicherweise zu erwartende Verhalten im Verbund. Jeder Agent im Verbund eines Kontrakt-Netztes bearbeitet die durch ihn übernommenen Aufträge völlig autonom und ohne Eingriff von außen (Stickel et al. 1997).

Im Hinblick auf die **Black-Board-Systeme** ist etwa folgender Bearbeitungsablauf zu erkennen:

- Die Aufgaben beziehungsweise Aufträge werden auf ein Black-Board, quasi wie auf eine *Tafel* (aus-)geschrieben.
- Diese werden von den betroffenen Agenten, das heißt diejenigen mit den relevanten Fähigkeiten, gelesen und bearbeitet.
- Die erarbeiteten Lösungen für die Aufträge werden auf das Black-Board zurückgeschrieben
- und von anderen interessierten Agenten (mit relevanten Fähigkeiten) gelesen und weiterbearbeitet.

Bei einem Black-Board-System nutzen die Agenten einen gemeinsamen Datenbereich als zentrale Struktur, der in der Regel in gemeinsame Regionen beziehungsweise Niveaus mit definierten Rechten (Lese- / Schreibzugriffe) unterteilt wird (Burkhard 2000). Insgesamt kommt dem Black-Board neben dem reinen Austausch von Daten die Synchronisation der Agenten als zentrale Aufgabe zu (Burkhard 2000).

4.5.3.3 Anwendungen von Softwareagenten

Die Anwendungen und Anwendungsmöglichkeiten von Softwareagenten sind bereits sehr vielfältig. Wie bereits unter den Stichpunkten Agentenorientierte Techniken oder Agentenorientierte Softwareentwicklung skizziert wurde, werden Softwareagenten auf den zwei Ebenen Einzelagent und Multi-Agentensystem von den Entwicklern gerne als eine Art „natürliche Metapher“ (Jennings u. Wooldridge 1998; Wooldridge u. Jennings 1999) für die Programmentwicklung (Konzeptentwicklung, Entwurf und Implementierung) von komplexen, verteilten Applikationen gesehen.

Die Anwendungen im industriellen Bereich reichen von der Prozesssteuerung in der Stromversorgung (z.B. ARCHON) (Jennings u. Wooldridge 1998), verschiedensten Nutzungen im Bereich der industriellen Fertigung (z.B. YAMS, Yet Another Manufacturing System) hin zu DASCh (Dynamical Analysis of Supply Chains) von Parunak (Parunak 2000; Jennings u. Wooldridge 98). Eine andere Supply Chain Management Anwendung ist das Integrated Supply Chain Management System (ISCM) von

Fox (Fox et al. 2000), welches verschiedene Agententypen definiert, unter anderem den TOVE Dispatcher Agenten und den Logistics Agenten. Weitere Anwendungen auf dem Gebiet der Industriellen Fertigung sind das bekannte Holonic Manufacturing System, wo Agenten für verschiedenste Aufgaben vom Scheduling (Disposition / Planung) (Gou et al. 1998; Marik et al. 2002) bis hin zur Steuerung und Koordination mit naturanalogen Methoden wie dem Konzept der „Stigmergy“ (Biologie; Auswirkungen auf das Verhalten durch anhaltende Umwelteffekte als Ergebnis vorherigen Verhaltens) (Valckenaers et al. 2001) oder sogar zur Real-Time Steuerung (Brennan et al. 2001) eingesetzt werden. Dabei wird der Aspekt der Real-Time oder Online-Steuerung im Holonic Manufacturing bis auf die Maschinensteuerungsebene, im Sinne der Automatisierungstechnik, vorangetrieben, beispielsweise unter Einbeziehung von mobilen Endgeräten (z.B. Laptops) (Marik et al. 2002). Weiterhin wird versucht unabhängig von Konzepten wie dem Holonic Manufacturing einen Ordnungsrahmen für Multi-Agentensysteme (z.B. Ressourcen und Service Agenten) in der industriellen Automatisierung zu schaffen, auch mit Bezug auf die Automatisierungspyramide (Wagner 2003). In diesem Zusammenhang ist eine wichtige Unterscheidung zwischen einer Steuerung auf der PPS-Ebene (Logistik- und Materialflussebene), dieses ist überwiegend Geschäftsprozesssteuerung auf Basis von Workflows, und der Steuerung auf der Fertigungsebene beziehungsweise der maschinennahen Steuerung (z.B. SPS, Lagerbediengeräte), zu treffen. Weiterhin wurden bereits auch erste Ergebnisse zu Untersuchungen publiziert, die den Einsatz von Agentensystemen im Bereich von Flexiblen Fertigungssystemen (FFS), insbesondere direkt auf der Ebene des Fertigungsprozesses betreffen (Cetnarowicz et al. 2002). Bei der Konzeption und Entwicklung von agentenbasierter Steuerungssoftware ergeben sich besondere Probleme bezüglich deren Anwendung bei der Montage. Hier müssten bei einer Produkt- beziehungsweise Logistikobjekt-orientierten Implementierung von Agenten (z.B. Wellen, Bleche, Bauteile, Komponenten), die Agenten quasi zu einem Zusammenbau verschmelzen (Vrba 2001). Erste Ansätze zu dieser Problematik wurden bereits untersucht, die auch die Demontage mit berücksichtigen (Pavlicka 2002).

4.5.4 Konzeption und Entwicklung selbststeuernder logistischer Prozesse mit Agentensystemen

Zielsetzung dieses Abschnitts ist es den Leser, nachdem er auf die Vielzahl von Merkmalen und Anforderungen von Softwareagenten hingewiesen wurde, für die besonderen Anforderungen beim Entwurf und der Konzep-

tion von Softwareagenten für selbststeuernde logistische Prozesse zu sensibilisieren. Wie bereits unter Abschnitt 4.5.3.2 erwähnt, werden auf dem Gebiet der Agentenorientierten Techniken und der Agentenorientierten Softwareentwicklung Agenten gerne als „natürliche Metapher“ bei der Gestaltung komplexer, verteilter Softwareapplikationen verwendet. Diesbezüglich muss dabei immer der einzelne Softwareagent (Mikroebene) sowie das Mehr- beziehungsweise Multiagentensystem (Makroebene) gestaltet werden. Die Unterscheidung in Mikro- und Makroebene wird beispielsweise von Weiß (Weiß u. Jakob 2005) aber auch anderen getroffen.

4.5.4.1 Identifikation und Entwurf auf der Mikroebene

Auf der Mikroebene des Entwurfs eines Softwareagenten werden insbesondere seine Fähigkeiten (Scholz-Reiter et al. 2005) festgelegt. Hier wird entschieden welchem Objekt (z.B. Bauteil, Maschine, Ladungsträger) der Softwareagent zugeordnet wird, wie dieses Objekt zu charakterisieren ist (z.B. Bauteilontologie) und welche Ziele es hat. Diese Informationen benötigt der Softwareagent, um quasi zu wissen *wer oder was er ist* (z.B. Name, Kennnummer, Endzustand). Die Ziele (z.B. Kennzahl für Durchlaufzeit oder Wartezeiten) benötigt der Softwareagent, um zu einem späteren Zeitpunkt entscheiden zu können, wie er (z.B. Bauteil) bearbeitet oder transportiert werden muss. Weiterhin benötigt der Softwareagent eine Bewertungsfunktion, mit der er unter Umständen verschiedene Alternativen im Abgleich mit seinen Zielen berechnen und auswerten (z.B. Suchalgorithmen) kann, bevor er eine Entscheidung (z.B. entscheidungstheoretische Modelle) trifft. Die Abbildung 4.5-2 zeigt ein Sichtenkonzept, welches zur Modellierung der Selbststeuerung in der Produktionslogistik entwickelt wird (Scholz-Reiter et al. 2005). In diesem Sichtenkonzept sind das Wissen, Ziele (Bestandteil des Wissens) und die Fähigkeiten des Softwareagenten als Umsetzungskonzept der Selbststeuerung explizite Bestandteile.

Über eine entsprechende Sensorik und Aktuatorik muss der Softwareagent schließlich noch in die Lage versetzt werden, wichtige Informationen aus der Werkstatt aufzunehmen und auch wieder ab- beziehungsweise weiterzugeben. Die konkrete Implementierung des Softwareagenten hängt jedoch wiederum sehr stark von der gewählten oder durch die Agentenplattform zur Verfügung gestellten Architektur des Softwareagenten ab (z.B. BDI-Agenten, Schichtenarchitektur) (Abschn. 4.5.3.2).

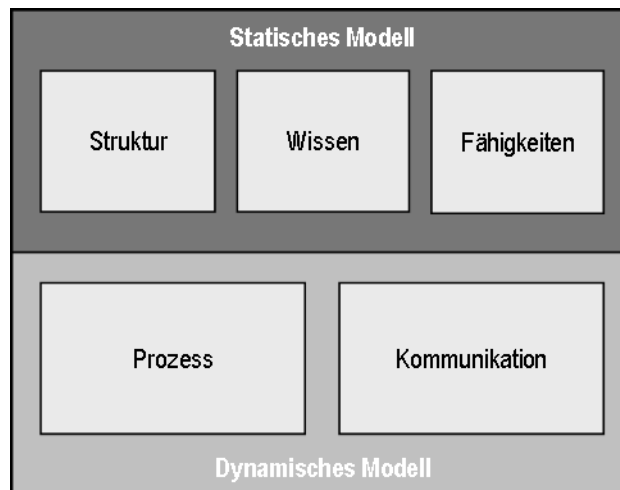


Abb. 4.5-2 Sichtenkonzept zur Modellierung der Selbststeuerung in der Produktionslogistik (Scholz-Reiter et al. 2005)

4.5.4.2 Entwurf und Konzeption auf der Makroebene

Auf der Makroebene wird entschieden, wie die Softwareagenten zusammenarbeiten. Hat zu Beispiel jeder einzelne Softwareagent zu allen anderen Kontakt (z.B. Peer to Peer) und kommuniziert er mit ihnen über den Austausch von Nachrichten. Der Aspekt der Kommunikation wird im Rahmen des dynamischen Modells des Sichtenkonzepts (Abb. 4.5-2) adressiert. Dieses setzt wiederum voraus, dass eine allen gemeinsame Ontologie (Definition Ontologie siehe Abschn. 4.5.5.3) konzipiert und implementiert wurde, die über alle kontextuellen Beziehungen zueinander Auskunft gibt. Diese Fragestellung kann auch durch die Verwendung von Verfahren wie dem Kontrakt-Netz oder einem Black-Board-Ansatz gelöst werden. Andere Ansätze, wie zum Beispiel im Holonic Manufacturing, versuchen qua Definition bestimmte Hierarchieebenen (Holone) mit bestimmen Agenten-Konstellationen (z.B. Maschinenagenten) zu schaffen. Multiagenten-Architekturen, wie beispielsweise von Huhns und Singh (Huhns und Singh 1998), orientieren sich mit der Definition verschiedener Agententypen an dem Entwurf verteilter Softwaresysteme (Abschn. 4.5.5.3). Ziel dieser Ansätze ist es, die erste Identifikation von Softwareagenten in dem zu entwerfenden System zu unterstützen und zu erleichtern. Eine weitere Fragestellung ist die der Synchronisation der verteilten Auftragsbearbeitung durch die verschiedenen Agenten. Die Bearbeitung und eine spätere Synthese von Teilaufträgen muss mit dem Austausch von Nachrichteninhalten und „logischen Sprechakten“ abgestimmt sein. Dabei

können beispielsweise zeitlich aufeinander folgende Teilaufträge durch die Softwareagenten auch nur bearbeitet werden, wenn der vorherige Schritt abgeschlossen und die relevanten Informationen per Nachricht übermittelt wurden. Abschließend muss auch noch geklärt werden, ob die Abstimmungsverfahren zwischen den einzelnen Softwareagenten prinzipiell kooperativ erfolgen sollen, oder ob hier jeder Agent für sich optimiert (opportunistische Koordination) und so unter Umständen eine gute Lösung der Gesamtaufgabe „aus den Augen verliert“. An dieser Stelle haben die Mikro- und die Makroebene den größten Berührungspunkt, da das jeweils im Softwareagent implementierte Entscheidungs- und Auswahlverfahren eben über „globale“ Kooperation oder lokale Optimierung entscheidet.

4.5.5 Selbststeuerung logistischer Prozesse mit Agentensystemen

Zielsetzung dieses Abschnitts ist es die Selbststeuerung logistischer Prozesse mit Agentensystemen auf verschiedenen Anwendungsgebieten in der Logistik zu skizzieren. Die Anwendungsgebiete sind zum einen die Produktionslogistik und die Transportlogistik sowie zum anderen das Supply Chain Management. Diesbezüglich wird auch auf besondere Anforderungen aus dem Bereich der Softwareagenten sowie potenzielle Ansatzpunkte bei bestehenden Modellen in der Produktionsplanung und -steuerung, dem Supply Chain Management und auf Automatisierungsaspekte näher eingegangen.

4.5.5.1 Selbststeuerung in der Produktionslogistik

Die logistischen Prozesse in der Produktion sind sehr komplex und hängen zudem stark von der betrachteten Fertigungsart (Einzel-, Serien-, Massenfertigung) sowie dem Fertigungsprinzip (Werkstätten-, Gruppen-, Fließprinzip) ab (Much u. Nicolai 1995). Die Produktionslogistik umfasst allgemein die Aufgaben der Planung und Steuerung der Fertigungs-, innerbetrieblichen Transport-, Umschlags- und Zwischenlagerungsprozesse. Dieses schließt alle relevanten Informationsprozesse (Sommerer 1998) mit ein, welche für die Versorgung der Produktionsprozesse mit Einsatzgütern (z.B. Roh-, Hilfs-, Betriebsstoffe) notwendig sind, sowie die Koordination der Abgabe von Halbfertig- und Fertigerzeugnissen an das Absatzlager (Jünemann u. Bayer 1998). Ziel ist insgesamt die Beschleunigung aller Flüsse, insbesondere des Materialflusses (Sommerer 1998), und die Minimierung der Aufwendungen beziehungsweise aller nicht wertschöpfenden Leistungen wie Blind- (z.B. Zwischenlager, Sicherheitspuffer) oder

Fehlleistungen (z.B. Ausschuss, Nacharbeit, Störungen) (Tomys 1995; Hummel u. Malorny 1997). Der Wirkungsbereich der Produktionslogistik wird dabei in der Regel auf die Phase vom Eintritt der Güter in den Fertigungsprozess bis zu deren Erreichen des Endlagers begrenzt (Becker u. Rosemann 1993).

Auch die Selbststeuerung, die den Steuerungsaspekt dieser Definition von Produktionslogistik aufnimmt, muss bezüglich der skizzierten Phasen, der Zielsetzung der Minimierung der Aufwendungen folgen und einen entsprechenden Beitrag zur Verbesserung der Prozesse leisten. Dabei werden unter Selbststeuerung hauptsächlich Informationsprozesse dezentraler Entscheidungsfindung in den heterarchischen Strukturen der Produktionslogistik verstanden. Diese ergeben sich aus den Fähigkeiten (Scholz-Reiter et al. 2005) und Möglichkeiten der informationellen Repräsentationen (z.B. Softwareagenten) interagierender Systemelemente (z.B. Güter, Maschinen, Fördermittel) mit dem Ziel, autonom und zielgerichtet Entscheidungen zu treffen. Diesbezüglich wird gemäß Jünemann (Jünemann u. Bayer 1998) eine starke Kopplung der Informationsflüsse an die jeweiligen Güter verfolgt, um so eine weitreichende Synchronisation der Informationsflüsse mit den Materialflüssen realisieren zu können.

In Anlehnung an Jünemann (Jünemann u. Bayer 1998) werden unter Informationsfluss die folgenden Funktionen beziehungsweise Arbeitsoperationen subsumiert, die im Rahmen der Selbststeuerung direkt an das jeweilige Objekt, insbesondere an das Gut, aber auch an die Maschine und das Fördermittel gekoppelt werden:

- Daten ein- und ausgeben beziehungsweise -lesen,
- Daten transportieren, insbesondere unter Nutzung direkter Kommunikation,
- lokale Datenverarbeitung (ordnen, aufbereiten, steuern, disponieren) und
- lokale Datenspeicherung (verwalten).

Durch die so direkt an den Objekten verankerte Möglichkeit zur zielgerichteten, dezentralen Informationsverarbeitung wird insgesamt das übergeordnete Ziel, die Realisierung einer höheren Systemrobustheit, also die Beibehaltung beziehungsweise Rückkehr in einen stabilen Systemzustand nach Einwirkung einer Klasse von Störungen, angestrebt. Die verteilte Bewältigung der inhärenten Dynamik, die den heterarchischen System- und Prozessstrukturen der Produktionslogistik innewohnt, sowie die grundlegenden Fragen nach der makroskopischen Komplexität (Gesamtverhalten) eines solchen Systems im Verhältnis zu dessen Kompliziertheit (Anzahl der Elemente) sind die Hauptfragestellungen bei der Konzeption

und Entwicklung von Selbststeuerungsstrategien für die Produktionslogistik.

Bei einer weitergehenden, detaillierteren Untersuchung der Produktionslogistik stößt man häufig auf eine funktionale und hierarchisch gegliederte Ebenenbeschreibung (Ebenen der Produktionssteuerung) (Pritschow et al. 1996) der Informations- und Materialflussebenen (Jünemann u. Bayer 1998), die bottom-up quasi von kleinen nach größeren Systemeinheiten unterschieden werden. In diesem Zusammenhang wird auch der Begriff der Automatisierungspyramide verwendet. Die Materialflussebene an sich, die derzeit meist in Antriebe und Geber (Ebene 1, z.B. an Lagerbedien- oder Bestückungsgerät), Elementsteuerung (Ebene 2, Klein-SPS oder Mikrocontroller) und Bereichssteuerung (Ebene 3, SPS oder Industrie-PC) unterteilt ist und Sensor/Aktor- oder Feldbussysteme verwendet (Jünemann u. Bayer 1998), wird nicht direkt mit der Selbststeuerung adressiert. Hier sind jedoch tiefgreifende Änderungen erkennbar und möglich, zum Beispiel durch die Miniaturisierung und Verbesserung (z.B. Funkreichweiten, Rechengeschwindigkeit) von Telematik-, Rechner- und Kommunikationstechnologien, sodass diese Gliederung zukünftig wohl nicht mehr in dieser Form bestehen wird. Hierfür werden bereits seit längerer Zeit Konzepte, wie technische Multiagentensysteme (Lüth 1998) diskutiert, die sich auf diesen Ebenen (1-3, insbesondere 3) bewegen.

Das Konzept der Selbststeuerung logistischer Objekte zielt vielmehr auf die darüber liegenden Funktionsebenen ab, die wohl zukünftig nicht mehr in dieser Form getrennt werden können, sondern immer mehr verschmelzen werden. Diese adressierten Ebenen werden derzeit wie folgt unterteilt (Jünemann u. Bayer 1998, Pritschow et al. 1996):

- Materialfluss-Steuerungsebene oder Subsystemsteuerung (Ebene 4), die dezentral sämtliche Operationen eines Teilsystems zielgerichtet steuert,
- Materialfluss-Leitebene oder Systemsteuerung (Ebene 5), in der zum Beispiel Fördermittel koordiniert werden sowie der Materialfluss verfolgt und optimiert wird,
- Logistik-Leitebene oder Darstellung und Kommunikation (Ebene 6), in der beispielsweise die Produktionsplanung, die Auftragsbearbeitung oder die Lagerverwaltung durchgeführt wird.

Diesbezüglich wird deutlich, dass durch die Konzeption und Einführung der selbststeuernden logistischen Objekte, die drei eben aufgeführten Ebenen weitestgehend durch die in einem Objekt vereinten Kompetenzen (z.B. Fördermittel, Maschine oder Gut), quasi objektorientiert gebündelt werden (The PABADIS consortium 2002). Ergibt sich auf den Ebenen 1 bis 3 ein vergleichbarer Entwicklungstrend, so würde sich zukünftig wohl nur noch

ein Gesamtsystem aus zwei bis drei Ebenen ergeben und nicht mehr wie bisher eine klassische Automatisierungspyramide aus mindestens 6 Ebenen.

Für die Selbststeuerung logistischer Prozesse ist das Fertigungsprinzip der Werkstattfertigung, deren Arbeitssysteme in der Regel nach dem Verichtungsprinzip (räumlich, organisatorische Einheiten) (Pfohl 2000) angeordnet sind, mit einer Fertigungsart von Einzel- bis Kleinserien besonders interessant. Das eher auf Großserien orientierte Fertigungsprinzip der Fließfertigung, mit ihrer zumeist starren Verkettung von Fertigungs- und Transportanlagen, lässt hingegen kaum Spielraum für die Idee der Selbststeuerung produktionslogistischer Prozesse.

Die Selbststeuerung bezieht sich im Rahmen der Werkstattfertigung insbesondere auf die an der Produktion beteiligten Güter, die sich quasi selbständig ihren Weg durch die Werkstatt bis hin zur Fertigstellung des Endprodukts suchen sollen. Dabei soll das jeweilige Gut beziehungsweise Werkstück (z.B. Rohmaterial, Halbfertig-, Zwischenerzeugnis) über die an ihm zu vollziehenden Bearbeitungsvorgänge bis hin zu einer besonders exponierten Zwischenstufe, oder gar bis hin zum Endproduktstadium, informiert sein und selbständig agieren. Dieses bedeutet, dass es zumindest einen weitgehenden Teil seiner Arbeitspläne mit sich führt und darüber urteilen kann, wie, wo und nach welchen Kriterien der jeweils folgende Arbeitsvorgang durchgeführt werden soll. Sind die jeweiligen Arbeitsvorgänge auf der mitgeführten Arbeitsplan-Wissensbasis dabei sehr unterschiedlich, so bedingt das Fertigungsprinzip der Werkstattfertigung in der Regel recht weite und unter Umständen sehr häufige, diskontinuierliche innerbetriebliche Transportvorgänge (Pfohl 2000). Sind es hingegen ähnliche Bearbeitungsvorgänge, so führt das Fertigungsprinzip der Werkstattfertigung dazu, dass ein Gut in verschiedenen Bearbeitungszuständen ein und dieselbe Werkstatt mehrfach durchlaufen kann (Pfohl 2000), welches dann zu Transporten mit eher kurzen, hoch frequentierten Wegstrecken führt.

Aus der Sicht des Gesamtsystems Werkstattfertigung, mit seinen räumlich, organisatorischen Subsystemen Werkstatt und dessen Elementen Arbeitssystem, Transportsystem und Lager (inkl. Zwischenlager, Puffer), entsteht der selbststeuernde produktionslogistische Prozess im Zusammenhang mit den jeweiligen Gütern (inkl. Bearbeitungszustände) im Rahmen der Auftragsabwicklung. Dabei dient der Auftragseingang quasi als Trigger. Aus der Auftragsabwicklung ergeben sich neben einer kundenorientierten Produktkonfiguration (Zusammensetzung beziehungsweise Stückliste des Endprodukts) über verschiedene Schritte, beispielsweise der derzeitigen Arbeitsplanung und der weitergehenden Programmplanung, terminierte Bearbeitungsreihenfolgen, die mit der jeweils anzutreffenden

Situation in der Werkstatt in der Regel nichts mehr zu tun haben. Daher ist es eine übergeordnete Zielsetzung bei der Selbststeuerung produktionslogistischer Prozesse, dass die jeweiligen Güter (inkl. Bearbeitungszustände) autonom und auftragsbezogen (z.B. Berücksichtigung von Start- und Fertigstellungsterminen, Bearbeitungsqualitäten (z.B. Oberflächengüte)), sowie unter Einbeziehung von aktuellen Zustandsinformationen über die Arbeits- und Transportsysteme, ihre Arbeitspläne adaptiv terminieren. Sie können dann als Reaktion unter Umständen Bearbeitungsreihenfolgen, sofern technologisch veränderbar, selbst wählen und abändern. Dieser Wahlbeziehungsweise Abstimmungsprozess wird durch die Fähigkeiten (Scholz-Reiter et al. 2005) der Güter sowie der Arbeits- und Transportsysteme zur direkten Kommunikation unterstützt. Dieses eröffnet ihnen die Möglichkeit sich über die Anzahl der wartenden Güter vor einem Arbeitssystem oder über einen Transport zu einer bestimmten Bearbeitungsstation auszutauschen. In diesem Zusammenhang sollen sowohl das Gut als auch das Arbeitssystem die Option zur Annahme oder Ablehnung der Bearbeitung haben. Dieses kann beispielsweise unter der Abwägung von maximaler Auslastung und minimalen Rüstzeiten und -kosten (nicht wertschöpfende Stützleistung) beim Arbeitssystem (Tomys 1995) erfolgen. Weiterhin kann eine Realisierung minimaler Durchlaufzeiten durch Umgehung von Sicherheitspuffern, inklusive An- und Abtransport (nicht wertschöpfende Blindleistung) (Tomys 1995), sowie ein Ausschließen von Ausschuss und Nacharbeit (wertmindernde Fehlleistung) (Tomys 1995) auf Seiten des Gutes angestrebt werden. Für jeden Bearbeitungsvorgang soll das Gut deshalb in der Lage sein, entsprechend des nächsten anzusteuern Arbeitssystems, sei es innerhalb derselben Werkstatt (i.S. von Subsystem) oder in einer anderen, ein relevantes Transportmittel (z.B. Gabelstapler) gezielt via direkter Kommunikation zu allokalieren. Weitere relevante produktionslogistische Prozesse im Rahmen der Selbststeuerung können das Kommissionieren, die Materialbereitstellung inklusive Behälterallokation, das Testen und gegebenenfalls die Selbstausschleusung im Rahmen einer Qualitätsprüfung oder das Lackieren sein. Die folgende Abbildung 4.5-3 soll einen ersten Eindruck der Selbststeuerung in der Produktionslogistik auf der Basis eines einfachen Modells einer zweistufigen Werkstattfertigung vermitteln.

Wesentlich jedoch ist, dass unabhängig davon, welche Objekte in einem produktionslogistischen Szenario konkret per Selbststeuerung „intelligent“ gemacht werden, sie müssen neben der reinen Fähigkeit zur direkten Kommunikation auch mit dem notwendigen Wissen sowie der grundlegenden Fähigkeit zur Abwägung von Alternativen und einer zielgerichteten Auswahl von Handlungsalternativen ausgestattet werden. Diesbezüglich kann die gegenseitige Abwägung und Auswahl von Handlungs-

alternativen zum Beispiel durch den Einsatz von Methoden der Entscheidungs- und Spieltheorie (Fischer et al. 1998) realisiert werden. Dabei hängen je nach dem späteren Umsetzungskonzept (hier Softwareagenten) Kommunikation, Abwägung und zielgerichtete Auswahl (inkl. Handlung) fast immer untrennbar zusammen.

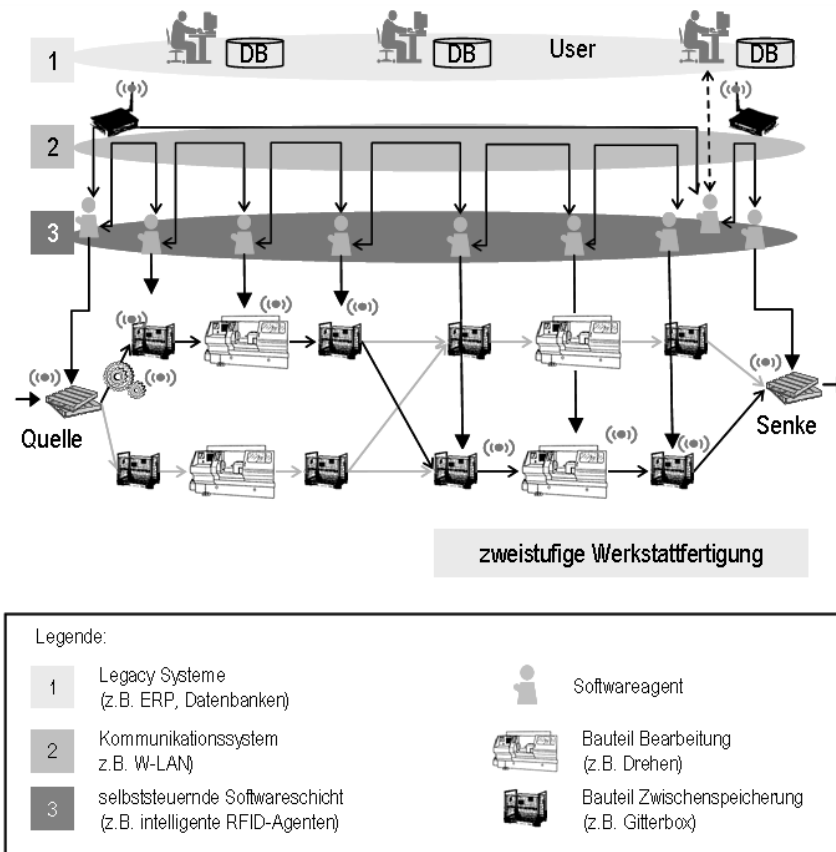


Abb. 4.5-3 Grundidee der Selbststeuerung in der Produktionslogistik (Scholz-Reiter et al. 2004)

4.5.5.2 Selbststeuerung in der Transportlogistik

Bei der Konzept- und Methodenentwicklung im Rahmen der Selbststeuerung logistischer Prozesse in der Transportlogistik wird untersucht, wie der multimodale Gütertransport – Nutzung unterschiedlicher Transportmodi im Verbund – durch neuartige Verfahren unterstützt werden kann. Diese

Verfahren sollen das selbststeuernde Routing von Stückgütern bei einem sich dynamisch verändernden Transportangebot unter unsicherem Wissen ermöglichen.

Diesbezüglich ist auch das Konzept der reaktiven Planung interessant (Scholz-Reiter 1998; Scholz-Reiter u. Scharke 2000; Scholz-Reiter u. Höhns 2001). Die reaktive Planung und in der Folge die damit verbundene reaktive Steuerung ist eine wichtige Ergänzung der kurzfristigen Planung, wenn durch unvorgesehene Störungen, wie beispielsweise LKW-Ausfall, Staus oder fehlende Ladung, die vorhandene Planung obsolet geworden ist. Die Methode der reaktiven Planung und Steuerung ist dadurch gekennzeichnet, dass parallel zu den laufenden Prozessen Informationen über die Umwelt aufgenommen werden, die in einer geeigneten Art und Weise verarbeitet werden müssen, um die Schlussfolgerungen zur Initiierung der notwendigen Änderungen sofort wieder in den laufenden Prozess einfließen zu lassen (Scholz-Reiter u. Scharke 2000; Scholz-Reiter u. Höhns 2001). Da a priori in der Regel kein Wissen darüber vorhanden ist welche Störungen an welcher Stelle auftreten und auch dann nur unvollständige Informationen vorliegen, versagt eine konventionelle, kurzfristige Umplanung und Umsteuerung am laufenden Prozess, da sich die Umweltinformationen im Laufe der Planung wieder verändern und nicht berücksichtigt werden können (Scholz-Reiter u. Scharke 2000; Scholz-Reiter u. Höhns 2001). Die Aufnahme und Verarbeitung der Informationen sowie die Ausgabe der Schlussfolgerungen setzt also eine ständige Interaktionen mit der Umwelt voraus, die beispielsweise den Einsatz von Multi-Agentensystemen mit den weiter oben skizzierten Merkmalen (Abschn. 4.5.3.2) nahe legt.

Im Zusammenhang mit der Selbststeuerung logistischer Prozesse in der Transportlogistik und hier insbesondere in den multimodalen Transportnetzwerken, ist die Analogie zu dem Transport von Datenpaketen in Kommunikationsnetzen sehr interessant. Diesbezüglich sind Verfahren aus der Datenkommunikation unter Umständen dazu geeignet, um auch auf den physischen Transport angewendet werden zu können (Peters et al. 2005). Bei Analogieschlüssen ist dabei jedoch immer zu berücksichtigen, dass sicherlich auch noch grundlegende Anpassungen und Veränderungen vorgenommen werden müssen. Festzustellen bleibt jedoch, dass Selbststeuernde logistische Prozesse in der Transportlogistik den Prozessen, die in Kommunikationsnetzen zum Einsatz kommen, ähneln. Für die Transportlogistik interessante Verfahren sind diejenigen, die beim Routing und bei der Störungsbehandlung im Bereich des Internets zum Einsatz kommen. Dieses betrifft eine Vielzahl von Algorithmen für die Wegewahl (z. B. Active Networks), sowie die Dienstgüteunterstützung, das heißt die Einhaltung von vorgegebenen Dienstgütekriterien (Quality of Service),

wie beispielsweise Verlustraten oder Verzögerungszeiten (Peters et al. 2005).

Auf der Basis von Softwareagenten könnten Systeme für das reaktive Routing von Stückgütern mit direkter Orientierung auf das Transportgut, sowie unter Nutzung unterschiedlicher Transportmittel und der Einbindung verschiedener Transportunternehmen realisiert werden. Kunden und Umschlagpunkte (Hubs) sind im Rahmen des Transportsnetzwerks sowohl Senken als auch Quellen für die Transportgüter (z.B. Pakete). Diese Knoten können nun variabel mit den verschiedensten Transportmodi (z.B. Zug, LKW, Flugzeug) bedient werden. Die Transportmodi bilden im Hinblick auf die Netzwerk Betrachtung die Kanten zwischen den Knoten. In einem transportlogistischen Szenario werden dann sowohl die Knoten als auch die Güter (z.B. ein Paket als atomisierte Ladung) und schließlich die jeweiligen Transportmodi (z.B. jeder LKW), durch einen Softwareagenten repräsentiert beziehungsweise vertreten. Jeder Softwareagent muss mit der Fähigkeit zur *Kommunikation* und der Aufnahme von Informationen aus der Umwelt ausgestattet sein (z.B. *Situiertheit*). Darüber hinaus muss er eigenes Zielsystem (*Zielgerichtetheit*) besitzen, welches er zum ständigen Abgleich mit den Umweltinformationen benutzt. Dabei könnte es beispielsweise für ein Transportgut interessant sein, bei Transportverzögerungen mit dem LKW an einem der nächsten Hubs die Entladung zu verlangen, um gegebenenfalls auf einen anderen LKW oder Transportmodus zu wechseln, der ihm zu diesem Zeitpunkt eine pünktliche Ablieferung beim Kunden (Senke) verspricht. Die bislang bekannten, agentenbasierten Konzepte und Verfahren im Bereich der Transportlogistik beziehen sich in der Regel nur auf einen Transportmodus (z.B. LKW) und berücksichtigen dann auch nur spezielle Funktionalitäten (z.B. Versendung, Disposition) (Bürcker et al. 1999; Funk et al. 1998; Lind u. Fischer 1998).

4.5.5.3 Selbststeuerung im Supply Chain Management

Im Rahmen eines Forschungsprojekts (vgl. Literaturverzeichnis SCHO 540/8-1), wurde die wissensbasierte, unternehmensübergreifende, reaktive Auftragskoordination in der Supply Chain mit Agentensystemen untersucht. Die reaktive Planung und Steuerung ist dabei quasi der *einfachste Fall* einer Selbststeuerungsstrategie, die im Prinzip nach einem Stimulus-Response-Ansatz mit *einfachen, reaktiven Agenten* ausgeführt werden kann (Scholz-Reiter u. Höhns 2001). Die folgende Abbildung 4.5-4 ordnet die Reaktive Planung und Steuerung in den Kontext der langfristigen bis kurzfristigen Planung ein (Scholz-Reiter u. Höhns 2001).

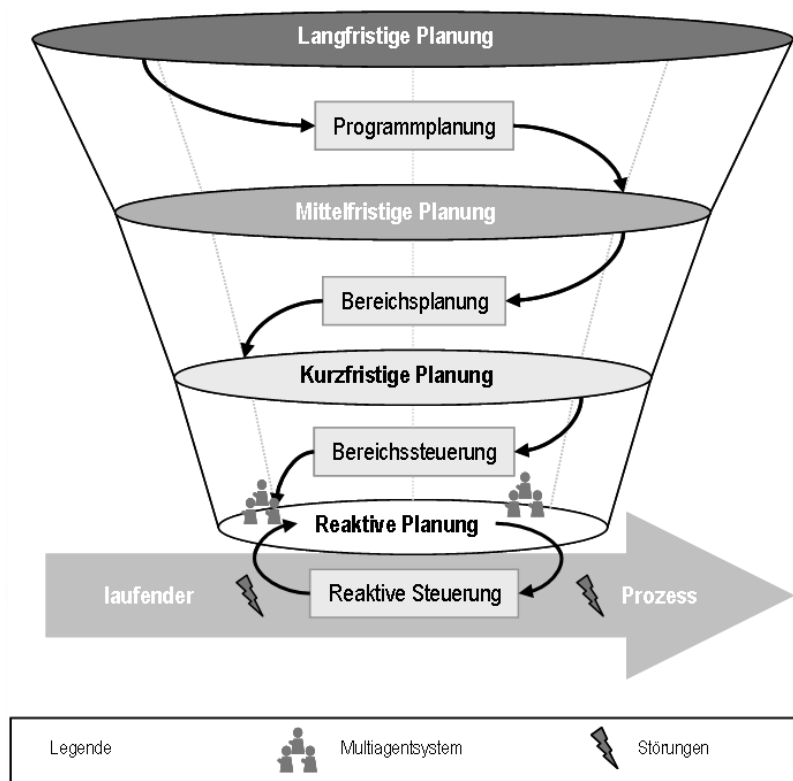


Abb. 4.5-4 Reaktive Planung und Steuerung (Scholz-Reiter u. Höhns 2001)

Weiterhin stellt sich insbesondere aus der Sicht der Anwendungsdomäne im Hinblick auf die Formalisierung des benötigten Domänenwissens die Frage, auf welche verbreiteten und akzeptierten theoretischen Modelle (z.B. PPS-Modelle) sowie Industriestandards zurückgegriffen werden kann. Bezüglich der Rahmenbedingungen für das Supply Chain Management, erscheint es hierbei zweckmäßig, den Quasistandard des SCOR-Modells 7.0 (Supply Chain Operations Reference Model) (Supply Chain Council 2005), als eine Ausgangsbasis für die Domänen- und Aufgaben-Ontologie zu wählen. Es dient im Bereich Planung und Steuerung unternehmensübergreifender Netzwerke zur Beschreibung, Bewertung und Evaluation verschiedener Konfigurationen von Lieferketten. Dabei wird in einem hierarchischen Top-Down-Modellierungsansatz ausgehend von fünf zentralen Prozessen (Beschaffen, Produzieren etc.) mit steigendem Detaillierungsgrad der Versuch unternommen, über standardisierte Prozesselemente (Planen, Ausführen, Unterstützen) bis auf die Task- und Aktivitätenebene zu gelangen, um explizit ein einheitliches Verständnis für das

Supply Chain Management zu entwickeln. Gerade dieser Aspekt lässt das SCOR-Modell im Sinne der Modellierung einer Ontologie besonders interessant erscheinen. Das SCOR-Modell 7.0 weist jedoch auf der Task-beziehungswise Aktivitätenebene immer noch erhebliche Lücken im Hinblick auf eine softwaretechnische Implementierung, beispielsweise mit Softwareagenten, auf (Supply Chain Council 2005). Diese wird zwar in gewisser Weise im SCOR-Modell angenommen, aber nicht direkt adressiert (Supply Chain Council 2005). An dieser Stelle ist also eine sinnvolle Ergänzung durch ein anderes Modell notwendig. Diesbezüglich erscheint beispielsweise eine Verknüpfung mit der im Aachener PPS-Modell – Referenzmodell einer integrierten PPS – beschriebenen Querschnittsaufgabe der Auftragskoordination (Luczak et al. 1998) interessant. Diese wird hier im Zusammenhang mit dem Supply Chain Management als unternehmensübergreifend interpretiert, welches zum Zeitpunkt der Durchführung des Forschungsprojekts im Originalmodell noch nicht der Fall war. Die wesentlichen Aufgaben der Auftragskoordination würden sich hierbei durch die Abstimmung (hier: dezentral durch Agenten) der Aktivitäten aller an der Auftragsabwicklung beteiligten Unternehmen und die Synchronisation der Aufgabenerfüllung innerhalb der relevanten Planungsebenen ergeben. Als besonders hilfreich können sich hierbei die einzelnen Teilmodelle beziehungsweise Komponenten des Aachener PPS-Modells erweisen, wie das Aufgaben-, Prozess-, Funktions- und schließlich das Datenmodell, die das SCOR-Modell auf den relevanten Ebenen ergänzen sollen. Im Hinblick auf die Formalisierung einer Ontologie stehen zum Beispiel Modellierungswerkzeuge wie Protégé 3.0 (Protégé Team 2005) zur Verfügung. Ontologien stellen dabei ein gemeinsames Verständnis einer Domäne zur Verfügung und ermöglichen so die zwischenmenschliche Kommunikation, aber auch die zwischen Anwendungssystemen und ihren Benutzern (Puppe et al. 2000). Sie beschreiben somit, welche Konzepte der realen Welt in einem Softwaresystem modelliert werden sollen (Puppe et al. 2000). Grundsätzlich lassen sich unter anderem folgende Arten von Ontologien unterscheiden (Puppe et al. 2000):

- Domänenontologien,
- Methodenontologien,
- Aufgabenontologien.

Für den Anwendungsbezug, hier Auftragskoordination im Rahmen des Supply Chain Managements, sind nun insbesondere die Domänen- und die Aufgabenontologie von besonderem Interesse (Abb. 4.5-5). Die Domänenontologie definiert die Begriffe (z.B. Prozesse und Prozesselemente) und Bezeichnungen, die für einen bestimmten Anwendungsbereich von Bedeutung sind (Puppe et al. 2000). Die Aufgabenontologie hingegen de-

finiert die Begrifflichkeiten verschiedener Typen von Aufgaben (z.B. Tasks und Aktivitäten) und setzt sie zueinander in Beziehung (z.B. Ereignis A erzeugt Reaktion B) (Puppe et al. 2000).

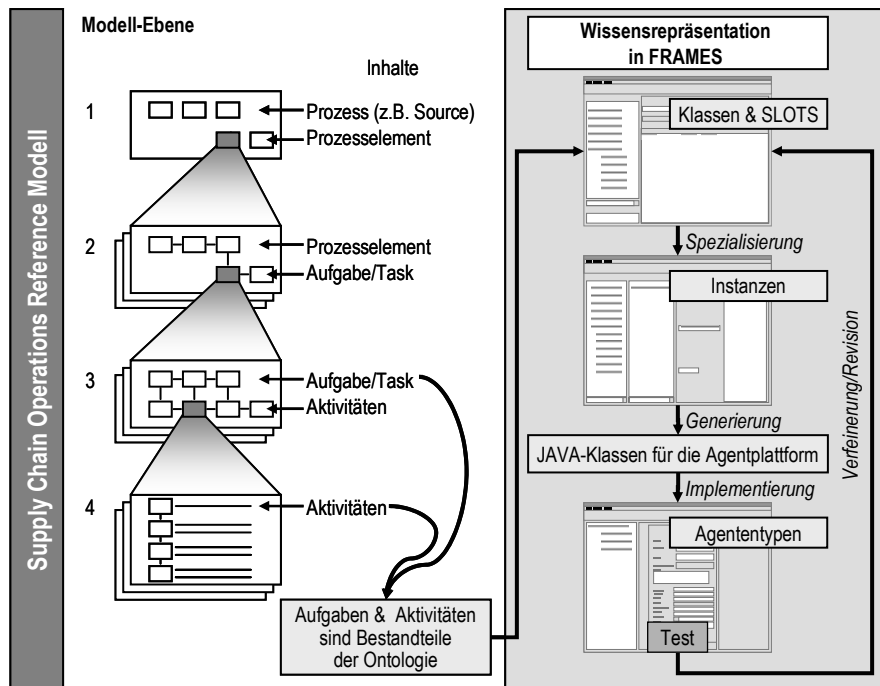


Abb. 4.5-5 Wissensmanagement für Softwareagenten im Supply Chain Management (Scholz-Reiter u. Höhns 2003)

Protégé 3.0 stellt eine abstrakte, agentensystemspezifische Ontologie zur Verfügung, mit den Konzepten Agent-ID, Agent-Activity und Prädikaten, die durch eine teilweise Automatisierung (z.B. Bean-Generator) für das freiverfügbare Agentenentwicklungstool JADE fallspezifisch angepasst werden kann. Insgesamt war es im Rahmen des Forschungsprojekts unter anderem das Ziel, zu einer ersten, groben Zuordnung der SCOR-Modell- und Aachener PPS-Modell-Bestandteile zu dieser abstrakten Ontologie zu gelangen. Hierbei ist insbesondere das Konzept der Agentenaktivität von großer Bedeutung, da unter anderem hiermit festgelegt wird, über welche Fähigkeiten der Agent im Hinblick auf die spätere verteilte Problemlösung (Teilaspekte der Auftragskoordination) verfügen soll. Über den soeben skizzierten Ansatz hinaus existieren im operativen Supply Chain Management eine Fülle von komplexen, unternehmensübergreifenden Prozessen, die jeweils einen spezifischen, hohen Koordinations- be-

ziehungsweise Abstimmungsbedarf aufweisen und die deshalb für die Selbststeuerung mit Softwareagenten potenziell interessant sind. Speziell im operativen Supply Chain Management ist es das vorrangige Ziel, die Kundenaufträge mit höchster Kundenorientierung zeit-, mengen- und termingerech im Hinblick auf die Fertigstellung des Endproduktes auszuführen. Dabei müssen aus der Perspektive des Endherstellers je nach Art und Größe des Kundenauftrags, diverse Ebenen von Zulieferern erreicht werden. Oftmals sind ihm diese jedoch nicht einmal bekannt. Zudem verfolgen alle Unternehmen einer Supply Chain gleichzeitig eine möglichst effiziente Auslastung ihrer verfügbaren Kapazitäten, sowie die Bevorratung von möglichst niedrigen Beständen. Hier deuten sich klassische Zielkonflikte an, die jedoch prinzipiell durch neue Ansätze aufgelöst werden können.

In der aktuellen Literatur werden hierzu derzeit vier Prozesse aufgezeigt, die für die Anwendung von Selbststeuerung in Betracht kommen und die zur Koordination des Kundenauftrags dienen sollen. Dieses sind zum einen Available to Promise (ATP) und Capable to Promise (CTP) (Kilger u. Schneeweiss 2000; Fleischmann u. Meyr 2001; Kuhn u. Hellingrath 2002; Alicke 2003), sowie zum anderen Plan-by-Exception und Bi-directional Change Propagation (Alicke 2003), die auf ATP und CTP aufbauen und diese ergänzen. Available to Promise (ATP) dient in erster Linie der zuverlässigen Bestimmung von Lieferterminen zu Kundenaufträgen entlang der Lieferkette (Alicke 2003; Kuhn u. Hellingrath 2002). Dabei werden in der Regel auch die disponiblen Bestände entlang der Supply Chain mit berücksichtigt und einkalkuliert (Fleischmann u. Meyr 2001), die in der Vergangenheit basierend auf den Prognosen über erwartete Aufträge beschafft oder produziert wurden. Capable to Promise (CTP) schließt sich an das ATP an und bezieht in die Bestimmung der Liefertermine zusätzlich auch noch die verfügbaren Kapazitäten auf den Produktionsressourcen der einzelnen Unternehmen im Supply Network mit ein, die an der Erbringung des Kundenauftrags beteiligt sind (Fleischmann u. Meyr 2001, Alicke 2003, Kuhn u. Hellingrath 2002). Im Hinblick auf die Koordination der Auftragsverteilung und -abwicklung der Teilkomponenten eines Kundenauftrags zwischen den Unternehmen der Supply Chain sind weiterhin die Konzepte des Plan-by-Exception sowie der Bi-directional Change Propagation (Alicke 2003) im Hinblick auf deren Umsetzung unter Verwendung von Aspekten der Selbststeuerung sehr interessant. So ist Plan-by-Exception zwar prinzipiell eher auf einen menschlichen Planer in der Logistik abgestellt, der von Routineaufgaben zugunsten von planerischen Ausnahmesituationen entlastet werden soll, welches jedoch durch einen stark gesteigerten Einsatz von „Intelligenz“ bereits auf der Ebene der logistischen Objekte (z.B. Bauteile, Komponenten, Transportmittel) poten-

ziell für Selbststeuerung geeignet ist. Im Plan-by-Exception Prozess (Alicke 2003) wird ein mehrstufiges ATP und CTP zur Bestimmung des Liefertermins mit folgenden Schritten

- Aufzeigen des Problems,
- Aufzeigen der Ursache,
- Durchspielen von Szenarien,

solange durchgeführt, bis die notwendigen disponiblen Bestände oder benötigten Fertigungskapazitäten, die zur Erfüllung des Kundenauftrags benötigt werden, allokiert sind. Dieser Prozess soll prinzipiell auch unternehmensübergreifend in der Supply Chain durchgeführt werden, welches neben geeigneten Koordinationsmechanismen, beispielsweise im Rahmen einer Selbststeuerung, auch höchste Anforderungen an die Integrationsfähigkeit der betroffenen Informations- und Kommunikationssysteme stellt, beispielsweise mit Softwareagenten. Im Zusammenhang mit dem Plan-by-Exception ist die Bi-directional Change Propagation (Alicke 2003) ein weiterer planerisch orientierter Logistikprozess, der für die Umsetzung mit Selbststeuerung sehr interessant ist. Hierbei wird beispielsweise der lokale Ausfall einer Produktionsressource untersucht, die zuvor bereits im CTP als disponible Kapazität zur Erbringung des Kundenauftrags eingeplant wurde. Im Rahmen der Bi-directional Change Propagation (Alicke 2003) wird nun eine lokale Neuberechnung und Aktualisierung des betroffenen Planabschnitts, beispielsweise durch Softwareagenten, durchgeführt, indem sie diese Ergebnisse in Richtung aller betroffenen Knoten beziehungsweise Unternehmen in der Supply Chain propagieren beziehungsweise kommunizieren. Für diesen Abschnitt passen sie die Produktions- oder Bestellmengen dahingehend an, dass beispielsweise andere Komponenten als Substitut verwendet werden können oder Liefertermine bis hin zu Auftragsreihenfolgen getauscht werden.

Anhand eines Supply Chain Szenarios aus der Maschinenbaubranche (Windt et al. 2002), welches auf der Basis eines Industrieprojekts gewonnen und modelliert wurde, sind die oben skizzierten, für die Selbststeuerung relevanten Prozesse des unternehmensübergreifenden Supply Chain Managements, wie ATP oder CTP, für das make-to-order (d.h. beim Vorliegen eines konkreten Kundenauftrags) untersucht worden (Scholz-Reiter et al. 2004; Scholz-Reiter et al. 2005). Darüber hinaus wurden wichtige Aspekte der Abbildung von Produktstrukturen auf die Supply Chain erforscht, die die Struktur und geografische Verteilung der Supply Chain maßgeblich beeinflussen (Scholz-Reiter et al. 2005). Das zugrunde liegende Supply Chain Modell verfügte über insgesamt drei Ebenen, OEM bis 2nd-Tier, mit verschiedenen Substitutionsmöglichkeiten für bestimmte

Bauteile und Komponenten, welches eine potenzielle Selbststeuerung überhaupt erst ermöglichen kann.

Die dabei gewonnene Flexibilität, wird weitestgehend durch das Multiple-Sourcing auf den verschiedenen Supply Chain Ebenen erreicht. Dieses erlaubt dann eine gute Anwendung und Beobachtung des Plan-by-Exception Ansatzes. Dieses Konzept (Scholz-Reiter et al. 2004) wurde in einem ersten Ansatz, der auch einigen der oben skizzierten Aspekte der Selbststeuerung genügt, agentenbasiert umgesetzt. So wurde auf der Basis einer von Huhns und Singh entworfenen Multiagentensystemarchitektur (Huhns u. Singh 1998) für verteilte, intelligente Informationssysteme ein Multiagentensystem konzipiert und entwickelt, welches vier Agententypen verwendet (User-, Broker-, Execution-, Mediator-Agenten), die unter anderem Maschinen, Lager und Kunden repräsentieren (Scholz-Reiter u. Höhns 2003) (Abb. 4.5-6). Die Agententypen werden nach (Huhns u. Singh 1998) etwa wie folgt definiert:

- **Broker-Agenten:** Fungieren als Gelbe Seiten zur Lokalisierung der Agenten mit den benötigten Fähigkeiten. (Diese Funktionalität wird bereits teilweise durch einen so genannten Directory Facilitator Agenten (DF), der beispielsweise in der JADE-Agentenplattform standardmäßig implementiert ist, übernommen.)
- **Ressourcenagenten:** Fungieren als Wrapper für verschiedene Ressourcentypen (z.B. Datenbanken- oder Maschinenagenten).
- **Execution-Agenten:** Ausführen, lokale Planung und Steuerung (in Form von Koordination)): Sind meist als regelbasiertes System zur Überwachung oder Steuerung (z.B. Workflowsteuerung) implementiert.
- **Mediator-Agenten:** Sind in der Regel als spezialisierte Execution-Agenten für die Suche von Ressourcen und das Handling von Kommunikation (z.B. Verhandlungen) ausgelegt.

Ein Ablauf wird durch die Produktkonfiguration eines Kunden (Configure to Promise (CoTP)) gestartet ((1) Abb. 4.5-6), wobei quasi die Produktkonfiguration als Stückliste auf die Supply Chain abgebildet wird. Für die Entscheidung über spätere Alternativen für das Plan-by-Exception und der Bi-directional Change Propagation wird derzeit eine Art Priorisierung (Alicke 2003) auf der Basis von Kundenpräferenzen verwendet. Ist ein machbarer Kundenauftrag angenommen und veranlasst ((2+3) → *confirmation oder cancellation*, Abb. 4.5-5), so wird dieser solange ausgeführt, bis ein Ausfall auf der 1st-Tier- oder 2nd-Tier-Ebene eintritt ((2 bis 5) → *Störung*, Abb. 4.5-5), um dann auf der Basis der Kundenpräferenzen ein Plan-by-Exception (mehrstufiges ATP/CTP) durchzuführen und für die verbleibende Liefer- beziehungsweise Produktionszeit einen alternativen Produzen-

ten zu ermitteln. Sind keine Präferenzen beziehungsweise Prioritäten verfügbar, so kann der OEM ein Substitut beziehungsweise passenden Produzenten auswählen, der unter logistischen Zielsetzungen, wie (Produkt-)Qualität, Zeit (Einhaltung der restlichen Lieferzeit) und Lieferservice, für ihn am geeignetsten erscheint (Scholz-Reiter et al. 2004; Scholz-Reiter et al. 2005).

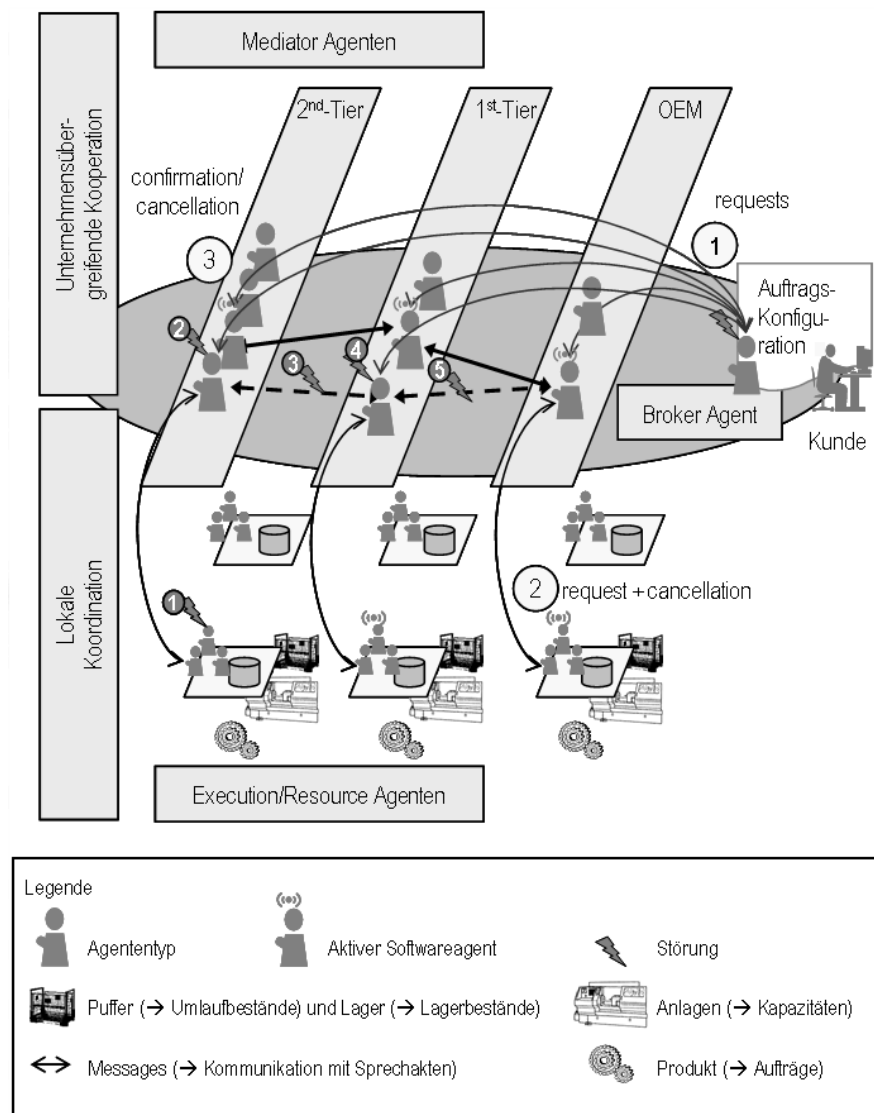


Abb. 4.5-6 Überblick über den Ablauf der Auftragskoordination

4.5.6 Zusammenfassung

Zielsetzung dieses Kapitels war es, eine Einführung und einen Überblick über die verschiedenen Einsatzmöglichkeiten von Softwareagenten im Hinblick auf die Planung und Steuerung logistischer Prozesse zu geben. Diesbezüglich wurden die Schwerpunkte der Betrachtungen auf den intra-organisationalen Bereich (Produktionslogistik), sowie auf den interorganisationalen Bereich (Transportlogistik und Supply Chain Management) gelegt. Die Selbststeuerung als Konzept und Verfahren, beschreibt in diesem Zusammenhang jedoch nicht nur eine kleine Klasse von Ideen und Methoden, sondern ist aufgrund ihrer skizzierten Herkunftsquellen ein breites Spektrum an neuen Ansätzen und Möglichkeiten zur lokalen, intelligenten Planung und Steuerung logistischer Prozesse, insbesondere aus der Ebene der logistischen Objekte heraus. Die Selbststeuerung logistischer Prozesse geht dabei von einer weiterhin sprunghaften Weiterentwicklung auf dem Gebiet der Informations- und Kommunikationstechnologie aus, sodass in zirka 10 bis 15 Jahren die logistischen Objekte (z.B. Pakete, Güter, Ladungsträger, Maschinen, Transportmittel) dazu befähigt werden können, auf der Basis kleinster, verteilter Rechnergrößen (Rechenleistung) lokal, autonom zu entscheiden und abzuwägen. Vor diesem Hintergrund befindet sich die Selbststeuerung logistischer Prozesse noch in der Konzeptphase und es ist davon auszugehen, dass es wohl keine „Lösungen von der Stange“ geben wird. Insgesamt bietet die Selbststeuerung logistischer Prozesse jedoch die Möglichkeit, ein neues Verständnis für die Logistikprozesse sowie für das logistische System und dessen Systemelemente zu entwickeln. Zielstellung ist es diesbezüglich die logistischen Prozesse robuster zu gestalten sowie die Dynamik und strukturelle Komplexität von Logistikprozess und -system insgesamt besser verstehen zu können.

4.5.7 Literatur

- Acronymics (2004) The AGENTBUILDER Toolkit Reference Manual,
<http://www.agentbuilder.com/Documentation/usermanual.html>
Download: 02.05.2005
- Alicke K (2003) Planung und Betrieb von Logistiknetzwerken, Unternehmensübergreifendes Supply Chain Management. Springer, Berlin Heidelberg New York
- Becker J, Rosemann M (1993) Logistik und CIM – Die effiziente Material- und Informationsflussgestaltung im Industrieunternehmen. Springer, Berlin Heidelberg New York
- Bellifemine F, Caire G, Poggi A, Rimassa G (2003) JADE, A White Paper.
<http://jade.tilab.com/papers/WhitePaperJADEEXP.pdf>

Download: 02.05.2005

- Brennan R W, Fletcher M, Norrie D H (2002) A Holonic Approach to Reconfiguring Real-Time Distributed Control Systems. In: Marik V, Stepankova O, Krautwurmova H, Luck M (eds) Multi-Agent Systems and Applications II. Springer, Berlin Heidelberg New York, pp 323-335
- Brenner W, Zarnekow R, Wittig H (1998) Intelligente Softwareagenten, Grundlagen und Anwendungen. Springer, Berlin Heidelberg New York
- Bürckert H-J, Funk P, Vierke G (1999) An Intercompany Dispatch Support System for Intermodal Transportation Chains. DFKI, Saarbrücken
- Burkhard H-D (2000) Software-Agenten. In: Görz G, Rollinger C-R, Schneeberger J (Hrsg) Handbuch der Künstlichen Intelligenz, 3. Aufl. Oldenbourg, München Wien, S 941-1018
- Cetnarowicz K, Kozlak J (2002) Multi-agent System for Flexible Manufacturing Systems Managemt. In: Dunin-Keplicz B, Nawarecki E (eds) From Theory to Practice in Multi-Agent Systems. Springer, Berlin Heidelberg New York, pp 73-82
- FIPA (2003) FIPA-OS
<http://www.fipa.org/resources/livesystems.html#os>, Download am 06.05.2005
- Fischer K, Ruß Ch, Vierke G (1998) Decision Theory and Coordination in Multi-agent Systems. DFKI, Saarbrücken
- Fleischmann B, Meyr H (2001) Supply Chain Planning. In: Sebastian H-J, Grünert T (Hrsg) Logistik Management, Supply Chain Management und e-Business. Teubner, Stuttgart, S 13-29
- Fox M S, Barbuceanu M, Teigen R (2000) Agent-oriented Supply Chain Management. The International Journal of Flexible Manufacturing Systems (2000)12:165-188
- Funk P, Vierke G, Bürckert H-J (1998) A Multi-Agent Perspective on Intermodal Transportation Chains. DFKI, Saarbrücken
- Gfk (Deutsche Gesellschaft für Kybernetik) (2002) Was versteht die GfK unter Kybernetik?,
<http://www.gesellschaft-fuer-kybernetik.org/>, Download am 06.05.2005
- Göbel E (1998) Theorie und Gestaltung der Selbstorganisation. Duncker & Humblodt, Berlin
- Gou L, Luh P B, Kyoya Y (1998) Holonic manufacturing scheduling, architecture, cooperation mechanism, and implementation. Computers in Industry 37(1998)3:213-231
- Haken H (1990) Über das Verhältnis der Synergetik zur Thermodynamik, Kybernetik und Informationstheorie. In: Niedersen U, Pohlmann L (Hrsg) Selbstorganisation und Determination. Duncker & Humblodt, Berlin, S 19-24
- Huhns M N, Singh M P (1998) Multi-agent Systems in Information-Rich Environments. In: Klusch M, Weiß G (eds) Cooperative Information Agents II. Springer, Berlin Heidelberg New York, pp 79-93
- Hummel Th, Malorny Ch (1997) Total Quality Management – Tipps für die Einführung. Carl Hanser, München Wien
- Jennings N R, Wooldridge M J (eds) (1998) Agent Technology, Foundations, Applications, and Markets. Springer, Berlin Heidelberg New York

- Jünemann R, Beyer A (1998) Steuerung von Materialfluss- und Logistiksystemen – Informations- und Steuerungssysteme, Automatisierungstechnik. Springer, Berlin Heidelberg New York
- Kilger C, Schneeweiss L (2000) Demand Fulfilment and ATP. In: Stadler H, Kilger C (eds) Supply Chain Management and Advanced Planning, Concept, Models, Software and Case Studies. Springer, Berlin Heidelberg New York, pp 135-148
- Klügl F (2001) Multiagentensimulation, Konzepte, Werkzeuge, Anwendungen. Addison-Wesley, München
- Krallmann H, Albayrak S (2002) Holonic Manufacturing, Agentenorientierte Techniken zur Umsetzung von holonischen Strukturen. TCW, München
- Kuhn A, Hellingrath H (2002) Supply Chain Management, Optimierte Zusammenarbeit in der Wertschöpfungskette. Springer, Berlin Heidelberg New York
- Lind J, Fischer K (1998) Transportation Scheduling and Simulation in a Railroad Scenario, A Multi-Agent Approach. DFKI, Saarbrücken
- Lind J (2001) Iterative Software Engineering for Multiagent Systems, The MASSIVE Method. Springer, Berlin Heidelberg New York
- Lüth T (1998) Technische Multiagentensysteme – Verteilte autonome Roboter- und Fertigungssysteme. Carl Hanser, München Wien
- Luczak H, Eversheim W, Schotten M (Hrsg) (1998) Produktionsplanung und -steuerung, Grundlagen, Gestaltung und Konzepte. Springer, Berlin Heidelberg New York
- Marik V, Fletcher M, Pechoucek M (2002) Holons & Agents, Recent Developments and Mutual Impacts. In: Marik V, Stepankova O, Krautwurmova H, Luck M (eds) Multi-Agent Systems and Applications II. Springer, Berlin Heidelberg New York, pp 233-267
- Müller J P (1996) The Design of Autonomous Agents, A Layered Approach. Springer, Berlin Heidelberg New York
- Much D, Nicolai H (1995) PPS-Lexikon. Cornelsen, Berlin
- Oechslein C (2004) Vorgehensmodell mit integrierter Spezifikations- und Implementierungssprache für Multiagentensimulationen. Shaker, Aachen
- Parunak H v D (2000) A Practitioners' Review of Industrial Agent Applications. Autonomous Agents and Multi-Agent Systems (2000)3:389-407
- Pavliška A, Srovnal V (2002) Robot Dissassembly Process Using Multi-agent System. In: Dunin-Keplicz B, Nawarecki E (eds) From Theory to Practice in Multi-Agent Systems. Springer, Berlin Heidelberg New York, pp 227-233
- Peters K, Becker M, Wenning B-L, Timm-Giel A, Görg C (2005) Autonomous Logistics Systems Integrating Communication, Computation and Transportation Networks. (Submitted to the First International IEEE WoWMoM Workshop on Autonomic Communications and Computing (ACC 2005))
- Pfohl H-Chr (2000) Logistiksysteme – Betriebswirtschaftliche Grundlagen. Springer, Berlin Heidelberg New York
- Pritschow G, Duellen G, Bender K (1996) Steuerung von Produktionssystemen, In: Eversheim W, Schuh G (1996) Betriebsstätte – Produktion und Management. Springer, Berlin Heidelberg New York, S 10-73
- Protégé Team (2005) Protégé 3.0.

- <http://protege.stanford.edu/index.html>, Download am 06.05.2005
- Puppe F, Stoyan H, Studer R (2000) Knowledge-Engineering. In: Görz G, Rollinger C-R, Schneeberger J (Hrsg) Handbuch der Künstlichen Intelligenz, 3. Aufl. Oldenbourg, München Wien, S 599-641
- SCHO 540/8-1 Das Forschungsprojekt „Reaktive Steuerung von Lieferketten mit Agentensystemen“ wurde unter dem Kennzeichen SCHO 540/8-1 von der Deutschen Forschungsgemeinschaft (DFG) im Rahmen des Schwerpunktprogramms (SPP) 1083 „Intelligente Agenten in betriebswirtschaftlichen Anwendungen“ im Zeitraum von Mai 2001 bis Mai 2003 gefördert.
http://www.ips.biba.uni-bremen.de/projekt.html?&no_cache=1&proj=8
(Stand: Mai2005)
- Scholz-Reiter B (1998) Chancen und Möglichkeiten der reaktiven Planung und Steuerung von intermodalen Stückguttransporten. In: Fluhr M (Hrsg) Innovative Lösungen für den Verkehr von morgen. in Time, Berlin, S 32-48
- Scholz-Reiter B, Scharke H (2000) Reaktive Planung. *Industrie Management* (2000)2:21-26
- Scholz-Reiter B, Höhns H (2001) Reaktive Planung und Steuerung logistischer Prozesse mit Multiagentensystemen (MAS). *Industrie Management* (2001)6:33-36
- Scholz-Reiter B, Höhns H (2003) Wissensbasierte Auftragskoordination im Supply Chain Management. *Industrie Management* (2003)3:26-29
- Scholz-Reiter B, Höhns H, Hamann T (2004) Adaptive Control of Supply Chains, Building Blocks and Tools of an Agent-based Simulation Framework. *Annals of the CIRP*, S 353-356
- Scholz-Reiter B, Windt K, Kolditz J, Böse F, Hildebrandt T, Philipp T, Höhns H (2004) New Concepts of Modelling and Evaluating Autonomous Logistic Processes. In: Chryssolouris G (ed) Proceedings of the IFAC-MIM'04 Conference on Manufacturing, Modelling, Management and Control (Pre-Prints). (To appear in formal Proceedings)
- Scholz-Reiter B, Höhns H, Kolditz J, Hildebrandt T (2005) Autonomous Supply Net Coordination. In: Proceedings of the 38th CIRP Seminar Manufacturing on Systems, Florianopolis Brazil (electronic Proceedings on CD-ROM)
- Scholz-Reiter B, Hildebrandt T, Kolditz J, Höhns H (2005) Selbststeuerung in der Produktion, Ein Modellierungskonzept. *Industrie Management* (2005)4:33-36
- SeSAM (2004) SeSAM, Shell for Simulated Agent Systems.
<http://www.simsesam.de>, Download am 06.05.2005
- SFB 637 „Selbststeuerung logistischer Prozesse – Ein Paradigmenwechsel und seine Grenzen“ an Universität Bremen, wird von der Deutschen Forschungsgemeinschaft (DFG) seit Januar 2004 gefördert.
<http://www.sfb637.uni-bremen.de/home.html> (Stand: Mai 2005)
- Smith R G (1980) The Contract-Net Protocol, High-Level Communication and Control in a Distributed Problem Solver. *IEEE Transactions on Computation* (1980)29:1104-1113
- Sommerer G (1998) Unternehmenslogistik – Ausgewählte Instrumentarien zur Planung und Organisation logistischer Prozesse. Carl Hanser, München Wien

- Stickel E, Groffmann H-D, Rau K-H (Hrsg) (1997), Gabler-Wirtschaftsinformatik-Lexikon, Gabler, Wiesbaden
- Supply Chain Council (2005) The Supply Chain Operations Reference Model 7.0. <http://www.supplychain.org/galleries/defaultfile/SCOR%207.0%20Overview.pdf>, Download am 06.05.2005
- The PABADIS consortium (2002) Development of a machine representation – Task 2.3. of Work Package 2. <http://www.pabadis.org/htdocs/info.html>, Download am 22.03.2005
- Tomys A-K (1995) Kostenorientiertes Qualitätsmanagement – Qualitätscontrolling zur ständigen Verbesserung der Unternehmensprozesse. Carl Hanser, München Wien
- Valckenaers P, Brussel H v, Kollingbaum M, Bochmann O (2001) Multi-agent Coordination and Control Using Stigmergy Applied to Manufacturing Control. In: Luck M, Marik V, Stepankova O, Trappl R (eds) Multi-Agent Systems and Applications. Springer, Berlin Heidelberg New York, pp 317-334
- Vrba P, Hrdonka V (2002) Material Handling Problem, FIPA Compliant Implementation. In: Marik V, Stepankova O, Krautwurmova H, Luck M (eds) Multi-Agent Systems and Applications II. Springer, Berlin Heidelberg New York, pp 268-279
- Wagner T (2003) An Agent-Oriented Approach to Industrial Automation Systems. In: Kowalczyk R, Müller J P, Tianfield H, Unland R (eds) Agent Technologies, Infrastructures, Tools, and Applications for E-Services. Springer, Berlin Heidelberg New York, pp 314-328
- Warnecke G (1993) Revolution der Unternehmenskultur, Das Fraktale Unternehmen. Springer, Berlin Heidelberg New York
- Weiß G, Jakob R (2005) Agentenorientierte Softwareentwicklung, Methoden und Tools. Springer, Berlin Heidelberg New York
- Windt K, Höhns H, Scholz-Reiter B (2002) Szenarien für die Reorganisation, Optimierung von Lager- und Distributionsstrukturen durch IT in Logistiknetzwerken. DVZ Deutsche Logistik-Zeitung (2002)56:10
- Wooldridge M J, Jennings N R (1999) Software engineering with agents, pitfalls and pratfalls. IEEE Internet Computing (1999)3:20-27
- Wooldrige M J, Jennings N R, Kinny D (2000) The Gaia Methodology for Agent-Oriented Analysis and Design. Autonomous Agents and Multi-Agent Systems (2000)3:285-312