

**AVALANCHE -
Ein agentenbasierter dezentraler
Koordinationsmechanismus
für elektronische Märkte**

Inaugural-Dissertation
zur Erlangung der Würde eines
Doktors der Wirtschaftswissenschaften
der Wirtschaftswissenschaftlichen Fakultät
der Albert-Ludwigs-Universität Freiburg im Breisgau

vorgelegt von
Torsten Eymann
aus Lübeck

2000

Erstgutachter: Prof. Dr. Günter Müller

Zweitgutachter: Prof. Dr. Franz Schober

Dekan: Prof. Dr. Heinz Rehkugler

Tag des Promotionsbeschlusses: 8. November 2000

Die Gutachter wurden durch die Fakultät bestimmt.

Vorwort

Die Gestaltung elektronischer Marktplätze ist eine wirtschaftswissenschaftliche Fragestellung, die zur Umsetzung auf die Technologien der Informatik angewiesen ist. Für die Interaktion zwischen Menschen sind bis auf den Bereich der Sicherheit diese Technologien längst verfügbar (vgl. Müller/Rannenberg 1999) und organisatorische, betriebswirtschaftliche und rechtliche Fragen treten jetzt in den Vordergrund. In einem völlig anderen Entwicklungsstadium befinden wir uns dann, wenn diese Interaktion durch Menschen auf Computerprogramme (Agenten) übertragen wird. Hier ist noch die Innovation der Technik gefordert, die am Anfang jeder Kulturentwicklung steht (vgl. Ropohl 1999). Sie nimmt die Probleme und Konzepte der Wirtschaftswissenschaften von vornherein als wichtigen Input für die Technikgestaltung. Daher war es von Anfang an wichtig, auch die Ziele klar zu formulieren - obwohl die technische Umsetzung in dieser Arbeit den meisten Raum einnimmt, geht es mir weniger um die Technik selbst, sondern um ihre Anwendung für die dezentrale Koordination in elektronischen, agentenbasierten Märkten. Das in dieser Arbeit behandelte Thema wurzelt damit in zwei vollkommen unterschiedlichen wissenschaftlichen Disziplinen, den Wirtschaftswissenschaften und der Informatik. Über die Jahre haben sowohl Informatiker als auch Ökonomen mit mir über AVALANCHE diskutiert. Viele Anregungen haben Eingang in die Implementation gefunden und die technischen Grundlagen verbessert; andere haben den Forschungsschwerpunkt kommentiert und mich die Fragestellung fokussierter darstellen lassen.

Meinem akademischen Lehrer, Prof. Dr. Günter Müller, schulde ich viel Dank für die fachliche Begleitung und Unterstützung in jeder Phase dieser Arbeit. Die Frage nach dem gegenwärtigen Grad und Stellenwert des Einbeziehens wirtschaftswissenschaftlicher Konzepte in eine noch unfertige, aber auf präzise Formalisierung angewiesene Technologie hat uns oft unterschiedlicher Meinung sein lassen. Es wurde nie zum persönlichen Konflikt und am Ende haben mich seine Bemerkungen, dass Wissenschaft und Esoterik keine Schnittmenge haben, bodenständig bleiben lassen, wenn die Begeisterung an dem Potential der Technik mit mir durchzugehen begann. Günter Müller hat solchen Streit immer als Teil einer akademischen Ausbildungssituation verstanden und nie gescheut, obgleich uns beiden diese unmittelbare Interpretation der Rolle eines Doktorvaters wahrscheinlich manche Nerven gekostet hat. Es hat sich gelohnt.

Einen wesentlichen Einfluß auf meine Arbeit haben meine Kollegen der Abteilung Telematik gehabt, die mit ihrer jeweiligen Herkunft aus Informatik oder Wirtschaftswissenschaften von beiden Seiten immer für fundamentale als auch detaillierte, immer aber konstruktive Kritik

sorgten. Herausheben möchte ich die Rolle von Dr. Detlef Schoder, der mit mir viele Stunden über betriebswirtschaftliche Fragestellungen diskutiert und immer wieder für Motivation und intellektuelle Stimulation gesorgt hat. Meinen beiden Kollegen im Projekt AVALANCHE, Boris Padovan und Stefan Sackmann, kann ich nur dadurch gerecht werden, dass ich sage, wir sind gute Freunde geworden und dass viele Ideen nur durch unsere gemeinschaftliche Arbeit an unserer Vision elektronischer Märkte entstehen konnten.

Günter Müller hat gerade für dieses Projekt immer Wert auf den Diskurs mit anderen Wissenschaftlern gelegt. Von diesen haben einige meine Arbeit und meine Denkweise besonders geprägt. Bernardo Huberman von Xerox PARC hat mich in einem kurzen Gespräch zu Anfang meines Projekts von Überlegungen der Simulation meiner Fragestellung mit Hilfe der Software SWARM des Santa Fe Institutes abgebracht. Leigh Tesfatsion habe ich für die Einführung in Agent-Based Computational Economics zu danken. Mit Paul Kearney, Robert Smith, Rosaria Conte und Gerard Weisbuch verbinden hilfreiche Diskussionen im Umfeld des „Future and Emerging Technologies Programme“ der Europäischen Union. Aus diesem Kontakt ergab sich ein Forschungsaufenthalt bei British Telecom in Martlesham Heath, der von AgentLink, dem Network of Excellence in Agent-Based Computing der EU, gefördert wurde. Stefan Kirn von der TU Ilmenau danke ich für die kollegiale Aufnahme und gemeinsame Vorbereitung eines Agenten-Workshops in Frankfurt Ende 1998. Ich gratuliere ihm, das aus dieser Keimzelle das DFG-Schwerpunktprogramm zu Software-Agenten entstand, dessen Koordinator er ist. Prof. Dr. Franz Schober danke ich für die Übernahme des Koreferats.

Avalanche ist aber auch das Ergebnis vieler Bemühungen von Studierenden der Telematik. Sie haben mit uns im Rahmen von Haus-, Seminar- oder Diplomarbeiten trotz einer unvollständigen Dokumentation teilweise erstaunliche Ergebnisse zustande gebracht. Thomas Albrecht und Kai Lessmann, die jeweils die erste (1998) und die zweite Version der Software (1999) grundlegend gestaltet haben, verdienen dabei besonderen Respekt.

Ein besonderer Dank gebührt auch meinen Eltern, meinem erweiterten eigenen Familienkreis und Steffis Familie, die mich angespornt und mir vielfältige Unterstützung und Motivation haben zukommen lassen. Von allen Menschen aus meinem privaten Kreis schulde ich jedoch Steffi am meisten Dank.

Freiburg, im November 2000

Torsten Eymann

Inhaltsverzeichnis

Vorwort	III
Inhaltsverzeichnis	V
Abbildungsverzeichnis	IX
1 Einführung	1
1.1 Software-Agenten in elektronischen Märkten	1
1.2 Koordinationskonzepte der Verteilten Künstlichen Intelligenz (VKI) in elektronischen Märkten	5
1.3 Eine Fallstudie: Der Schwarze Montag an der Wall Street, 1987	7
1.4 Folgen für die Gestaltung marktlicher Koordinationsmechanismen	11
1.5 Koordination im Schichtenmodell der Telematik	13
1.6 Das Koordinationsproblem des elektronischen Marktplatzes	16
1.7 Anforderungen an den Koordinationsmechanismus	20
1.8 Aufbau der Arbeit	24
2 Der Markt als Koordinationsverfahren	27
2.1 Koordination als Ausgleich von Interdependenzen	28
2.2 Der Markt als dynamischer Koordinationsmechanismus	30
2.2.1 Spieltheoretische Konzepte der marktlichen Koordination	32
2.2.2 Marktliche Koordination als Ressourcenallokationsverfahren	36
2.3 Der Markt als Arbitrationsverfahren	40
2.4 Der Markt als dezentraler Koordinationsmechanismus	47
2.4.1 Der Walras'sche Auktionator und Market-Oriented Programming	48
2.4.2 Der Ansatz der "Agoric Open Systems"	53
2.4.3 Auktionsverfahren	56
2.4.4 Konsequenzen für den Einsatz von Auktionsverfahren	59
2.5 Koordination durch marktliche Selbstorganisation	62
2.5.1 Die Katallaxie als Erklärungsmodell	65
2.5.2 Selbstorganisatorische Koordination für offene Märkte	74

3	Koordination durch autonome Software-Agenten in Mehragenten-Systemen	77
3.1	Komponenten für Software-Agenten	79
3.1.1	Einfluß der Umgebung auf die Gestaltung der Agenten	80
3.1.2	Modellierung der Agenten	83
3.2	Mehragentensysteme	91
3.3	Koordination in Mehragentensystemen	92
3.3.1	Koordination durch verteiltes Problemlösen	95
3.3.2	Marktliche Koordination in Multi-Agenten-Systemen	100
3.4	Kommunikation zwischen Agenten	104
3.5	Automatisiertes Verhandeln zwischen Agenten	109
3.5.1	Die Handelsgüter	110
3.5.2	Das Ziel der Agenten	111
3.5.3	Ansätze für Verhandlungsstrategien	113
3.5.3.1	Regelbasierte Ansätze	116
3.5.3.2	Spieltheoretische Ansätze	117
3.5.3.3	Adaptive Ansätze	120
3.5.3.4	Welche Variablen bestimmen die Strategie?	122
3.5.3.5	Adaption der Strategie mit evolutionären Algorithmen	127
3.6	Zusammenfassung der Anforderungen an die Implementierung	131
4	Realisierung marktlicher Koordination in einem MAS	135
4.1	Referenzszenario der Koordination einer Wertschöpfungskette	136
4.2	Architektur eines agentenbasierten Marktes	139
4.3	Strukturierung der Datentypen	141
4.4	Datentypen und Methoden der Agentenklassen	143
4.5	Datentypen und Methoden der handelnden Agenten	144
4.5.1	Datentypen des individuellen Verhaltens	149
4.5.2	Kommunikation zwischen den Agenten	150
4.5.3	Kooperation, Interaktion und Verhandlungen	153
4.5.4	Güter	154
4.5.5	Zielbildung der Agenten	155
4.5.6	Verhandlungsstrategie	156
4.5.7	Realisierung der Adaptivität	161

4.5.8	Das Verhandlungsprotokoll	162
4.6	Klassen und Methoden des Marktplatzes	169
4.7	Datentypen und Methoden der Experimentsteuerung	171
4.8	Vorläufige Ergebnisse der Realisierung	173
5	Avalanche als Experimentierumgebung für marktliche Koordination	175
5.1	Veränderungen der Güterpreise	177
5.2	Anzahl der Transaktionen	181
5.3	Gewinnentwicklung bei unterschiedlicher Konzessionsrate	182
5.4	Anpassung von Strategieparametern durch STDEA	185
5.5	Erreichen von Preisebenen aus zufälligen Startpreisen	189
5.6	Veränderungen der Konzessionshöhe	194
6	Ergebnisse und Ausblick	203
6.1	Thematische Einordnung der Ergebnisse	205
6.2	Avalanche - ein Prototyp für ein Katallaktisches Informationssystem	208
6.2.1	Herausforderungen an den praktischen Einsatz von MAS	209
6.2.2	Zukünftiger Einsatz von MAS im digitalen Wirtschaften	212
	Abkürzungsverzeichnis	213
	Literaturverzeichnis	216
	Autorenindex	234
	Stichwortverzeichnis	241
	Glossar	245
	Grafische Übersicht von Avalanche	250
	Programmdokumentation Avalanche	Anhang

Abbildungsverzeichnis

Abb. 1:	Einsatzmöglichkeiten von VKI-Methoden auf elektronischen Marktplätzen	6
Abb. 2:	Schichtenmodell der Telematik (Müller et al. 1997, S. 41)	13
Abb. 3:	Koordination durch Marktmechanismen in Multi-Agenten-Systemen	21
Abb. 4:	Kapitelübersicht in grafischer Darstellung	25
Abb. 5:	Koordinationsmechanismen nach Moulin (1995)	33
Abb. 6:	Market-Oriented Programming	51
Abb. 7:	Katallaxie als Erklärungsmodell des Koordinationsmechanismus "Markt"	70
Abb. 8:	Rückkopplung zwischen Agent und Umgebung (nach Wooldridge 1999, S. 29)	80
Abb. 9:	Übergang von geschlossenen zu offenen Mehragentensystemen	82
Abb. 10:	Das k.-Ebene-Konzept von Vidal/Durfee (1996)	84
Abb. 11:	Ergebnisse der Experimente von Gode/Sunder (1993), aus Cliff (1997, S. 25)	88
Abb. 12:	Konvergieren von ZIP-Agenten auf einen Gleichgewichtspreis	90
Abb. 13:	Aufbau einer ACL Nachricht (aus FIPA 1997, S. 11)	108
Abb. 14:	Bilateraler Verhandlungsprozeß (nach de Paula et al. 2000, S. 3)	114
Abb. 15:	Prinzip eines agentenbasierten, dezentralen evolutionären Algorithmus (Smith/Taylor 1998, Fig. 2)	129
Abb. 16:	Agentenbasierte Realisierung eines elektronischen Marktes	133
Abb. 17:	Avalanche zwischen Markt und Multi-Agenten-System	135
Abb. 18:	Szenario der Wertschöpfungskette in Avalanche	137
Abb. 19:	UML-Anwendungsfalldiagramm von Avalanche	140
Abb. 20:	UML-Aktivitätendiagramm für AvTradeAgent	146
Abb. 21:	UML-Komponentendiagramm für AvTradeAgent	149
Abb. 22:	AUML-Sequenzdiagramm der Verhandlung zwischen Agenten	164
Abb. 23:	Zustandsdiagramm eines Agenten in der Käuferrolle	165
Abb. 24:	AUML-Sequenzdiagramm der Verzeichnisdienstabfrage	171
Abb. 25:	Startphase mit festgelegten Preisen	178
Abb. 26:	Experimentverlauf bei festgelegten Startpreisen und homogener Strategieverteilung	179
Abb. 27:	Gewinnentwicklung bei festgelegten Startpreisen	180
Abb. 28:	Preisentwicklung bei einer Acquisitiveness-Vorgabe von 0.9	182
Abb. 29:	Preisentwicklung mit hartnäckigen Zimmermann-Agenten	183
Abb. 30:	Gewinnentwicklung bei hartnäckigen Zimmermannagenten	184
Abb. 31:	Entwicklung des Acquisitiveness-Parameters durch den evolutionären Algorithmus ($p_{acq} = 0.5$)	185
Abb. 32:	Entwicklung des Acquisitiveness-Parameters durch den evolutionären Algorithmus	

(p_acq = 0.6)	186
Abb. 33: Einschwingen der Preise bei Koevolution	188
Abb. 34: Koevolution des Acquisitiveness-Parameters	189
Abb. 35: Experiment mit zufälliger Preisausstattung (173 Transaktionen)	191
Abb. 36: Experiment mit zufälliger Preisausstattung (973 Transaktionen)	192
Abb. 37: Experiment mit zufälliger Preisausstattung (1973 Transaktionen)	192
Abb. 38: Gewinnentwicklung bei zufälliger Preisausstattung	193
Abb. 39: Evolution der Konzessionshöhe	194
Abb. 40: Preisentwicklung bei Evolution der Konzessionshöhe	195
Abb. 41: Evolution der Konzessionhöhe (2. Versuch)	196
Abb. 42: Preisentwicklung bei simultaner Evolution von p_acq und del_change	197
Abb. 43: Koevolution der Parameter p_acq und del_change	198
Abb. 44: Preisentwicklung bei zufälligen Startpreisen und Variation von p_acq und del_change	199
Abb. 45: Koevolution von p_acq, del_change und p_sat	200
Abb. 46: Vom MAS zum Markt: Konzepte und Implementationen	204
Abb. 47: Prinzipdarstellung der Remote Method Invocation (RMI)	252
Abb. 48: Zugriff auf entfernte Objekte in Voyager (aus Objectspace 1998, S. 9)	254
Abb. 49: UML-Klassendiagramm von Avalanche	257
Abb. 50: UML-Klassendiagramm der handelnden Agenten	258
Abb. 51: UML-Klassendiagramm des Marktplatzes	259
Abb. 52: UML-Klassendiagramm des evolutionären Algorithmus STDEA	260
Abb. 53: UML-Klassendiagramm der Inter-Agenten-Kommunikation	261

1 Einführung

1.1 Software-Agenten in elektronischen Märkten

„We believe that, over the course of the next decade, the global economy and the Internet will merge into an information economy bustling with billions of autonomous software agents that exchange information goods and services with humans and other agents. Agents will represent – and be – consumers, producers, and intermediaries. They will facilitate all facets of electronic commerce, including shopping, advertising, negotiation, payment, delivery, and marketing and sales analysis. The agents we envision will not be mere adjuncts to business processes. They will be economic software agents: independent, self-motivated economic players, endowed with algorithms for maximizing utility and profit on behalf of their human owners. From other agents, they will purchase inputs, such as network bandwidth, processing power, or database access rights, as well as more refined information products and services. They will add value to these inputs by synthesizing, filtering, translating, mining, or otherwise processing them, and will sell the resultant product or service to other agents. In essence, these agents will function as miniature automated businesses that create and sell value to other agents, and in so doing will form complex, efficient economic webs of information goods and services that respond adaptively to the ever-changing needs of humans for physical and information-based products and services.“ (Kephart et al. 2000, S.2)¹

Der Einsatz von Software-Agenten, die im Namen menschlicher Anwender autonom wirtschaftliche Transaktionen durchführen, z.B. im Rahmen elektronischer Märkte und des elektronischen Geschäftsverkehrs, stellt eine Synthese von Konzepten der Künstlichen Intelligenz, der Wirtschaftsinformatik und der Wirtschaftswissenschaften dar. Der Begriff des Software-Agenten wird im folgenden so verstanden:

„An agent is an encapsulated computer system that is situated in some environment, and that is capable of flexible autonomous action in that environment in order to meet its design objectives“ (Wooldridge/Jennings 1999, S. 2)

¹ Implizit liegt diesem Dienstekreislauf ein entgegengesetzter Geldkreislauf zugrunde, der im Zitat durch die Wortwahl („purchase inputs“, „sell [...] services“) deutlich gemacht wird.

Agenten unterscheiden sich von traditionellen Software-Anwendungen vor allem durch ihre Autonomie, die als zielgerichtetes und proaktives, d.h. selbst-startendes Verhalten begriffen werden kann. Diese Eigenschaft erlaubt es ihnen, in einer definierten Umgebung zusammen mit anderen Agenten kontinuierlich und selbständig Aufgaben zielgerichtet auszuführen. Sie sind daher gerade für den Einsatz in dynamischen Umgebungen mit einer Vielzahl sich ständig ändernder Informationen und parallel durchgeführten Prozesse geeignet. Der elektronische Geschäftsverkehr in offenen Netzen und dort insbesondere elektronische Märkte, stellen eine solche Umgebung dar.

Ein elektronischer Markt „basiert auf einem oder mehreren zusammenarbeitenden Informationssystemen, die einzelne oder alle Phasen einer Markttransaktion elektronisch unterstützen, integrieren oder gar automatisieren“ (Müller et al. 1997, S. 299)². Die Marktplätze selbst können dabei auf unterschiedliche Produkte spezialisiert sein, z.B. Rohstoffe, Wirtschaftsgüter, Informationsgüter oder Dienstleistungen, was sich letztendlich aus der Zusammensetzung der Teilnehmer ergibt. Software-Agenten könnten diese Vielfalt an Märkten nicht nur permanent im Auftrag ihres Absenders beobachten (Anbahnungsphase), sondern auf ihnen auch Transaktionen initiieren. Menschliche Anwender, die an elektronischen Märkten teilnehmen wollen, könnten Software-Agenten als Repräsentanten einsetzen, um ihre Transaktionskosten über alle Phasen zu senken. Durch die Entwicklung automatisierter Verhandlungskonzepte könnten sowohl Käufer- als auch Verkäufer-Agenten automatisiert und autonom auf elektronischen Marktplätzen im Auftrag ihres Anwenders handeln. Wenn der Agent ein interessantes Kauf- oder Verkaufsangebot identifizieren würde, so würde er umgehend Verhandlungen mit dem betreffenden Partneragenten (Vereinbarungsphase) beginnen. Entsprechen sich Angebot und Nachfrage, könnten die Agenten selbständig eine Transaktion durchführen und den Nutzen des menschlichen Besitzers vermehren (Abwicklungsphase). Ob Software-Agenten jemals in dieser Form auf elektronischen Marktplätzen eingesetzt werden, kann heute noch nicht mit Sicherheit vorausgesagt werden. Damit diese Entwicklung eintreffen kann, müssen vier Voraussetzungen erfüllt sein.

1. Der menschliche Anwender und Besitzer des Agenten (im folgenden vereinfacht Prinzipal³ genannt) muß seine Präferenzen bezüglich eines optimalen Zielzustandes definieren kön-

2 Es gibt eine Vielzahl unterschiedlicher Phasenmodelle für elektronische Märkte (vgl. Guttman et al. 1998, Schmid/Lindemann 1998, Kalakota/Whinston 1996). Im folgenden wird das dreistufige Phasenmodell von Müller et al. (1997, S. 300) verwendet, welches die Anbahnungs-, Vereinbarungs- und Abwicklungsphase unterscheidet.

3 Die Wortwahl „Prinzipal“ und „Agent“ ist der Principal-Agency-Theory entnommen (Ross 1973). Dies stellt keineswegs eine Einschränkung oder Problematisierung im Sinne dieser Theorie dar. Die Benutzung dieser Wörter soll lediglich die enge hierarchische Beziehung zwischen Mensch und Software verdeutlichen, bei der jeder Agent einem Menschen gehört, der ihn kontrolliert; die Handlungen des Agenten werden dem ihn kontrollierenden Menschen zugeschrieben.

nen. Dieser Zustand lässt sich üblicherweise als Allokation von Ressourcen definieren, z.B. als eine Zusammenstellung von Gütern und Geld, die für den Besitzer den größtmöglichen Nutzen darstellt. Außerdem muß er eine Strategie formal beschreiben können, um im Zeitablauf vom gegenwärtigen Zustand zu dem präferierten Endzustand zu gelangen.

2. Es besteht ein Vorteil oder eine Notwendigkeit in der Vertretung des Prinzipals durch einen Software-Agenten. Entweder erfordert die technische Nutzung eines elektronischen Marktsystems den Einsatz von Software-Agenten, oder es gibt eine wirtschaftliche Nutzenbetrachtung, in der durch den Einsatz Transaktionskosten gesenkt werden können.
3. Der Software-Agent (im folgenden meistens vereinfacht Agent genannt) führt Aktionen aus, deren kausale und zeitliche Aufeinanderfolge durch die Strategie des Prinzipals bestimmt wird. Es besteht somit eine eindeutige hierarchische Ordnung zwischen Prinzipal und Agent. Die Prinzipale stehen in keiner hierarchischen Beziehung zueinander und vertrauen im wesentlichen dem abstrakten Koordinationsmechanismus bezüglich der Auflösung von Konflikten und der Möglichkeit, die eigene Nutzenmaximierung verfolgen zu können. Agenten ohne Besitzer werden im folgenden nicht betrachtet. Nicht betrachtet werden auch Probleme der Repräsentanz einer juristischen Person "Unternehmen" durch natürliche Personen (Eigentümer oder angestellte Mitarbeiter), die wiederum Agenten einsetzen, d.h. rechtliche Vertretungsprobleme oder betriebswirtschaftliche „principal-agency“-Probleme, deren Lösung in der nichttechnischen Sphäre zu suchen ist.
4. Die wichtigste Voraussetzung ist die Bereitstellung eines Regelwerkes, welches eine planmäßige Zielverfolgung der Agenten auf den elektronischen Marktplätzen zuläßt. Erst dieses ermöglicht eine Akzeptanz einer agentenbasierten Technologie digitalen Wirtschaftens durch die menschlichen Prinzipale. Diese werden Agenten nur dann einsetzen, wenn überhaupt ein Ergebnis prognostiziert und Abweichungen begründet werden können. Ein solches Regelwerk kann jedoch allein kein Ergebnis durchsetzen, so dass Steuerungsmechanismen für elektronische Marktplätze benötigt werden.

Elektronische Marktplätze bilden einen Teil einer zukünftigen Globalen Informations-Infrastruktur (GII). Das Internet als Prototyp der GII besitzt bereits eine technisch offene Architektur, die voneinander unabhängige Anwendungen und autonomes Handeln von menschlichen Anwendern und Diensteanbietern erlaubt. Jeder Teilnehmer kann grundsätzlich sowohl als Güter- oder Diensteanbieter als auch als Nachfrager auftreten und ad-hoc über große Entfernungen mit vorher unbekanntem Teilnehmern kooperieren (Müller et al. 1997, S. 25ff.). Diese Teilnehmer können neben Menschen auch Software-Agenten sein, die gleichzeitig in autonomer, individualistischer und nicht notwendigerweise kooperativer Weise ihren Aufgaben nachgehen. In diesem Fall werden sie unweigerlich unterschiedlichen Organisationen, Autoritäten

und Kulturen angehören, die in ebenso unterschiedlichen Weisen Kommunikation und Kooperation implementieren. Um trotz dieser Vielfalt und Unsicherheit eine dynamische Organisation zu schaffen, welche ein zielgerichtetes Handeln des Einzelnen ermöglicht, sind technische und nichttechnische Institutionen⁴ erforderlich. Ähnliche Forschungsfragen sind z.B. wie individualistische Software-Agenten zu einer Ökonomie zusammenfinden (Lane 1993), oder wie sich Normen und Regeln in Mehragenten-Systemen entwickeln (Conte/Castelfranchi 1995). Solche Probleme sind bisher wenig experimentell erforscht und es besteht ein fundamentaler Unterschied in den Lösungsmöglichkeiten zwischen konventionellen Informationssystemen und der offenen Umgebung der GII.

In **verteilten, aber geschlossenen** Umgebungen ist es möglich, Agenten, Koordinationsmechanismus und Umgebung zusammenpassend zu gestalten und von einer zentralen Stelle aus zu steuern. In diesem Fall ist durch den Systemverwalter ein globaler Zugriff auf alle Agenten möglich. Agenten, welche die Erreichung eines möglicherweise existierenden Systemziels beeinträchtigen, können aus der Umgebung nach Maßgabe des Systemverwalters entfernt werden. Die Austragung von Konflikten zwischen den Teilnehmern wird meist bereits im Vorfeld des Systemdesigns ausgeschlossen. Die Unterordnung jedes Individualziels unter das Systemziel fixiert auch eine Hierarchie, an deren Spitze der Systemdesigner steht und dem die einzelnen Teilnehmer und Software-Agenten untergeordnet sind.

In **offenen** Netzen muß die Steuerung der Agenten auf anderem Wege bewerkstelligt werden. Ein koordiniertes Handeln der Agenten kann nicht erzwungen werden. Die Ausrichtung von autonomen, dezentralen Entscheidungsträgern verlangt nach einem Ausgleich zwischen der Verfolgung der jeweiligen individuellen Ziele. Konflikte⁵ zwischen Individualzielen oder Individualziel und dem Ziel des Systemdesigners lassen sich nicht durch soziale Kontakte der Agentenbesitzer außerhalb des technischen Systems lösen. Zur Konfliktlösung gibt es nur die Möglichkeit, innerhalb des Systems über Verhandlungen eine Einigung herbeizuführen, d.h. die Agenten zu koordinieren.

Sollte eine derartige Koordination nicht erreichbar sein, können die Agenten nicht planmäßig ihre Ziele verfolgen und es ergibt sich ein Systemzustand, der von Thomas Hobbes so beschrieben wurde:

„A condition of war of every one against every one, in which case every one is governed by his own reason, and there is nothing he can make use of that may not be a help unto

4 Der Institutionenbegriff wird im Sinne von North (1998) verstanden und umfaßt neben physischen Objekten wie z.B. Gerichten auch immaterielle Objekte wie Regeln und Normen.

5 Zwei Ziele stehen dann in Konflikt miteinander, wenn die Nutzenerhöhung eines Entscheidungsträgers diejenige eines anderen Entscheidungsträgers ausschließt oder vermindert.

him in preserving his life against his enemies; it followeth that in such a condition every man has a right to every thing, even to one another's body“ (Hobbes 1651/1980, Kap. 13).

Es braucht nicht betont zu werden, dass ein solcher Zustand in einem Mehragenten-System unbedingt durch die Entwicklung geeigneter Verfahren vermieden werden muß. Anderenfalls werden Agenten und Agentensysteme nicht genutzt, so dass jede technische, ökonomische und intellektuelle Investition in diese Technologie vergeblich ist.

Für die Akzeptanz der Technologie der Softwareagenten im digitalen Wirtschaften ist es entscheidend, Koordinationsmechanismen zu entwickeln, die als Regelwerke eine planmäßige Verfolgung individualistischer Ziele in elektronischen Märkten ermöglichen. Die wesentliche Fragestellung dieser Arbeit besteht daher darin, einen automatisierten Koordinationsmechanismus für Multi-Agenten-Systeme zu konstruieren, der eine menschlichen Teilnehmern vergleichbare Koordinationsleistung erreicht. Wenn dieses Ergebnis durch experimentelle Untersuchungen eines Prototypen erreicht werden kann, ergeben sich sowohl neuartige Möglichkeiten der Koordination von Multi-Agenten-Systemen als auch der Gestaltung elektronischer Marktplätze.

1.2 Koordinationskonzepte der Verteilten Künstlichen Intelligenz (VKI) in elektronischen Märkten

Menschen versuchen seit jeher, Aufgaben, die formalisierbar sind und wiederholt werden können, an Automaten zu delegieren. Der Computer bietet diese Möglichkeit eigentlich nur bei hochstrukturierten und mathematisch beschreibbaren Problemen. Hier hat die Künstliche Intelligenz (KI) Methoden entwickelt, um Menschen mit Software-Agenten zusammenarbeiten zu lassen, so dass diese Agenten im Namen eines Anwenders unstrukturierte Problemstellungen lösen können. Eine Untermenge der KI, die sich speziell mit aus Agenten bestehenden Vielkomponentensystemen beschäftigt, ist das Feld der verteilten künstlichen Intelligenz (VKI, engl. Distributed Artificial Intelligence - DAI).

„DAI is the study, construction, and application of multiagent systems, that is, systems in which several interacting, intelligent agents pursue some set of goals or perform some set of tasks.“ (Weiss 1999, p. 1).

Drei unterschiedliche VKI-Probleme lassen sich auf einem elektronischen Marktplatz, auf dem verschiedene Unternehmen versuchen, mit Hilfe von Verkäufer- und Käuferagenten Handelstransaktionen durchzuführen, finden. Jedes der folgenden drei Szenarien ist durch Unter-

schiede im Verhältnis der Agenten zueinander, ihren Interaktionsfähigkeiten und insbesondere den Interdependenzen zwischen ihren Zielen charakterisiert (Kraus 1997, S. 81):

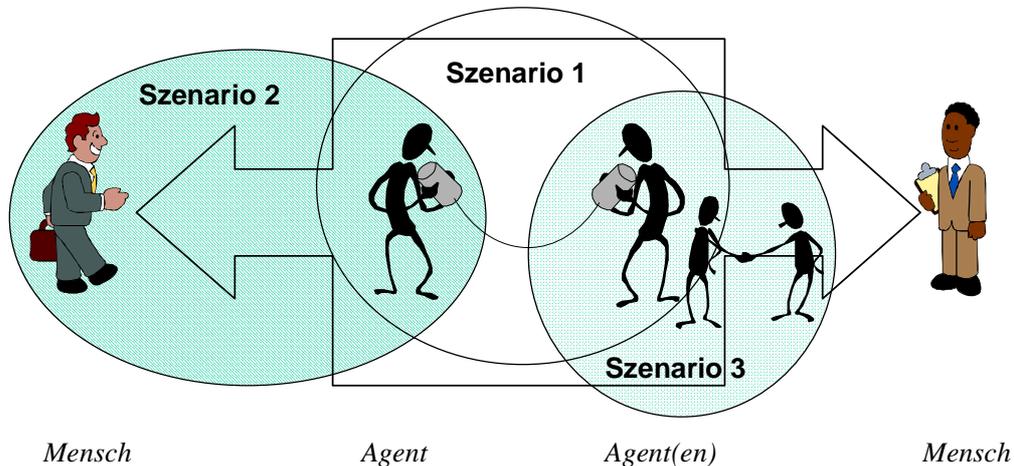


Abb. 1: Einsatzmöglichkeiten von VKI-Methoden auf elektronischen Marktplätzen

1. Szenario 1: Bei der Interaktion zwischen autonomen Softwareagenten, die zu unterschiedlichen Unternehmen gehören, sind die Agenten eigennützig motiviert, können aber von einer Zusammenarbeit profitieren. Die Designer der Agenten verständigen sich über Regeln für die Interaktion, die Zahl der Agenten in jeder einzelnen Interaktion ist begrenzt, die Agenten können miteinander kommunizieren und eigene Berechnungen anstellen.
2. Szenario 2: Ein Verkäuferagent eines Unternehmens versucht, Produkte an einen Menschen zu verkaufen. In diesem Fall wird der Mensch eine nicht strukturierte Interaktion durchführen und es ist schwieriger, die Regeln und Protokolle für eine bestimmte Interaktion im vorhinein festzulegen.
3. Szenario 3: Zwei Agenten des gleichen Unternehmens arbeiten zusammen auf das Ziel hin, den Nutzen (z.B. den Gewinn) des Unternehmens zu verbessern. Die Agenten kooperieren miteinander, Regeln und Protokolle können im vorhinein festgelegt werden, die Zahl der Agenten ist insgesamt begrenzt und sie können eigene Berechnungen anstellen. In diesem Fall gibt es die Möglichkeit, gezielt autonome Softwareagenten zu konstruieren, die in verteilter und kooperativer Weise dieses Problem lösen. Dieser Bereich der VKI wird als Distributed Problem Solving (DPS) bezeichnet. Die meisten Ansätze der VKI zur Koordination in Mehragentensystemen gehen von solchen explizit kooperativen Situationen aus.

Im folgenden wird immer von einer Marktsituation ausgegangen, die dem Szenario 1 entspricht. Die Software-Agenten gehören zu unterschiedlichen Unternehmen, mit jeweils unter-

schiedlichen Zielen und Strategien, die durchaus inkompatibel bzw. antagonistisch sein können (Ferber 1995). Sie führen eine Kauf- oder Verkaufstransaktion dann durch, wenn bei allen beteiligten Agenten dadurch der Nutzen erhöht wird. In diesem Fall kann das VKI-Konzept der Multi-Agenten-Systeme (MAS) angewendet werden.

„In contrast, research in MAS is concerned with the behaviour of a collection of possibly pre-existing autonomous agents aiming at solving a given problem” (Jennings et al. 1998, S. 285).

Charakteristisch für MAS ist, dass (1) jeder Agent nur unvollständige Information über den Zustand der Domäne und der anderen Agenten besitzt und (2) auch nur unvollständige Fähigkeiten zur Lösung des Gesamtproblems beisteuern kann, es (3) keine globale Systemsteuerung gibt, (4) alle Daten dezentralisiert vorliegen und (5) alle Berechnungen in den Agenten parallel, gleichzeitig und asynchron stattfinden. Für das Szenario 1 ist daher die Betrachtungsweise der Multi-Agenten-Systeme erfolgsversprechender als diejenige des Distributed Problem Solving (vgl. dazu auch Kapitel 3.3.2).

Im folgenden Abschnitt soll exemplarisch gezeigt werden, zu welchen Auswirkungen das Zusammenspiel solcher Softwareobjekte führen kann, die autonome Entscheidungen ökonomischer Bedeutung fällen, wenn das Koordinationsproblem falsch verstanden und der Koordinationsmechanismus falsch gewählt wird.

1.3 Eine Fallstudie: Der Schwarze Montag an der Wall Street, 1987

Der Fall des „Schwarzen Montags“ zeigt, welches wirtschaftliche Potential von einem falsch verstandenen Modell eines wirtschaftlichen Gesamtsystems, insbesondere der Fiktion einer kontrollierbaren, stabilen Umwelt, ausgehen kann. Die im einzelnen sinnvollen Eigenschaften der lokalen Softwareobjekten beeinflussten in ihrem Zusammenwirken das Verhalten des Gesamtsystems in negativer und unvorhergesehener Weise.

Am Montag, dem 19. Oktober 1987⁶, vernichtete ein Zusammenbruch der Aktienbörse an der New Yorker Wall Street durch den Kursverlust, über alle US-Aktien gerechnet, 1 Billion US-Dollar an Werten. Der Dow Jones Index fiel absolut um 508 Punkte bzw. relativ zum Kursniveau um 22.6%, der bisher größte Tagesverlust in der Geschichte der Börse. Die Verluste wiederholten sich in unterschiedlich starker Ausprägung weltweit auf allen Aktienmärkten.

Der Anlass des Kursverfalls lag in verschiedenen makroökonomischen Faktoren. Die Erträge für langfristige Anlagen kletterten von 7.6% im Januar 1987 auf etwa 10% im Oktober 1987. Damit stellten sie eine lukrative Alternative zu Aktien dar. Das US-Handelsdefizit stieg und der Wert des US-Dollars begann zu sinken. Nach einer Rede von Finanzminister Jim Baker am Wochenende vor dem 19. Oktober fürchteten Investoren, dass der schwache Dollar zu weiterer Inflation führen würde. Am Montag fiel daraufhin der Dow Jones Index in den ersten anderthalb Stunden des Börsenhandels um 200 Punkte bzw. 9% seines Wertes. Dieser Kursverfall alleine hätte jedoch ohne die Charakteristika der verwendeten Computertechnologie nicht diese katastrophalen Auswirkungen gehabt.

Um eine ständig wachsende Menge an Informationen verarbeiten zu können, hatten die Börsenplätze während der 1980er Jahre Millionenbeträge an US-Dollar in die Installierung elektronischer Netzwerke und komplexer, computerunterstützter Handelsmechanismen investiert. Gleichzeitig fand ein Übergang von zentralen Großrechnern zu Personal Computern und der auf ihnen lokal installierten Software statt. Diese Software wurde programmiert, um Gewinnmöglichkeiten aufzuspüren und um ständig die Werte von Optionen und Futures mit den Preisen der diesen Finanzinstrumenten zugrundeliegenden Aktien zu vergleichen. Unter der Annahme, dass Informationen über Preisänderungen unmittelbar dargestellt wurden, ermutigten sie Investoren, auf mehr als einem Markt simultan zu handeln und minimierten dadurch Arbitragevorteile zwischen den Märkten. Als weiterer Effekt ergab sich, dass Investoren und Händler routinemäßig größere Risiken wagten. Große Anleger, die Vorteile in Sekunden für sich auszunutzen suchten, ließen computerprogrammierte Handelstrategien ausführen, statt menschliche Aktienhändler einzusetzen. Der Gewöhnungseffekt dieser Technologie führte zu einem falschen Sicherheitsverständnis seitens der Investoren, die glaubten, dass die gleichen computergestützten Handelstechniken sie auch in einem fallenden Markt schützen würden.

6 Die Beschreibung der Ereignisse folgt den Informationen aus der "Washington Post" "Stock Market Suffers Largest Loss In History As Dow Industrial Average Drops 508 Points - Plunge Blamed On Anxieties And Computerized Trading" by Peter Behr and David A. Vise, Washington Post, Tuesday, October 20, 1987; Page A01, the "New York Times" New York Times, Tuesday, December 15, 1987. Page 1. The Computer's Contribution to the Rise and Fall of Stocks, by David E. Sanger, "Science" "Computers Amplify Black Monday", Science, 30 October 1987, Volume 238, Number 4827 und von der Internet-Seite von CNNfn.com unter <http://www.cnnfn.com/markets/crash/>.

Zu diesem Zweck verwendeten die meisten institutionellen Anleger verschiedene computergestützte Portfolio-Absicherungsprogramme („Portfolio Insurance“). Diese Programme begannen automatisch Aktien zu verkaufen, wenn die Preise unter eine bestimmte Grenze fielen und verstärkten diese Bemühungen bei weiter fallenden Kursen. Daneben versuchten Wertpapierarbitrage-Programme („Index Arbitrage“), durch gleichzeitiges Verkaufen von Aktien auf einem Wertpapiermarkt (am Börsenplatz New York) und Kaufen von Aktienoptionen auf einem Optionsmarkt (am Börsenplatz Chicago) Arbitragegewinne zu sichern. Da der Aktienmarkt an diesem Tag jedoch keine Liquidität hatte, um Aktien zu kaufen, wurden die Aktienkurse nicht gestützt und der Markt fiel weiter. Prospektive Käufer hingegen warteten, da sie davon ausgehen konnten, dass die Kurse weiter fielen.

Befürworter der Portfolio-Absicherungsprogramme hatten vor dem Börsencrash behauptet, dass Verluste nicht über 5% des Portfoliowertes hinausgehen könnten. Diese Verkaufsversprechungen hatten gewirkt und im Oktober 1987 waren zwischen 70 und 90 Milliarden US-Dollar in Fonds angelegt, die mit Portfolio-Absicherungsprogrammen arbeiteten. Manche Investoren verwarfen bekannte Sicherungstechniken, wie z.B. das Umwandeln eines Teils ihrer Aktien in Bargeld, da sie zuversichtlich waren, dass die Programme arbeiten würden.

Die Funktionsweise der Börse, die in den Programmen nachgebildet worden war, beruhte im wesentlichen auf zwei Annahmen, die sich als falsch erwiesen. Eine Annahme war, dass die Märkte eine Art von „Wohlverhalten“ zeigten, d.h. dass die Aktienkurse und die Optionspreiskurse sich zeitnah gegenseitig verfolgen würden. Eine andere Annahme war, dass zum Zeitpunkt des Vorschlags eines einzelnen Kauf- oder Verkaufsgebotes genügend Käufer bzw. Verkäufer vorhanden wären.

Aktienbörsen sind normalerweise hochliquide Märkte, vor allem wegen der Finanzintermediäre, die eine Ausführung der Transaktionen garantieren, indem sie zwischen Käufern und Verkäufern stehen. Börsenspezialisten z.B. kaufen auch in fallenden Märkten und stützen damit implizit die Kurse. Dieses System arbeitet sehr gut, um Liquidität zu gewährleisten, wenn die Zahl der Kauf- und Verkaufsaufträge nicht zu weit auseinander ist. Normalerweise reicht eine kleine Menge von Beständen relativ zu den Auftragseingängen aus, um die Lücke zwischen Kauf- und Verkaufsaufträgen zu schließen. Die Zahl der Verkaufsaufträge am 19. Oktober 1987 überstieg die Zahl der Kaufaufträge bei weitem und die Aktienkurse in New York und die Optionskurse in Chicago zeigten deutliche Differenzen.

In dieser Situation war es nicht mehr möglich, diese Entwicklung von einer zentralen Stelle aus anzuhalten, da es keine einzelne Institution mit einem derartigen Einfluß gab. Am Abend des 19. Oktober wurde z.B. in New York von der Börsenaufsicht erwogen, das System einfach abzuschalten. Zu dieser Zeit hatte sich der Börsencrash in seiner Tendenz bereits auf die ver-

bundenen Handelsplätze in Chicago und weltweit ausgedehnt und eine verstärkte Rückkopplung auf New York bewirkt. Gerade wegen dieser Überlegungen fürchteten Investoren, dass sie auf ihren Aktien „sitzen bleiben“ würden. Die Finanzintermediäre führten daher am Tagesende wesentlich größere Bestände als üblich. Für diese Aktien mußten sie laut Börsenregeln innerhalb von fünf Geschäftstagen zahlen. Die dafür notwendigen Kredite blieben aus, weil die Solvenz der Intermediäre bezweifelt wurde. Die Banken wollten kein Geld mehr verleihen, da der Verfall der Wertpapierkurse die Kreditwürdigkeit aller Beteiligten bereits beeinträchtigt hatte. Die Zurückhaltung der Banken und anderer Kreditgeber, weitere Kredite zu geben, erhöhte wiederum das Risiko, dass sich die Erwartungen über einen Marktzusammenbruch bewahrheiten würden.

Technisch gesehen verstärkte der Einsatz der Software-Mechanismen „Portfolioabsicherung“ und „Wertpapierarbitrage“ die beginnende Entwicklung auf dem Aktienmarkt, da Aktien und Optionen kombiniert gehandelt wurden. Wenn der Preis der Aktien in New York von dem Preis der entsprechenden Optionen in Chicago unterschiedlich war, konnten Investoren Gewinne sichern, indem sie gleichzeitig Aktien verkauften und Optionen auf diese Aktien kauften. Das Problem lag schlußendlich in den Marktmechanismen selbst, die ein Entflechten der Optionsmärkte in Chicago und der Aktienmärkte in New York verhinderten. Nach dem initialen Kursverlust von 200 Punkten setzte eine programmierte Wertpapierarbitrage ein, die von den Portfolioabsicherung-Programmen verstärkt wurde. Da letztere umso mehr Aktien verkaufen wollten, je tiefer der Kurs fiel, wurden immer mehr Aktien verkauft, bis der Markt keine Liquidität mehr bereitstellen konnte. Diejenigen Käufer, die diesen Mechanismus durchschauten, warteten ohnehin auf ein weiteres Fallen der Preise.

Eine Untersuchungskommission der „Securities Exchange Commission“ (SEC) kam später zu dem Urteil, dass kein einzelnes Ereignis als Ursache des Börsenzusammenbruchs isoliert werden konnte. Insgesamt wurde die Schuld dem Zusammenwirken zwischen den Menschen und den Computerprogrammen gegeben. Im Sinne dieser Arbeit lassen sich folgende Resultate aus den Ereignissen von 1987 festhalten:

1. Es gab keine technischen Probleme. Alle dezentralen Systeme arbeiteten zuverlässig. Der Ablauf des Börsenzusammenbruchs wurde nicht beeinflusst durch Programmfehler, technische Probleme oder Sicherheitsprobleme – es gab keinen Einbruch, keine Fälschung von Information und keinen Verlust von Kommunikationskanälen. Die Zahl der angeschlossenen Käufer und Verkäufer blieb in dem geschlossenen Netzwerk konstant.
2. Das schwächste Glied der zeitnahen Verarbeitung aller Daten war das zentrale Marktinformationssystem. Die Aufgabe, die ökonomische Koordination zwischen einer großen Zahl relativ langsam agierender menschlicher Agenten zu unterstützen, konnte relativ gut gelöst

werden. Mit zunehmendem Einsatz dezentraler Informationsverarbeitung, d.h. der Ersetzung menschlicher Agenten durch Software-Programme, konnten die einzelnen Teilnehmer schneller agieren, als der zentral implementierte Marktmechanismus den gegenwärtigen Informationsstand verarbeiten und seine vollständige Information in Form von Aktienkursen ausgeben konnte.

3. Die Rückkoppelung zwischen den Märkten und ihrer relevanten Umwelt wurde von den einfachen, regelbasierten Programmen (und ihren Programmierern) nicht erkannt. Das regelbasierte und nicht adaptive Verhalten der Portfolioabsicherungs- und Wertpapierarbitrage-Programme bewirkte in der Summe genau diejenigen Probleme, die ihnen schließlich zum Verhängnis wurden. Es kann behauptet werden, dass das adaptive, komplexere Verhalten menschlicher Aktienhändler an dieser Stelle eine Abschwächung der Kursentwicklung bewirkt hätte.

Das Problem, das zum Schwarzen Montag führte, ist prinzipiell bis heute nicht gelöst. Schnelle Handelstransaktionen, deren Auswirkungen in der Gesamtheit a priori kaum erkannt werden können, werden die Regel und nicht die Ausnahme sein. Die offene Architektur der GII mit ihrer variierenden Anzahl von Teilnehmern und den dazugehörigen Sicherheitsproblemen ist fundamental verschieden von den geschlossenen Informationssystemen von gestern. Trotzdem werden diese einzelnen Systeme als Diensteanbieter und -nachfrager, ohne Änderung in ihrer Arbeitsweise, in zunehmendem Maße miteinander vernetzt; ein spiegelbildliches Vorgehen zur Vernetzung der Aktienmärkte, jedoch in einer weit größeren Dimension.

1.4 Folgen für die Gestaltung marktlicher Koordinationsmechanismen

Gerade im Bereich des Electronic Commerce werden sich voraussichtlich Möglichkeiten zur den Börsensystemen vergleichbaren Koordination und Verhandlung zwischen vernetzten autonomen Systemen ergeben. Solche Anwendungsgebiete können z.B. elektronische Märkte in der Versorgungsindustrie (z.B. Elektrizität, Wasser, Erdgas), in der Telekommunikation (Bandbreitenhandel), oder in der Beschaffung von Verbrauchs- und Hilfsmaterialien sein. Auch die Produktionssteuerung in solchen Bereichen, in denen die Zwischenprodukte eindeutig definiert werden können, gehört dazu. Die Koordinationsmechanismen für diese Anwendungsfelder werden zuerst von den Aktienbörsen übernommen, weil diese bereits vorhanden und ausreichend detailliert beschrieben sind⁷. Damit werden jedoch auch ähnliche Risiken übernommen, die im dem vorhergehenden Abschnitt beschrieben worden sind.

Der Vorteil der Software-Agenten im Vergleich zu menschlichen Marktteilnehmern, dass sie mit vielen potentiellen Handelspartnern gleichzeitig in automatisierte Verhandlungen treten, einen nutzenverbessernden Preis erreichen oder einen vom Prinzipal nur noch zu bestätigenden Vertrag aufsetzen können und dies alles innerhalb von Millisekunden (Preist 1998, Guttman et. al. 1998). Diese Möglichkeiten motivieren auch den Einsatz des in dieser Arbeit zu entwickelnden Marktmechanismus. Erste Anwendungsdomänen in diesem Bereich werden voraussichtlich durch folgende Eigenschaften charakterisiert werden können (Sierra 2000, S.10):

1. Die Verhandlungsabläufe zeichnen sich durch eine sehr hohe Geschwindigkeit aus. Als Beispiel seien Aushandlungen bezüglich der Kommunikationsbandbreite in paketbasierten Netzwerken angeführt (vgl. Gibney et al. 1999), die so schnell ablaufen werden, dass eine Rückmeldung an einen menschlichen Benutzer zwischen den einzelnen Verhandlungsrunden nicht möglich sein wird.
2. Die Interaktionen zwischen den Agenten werden häufig wiederholt. Dadurch kann ein Lernverhalten in den Agenten aus der Erfahrung früherer Interaktionen gespeist werden, sofern die Komplexität der Anwendungsdomäne begrenzt ist. Gleichzeitig ergeben sich Größenvorteile in Bezug auf den Einsatz von Software, Hardware oder Training der Agenten. Insbesondere letzteres ist von hoher Bedeutung, da als größter Kostenanteil des Einsatzes von Agenten die explizite Eingabe von Anwenderpräferenzen viel Zeit kosten wird.
3. Transaktionen werden nur einen relativ geringen Wert aufweisen. Dadurch kann der Prozess überwacht und der automatisierte Handel nach einer kurzen Zeit gestoppt werden, ohne dass signifikante Verluste eingetreten sind. Bei geringwertigen Transaktionen ist der Anteil der Transaktionskosten an den Güterkosten außerdem relativ hoch, was den Übergang von menschlichem auf agentenbasiertes Handeln mit geringen Transaktionskosten zusätzlich motiviert.
4. Die gehandelten Wirtschaftsgüter sind mit wenigen Verhandlungsattributen charakterisierbar. Wenn zwei Objekte nur über den Preis unterschieden werden können (sogenannte *Commodities*), so ist es möglich, eine Entscheidung über eine Kauftransaktion anhand dieser einen Preisdimension zu fällen. Je mehr Verhandlungsattribute existieren, desto komplexer muß die Entscheidungslogik sein. Zusätzlich treten zwischen den Attributen kombinatorische Probleme auf.

7 Ein Beispiel für diese Vorgehensweise stellt die Elektrizitäts- und Energiebörse EEX in Frankfurt am Main dar (www.eex.de). Das dort verwendete MAS ENTRAS ist ohne wesentliche Änderungen an seiner Arbeitsweise von dem MAS AMTRAS abgeleitet, welches für den Einsatz in einem Rentenfinanzmarkt entwickelt wurde (vgl. Budimir/Gomber 1999).

Um die Chancen und Risiken der Gestaltung agentenbasierter elektronischer Märkte beurteilen zu können, ist sowohl ein besseres Verständnis des Koordinationsproblems als auch eine darauf aufbauende verbesserte Gestaltung des Koordinationsmechanismus vonnöten. Diese beiden Aspekte werden in den Abschnitten 1.6 und 1.7 detaillierter ausgeführt. Zunächst soll jedoch der Begriff der Koordination definiert und eingeordnet werden.

1.5 Koordination im Schichtenmodell der Telematik

Um mögliche Realisierungen von Koordinationsmechanismen in der zutreffenden Abstraktionsebene einordnen zu können, wird das Freiburger Schichtenmodell der Telematik genutzt. Es unterteilt Anwendungen in verteilten Netzen in einen informationstechnischen und einen nichttechnischen, organisatorischen Kontext. Das Verhältnis der informationstechnischen Schichten zu den nichttechnischen, organisatorischen Schichten kann an einem „nach oben“ erweiterten allgemeinen Modell eines technischen Netzwerkprotokolls veranschaulicht werden. Grundidee ist es, das Konzept der Schichtung in Netzwerkprotokollen aufzunehmen und nichttechnische Schichten einzuführen, die die Verwendung der Informationstechnik betreffen (Schoder et al. 1997, S. 282).

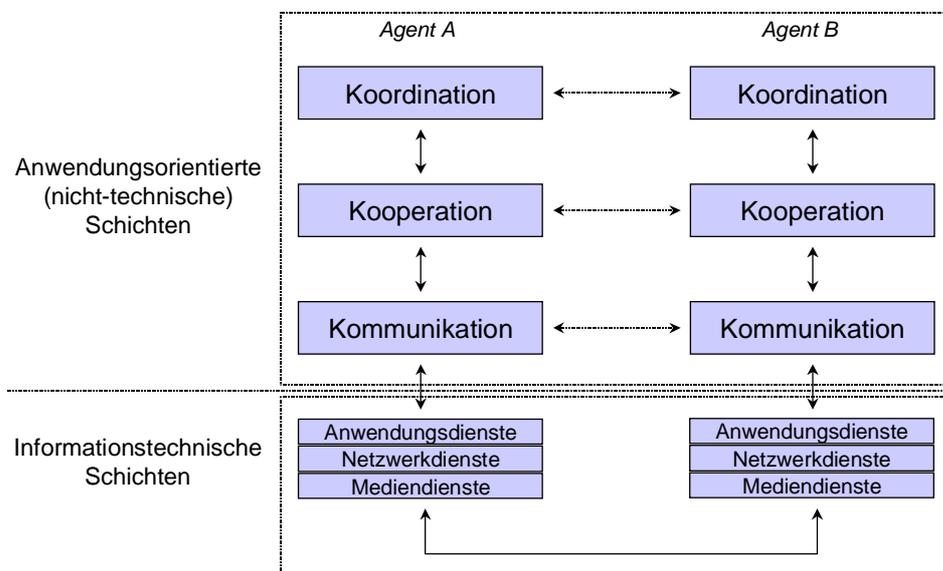


Abb. 2: Schichtenmodell der Telematik (Müller et al. 1997, S. 41)

Die Schichtung des als Vorbild fungierenden ISO-OSI Referenzmodells wendet gleichberechtigt zur Abstraktion der Aufgaben und Dienste zwei Prinzipien an. Das technische oder Defini-

tionsprinzip basiert auf einer formalen Definition von Schnittstellen, Diensten und Protokollen. Das zweite, das Dienstleistungsprinzip, definiert eine Arbeitsteilungsstrategie und legt so die Reihenfolge bei der Erbringung einer Dienstleistung fest. Die Anordnung der Schichten erfolgt entsprechend ihrer Fähigkeiten, „Dienstleistungen“ für mehrwertige Aufgaben zu erbringen. Die Funktionalität einer Schicht kapselt die Funktionalität aller darunterliegenden Schichten und verbirgt deren konkrete Implementation vor den höheren Schichten. Vor dem Hintergrund dieser Sichtweise lassen sich Aspekte der informationstechnischen Unterstützung von Kooperationsprozessen und Koordinationsaufgaben in einen durchgängigen Kontext zusammenfassen. Die Grundidee ist, das Konzept und die Vorteile der Schichtung der Rechnerkommunikation als Beschreibungsprinzip für die nicht-technischen Schichten einzuführen (Müller et al. 1997, S.41).

Ausgangspunkt des organisatorischen Kontexts ist die Abgrenzung des Koordinationsproblems als die gesamtzielkonforme Abstimmung interdependenter Teilaufgaben durch die beteiligten Aufgabenträger. Diese Abstimmung erfolgt in Kooperationsprozessen, die wiederum über Interaktionen zwischen den betroffenen Individuen ablaufen. Eine Interaktion ist als ein Kommunikationsprozess im organisatorischen Kontext anzusehen, in dem Informationen unter den Beteiligten ausgetauscht werden. Die Informationstechnik kann Informationen lediglich im Rahmen technischer Kommunikationsprozesse als Daten zwischen den Knoten eines Kommunikationsnetzwerks übertragen (Müller et al. 1997, S. 41ff.) Die Trennung und Funktionalität der nichttechnischen Schichten kann aufgrund der Vieldimensionalität der betriebswirtschaftlichen Realität nicht so scharf abgegrenzt geschehen, wie in den technischen Referenzmodellen, z.B. dem ISO-OSI-Modell (Effelsberg/Fleischmann 1986). Trotzdem ist diese Darstellung als komplexitätsreduzierendes Beschreibungsmodell nützlich, auch wenn die Realisierung der nichttechnischen Schichten in so starkem Maße vom konkreten Anwendungsfeld abhängt, dass eine Generalisierung nur auf der hier vorgestellten abstrakten Ebene sinnvoll erscheint. Eine Diskussion der Begriffe „Koordination“, „Kooperation“ und „Kommunikation“ im Zusammenhang betriebswirtschaftlicher Anwendungsfelder findet sich ausführlich bei Hummel (1995, S. 37ff.) oder Müller et al. (1997, S. 39ff). An dieser Stelle sollen nur die für diese Arbeit wesentlichen Ergebnisse zusammengefasst werden.

Das betriebswirtschaftliche Ziel der Unternehmung, die Gewinnmaximierung, kann im Allgemeinen nur durch Arbeitsteilung von Akteuren erreicht werden. Die Gesamtaufgabe der Zielerreichung wird durch Erfüllung von Teilaufgaben durch die Akteure, sowie die Synthetisierung ihrer Teilergebnisse, ausgeführt. Die Durchführung der Teilaufgaben wiederum bedarf der Abstimmung zwischen den Akteuren, sei es bezüglich der zeitlichen Reihenfolge oder der Auflösung von Zugriffskonflikten auf gemeinsame Ressourcen: *“ensuring the collaborative actions of the individuals working on a shared set of decisions are coordinated to achieve the*

desired result efficiently” (Klein 1997, S. 161). **Koordination** läßt sich vor diesem Hintergrund als die „gesamtzielkonforme Abstimmung interdependenter Teilaufgaben durch die beteiligten Aufgabenträger“ untereinander definieren (Laßmann 1992, S. 2), wobei diese zentral oder dezentral erfolgen kann. Auf die Fragestellung der zentralen oder dezentralen Koordination wird später noch ausführlich eingegangen.

„Die konkrete Interaktion zweier oder mehrerer Aufgabenträger zum Zweck der Abstimmung und Durchführung von Teilaufgaben wird hier als **Kooperation** bezeichnet⁸“ (Hummel 1995, S.10). Die Ausgestaltung der Kooperation hängt in hohem Maße von der hierarchischen Position der Aufgabenträger zueinander ab. Im Fall der direkten Über- oder Unterordnung zwischen den Aufgabenträgern bestimmt die Abstimmung im Extremfall aus der Anweisung zu zielkonformen Verhalten. Besteht keine solche Weisungsbefugnis, ergibt sich erneut ein Koordinationsproblem in der Abstimmung zwischen den Teilaufgaben, welches generell durch das Eingehen von Konzessionen oder das argumentative Überzeugen gelöst werden kann (Pruitt 1981, S. 91). Wie auch bei der Arbeit von Hummel wird im Folgenden der Fall „lateralen Kooperation“ (Bleicher 1991, S. 148), d.h. ohne hierarchische Ordnung betrachtet.

„Die Besonderheit dieser “lateralen” Kooperation liegt in der horizontalen, funktionsübergreifenden Zusammenarbeit zwischen Gruppen oder Individuen, wobei zwischen den betroffenen Akteuren keine formalen Weisungsbeziehungen bestehen. In dieser Situation ist die (aus kooperativen Prozessen resultierende) übergeordnete Koordinationsleistung von keiner Stelle einforderbar, sie muß sich vielmehr aus der von den betroffenen Akteuren dezentral gefällten Entscheidung zur Kommunikation, Kooperation und schließlich Koordination eigenständig ergeben.“ (Schoder et al. 1997, S. 281)

Jegliche Abstimmung und Durchführung von Teilaufgaben erfordert den Informationsaustausch zwischen den Kooperationspartnern und somit Kommunikationsprozesse (Hummel 1995, S.12). Auch zum Begriff der **Kommunikation** gibt es unterschiedliche Definitionen, von denen im folgenden die reine Übertragung von Informationen, d.h. *“the exchange of thoughts, messages, or information, as by speech, signals, writing, or behavior.”* (American Heritage 1996) als für die Zwecke dieser Arbeit ausreichend angesehen wird. Die semantische Interpretation der übertragenen Informationen findet an der Schnittstelle zwischen Kommunikation und Kooperation statt und wird hier nicht weiter problematisiert. Es sei darauf hingewiesen, dass der physische Transport von Nachrichten auf der technischen Ebene nicht das

⁸ „vgl. zu einer eingehenden Diskussion des Kooperationsbegriffs Klimecki (1984), S. 71ff.“ (Hummel 1995, S. 10).

gleiche ist, wie die Übertragung von bedeutungsvollen Inhalten auf der nichttechnischen Ebene: „die Technik selbst ist vollkommen neutral, erst die konkrete Verwendung konstituiert ihren Zweck“ (Hummel 1995, S. 39).

1.6 Das Koordinationsproblem des elektronischen Marktplatzes

Das Koordinationsproblem eines elektronischen Marktplatzes lässt sich wie folgt charakterisieren:

In einem elektronischen Marktplatz handeln menschliche Teilnehmer miteinander unter Nutzung computergestützter Kommunikationsmedien. Durch bilaterale Kooperation (Güter- bzw. Werteübergang in Handelstransaktionen) versuchen sie, im Zeitablauf ihren Nutzen zu maximieren. Dazu müssen sie aus einer Menge an alternativen Entscheidungen im Markt auswählen, die Verwendung, Kauf oder Verkauf eigener und fremder Ressourcen (Güter und/oder Geld) betreffen. Menschen werden in Zukunft diese Entscheidungsaufgabe an autonome Softwareobjekte delegieren, um ihre eigene Zeit besser nutzen zu können und ihre Transaktionskosten zu senken. Diese Softwareobjekte (Agenten) werden von ihren menschlichen Besitzern (Prinzipalen) mit Informationen über ihre vorhandenen Ressourcen, den anzustrebenden Zielzustand, sowie Regeln und Strategien über den dynamischen Vorgang der Erreichung des Zielzustandes ausgestattet. Damit die Agenten miteinander interagieren zu können, haben sich die Prinzipale im Vorhinein auf eine technische Basis sowie einen Kommunikationsstandard geeinigt.

Der Marktplatz⁹ ist ein technischer Ort, der von den Teilnehmern zu erfüllende Anforderungen an Technik und Kommunikationsformen stellt. Der Marktplatz implementiert zum einen die technische Marktplattform, zum anderen den Marktmechanismus als Regeln und Verfahrensweisen, die zu einer veränderten Ressourcenallokation führen. Zur Laufzeit treten auf dem Marktplatz Agenten mit einer definierten Ausstattung an Ressourcen und Präferenzen auf. Die Marktplätze sind prinzipiell offen für neue Agenten unbekannter Teilnehmer, da diese neue

⁹ Zur Abgrenzung der Begriffe Markt, Marktplatz und Marktplattform ist folgendes festzustellen. Ein Markt ist der abstrakte Ort des Tausches. Ein Marktplatz ist der konkrete Ort, an dem Nachfrager und Anbieter einander treffen. Eine Marktplattform ist eine technische Realisierung eines Marktplatzes in Form einer Softwareimplementation, über die Anbieter und Nachfrager über telematische Medien kommunizieren und kooperieren. Da im folgenden fast immer eine technische Realisierung besprochen wird, sind Marktplatz und Marktplattform in dieser Arbeit synonym.

Marktchancen und eine erhöhte Liquidität darstellen. Die Zahl der Agenten auf einem Marktplatz ist weder nach oben noch nach unten begrenzt; dadurch ergibt sich eine dynamische Population an Marktteilnehmern. Es ist möglich, dass Agenten zwischen Marktplätzen, auf denen gleiche Güter gehandelt werden, wechseln oder der Prinzipal mehrere Agenten auf verschiedenen Marktplätzen gleichzeitig nutzt. Auf diese Weise werden auch Marktinformationen zwischen den Marktplätzen weitergegeben.

Jeder Marktplatzbetreiber kann zwischen unterschiedlichen Marktmechanismen wählen, z.B. verschiedene Auktionsverfahren. Der Marktplatzbetreiber erzielt einen Gewinn aus Zutrittsgebühren, Transaktionsgebühren, oder einer Kombination beider und hat daher ein eigenes Interesse an der Durchführung von möglichst vielen Transaktionen auf seinem Marktplatz. Seine Strategie und seine Handlungsalternativen sind jedoch nicht Gegenstand dieser Betrachtungen. Insofern wird der Marktplatz als existent angenommen und sein Einkommen ist nicht beeinflussbar durch das Verhalten der auf diesem handelnden Software-Agenten. Der Marktplatzbetreiber kann eine die Koordination unterstützende zentrale Instanz aufstellen, für deren Nutzung diese Transaktionsgebühren fällig werden. Im einfachsten Fall handelt es sich um einen Verzeichnisdienst, damit die Software-Agenten sich finden und miteinander kommunizieren können. Im kompliziertesten Fall wird der komplette Steuerungsmechanismus für den Marktplatz und alle seine Teilnehmer in dieser Instanz konzentriert. Da es sich um ein offenes System handelt, kann es mehrere Marktplätze geben, die zueinander in Konkurrenz stehen. Der einzelne Software-Agent hat prinzipiell die Möglichkeit, eine vorhandene zentrale Koordinationsinstanz zu nutzen, er kann diese aber auch umgehen und direkt mit einem anderen Software-Agenten kommunizieren.

Jeder Agent besitzt in diesem Modell eine Startverteilung an Ressourcen, sowie eine ebensolche Zielverteilung, die sich von der Startverteilung unterscheidet und die er durch Teilnahme am Markt erreichen möchte. Die Zielverteilung stellt dabei für ihn einen höheren Nutzen dar als die Startverteilung. Um von der Startverteilung zur Zielverteilung zu gelangen, kooperiert ein Agent für die Dauer einer Gütertransaktion mit einem anderen Agenten, wodurch Geld und Güter ausgetauscht werden und daher für beide Agenten eine neue Ressourcenverteilung entsteht. Eine Transaktion (ein Ressourcenübergang) wird dann durchgeführt, wenn dadurch für die beteiligten Agenten eine Nutzenerhöhung stattfindet¹⁰. Die Entscheidung darüber, ob eine derartige Transaktion durchgeführt werden kann, wird in hohem Maße von dem Zustand und den Präferenzen des prospektiven Partners beeinflusst. Durch dieses Vorgehen verändert sich

10 Damit wird ein benevolentes Handeln der Agenten ausgeschlossen. Möglich ist dennoch ein Handeln auf unterschiedlichen Nutzenwertskalen, d.h. es muß nicht beiden Agenten gleichermaßen um Geldgewinne gehen.

durch die Durchführung von Transaktionen ständig die Angebots- und Nachfragesituation auf dem Markt. Die Agenten müssen an allen Marktphasen teilnehmen und in jeder Situation notwendige Entscheidungen selber treffen können, ohne ihre Prinzipale rückzufragen¹¹. Dazu benötigen sie eine eigene Informationsverarbeitungskapazität sowie die Fähigkeit, aus ihren in der Vergangenheit erzielten Ergebnissen lernen zu können, d.h. sie müssen einerseits diese messen und andererseits ihre Strategien anpassen können. Diese letztere Fähigkeit führt wiederum zu einer starken Veränderung der Marktsituation, da die gleiche Entscheidungssituation durch denselben Agenten im Zeitablauf unterschiedlich beurteilt werden kann.

Jeder Software-Agent erhält Informationen von seinem Prinzipal, die ihn zu zielgerichtetem Handeln befähigen. Dazu gehören eine ständig aktualisierte Information über die vorhandenen Ressourcen, eine Zielvorgabe (abstrakt als Nutzenfunktion, praktisch in Form von Mengen- und Preisgrenzen einzukaufender oder zu verkaufender Güter), sowie die Formalisierung des strategischen Verhaltens (im Sinne der Kooperationsdefinition des vorhergehenden Kapitels). Diese Informationen sind für jeden Prinzipal und damit auch für den Software-Agenten privat. Eine Veröffentlichung dieser Informationen durch Dritte (auch durch den Marktplatzbetreiber), so dass andere Agenten diese einsehen können, führt zu der Möglichkeit der ökonomischen Ausnutzung des Software-Agenten, ist stark nutzensenkend und führt u.U. zur Nichtteilnahme des Agenten am Markt. Es muß daher sichergestellt werden, dass kein anderer Marktteilnehmer die Informationen des Agenten auslesen kann. Falls der Agent eine Entscheidung nicht selber treffen kann oder können soll, kann der Prinzipal während der Lebenszeit des Software-Agenten Ziele oder Strategien zu verändern, oder eine adaptive Prozedur spezifizieren, die Ziele oder Strategien verändert, ohne dass er direkt eingreifen muß. Außerdem können erhaltene Ressourcen konsumiert oder durch Produktionsprozesse in andere Güter umgewandelt werden.

Die Ausgangssituation des Koordinationsproblems wird zusammenfassend durch folgende Eigenschaften charakterisiert, die hier als Annahmen postuliert werden, da sie stark von dem betrachteten Anwendungsfall und der verfügbaren Technologie abhängen:

1. **Annahme allgegenwärtiger (ubiquitärer) Kooperation:** In der zukünftigen GII werden zunehmend komplexere Problemstellungen von vielen, miteinander vernetzten Akteuren bearbeitet werden. Das zur Problemlösung notwendige Wissen erfordert Spezialisierung. Jeder Akteur geht davon aus, dass es jederzeit möglich ist, mit anderen Akteuren (und deren Problemlösungswissen) zu kooperieren, um eine dem Einzelnen nicht mögliche Problemlö-

11 Tatsächlich sind agentenbasierte elektronische Märkte heute so angelegt, dass Agenten vor jeder Entscheidung bei den Prinzipalen rückfragen. Eine echte Transaktionskostenminimierung ergibt sich für den Prinzipal erst, wenn die Agenten im Rahmen ihrer Vorgaben vollkommen selbständig entscheiden und handeln.

sung zu erreichen. Intermediäre können umgangen werden und werden nur in Anspruch genommen, wenn sie ökonomisch rational sind, d.h. einen besseren Nutzen gewährleisten. Intermediäre sind selbst wirtschaftlichen Zwängen unterworfen und konkurrieren vielleicht sogar miteinander. Damit kann nicht von vornherein von einer Pflicht zur Teilnahme an einem zentralen Koordinationsmechanismus ausgegangen werden.

2. **Annahme nicht-benevolenten Handelns:** Die vernetzten Akteure stehen in keiner hierarchischen Beziehung zueinander. Ihr jeweiliges Ziel ist es, ihren individuellen Nutzen zu maximieren. Eine Kooperationsbeziehung wird nur dann eingegangen, wenn die beteiligten Partner sich von der Erreichung des Kooperationszieles auch eine Erhöhung des eigenen Nutzens versprechen. Dies gilt auch und gerade bei der Inanspruchnahme Dritter, z.B. Marktintermediäre. Eine Kooperation zum alleinigen Nutzen des Partners oder einer dritten Partei (benevolentes Handeln) findet nicht statt. Ein wichtiger Parameter der Strategie der Agenten ist die individuelle Bereitwilligkeit zur nutzenverbessernden Kooperation mit anderen, a priori unbekanntem Agenten, die während der Laufzeit in Marktplätze eintreten oder diese verlassen.
3. **Annahme begrenzter Rationalität:** Die Veränderung von Zuständen der Ressourcenallokation wird im wesentlichen durch Entscheidungen strategisch handelnder Software-Agenten bewirkt, deren Strategie nicht öffentlich ist. Die Berechnung der „besten Alternative“ (im Sinne maximalen Nutzenzuwachses) kann durch den Agenten nicht eindeutig durchgeführt werden. Daran ist zum einen die Rekursion schuld, die das prognostizierte Verhalten des Partners jeweils zu einer wesentlichen Ursache für das eigene Verhalten macht, zum anderen die Notwendigkeit, sämtliche Informationen der Umwelt in eine derartige Optimierung einbeziehen zu müssen. Die Agenten müssen daher mit unvollständiger Information über die Durchführung einer Transaktion entscheiden. Ohne Ergebnisse zu einem Optimum in Beziehung setzen zu können, bleibt ihnen nur die Entscheidung auf der Grundlage des Überschreitens von subjektiven Zufriedenheitsniveaus.

Diese Charakteristika des grundlegenden Koordinationsproblems determinieren die Auswahl von Koordinationsmechanismen, die diese Annahmen nicht verletzen dürfen. Der Einsatz eines Koordinationsmechanismus, der auf benevolentem Verhalten basiert, kann bspw. an sogenannten Trittbrettfahrerproblemen (Varian 1991, S. 547f.) scheitern, wie Hummel (1995) im Kontext eines Gruppenterminkalenders oder Adar/Huberman (2000) anhand des Filesharingprogrammes GNUTELLA zeigen konnten.

1.7 Anforderungen an den Koordinationsmechanismus

Um das Koordinationsproblem einer Lösung zuzuführen, wird ein abstraktes Regelwerk zur Durchführung der Koordination benötigt. Dieses erlaubt es in klassischen Märkten menschlichen Marktteilnehmern, aus einer Vielzahl möglicher Aktionen diejenigen auszuwählen, die mit großer Wahrscheinlichkeit zu einem Ergebnis führen. Für den vergleichbaren Einsatz von Software-Agenten in elektronischen Märkten benötigt man eine maschinelle Realisierung dieses Regelwerks - einen Koordinationsmechanismus.

Ziel der Arbeit ist es vor diesem Hintergrund, einen Koordinationsmechanismus für elektronische Märkte zu entwickeln, welcher es den einzelnen Agenten ermöglicht, planmäßig eine nutzenverbessernde Ressourcenallokation anzustreben. Diese Märkte bestehen aus von vielen menschlichen Anwendern erstellten Software-Agenten, die individualistisches und informationsstrategisches Verhalten, und deren Zusammensetzung häufig wechselt. Die Realisierung eines solchen Koordinationsmechanismus stellt eine beidseitige Verknüpfung zwischen ökonomischer Analyse und technischer Implementierung her, wie in Abbildung 3 dargestellt ist. Auf der *Ebene der ökonomischen Analyse* konkurrieren verschiedene Erklärungsansätze des Phänomens Markt miteinander, die in Kapitel 3 diskutiert werden. Abhängig davon ergeben sich auf der *Ebene der technischen Implementierung* alternative Möglichkeiten der Formalisierung, die mit bestehender Technologie (siehe Kapitel 2) besser oder schlechter umgesetzt werden können.

Aus dem Ziel dieser Arbeit folgen als angestrebte Ergebnisse:

1. **Ein dezentraler Koordinationsmechanismus für Multi-Agenten-Systeme.** Die Möglichkeiten einer Marktökonomie, die konflikthafte Bedürfnisse von Millionen menschlicher Agenten zu befriedigen, empfiehlt sie als ein Prinzip dezentraler Koordination auch für technische Vielkomponentensysteme (Miller/Drexler 1988, Huberman 1988, Clearwater 1995, Wellman 1996). In der Darstellung von Abbildung 3 entspricht dies der Pfeilrichtung von oben nach unten.
1. **Ein Werkzeug zur dynamischen Untersuchung dezentraler Marktkoordination.** Eine formale Beschreibung eines Koordinationsalgorithmus „Markt“ ist bisher von der Volkswirtschaftslehre nicht eindeutig geliefert worden, so dass eine widerspruchsfreie Implementation nicht existieren kann. Die in dieser Arbeit vorgestellte Realisierung einer dezentralen marktlichen Koordination stellt damit auch ein Werkzeug dar, um durch Messungen und Analysen marktliche Koordination zu erforschen. In Abbildung 3 wird dieser Aspekt durch die Pfeilrichtung von unten nach oben symbolisiert.

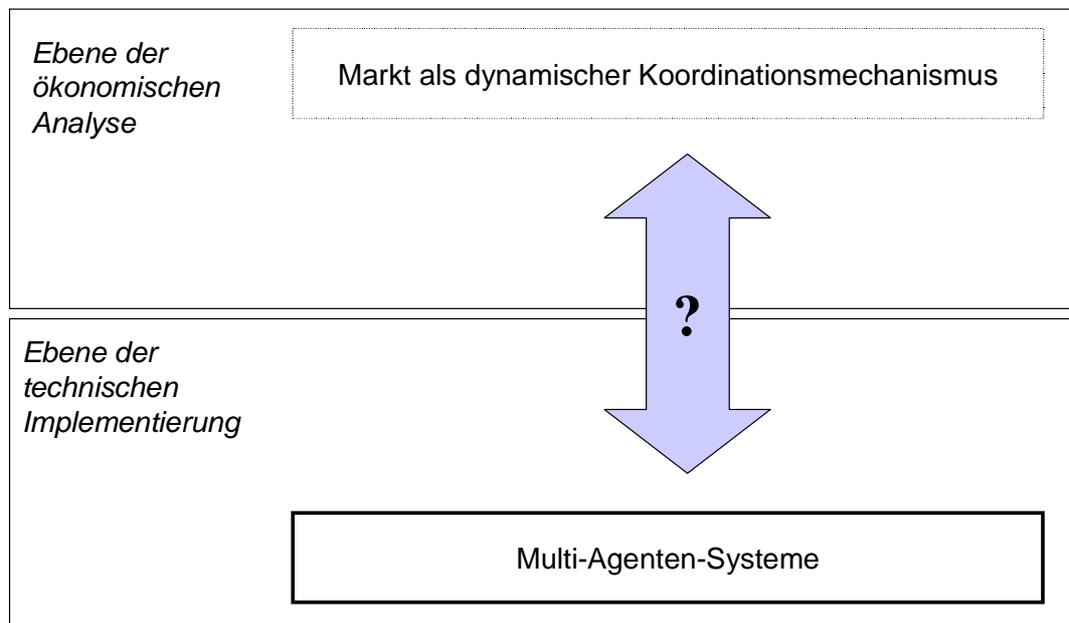


Abb. 3: Koordination durch Marktmechanismen in Multi-Agenten-Systemen

In einer solchen Modellwelt besteht die Aufgabe eines Koordinationsmechanismus darin, Konflikte in den Zielen zweier oder mehrerer Agenten aufzulösen. Welcher Allokationsmechanismus ausgewählt wird, entscheidet sich an bestimmten Anforderungen aus generellen Überlegungen zur wirtschaftsinformatischen Realisierung und den speziellen Anforderungen aus dem Szenario.

"In designing a mechanism to solve allocation problems, we consider desirable properties for outcome allocations, as well as computational issues in determining allocations from agent interaction protocols." (Wellman/Wurman 1997, S.5)

Diese allgemeingültigen Charakteristika des Mechanismus bzw. dessen Ergebnis lassen sich detailliert wie folgt darstellen (Tygar 1998, Tesch/Fankhauser 1999, Varian 1998, Rosenschein/Zlotkin 1994, Wellman 1997):

1. **Verfolgung ökonomischer Prinzipien:** Der Mechanismus sollte ökonomische Prinzipien verfolgen, d.h. faire Ergebnisse für alle Parteien erzielen. Dies kann am besten erreicht werden, wenn es keine Anreize und Optionen zu lügen und betrügen gibt und wenn statt einer spekulativen Handlungsweise der tatsächliche Indifferenzpreis für ein Gut geboten wird ("*incentive compatibility*"). Auch sollte ein einzelner Marktteilnehmer die Allokation nicht zu seinen Gunsten einseitig verändern können. Diese Argumente kommen aus einer ökonomischen

misch-analytischen Perspektive und erzeugen eigene Probleme in der Implementation (vgl. zu Spekulation Rasmusson/Janson 1999, zu Auktionen Sandholm 1998).

2. **Schnelle Ausführung:** Der Mechanismus sollte eine schnelle Ausführung ermöglichen, d.h. der einzelne Agent sollte schnell zu seinem Ergebnis kommen können. Je schneller eine Rückmeldung an den Agenten bezüglich der Auswirkungen seiner Aktionen erfolgt, desto eher kann sich dieser an eine veränderte Umwelt anpassen. Dieses technische Problem wird durch die Verarbeitungsgeschwindigkeit eines Mediators bzw. durch die Verteilung und Kommunikationsgeschwindigkeit zwischen den Agenten beeinflusst. In Simulation und geschlossenen Systemen wird meist auf den Kunstgriff der Rundenbasiertheit (engl. "*time slicing*" bzw. "*time slotting*") zurückgegriffen, d.h. die Ausführungszeit wird in diskrete Runden unterteilt, die keine Abhängigkeiten voneinander aufweisen. Diese Möglichkeit besteht in offenen Netzwerken und mit zunehmender Realitätsnähe nicht¹².
3. **Privatheit:** Der Mechanismus soll Sicherheit bieten und Privatheit schützen. Die eigenen, geheimen Präferenzen des Agenten dürfen nicht nach außen gegeben werden. Weder der Marktbetreiber noch dritte Agenten sollen von den wechselnden Geboten in einer Verhandlung mit einem anderen Agenten erfahren. Lediglich der tatsächliche Verkaufspreis kann am Ende mitgeteilt werden. Anderenfalls kann die Information zum Ausspähen der Strategie in späteren Verhandlungen genutzt werden, was keinesfalls im Sinne des Agenten (und seines Prinzipals ist): "*All else being equal, privacy reduces the ability of agents to gain strategic advantage by reasoning about or manipulating the beliefs of others.*" (Wellman 1997, S.8). In diesem Zusammenhang ist auch Anonymität zu nennen; sollten Agenten ihre Identität schützen wollen, muß der Koordinationsmechanismus anonyme bzw. pseudonyme Teilnehmer ermöglichen.
4. **Skalierbarkeit:** Der Mechanismus muß für eine prinzipiell beliebig große Teilnehmerzahl anwendbar sein und unabhängig von der Zahl der im Markt agierenden Agenten und deren Kommunikationsaufkommen arbeiten. Eine Informationsverarbeitung, die an einem Punkt im System warten muß, führt zu einer verzögerten Rückmeldung über die Auswirkungen bestimmter Aktionen und damit tendenziell zu einer Selbstverstärkung einer begonnenen Entwicklung.

12 Dieses Argument gilt für Szenarien, in denen Agenten nicht rundenbasiert handeln und mehrere Aktionen gleichzeitig verfolgen können. In Szenarien, in denen ein Agent nur eine Aktion verfolgt und sich die Umwelt schrittweise und für alle Agenten gleichzeitig verändert (z.B. in Auktionen), tritt dieses Problem naturgemäß nicht auf.

5. **Individuelle Rationalität:** Jeder Agent muß die Chance haben, durch eigene Aktionen seinen Nutzen verbessern zu können. Im Zeitablauf des Koordinationsmechanismus kann es schließlich Gewinner und Verlierer geben, aber dieses Risiko muß grundsätzlich auf Entscheidungen des Agenten zurückzuführen sein. Falls dies nicht gewährleistet werden kann, wird der einzelne Agent nicht an dem Allokationsmechanismus teilnehmen und durch externes Handeln diesen letztendlich sogar im Ergebnis stören. Diese Eigenschaft wird als individuelle Rationalität bezeichnet: "*A mechanism is individually rational if no agent would not prefer not to participate in the mechanism at all*" (Wellman 1997, S.8).
6. **Terminierung:** Die Vorgehensweise eines Koordinationsmechanismus kann in zwei Teilschritte unterteilt werden. Die zuerst stattfindende Ressourcenallokation verteilt lediglich die Verfügungsrechte über eine Ressource auf die Teilnehmer. Im zweiten Schritt folgt erst die Ressourcenverwendung, z.B. innerhalb eines Produktionsvorgangs. Vom Blickpunkt der neoklassischen Ökonomik bedeutet dies, dass der Allokationsmechanismus von vorläufigen (tentative) Lösungen zu einer endgültigen Lösung konvergiert, die auch gewählt worden wäre, wenn alle Agenten von vornherein vollständige Information gehabt hätten. Diese Lösung wird meist das allgemeine Gleichgewicht sein (*general equilibrium*). Das bedeutet jedoch auch, dass während der Laufzeit des Allokationsmechanismus keine Ressourcenverwendung stattfindet und dass jede vorläufige Lösung reversibel ist. Diese Annahme gilt für alle Realisierungen, die von einer Verankerung des Mechanismus in der neoklassischen Ökonomik ausgehen, nicht jedoch für Non-Tâtonnement-Prozesse (vgl. Kapitel 2.5) und solche Verhandlungen, bei denen keine Unabhängigkeit der Verhandlungen besteht: "*if an agent's ability to produce X is contingent on its acquisition of Y, its negotiations for these resources are inherently coupled*" (Wellman 1997, S.9). Wenn die Schritte der Verfügungsrechteübertragung und des Ressourcenverbrauchs ineinandergreifen, wird jede (auch vorläufige) Lösung zu einer Veränderung des Problems führen, so dass keine iterative Annäherung möglich ist. In diesem Fall würden die Agenten ständig neue und veränderte Preisentscheidungen treffen und der Allokationsprozeß ein kontinuierlicher und nicht an einem Gleichgewicht endender sein.
7. **Qualitative Verbesserung:** Das Ergebnis des Mechanismus muß in mindestens einer Dimension relativ besser im Vergleich zu allen anderen erreichbaren Allokationen sein. Ob das Ergebnis auch theoretisch das beste ist, kann nicht beurteilt werden; im praktischen Fall gilt "*it is also the most we could require without introducing some judgement about the relative importance of satisfying various agents*" (Wellman 1997, S.6).
8. **Adaptivität und dezentrale Steuerung:** Der Agent muß auf eine Veränderung der Umweltbedingungen reagieren können. Die Verfolgung fester Regeln in Zeiten, in denen

sich das eigene Weltmodell schnell und vielleicht unbemerkt ändern kann, verstärkt nur eine einmal begonnene Entwicklung mit unabsehbaren Auswirkungen für das Gesamtsystem. Andererseits können die Agenten gerade wegen der dezentralen Struktur der Märkte und der inhärenten Vernetzung der Agenten auf die Entwicklung des Gesamtsystems selber einwirken (sie sind nicht nur Preisnehmer). Durch geeignete Regeln und Anreize in den Agenten ergeben sich Möglichkeiten, derartige Systeme dezentral zu steuern.

Mit diesen konzeptionellen und technischen Charakteristika, die ein Koordinations*mechanismus* zeigen muß, um von den Teilnehmern bzw. einer dritten Partei als nutzenstiftend eingestuft zu werden, kann der Kreis der in Frage kommenden Koordinationsmechanismen bereits stark eingeschränkt werden. Zusammen mit den Anforderungen des Koordinationsproblems ergibt sich ein Raster, welches die Suche nach ökonomischen Erklärungsansätzen und deren möglicher technischer Umsetzung in den folgenden Kapiteln determiniert.

1.8 Aufbau der Arbeit

Der Einsatz von Agenten, die in elektronischen Märkten miteinander handeln und damit Agentensysteme bilden, führt zur Frage nach in diesen agentenbasierten Märkten anwendbaren Koordinationsmechanismen. Die Fallstudie des Schwarzen Montags 1987 zeigt, dass die Frage der Gestaltung des Koordinationsmechanismus wesentlich für den Einsatz von aus vielen individualistischen Agenten bestehenden Multi-Agenten-Systemen in offenen Umgebungen ist und die Akzeptanz dieser Technologie determiniert. Die Anforderungen an die Koordination in Multi-Agenten-Systemen können anhand eines Abstraktionsmodells für Telematik-Systeme (Schichtenmodell) abgebildet werden.

Die Konzepte für Koordinationsmechanismen aus den Wirtschaftswissenschaften, die in **Kapitel 2** vorgestellt werden, eignen sich in unterschiedlicher Weise zum Einsatz in agentenbasierten elektronischen Märkten. Diejenigen Leser, die mit den Marktkonzepten der Wirtschaftswissenschaften (insbesondere dem Walras'schen Auktionator und Hayeks Katallaxie) vertraut sind, können direkt zu den folgenden Kapiteln übergehen. Die Nachteile zentraler Ansätze führen in Kapitel 2 zur Suche nach einer formalisierbaren Beschreibung eines dezentralen Koordinationsmechanismus, der in selbstorganisierender Weise den Ausgleich zwischen den Software-Agenten durchführt. Dieses Konzept dient als Vorlage für die in Kapitel 4 beschriebene Realisierung eines agentenbasierten Marktplatzes mit dem Namen AVALANCHE.

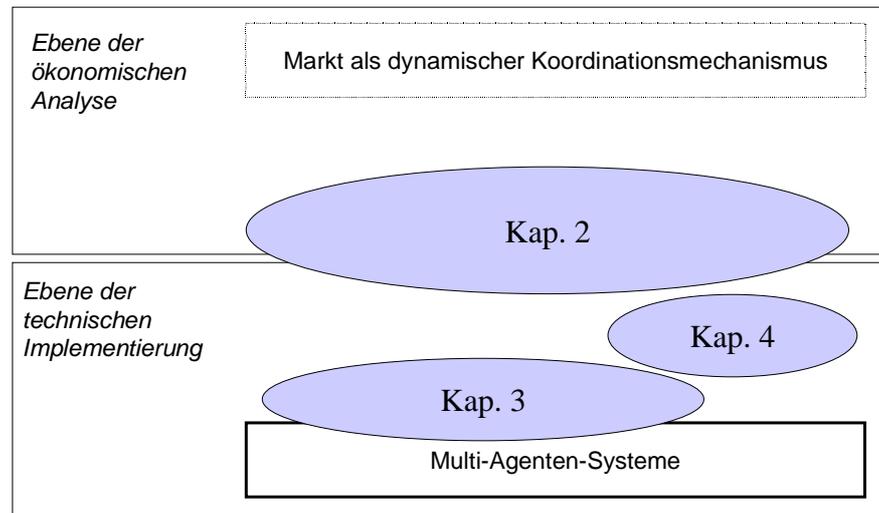


Abb. 4: Kapitelübersicht in grafischer Darstellung

In **Kapitel 3** werden die informatischen Grundlagen der VKI und damit der Software-Agenten und Multi-Agenten-Systeme, sofern sie für AVALANCHE relevant sind, erläutert. Leser, die mit diesen Aspekten der Informatik vertraut sind, können dieses Kapitel übergehen. Darin geht es um die Frage, wie Software-Agenten miteinander kommunizieren und kooperieren. Die informatischen Konzepte der Verteilten Künstlichen Intelligenz, Verteiltes Problemlösen und Multi-Agenten-Systeme, werden dargestellt und auf ihre Eignung für die Beschreibung elektronischer Marktplätze untersucht.

In **Kapitel 4** wird der Aufbau und die Funktionsweise des Multi-Agenten-Systems AVALANCHE dargestellt. Dieses stellt die Realisierung eines strikt dezentralen Koordinationsmechanismus dar. Die Beschreibung der technischen Realisation wird begleitet von einer Darstellung der Gestaltungsoptionen und den Begründungen, warum auf den unterschiedlichen Ebenen des Schichtenmodells der Telematik jeweils eine bestimmte Implementation realisiert wurde.

Anschließend wird in **Kapitel 5** das dynamische Verhalten des Koordinationsmechanismus in AVALANCHE in experimenteller Weise untersucht. Dieses Vorgehen stellt nicht nur einen Funktionstest dar, sondern geht der Frage nach, ob es sich bei AVALANCHE im Wortsinne um einen „elektronischen Markt“ handelt. Ausgehend von unterschiedlichen Umgebungs- und Ausgangszuständen kann gezeigt werden, welchen Einfluß diese auf das Ergebnis der Koordination haben, z.B. wie sich die Strategien der Agenten zur Nutzenverfolgung im Zeitablauf anpassen und welche Preis- und Gewinnentwicklungen sich ergeben. Daraus lassen sich wiederum Rückschlüsse auf die Eignung von Multi-Agenten-Systemen für die Koordination wirtschaftlicher Anwendungsszenarien ziehen.

Die Arbeit schließt in **Kapitel 6** mit einer Gesamtbetrachtung der Ergebnisse und einem Ausblick auf noch offene Fragestellungen. AVALANCHE wird als Prototyp einer bestimmten Klasse marktkoordinierter MAS eingeordnet. Zum Schluß werden mögliche Weiterentwicklungen des Konzepts in anwendungsspezifischer und technologischer Hinsicht kurz vorgestellt.

2 Der Markt als Koordinationsverfahren

In diesem Kapitel werden Erklärungs- und Gestaltungsansätze zur Koordination zwischen individualistisch motivierten Akteuren beschrieben, die unabhängig von der Realisierung in „menschenbasierten“ Märkten oder auf Basis der Technologie der Multi-Agenten-Systeme sind. Wenn angenommen werden kann, dass Software-Agenten in ähnlicher Weise wie Menschen handeln können, dann stellt sich die Frage der Realisierung erst in zweiter Linie. Zuerst geht es daher um die technologie-unabhängige Frage des "*Economic Mechanism Design*" (Varian 1995), d.h. der Festlegung eines dynamischen Verfahrens zur Koordination:

„Suppose we are given a set of information processors each limited in its ability to accomodate and analyze incoming information (for example, workers in a firm or CPUs in a computer). What is the most effective way to organize these processors?“ (Miller 1996, S.2)

Diese Frage lässt sich methodisch entweder durch mathematische Modelle oder durch computergestützte Experimente erforschen. Dabei ist zwischen der Untersuchung des *Mechanismus* und des *Effekts* (Kearney/Merlat 1999) zu unterscheiden. Der *Mechanismus* beschreibt das (dynamische) Verfahren, welches von einem Ausgangszustand zu einem Endzustand führt. Der *Effekt* ist dagegen die Beschreibung des (statischen) Endzustandes. Der Vorteil experimenteller Untersuchung mittels Computern läßt sich so zusammenfassen:

“Computational models enable the study of complex, dynamic worlds because they themselves are dynamic.“ (Page 1999, S. 2).

Da im Fokus dieser Arbeit die Untersuchung des Koordinationsmechanismus steht, werden Computereperimente im Lichte dieser Bemerkung als zweckmäßiger angesehen. Prototypen elektronischer Marktplätze können dabei, wie auch das in dieser Arbeit vorgestellte Multi-Agenten-System AVALANCHE, bereits *qua Existenz* eine Experimentierumgebung darstellen.

Die Anwendungsdomäne elektronischer Marktplätze zeichnet sich dadurch aus, dass Softwareagenten, die im Auftrag eigennutzenmaximierender Wirtschaftssubjekte handeln sollen, vor einem verteilten Ressourcenallokationsproblem stehen. Die Organisation der Software-Agenten, so dass sie sich koordinieren, kann in einem offenen Markt durch niemanden (im Wortsinne) angeordnet werden. In diesem Kapitel sollen neben den Konzepten mögliche Realisierungen solcher Koordinationsmechanismen vorgestellt und diskutiert werden, die einen Markt oder einen marktähnlichen Mechanismus darstellen, der technisch zu einer über die

Gesamtdauer der Koordination für alle Teilnehmer befriedigenden (nicht notwendigerweise optimalen) Allokation führt.

Auch wenn für den gesamten Markt ein optimaler Zustand angegeben werden könnte, wie z.B. eine optimale Wohlfahrt, so fehlt es doch an den Möglichkeiten, die Softwareobjekte zur Ausführung bestimmter Aktionen zu zwingen, um diesen Zustand zu erreichen. Die Autonomie der Agenten erstreckt sich letztendlich auch auf die freiwillige Unterwerfung unter die Ergebnisse eines Koordinationsmechanismus. Ein Agent wird seine Teilnahme grundsätzlich auf solche Mechanismen beschränken, bei denen er von einem gleichen oder besseren Ergebnis als bei anderen Mechanismen ausgeht, oder anders gesagt, bei denen er von einer Verbesserung seiner Startallokation überzeugt ist (Rasmusson/Janson 1999). Ein „erfolgreicher“ Koordinationsmechanismus muß Anreize möglicher Nutzenmaximierung bereitstellen, um die Agenten aus deren eigener Nutzenverfolgung zu bestimmten Aktionen zu verleiten. Diese Ansichtweise unterscheidet das Vorgehen der vorliegenden Arbeit von der Untersuchung geschlossener Marktexperimente, bei denen Aktionen zur Erreichung eines Gesamtsystemzustandes erzwungen werden können.

Um darzustellen, welchen Annahmen die Koordination eines offenen elektronischen Marktes folgt und welche alternativen Koordinationsmechanismen es gibt, soll im folgenden auf das „Warum“ und das „Wie“ von Märkten als Koordinationsinstrumenten hingeführt werden. Dazu wird zuerst auf die Definition von Koordination an sich eingegangen. Anschließend wird dargestellt, welche Eigenschaften der Anwendungsumgebung die Wahl eines bestimmten Mechanismus grundsätzlich beeinflussen. Die drei möglichen Realisierungskonzepte für Marktmechanismen (Arbitrator, Auktionator und Katallaxie) werden anhand der Charakteristika des Koordinationsproblems und der Anforderungen an einen Koordinationsmechanismus aus Kapitel 1 für ihre Eignung für Multi-Agenten-Systeme bzw. offene elektronische Marktplätze untersucht.

2.1 Koordination als Ausgleich von Interdependenzen

Koordination kann als bilaterale Koordination zwischen zwei Akteuren oder als multilaterale Koordination in einem System, möglicherweise durch eine dritte Partei verstanden werden (Klein 1997). Ziel und Aufgabe der Koordination ist *"managing interdependencies between sharing resources"* (Malone/Crowston 1994) und damit die Aufteilung und Zuweisung der Allokation von gemeinsamen Ressourcen zwischen Akteuren.

Diejenigen Voraussetzungen, welche die Anwendung eines Koordinationsmechanismus notwendig machen, sind das Vorliegen solcher Interdependenzen. Nur dann kann durch Koordination überhaupt ein Problem gelöst werden:

"Coordination is only required, of course, when distributed activities are interdependent"
(Klein 1997, S. 162).

Diese Interdependenzen können entlang dreier Achsen verteilt sein, welche die einsetzbare Technologie determinieren (Klein 1997, S.162):

- Besteht eine Verteilung der **Interdependenzen zwischen Akteuren** ("*across participants*"), muß die eingesetzte Technologie die Sequentialisierung der Aufgabenverteilung und den Informationsfluß zwischen Akteuren ermöglichen. Gemeinsame Ressourcen sind also die Verfügbarkeit physischer Ressourcen zu gleichen Zeitpunkten. Als Beispiel wird Prozessmanagementsoftware und Planning/Scheduling angeführt. Nachteile dieser technischen Lösungen sind die mangelnde Skalierbarkeit und die Ausnahmenbehandlung (exception handling), auf die später noch detailliert eingegangen wird.
- Eine Verteilung der **Interdependenzen über Zielvorstellungen** ("*across perspectives*") findet statt, wenn die Akteure an unterschiedlichen Aspekten einer Gesamtentscheidung arbeiten und/oder häufig inkompatible Individualziele verfolgen. Hier ist die gemeinsame Ressource die Gesamtentscheidung bzw. das zu erreichende Gesamtziel. Die Konsistenz der Aktionen der Individuen wird technisch z.B. durch Konfliktmanagementsoftware sichergestellt (Klein 1997, S. 170).
- Die Verteilung der **Interdependenzen in der Zeit** schließlich geht von der Notwendigkeit aus, Entscheidungswege (z.B. Diskussionsabläufe und deren Ergebnisse) zu speichern, um diese Informationen zu einem späteren Zeitpunkt bei ähnlichen Problemstellungen erneut nutzen zu können. Diese Dimension geht technisch in die Gebiete des Wissensmanagement bzw. des Organizational Learning (Strauss 1995) ein und wird im Verlauf dieser Arbeit nicht weiter betrachtet.

Es läßt sich festhalten, dass das eigentliche Koordinationsproblem die Interdependenz der persönlichen Ziele jeweils mindestens zweier Individuen umfaßt. Versteht man das individuelle Ziel als den Wunsch nach Zugriff auf eine physische Ressource, so ist die "Interdependenz zwischen Akteuren" auch unter diesem Aspekt subsumiert. Ein Beispiel mag dies verdeutlichen¹³: Wenn Anna und Benno gemeinsam in den Urlaub fahren möchten, Anna aber ans Meer und Benno in die Berge, so liegen unterschiedliche Individualziele vor und damit eine Interdependenz.

13 Dieses Beispiel ist in ähnlicher Form in der Spieltheorie unter dem Begriff "Kampf der Geschlechter" verbreitet (vgl. Holler/illing 1996, S. 11f.)

denz über Zielvorstellungen. Falls die Zielvorstellung lautet "mit dem gemeinsamen Auto ans Meer" bzw. "in die Berge" zu fahren, dann besteht gleichzeitig eine Interdependenz über physische Ressourcen. Das Ziel "Zugriff auf das Auto" ist aber eigentlich Ausdruck konfliktionärer Zielvorstellungen. Die Abgrenzung zwischen beiden Perspektiven ist im Einzelfall schwierig. Der Einfachheit halber wird im folgenden bei Konflikten von Zuteilungs- bzw. Zugriffsproblemen ausgegangen, welche die explizite Auflösung eines (existierenden oder prognostizierten) konfliktionären Zugriffs auf gemeinsame Ressourcen erfordern.

Wie jedoch funktioniert der Mechanismus des Marktes, der die Koordination zustande bringt? Kann ein solches Marktkonzept mit Informationstechnik realisiert werden? Aus welchen Bestandteilen ist dieser Marktmechanismus zusammengesetzt, welche Objekte müssen berücksichtigt werden und wie gestaltet sich die konkrete Interaktion dieser Objekte? Diese Fragen werden im weiteren ausgeführt und führen zur Synthese eines Konzeptes, das den "Markt" als ein Vielkomponentensystem ohne zentralen Koordinator begreift und der Möglichkeit seiner Formalisierung anhand der Technologie der Multi-Agenten-Systeme (vgl. Abbildung 3 auf Seite 21). In nächsten Abschnitt wird der Frage nachgegangen, ob die Wirtschaftswissenschaften eine Definition des Marktmechanismus bieten können, welcher diese konzeptionelle Klarheit ermöglicht.

2.2 Der Markt als dynamischer Koordinationsmechanismus

Definitionen des Begriffs „Markt“ wie die folgende: *"Märkte sind ein Koordinationssystem, das Angebot und Nachfrage aufeinander abstimmt, ohne dass Anbieter und Nachfrager gleiche oder auch nur ähnliche Ziele verfolgen müssen."* (Kieser/Kubicek 1992, S. 118) sind nicht ausreichend für die wirtschaftsinformatische Umsetzung. Der Markt als Untersuchungsobjekt der Wirtschaftswissenschaften ist in seiner technischen Funktion nur wenig untersucht und die Definition eines Marktes variiert je nach Fragestellung:

"Now it is paradoxical how variously and vaguely defined the concept of market is"
(Ménard 1995, S. 168).

Der Mechanismus des Zustandekommens der Preise in einem Markt wird größtenteils auf die "unsichtbare Hand" von Adam Smith (1776) zurückgeführt, ohne dass bis heute eine detaillierte Beschreibung dieses Marktmechanismus existiert. In Bezug auf die in den Wirtschaftswissenschaften dominante neoklassische Gleichgewichtstheorie wird beispielsweise kritisiert, dass sie *"zur Beantwortung der Frage, welche Mechanismen einen Gleichgewichtszustand herbeiführen, recht wenig und nur sehr Unbefriedigendes beizutragen [hat]"* (von Böventer/ Illing 1995, S. 216).

Weiter helfen Definitionen im Kontext von Wertpapierbörsen, wo der Begriff des Marktmodells *"den Mechanismus, auf welche Weise Orders zu Geschäftsabschlüssen im börslichen Handelssystem zusammengeführt werden"* bezeichnet (Deutsche Börse 2000, S. 4). Der Markt als abstrakter Ort der Abstimmung wird hier als konkreter Abstimmungsmechanismus verstanden. Diese Vereinfachung geht zu weit – in Konsequenz wäre der Markt letztendlich nicht mehr und nicht weniger als eine mathematische Formel. Dass wirtschaftliche Koordination und Märkte nicht programmiert werden können, ist eine Lehre aus dem Schwarzen Montag an der Wall Street (s. Kapitel 1) und, wenn man so will, auch aus der Niederlage der sozialistischen Planwirtschaft (vgl. Hayek 1988).

Märkte sind ökonomisch vor allem mit dem Auftreten vieler Individuen und deren Handeln von privatem Eigentum (bzw. Verfügungsrechten) verknüpft. Märkte sind in einem ökonomischen Sinne Räume, in denen Anbieter und Nachfrager aufeinandertreffen. Erstere bieten Güter an und fragen Geld nach, letztere bieten Geld an und fragen Güter nach. Findet ein Handel statt, so werden Geld und Waren ausgetauscht. Witt (1980, S. 17f.) bezeichnet beispielsweise solche Objekte als "Markt", bei denen *"das Auftreten von mehr als einem Individuum, das für die entsprechende Klasse von Gütern Austauschofferten gegen andere oder eine bestimmte Klasse von anderen Gütern (z.B. Geld) macht; Vertragsfreiheit der Individuen sowohl hinsichtlich des Transaktionsumfangs wie des Austauschverhältnisses (Preises) und ggf. der Konditionen in dem Sinn, dass keine Seite der anderen die Transaktion zu Bedingungen auferlegen kann, die diese nicht akzeptiert hat"*, beobachtet werden kann. Auch Moulin hebt die Vielzahl an unterscheidbaren Teilnehmern mit eigenem Besitz und eigenen Zielen hervor:

"Markets describe the exchange and production of commodities in the private property regime. The ownership structure assigns each unit of good and each technology to exactly one participant. Under specialization and division of labor, the initial allocation of resources is not expected to be Pareto-optimal [...]. Markets are meant to exploit these cooperative opportunities by eliciting Pareto improving trades" (Moulin 1995, S. 45).

Märkte bieten zusammengefaßt eine Plattform, auf der Agenten miteinander in Kooperation treten können, wenn sie dies aus Gründen der individuellen Nutzensteigerung anstreben. Aufgrund der Arbeitsteilung der Agenten innerhalb eines grösseren Systems wird a priori von einer nicht-effizienten Verteilung der Ressourcen ausgegangen. Um eine Realisierung eines Koordinationsmechanismus im Sinne dieser Ausführungen als "Markt" bezeichnen zu können, resultieren drei charakteristische Anforderungen an die Technik:

1. Jeder Marktteilnehmer stellt ein eigenes, unterscheidbares Objekt in der Realisierung dar. Damit scheidet ein Totalansatz, der den Markt insgesamt modelliert, aus. Grundsätzlich

werden die Bestandteile eines verteilten Informationssystems als ökonomische Agenten betrachtet (Kurose et al. 1985). Eine Modellierung der Marktteilnehmer, deren Koordination, Kooperation und Kommunikation sowie notwendiger Institutionen des Marktes mit Ansätzen objektorientierter Modellierung bietet sich hingegen an.

2. Die Individuen kommunizieren über Austauschofferten, die dem Zweck der individuellen Nutzenerhöhung dienen, über einen Übergang des Verfügungsrechtes für im privaten Besitz befindliche Ressourcen.
3. Die Vertragsfreiheit der Objekte kann derart gewährleistet werden, dass die Aufnahme der Transaktion den teilnehmenden Objekten freigestellt ist. Dies erfordert eine Autonomie der Objekte und damit die Nutzung eines agentenbasierten Ansatzes.

Aus diesen Anforderungen ergibt sich, dass eine technische Realisierung eines Marktes fast zwangsläufig mit einem Ansatz als dezentralem Multi-Agenten-System einhergeht, wenn er für die ökonomische Analyse, Messung und Steuerung zugänglich sein soll. Welche dezentralen technischen Konzepte gibt es in der Ökonomie? Dazu wenden wir uns im nächsten Abschnitt zuerst der ökonomischen Spieltheorie zu, die das Abstraktum "Markt" als erste in die einzelnen Teilnehmer (Agenten) zerlegt hat.

2.2.1 Spieltheoretische Konzepte der marktlichen Koordination

Moulin (1995, S. 11) definiert aus Sicht der ökonomischen Spieltheorie drei idealisierte Konzepte der Koordination, die das dynamische Zusammenspiel der einzelnen Agenten beschreiben. Jeder Realisierung eines Marktes liegt eines dieser Konzepte implizit zugrunde. Zweck dieses Abschnittes ist es, die Grundlagen und Annahmen dieser Konzepte darzulegen, um später die Ausgangsbasis der einzelnen Realisierung besser verstehen zu können.

Zweck der Kooperation zwischen den Agenten kann nur eine Nutzenmaximierung sein, die quantitativ über dem Niveau liegt, welches ein Agent allein erreichen könnte:

"Cooperation in the economic tradition is mutual assistance between egoists." (Moulin 1995, S. 5).

Voraussetzung dafür ist eine Arbeitsteilung zwischen den Agenten, die in den bereits angeführten Interdependenzen mündet. Jede Kooperation zwischen zwei oder mehr Agenten ist Teil des Interdependenzmanagement-Prozesses (Malone/Crowston 1994) des Gesamtsystems, im in dieser Arbeit verwendeten Sinne also der Koordination. Moulin selbst spricht nicht von Koordination, da er nicht das Gesamtsystem betrachtet, sondern er beschränkt sich auf die Kooperation zwischen Agenten und stellt dazu spieltheoretische Betrachtungen an. Der Nutz-

engewinn, den die Agenten aus ihrer gegenseitigen Hilfestellung ziehen, kann entweder nur für den eigenen Agenten betrachtet werden oder für alle an einer Transaktion beteiligten Agenten. Daraus ergibt sich als Qualitätskriterium für den Gesamtzustand das Konzept der Pareto-Optimalität:

"The concept of Pareto optimality (also called economic efficiency, or simply efficiency) follows logically from this view of cooperation" (Moulin 1995, S. 5).

Die Maximierung der Nutzensumme liefert im Allgemeinen das gleiche Ergebnis wie die Lösung des Problems der Pareto-Effizienz (Varian 1994, S. 336ff.). Aus dieser Kombination resultiert das Streben nach einem gemeinsamen besten Nutzen aus der Kooperation.

Der dynamische Prozeß der Koordination wird bei Moulin in drei Formen eingeteilt: den Gerechtigkeitsmodus, den Modus dezentralen Verhaltens, sowie den Modus direkter Verhandlung (vgl. Abb. 5).

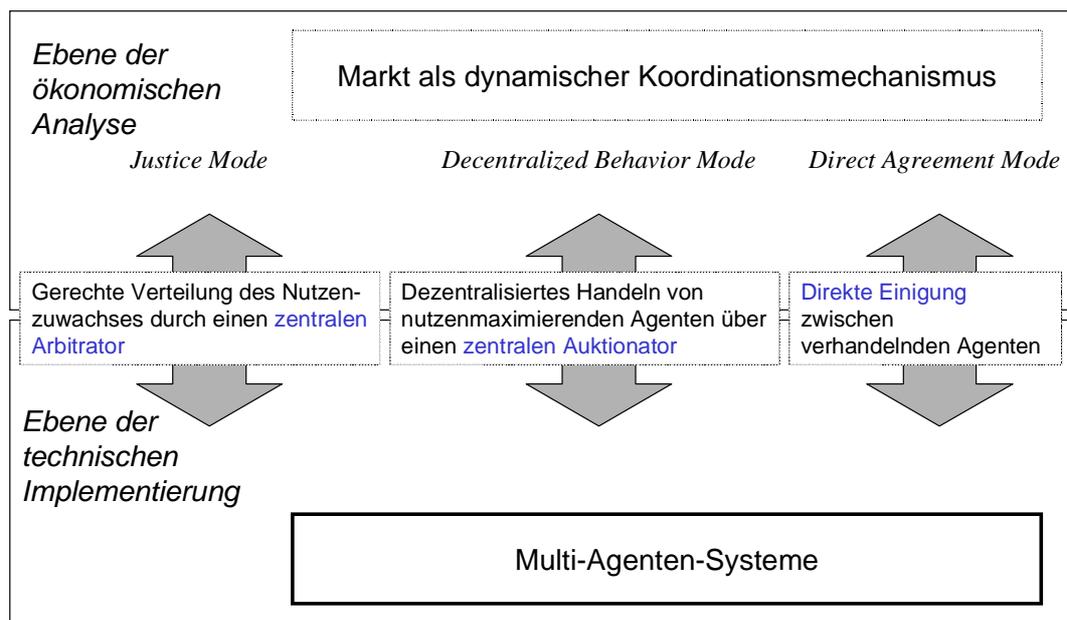


Abb. 5: Koordinationsmechanismen nach Moulin (1995)

Im Konzept des **Gerechtigkeitsmodus** (*justice mode*) (Moulin 1995, S.19) existiert ein zentraler Arbitrator, der über die Verteilung des Kooperationsgewinns entscheidet. Eine gerechte (*justified*) Verteilung des Gewinns ist eine solche, die alle beteiligten Agenten als fair ansehen. Das setzt voraus, dass der Mechanismus der Verteilung für die Agenten transparent ist.

"These two requirements leave not much room to the arbitrator: she must select her decision by means of an equitable mechanical formula." (Moulin 1995, S. 19)

Der einfachste Mechanismus ist offensichtlich die Verteilung zu gleichen Teilen¹⁴. Im Fall eines elektronischen Marktes sei folgendes Beispiel gegeben:

Anna besitzt ein Gut, welches für sie € 20 wert ist. Benno möchte dieses Gut gerne haben, hat genügend Geld und schätzt den Wert des Gutes auf € 24. Der Arbitrator entscheidet, dass das Gut für € 22 seinen Besitzer wechselt, so dass sowohl Anna als auch Benno einen gleichen Gewinn von € 2 realisieren können.

Um diese Fairness gewährleisten zu können, sollten sämtliche Informationen allen Teilnehmern offengelegt werden können. Spekulatives Verhalten, z.B. das Übermitteln falscher Präferenzen, ist konzeptionell ausgeschlossen. Der Arbitrator besitzt eine vollkommene Kenntnis aller Präferenzen und Nutzenfunktionen. Die einzelnen Agenten übertragen diese Informationen an den Arbitrator und warten bis zum Abschluß der Berechnung auf den neuen Allokationsvektor. Dadurch lässt sich sehr einfach eine gemeinsame Zielverfolgung erreichen, z.B. zur Realisierung der optimalen Wohlfahrt bei der Kompensation der Inanspruchnahme öffentlicher Güter durch die Gemeinschaft. Dieses *fair-share-Prinzip* wird von Moulin folgendermaßen definiert:

"If agent i 's utility [...] is u_i , the fair share of the resources is simply u_i/n , where n is the number of agents. The fair share property requires that every agent's net gain [...] be at least u_i/n " (Moulin 1995, S. 176).

Ein anderes Konzept zur fairen Berechnung ist der sogenannte *no-envy-Test*:

"The no-envy-Test says that no agent should wish to exchange his own allocation (awarded by a particular solution) for the allocation of any other agent" (Moulin 1995, S.178).

Die Aufgabe des Arbitrators ist es, die Fairness entweder nach dem *fair-share-Prinzip* oder dem *no-envy-Test* sicherzustellen¹⁵.

14 Hier wäre erneut auf die Annahme des identischen Grenznutzens zu verweisen, d.h. in allen hier besprochenen Beispielen hat eine Geldeinheit für jeden Agenten den gleichen Nutzen, unabhängig von der Kapitalausstattung, den Handlungsalternativen, oder den Zielen. €2 sind für Anna in jeder Situation genauso viel wert wie für Benno.

15 Hier kann noch unterschieden werden zwischen *end-state justice* und *procedural justice*. Der erste Fall geht von einem allwissenden Arbitrator aus, der am Ende der Kooperation die Nutzen- bzw. Geldverteilung vornimmt. Im zweiten Fall werden die Agenten mit gleichen Startausstattungen an Gütern und Aktionsmöglichkeiten ausgerüstet, so dass das Ergebnis automatisch als gerecht bezeichnet werden kann (Moulin 1995, S.37). Dieser zweite Fall ist letztendlich die Grundlage eines libertären Gedankengutes (Hayek 1976, Nozick 1974), wohingegen *end-state justice* eine Doktrin planwirtschaftlichen Vorgehens ist.

Gerade entgegengesetzt dazu steht der **Modus direkter Verhandlung** (*Direct Agreement Mode*), in dem dezentrale Lösungen ohne Koordinatorinstanz erreicht werden. Hier kann es sich beispielsweise um bilaterales Tauschen von Gütern handeln, so dass sich nach jedem Tausch der Nutzen des Agenten erhöht.

Wenn Anna aus dem obigen Beispiel ein Angebot bekommt, welches höher ist € 20, dann wird sie das Gut an den Meistbietenden verkaufen. Will Benno das Gut kaufen, wird er das geringere von zwei Angeboten wählen. Jeder Preis zwischen € 20 und € 24 befriedigt beide Agenten und da es sich um ein Nullsummenspiel handelt, ändert sich die Nutzensumme (= Wohlfahrt) damit nicht¹⁶. Was sich ändern kann, ist das Verhältnis des jeweiligen Nutzenzuwachses. Wenn Benno zuerst spricht und € 21 vorschlägt, wird Anna einwilligen und € 1 realisieren, gegenüber € 3 von Benno. Auch diese Verteilung ist pareto-effizient, denn Anna kann z.B. keine neue Verteilung vorschlagen, ohne dass Benno schlechter dastünde.

Die Verteilung ist nicht fair im Sinne des *fair-share-Prinzips*, damit wird aber auch deutlich, dass ohne die Kenntnis der Indifferenzpreise eine solche Fairness überhaupt nicht erreicht werden kann. Im Modus direkter Verhandlung haben alle Agenten grundsätzlich die Möglichkeit, ihren Indifferenzpreis geheimzuhalten. Ein strategisches Vorgehen hat einen deutlichen Einfluß auf den eigenen Nutzen.

"The question is: what agreements (what Pareto optimal outcomes) are then likely to occur? What agreements are deemed stable?" (Moulin 1995, S. 14).

Im ökonomischen Sinn wird sich eine solche Menge an Transaktionen ergeben, die ein Kernergebnis darstellen; d.h. eine Verteilung, die nicht von einer anderen Verteilung dominiert wird. Ein Ergebnis A wird dann von einem Ergebnis B dominiert, wenn es eine Untermenge von Agenten (eine "Koalition") gibt, die das Verlangen, aber auch die Macht haben, Ergebnis B durchzusetzen. Bei einer Situation, in der sich alternative Ergebnisse gegenseitig dominieren (z.B. A dominiert B dominiert C dominiert A), ist der Kern leer und damit das Verhalten der Effizienz suchenden Teilnehmer instabil.

"[...] Instability can only be revealed in a dynamic context [...], but always implies (even in the static context) a lack of predictability, a genuine indeterminacy of the final outcome." (Moulin 1995, S.17)

16 Unter der Annahme, dass Geld = Nutzen ist.

Eine stabile Kernallokation zeichnet sich dadurch aus, dass (1) der Handel freiwillig zwischen den Agenten stattfindet, da jeder Agent von diesem profitiert und seinen Nutzen erhöht, (2) die Allokation effizient ist, d.h. Pareto-optimal und (3) keine Koalition von Agenten eine bessere alternative Ressourcenverteilung finden kann.

Im **Modus dezentralen Verhaltens** (*Decentralized Behavior Mode*) verhalten sich die Agenten strategisch, d.h. sie wählen aus einer Anzahl ihnen von einer zentralen Instanz kostenlos zur Verfügung gestellten Aktionsalternativen. Die Agenten kommunizieren nicht direkt miteinander und können sich auch nicht auf ein Pareto-optimales Ergebnis einigen. Das Ergebnis kommt allein aufgrund der Aktionswahl der einzelnen Agenten zustande, ohne dass diese Kenntnis über die systemweiten Auswirkungen haben. Dieses Ergebnis wird meist als kompetitives Gleichgewicht bezeichnet und es zeichnet sich dadurch aus, dass jeder Agent ein Preissignal liest und daraufhin, unabhängig von jedem anderen Agenten, seine bevorzugte Transaktion auswählt (d.h. wieviel von jedem Gut er kaufen oder verkaufen soll). Nachdem alle Entscheidungen aller Agenten vorliegen, ergibt sich aggregiert eine entsprechende Nachfrage- und Angebotsituation, d.h. alle Transaktionen können durchgeführt werden. Der dynamische Prozeß der Aggregation der Preise wird zumeist über den Walras'schen Auktionator (Walras 1874) dargestellt. Ein wesentlicher Unterschied zwischen dem Ergebnis des Modus direkter Verhandlung und des Modus dezentralen Verhaltens ist die Preisbildung der Transaktionen. Im Modus dezentralen Verhaltens bezahlen alle Käufer den gleichen Preis für ein bestimmtes Gut. Im Fall des Modus direkter Verhandlung können die Käufer unterschiedliche Preise je nach Transaktion bezahlen. Damit wird deutlich, dass ein Intermediär, dessen Aufgabe die Aggregation und Bereitstellung der Preise ist, hier als einziger Kommunikationspartner der verteilten Agenten auftritt. Dieser hat aber, im Unterschied zum Gerechtigkeitsmodus, keine Aufgabe der Berechnung einer Allokationslösung. Dieser ergibt sich vielmehr durch dezentrale Bewertung und Entscheidungsfindung:

"Centralized control implies that the control functions are implemented in only one process. When several processes cooperatively carry out these functions, then the control is said to be decentralized." (Gagliano/Mitchem 1995, S. 29).

2.2.2 Marktliche Koordination als Ressourcenallokationsverfahren

Um entscheiden zu können, welche Klasse der Koordinationsmechanismen für welche Art von Koordinationsproblemen angewendet werden kann, muß das Koordinationsproblem selbst, welches am Beginn der Anwendung des Mechanismus steht, zuerst formal beschrieben werden.

"Computational economies were mainly regarded as mechanisms to achieve system global goals, i.e. a class of decentralized algorithms for resource allocation built into a system by top-down design." (Rasmusson/Janson 1999, S. 143).

Das Koordinationsproblem wird im folgenden als ein Problem verteilter Ressourcenallokation aufgefaßt. Der Koordinationsmechanismus wird als dynamischer Vorgang der Ressourcenallokation verstanden und beschrieben.

Um ein ökonomisches Modell einer Ressourcenallokation anwenden zu können, muß die Ressource auf einer Wertskala darstellbar sein.

"Most studies of resource allocation mechanisms have used a performance model of the resource, where the very concept of the resource is defined in terms of measurable qualities of the service such as utilization, throughput and delays." (Ferguson et al. 1995, S. 162)

Unterschiedliche Allokationen der Ressource haben Auswirkungen auf meßbare Größen. Im Raum der Kombinationen aller meßbaren Größen existiert eine Allokation, die sich als optimal definieren läßt. Diese Herangehensweise führt zu einer zweistufigen Abstraktion in der Realisierung, die sich nach Ferguson et al. (1995) so darstellen läßt:

Resource sharing → performance model → economic optimization model

Die quantifizierbare Parameterisierung eines Modells, welche Ressourcen existieren und wie sich diese teilen lassen, führt zu dem mathematisch beschreibbaren *performance model*. In dieser Abstraktionsebene ist der konkrete Anwendungsfall bereits herausgenommen. Dieser Schritt *"eliminates the details of the resource and captures its relevant behaviors and optimization parameters"* (Ferguson et al. 1995, S.162). Wenn alle Parameter vorliegen, kann im zweiten Schritt ein ökonomisches Modell herangezogen werden, um das *performance model* zu optimieren. Ferguson et al. (1995) klassifizieren zwei unterschiedliche Arten ökonomischer Modelle:

- Zum einen einen nicht-preisbasierten Ansatz, der spieltheoretische Modelle nutzt, um dezentralisierte Steuerungsmechanismen für die Ressourcenallokation zu konstruieren. Rosenschein/Zlotkin (1994a) unterscheiden beim Koordinationsproblem hier zwischen *Task-Oriented Domains* (ebda., S. 29ff.) und *State-Oriented Domains* (ebda., S. 87ff.), die nur exklusive Zugriffe auf Ressourcen zulassen und keine Abstufung in Form von Konzessionen vorsehen.

- Zum anderen ein Preismechanismus, bei dem Käufer und Verkäufer mittels Marktmechanismen interagieren und der Güterpreise und Geldübertragung einbezieht. Dieser kann angewendet werden, wenn es sich beim Koordinationsproblem um eines einer *Worth-Oriented Domain* handelt, d.h. deren Ziel "is subsumed by a worth function over all possible final states. Those states with the highest value of worth might be thought of as those that satisfy the full goal, while other, with lower worth values, only partially satisfy the goal" (Rosenschein/Zlotkin 1994a, S. 149). In diesem Fall können Konzessionen auf einer gemeinsamen Nutzenwertskala (Geld) eingegangen werden.

Auf derartige Probleme können jeweils die von Preist (1998) beschriebenen Möglichkeiten, welche Arten von Verhandlungsstrategien in ökonomisch handelnden Agenten implementiert werden können, in unterschiedlicher Weise angesetzt werden (vgl. Kapitel 3.5.3). Im folgenden wird immer von der Nutzung eines Preismechanismus in einer *Worth-Oriented-Domain* ausgegangen und die mathematische Beschreibung des Problems der ökonomischen Ressourcenallokation entsprechend dargestellt. Diese Darstellung liegt den weitergehenden Ausführungen im Verlauf dieses Kapitels zugrunde und ist Ferguson et al. (1995, S.160) entnommen, ähnlich auch Varian (1994, S. 391 ff.).

Eine Ökonomie enthält eine Menge von Ressourcen $\vec{r} = \{r_1, r_2, \dots, r_M\}$. Eine Allokation von Ressourcen eines Agenten wird als Vektor $\vec{x} = \{x_1, x_2, \dots, x_M\}$ dargestellt, wobei x_i diejenige Menge der Ressource r_i ist, die diesem Agenten augenblicklich zugeordnet sind. Ein Agent präferiert bestimmte Allokationen gegenüber anderen, was durch Nutzenfunktionen dargestellt wird. Diese Funktionen sind Abbildungen der Allokation auf die realen Zahlen. Sei $U(\cdot)$ eine Nutzenfunktion und seien x, y zwei Allokationen, dann präferiert der Agent $x \succ y$, wenn $U(x) > U(y)$.

Die Menge derjenigen Allokationen, die unter Budgetrestriktionen erreichbar ist (die Budgetmenge) wird durch folgende Funktion dargestellt:

$$B(\vec{p}) = \{x | \vec{p} \bullet x \leq w\} \quad (1)$$

mit $\vec{p} = \{p_1, p_2, \dots, p_M\}$ als Preisvektor, bei dem p_i der Preis für eine Einheit der Ressource r_i ist und w der Reichtum des Agenten (d.h. sein Gesamtbudget in Geldeinheiten). Die Nachfragemenge für einen Agenten ergibt sich als Menge an Allokationen, die in der Budgetmenge liegt und die am meisten präferiert wird:

$$\Phi(\vec{p}) = \{x | x \in B(\vec{p}), U(x) \geq U(x'), \forall x' \in B(\vec{p})\} \quad (2)$$

Einfach gesagt, fragt jeder Agent immer die meistpräferierte, für ihn bezahlbare Allokation von Ressourcen nach. Falls Interdependenzen vorliegen, können Agenten genau diese Alloka-

tion nicht erreichen, da sie in Konflikt mit einem anderen Agenten stehen. Damit ergibt sich eine Interdependenz "*across perspectives*", wenn beide Agenten das Ziel verfolgen, ihr jeweiliges $\Phi(\vec{p})$ zu erreichen und davon abgeleitet eine Interdependenz "*across participants*", wenn beide Agenten a, b auf die gleiche Ressource r_i in der Menge $x_{i,a}$ bzw. $x_{i,b}$ zugreifen wollen und es gilt $x_{i,a} + x_{i,b} > x_I$, mit $x_{i,a} \in \Phi_a(\vec{p})$ und $x_{i,b} \in \Phi_b(\vec{p})$, so dass die verfügbare Gesamtmenge x_I der Ressource (zur gleichen Zeit) geringer ist als die Summe der Zugriffswünsche. Genau diese Konfliktsituation ist der Ausgangspunkt des dargestellten Koordinationsproblems. Kann organisatorisch sichergestellt werden, dass diese Konfliktsituation nie eintritt, ist jeder Koordinationsmechanismus sinnlos.

Wie eine derartige Situation aufgelöst werden kann, d.h. in welcher Weise beide Agenten Konzessionen eingehen, so dass sie jeweils eine von der ursprünglichen Nachfragemenge nicht zu weit entfernte Allokation erreichen, wird durch einen Koordinationsmechanismus beschrieben. Dieser basiert grundsätzlich auf einem Kommunikationsprotokoll:

"a mechanism describes a communication protocol, in effect defining who can communicate with whom and what message types are allowed" (Wellman 1997, S.5).

Auf Basis dessen können wiederum Agenten miteinander kooperieren, um sich selbst zielgerichtet an der Koordination zu beteiligen:

"Coordination occurs when bargainers work together in search of a mutually acceptable agreement" (Pruitt 1989, S. 91).

Es ist auch möglich, dass Akteure sich unbewußt koordinieren, ohne direkt miteinander in Interaktion zu treten, z.B. wenn eine Kommunikation unmöglich ist (vgl. Schelling 1960, Fenster et al. 1995). Dieser Fall wird im folgenden nicht betrachtet.

Es lässt sich festhalten, dass für einen Koordinationsmechanismus eines Mehragentensystems, bei dem Konflikte beim Zugriff auf die knappen Ressourcen entstehen können, im wesentlichen zwei Aufgaben auf unterschiedliche Träger verteilt werden müssen:

1. Die einzelnen Akteure müssen aus einer Anzahl an Entscheidungsalternativen eine Aktion auswählen, mit der sie auf ihre Umweltsituation einwirken und diese kommunizieren (Effektor, vgl. Abbildung 8 auf Seite 80).
2. Die Ergebnisse der Summe der gewählten Aktionen der einzelnen Akteure müssen festgestellt und an die Akteure zurückgemeldet werden (Sensorik, vgl. Abb. 8).

Diese Aufgabenverteilung wird in den nächsten Abschnitten in unterschiedlichen Kombinationen dargestellt und auf die jeweilige Eignung für die Realisierung offener agentenbasierter Märkte untersucht.

Tabelle 1: Aufgabenverteilung der Koordinationsmodi

Aktionsauswahl	Ergebnisberechnung	Koordinationsmodus
zentral	zentral	Gerechtigkeitsmodus bzw. Arbitration (Kap. 2.3)
dezentral	zentral	Modus dezentralen Verhaltens bzw. Auktion (Kap. 2.4)
dezentral	dezentral	Modus direkten Handelns (Kap. 2.5)

2.3 Der Markt als Arbitrationsverfahren

Die Aufgabe der Koordination in Informationssystemen kann in der Definition von Malone/Crowston (1994), *"managing dependencies between activities"* zusammengefaßt werden. Gerade durch die Notwendigkeit der Integration verschiedener Informationssysteme in Unternehmen oder die Notwendigkeit der Kooperation zwischen Informationssystemen verschiedener Unternehmen (z.B. in Vernetzten KMU) ergibt sich Koordination als Abstimmung interdependenter Teilziele, sei es "nur" die Gewährleistung konsistenter Datenhaltung über mehrere Informationssysteme hinweg. Dabei wird immer die Kooperationsbereitschaft der Teilsysteme vorausgesetzt. Bei dieser Art von Architektur ist bereits die Verteilung der Anwendungsobjekte zentral geplant. Die Zielvorgabe des Gesamtsystems ist identisch mit den Zielen der einzelnen Objekte, die keinerlei Autonomie besitzen. Aufgabenzuweisung, Kontrolle und Ausnahmebehandlung sind alleine Aufgabe der Zentralinstanz.

Arbitration/Matchmaking benutzt einen zentralern Arbitrator, um die unterschiedlichen Präferenzen zwischen Anbietern und Nachfragern auszugleichen. Dieser Ansatz geht davon aus, dass der Arbitrator in jedem Fall vertrauenswürdig ist, dass die Informationen, die er von den Teilnehmern bekommt, korrekt sind und dass seine Informationsverarbeitungskapazität für einen schnellen Ablauf des Ressourcenallokations-Algorithmus ausreicht. Anforderungen an die Teilnehmer sind neben der Übermittlung von formalisierten Verhandlungspositionen sowie des jeweils erwarteten Nutzens die Teilnahme am Koordinationsverfahren des Arbitrators.

Diesem Verständnis von Koordination sind folgende Annahmen implizit:

- Es existiert eine zentrale Stelle, die jederzeit Risiken der globalen Zielerreichung erfassen kann.
- Diese Stelle besitzt die Autorität, jedes Subsystem zu einer Aktion zu zwingen.
- Die Subsysteme sind uneingeschränkt kooperativ, sowohl in der vollständigen Informationsweitergabe als auch in der Annahme und Bearbeitung von Aktionsaufträgen der zentralen Autorität.
- Die lokale Zielverfolgung durch Subsysteme ist der globalen Zielverfolgung unbedingt untergeordnet.

Als Beispiel einer derartigen "control architecture" in einer realen Anwendung beschreiben Norrie/Wunderli (1995) die Koordination von Aufgaben innerhalb eines CIM-Systems:

"Flexible and dynamic integration can be attained through a control layer which coordinates application tasks based on explicit inter-system dependencies. [...] Repository technology has been proposed as the basis for such a control layer. [...] The coordination activity is supervised by a central authority called the Global Coordinator." (Norrie/Wunderli 1995, S.190) und später "The Global Coordinator's main task is to maintain global consistency constraints between objects and object types of different subsystems" (Norrie/Wunderli 1995, S. 199).

Das Problem der Ressourcenallokation wird verstanden als die Zuweisung von passenden Ressourcen zu einer Menge von Aufgaben mit festgelegten Ausführungszeiten, mit der Nebenbedingung, dass keine Ressource gleichzeitig mehreren Aufgaben zugewiesen werden kann. In diesem Sinne ist Ressourcenallokation ein Beispiel eines kombinatorisch NP-vollständigen Problems. Der Koordinationsmechanismus ist innerhalb des *Global Coordinators* gekapselt, während die verteilten Agenten keine autonomen Verarbeitungsaufgaben haben. Diese Realisierung der Koordination in einer Instanz entspricht letztendlich auf den abstrakten Annahmen von Moulins "Gerechtigkeitsmodus". Im Sinne des Schichtenmodells können die wesentlichen Eigenschaften einer derartigen Realisierung so zusammenfasst werden:

- Eine Kommunikation zwischen den einzelnen Teilnehmern findet nicht statt. Jeder Teilnehmer kommuniziert zu Beginn des Mechanismus einmalig mit dem Arbitrator, um seine Präferenzen zu übermitteln. Am Ende des Mechanismus übermittelt der Arbitrator die ausgerechnete Ressourcenallokation zurück. In der Zwischenzeit wartet der Teilnehmer und unternimmt keine eigenen Aktionen.

- Die Kooperationschicht kann im "Gerechtigkeitsmodus" als nicht implementiert angesehen werden. Die Agenten sind notwendigerweise unbedingt kooperativ gegenüber dem Arbitrator, der außerdem ihr einziger Kommunikationspartner ist. Ein strategisches Verhalten der Agenten gibt es nicht.
- Der Allokationsmechanismus ermittelt am Ende des Berechnungsprozeß keine oder genau eine Lösung, es gibt weder Zwischenschritte noch Teilergebnisse. Diese Koordination ist statisch in dem Sinne, dass für die Teilnehmer zwischen Problemstellung und –lösung keine Zeit vergeht, d.h. dass es keinen für sie wahrnehmbaren dynamischen Prozess gibt. Gleichzeitig gilt, dass die Umgebung durch den Berechnungsvorgang des Arbitrator selbst nicht verändert wird.

Luce/Raiffa (1989) führen ein Arbitrationsschema ein, bei dem eine neutrale dritte Partei als Arbitrator auftritt. Käufer und Verkäufer übermitteln dabei ihre Präferenzen an den Arbitrator, der einen für beide akzeptablen Kompromiß berechnet. Wenn der Arbitrator als vertrauenswürdige Instanz (*trustbroker*) auftritt, wird per se ausgeschlossen, dass dieser die Präferenzen der Teilnehmer weitergibt, so dass Privatheit der Präferenzen und auch die Anonymität der Teilnehmer gewahrt wird.

Das Verfahren geht von einer Population miteinander handelnder Agenten aus. Eine Handelstransaktion beginnt mit zwei Agenten, die eine Transaktionsvereinbarung aus einer Menge alternativer Transaktionen erreichen wollen. Diese Menge wird "*negotiation set*" genannt. Jeder Agent übermittelt zu einem Zeitpunkt t_0 seine Präferenzen an den Arbitrator, d.h. zu jeder möglichen Transaktion aus der Menge denjenigen Nutzen, der ihm aus deren Durchführung entsteht. Im Allgemeinen kann davon ausgegangen werden, dass die Agenten in ihrer Nutzenbewertung nicht übereinstimmen, so dass ein Konflikt besteht. Die Aufgabe des Arbitrators ist es, die konfliktionären Interessen aufzulösen und zu entscheiden, welche Transaktionen ausgeführt werden und welche nicht. Der Allokationsmechanismus hat als Ziel, die Summe der Nutzen aller Marktteilnehmer zum Ausführungszeitpunkt t_1 zu maximieren (optimale Wohlfahrt). Dies bedeutet, dass der Arbitrator ausgehend von den ihm in t_0 übermittelten nutzenbewerteten Angeboten und Nachfragen eine Verteilung von überschneidungsfreien Kontrakten findet.

Das Gestaltungsziel solcher Arbitrationsschemata ist es, Konflikte so auflösen zu können, dass die Summe aller individuellen Nutzen maximiert wird und Agenten ihre wahren Präferenzen dem Arbitrator gegenüber offenbaren, ohne sie allen anderen Agenten ebenfalls verfügbar zu machen (Tesch/Fankhauser 1999, S.4).

Das einfachste Verfahren ist das Arbitrationsschema von Nash (Nash 1950). Dieses wählt aus der dem Arbitrator vorliegenden Transaktionsmenge zweier Agenten diejenige Transaktion

aus, bei der das Produkt der individuellen Nutzen der Beteiligten maximal ist. Unter der Annahme, dass beide Agenten ihre wahren Präferenzen offenbart haben, ist das Ergebnis pareto-optimal.

Das Verfahren geht von der Annahme öffentlich bekannter und wahrer Information aus, ist jedoch durch einfaches Lügen der Beteiligten in Bezug auf ihre Präferenzen kompromittierbar. Bei sogenannten "*greedy lies*" werden einfach diejenigen Alternativen, die für den Agenten besonders attraktiv sind, mit einem überhöhten Nutzen dargestellt, der nicht den wahren Werten entspricht. Um eine solche Vorgehensweise auszuschließen, schlagen Tesch/Fankhauser (1999, S. 6) eine Gewichtung der Nutzenangaben durch den Arbitrator vor, die im wesentlichen die Unterschiede der absoluten Höhe der Nutzenzuweisungen nivelliert. Damit wird das Problem abgeschwächt, jedoch nicht grundsätzlich zum Verschwinden gebracht. Insbesondere gibt es kein Mittel gegen sogenannte "*informed lies*". Ein Agent, der auch nur ein teilweises Wissen über die Präferenzen seines Gegenübers hat, kann seine Nutzenangaben gegenüber dem Arbitrator exakt so dimensionieren, dass die von ihm bevorzugte Alternative ausgewählt wird.

Tesch/Fankhauser (1999, S.7) schlagen zusätzlich ein nichtdeterministisches Auswahlverfahren des Arbitrators vor. Damit ist die konkrete Entscheidung des Arbitrators für den Agenten nicht vorhersehbar, so dass nicht mit Sicherheit gesagt werden kann, dass eine spekulative Nutzenangabe zu dem erwünschten Ziel führt. Auch der weniger über seinen Gegenspieler informierte Agent kann für ihn befriedigende Ergebnisse erzielen. Insgesamt führt dieses Vorgehen zu einem suboptimalen Ergebnis, da nicht jedes Mal diejenige Alternative mit dem maximalen Nutzen ausgewählt wird. Durch die immer noch vorhandene Gewichtung der Alternativen soll ein besseres Ergebnis als bei vollkommen zufälliger Alternativenwahl gewährleistet sein:

"With these characteristics, the probabilistic selection scheme has its strength in environments where the traders are well-aware of the opponent's preferences. Especially for asymmetric public information, it guarantees the less-informed agent fair outcomes" (Tesch/Fankhauser 1999, S. 8).

Generell wird beim Arbitrator von einem idealisierten Zustand ausgegangen, bei dem aus einer Menge an Agenten zufällige Paare gebildet werden, die mit ihrem jeweiligen "*negotiation set*" vom Arbitrator koordiniert werden sollen. Eine Erweiterung dessen ist die Einführung eines *Matchmakers*, der aus einer Menge an Agenten die jeweils besten zusammenführt und damit den Gesamtnutzen zu verbessern versucht. Dafür muß er für eine Population von i Käufern und j Verkäufern in einem Markt eine $i \times j$ Tabelle mit allen möglichen Kombinationen aufbauen. In jeder Zelle dieser Tabelle kann dann ein Arbitrationsschema angewandt werden,

welches den besten Nutzen aus der Paarung errechnet. Das von Tesch/Fankhauser (1999, S. 9) angewendete heuristische Verfahren beginnt mit einem zufällig ausgewählten Agenten und sucht für diesen denjenigen Partner, dessen Paarung den höchsten Nutzen hat. Beide Agenten werden zusammengeführt, anschließend aus der Suchtabelle entfernt und der Vorgang wiederholt. Nachdem so alle Paare gebildet worden sind, wird jeweils paarweise eine Arbitration durchgeführt. Als Vorteile des Verfahrens wird die höhere Flexibilität in Bezug auf Katalogrealisierungen und die höhere Robustheit gegen die spekulative Präferenznennung durch die Agenten angeführt.

Eine Diskussion der Eignung von Arbitrationsverfahren für die Koordination in offenen Märkten orientiert sich an den Charakteristika von Koordinationsproblem und -mechanismus, die in Kapitel 1 aufgestellt wurden. Bei den Charakteristika des Problems (ubiquitäre Kooperation, kein benevolentes Handeln, begrenzte Rationalität der Entscheidung) handelt es sich um Eigenschaften, die eine realweltliche Marktsituation beschreiben und für die ein leistungsfähiger Marktmechanismus gesucht wird. Bei den Charakteristika des Mechanismus (Verfolgung ökonomischer Prinzipien, schnelle zeitnahe Ausführung, Privatheit, Skalierbarkeit, Adaptivität und Dezentralität) handelt es sich hingegen um Eigenschaften der formalen Lösung, d.h. des Marktmechanismus selbst.

Bezüglich der **Ausführungsgeschwindigkeit** und der **Skalierbarkeit** läßt sich folgendes festhalten. Das o.a. zentrale Verfahren besitzt zur exakten Berechnung der maximalen Nutzen-summe eine Komplexität, die sich für n Agenten zu $O(2^n)$ verhält und den Formalismus praktisch nur für kleine n anwendbar macht. Es kann in zentraler Form nicht mit einem deterministischen Algorithmus in polynomialer Zeit gelöst werden, so dass ein Koordinationsmechanismus des "Gerechtigkeitsmodus" für realistische Teilnehmerzahlen in offenen Systemen nicht implementierbar ist. In ähnlichen Anwendungen (z.B. der Produktionsplanung) werden meist Heuristiken angewendet, die auf die Erreichung einer optimalen Allokation verzichten, aber eine je nach Definition zufriedenstellende Allokation gewährleisten können. Die Qualität der Lösung kann im vorliegenden Fall nicht überprüft werden und hängt stark von den nicht-deterministischen Aspekten, d.h. von der zufälligen Auswahl der Agentenpaare und der gewichteten Zufallsauswahl der schließlichen Transaktion ab.

Dadurch wird zweitens jegliche **Adaptivität** der Agentenstrategie an zukünftige Entscheidungen in Frage gestellt. Der Arbitrator führt alle anstehenden Transaktionen unter der Fiktion der Gleichzeitigkeit durch. Damit ergibt sich eine Reaktion auf Angebot und Nachfrage erst nach der Durchführung aller Transaktionen und die Leitfunktion der Preisveränderung in Bezug auf die Ressourcenknappheit wird unterschlagen. Eine Rückmeldung an die einzelnen Teilnehmer, inwiefern ihre Nutzenangabe und damit ihre Handelsstrategie den Lösungsalgorithmus bzw.

ihr Ergebnis beeinflusst hat, findet nicht statt. In diesem Fall ergibt sich kein Anlass für die Agenten, in Zukunft ihre Nutzenbewertungen zu verändern.

In der Frage der **Echtzeitverarbeitung** und der **Ausführungsgeschwindigkeit** gilt, dass die Annahme der Ökonomie, das Allokationsverfahren in unendlich hoher Geschwindigkeit und damit praktisch sofort durchführen zu können, in der Realisierung in Computersystemen nicht umgesetzt werden kann. Eine suboptimale Ressourcenallokation zum Zeitpunkt t stellt das Problem dar. Die Berechnung einer besseren Ressourcenallokation dauert eine Zeit Δt , so dass eine Lösung des ursprünglichen Allokationsproblems erst zum Zeitpunkt $t' = t + \Delta t$ vorliegt. Die Anwendbarkeit der Lösung hängt jetzt davon ab, dass zwischen den Zeitpunkten t und t' keine Veränderung der Ausgangssituation passiert ist. In einer ArbitratorRealisierung findet keine Veränderung der Ausgangssituation statt, da der zentrale Arbitrator entweder die einzige Instanz mit direkten Veränderungsbefugnissen ist oder die einzelnen Agenten dessen Befehle zur Veränderung ausführen, nachdem der Berechnungsvorgang abgeschlossen wurde. Das Problem selbst wird umgangen und muss auf einer nicht-technischen Ebene gelöst werden (z.B. durch Vereinbarungen, dass alle Agenten auf ein Signal des Arbitrators warten, bevor sie mit ihrer Verarbeitung fortfahren.)

Bezüglich der **Verfolgung ökonomischer Prinzipien** und der Möglichkeit der **ubiquitären Kooperation** wird ein am Allokationsverfahren teilnehmender Agent nicht in jedem Fall seinen Eigennutz befriedigen können. Für ihn und jeden anderen Agenten muss die Summe aus Kosten des Arbitrators und Opportunitätskosten des eigenen Verkaufens geringer sein als der Nutzen aus den Aktionen des Arbitrators. Wenn Agenten an der Effektivität der Arbitratorlösungen Zweifel haben, werden sie an diesem vorbei bilaterale Lösungen vorziehen und die Effektivität des Arbitrators weiter beschneiden, so dass dieser schließlich nutzlos wird. In den in dieser Arbeit thematisierten offenen, elektronischen Märkten gibt es immer die Möglichkeit für die Agenten, bilaterale Verhandlungen miteinander aufzunehmen. Dies wird von ihnen dann in Angriff genommen, wenn der durch die Koordinationsleistung des Arbitrators erreichte Nutzen geringer als der prognostizierte Nutzen jeder anderen Lösung ist. Der mit der Leistung des Arbitrators nicht zufriedene Agent wird in der Zukunft nach bilateralen, spekulativen Lösungen suchen (vgl. Rasmusson/Janson 1999).

Schließlich wird die **Privatheit** des Agenten nicht beachtet. Die Nutzenfunktion eines Agenten ist sein wertvollster Besitz. In Kenntnis der Nutzenfunktion eines Verhandlungspartners können Preisober- und -untergrenzen berechnet werden. Die Verhandlungsstrategie wird fundamental ebenfalls durch die Nutzenfunktion determiniert. Ein ökonomisches, strategisches, spekulatives Handeln ist für einen Agenten in dem Moment nicht mehr möglich, in dem seine Nutzenfunktion anderen Teilnehmern bekannt wird. Aus dieser Sichtweise darf der Agent nie

seine Nutzenfunktion publizieren, genausowenig, wie er bei asymmetrischen Verschlüsselungsverfahren einen privaten Schlüssel herausgeben würde. Neben der Geheimhaltung der Nutzenfunktion ist es in bestimmten Fällen auch nötig, die Anonymität des Prinzipals des Agenten sicherzustellen:

"Anonymous trading in stock markets is considered to increase the liquidity, especially since large traders can participate incognito. Large traders, if identified, risk altering the prices in a for them negative way by trading" (Rasmusson/Janson 1999, S. 148).

Vollkommene Anonymität kann es nicht geben, da die Unabstreitbarkeit der Transaktion sichergestellt werden muß. Auch ein Lernen aus dem vergangenen Verhalten eines Agenten, im Sinne eines Reputation Tracking, ist nur möglich, wenn der Agent selbst eine eindeutige und unveränderliche Identität besitzt bzw. einem Prinzipal eindeutig zugeordnet werden kann. Im Umkehrschluß kann bereits diese Eigenschaft in die Nutzenfunktion der handelnden Agenten einfließen, d.h. das von vornherein nur mit solchen Agenten gehandelt wird, deren Identität bekannt ist oder überprüft werden kann. Schließlich sollte die Privatheit eines Agenten in der Form gewährleistet sein, dass die Informationen über ihn nicht an einer zentralen Stelle gebündelt gespeichert und abgerufen werden können. In diesem Fall wäre es möglich, ohne großen Aufwand seine Präferenzen zu rekonstruieren, indem seine Vorgehensweise in historischen Verhandlungen mit unterschiedlichen Partnern, deren Präferenzen ebenfalls annähernd bekannt sind, analysiert wird. Als Fazit dürfen diese Informationen entweder nicht an einer zentralen Stelle gespeichert sein oder diese Stelle verpflichtet sich, die Informationen weder in diesem Sinne zu nutzen noch sie weiterzugeben, um eine strikte Neutralität zu wahren. Im "Gerechtigkeitsmodus" kann eine zentrale Berechnung durch den Arbitrator nur dann erfolgen, wenn diesem sämtliche Nutzenfunktionen aller Beteiligten bekannt sind und die Agenten ihre privaten Präferenzen dem Arbitrator offenlegen. Diese Information ist für Konkurrenten wie für Verkäufer sowohl für die gegenwärtige als auch für zukünftige Situationen extrem wertvoll. Der Arbitrator verletzt in hohem Maße die Anforderungen an die Geheimhaltung und kann nur dort eingesetzt werden, wo seine Neutralität gesichert ist und er Bestechungsversuchen der anderen Teilnehmer unbedingt widerstehen kann. Anderenfalls werden die Agenten nicht mehr ihre wahren Präferenzen offenbaren, sondern spekulativ handeln, wodurch die grundlegende Annahme der gerechten Nutzenverteilung (daher ja die Bezeichnung: "Gerechtigkeitsmodus") wegfällt.

Der Einsatz des Arbitrators ist konsequenterweise in solchen Umgebungen sinnvoll, in denen sich die relevante Umwelt bzw. die Menge der Teilnehmer qualitativ und quantitativ nur wenig verändert. Insgesamt kann die Realisierung von Koordination mit Hilfe zentraler Arbitration nur in geschlossenen, möglichst kleinzahligen Multi-Agenten-Systemen positiv bewertet wer-

den. Ein Schlüssel zu einem verbesserten Konzept ist die Verlagerung der Berechnungs- und Bewertungsfunktion auf die einzelnen Teilnehmer, die jeweils nach ihren eigenen lokalen Entscheidungsregeln über ihre nächste Aktion entscheiden.

2.4 Der Markt als dezentraler Koordinationsmechanismus

Die Autonomie der Teilnehmer wird in Koordinationsmechanismen des "Modus dezentralen Verhaltens" wesentlich stärker in den Vordergrund gestellt als in denjenigen des "Gerechtigkeitsmodus". Während in letzterem der einzelne Teilnehmer mit der Allokation durch den Arbitrator zwangsweise zufriedengestellt wurde, agiert und reagiert der einzelne Agent im "Modus dezentralen Verhaltens" in einem dynamischen Verfahren selbständig auf Veränderungen seiner Umwelt. Diese Umwelt konzentriert sich in einem zentralen Mediator, der sämtliche Informationen sammelt und den Agenten in kooperativer Form zur Verfügung stellt. In Bezug auf das Schichtenmodell der Telematik charakterisieren den "Modus dezentralen Verhaltens" folgende Eigenschaften:

- Eine Kommunikation zwischen den einzelnen Teilnehmern findet nicht statt. Jeder Teilnehmer kommuniziert ausschließlich mit dem zentralen Mediator.
- Die Kooperation zwischen Teilnehmer und Mediator ist von Vertrauen geprägt. Der Mediator selbst verfolgt kein eigenes Ziel in der Koordination, sondern sammelt die ihm von den autonom entscheidenden Teilnehmern gesendeten Informationen und stellt sie (meist aggregiert) diesen entweder wieder zur Abfrage zur Verfügung oder sendet sie ihnen unaufgefordert zu. Er muß als vertrauenswürdige und (im Rahmen der ihm zur Verfügung gestellten Informationen) objektive dritte Partei angesehen werden.

Ausgangspunkt der Koordination ist eine nicht-optimale Ressourcenallokation. Der dynamische Koordinationsprozeß nähert sich über (auch simulierte) Zwischenlösungen im Zeitablauf einer optimalen Lösung an. Die Präferenzen der Teilnehmer verändern sich im Zeitablauf und werden dem Mediator anschließend kommuniziert, was den Prozessverlauf wiederum verändert. Es ergibt sich ein in der Funktion zweigeteiltes Verfahren mit zentraler Informations- und Ergebnisbereitstellung, aber dezentraler Entscheidungskapazität. Das Konzept des Walras'schen Auktionators ist ein ökonomisches Verfahren, das einen derartigen Koordinationsmechanismus in arbeitsfähiger Detailtreue expliziert.

2.4.1 Der Walras'sche Auktionator und Market-Oriented Programming

Nach dem Erscheinen von Adam Smith' "Reichtum der Nationen" 1784, in welchem das Konzept der "unsichtbaren Hand" aufkam, schuf Leon Walras 1874 die erste konkrete Beschreibung eines ökonomischen Allokationsmechanismus. Anders als in der Intention von Smith ist die "unsichtbare Hand" hier sichtbar geworden und in einem zentralen Auktionator manifestiert. Der offensichtliche Widerspruch, der zwischen Realität und Walras'schem Auktionator liegt, lässt sich historisch begründen, dass die ökonomische Fragestellung nach den Eigenschaften des allgemeinen Gleichgewichts forschte und der Prozeß zur Erreichung des Gleichgewichts in den Hintergrund rückte. Mit den Worten von Varian:

"Im Großteil der Volkswirtschaftslehre verwenden wir einen Rahmen, der aus den beiden folgenden einfachen Grundprinzipien besteht:

- 1. Das Optimierungsprinzip – die Leute/Agenten versuchen das beste Konsummuster zu wählen, das sie sich leisten können.*
- 2. Das Gleichgewichtsprinzip – die Preise passen sich solange an, bis die Menge, welche die Leute/Agenten von irgendetwas nachfragen, gleich der angebotenen Menge ist" (Varian 1991, S.2)*

und weiter:

"Es ist zumindest vorstellbar, dass zu einer bestimmten Zeit Nachfrage und Angebot der Leute nicht übereinstimmen und sich daher etwas ändern muß. [... Aber der] Gleichgewichtspreis interessiert uns, nicht jedoch wie der Markt zu diesem Gleichgewicht kommt, oder wie es sich über längere Zeiträume hinweg verändert" (Varian 1991, S.3).

Genau diese Vernachlässigung des Effekts gegenüber dem Mechanismus macht die Ableitung des Mechanismus aus den vereinfachenden Annahmen der neoklassischen Ökonomik äußerst schwierig.

Der Preisanpassungsprozeß wurde von Leon Walras (1874, bzw. Arrow/Hahn 1971) als "tâtonnement" bezeichnet. Agenten (d.h. Produzenten und Konsumenten) reagieren mit ihren Kauf- und Verkaufswünsche dabei auf Preissignale für die einzelnen Güter. Die Interaktionen der Agenten werden von einem zentralen Auktionator koordiniert, der das generelle Preisniveau in Richtung auf einen Angebot-Nachfrage-Ausgleich verschiebt und dabei Interimspreise signalisiert, um Antworten von den Agenten zu erhalten. Das Verfahren funktioniert im Detail so, dass es zuerst für jedes Gut einen Auktionator gibt. Alle Gütermärkte werden zufällig in eine Reihenfolge gebracht. Zu Beginn des Prozesses senden die Agenten dem Auktionator des

ersten Gutes ihre mehr oder weniger willkürlichen Preise, die Kauf oder Verkauf dieses Gutes anzeigen. Die Überschußnachfrage, d.h. diejenige Nachfragemenge, die das Angebot übersteigt, ergibt sich dann als positiv, null, oder negativ. Der erste Markt wird ins Gleichgewicht gebracht, indem der Preis so gesetzt wird, dass Angebot und Nachfrage übereinstimmen. Dieser Ausgleich wirkt sich über die Budgetrestriktionen der Agenten auf die Nachfrage in den anderen Gütermärkten aus. Als nächstes wird sukzessive der Preis auf allen anderen Märkten so gesetzt, dass dort ebenfalls jeweils Angebot und Nachfrage ausgeglichen sind, was als Markträumung bezeichnet wird. Am Ende jeder Runde ist so lediglich der jeweils letzte Markt im Gleichgewicht, alle übrigen werden durch diesen Ausgleich so beeinflusst, dass auch bereits als geräumt berechnete Märkte wieder ins Ungleichgewicht überführt werden. Nach Walras' Argumentation ist aber der Einfluß anderer Gütermärkte immer geringer als die Markträumung auf dem gerade betrachteten Gütermarkt selbst. Insofern sollte es ein iteratives Vorgehen geben, welches die Summe der Ungleichgewichte durch sukzessive Markträumung minimiert. Dieses Vorgehen wurde von Samuelson (1947) mathematisch formuliert und in einigen Aspekten verändert. In Samuelsons Version erhalten die Agenten erst ein Preissignal und melden dann ihre Überschußnachfrage zurück. Der Auktionator paßt die Preise inkrementell im Verhältnis zur Größenordnung der Überschußnachfrage an und meldet diese an die Agenten zurück. In jeder Runde berechnen die Agenten ihre Überschußnachfrage neu und melden diese an den Auktionator zurück. Der Prozeß wird solange fortgeführt, bis die Preise zum stabilen Gleichgewicht konvergieren.

Wesentliche Merkmale des Walras'schen Auktionators sind (Gravelle/Rees 1992, S. 254-258):

- Die Wirtschaftssubjekte erfahren den Marktpreis ausschließlich durch den zentralen Auktionator, es gibt also keine Kommunikation zwischen den Agenten.
- Der Nachfragepreis richtet sich proportional nach der Angebotsmenge und umgekehrt. Keine weiteren Merkmale werden einbezogen. Insbesondere haben die Erwartungen der Teilnehmer keinen Einfluß auf den Preis.
- Handel findet erst statt, wenn der Auktionator zu einem Gleichgewicht gefunden hat und ein Gleichgewichtspreis eingestellt ist. Dieser Preis wird den Teilnehmern umsonst zur Verfügung gestellt, d.h. den Teilnehmern entstehen keine Suchkosten. Es gibt keinen Handel zu unterschiedlichen Preisen.

Die Nutzung des "tâtonnement"-Preismechanismus zur Koordination in Mehragenten-Systemen ist von Wellman unter dem Begriff Market-Oriented Programming (MOP) eingeführt worden:

"In market-oriented programming we take the metaphor of an economy computing a

multiagent behavior literally, and directly implement the distributed computation as a market price system. That is, the models, or agents, interact by offering to buy or sell commodities at fixed unit prices. When this system reaches equilibrium, the computational market has indeed computed the allocation of resources throughout the system, and dictates the activities and consumptions of the various modules" (Wellman 1995, S.76).

Wellman geht von einem neo-klassischen Ökonomiebegriff aus, in welchem dem Konzept eines Gleichgewichtspreises eine hohe Bedeutung zukommt:

"In market-oriented programming, we take general equilibrium to be the gold standard solution, but admit partial equilibrium as one way to trade solution quality for computational efficiency" (Wellman 1995, S.75).

MOP ist nach Wellman *"a methodology for solving particular distributed control problems, but also a generic programming paradigm for the development of distributed systems based on these methods. By a programming paradigm, I mean that the exercise of defining a computational market leads to the specification of a procedural solution to the underlying allocation problem facing that market" (Wellman 1995, S.74).* Die Verbindung von der reinen informatischen Realisierung zur ökonomischen Konzeption und Analyse findet über die ökonomische Fragestellung des *"mechanism design"* (Varian 1995) statt:

"Within economics, the problem of synthesizing an interaction protocol via which rational agents achieve a socially desirable end is called mechanism design. This is exactly the problem we face in designing distributed software systems, which suggests an opportunity exploit existing economic ideas" (Wellman/Wurman 1997, S.120).

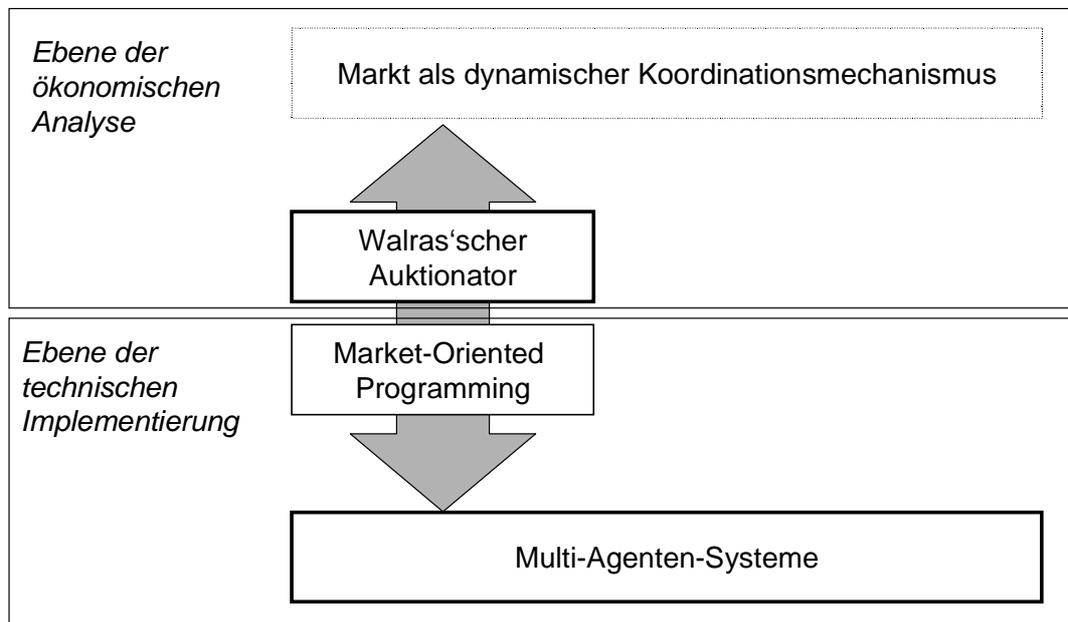


Abb. 6: Market-Oriented Programming

Wellman nennt folgende Vorteile von MOP gegenüber nicht-marktlichen Ansätzen der Koordination:

- Der marktbasierter Ansatz führt immer zu pareto-optimaler Allokation unter der Bedingung, dass *"preferences and technologies are monotone, smooth, and convex"*. Darüber hinaus kann gezeigt werden, dass Pareto-Effizienz sogar eine untere Grenze für die Qualität der Problemlösung darstellt (Wellman 1995, S.77).
- Die Dekomposition der Problemlösung führt zu einer vergleichbaren Komplexität wie bei kooperativen Lösungen. Die Dezentralisierung der Aufgaben wird aber durch die Preisbildung unterstützt. Zum einen kapseln die Preise sämtliche Information: *"the preference, capabilities, and even existence of other agents is separable from the agent's decision problem"* (Wellman 1995, S.77). Zum anderen kann für jedes Gut (in einem Markt, auf dem mehrere Güter gehandelt werden) separat ein Markträumungspreis berechnet werden, so dass jeder Auktionator auf dem Markt nur mit wenigen Dimensionen umgehen können muß.
- Die größten Vorteile werden im Design verteilter Systeme gesehen. Dazu gehört u.a. die inhärente Modularität durch Aufteilung der Berechnungskapazität auf solche Agenten, *"which has a well-scoped set of capabilities (technology) or resources (endowment), and a well-defined and limited objective to maximize (profits or utility)"* (Wellman 1995, S.78). Weiterhin finden alle Interaktionen zwischen Agenten als Handelsprozesse statt,

vermittelt durch einen allgemeinen Marktmechanismus. Drittens kann das System selbst mit Hilfe bekannter neo-klassischer Analysemethoden beschrieben werden. Die Tatsache, dass ein Gleichgewicht existiert und dieses vom Auktionator gefunden wird, kann unter Annahme bestimmter Systemeigenschaften bewiesen werden. Unter zusätzlichen Annahmen¹⁷ ist dieses Gleichgewicht stabil und der Preisanpassungsprozeß konvergiert auf dieses Gleichgewicht hin.

Wellman's Implementation von Market-Oriented Programming ist das Projekt WALRAS, "*essentially an object-oriented implementation of general equilibrium theory*" (Wellman 1995, S.78). Alle Agenten innerhalb von WALRAS werden von zwei Grundklassen abgeleitet, Konsumenten und Produzenten. Die Agenten in WALRAS interagieren ausschließlich über ein verteiltes "*bidding protocol*". WALRAS nutzt einen Auktionator, der von allen Anbietern und Nachfragern Informationen über das jeweilige Angebot und die Nachfrage einholt. Sowohl die Konsumenten als die Produzenten-Agenten bieten mit dem Ziel, Nutzen bzw. Gewinn zu maximieren, abhängig von ihren Möglichkeiten und Budgetgrenzen und der Annahme des Wettbewerbs zwischen allen Teilnehmern. Jeder Agent sendet ein Gebot für jedes Gut, welches ihn interessiert, an den Auktionator. Die Gebote sind Nachfragefunktionen, die die Menge für jeden möglichen Güterpreis spezifizieren, unter der Annahme dass die Preise aller übrigen Güter mit ihren derzeitigen Werten fixiert sind. Neue Markträumungspreise werden berechnet, wenn die Gebote verändert werden. Der Auktionator benachrichtigt betroffene Agenten über relevante Veränderungen. In der Rückmeldung an die Anbieter und Nachfrager wird der errechnete Gleichgewichtspreis mitgeteilt, zu dem in der nächsten Runde der Handel stattfinden wird.

Als Ergebnisse aus dem Einsatz von WALRAS in verschiedenen Anwendungen führt Wellman (1995, S.87ff.) folgende "*lessons learned*" an:

- "*Our first lesson is that the most important design parameter of a computational economy is the set of goods.*" Von den drei Gestaltungsmerkmalen der Kooperationsschicht (Güter-, Ziel- und Strategiedefinition) ist die Güterdefinition am wichtigsten und die Definition der Ziele und Strategien der Agenten folgt aus dieser.
- "*Avoid saturation of utility.*" Wenn der Nutzen der Agenten befriedigt werden kann, ohne dass diese ihr komplettes Budget ausgeben müssen, gibt es Indifferenzen in Bezug auf die Nachfragestrategie und die Eindeutigkeit der Lösung geht verloren. Der Markt als

17 Insbesondere "gross substitutability" – die Eigenschaft, dass die Preiserhöhung für ein Gut keine Nachfrageverminderung für ein anderes Gut nach sich zieht (Wellman 1995, S. 78).

Allokationsmechanismus ist dort sinnvoll eingesetzt, wo knappe Ressourcen entsprechend ihres Nutzens eingesetzt werden können. Ohne Knappheit von Ressourcen geht diese zwingende Voraussetzung verloren.

- *"Producers do all the work"*. Der Zweck von MOP ist die Berechnung von Aktionen, d.h. die Entscheidung, welche verteilten Ressourcen von wem genutzt werden sollen. Um eine Aktionsauswahl zu modellieren, wird ein Produzent definiert, dessen Technologie aus diesen Aktionen besteht. Damit entsteht der Eindruck, dass die Konsumenten keine funktionale Bedeutung in der Realisierung haben. Tatsächlich bewerten die Konsumenten über ihre Nutzenfunktion aber die Aktionen und bilden eine Rangordnung.
- *"Privatization averts commons tragedies"*. Die Übernutzung gemeinsamer Ressourcen (*tragedy of the commons*) stellt ein häufig auftretendes Problem in verteilten Systemen dar (Turner 1993). Indem diese Ressourcen ebenfalls von Agenten repräsentiert werden, werden diese Externalitäten internalisiert und die Lösung verbessert sich von einem suboptimalen hin zu einem systemweiten Gleichgewicht (Wellman 1995, S. 88).
- *"Arbitrageurs increase decentralization"*. Das Einführen von Händleragenten dezentralisiert die Problemlösung. Neben dem zentralen Auktionator gibt es viele Händler, die im wesentlichen das gleiche Produkt anbieten: *"The fact that all agents interested in a particular good must communicate with a central auction for that good may seem like a strong centralization constraint. With arbitrageurs, one can decentralize the communication arbitrarily by defining multiple versions of the same good"* (Wellman 1995, S. 89). Agenten können dann irgendeine Version des Produktes einkaufen; die Technologie der Arbitrageure ist die Transformation einer Version in eine andere, womit ein identischer Preis für alle Versionen festgelegt wird. Das Gleichgewicht wird dadurch nicht beeinflusst.

Das Verfahren wurde bisher u.a. eingesetzt für die Gestaltung von Elektrizitätsmärkten (Ygge 1998), Multi-Commodity Flow Problems (Wellman 1993) oder für Verhandlungen über die Dienstqualität in Multimedia-Netzwerken (Yamaki et al. 1996). Für eine Diskussion über die Einsatzmöglichkeiten von MOP in dem Szenario dieser Arbeit sei auf das Kapitel 2.4.3 "Auktionsverfahren" auf Seite 56 verwiesen.

2.4.2 Der Ansatz der "Agoric Open Systems"

Einen anderen Ansatz, mit Hilfe von preisbasierten Auktionen Ressourcenallokation in Computersystemen zu betreiben, stellen die „Agoric Open Systems“ dar, die bei Betriebssystemen Ressourcen (z.B. Prozessorzeit, internen und externen Speicherplatz) zuweisen sollen. Die

Grundidee der Agoric Open Systems ist wie auch beim Market-Oriented Programming, " [to examine] markets as a model for computation and [to propose] a framework – agoric systems – for applying the power of market mechanisms to the software domain" (Miller/Drexler 1988, S. 133).

"An agoric system is defined as a software system using market mechanisms, based on foundations that provide for the encapsulation and communication of information, access and resources among objects" ((Miller/Drexler 1988, S.135).

In dem Artikel von Miller/Drexler von 1988 wird die Nutzung von Hypertext-Systemen und objektorientierter Programmierung propagiert. Die heutige Verbreitung und tatsächliche Anwendung entsprechender Technologie hat diese Grundlagen bereits z.B. auf Basis von HTML, Internet und Java realisiert. Die von Miller/Drexler angesprochenen informatischen Konzepte (z.B. das Betriebssystem MACH, das Hypertextsystem XANADU oder die Programmiersprache ACTOR) sind bereits überholt. Im folgenden soll daher stärker das Konzept und weniger die technische Realisierung beschrieben werden.

Das zentrale Anwendungsbeispiel geht von der Preisstellung der Inanspruchnahme von Maschinenressourcen (Speicherplatz, Prozessorzeit usw.) für Softwareobjekte aus: *"Extending encapsulation to include computational resources means holding each object accountable for the cost of its activity."* (ebda., S.135) und *"In agoric systems, objects will charge each other and the machine will charge the objects."* (ebda., S. 148) Diese Softwareobjekte wiederum stellen die ihnen zugerechneten Kosten wiederum denjenigen Objekten (oder menschlichen Anwendern) in Rechnung, von denen sie selbst als Ressource in Anspruch genommen werden, möglicherweise mit einem Gewinnaufschlag. Der Endanwender zahlt letztendlich für den insgesamt angefallenen Ressourcenzugriff, den er durch die Abarbeitung seiner Problemstellung (z.B. Datenbankabfrage, Preiskalkulation) getätigt hat.

Bei den Agoric Open Systems verhandeln Softwareobjekte marktähnlich um Computerressourcen. Aus dem Erfolg bei diesen Verhandlungen können zwei Nutzen gezogen werden: erstens werden zur Laufzeit des Systems Softwareobjekten dynamisch und bedarfsabhängig Ressourcen zugewiesen, ohne dass ein zentraler Steuerungsmechanismus eingreifen müsste, zum anderen wird mit Blick auf den Designvorgang eine Bewertung wiederverwendbarer Softwarekomponenten durchgeführt.

Nach Miller/Drexler ist ein erwünschter Nebeneffekt der Kapselung von Objekten, dass *"motivated by the need for decentralized planning and division of labor, computer science has reinvented the notion of property rights"* (ebda., S.138), was wiederum die Anwendung ökonomisch motivierter Mechanismen rechtfertigt.

Diese Ressourcenallokation in Form einer *"spontaneous order"* (Miller/Drexler 1988, S.139) innerhalb von Rechnersystemen wird als grosser Vorteil gegenüber zentralen Mechanismen gesehen, wenn von einer Welt ausgegangen wird, welche *"is becoming populated with hyper-cubes, Connection Machines, shared-memory multi-processors, special-purpose systolic arrays, vectorizing super-computers, neural-net simulators, and millions of personal computers. More and more, these are being linked by local area networks, satellites, phones, packet radio, optical fiber, and people carrying floppy disks"* (ebda., S. 139). In direkter Fortsetzung dieser Aufzählung hat das Internet diese Vision bereits zu grossen Teilen verwirklicht.

Auch bei Agoric Systems stellt sich die Frage, wie die einzelnen Softwareobjekte aussehen müssten, um erfolgreich in einem Marktsystem arbeiten zu können: *"explicit attention must be paid to the question of the minimal competence and complexity necessary for an object to participate in a market system"* (ebda., S. 141). Hier ergibt sich mit anderen Worten die Frage, wie einfach Agenten und Mechanismen aufgebaut sein müssen, damit sich tatsächlich ein Markt bildet. Diese Frage wurde später von Gode/Sunder (1993) und Cliff/Bruten (1998) unter dem Begriff der Zero-Intelligence Agents wieder aufgenommen. Agoric Systems stellen hingegen idealisierte Märkte dar, da von seiten der Designer andere Gesetze angewendet werden können:

"Such a market is idealized because certain complicating characteristics of real markets are excluded. The basic rules of human markets are typically encoded in legal systems and enforced by attempting to catch and punish violators" (Miller et al. 1995, S. 100).

Als die Realisierung vereinfachende Unterschiede zu menschlichen Märkten werden folgende Charakteristika angeführt:

- Diebstahl von Präferenzen kann durch Kapselung in der objektorientierten Software implizit verhindert werden. Nicht-fälschbare Währung und Warenzeichen werden über eine interne Public-Key-Infrastructure implementiert. Das gilt nur, wenn das Agoric System keine Verbindung zur "realen Welt" hat, d.h. wenn es ein geschlossenes System ist.
- Negative Externalitäten werden vermieden, d.h. die Übernutzung öffentlicher Güter (tragedy of the commons, vgl. Turner 1993). Das kann nach Ansicht von Miller et al. (1995) in Agoric Systems vermieden werden, indem die öffentlichen Güter ebenfalls durch Agenten repräsentiert und damit internalisiert werden. Wellman (1995) hat dieses Argument später für Market-Oriented Programming aufgegriffen (s.o.).

- Software-Agenten sind keine Menschen, *"hence they are not hurt when they go broke"* (Miller et al. 1995, S. 100). Das ist physikalisch richtig, aber der wirtschaftliche Ausfall des Agenten trifft den Prinzipal genauso, als wenn er selbst handeln würde.

Miller/Drexler (1988) sowie Miller et al. (1995) nehmen in ihrem Konzept der Agoric Open Systems vieles von dem Konzept der vorliegenden Arbeit voraus, behandeln neben dem hier angesprochenen Ressourcenallokationsproblem aber ebenso die Implikationen des Konzepts u.a. für Softwareentwicklung¹⁸, Sicherheit, Nutzungsabrechnung von Software und die Verknüpfung zu Konzepten menschlicher Intelligenz in verteilten Systemen (vgl. Minsky 1981). Durch die Beschränkungen der 1988 zur Verfügung stehenden Technologie wurde das Konzept nur experimentell umgesetzt und nicht in einem konkreten Anwendungsfall weiterentwickelt. Für den vorliegenden Fall eignen sich die Agoric Open Systems auch deswegen nicht, weil die Systementwickler als unbedingt kooperativ angenommen werden. In einem elektronischen Markt, dessen Agenten von Menschen definiert werden, kann dies keinesfalls vorausgesetzt werden.

2.4.3 Auktionsverfahren

Die mathematische Formulierbarkeit und die einfache Logik des Konzepts haben dazu geführt, dass nahezu alle Realisierungen der Ressourcenallokation in Märkten einen Auktionator vorsehen.

"Auctions provide efficient, distributed and autonomy preserving ways of solving task and resource allocation problems in computational multiagent systems" (Sandholm 1998, S. 1).

Ein wesentliches Argument ist, dass Auktionen normativ untersucht werden können. Welche Strategien für bestimmte Situationen am besten sind und welche Eigenschaften das Ergebnis in diesem Fall besitzt, z.B. optimale Wohlfahrt oder eine effiziente Allokation, kann analytisch und theoretisch begründet werden. Auktionen sind jedoch nur eine Form des wirtschaftlichen Handelns. Als alternative Realisierungen können insbesondere noch bilaterale Verhandlungen und Kataloge angeführt werden:

"Auctions are just one form of business negotiations. Other examples are competitive bids for procurement, brokerages/exchanges/cartels, and two-party negotiations. While these negotiation techniques appear to be fundamentally different from each other, from

18 "Agoric systems can serve as a software publishing and distribution marketplace providing strong incentives for the development of reusable software components" (Miller/Drexler 1988, S. 134).

the business process modeling or work flow point of view, they can be strikingly similar"
(Kumar/Feldman 1998, S.2).

Auktionen werden im Internet zwischen menschlichen Agenten in einer Vielzahl von Anwendungsfeldern eingesetzt. Im Bereich des *Agent-mediated Electronic Commerce* (Guttman et al. 1998) setzen die Projekte FM100 (Gimenez et al. 1998) und Auctionbot (Wurman et al. 1998) z.B. jeweils ein zentrales Auktionshaus mit standardisierten Protokollen ein, welches von Software Agenten mit parametrisierbaren Bietstrategien genutzt werden kann.

Der bekannteste Einsatz computergestützter Auktionsverfahren findet an Wertpapierbörsen statt. Im folgenden soll eine beispielhafte Implementation eines Auktionsverfahrens für Rentenpapiere mittels eines Multi-Agenten-Systems vorgestellt werden.

Der an einer Wertpapierbörse verwendete Koordinationsmechanismus wird als *Marktmodell* bezeichnet: *"Das Marktmodell definiert den Mechanismus, auf welche Weise Orders zu Geschäftsabschlüssen im börslichen Handelssystem zusammengeführt werden. Es beschreibt die Preisermittlung, die Priorisierung börslicher Orders sowie die Art und den Umfang der Informationen, die den Marktteilnehmern während der Handelszeit zur Verfügung gestellt werden"* (Deutsche Börse 2000, S. 5). Dabei kann zwischen einer Vor-, Haupt- und Nachhandelsphase unterschieden werden. Die Marktteilnehmer haben lediglich in der Vor- und Nachhandelsphase die Möglichkeit der Interaktion mit dem Auktionator, während in der Haupthandelsphase eine Ressourcenallokation für ein festgehaltenes Problem gesucht wird. Die Haupthandelsphase kann als Auktion oder fortlaufender Handel implementiert sein.

Der fortlaufende Handel (Deutsche Börse 2000, S. 19ff.) beginnt mit einer Eröffnungsauktion, kann durch eine oder mehrere untertägige Auktionen unterbrochen werden und endet mit einer Schlussauktion. Die Eröffnungsauktion unterteilt sich in die Aufruf-, Preisermittlungs- und Marktausgleichphase. Ihr Zweck ist die Vermeidung eines "gekreuzten Orderbuches" zu Beginn der Haupthandelsphase, d.h. das Vorliegen von Geboten mit höheren Kauf- als Verkaufspreisen. Diese Gebote werden bereits hier einander zugeordnet.

In der Aufrufphase interagieren die Marktteilnehmer mit dem zentralen Auktionssystem und geben ihre Gebote in das zentrale Orderbuch ein. Die Preisermittlungsphase wird durch das Xetra-System durchgeführt, indem ein Auktionspreis nach dem Meistausführungspreis ermittelt wird. *"Der Auktionspreis ist der Preis, zu dem das höchste ausführbare Ordervolumen und der niedrigste Überhang je im Orderbuch vorhandenen Limit besteht"* (Deutsche Börse 2000, S. 20, für eine ausführliche Darstellung vgl. ebda, S. 31ff.). In der Marktausgleichsphase werden Orders, die aufgrund eines Angebots- oder Nachfrageüberhangs nicht ausgeführt werden konnten, dem Markt zum Auktionspreis für einen begrenzten Zeitraum angeboten.

An die Eröffnungsauktion schließt sich der fortlaufende Handel an. In diesem werden eintreffende Orders (Kaufgebote) und Quotes (Verkaufsggebote) sofort mit der entsprechenden Gegenseite geprüft und möglicherweise ausgeführt. Ansonsten wird das Orderbuch solange gefüllt, bis entweder eine untertägige (Zwischen-) Auktion oder die Schlussauktion durchgeführt wird. Auch diese Auktionen unterteilen sich jeweils in die Aufruf-, Preisermittlungs- und Marktausgleichsphase. Die in der Aufrufphase berücksichtigten Gebote stammen aus dem im fortlaufenden Handel gefüllten Orderbuch, welches während der Auktion selbst geschlossen bleibt. In der Schlussauktion nicht ausgeführte Orders werden auf den nächsten Tag übertragen.

Budimir/Gomber (1999, S. 263ff.) zeigt mit dem Projekt AMTRAS, wie auf dieser Mechanismus in einem agentenbasierten, elektronischen Wertpapierhandelssystem prototypisch umgesetzt werden kann. *"Statt der konventionellen Orderbücher existieren auf einem elektronischen Wertpapiermarkt Agenten, die jeweils die Transaktionsinteressen der Händler repräsentieren und nach anderen Agenten, d.h. Verhandlungspartner, mit kompatiblen Transaktionsinteressen suchen. Jeder Agent repräsentiert eine Order, die der Händler hinsichtlich der relevanten Parameter frei spezifizieren kann. Die Agenten sind beispielsweise in der Lage, einen Verhandlungspartner nach konkreten oder vagen Spezifikationen zu suchen, Auktionen durchzuführen und Verhandlungen des Händlers vollständig elektronisch – entweder nach einer vorgegebenen Strategie oder durch aktives Eingreifen des Händlers – durchzuführen"* (Budimir/Gomber 1999, S. 264).

Der Mechanismus überträgt die Entscheidungsfindung des menschlichen Teilnehmers auf den Agenten. Die Präferenzen und die Strategie, die der Händler sonst zwischen den Auktionen durch Veränderung seiner Orders manifestiert, wird hier im Software-Agenten fest kodiert. *"Als Auktion sieht das Marktmodell von Amtras eine Single auction vor. Dabei trifft die eingestellte Order auf eine oder mehrere Orders auf der Marktgegenseite. Zur Teilnahme an einer solchen Auktion müssen die Händler konkrete Wertpapiere, das gewünschte Volumen, sowie einen Limitpreis spezifizieren. Amtras ermöglicht auch die Abbildung elektronischer Verhandlungen. Für diesen Zweck müssen die Händler neben dem Wertpapier auch die von ihnen gewünschten maximal zu handelnden Volumina angeben. Anschließend können die Händler in einem Preis/Zeit-Diagramm ihre Handelspräferenzen in Form einer Funktion – der sogenannten Verhandlungsfunktion – angeben. Dabei handelt es sich um eine diskrete Funktion, die jedem Verhandlungszeitpunkt genau einen Limitpreis zuordnet. Diese Funktion kann als Verhandlungsstrategie eines Händlers interpretiert werden. Die einzelnen Zeitabstände (Ticks) sind dabei vom System fest vorgegeben. Findet nun ein Produkt- und Partner-matching statt, kommt es zu einer elektronischen Verhandlung. Die Verhandlungsfunktionen der Händler werden nun gegenübergestellt und bei jedem Tick miteinander verglichen"* (Budi-

mir/Gomber 1999, S. 265). "Wählt der Händler im Marktmodell die Möglichkeit der verbindlichen Verhandlungen, so erfolgt nach einer erfolgreichen Verhandlung und Preisfindung mit seinem Kontraktpartner ein sofortiger, automatischer Abschluß des Geschäfts. Bei einer unverbindlichen Verhandlung hingegen folgt nach der erfolgreichen Preisfindung nur eine Bekanntgabe der Verhandlungsergebnisse an die Händler. Um zu einem Abschluß zu gelangen, müssen beide Händler das Verhandlungsergebnis der Agenten bestätigen" (Budimir/Gomber 1999, S. 266).

2.4.4 Konsequenzen für den Einsatz von Auktionsverfahren

Auch die Diskussion der Auktionsverfahren (einschließlich MOP und Agoric Open Systems) orientiert sich an den bereits bei den Arbitrationsverfahren wiederholten Charakteristika von Koordinationsproblem und -mechanismus aus Kapitel 1.

Die Charakteristika der **ubiquitären Kooperation**, der **begrenzten Rationalität** der Entscheidung und der **Verfolgung ökonomischer Prinzipien** tauchen bei Auktionen zusammengefaßt in der Frage des spekulativen Verhaltens der Agenten auf: ist es möglich, dass ein einzelner Agent durch strategisches Verhalten (z.B. im richtigen Moment die Unwahrheit zu sagen) das Ergebnis des Allokationsverfahrens zu seinen Gunsten beeinflussen kann?

Für die Realisierung von Verhandlungsstrategien in Agenten wäre es einfacher, einen Mechanismus wählen zu können, der kein spekulatives Verhalten erfordert und damit auch keine entsprechende Intelligenz (im Sinne der k.-Ebene-Modelle (Vidal/Durfee 1998)) und Anpassungs- bzw. Lernfähigkeit. Die Forschungsrichtung des Economic Mechanism Design (Varian 1995) schlägt in diesem Fall "*incentive compatible*" Protokolle vor, bei denen die beste Strategie nicht darin besteht, zu spekulieren, sondern die eigenen Präferenzen ehrlich zu offenbaren. Ein bekanntes Beispiel eines solchen Mechanismus ist die Vickrey-Auktion, bei der der Gewinner nur den Preis des zweitbesten Gebotes zahlt. Sie wird trotz ihrer geringen Verbreitung in der Realität (Sandholm 1996b, S.300) für die Anwendung in Multi-Agenten-Systemen von vielen Forschern favorisiert (Gomber/Weinhardt 1999, Rosenschein/Zlotkin 1994). Für die Vickrey-Auktion gelten aber bestimmte Annahmen, die in bestehenden Realisierungen häufig verletzt werden (Sandholm 1996b). Die beiden wichtigsten Annahmen sind, dass die Bieter keine Möglichkeit haben, neben der Auktion miteinander zu kommunizieren und dass sie keine Information über die Gebote der anderen Teilnehmer haben. Durch die Kenntnis früherer Gebote des gleichen Nachfragers kann dessen Strategie angenähert und Markträumungspreise antizipiert werden. Durch Nebenkommunikation können die Nachfrager ein Kartell bilden, um ihre Gebote abzusprechen. Dieses Vorgehen wird z.B. von Vulkan (1999, S. 12) für die FCC Spectrum Auktion in den USA zitiert, bei denen die letzten numerischen Stellen der

Gebote zur Signalisierung zwischen den Bietern genutzt wurden. Da diese Annahmen im Internet relativ einfach technisch verletzt werden können und auch deutliche ökonomische Motive dahingehend bestehen, ist die Vickrey-Auktion nur unter bestimmten Umständen einsetzbar. Die iterative englische Auktion wurde unter diesen Umständen von Vulkan/Jennings (1998, S.9) für offene Agentensysteme als passender vorgeschlagen, da sie *"uses open bids and proves that for private value goods the auction is incentive safe if prices are completely exogenous."*

Die Probleme der **schnellen Verarbeitung** und der **Skalierbarkeit** lassen sich zusammenfassend betrachten. Auktionen arbeiten prinzipiell ohne zeitliche Beschränkungen, d.h. mit einer Art künstlichen Zeitrechnung, die durch den Auktionator getaktet wird. Diese Rundenbasiertheit erlaubt die zeitgleiche Beschaffung und Verarbeitung von verteilten Informationen in einem Netzwerk, so dass der Auktionator zu jedem Taktzeitpunkt über globale Information verfügt. Der Auktionator sammelt sequentiell von jedem Agenten Informationen über die Präferenzen, um die Verhaltensregeln für den nächsten Handelstag vorzugeben. Dieser nächste Handelstag beginnt erst nach dem vollständigen Abschluss des Berechnungsvorgangs (Kearney/Merlat 1999, S.149). Eine Änderung der Preise, um im Walras'schen Sinne Angebot und Nachfrage zu steuern, dient gleichzeitig zur Synchronisierung aller Beteiligten. Welche praktischen zeitlichen Probleme dabei in der Realisierung von Internet-Auktionen lauern, haben Kumar/Feldman (1998, S. 14f.) so beschrieben:

"Most of the challenges arise from the fact that while open cry auctions in the real world move rapidly with bidders responding to posted bids in seconds or sub seconds, Internet auctions must move at a slower pace because of Internet latencies."

Diese Latenzzeit liegt vor allem an der Verteilung der (menschlichen) Agenten im Netzwerk und damit der unterschiedlichen Laufzeit von Informationen und Preissignalen, die in realen Auktionsräumen vernachlässigbar ist. In Auktionen auf offenen Netzen muß der Auktionator dafür sorgen, dass alle Informationen, d.h. das Eintreffen von Geboten und die Erteilung des Zuschlages, gleichzeitig an alle Teilnehmer versendet werden. Insofern bestimmt die Latenzzeit des langsamsten Bieters die Geschwindigkeit des gesamten Verfahrens. Die Zentralisierung der Informationssammlung und -weitergabe beeinträchtigt außerdem die Skalierbarkeit der durch einen Auktionator gefundenen Lösung. Da die Berechnungsvorgänge selbst parallel in den Agenten durchgeführt werden, gibt es hier kein Komplexitätsproblem. Die Skalierbarkeit dieser Lösung wird aber durch die Informationsverarbeitungskapazität des Auktionators begrenzt. Die Zeit, die der Informationsbeschaffung und dem Wiederaussenden der Preise dient, steigt mit zunehmender Agentenanzahl und zunehmender Ausbreitung des Netzwerks. Oberhalb einer gewissen Anzahl wird die Zeit zur Informationsbeschaffung schließlich größer

sein als die erlaubte Wartezeit für die einzelnen Agenten, so dass diese wegen Zeitüberschreitung von sich aus den Koordinationsversuch abbrechen.

Ein wichtiger Aspekt, der die operative Durchführung von Auktionen stark beeinflusst, ist die Gewährung von **Sicherheit** und **Privatheit** der Teilnehmerinteressen. Auch hier gilt, dass sämtliche Informationen einer Verhandlungssituation dem Koordinator zugänglich gemacht werden, indem alle beteiligten Parteien (Käufer und Verkäufer) ihre Verhandlungspositionen und Nutzen übermitteln. Dies geschieht in der Annahme, dass der Koordinator diese Informationen keinesfalls Verfahrensbeteiligten weitergibt und dass die Entscheidungen des Koordinators objektiv nachvollziehbar sind. Im Gegensatz zum Arbitrationsverfahren, bei dem die gesamte Nutzenfunktion übermittelt werden musste, genügt bei der Realisierung mit einem zentralen Auktionator die punktweise Publikation des eigenen Nutzens. In offenen Realisierungen wird diese Information an sämtliche Beteiligten weitergeleitet, wobei in geschlossenen Realisierungen lediglich der Auktionator über diese Informationen verfügt.

Um sicherzustellen, dass der Auktionator und der Mechanismus nicht durch einen externen Angreifer, z.B. durch unautorisierte Nachrichten oder Veränderungen existierender Nachrichten sabotiert werden können, werden unterschiedliche technische Sicherheitsmechanismen benötigt. Mit Zeitstempeln muß die zeitliche Gültigkeit und Zugänglichkeit von Geboten und Ankündigungen des Auktionators dokumentiert werden können. Während der Gebotsphase müssen Gebote verschlüsselt werden, damit sie nicht verändert werden können oder (bei geschlossenen Auktionen) in Verletzung der Auktionsregeln anderen Teilnehmern mitgeteilt werden. In offenen Auktionen muß die Anwendung von preistreibenden Phantombietern (engl. *shills*) unterbunden werden, indem durch eine digitale Signatur eine eindeutige Zuordnung eines Gebotes zu einem existierenden Bieter getätigt werden kann. Schließlich ist der Auktionator zur Wahrung der Rechtssicherheit verpflichtet, ein fälschungssicheres Log des gesamten Verkaufsvorganges zu führen, abzuspeichern und mit seiner digitalen Signatur zu unterzeichnen.

Neben der technischen Sicherheit steht das **Vertrauen in die Objektivität und Neutralität des Auktionators** als vertrauenswürdiger dritter Instanz. Der Auktionator ist allerdings wirtschaftlich nicht unabhängig von den Teilnehmern, da die Bezahlung seiner Dienste (wie üblich) in Form von anteiligen Provisionen an den Verkäufen stattfinden. Er ist beispielsweise daran interessiert, möglichst hohe Preise bei den Verkäufen zu erzielen. Im Internet kann der Auktionator aber bspw. leicht unendlich viele Phantombieter generieren und mit einer eigenen Identität versehen.

Schließlich kann das Argument der **mangelnden Dezentralität** gegen Auktionsverfahren angeführt werden. Der Auktionator ist eine vollkommen zentrale Institution. Die Informatio-

nen werden von einer zentralen Stelle gesammelt, welche die Gesamtheit dieser Informationen nutzt, um die Preise festlegen zu können, die das zukünftige Verhalten der Agenten bestimmen (Kearney/Merlat 1999, S.149). Der Auktionator handelt unter der Annahme vollkommener Information, denn nur dann kann ein im ökonomischen Sinn optimales Ergebnis berechnet werden. Im praktischen Einsatz ergibt sich durch die Zentralität das Problem, dass der Auktionator einen einzelnen Angriffspunkt darstellt, um das Verfahren zu beeinflussen. Da er eigene wirtschaftliche Interessen hat, ist er anfällig für Bestechungsversuche, um z.B. in *sealed-bid* Implementationen Informationen über Konkurrenzangebote weiterzugeben oder um direkt die Allokation zu Gunsten eines Agenten zu verändern. Die Insiderproblematik an Wertpapierbörsen oder bei der Vergabe von öffentlichen Bauaufträgen zeigt diese Problematik in der physischen Welt, die in agentenbasierten Auktionen keinesfalls als nicht-existent angenommen werden darf. In technischer Hinsicht und ohne Vorbild in der physischen Welt stellen *Denial-of-Service*-Angriffe (DoS) ein großes Problem für zentrale Koordinationsstellen in offenen, paketbasierten Netzen wie dem Internet dar. In diesem Fall wird ein zentraler Auktionator mit so vielen Informationsanfragen überhäuft, dass eine technische Abarbeitung unmöglich wird. Sowohl der Schwarze Montag an der Wall Street, in jüngster Zeit aber besonders DoS-Angriffe auf zentrale Auktionatoren wie *eBay* (Rötzer 2000) zeigen die Tragweite dieses Problems.

Auktionen können daher in offenen Netzen nur dort eingesetzt werden, wo die Neutralität des Auktionators gewährleistet werden kann, die Agenten wahrheitsgemäße Gebote abgeben und die Kommunikation sicher ist. Diese Eigenschaften erfordern die organisatorische und rechtliche Absicherung aller Beteiligten vor der Teilnahme am Koordinationsverfahren, verhindern aber eine ad-hoc Kooperation mit unbekanntem Teilnehmern. Die Zentralität des Auktionsverfahrens stellt einen prinzipiellen Angriffspunkt sowohl in ökonomischer als auch in technischer Hinsicht dar. Für die Anwendung in Telematiksystemen, in denen erst eine ad-hoc-Kooperation die Potentiale der Technik ausschöpfen kann (Müller et al. 1997, S. 24ff.), sind Auktionsverfahren zu restriktiv.

2.5 Koordination durch marktliche Selbstorganisation

Als dritte Möglichkeit der Koordination ergibt sich schließlich eine Realisierung ohne jegliche zentrale Koordinationsinstanz. Ausgehend von dem Konzept von Moulin (Modus direkten Verhandeln, vgl. auch den Begriff der „*concurrent coordination*“ bei Schiefloe/Syvetsen (1997)) handelt es sich um solche Koordinationsergebnisse, die durch direktes Handeln der Teilnehmer untereinander erzielt werden. Während in Walras' *tâtonnement*-Prozess die Agenten jeweils auf die Beendigung des Berechnungsprozesses durch den zentralen Auktionator

warten mußten, handeln sie in sogenannten *non-tâtonnement*-Prozessen miteinander, bevor eine Gleichgewichtssituation erreicht worden ist.

"Die Vorstellung von einem Tâtonnement ist in zwei Arten von Situationen sinnvoll, nämlich entweder, wenn nur zu Gleichgewichtspreisen gehandelt wird, oder, wenn keines der Güter lagerfähig ist, so dass die Konsumenten in jeder Periode dieselbe Anfangsausstattung haben. Wenn Güter angehäuft werden können, werden sich die Anfangsausstattungen der Konsumenten mit der Zeit verändern und das wird wiederum das Nachfrageverhalten beeinflussen. Modelle, die diese Veränderung der Anfangsausstattung einbeziehen, werden Nicht-Tâtonnement Modelle genannt" (Varian 1994, S. 406).

In der ökonomischen Betrachtung wird aus Vereinfachungsgründen irreversible Konsumption und Güterproduktion typischerweise ausgeschlossen (Cheng/Wellman 1998, S.3).

In Bezug auf das Schichtenmodell der Telematik lassen sich die auf Non-Tâtonnement, dem Modus direkten Verhandeln bzw. „*concurrent coordination*“ basierenden Realisierungen folgendermaßen charakterisieren:

- Eine Kommunikation findet direkt zwischen den einzelnen Teilnehmern statt. Es gibt keinen zentralen Mediator, was auch zur Konsequenz hat, dass die ausgetauschten Informationen immer nur lokales Wissen der Kommunikationspartner widerspiegeln.
- Ausgangspunkt der Koordination ist eine suboptimale Ressourcenallokation. Der dynamische Koordinationsprozeß wird über Transaktionen der Teilnehmer, die für diese jeweils nutzenerhöhend sind, getrieben. Die Präferenzen der Teilnehmer verändern sich im Zeitablauf und steuern den sich dadurch selbst organisierenden Prozeß.
- Die Kooperation zwischen den Teilnehmern ist von individueller Nutzenverfolgung geprägt.

Der sogenannte *Edgeworth-Prozeß* (Varian 1994, S. 406f.) geht z.B. davon aus, dass die Wirtschaftssubjekte bilateral im Sinne des „Modus direkten Verhandeln“ nur dann freiwillig miteinander handeln, wenn ihr Nutzen nach jedem Tauschgeschäft steigt. "Die Nutzensumme muß annahmegemäß über die Zeit zunehmen, so dass einfaches Beschränktheitsargument ausreicht, um einen Konvergenzbeweis zu führen" (Varian 1994, S. 406). Der Endzustand ist dann pareto-optimal und stellt eine optimale Wohlfahrt (als Summe aller Einzelnutzen) dar. Alternativ dazu ist der *Hahn-Prozeß* anzuführen (Arrow/Hahn 1971, Varian 1994, S.406f.). Hier werden Handelsgeschäfte nur abgeschlossen, wenn für ein Gut gleichzeitig bei einem Wirtschaftssubjekt eine Überschussnachfrage und bei einem anderen ein Überschussangebot besteht. Da vorausgesetzt wird, dass eine Überschussnachfrage nach einem Gut dessen Preis erhöht, wird

der indirekte Nutzen der Nachfrager über die Zeit abnehmen, solange Angebot und Nachfrage nicht ausgeglichen sind. Der Zustand des Gleichgewichts ist dann erreicht, wenn jeder einzelne Teilnehmer seinen höchsten Nutzen erzielt hat.

Der Vorteil aller Auktionsverfahren (einschließlich des tâtonnement-Prozesses) ist ihre formale Beschreibung, beweisbare Terminierung und ihre einfache Umsetzung. Vorteile in der Realisierung sind eher bei Non-Tâtonnement-Prozessen zu finden:

"One nice feature of non-tatonnement pure exchange processes is that they are usually more stable than tatonnement processes" (Cheng/Wellman 1998, S.4).

Als Nachteile von non-tâtonnement wird angeführt, dass vorläufige Preise zu strategischem Verhalten führen können, sowie dass irreversible Entscheidungen (also z.B. Produktionsprozesse, die eingekaufte Ressourcen sofort umwandeln und damit dem Mechanismus entziehen) zu ineffizienter Ressourcenallokation führen. Im Vergleich zwischen beiden Marktprozessen schließen Cheng/Wellman (1998, S.4):

"The tentative quality of tâtonnement makes it a more conservative, if somewhat less dynamic, approach."

In der Realisierung ergibt sich die größte Änderung in der zeitlichen Betrachtung des Berechnungsvorgangs. Sowohl bei den Arbitrations- wie den Auktionsverfahren kann solange gerechnet werden (prinzipiell unendlich lange), bis ein Gleichgewichtspreis feststeht. Voraussetzung ist, dass sich die Situation für keinen Teilnehmer ändert, d.h. dass keine irreversible Ressourcenallokation durchgeführt wird. Das prinzipielle Konzept spiegelt eine Berechnung in unendlich kurzer Zeit wider. Für den Agent bietet sich das Bild eines Anfangsgleichgewichtes, gefolgt von einer einzelnen exogenen Änderung und dem sofortigen Umschlagen in ein Endgleichgewicht. Hier ist der Marktmechanismus kein Prozeß, sondern ein Hüpfen von Zustand zu Zustand. In der Realisierung wird dies durch Rundenbasiertheit ("*time slotting*") simuliert. Die gesamte relevante Umwelt inklusive aller Teilnehmer wird während des Berechnungsprozesses des Arbitrators bzw. Auktionators als statisch angenommen.

Diese Annahme ist nicht nur implementativ von wesentlicher Bedeutung, sondern leitet zur grundsätzlichen Notwendigkeit der Realisierung von Lernmethoden in den Agenten hin. Während eine Koordination auf Grundlage zentraler, neoklassischer Konzepte mit regelbasierten oder spieltheoretischen Konzepten auskommt, wird ein stetiger, dynamischer Prozeß die Notwendigkeit adaptiver Anpassung an eine ständige Veränderung der Umwelt nach sich ziehen. Diese Veränderung ist unumkehrbar und ergibt ständig neue Situationen. Erfahrungen aus früheren Situationen müssen durch den Agenten generalisiert und angepasst werden, da die gleiche

Entscheidungssituation nicht zweimal durchlaufen wird. Lernen und ein unumkehrbarer Zeitbegriff bedingen sich:

"Historical time, which is the only definition of time which has any empirical meaning, is a continuum along which structures exhibit a degree of irreversibility. It is precisely because of structural irreversibility that systems have to evolve in the face of new conditions. Relaxation of the irreversibility condition of structure, by definition, removes evolutionary change and also the meaning of time" (Foster/Wild 1994, S. 2).

Dies ist die ökonomische Variante von Heraklits Ausspruch, dass man nicht zweimal in denselben Fluß steigen kann. Die Koordination in selbstorganisatorischen Prozessen verlangt explizit nach adaptiven Kooperationsstrategien, die eine Lernfähigkeit der Agenten voraussetzen. Im folgenden sollen ökonomische Konzepte betrachtet werden, die deutlich machen, welche Bedeutungen hinter den Begriffen des Agenten, seiner Umwelt und der Adaptivität des Agenten an seine Umwelt stehen. Die Darstellung eines solchen Modells, das der Katallaxie bzw. "spontanen Ordnung" von Hayek (1974, 1988), führt zu Realisierungskonzepten selbstorganisatorischer Koordination in Multi-Agenten-Systemen.

2.5.1 Die Katallaxie als Erklärungsmodell

Das Wort Katallaktik wurde 1838 von Erzbischof Whately als Bezeichnung für die theoretische Wissenschaft, die sich mit dem Marktgeschehen befaßt, vorgeschlagen und von Ludwig von Mises und F.A. von Hayek zu Beginn des 20. Jahrhunderts wieder aufgegriffen. Das altgriechische Wort *katallatein* bedeutet nicht nur "austauschen", sondern auch "versöhnen" und "Feindschaft beilegen"¹⁹ (vgl. Hayek 1996, S. 122). **Katallaxie** wird von Hayek als Synonym für den Begriff Marktwirtschaft aufgefasst, ohne dessen Mehrdeutigkeiten zu übernehmen; im engeren Sinne geht es um den wirtschaftlichen Koordinationsmechanismus, der in Märkten auf der Grundlage des Preissystems und der Signalisierung von subjektiver Wertschätzung der Güter durch den Einzelnen wirkt. "Die Menschen haben, ohne das bewußt zu beabsichtigen und zu wissen, Katallaxie als eine Methode der Wissensbildung entwickelt, die zugleich Motivationen erzeugt, nach diesen Informationen zu handeln. Adam Smith nannte dies eine unsichtbare Hand, die das Verfolgen des Eigeninteresses in den Dienst der Anderen stellt" (Hoppmann 1994, S. 141f.). Bei der Katallaxie handelt es sich um eine "spontane Handlungsordnung",

¹⁹ Der politisch-philosophische Anspruch der Katallaxie, der hier nicht weiter verfolgt werden soll, liegt auch in der offenen Austragung von Konflikten in geregelter, kooperativer Form, mit anderen Worten: wer Handel treibt, greift nicht zur Waffe. In den folgenden Ausführungen wird das Wort Agent im ökonomischen Sinne benutzt, kennzeichnet also jemanden, der ökonomisch handelt. Ob dieser Agent ein menschlicher Teilnehmer oder ein Software-Agent ist, wird an dieser Stelle offengelassen.

ein aus koordiniertem Verhalten entstehendes Muster, welches nur abstrakt beschrieben werden kann. Die jeweilige Ausprägung der Katallaxie ist von den Plänen und Handlungen der Teilnehmer bestimmt, die in der Summe unbekannt sind und auch nicht prognostiziert werden können. "Die Menschen bedienen sich daher der spontanen Handelsordnung, ohne zu wissen, wie sie entsteht und ohne ihre besonderen Ergebnisse vorher zu kennen. Wir wissen lediglich, wie sie funktioniert, nachdem wir ihr Funktionsprinzip gedanklich rekonstruiert haben" (Hoppmann 1994, S. 145f.). Vier wesentliche Voraussetzungen, unter denen Katallaxie zustande kommt, können aufgezeigt werden (Hoppmann 1994, S. 139ff.):

1. Agenten arbeiten in ihrem eigenen Selbstinteresse, um ein Einkommen zu erzielen. Dazu benötigen sie eine Formalisierung eigener Ziele, eine Verknüpfung von Einkommen und Nutzen, eine Prognosefähigkeit zukünftiger Umweltzustände und eine Anzahl von Handlungsalternativen, um die Umwelt zur eigenen Nutzensteigerung zu beeinflussen.
2. Agenten wägen in subjektiver Weise präferierte Alternativen gegeneinander ab. Dazu müssen sie Informationen über Alternativen beschaffen, diese bewerten und anhand formalisierter eigener Präferenzen in eine Reihenfolge bringen.
3. Agenten haben Zugang zu Märkten als Ort des Tausches von Ressourcen. Märkte bieten die Möglichkeit, wirtschaftliche Transaktionen zu tätigen und Informationen über Umweltzustände in gebündelter Form zu erfassen. Die Aktionen der Agenten wiederum verändern diese Zustände.
4. Schließlich ist es den Agenten prinzipiell unmöglich, zukünftige Marktzustände exakt zu prognostizieren, was Hayek als "konstitutionale Unwissenheit" (constitutional ignorance) bezeichnet. Daraus folgt eine größtmögliche Adaptivität in den Handlungsmöglichkeiten der Agenten und die Notwendigkeit, anhand der Erfahrung der eigenen Erfolge durch Lernprozesse seine Handlungsstrategien zu ändern.

Generell müssen die wirtschaftlich handelnden Agenten zwei wesentliche Fragen lösen:

1. Wie muß etwas getan werden, d.h. welche Handlungsalternativen stehen zur Auswahl und wodurch werden sie bestimmt?

Diese Frage wird durch Instrumente wie Erfahrungen, Gebräuche und Institutionen beantwortet, die bei Erfolg durch die kulturelle Evolution, Erziehung oder Imitation an zukünftige Generationen weitergegeben werden. Beispiele dafür sind Sprache, Schrift und Geld, die Erfahrung und Wissen einer Vielzahl vergangener Generationen in ihrer Handhabung beinhalten. Sie stellen eine Anpassung des Gehirns an die unbekanntesten Bedingungen seiner natürlichen und sozialen Umwelt dar, welche die Außenwelt der Katallaxie darstellt. Das Lernen der Agenten bezieht sich nicht nur auf das jeweilige eigene interne Modell, sondern

erschafft Rahmenbedingungen für zukünftige Generationen, die nicht erneut von Null anfangen müssen, sondern denen diese Erkenntnisse durch eine "kulturelle Evolution" weitergegeben werden.

2. Was muß getan werden, d.h. welche dieser Alternativen wird mit welcher Begründung ausgewählt?

Um von einem derzeitigen Zustand des Agenten in einen präferierten Zustand zu wechseln, muß eine aktive Handlung durchgeführt werden. Damit diese erfolgreich ist, muß der Handelnde die Pläne, Erwartungen und Aktionen seiner Umwelt und aller anderen handelnden Personen prognostizieren, ohne aufgrund der konstitutionellen Unwissenheit seiner Entscheidung vollständiges Wissen zugrundelegen zu können. Durch die Katallaxie ist es dem Handelnden möglich, Information über das Wissen der Anderen zu erhalten. Der Austausch von Informationen führt zur Generierung von Preisen, die den subjektiven Wert widerspiegeln, den jeder Tauschpartner dieser Information zumißt, ohne dass die Gründe für diese Bewertung expliziert werden. Die einzelnen Handlungsalternativen können durch Orientierung an den Preisrelationen in optimaler Weise verglichen und bewertet werden: "Denn außer der Verteilung von Gütern im Wettbewerbsprozeß des Marktes kennen wir keine Methode, den Einzelnen zu informieren, in welche Richtung seine individuellen Anstrengungen zielen müssen, um möglichst viel zum Gesamtprodukt beizutragen" (Hayek 1996, S.3).

Die einzelnen Marktteilnehmer werden motiviert und geleitet durch ihre Einkommenserwartungen. Ausgehend von den beobachteten Preisrelationen erstellen sie Pläne, bestimmen ihr Marktverhalten und verändern beides anhand der vom Markt gelernten Erfahrungen. Die Preisrelationen verändern sich konstant und diese Veränderung bieten sowohl Information als auch Motivation. Der einzelne Marktteilnehmer betrachtet zwei Wertskalen, zum einen die seiner eigenen Ziele, zum anderen die Marktpreise (Hayek 1996, S. 103ff.). Marktpreise bewerten die verfügbare Handlungsalternative und die Differenz zwischen interner Bewertung und erzieltm oder erzielbarem Marktpreis verändert den Gesamtnutzen des Marktteilnehmers. Inwiefern der Gesamtnutzen mit einem längerfristigen Ziel eines Marktteilnehmers übereinstimmt, ist für den Einzelnen nicht einfach zu beantworten: "Die meisten Menschen müssen, um ihre eigenen Ziele zu erreichen, etwas verfolgen, was sowohl für sie als auch für andere bloße Mittel sind. Das heißt, sie müssen sich an irgendeinem Punkt in eine lange Kette von Aktivitäten eingliedern, die schließlich in ferner Zukunft und an einem fernen Ort zur Befriedigung eines ihnen unbekanntes Bedürfnisses führen werden, nachdem sie durch viele Zwischenstadien mit anderer Zielrichtung hindurchgegangen sind" (Hayek 1996, S. 104). Die Ein-

kommenserwartungen und die Preisrelationen sind die Stabilisatoren des Gesamtsystems, ohne die die Koordination der individuellen Handlungen bald beendet wäre.

Alle Märkte sind bilateral miteinander verbunden. Diese "Interdependenz der Preise", die das gesamte als System arbeiten läßt, arbeitet gegen die konstitutionelle Unwissenheit, indem Wissen, welches weit über die Marktteilnehmer verstreut ist, eingebracht wird. Das Preissystem ist eine Informationsmethode über konkrete Umweltbedingungen, die von Menschen ohne ihre explizite Intention entwickelt wurde:

"Einem schlichten Geist, der sich Ordnung nur als Ergebnis vorsätzlicher Gestaltung vorstellen kann, mag es absurd scheinen, dass unter komplexen Bedingungen eine Ordnung und die Anpassung an Unbekanntes wirksamer durch Dezentralisierung von Entscheidungen erreicht werden können und dass eine Aufteilung der Entscheidungsgewalt die Möglichkeit einer Ordnung des Ganzen regelrecht vergrößert. Doch solche Dezentralisierung bewirkt eben, dass mehr Information berücksichtigt wird" (Hayek 1996, S. 81).

Die Katallaxie arbeitet als selbst-regulierendes System, dessen genaue Umstände dem Menschen unbekannt sind. An dieser Stelle manifestiert sich die konstitutionelle Unwissenheit, die aber gleichzeitig als Motivator für eine andauernde Suche nach Wissen dient. Die evolutionären Prozesse der Katallaxie sind nicht darwinistisch-biologisch, sondern kulturell motiviert. Gleich ist, dass die Katallaxie zwar selbst-organisierende Prozesse entwickelt, die Prognose der aus dem evolutionären Selektionsprozeß herrührenden Entwicklung unmöglich ist: "... ist jegliche Evolution, die kulturelle ebenso wie die biologische, ein Prozeß unablässiger Anpassung an unvorhersehbare Ereignisse, an zufällige Umstände, die nicht hätten vorhergesagt werden können" (Hayek 1996, S. 23). Das Muster, welches durch das koordinierte Verhalten der Einzelnen entsteht, wird deshalb als "spontane Ordnung" von Märkten bezeichnet. Die Details dieser spontanen Ordnung sind determiniert durch die jeweiligen Pläne und Handlungen der Marktteilnehmer, die a priori unbekannt und unmöglich zu prognostizieren sind. Trotzdem ist es möglich, die Eigenschaften der spontanen Ordnung zu nutzen, ohne zu wissen, wie diese entstanden ist und welche exakten Resultate sie hervorbringt. Schließlich folgt aus diesen Ausführungen die nicht-teleologische Sichtweise von Hayek, da die einzelnen Agenten nicht auf ein gemeinsames Ziel des Marktgleichgewichtes hinarbeiten, sondern sich die Koordinationsleistung emergent als Folge der jeweils unterschiedlichen Zielverfolgung des Einzelnen ergibt.

Niemand kann einen optimalen Zustand einer Volkswirtschaft, einen Zustand maximaler Effizienz oder die maximale Wohlfahrt benennen: "Der Markt ist das einzige bekannte Verfahren

zur Beschaffung von Informationen, die es dem einzelnen ermöglicht, die komparativen Vorteile verschiedener Verwendungen wirtschaftlicher Mittel abzuwägen, über die er aus eigener Anschauung etwas weiß und durch deren Einsatz er, ob er will oder nicht, der Bedürfnisbefriedigung weit entfernter unbekannter Personen dient. Dieses verstreute Wissen ist *seiner Natur nach* verstreut und läßt sich keinesfalls sammeln und einer Behörde übermitteln, der die Aufgabe der vorsätzlichen Schaffung von Ordnung übertragen wäre." (Hayek 1996, S. 82). In seinem Buch "Die verhängnisvolle Anmaßung" hat Hayek (1988) diesen Gedanken als grundlegend für das notwendige Scheitern jeglicher planwirtschaftlichen Ökonomie ausgeführt.

Solange die Teilnehmer den allgemeinen Verhaltensregeln folgen, gibt es eine spontane Koordination ihrer Handlungen. Es entsteht eine Ordnung, in der es den Teilnehmern möglich ist, ihre Pläne mehr oder weniger erfolgreich zu realisieren. Dieser Zustand unterscheidet sich von chaotischem Verhalten, in dem kein Ziel mit Hoffnung auf Erfolg planmäßig verfolgt werden könnte. Welche Verhaltensregeln ermöglichen die Katallaxie und wie kann unerwünschten Entwicklungen gegengesteuert werden? "Die Existenzbedingung für solches Eigentum, für Freiheit und Ordnung ist seit den alten Griechen bis heute dieselbe geblieben: Recht im Sinne abstrakter Regeln, welche es dem einzelnen ermöglichen, jederzeit festzustellen, wer berechtigt ist, über eine bestimmte Sache zu verfügen" (Hayek 1996, S. 28). Die zwei wichtigsten Verhaltensregeln stellen die Autonomie in der Handlung sowie die Zulassung von (zu schützendem und zu mehrendem) Privateigentum dar.

Es ist jedoch nicht möglich, eine Zentralstelle quasi außerhalb der Katallaxie zu erschaffen, die über die Einhaltung dieser Regeln *ex cathedra* wacht. Sämtliche Institutionen, die auf die Veränderung der Umwelt einwirken, müssen selbst wiederum Teil der Katallaxie sein: "Denn in Wirklichkeit können wir eine Ordnung von Unbekanntem nur dadurch bewirken, dass wir es *veranlassen, sich selbst zu ordnen*" (Hayek 1996, S. 88). Diese Institutionen schränken den Freiheitsgrad des Systems insgesamt ein, wodurch eine Fokussierung des einzelnen Agenten auf die existierenden Möglichkeiten erfolgt:

"Institutions are frictions which, like frictions in mechanical systems, by restricting movement may make controlled movement possible. [...] It is by preventing the exploration of many possibilities that institutions economics each individual's scarce resource of cognition and focus the attention of that individual on a particular range of options" (Loasby 2000, S. 299).

Dies ist der tiefere Sinn der ökonomischen Selbstorganisation und der führt zur Suche nach Institutionen (Instanzen, Regeln oder Normen), die eine Selbstorganisation in Multi-Agenten-Systemen kontrollieren und steuern können.

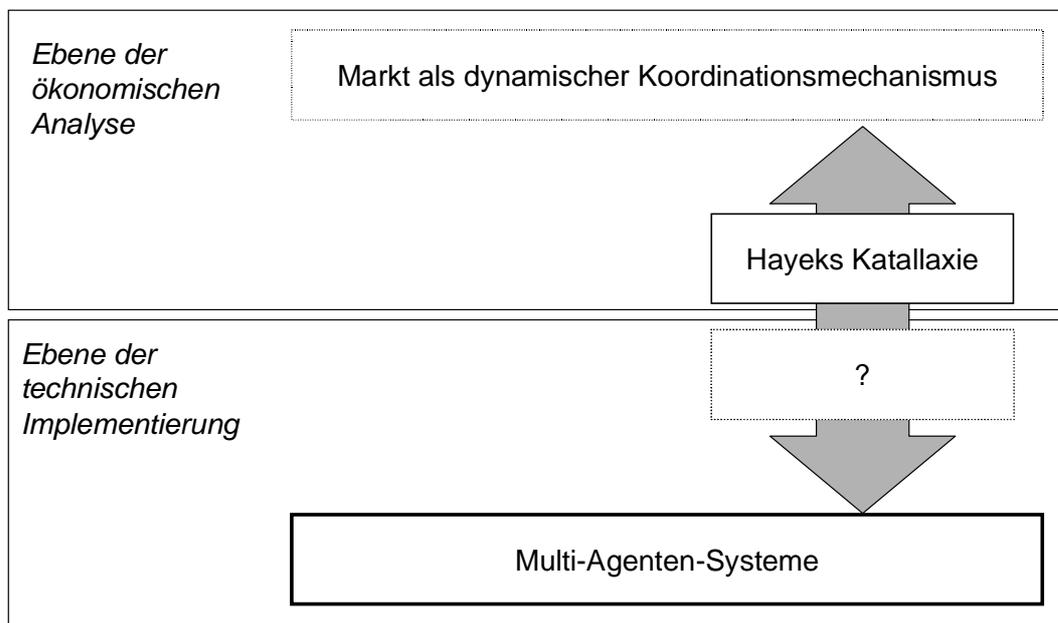


Abb. 7: Katallaxie als Erklärungsmodell des Koordinationsmechanismus "Markt"

Im folgenden sollen zwei Implementierungen von Multi-Agenten-Systemen, die implizit auf ökonomischer Selbstorganisation basieren, beispielhaft vorgestellt werden. Dabei handelt es sich einmal um eine Anwendung in der Fertigungssteuerung, zum anderen um die Allokation von Bandbreite in einem Computernetzwerk.

Agentenbasierte Fertigungssteuerung folgt der Vision, dass *"fully operational Computer Integrated Manufacturing systems can be installed by merely plugging together manufacturing resources which are each controlled by an individual agent"* (Baker 1995, S. 185). Ein wesentliches Element der Fertigungssteuerung ist die Planung der zukünftigen Nutzung der Produktionsressourcen. In der von Baker (1995) beschriebenen Anwendung wird das Kontraktnetzverfahren des Distributed Problem Solving für diesen Zweck modifiziert und als **"Market-Driven Contract Net"** (MDCN) bezeichnet.

Im Market-Driven Contract Net werden reale Gebote, die Preise, Verfügbarkeit sowie Mengen des Produktionsablaufs berücksichtigen, unmittelbar zur Steuerung eingesetzt. Dadurch kann der Produktionsablauf wesentlich näher an die Dynamik der Einkaufs- und Verkaufsmärkte gebunden werden. Virtuelle Unternehmen könnten ihre Fertigungsabläufe direkt mit der Beschaffungs- und Absatzseite vernetzen:

"Production Reservation Scheduling manages the demand side of the factory control problem by a direct interaction between customers and the scheduler. This interaction

assures jobs arrive when they can be produced with the least waiting. This promises to proportionally reduce lead-times and inventories" (Baker 1995, S.187).

Das Konzept von MDCN versucht, Agilität und Produktionseffizienz gleichermaßen zu optimieren. Agilität benötigt Reserven und Freiraum in der Ressourcenallokation, Effizienz wird üblicherweise mit vollständiger und fixierter Allokation gleichgesetzt. Im MDCN steuern Agenten Fertigungsressourcen, z.B. Maschinen, Materialtransportsysteme oder die Lagerverwaltung. Die Nachfrage in dem Netzwerk wird durch Verkaufsagenten repräsentiert. Externe Nachfrage wird durch die Verkaufsagenten als "request for bid"-Nachricht in das Netzwerk als Broadcast eingebracht. Diejenige Ressourcenagenten, die diese Nachfrage befriedigen können, senden eine "will_bid" Nachricht zurück. Wenn sie das Endprodukt nicht direkt selbst herstellen können, werden sie eine "request for bid" Nachricht für die entsprechenden benötigten Bauteile herausgeben, die damit rekursiv bis zu der Stufe erzeugt werden, an der ein Agent ein Gut direkt herstellen kann. Dieser Agent wird mit einem "bid" antworten, aus dem heraus die fragenden Agenten selber Gebote berechnen können, bis ein einzelnes "bid" wieder bei dem Verkaufsagenten ankommt. Gebote werden als Kostenfunktion abhängig von Lieferzeit und Losgröße berechnet und übermittelt; sie kapseln viele Informationen, die Ressourcen definieren, z.B. Komponentenkosten, Lohnkosten menschlicher Maschinenbediener, Zwischenlagerkosten.

Abhängig von den erhaltenen Geboten wählt der Nachfrager dann diejenigen Kosten, Lieferzeiten und Mengen, die er zu erhalten wünscht. Als "purchase order" wird dann korrespondierend zu einem "bid" erneut rekursiv durch den Baum gegangen, bis Produktionsressourcen reserviert sind. Da alle Nachrichten als Broadcast versendet werden, können konkurrierende, aber nicht beschäftigte Agenten ihre Gebote dann wieder löschen.

Baker (1995, S.193f.) beschreibt als Anwendung dieses Mechanismus den General Electric Small Parts Shop, ein Unterbetrieb der Kraftwerkserstellung von General Electric in Schenectady, New York, USA.

"The Small Parts Shop is a classic job-shop which manufactures a number of product lines including nuts, bolts, studs, pins, packing rings, spill strips, packing casings, and oil deflectors. In 1984, the Small Parts Shop produced 350,000 parts to fill over 15,000 orders. The shop's database contained 25,000 unique part designs, and its production equipment included 115 machine tools, of which 24 were numerically controlled." (Baker 1995, S.193)

Ein abgeschlossener Bereich dieser umfangreichen Produktionsumgebung wird als Feldversuch durch MDCN simuliert. Die Bids der Agenten spiegeln dabei echte Produktionskosten wieder, ebenso wie die Nachfrage durch reale Marktpreise gesteuert wird.

Baker (1995) stellt auch Vergleiche zwischen der tatsächlichen Effizienz eines eingesetzten *Flexible Manufacturing Systems* (FMS) und der simulierten MDCN-Umgebung durchgeführt. Dabei ergibt sich, dass vor allem aufgrund der besseren Verteilung der Jobs auf die Ressourcen eine Kostenreduktion von 3,2% der Stückkosten durch MDCN erzielt werden könnte, wenn Maschinen um Jobs konkurrieren können. Die Lagerkosten für maschinennahe Zwischenläger konnten in der Simulation um 45% gesenkt werden. Die durchschnittliche Verweilzeit auf den Maschinen sank von 9,03 Tagen auf 3,72 Tage. Diese Ergebnisse können nicht generalisiert werden, aber sie zeigen doch schon die Größenordnungen, die durch Marktmechanismen im Vergleich zur hierarchischen Planung des FMS erreicht werden können. Als weiterer Vorteil von MDCN wird die ständige Kosteninformation angeführt, die mit den Produkten und den Operationen verbunden ist. Dies erlaubt die Verknüpfung mit einem Kostenrechnungssystem, welches bis auf Maschinenebene durchschnittliche und aktuelle Kosten online überwachen kann.

Die zeitliche Komplexität dieses Verfahrens wurde von Baker (1995, S.208f.) für den Fall, n Aufträge auf m Maschinen zu verteilen, untersucht. Bei Berechnung auf einem zentralen Prozessor ist die Zahl der Maschineninstruktionen, die für MDCN notwendig sind, proportional zu $O(nm^2)$, bei Berechnung auf m parallelen Prozessoren jedoch proportional zu $O(nm)$. Eine verteilte Berechnung macht aber nur dann Sinn, wenn die verfügbare Netzwerkbandbreite die Kommunikation der Gebote zwischen den Prozessoren nicht verzögert.

Es bleibt zu bemerken, dass das Netzwerk der Prozessoren und Maschinen organisatorisch abgeschlossen ist. Alle Agenten haben den gleichen Prinzipal und damit eine gemeinsame Nutzenfunktion. Diese Konfiguration erlaubt es, trotz Eigennutzmaximierung der Ressourcen Nachrichten als Broadcast zu versenden. Sollten die Produktionsressourcen jedoch unterschiedlichen Prinzipalen gehören, so wird entweder ein strategischer Umgang mit Informationen einsetzen (so dass das Broadcast-Prinzip geändert werden muß), oder es muß eine organisatorische Regelung zwischen den Prinzipalen außerhalb des MDCN-Netzwerkes gefunden werden. Diese Erweiterung wird von Baker nicht getroffen, obwohl die Eignung für Virtuelle Unternehmen in der Motivation hervorgehoben wird.

Das Anwendungsbeispiel von Kuwabara et al. (1995, S. 54 ff.) beschreibt die Implementation der Miete von Kommunikationsbandbreite in einem lose vermaschten Rechnernetz. Bei diesem Marktmodell gibt es zwei Arten von Agenten:

"Agents associated with activities (activity agents) request agents at resources

(resource agents) for the resource they require, and resource agents decide whether requests are granted" (Kuwabara 1995, S.54).

Jeder Ressourcenagent ist einer Verbindung zwischen zwei Netzknoten zugeordnet und vermietet auf dieser Verbindung die verfügbare Bandbreite. Die Aktivitätsagenten versuchen, in diesem Netz mehrteilige Verbindungen aufzubauen und müssen dafür Bandbreite kaufen; dabei gibt es mehrere alternative Verbindungen (ebda., S. 58f.). Die Agenten selbst haben keine globalen Informationen und ihre Aktivitäten werden nur von ihrem lokalen Wissen determiniert.

Die Ressourcenagenten versuchen, ihre Einkünfte aus dem Verkaufen der Ressourcen zu maximieren, während die Aktivitätsagenten versuchen, ihre Ausgaben zu minimieren (unter der Annahme, dass die Ausgabenminimierung umgekehrt proportional zu einer Nutzensteigerung ist). Die Agenten eines Typs kommunizieren nicht untereinander; die einzige Kommunikation läuft über *resource requests* (von Aktivitätsagenten zu Ressourcenagenten) und *resource price notifications* (von Ressourcenagenten zu Aktivitätsagenten). Die Strategien der Agenten beziehen sich bei den Verkäufern auf den Preis, bei den Käufern auf die Menge an Ressourcen, die nachgefragt werden.

Im Gegensatz zu zentralistischen Ansätzen implementieren Kuwabara et al. (1995) einen von ihnen so genannten "*equilibratory approach*", bei dem die Ressourcenpreise abhängig von der ihm bekannten Nachfrage direkt von dem Verkäufer bestimmt werden und nicht durch einen zentralen Auktionsmechanismus. Der Mietpreis berechnet sich als Quotient aus der Nachfragemenge und der noch verfügbaren Menge: wenn die Nachfrage hoch ist, steigt der Preis, wenn die Nachfrage niedrig ist, fällt der Preis. Die Simulationen arbeiten mit "slotted time", d.h. der Einteilung der Berechnungszeit in feste Zeitraster, die "slots". Das Senden eines Kommunikationsaktes wird innerhalb eines „slots“ abgewickelt, so dass eine garantierte Zustellung innerhalb des Netzes im gleichen „slot“ gewährleistet wird. Beim Übergang von einem slot zum nächsten existieren keine offenen Kommunikationsbeziehungen. Die Ressourcenagenten kalkulieren ihre neuen Preise auf Basis der Preise der letzten Runde, wohingegen die Aktivitätsagenten ihren Mengenbedarf anhand der kommunizierten Preise in dem aktuellen Zeitraster berechnen. Auch die Anwendung von Kuwabara et al. funktioniert nur in einem geschlossenen Netzwerk, in dem das Gesamtsystem von einer zentralen Stelle synchronisiert werden kann.

2.5.2 Selbstorganisatorische Koordination für offene Märkte

Eine Realisierung der Katallaxie oder vergleichbarer selbstorganisatorischer Marktmechanismen, die dem Paradigma des *Modus der direkten Verhandlung* folgen, muß sich ebenfalls den in Kapitel 1 vorgestellten Kriterien stellen.

Die **Berechnung** eines Allokationszustandes wird bei "direct agreement" Realisierungen in Echtzeit durchgeführt, da die Zeitphasen eines Verhandlungsprozesses durch die Agenten selbst gesetzt werden. Das Aussenden und Empfangen von Kommunikationsakten taktet die Verhandlungszeit für die Agenten, die in der Zwischenzeit Berechnungen für ihre Folgeschritte durchführen können. Alle Zwischenergebnisse, die sich aus einem Übergang von Ressourcen (bzw. Verfügungsrechten) von einem Agenten auf einen anderen Agenten ergeben, sind existent. Sollte die Koordination an dieser Stelle abbrechen (z.B. aufgrund technischer Fehler oder durch Erreichen einer Zeitschranke), so wird keine zusätzliche Phase benötigt, um den Mechanismus abzuschließen.

Gerade die **Privatheit** der Information wird durch Realisierungen des "Modus direkter Verhandlung" wesentlich stärker unterstützt als bei den beiden anderen Kategorien. In bilateralen Verhandlungen werden nur dem jeweiligen Verhandlungspartner punktweise Ausschnitte der Nutzenfunktion bekanntgemacht. Dadurch wird das Wissen über die Strategie des Agenten notwendigerweise über die (konkurrierenden) Verhandlungspartner verteilt. Diese haben ihrerseits im Allgemeinen keinen Anreiz, diese Information weiterzugeben, weil sie Wettbewerbsvorteile darstellen. Zusätzlich verlangt diese Unvollständigkeit der Information nach Maßnahmen der Adaption an die Umwelt (z.B. durch Lernprozesse), wodurch sich die Nutzenfunktion der Agenten selbst im Zeitablauf verändert. Jeder Agent muß davon ausgehen, dass die Informationen, die andere Agenten liefern können, bereits historisch sind und das aktuelle Verhalten des Gegenübers nur noch annähernd beschreiben können, was deren ökonomischen Wert weiter vermindert. Durch sparsame Datenweitergabe, einen schnellen Alterungsprozeß der Informationen und die verteilte Speicherung wird implizit eine im Vergleich der Koordinationsmechanismen maximale Geheimhaltung ermöglicht.

In Bezug auf die **Skalierbarkeit** läßt sich feststellen, dass die Realisierung des "Modus direkter Verhandlung" Informationen in paralleler Weise verarbeitet, so dass hier wie bereits bei den Auktionsverfahren kein Komplexitätsproblem entsteht. Durch die jeweils bilaterale Weitergabe von Informationen wird das Problem der zentralen Informationsverarbeitung umgangen. Erkauft werden diese Vorteile durch den ungleich höheren Kommunikationsaufwand zur Informationsbeschaffung und die damit verbundene konstitutionelle Unsicherheit über den augenblicklichen Gesamtzustand des Systems. Das Design von Agenten, die in einem derarti-

gen Koordinationsumfeld arbeiten, erfordert eine im Vergleich zu zentralen Ansätzen höhere "Intelligenz" und Komplexität in der Kommunikations- und Kooperationsschicht.

In der Einordnung des Schichtenmodells der Telematik bedingt die Koordination durch Selbstorganisation die Kooperation mittels adaptiver Strategien. Eine Koordination in immer ähnlich wiederkehrenden Situationen lässt sich durch Agenten implementieren, die diese endliche Menge an Entscheidungssituationen durch Regeln oder spieltheoretische Analyse auflösen. Wenn sämtliche alternativen Ereignisse, die aus einer bestimmten Handlung resultieren können, mit ihren Eintrittswahrscheinlichkeiten bekannt sind, ist es möglich, die beste Handlung mathematisch zu bestimmen. Tatsächlich besteht eine konstitutionelle Unsicherheit, da es keine vollständige Auflistung der möglichen Ereignisse gibt. Als Ergebnis dieser Unsicherheit werden die Wirtschaftssubjekte auch fehlerhafte, aber kreative Entscheidungen begehen und damit die Unsicherheit weiter verstärken. Der Marktprozeß führt dazu, dass trotz dieser Unsicherheit Koordination stattfindet, die Allokationsleistung verbessert und die Auswirkung fehlerhafter Entscheidungen vermindert wird. Die Rückmeldungen aus dem Marktprozeß lenken das Verhalten und die Entscheidungen des Einzelnen. Dieser Vorgang kann als "Lernen" bezeichnet werden. Lernen bezeichnet nicht allein das Anpassen von Variablen in einer sonst unveränderten Lernfunktion, vielmehr kann aus dem Lernvorgang neues Wissen entstehen, das die Lernfunktion selbst verändert (O'Driscoll/Rizzo 1985, S.9f.).

In realistischen Szenarien können sämtliche hier beschriebenen Koordinationsmechanismen nur Annäherungen an die Wirklichkeit darstellen. In Betrachtung der o.a. Anforderungen läßt sich aber festhalten, dass die Implementierung einer "direct agreement"-Koordination die größten Vorteile in Bezug auf Geschwindigkeit, Flexibilität und Freiheit der Eigennutzverfolgung für die Teilnehmer bietet. Aufgrund vor allem historischer Entwicklungen sind als Koordinationsmechanismen für Multi-Agenten-Systemen bisher fast ausschließlich Arbitrations- bzw. Auktionsmechanismen entwickelt worden. Das im folgenden Kapitel beschriebene Multi-Agenten-System AVALANCHE nimmt hingegen das Konzept der spontanen Ordnung von Hayek als Vorlage für eine Koordinationsrealisierung. Die Ergebnisse dieser Realisierung in Bezug auf Agentenverhalten und Koordinationsleistung werden daran anschließend beschrieben.

3 Koordination durch autonome Software-Agenten in Mehragenten-Systemen

Der Begriff des Software-Agenten wird in der Praxis und in der Forschung sehr vielfältig verwendet. Die Schwierigkeit, eine eindeutige Definition zu finden, wird in der Heterogenität der vorgeschlagenen Taxonomien deutlich (Bradshaw 1997), die zwischen den Dimensionen Mobilität, Intelligenz und Autonomie (Gilbert 1995), Kooperativität, Lernfähigkeit und Autonomie (Nwana 1996), Hardware-/Softwarebasierung, Einzelagent-/Mehragentensystem und Anwendung (Franklin/Graesser 1996) unterscheiden. Die Schwierigkeiten liegen vor allem in der Problematik, die Begriffe „Autonomie“ und „Intelligenz“ sauber zu definieren (Petrie 1996), die in allen Taxonomien als grundlegend angesehen werden.

Einig ist sich die Forschung nur im Begriff der Autonomie, der einen Agenten von einem normalen Programm unterscheidet. Autonomie wird als „zielgerichtetes, proaktives und selbststartendes Verhalten“ verstanden (Bradshaw 1997, S.8). Damit werden als Software-Agenten solche Prozesse bezeichnet, *„which function continuously and autonomously in an environment in which other processes take place and other agents exist“* (Shoham 1997, S. 272). Maes führt folgende Definition an:

„An agent is a system that tries to fulfill a set of goals in a complex, dynamic environment. An agent is situated in the environment: it can sense the environment through its sensors and act upon the environment using its actuators.“ (Maes 1994b, S. 2)

Wooldridge (1999, S.32) definiert Agenten recht einfach und pragmatisch:

“To summarize, agents are simply computer systems that are capable of autonomous action in some environment in order to meet their design objectives.”

Mit den Objekten der objektorientierten Programmierung haben Agenten das Prinzip der Kapselung gemeinsam, d.h. des Verbergens der konkreten Implementation gegenüber anderen Objekten und der ausschließlichen Kommunikation über explizit freigegebene Schnittstellen. Die Deklaration einer Objektvariablen oder einer –methode als *private* lässt keine Veränderung durch ein anderes Objekt zu. Da Objekte miteinander kommunizieren und interagieren sollen, müssen bestimmte Methoden anderen Objekten zugänglich gemacht werden (z.B. als *public* deklariert). In diesem Fall gibt das Objekt die Kontrolle über sein Verhalten auf. Ein anderes Objekt kann die *public* deklarierte Methode aufrufen (*invoke*) und das aufgerufene Objekt zwingen, diese Methode von Anfang bis Ende durchzuführen.

Im Gegensatz dazu zeigen Agenten ein anderes Verhalten, wenn von ihnen ein Dienst angefordert wird:

“If an agent requires a service that is managed by another agent it cannot simply instruct it to start the service” (Jennings et al. 1999, S.5).

Agenten senden einander Anforderungen (*request*), so dass der Agent, von dem ein Dienst angefordert wurde, diese Anforderung auf Kompatibilität mit seinem eigenen Zielsystem überprüfen und über Ausführung oder Nichtausführung entscheiden kann. Die Entscheidung darüber, ob eine bestimmte Methode aufgerufen wird, liegt im objektorientierten Fall bei dem aufrufenden, im agentenbasierten Fall beim aufgerufenen Objekt (Wooldridge 1999, S. 35). Bauer et al. (2000, S. 2) charakterisieren den Unterschied zwischen Agenten und Objekten folgendermaßen:

„an agent is an object that can say "go" (dynamic autonomy as the ability to initiate action without external invocation) and "no" (deterministic autonomy as the ability to refuse or modify an external request).“

Damit eignen sich von einer organisatorischen Sicht aus Agentensysteme für solche Domänen, in denen keine hierarchische Beziehung zwischen den Softwareobjekten besteht. Stattdessen werden Agenten miteinander kooperieren, wenn es den eigenen Nutzen verbessert, was im Falle wirtschaftlicher Domänen zu der plakativen Unterscheidung führt:

“objects do it for free, agents do it for money.” (Wooldridge 1999, S.35).

Zusammenfassend kann ein Softwareagent als eine gekapselte Problemlösungseinheit betrachtet werden, die folgende Eigenschaften besitzt (Wooldridge 1999, S. 32ff.):

1. **Autonomie:** Agenten führen die Mehrzahl ihrer Aufgaben zur Problemlösung ohne den direkten Eingriff von Menschen oder anderer Agenten aus und sie haben steuernden Zugriff über ihre eigenen Aktionen und ihren eigenen internen Zustand. Die Autonomie von Software-Agenten kann nur sinnvoll erreicht werden, wenn den Agenten ein gewisser Grad an Entscheidungsfreiheit zugestanden wird. Abhängig von eigenen Aktionen der Vergangenheit, Einflüssen der Umwelt und eigener Planung wählt der Agent seine nächste Aktion aus. Steels (1996) unterscheidet zwischen der *operational* und der *behavioral autonomy*. Dabei ist die Ausführung von Aufgaben ohne (menschliche) Fremdeinwirkung der *operational autonomy* zuzurechnen, während das Treffen eigener Entscheidungen, die eigene Zielbildung sowie die Fähigkeit zu Lernen eine *behavioral autonomy* implizieren. *Behavioral Autonomy* setzt *Operational Autonomy* voraus, umgekehrt ist dies jedoch nicht der Fall. Diese Unterscheidung spielt auch bei der späteren Abgrenzung von deliberativen zu reaktiven Agenten eine Rolle.

2. **Sozialverhalten:** Agenten interagieren, wenn es ihnen angemessen erscheint, mit anderen Agenten, um ihre eigene Problemlösung abzuschließen oder um andere Agenten in deren Problemlösungsaktivitäten zu unterstützen. Hierbei geht es aber nicht nur um den reinen Informationsaustausch, so wie Millionen von Computern im Internet jeden Tag untereinander kommunizieren, sondern tatsächlich um die Fähigkeit zur Kooperation und zur Verhandlung, um gemeinsame Ziele zu erreichen. Diese Eigenschaft ist eine Grundvoraussetzung zur Gestaltung von Mehragentensystemen.
3. **Proaktivität:** Agenten übernehmen die Initiative und verfolgen neue Möglichkeiten, wenn sich dies ergeben sollte. Zielgerichtetes Handeln alleine kann durch einfache Programmierung erreicht werden, indem der Designer von einem Anfangszustand ausgeht (pre-condition) und sowohl Ziel (post-condition) als auch Verfahren vorgibt. Unter der Bedingung einer statischen Umgebung und eines korrekten Verfahrens kann angenommen werden, dass bei Terminierung der Zielzustand eingetreten ist. Trifft diese Annahme nicht zu, muß der Agent reagieren und sein Verfahren der Zielverfolgung den veränderten Bedingungen anpassen.
4. **Reaktivität:** Agenten überwachen ihre Umgebung und reagieren zeitnah auf Veränderungen, indem sie ihr Verfahren zur Zielerreichung verändern. Reaktivität und Proaktivität stehen in einer Wechselbeziehung. Hierbei muß ein Gleichgewicht gefunden werden, damit der Agent lange genug auf einem Ziel fokussiert bleibt und nicht ständig zwischen alternativen Verfahren wechselt (Wooldridge 1999, S.33).

Weitere Eigenschaften können sein: Mobilität (räumliche Autonomie im Sinne der freien Wahl des Ausführungsortes), zeitliche Kontinuität (Agenten sind kontinuierlich laufende Prozesse, keine einmaligen Berechnungen, die nach Beendigung der Aufgabe terminieren), Adaptivität (Agenten passen sich kontinuierlich den Veränderungen der Umgebung an) und Personalisierbarkeit (jeder Benutzer wird die gleichen Aufgaben anders durchführen; um dem Benutzer helfen zu können, muß sich der Agent individuell anpassen können, indem er vom Benutzer lernt). Diese letzteren Eigenschaften finden sich nicht in allen Agentensystemen, sondern sind je nach Einsatzgebiet und verwendeter Technologie unterschiedlich stark ausgeprägt.

3.1 Komponenten für Software-Agenten

Um die o.a. Eigenschaften umzusetzen, besteht ein Agent im Allgemeinen aus folgenden Grundbestandteilen (Crutchfield 1994, S.15, Holland 1992, S. 196f., Wooldridge 1999, S. 32)

1. Ein internes Modell als Repräsentation seiner Umwelt, durch das der Agent versucht, die Zukunft zu antizipieren und seine gegenwärtigen Aktionen aufgrund seiner Erwartungen zu

planen. Das interne Modell sollte fähig sein, die Konsequenzen für Handlungen zu bestimmen, ohne sie wirklich durchzuführen. Außerdem sollte sich das Modell anpassen und verbessern können, der Agent sollte also aus Fehlern lernen. Die Eigenschaften „Autonomie“ und „Sozialverhalten“ werden durch den Aufbau des internen Modells determiniert.

2. Sensoren (sensor input) erlauben es dem Agenten, seine Umwelt wahrzunehmen. Diese Informationen werden im internen Modell verarbeitet. Sensoren sind ein wesentlicher Bestandteil der Eigenschaft „Reaktivität“.
3. Durch Effektoren (action output) führt der Agent seine Aktionen aus und verändert damit die Umwelt. Effektoren sind wesentlich für die Implementation der Eigenschaft „Proaktivität“.

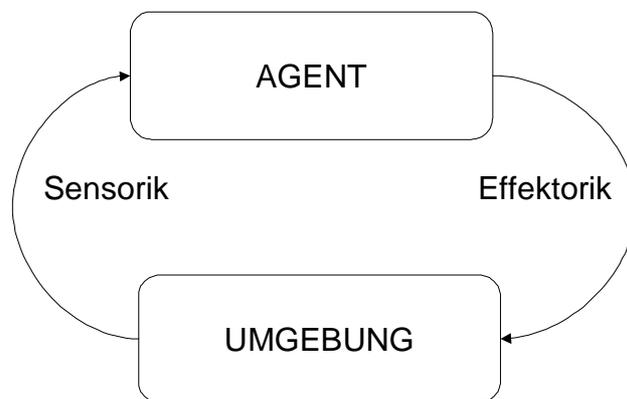


Abb. 8: Rückkopplung zwischen Agent und Umgebung (nach Wooldridge 1999, S. 29)

3.1.1 Einfluß der Umgebung auf die Gestaltung der Agenten

Die Ausprägung der Umgebung ist von entscheidender Bedeutung für die Gestaltung des Agenten, z.B. für die erforderliche Komplexität des internen Modells. Russel/Norvig (1995, S. 46), auch Wooldridge (1999, S. 30) unterscheiden zwischen diesen jeweiligen Dimensionen:

1. **Zugänglich oder nicht-zugänglich:** Die Domäne wird als zugänglich bezeichnet, wenn der Agent fähig ist, vollständige, genaue und zeitnahe Information über den gesamten Umgebungszustand zu erhalten. Zugängliche Umgebungen erlauben eine einfache Agentenimplementation, realistische Szenarien (z.B. das Internet) sind meist nicht-zugänglich.
2. **Deterministisch oder nicht-deterministisch:** In einer deterministischen Umgebung resultiert jede Handlung in einem isolierbaren Effekt. Aus dem aktuellen Zustand der Umgebung und des Agenten lässt sich ein zukünftiger Zustand nach der nächsten Aktion des Agenten

berechnen. Die physische Welt ist dagegen ein Beispiel für eine nicht-deterministische Umgebung. Russel/Norvig (1995, S.46) stellen fest, dass alle Umgebungen, die für praktische Anwendungen und Zwecke vielversprechend erscheinen, aufgrund ihrer Komplexität als nicht-deterministisch betrachtet werden müssen.

3. **Episodisch oder nicht-episodisch:** In einer episodischen Umgebung hängt der Erfolg eines Agenten nur von seinem Verhalten in der konkret vorliegenden Situation (der Episode) ab. In einer nicht-episodischen Umgebung hat das gegenwärtige Verhalten des Agenten Auswirkungen bspw. auf seinen Erfolg in zukünftigen Situationen bzw. Episoden.
4. **Statisch oder Dynamisch:** Eine statische Umgebung kann als ausschließlich durch die Handlungen des betrachteten Agenten veränderbar angesehen werden. In einer dynamischen Umgebung arbeiten gleichzeitig weitere Prozesse oder Agenten. Die Umgebung verändert sich durch deren Handlungen, ohne dass der einzelne Agent dies steuern könnte. Die physische Welt ist eine in hohem Maße dynamische Umgebung.
5. **Diskret oder kontinuierlich:** Eine Umgebung ist diskret, falls es in ihr eine festgelegte, endliche Zahl an Aktionen und wahrnehmbaren Zuständen gibt, wie z.B. bei einem Schachspiel. Die physische Welt ist hingegen kontinuierlich.

Aus dieser Aufstellung wird deutlich, dass eine Umgebung, die zugänglich, deterministisch, nicht-episodisch, statisch und diskret ist, einen sehr hohen Abstraktionsgrad von der wirklichen Welt aufweist. In so einer Umgebung ist die Implementation von Agenten zwar einfach durchführbar, die Klasse der Probleme, die derart angegangen werden kann, ist jedoch begrenzt. Je realitätsnäher die Umgebung ist, in der Agenten eingesetzt werden sollen, desto mehr der vereinfachenden Eigenschaften müssen in der Technikgestaltung fallengelassen werden und desto komplexer wird auch das interne Modell des Agenten. Die reale Welt als komplexeste Umgebung ist als nicht-zugänglich, nicht-deterministisch, nicht-episodisch, dynamisch und kontinuierlich charakterisierbar. Forschungsprojekte beginnen notwendigerweise in Laborumgebungen, die von diesem Eigenschaftenbündel weit abstrahieren und sich erst mit der Zeit den realweltlichen Ausprägungen öffnen²⁰. Die Aussagefähigkeit von Laborversuchen für reale Anwendungsdomänen ist soweit begrenzt. Um Agenten und Mehragentensysteme für praktische Anwendungen einsetzen zu können, muß ein Schritt gemacht werden von den heutigen simulativen Laborsystemen hin zu offenen Implementationen (siehe Abb. 9).

20 Ein Beispiel für eine Agentenimplementation mittlerer Komplexität in einer zugänglichen, deterministischen, nicht-episodischen, dynamischen und kontinuierlichen Umgebung stellt der Roboterfußball „Robocup“ dar. Die Zugrundelegung fester Spielregeln und eines abgegrenzten Spielfeldes schränkt die Komplexität bereits dermaßen ein, dass Robocup als ernstzunehmendes Forschungsproblem für Informatikdisziplinen wie verteilte Planung, Bilderkennung und Robotik einzustufen ist (vgl. Gutmann et al. 2000).

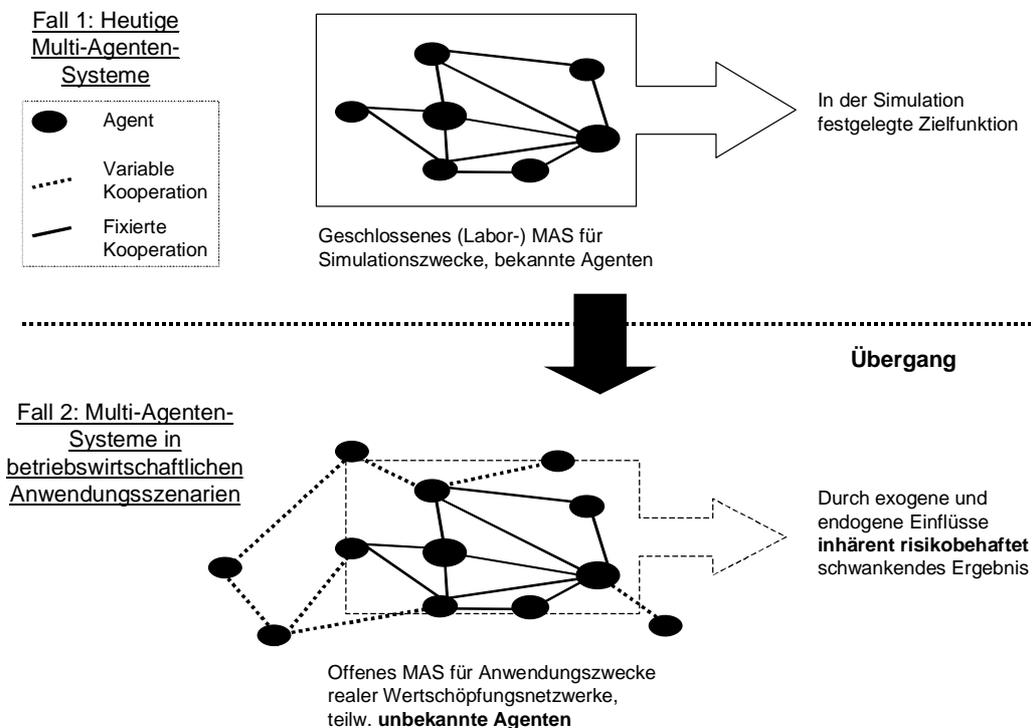


Abb. 9: Übergang von geschlossenen zu offenen Mehragentensystemen

Im hier betrachteten, realistischen Fall eines elektronischen Marktplatzes kann die Umgebung, die vor allem aus den anderen Software-Agenten besteht, als nicht-zugänglich (keine vollständige Marktinformation möglich), nicht-deterministisch (Nutzen aus einer eigenen Aktion kann nicht berechnet werden, sondern hängt von den Aktionen aller anderen Agenten ab), nicht-episodisch (Agenten verändern ihre Strategie über die Zeit), dynamisch und kontinuierlich (Zustände anderer Agenten sind unbekannt) charakterisiert werden. Da die Anwendung nicht unveränderlich und als Ganzes vordefiniert (d.h. geschlossen) ist, sondern grundsätzlich rekonfigurierbar und offen für neue Teilnehmer bzw. Agenten, muß das interne Modell relativ komplex gestaltet werden. Welche Gestaltungsoptionen des internen Modells es gibt und welche in einem elektronischen Markt eingesetzt werden sollten, wird im folgenden Kapitel beschrieben.

3.1.2 Modellierung der Agenten

Der wichtigste Bestandteil eines Agenten ist das interne Modell, da es für die gesamte Informationsverarbeitung, die Ziehung von Schlußfolgerungen, Entscheidungsfindung und Lernvorgänge verantwortlich ist. In der Konzeption des internen Modells lassen sich zwei Ansätze unterscheiden, die der deliberativen Agenten und die der reaktiven Agenten.

Ein **deliberativer Agent** (Genesereth/Nilsson 1987, pp. 325ff.) zeichnet sich durch drei Eigenschaften aus (Wooldridge 1992, S. 36): Er beinhaltet eine explizit repräsentierte Datenbasis mit Formeln, die das Wissen des Agenten bzw. seine Prognose der Umwelt beinhalten, er operiert in einem kontinuierlichen Zyklus von beobachten, entscheiden und handeln und seine Schlußfolgerungen und Entscheidung basieren auf einer Form logischer Inferenz. Die Umwelt wird in einer Form symbolischer Darstellung übersetzt, woraus gleichzeitig die zwei wesentlichen Probleme deliberativer Agenten resultieren. Zum einen muß der Agent, um sinnvoll entscheiden zu können, seine Wahrnehmung der realen Welt in einer möglichst kurzen Zeit in eine korrekte, adequate, symbolische Beschreibung übersetzen. Zum anderen muß er aus dieser Darstellung durch logische Wissensverarbeitung in erneut möglichst kurzer Zeit zu sinnvollen Entscheidungen gelangen.

Das k.-Ebene („k-level“) Konzept von Vidal und Durfee (Vidal/Durfee 1996) versucht eine Einordnung von unterschiedlichen Agententypen anhand des in ihnen implementierten internen Modells.

Agenten ohne explizites Modell können als Agenten der 0. Ebene bezeichnet werden. Ihre Aktionen werden bestimmt von ihrem Eingaben und den Belohnungen, die sie für ihre eigenen Aktionen bekommen. Für Agenten der 0. Ebene gilt:

“0-level agents must learn everything they know from observations they make about the environment, and from any rewards they get.” (Vidal/Durfee 1996, S. 6)

Diese Agenten erhalten eine Eingabe (*stimulus*), führen eine Aktion (*response*) aus und erhalten eine Belohnung. In dem Szenario eines agentenbasierten elektronischen Marktes erfassen die Agenten eine Marktsituation durch die anonyme Aggregation verschiedener Angebote und Nachfragen. Eine Entscheidung zur Auswahl eines Transaktionspartners und Durchführung einer Markttransaktion treffen sie allein aufgrund des relativ zu anderen Angeboten günstigeren Preises. Als Ergebnis der Transaktion lässt sich der monetäre Nutzenzuwachs als Belohnung des Agenten errechnen.

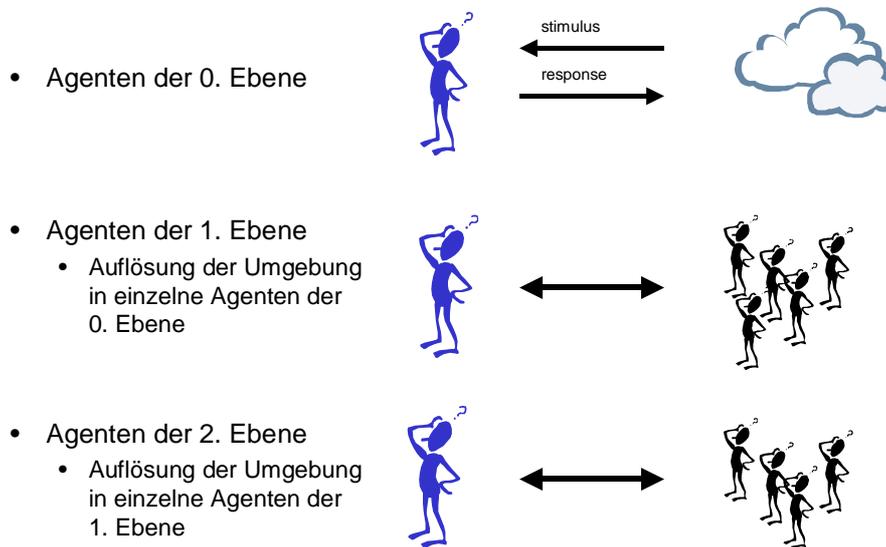


Abb. 10: Das k.-Ebene-Konzept von Vidal/Durfee (1996)

Agenten der 1. Ebene beziehen die Existenz anderer Agenten in ihre Modelle ein, haben jedoch keine Möglichkeit, deren Entscheidungswege nachzuvollziehen oder ihnen Absichten zuzuschreiben. Ihre Aktionen basieren auf den vorhergehenden Aktionen anderer Agenten:

“1-level agents can only model others by looking at their past behavior and trying to predict, from it, their future actions”. (Vidal/Durfee 1996, S. 7).

Die Einführung eines Gedächtnisses, so dass jeder Agent Informationen über ein früheres Verhalten anderer Agenten hat (z.B. als Reputationsliste), erfüllt diese Bedingung. Agenten der 1. Ebene modellieren ihre Umgebung als Agenten der 0. Ebene und nehmen deren Verhalten als vorgegeben an. Agenten der 1. Ebene sind beispielsweise notwendig für nicht-episodische Domänen, in denen aktuelles Verhalten Einfluß auf zukünftige Entscheidungen hat.

Agenten mit einem expliziten Modell über andere Agenten können als Agenten der 2. Ebene beschrieben werden. Agenten der 2. Ebene versuchen, auf Basis eines 1.-Ebene-Modells der anderen Agenten deren Aktionen zu prognostizieren. Der eher auf dieser Ebene anzusiedelnde Ansatz der sogenannten „symbolischen KI“ unterstellt, dass intelligentes Verhalten dadurch generiert werden kann, indem das System eine symbolische Repräsentation der Umgebung und seines erwünschten Verhaltens erzeugt und diese Repräsentation in syntaktischer Weise

manipuliert (Wooldridge 1999, S. 42). In logikbasierten Agenten-Architekturen ist die symbolische Repräsentation in logischen Formeln kodiert und die syntaktische Manipulation besteht aus logischer Deduktion und dem Beweisen von Theoremen. Agenten werden dabei durch mentale Eigenschaften spezifiziert, die weder eine bestimmte Implementation noch eine kognitive Architektur implizieren. Diese können z.B. sein (Rao/Georgeff 1995, FIPA 1997, S.7):

1. **Belief** (Glauben) – die Menge an Propositionen (Aussagen, die entweder wahr oder falsch sein können), von denen der Agent glaubt, dass sie zum aktuellen Zeitpunkt wahr sind. Propositionen, von denen geglaubt wird, dass sie falsch sind, werden dabei durch Glauben an die Negation dieser Proposition repräsentiert.
2. **Uncertainty** (Unsicherheit) – die Menge an Propositionen, die der Agent kennt, von denen er aber nicht mit Sicherheit sagen kann, ob sie derzeit wahr oder falsch sind. Die explizite Repräsentation von Unsicherheit ermöglicht es den Agenten, auch ungenaue Informationen zu kommunizieren und in Entscheidungen einzubeziehen.
3. **Desire** (Wunsch) - eine Menge langfristiger Ziel, als Beschreibung eines erwünschten Umweltzustandes. Wünsche stellen eine Motivation zum Handeln dar und können widersprüchlich zueinander sein.
4. **Intention** – eine Wahlmöglichkeit, Eigenschaft oder eine Menge von Propositionen der Umwelt, von denen der Agent glaubt, dass sie derzeit falsch sind, aber die er wahr haben möchte. Ein Agent, der eine Intention verfolgt, wird einen Handlungsplan ausarbeiten, um den Zustand der Umwelt in diesem Sinne zu verändern.

Der größte Nachteil dieser sogenannten **BDI-Architekturen** (für „Belief, Desire, Intention“, vgl. Rao/Georgeff 1995) für ihre praktische Anwendung besteht in der Zeitbegrenzung, die in realen Anwendungen verlangt wird:

„Decision making in such agents is predicated on the assumption of calculative rationality – the assumption that the world will not change in any significant way while the agent is deciding what to do, and that an action which is rational when decision making begins will be rational when it concludes“ (Wooldridge 1999, S. 47).

Ferber setzt dem Begriff der deliberativen (symbolischen) Agenten die **reaktiven Agenten** gegenüber.

“Reactive agents are simple situated agents which have no representation of their environment and act using a stimulus/response type of behavior; they respond to the present state of the environment in which they are embedded” (Ferber 1994, S. 9).

Reaktive Agenten entsprechen damit den Agenten der 0. Ebene in der Kategorisierung von Vidal/Durfee (1996). Als Charakteristik reaktiver Agenten kann das Fehlen eines Speichers/Gedächtnis und das Fehlen einer Planungskomponente (*reasoning*) angesehen werden. Die Implementation des internen Modells der reaktiven Agenten ist deutlich schwächer als bei der symbolischen KI ausgeprägt; Intelligenz wird von den Verfechtern der *Behavior-Based AI* (Maes 1993) bzw. *Bottom-Up AI* (Brooks 1986) als biologische Eigenschaft (*Enaction*) (Varela et al. 1991) ausgelegt, d.h. als die Summe der Fähigkeiten eines Agenten, mit seiner Umgebung interagieren zu können (Maturana/Varela 1987).

Als Anwendungsdomäne der reaktiven Agenten hat sich insbesondere die Simulation biologischen Verhaltens und die Robotik ergeben (dort z.B. die Subsumptionsarchitektur von Brooks (1996)). Im Verständnis der reaktiven Agenten und zur Abgrenzung gegenüber den deliberativen bzw. kognitiven Agenten ist besonders die Veränderung des Fokus von der Gestaltung einzelner Agenten zur Organisation von Multi-Agenten-Systemen bemerkenswert:

“Because of their complexity, cognitive agents are often considered as self-sufficient: they can work alone or with only a few other agents. On the contrary, reactive agents need companionship. They cannot work isolated and they usually achieve their tasks in groups” (Ferber 1994, S.10).

Als Vorteil dieser Gestaltung kann insbesondere die redundante Existenz einzelner Agenten angesehen werden – eine Agentengruppe soll auch dann ihr Ziel erreichen können, wenn einzelne Agenten ausfallen, was die Robustheit und Fehlertoleranz des Gesamtsystems erhöht. Reaktive Agenten erhalten eine Stimulierung aus ihrer Umgebung über zwei Arten von Rückmeldungen: eine lokale Rückmeldung (*feedback*) ist Teil der Konstruktion des Agenten, von dessen Gestalter vorherbestimmt und damit deterministisch. Die globale Rückmeldung ist ein Resultat der Interaktion zwischen den einzelnen Agenten des Vielkomponentensystems, welches deterministisch nicht sicher vorhersagbar ist und auf die eine Antwort nicht explizit im Agenten vordefiniert werden kann.

Ferbers Konzept der reaktiven Agenten hat in der KI starken Widerspruch ausgelöst. Insbesondere der Verzicht auf eine Planungskomponente macht die Zuordnung des Konzeptes unter den Begriff des “autonomen Agenten” zweifelhaft – bei einem reinen stimulus/response-Mechanismus kann von einer Autonomie des Agenten in Bezug auf die Wahl seiner Verhaltensweise keine Rede sein. Diese Ansicht greift jedoch nur dort, wo die Architektur des einzelnen Agenten in den Mittelpunkt gestellt wird. Ferber geht es stattdessen um die Zielverfolgung und Koordination mit vielen Agenten - für die von ihm beschriebenen Systeme sind als Untergrenze der Autonomie reaktive Agenten ausreichend. Reaktive Agenten finden sich als Konzept des internen Modells besonders in der Koordinationssicht der Multi-Agenten-Systeme.

Für den Einsatz auf elektronischen Marktplätzen stellt sich die Frage, ob das interne Modell der Agenten reaktiv oder deliberativ ausgeprägt sein soll und welche Ebene (im Sinne des k.-Ebene-Konzepts) implementiert sein muß, damit Software-Agenten überhaupt sinnvoll in einem Markt agieren können.

Über das Konzept der **“Zero Intelligence” (ZI) Agenten** haben die reaktiven Agenten in der ökonomischen Simulation Einzug gehalten (Gode/Sunder 1993, Cliff/Bruten 1998). Dort konnte gezeigt werden, dass bereits mit sehr einfacher Modellbildung ökonomisches Verhalten beobachtet werden kann. In der Simulation von Gode/Sunder handelt es sich um eine sogenannte kontinuierliche Doppelauktion (*Continous Double Auction*, CDA), in der alle Beteiligten ihre Gebote per Ausruf (*Broadcast*) an alle anderen übermitteln. Die Verkäufer führen eine Holländische Auktion durch, indem sie ihre Gebote im Zeitablauf senken, die Käufer gleichzeitig eine Englische Auktion, bei der sie ihre Kaufgebote im Wert steigern. Gode/Sunder bildeten durch Software-Agenten ähnliche Experimente mit menschlichen Marktteilnehmern nach, die Smith (1962) durchgeführt hatte. Die Käufer- und Verkäuferagenten rufen zu Beginn zufällige Preise aus und folgen danach jeweils auf- oder absteigend dem Auktionsverfahren der New Yorker Aktienbörse (NYSE).

Gode/Sunder führten fünf Experimente durch, bei denen jeweils zwei verschiedene Arten von ZI-Agenten und menschliche Teilnehmer gleichen Marktsituationen ausgesetzt wurden. Alle Agenten besitzen einen Limitpreis, der ihre Wertschätzung des Gutes ausdrückt. Sogenannte ZI-U-Agenten (für „*unconstrained*“) geben Gebote zwischen 1 und 200 Geldeinheiten (GE) mit gleicher Wahrscheinlichkeit ab. ZI-C-Agenten („*constrained*“) geben Gebote ab, die für Verkäufer zwischen dem Limitpreis und 200 GE liegen. Die Gebote von ZI-C-Käufern liegen zwischen 1 GE und dem Limitpreis. ZI-U-Agenten können daher Verluste machen, ZI-C-Agenten nicht, da sie einem Budgetzwang (*budget constraint*) unterliegen. Es wird jeweils nur 1 Stück des Gutes gehandelt. Die Software-Agenten besitzen überhaupt keine Adaptivität oder Lernfähigkeit, ihre „Intelligenz“ kann daher noch unterhalb der 0.-Ebene (nach Vidal/Durfee 1996) eingeordnet werden.

In der Abbildung 11 zeigt die linke Grafik die Anfangsausstattung nach Güterpreis und Gütermenge der Agenten, die rechte Grafik den zeitlichen Verlauf des CDA-Mechanismus über 6 Handelstage. P_0 bzw. Q_0 kennzeichnen Gleichgewichtspreis und -menge, auf die ein neoklassischer Markt konvergieren sollte. Die drei Reihen zeigen das gleiche Experiment mit von oben nach unten ZI-U-Agenten, ZI-C-Agenten und menschlichen Experimentteilnehmern, das zusammengefaßt auch bei Cliff (1997, S. 23ff.) beschrieben ist. Der Unterschied zwischen den einzelnen Ergebnissen liegt nicht in den internen Modellen der Agenten begründet, sondern einzig in der Organisation des CDA-Verfahrens (Gode/Sunder 1993, S. 123). In den Experi-

menten mit ZI-U-Agenten ergaben sich keine systematischen Muster und keine Tendenz, auf eine bestimmte Preisebene zu konvergieren (Gode/Sunder 1993, S. 126), wie die beiden oberen Grafiken zeigen.

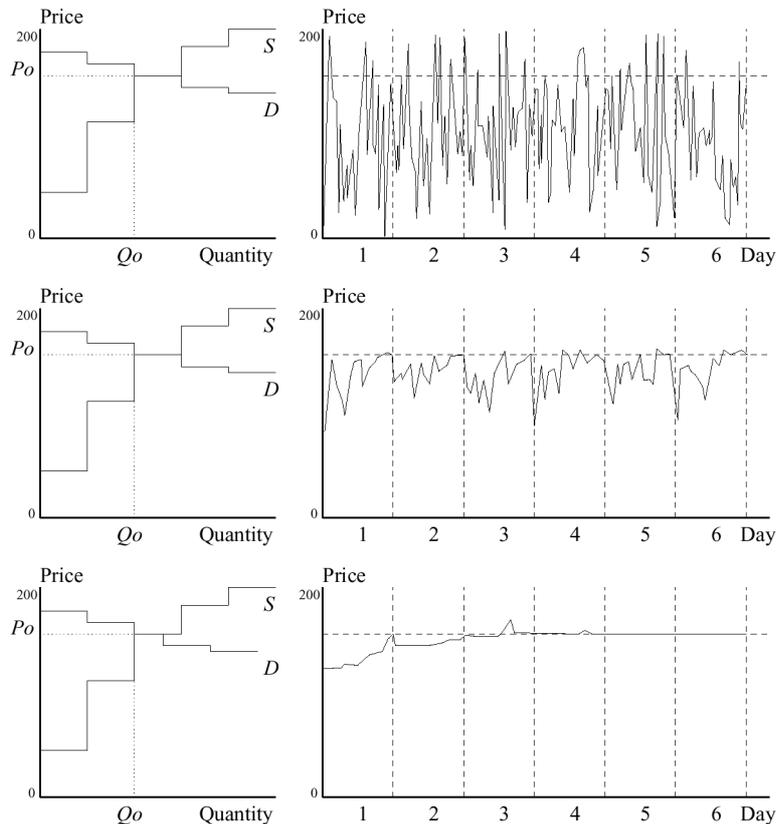


Abb. 11: Ergebnisse der Experimente von Gode/Sunder (1993), aus Cliff (1997, S. 25)

Auf der anderen Seite unterlagen die menschlichen Teilnehmer einem Budgetzwang und passten ihre Preise adaptiv während des Experiments an. Ihr Streben nach Nutzenmaximierung führte zu einer schnellen Konvergenz der CDA auf den aus der Anfangsausstattung an Preisen und Mengen errechenbaren Gleichgewichtspreis. Dies ist in Abb. 11 in der rechten unteren Grafik dargestellt, wobei aus der Grafik unten links eine zu den ZI-U-Agenten vergleichbare Anfangsausstattung abzulesen ist.

Die interessanteste Marktsituation ergibt sich aber beim Einsatz von ZI-C-Agenten (mittlere Grafiken in Abb. 11). Drei Ergebnisse werden von Gode/Sunder (1993, S. 129) bzw. Cliff (1997, S. 24) angeführt. Zum einen lernten die Agenten nicht, so dass sie, anders als die Menschen, jeden Handelstag erneut mit Preisen ausserhalb des Gleichgewichts begannen. Zweitens ist die Volatilität ihrer Preise zwischen den ZI-U-Agenten und den Menschen angesiedelt,

was eine Folge der Budgetbeschränkung sein muß. Drittens konvergieren die Preise im Verlauf eines Handelstages in die Nähe des Gleichgewichtspreises. Das Ergebnis der Experimente und ihre Relevanz zur vorliegenden Arbeit fasst Cliff folgendermaßen zusammen:

„Gode and Sunder's results appear to indicate that groups of so-called 'zero-intelligence' trading agents, that simply announce random prices for bids and offers, can give results that are remarkably similar to the performance of human traders. The implication drawn from their work is that the structure of the market plays a large part in determining the observable behavior of the agents in that market, and the intelligence (or lack of it) of the agents in the market is secondary. If this is true, the relevance of developing computational mechanisms for bargaining agents is highly questionable.“
(Cliff 1997, S. 4)

Der Grad an Intelligenz, der für die Bewältigung einer Marktsituation benötigt wird, wäre demzufolge verschwindend gering. Zeitliche, intellektuelle und monetäre Investitionen in die Entwicklung komplexer interner Modelle von Software-Agenten brauchten nicht getätigt zu werden. Auch für „menschliche“ Märkte stellen diese Ergebnisse eine analytische Herausforderung dar:

„Thus, the main message of Gode and Sunder's paper is that allocative efficiency appears to be almost entirely a product of market structure: prior to these experiments, it seemed fair to assume that the high efficiency of human markets is a consequence of human cognitive prowess; in light of Gode and Sunder's results, such assumptions are clearly highly doubtful.“ (Cliff 1997, S. 26)

Cliff (1997) konnte zeigen, dass die Konvergenz der Preise auf den Gleichgewichtspreis der ZI-C-Agenten ein Artefakt der Überschneidung der Wahrscheinlichkeitsdichtefunktion jeweils der Kauf- und Verkaufspreise darstellt (Cliff 1997, S. 27ff). Für den genauen Beweis sei auf die Ausführungen in der Quelle verwiesen. Cliff (1997) bzw. Cliff/Bruten (1998) entwickelten daraufhin sogenannte „Zero-Intelligence-Plus“ (ZIP) Agenten, die lediglich ihre Gewinnschwelle (profit margin) im Verlauf der Verhandlungen veränderten. Die ZIP-Agenten ermitteln ihre Gebotspreise über folgende Formel: $p_i(t) = \lambda_{i,j}(1 + \mu_i(t))$, wobei $p_i(t)$ der ausgerufene Gebotspreis des Agenten i zum Zeitpunkt t für das Gut j ist. $\lambda_{i,j}$ stellt den Limitpreis dar und $\mu_i(t)$ eine prozentuale Gewinnmarge, die adaptiv im Zeitablauf angepasst wird. Eine Gewinnmarge von +5% für Verkäufer bedeutet einen Gebotspreis, der um 105% über dem eigenen Limitpreis liegt; Käufer mit einer Gewinnmarge von -5% setzen Kaufgebote von 95% des Limitpreises ab. Das Adaptionsverfahren für $\mu_i(t)$ entspricht dem ähnlichen Mechanismus von Preist (1998), der in Kapitel 3.5.3.3 "Adaptive Ansätze" auf Seite 120 erläutert wird

und bei Cliff (1997, S. 42ff.) detailliert beschrieben ist. In diesem Zusammenhang ist nicht die Adaption als solche, sondern ihre Konsequenz für die Gestaltung des internen Modells von Interesse.

Der Anlass für die Veränderung ist die Beobachtung der Marktumgebung durch den einzelnen Agenten. Dabei werden diejenigen Preise, zu denen Gütertransaktionen stattfinden, mit den eigenen Geboten verglichen. Falls ein Verkäuferagent feststellt, dass sein Gebot unterhalb der aktuellen Abschlußpreise liegt, so wird er beim nächsten Mal ein höheres Gebot absetzen, in dem er seine Gewinnschwelle erhöht. Käuferagenten, die feststellen, dass ihre Gebote zu hoch liegen, werden ihre Gewinnmarge ebenfalls absolut erhöhen²¹ und ihre Gebotspreise dadurch senken. Anzumerken ist hier, dass die Agenten die Marktumgebung als solche beobachten, die einzelnen Gebote jedoch nicht bestimmten Agenten zuordnen. In der Kategorisierung von Vidal/Durfee (1996) handelt es sich bei den ZIP-Agenten um interne Modelle der 0.-Ebene. Cliff (1997, S. 46) bzw. Cliff/Bruten (1998, S. 28) zeigen, dass die ZIP-Agenten auf einen Gleichgewichtspreis konvergieren (siehe Abb. 12).

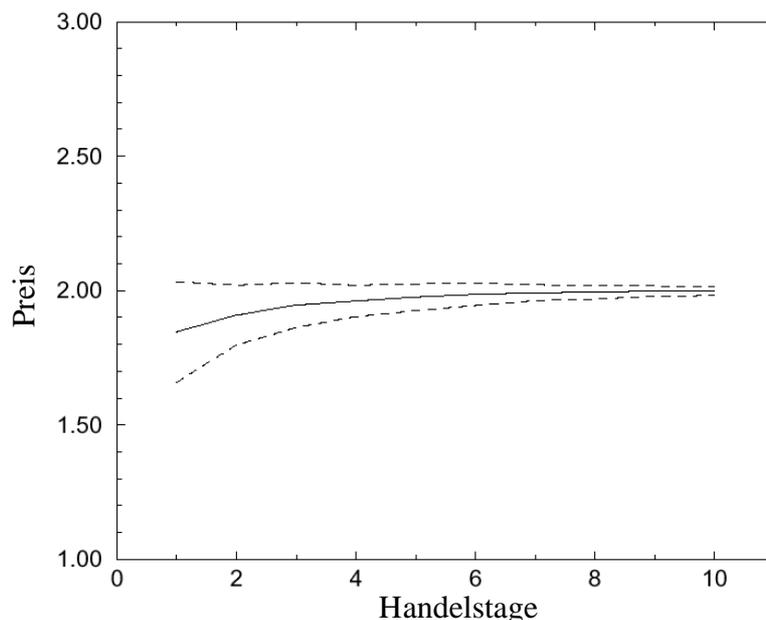


Abb. 12: Konvergieren von ZIP-Agenten auf einen Gleichgewichtspreis²²

- 21 Käuferagenten besitzen in diesem Verfahren eine negative Gewinnmarge. Eine absolute Erhöhung bedeutet z.B. eine Veränderung von -5% auf -8% und damit eine Verminderung des Gebotspreises von 95% auf 92% des Limitpreises.
- 22 Diese Grafik zeigt das konsolidierte Ergebnis von 50 Experimenten. Die Abszisse zeigt den Zeitverlauf in Handelstagen, die Ordinate den Preis in Geldeinheiten. Die mittlere durchgezogene Linie ist der Mittelwert, die gestrichelten Linien jeweils die einfache Standardeinweichung nach oben und unten. (Cliff 1997, S. 34, Figure 25).

Aus diesem Ergebnis folgern Cliff/Bruten (1998), dass zur Generierung ökonomischen Verhaltens bereits eine geringe, aber nicht zu geringe Komplexität des internen Modells ausreicht:

„Gode and Sunder's work was an important contribution to the field of experimental economics, providing an absolute lower limit on the mechanistic complexity of trading agents, and demonstrating that allocative efficiency is a poor indicator of the intelligence of agents in a double-auction market. [...] The failings of the zi-c traders indicates a need for bargaining mechanisms more complex than the simple stochastic generation of bid and offer prices. The work on ZIP traders should be viewed as a preliminary sketch of what forms such bargaining mechanisms might take. The ZIP traders are more complex than Gode and Sunder's ZI-C traders, but only slightly, and in any case are manifestly much less complex than humans. Nevertheless, the results from the zip traders, both in terms of equilibration and profit dispersion, are clearly closer to those from human experimental markets than are the results from ZI-C traders. It is reassuring to see that very simple mechanisms can give such human-like results, but there is much further work that could be done in exploring behavior of zip traders in more complex market environments, and in attempting to extend the behavioral sophistication of such traders without unduly adding to their complexity.“ (Cliff/Bruten 1998, S. 34)

Für das Design des internen Modells von Agenten bedeuten diese Ergebnisse, dass in der vorliegenden Arbeit von 0.-Ebene-Agenten ausgegangen wird, die ihre Informationen aus der Marktumgebung beziehen, ohne diese in andere Agenten aufzulösen. Diese Erkenntnis vereinfacht die Implementation agentenbasierter elektronischer Märkte und vor allem handelnder Agenten um ein Vielfaches. Als Umkehrschluss ließe sich ebenfalls festhalten, dass Methoden der symbolischen KI (2.-Ebene-Agenten) für das erfolgreiche Bestreiten von Marktsituationen nicht zwingend notwendig sind.

3.2 Mehragentensysteme

In fast allen Anwendungsfällen, in denen Software-Agenten eingesetzt werden, handelt es sich um Realisierungen mit mehr als einem Agenten. Die Anzahl der Agenten variiert dabei zwischen etwa 10 (z.B. im Agent-enhanced Workflow (vgl. Jennings et al. 1998) und vielen tausenden Agenten (z.B. in der Information Economy von IBM (Kephart et al. 2000)). Damit wird die Dynamik der Umwelt vor allem von den Aktionen und Entscheidungen anderer Agenten bestimmt, die alle Teil des gleichen technischen Vielkomponentensystems sind. Um in einem solchen System agieren zu können, ist eine gewisse Mindestkomplexität im Entwurf des inter-

nen Modells sowie der Umgebung erforderlich. Gasser (1991, S. 390) nennt sechs Problemfelder, die für eine Implementation eines Mehragentensystems²³ betrachtet werden müssen. Anhand des Schichtenmodells der Telematik lassen sich diese Problemfelder folgendermaßen zusammenfassen:

1. **Kommunikation** – wie können Agenten dazu gebracht werden, miteinander zu kommunizieren? Welche Kommunikationsprotokolle sollen verwendet werden? *„How to enable agents to communicate and interact: what communication languages or protocols to use, and what and when to communicate.“*
2. **Kooperation** und Interaktion – welche Sprache sollen die Agenten verwenden, um zu interagieren und ihre Vorgehensweise zu kombinieren? *„How to enable individual agents to represent and reason about the actions, plans and knowledge of other agents in order to coordinate with them; how to reason about the state of their coordinated process (e.g., initiation and termination). [...] How to ensure that agents act coherently in making decisions or taking action, accomodating the nonlocal effects of local decisions and avoiding harmful interactions.“*
3. **Koordination** – wie stellt man sicher, dass die Agenten sich koordinieren, um eine kohe-rente Lösung für das Gesamtproblem zustande zu bringen? *„How to formulate, describe, decompose, and allocate problems and synthesize results among a group of intelligent agents.“* und *„How to recognize and reconcile disparate viewpoints and conflicting intentions among a collection of agents trying to coordinate their actions.“*
4. Insgesamt über **alle Schichten** - wie entwickelt und implementiert man Mehragentensysteme? *„How to engineer and construct practical DAI systems; how to design technology platforms and development methodologies for DAI.“*

Im folgenden sollen existierende Konzepte und Realisierungen für diese Fragestellungen vorgestellt und diskutiert werden. Da sich diese Arbeit vor allem mit der ökonomischen Koordination beschäftigt, beginnt der nächste Abschnitt mit Konzepten der Koordination in Mehragentensystemen.

3.3 Koordination in Mehragentensystemen

Im Fall der Betrachtungen eines aus mehreren Agenten bestehenden Systems kommt zu den Grundbestandteilen der Agenten (Internes Modell, Sensoren und Effektoren) sowie der Umge-

²³ Aufgrund der nicht eindeutigen Sprachregelung in der VKI wird im folgenden von einem Mehragentensystem gesprochen, wenn die generelle Klasse eines aus vielen autonomen Komponenten bestehenden Informationssystems gemeint ist; ein Multi-Agenten-System hingegen wird einem Distributed Problem Solving-Ansatz gegenübergestellt und bezeichnet eine spezielle Ausprägung eines Mehragentensystems.

bung noch das Gesamtsystem selbst als Betrachtungsobjekt hinzu. Dieses Gesamtsystem stellt eine Gesellschaft oder Population von Agenten dar. Die isolierte Betrachtung des Aufbaus einzelner Agenten (oder von Agentenpaaren) entspricht nicht der Komplexität des Problems:

„Much of traditional AI has been concerned with how an agent can be constructed to function intelligently, with a single locus of internal reasoning and control implemented in a Von Neumann architecture. But intelligent systems do not function in isolation - they are at the very least a part of the environment in which they operate, and the environment typically contains other such intelligent systems. Thus, it makes sense to view such systems in societal terms“ (Huhns/Stephens 1999, S.81).

Autonome Agenten und Mehragentensysteme stellen einen Ansatz zur Analyse, zum Design und zur Implementierung von komplexen Softwaresystemen dar (Bradshaw 1997). Für die drei Objekte Agenten, Umgebung und System lassen sich die wesentlichen Forschungsfragen etwa so zusammenfassen (Durfee/Rosenschein 1994, S.7):

1. Charakteristika des einzelnen Agenten: Handelt der Agent rational? Sind seine Präferenzen allgemein bekannt? Werden sie von anderen geteilt? Welche Fähigkeiten hat der Agent? Sind diese anderen bekannt?
2. Charakteristika der Umgebung: Ist die Umgebung statisch? Ist sie geschlossen? Feindselig? Sind die Ergebnisse der Handlungen von den Agenten vorhersagbar? Sind sie zeitlich beschränkt?
3. Charakteristika des Gesamtsystems: Kann ein gerechter Zugang zu Ressourcen und ein faires Verhalten der Agenten sichergestellt werden? Müssen Annahmen über bestimmte externe Eigenschaften, z.B. Antwortzeitverhalten der Software, getroffen werden?

Während die ersten beiden Fragengruppen bereits bei Betrachtung eines einzelnen Agenten in Interaktion mit seiner Umwelt von Bedeutung sind, ist der Aufbau und das Verhalten eines Gesamtsystems aus vielen Agenten die zentrale Forschungsfrage der **Verteilten Künstlichen Intelligenz (VKI)**. Grundlage der VKI ist die Metapher der kooperierenden bzw. kollaborierenden Spezialisten, d.h. ein größeres Problem durch Zusammenarbeit autonomer Einheiten zu lösen. Eine Definition der VKI lautet:

*„Distributed AI involves agents who act in an environment to comprise a system.“
(Durfee/Rosenschein 1994, S.7).*

Diese Dreiteilung zwischen Agenten, Umgebung und dem Gesamtsystem kennzeichnet den Entwurf und Konstruktion von Mehragentensystemen. Vorteile eines Mehragentensystems werden in solchen Domänen gesehen, in denen Probleme gelöst werden sollen, die (1) aufgrund von Ressourcenbeschränkungen oder des Sicherheitsrisikos eines zentralen Systems zu

umfangreich für einen einzelnen Agenten sind, (2) Lösungen für verteilte Probleme gesucht werden, oder in technischer Hinsicht (3) verbesserte Modularität (um Komplexität reduzieren zu können), (4) erhöhte Ablaufgeschwindigkeit (durch Parallelverarbeitung), (5) gesteigerte Zuverlässigkeit (durch Redundanz der Agenten), (6) größere Flexibilität und (7) Wiederverwendbarkeit verlangt wird (Nwana 1996, S.214).

Im folgenden werden zwei sich ergänzende Ansätze beschrieben. Sowohl der Ansatz des Verteilten Problemlösens (**Distributed Problem Solving, DPS**) als auch der Multi-Agenten-Systeme (**Multi-Agent Systems, MAS**) beschäftigt sich mit der Beschreibung von Systemen, welche aus vielen interagierenden Agenten bestehen.

“Distributed AI can be broken down into two related areas. These two areas constitute distinctions between research areas; they are not really appropriate as descriptions of running systems” (Rosenschein/Zlotkin 1994b, S.31).

Der Unterschied zwischen beiden Ansätzen besteht in der Ausprägung der Weisungsbefugnis zu den ansonsten autonom arbeitenden Agenten:

„In distributed problem solving, there is assumed to be a single body that is able, at design time, to influence the preferences of all agents in the system. [...] In multiagent systems, you again have multiple agents in a distributed system, but you do not assume that there is a single designer who stands behind all of them, or put another way, you do not assume that the individual agents have a group sense of utility. Each of the agents in the system can be working at different goals, even conflicting goals. [...] The agents’ preferences arise from distinct designers” (Rosenschein/Zlotkin 1994b, S. 32).

Bei DPS kann durch den Designer ein explizit kooperatives Verhalten der Agenten notfalls erzwungen werden, während bei MAS im Prinzip von einer antagonistischen Situation ausgegangen wird, in der sich Kooperation ausschließlich durch individuelle Nutzenüberlegungen ergibt.

Kooperation zwischen vielen Agenten hat meist den Zweck, ein gemeinsames globales Ziel zu erreichen, von dem die Agenten nicht notwendigerweise Kenntnis haben müssen (Lee et al. 1997, Jennings 1996). Bei Nichtexistenz eines solchen Zieles dient die Kooperation zumindest der Erreichung kohärenter Aktivität der Agentengemeinschaft (Schumacher et al. 1999). Für diesen Zweck sind in der Forschung der Verteilten Künstlichen Intelligenz verschiedene Koordinationstechniken erfunden worden, die nach Nwana (1996) in vier Kategorien unterteilt werden können:

1. Techniken des “*Organizational Structuring*” (Durfee et al. 1987, Hayer-Roth 1985, Werkman 1990, Kearney et al. 1994): Organisatorische Muster werden a priori definiert, um implizit Verantwortlichkeiten, Fähigkeiten, Konnektivität und Kontrollfluß der Agenten festzulegen. Dabei handelt es sich um langfristige Beziehungen zwischen den Agenten. Solchen Systemen liegt häufig ein Master/Slave- bzw. Client/Server-Paradigma zugrunde.
2. In Multi-Agent Planungstechniken (Georgeff 1983) erstellen Agenten einen Plan, der sämtliche Aktionen und Interaktionen beinhaltet, die notwendig sind, um ihre jeweiligen Ziele zu erreichen. Die Planung kann dabei zentral oder dezentral erfolgen.
3. Das Kontraktnetz-Protokoll (Smith 1980, Parunak 1989, Huhns/Singh 1994, Conry et al. 1986) basiert auf der Metapher einer dezentralisierten Marktstruktur. Ein Kontraktnetz ist ein Allokationskonzept, um Aufträge verteilten Prozessoren innerhalb eines verteilten Computersystems zuzuweisen. Seine Funktionsweise wird in Kapitel 3.3.1 genauer erläutert.
4. In Verhandlungstechniken rasonieren Agenten über die Ziele und Strategien anderer Agenten (Sycara 1989): „*...negotiation is the communication process of a group of agents in order to reach a mutually accepted agreement on some matter*” (Bussmann/Müller 1992).

Die ersten beiden Kategorien (Planungstechniken und *Organizational Structuring*) gehen von einer unbedingten Kooperativität der Agenten zueinander aus. Sie sind vor allem in solchen Domänen anwendbar, in denen das Problem a priori strukturiert werden kann und alle Lösungsalternativen formal dargestellt werden können. Diese Koordinationstechniken werden verworfen, weil ihre Annahmen zur Koordination den Eigenschaften des Anwendungsproblems nicht gerecht werden. Kontraktnetze gehen ebenfalls von kooperativen Beziehungen der Agenten aus, entscheiden jedoch ad hoc über den zu wählenden Lösungsweg. Sie werden häufig in Mehragentensystemen eingesetzt, die dem Paradigma des Distributed Problem Solving folgen. Verhandlungstechniken hingegen sehen die Agenten als Konkurrenten, die eigene Ziele notfalls auch zum Schaden anderer Agenten verfolgen. Mehragentensysteme, die Verhandlungstechniken einsetzen, sind in den Bereich der Multi-Agenten-Systeme einzuordnen. Im folgenden werden DPS und MAS als komplementäre Paradigmen der Gestaltung von Mehragentensystemen vorgestellt und die Eignung von DPS/Kontraktnetzen bzw. MAS/Verhandlungstechniken für das hier behandelte Szenario diskutiert.

3.3.1 Koordination durch verteiltes Problemlösen

Verteiltes Problemlösen (engl. Distributed Problem Solving, DPS) geht von der Fragestellung aus, wie ein bekanntes Problem durch eine Menge von Modulen (Knoten) gelöst werden kann.

Diese kooperieren miteinander, indem sie das insgesamt bekannte Wissen über das Problem sowie sich entwickelnde Lösungen miteinander austauschen. In solchen Systemen, in denen es möglich ist, exakt festzustellen, was jeder Agent gegenwärtig tut und was er in der Zukunft ausführen wird, wäre es möglich, konflikthafte und überflüssige Aktionen zu vermeiden und das System könnte perfekt koordiniert werden (Malone 1987, Corkill/Lesser 1986).

Ausgehend von der Annahme vollkommener Information über das Problem wird daraufhin das verteilte System gestaltet, welches das Problem lösen soll:

„Early DPS work thus concentrated on harnessing and applying the power of networked systems to a problem, as exemplified by the contract net approach for decomposing and allocating task in a network. [...] In all of this work, the emphasis was on the problem, and how to get multiple agents to work together to solve it in a coherent, robust, and efficient manner” (Durfee/Rosenschein 1994, S.2).

Die Agenten werden grundsätzlich als kooperativ angenommen. Ein Gesamtproblem wird immer nach Kompetenz auf die einzelnen Agenten verteilt.

„In Distributed Problem Solving, we typically assume a fair degree of coherence is already present: the agents have been designed to work together; or the payoffs to self-interested agents are only accrued through collective efforts; or social engineering has introduced disincentives for agent individualism; etc.“ (Durfee 1999, S. 121).

Diesen Implementierungen ist gemeinsam, dass alle Agenten auf die Erreichung eines gemeinsamen Zieles hinarbeiten. Der Designer der Agenten kann im vorhinein Protokolle für die Kooperation zwischen den Agenten entwickeln. Die Anzahl der Agenten ist nicht groß und die Agenten können kommunizieren und eigene Berechnungen anstellen.

„DPS has focused on getting external properties such as robust and efficient performance under varying environmental conditions, from agents with established properties. DPS asks how can a particular collection of agents attain some level of collective performance if the properties of their environment are dynamic and uncontrollable“ (Rosenschein/Durfee 1994, S.8).

Das Problem wird immer als bekannt angenommen und auch der Lösungsweg ist zumindest theoretisch oder konzeptionell bekannt. Die Forschung in DPS konzentriert sich auf die Organisation und Planung, den Austausch von Zielen und Teilergebnissen mit dem Ziel, *“to increase the coherence of collective activity without increasing the overhead significantly” (Durfee/Rosenschein 1994, S.3).*

Prinzipiell können drei unterschiedliche, sich gegenseitig ergänzende Vorgehensweisen beschrieben werden: die Aufteilung von Aufgaben (Task Sharing), die Zusammenführung von Ergebnissen (Result Sharing) und die Planung des weiteren Vorgehens in und für ein verteiltes System (Distributed Planning). Aus deren Beschreibung lassen sich die generellen Annahmen, die einem DPS-System zugrunde liegen, dann zusammenfassen.

Task Sharing beschreibt die Parallelisierung von Aufgaben durch Delegieren von Unteraufgaben. Der delegierende Agent muß eine Dekomposition seiner Aufgabe durchführen, dann die Unteraufgaben delegieren und schließlich die Ergebnisse wieder zusammenführen. Diese Vorgehensweise kann über mehrere Ebenen rekursiv fortgeführt werden. Montgomery/Durfee (1993) zeigen diese Vorgehensweise beispielhaft anhand des Turm-von-Hanoi-Problems.

Der am häufigsten angewendete Mechanismus zum Task Sharing sind Kontraktnetze (Smith 1980). Ein Kontraktnetz ist ein Allokationskonzept, um Aufträge verteilten Prozessoren innerhalb eines verteilten Computersystems zuzuweisen. Grundsätzlich funktioniert ein Kontraktnetz folgendermaßen: Ein Managerknoten hat Aufträge, die innerhalb des Verteilten Systems berechnet werden müssen. Er versendet sogenannte Ankündigungen (*Announcements*) an alle verfügbaren Knoten oder (seltener) direkt an diejenigen Knoten, die seines Wissens nach diese Aufträge ausführen können. Die Knoten (bzw. Bieter) senden je nach ihren Fähigkeiten unterschiedliche Gebote an den Managerknoten zurück. Dieser evaluiert die eingegangenen Gebote und gibt den Zuschlag für den oder die Aufträge dem bevorzugten Knoten, der als Kontraktor bezeichnet wird. Die Methoden für die Formalisierung der Gebote und für die Evaluierung hängen stark vom Anwendungskontext ab.

In einem Fertigungsumfeld handelt es sich z.B. um Fertigungsaufträge und um Arbeitstationen bzw. Produktionsressourcen als Knoten. Durch die Aushandlung im Kontraktnetz soll für jeden Auftrag die beste Maschine gefunden werden. Nicht alle Maschinen werden jeden Auftrag ausführen können, aber durch den Bekanntmachung aller Aufträge im Kontraktnetz können die Maschinen darüber autonom entscheiden.

Zwei grundsätzliche Probleme werden in Bezug auf Kontraktnetze angeführt (Tilley 1995, S.229):

1. Das System reagiert immer in Bezug auf seinen derzeitigen Zustand und führt keinerlei Planung durch. Damit besteht zwar keine Abhängigkeit von der Verfügbarkeit bestimmter Maschinen, allerdings kann das zukünftige Verhalten des Gesamtsystems nicht einbezogen werden.
2. Das Kontraktnetz optimiert nicht in Bezug auf ein möglicherweise existentes globales Ziel. Es gibt keine unterschiedlichen Prioritäten der Aufträge. Dadurch kann ein Auftrag niedri-

ger Priorität durchaus dringender benötigte Ressourcen belegen und somit die Produktivität verschlechtern. Wenn die Reihenfolge der Auftragsvergabe durch einen zentralen Managerknoten festgelegt wird, so ist die Priorisierung ein Problem des Managerknotens und nicht des Kontraktnetzmechanismus. Andererseits gibt es Ansätze für mehrstufige Kontraktnetze, bei denen alle Maschinen Unteraufträge vergeben dürfen und damit auch eine Umbelegung einfacher möglich wird.

Nachdem die Aufgaben verteilt und durch die einzelnen Agenten bearbeitet worden sind, müssen durch Verfahren des **Result Sharing** Agenten u.U. Ergebnisse anderer Agenten in ihre eigenen einbeziehen. Das ist einfach beim Turm-von-Hanoi-Problem, weil das Problem klar abgegrenzt ist, die komplette Information zur Berechnung vorliegt und die Agenten unabhängig voneinander operieren. Im Gegensatz dazu steht z.B. eine realistische Transportlogistik-Anwendung wie das *Distributed Vehicle Monitoring* (Durfee 1999, S. 123), wo den Agenten unvollständige Informationen vorliegen und lokale Ergebnisse häufig Neuberechnet werden müssen, weil sich die Umwelt ständig ändert. Die richtige Antwort ergibt sich nach einer bestimmten Zeitdauer, aber bis dahin muß mit fehlerbehafteten Zwischenergebnissen gerechnet werden (vgl. den Begriff der *Functionally-Accurate Cooperation* bei Lesser/Corkill 1981, S.7). Agenten müssen vorläufige Ergebnisse kommunizieren können und diese in ihre eigenen Berechnungen auch als vorläufig (*tentative*) einbauen.

Die Planung zukünftigen Vorgehens ist schließlich Aufgabe des **Distributed Planning**. Die zentrale Erstellung eines später auf viele Agenten zu verteilenden Plans ist im wesentlichen ein Task Sharing-Problem, welches durch einen „*Partial Order Planner*“, der die Teilaufgaben in eine Ordnung bringt und sie den Subagenten zur Ausführung übergibt, gelöst wird. Die verteilte Erstellung eines zentralen Gesamtplans ist hingegen eine Aufgabe des Result Sharing. Das Problem, einen verteilten Plan zu erzeugen, der von vielen Einheiten ausgeführt wird, kann selbstverständlich ebenfalls in verteilter Weise gelöst werden. Für diese komplexeste Kombination, die verteilte Erstellung eines verteilten Plans, sei auf Durfee (1999) hingewiesen.

DPS geht von drei grundsätzlichen Annahmen aus, die diesen Zweig der VKI für das Problem eines offenen elektronischen Marktes ungeeignet erscheinen lassen (Durfee/Rosenschein 1994, S.4): die Annahme benevolenten Handelns, die Annahme gemeinsamer Ziele der Agenten und die Annahme eines zentralen Schöpfers.

Rosenschein/Genesereth (1985) argumentieren, dass ein DPS-System immer von **benevolentem Handeln** der einzelnen Agenten charakterisiert wird. Als „benevolent“ wird ein Agent eingestuft, wenn er grundsätzlich anderen Agenten helfen *möchte*, ohne seinen eigenen Nutzen zu betrachten²⁴. Derartige Agenten werden immer wahrheitsgemäß auf Dienstanforderungen

reagieren und müssen nicht durch Aushandlung oder das Anbieten von Konzessionen von der Übernahme einer Aufgabe überzeugt werden. Das Kontraktnetz-Protokoll z.B. weist Aufgaben den Agenten nach technischer Fähigkeit und Verfügbarkeit zu, ohne Nutzenüberlegungen bzw. Nutzenübertragungen (*transfer of utility*) durchzuführen. Auch bei benevolenter Kooperation kann daher nicht davon ausgegangen werden, dass eine optimale Koordination zustandekommt. Für das Kontraktnetz-Protokoll stellt sich bspw. problematisch dar, dass wichtige Aufgaben nicht zugewiesen werden können, wenn Agenten mit unwichtigen Aufgaben beschäftigt sind, die eigentlich hätten delegiert werden können (Durfee/Rosenschein 1994, S.4).

Eine weitere wichtige Eigenschaft ist das Vorhandensein eines **gemeinsamen (globalen) Ziels**:

„A motivation for benevolence among agents is having a common goal. That is, if the agents all value the same outcome of group activity, they will each attempt to contribute in whatever way they can to the global outcome.“ (Durfee/Rosenschein 1994, S.4).

Wenn ein übergeordnetes, globales Ziel erreicht werden soll, kann dieses (mgln. über mehrere Stufen) in Teilziele heruntergebrochen und den einzelnen Agenten zugewiesen werden. Die Aufgabe jedes einzelnen Agenten ist es dann, *„to help the system converge on the global solution, [...] trying to help the others form good local interpretations as quickly as possible“ (Durfee/Rosenschein 1994, S.4).* Die Zielerreichung jedes Agenten kann als Voraussetzung sowohl in Bezug auf dessen eigenes Teilziel wie auch bezüglich der Erreichung des Gesamtzieles gemessen werden. Die Teilziele müssen aber so gestaltet sein, dass lokale Entscheidungen nicht zu einem inkohärenten Gesamtzustand führen, was eine strikte Regelung der Verantwortlichkeiten erfordert.

Schließlich wird von einem **zentralen Designer** ausgegangen:

„Most recently, the argument has been put forward that a DPS system is a system with a centralized designer. This perspective subsumes the previous assumptions, since the central designer’s goals would be embodied in the agents (giving them common goals) and the designer, being concerned about getting the parts to work as a whole, would likely make each agent benevolent. Moreover, since the designer has the „big picture“, the preferences/foci of the agents could be calibrated and the mechanisms for

24 Benevolenz ist nicht gleichbedeutend mit Kooperativität: ein Agent, der kooperativ ist, hilft anderen Agenten nur, wenn er damit seinen eigenen Nutzen optimiert. Ein benevolenter Agent hilft immer, ohne seinen eigenen Nutzen abzuwägen. Wie dieser Nutzen gemessen wird, kann beim helfenden wie geholten Agenten unterschiedlich sein; für einen „Heilsarmee“-Agenten würde allein der Vorgang des Helfens bereits eine Nutzenverbesserung darstellen.

expressing and acting on those preferences can be standardized.“ (Durfee/Rosenschein 1994, S.5)

Solange ein einzelner Designer/Schöpfer das Gesamtsystem konzipiert und implementiert, werden die anderen Annahmen benevolenten Handelns und gemeinsamer Ziele aus der Eigennutzverfolgung des Designers heraus nicht verletzt. Inwiefern diese Vorgehensweise zu einer erfolgreichen Problemlösung führt, kann hieraus nicht gefolgert werden. Die Konsistenz des Systems alleine kann keine Aussage über die Qualität der Problemlösung treffen.

In dem Szenario offener elektronischer Märkte werden alle diese Annahmen verletzt. Da die Agenten notwendigerweise kompetitiv zueinander auf sehr ähnlichen Nutzenwertskalen agieren, kann Benevolenz ausgeschlossen werden (vgl. Rasmusson/Janson 1999, S. 144). Auch ein gemeinsames Ziel existiert nicht, wenn die reine Aufrechterhaltung der Koordination als solcher ausgeschlossen wird. Schließlich gibt es keinen zentralen Designer, sondern die einzelnen Agenten werden von ihren jeweiligen Prinzipalen mit einer Strategie ausgestattet und die Existenz antagonistischer Ziele kann nicht von vornherein ausgeschlossen werden. Zusammenfassend kann gesagt werden, dass DPS eine restriktive Darstellung eines aus vielen verteilten Einheiten bestehenden Systems und von seinen Annahmen her eine konsequente Weiterentwicklung eines zentralen Informationssystem-Ansatzes ist. Es ist als Konzept anwendbar für eine Domäne, in der das Verhalten der Teilnehmer (bzw. der sie repräsentierenden Softwareagenten) durch organisatorische Maßnahmen sanktioniert werden kann, nicht jedoch für ein offenes Netzwerk mit egoistischen Entitäten, die keine hierarchische Organisationsbeziehungen miteinander teilen.

3.3.2 Marktliche Koordination in Multi-Agenten-Systemen

Multi-Agenten-Systeme (MAS) im engeren Sinne gehen von einem komplementären Ansatz zum verteilten Problemlösen aus. DPS analysiert ein existierendes Problem und konstruiert daraufhin die handelnden Agenten. Bei MAS werden bereits existierende Agenten zusammengeführt, um ein neues Problem zu lösen:

„In contrast, research in MAS is concerned with the behaviour of a collection of possibly pre-existing autonomous agents aiming at solving a given problem“ (Jennings et al. 1998, S. 285).

Diese Problemlösungsentitäten (Agenten) sind autonom und können grundsätzlich voneinander verschieden sein. Charakteristisch für MAS ist (Jennings et al. 1998), dass jeder Agent nur unvollständige Information über den Zustand der Domäne und der anderen Agenten besitzt und auch nur unvollständige Fähigkeiten zur Lösung des Gesamtproblems beisteuern kann. Es

gibt keine globale Systemsteuerung und alle Daten liegen dezentralisiert vor. Alle Berechnungen werden in den Agenten parallel, gleichzeitig und asynchron durchgeführt, ohne dass eine Synchronisation z.B. durch einen Taktgeber stattfindet.

Außerdem wird die Annahme, dass die Agenten benevolent sind, fallengelassen:

„MAS research borrowed from the social science literature the underlying assumption that an agent should be rational: That, whatever it is doing, it should endeavor to maximize its own benefit/payoff.“ (Durfee/Rosenschein 1994, S.3)

Die Ableitung der Eigenschaften eines Agenten aus dem zu lösenden Problem, wie bei DPS, wird hier ersetzt durch die Ableitung der Eigenschaften aus den repräsentierten Individuen.

„That is, MAS generally only makes assumptions about the properties of individuals (most typically, that they are rational utility-maximizers), and considers what properties will emerge internally among agents given the incentives (payoffs) and features of their environment.“ (Durfee/Rosenschein 1994, S.6)

Der zentrale Designer, als weiteres Merkmal von DPS, kommt in MAS ebenfalls nicht vor:

“In particular, if a distributed problem-solving researcher can show that acting in a particular way is good for the system as a whole, he/she can impose this behavior on all the agent in the system at design time. For the multiagent system researcher, such an alternative is unavailable. At best, he/she might be able to design aspects of the environment that motivate all the (selfish) agents to act in a certain way. This need for indirect incentives is one element that distinguishes multi-agent system research from distributed problem-solving research” (Rosenschein/Zlotkin 1994b, S.32).

Rasmusson/Janson (1999, S.144) nennen 3 aufeinander aufbauende Argumente, warum Agenten in offenen Systemen immer eigennützig handeln werden und damit Multi-Agenten-Systeme formen:

1. Wenn Anwender ihre eigenen Softwareagenten kreieren können, werden sie ihr Eigeninteresse in das Verhalten der Agenten einprogrammieren.
2. Wenn Anwender zwischen eigennutzmaximierenden Agenten und solchen, die den Profit Dritter auf ihre Kosten erhöhen, wählen müssen, werden sie eigennutzmaximierende Agenten wählen.
3. Wenn Anwender nur solche Agenten benutzen dürfen, die das Ergebnis anderer Anwender auf ihre eigenen Kosten erhöhen, werden Agenten nur für solche Aufgaben benutzt werden, bei denen der Anwender ohne Agentennutzung noch schlechter dran wäre.

Was passiert, wenn man nicht-egoistische Agenten in einen eigennutzmaximierenden Verhandlungskontext setzt und die Koordination statt über ökonomische Mechanismen (Kompensation und Konzessionierung) über Argumentation führt? Rasmusson/Janson (1999, S. 145) führen das Beispiel zweier Software-Agenten an, die jeweils für ihren Anwender einen Hotel-aufenthalt buchen sollen. Es ist jedoch nur noch ein Zimmer frei. Sollte der Agent, der das Zimmer zufällig als erster belegt hat, es dem anderen Agenten geben, nur weil dieser ihn darum bittet? Die Agenten des MAS können nicht direkt angewiesen werden, eine bestimmte Aktion auszuführen, um das Erreichen des Gesamtzieles zu unterstützen. Es gibt keinen Gradmesser, der zum Vergleich des Nutzens beider Agenten herangezogen werden kann, weder global noch bilateral. In diesem Fall gibt es keine Möglichkeit, einen in irgendeiner Weise fairen Ausgleich durchzuführen. Die Autonomie der Agenten, die sie von Objekten unterscheidet, verbietet diese Vorgehensweise. Hingegen benötigt man hier eine Implementierung bi- oder multilateraler Koordination durch Aushandlungsprozesse, in denen Angebote formuliert und Konzessionen eingegangen werden. Die Einführung egoistischer Agenten führt zu einer Notwendigkeit ökonomischer Kompensation, die frei aushandelbar ist und letztendlich zu einer pareto-effizienten Allokation führen muß. In diesem Sinne muß Koordination in offenen Systemen als Konzessionierung bzw. mit Hilfe von Kompensationsmechanismen implementiert werden.

Eine fundamentale Fragestellung der MAS ist die Verknüpfung der Zielverfolgung der Agenten mit dem zu lösenden Gesamtproblem und damit die Frage, wie man individualistisch handelnde Agenten dazu bringt, mit anderen Agenten zu kooperieren:

*„MAS studies how individual, self-interested decisionmakers might discover (or be coerced into) stable, predictable, and desirable ways if interacting among themselves“
(Durfee/Rosenschein 1994, S.6).*

In Systemen, deren Agenten unterschiedlichen Besitzern gehören, ist eine argumentative Einigung nicht anwendbar, da keine globale Nutzenfunktion existiert. Aufgabe der Agenten ist es hier, die Wichtigkeit des jeweiligen Zieles durch Signalisierung gegenüber anderen Agenten offenzulegen. Diese Aufgabe übernehmen Kauf- und Verkaufsangebote, die unter Beachtung von Budgetrestriktionen Konflikte um Ressourcen lösen. Die von den Agenten in ihren Geboten übermittelten Informationen müssen formalisierbar und diskret sein. Die Formalisierbarkeit des Einzelnutzens kann in realistischen Szenarien weder in der Realität noch in agentenbasierten Marktsystemen als gelöst vorausgesetzt werden. Beispielsweise hängt der Nutzen eines Händlers vom Wiederverkaufswert des verhandelten Gutes ab. Dieser wird in der Zukunft bestimmt und hängt seinerseits ab von der Nutzenfunktion eines im vorhinein unbekanntem Verkäufers. Dem Händler kann sein Nutzen bestenfalls als Schätzwert innerhalb einer

Bandbreite bekannt sein, was die Entscheidung zwischen Handlungsalternativen mit Ungewißheit behaftet und möglicherweise keine eindeutig beste Alternative berechnen läßt.

Der Schwerpunkt der MAS liegt auf der Interaktion vieler Agenten mit einer sich dynamisch ändernden und äußerst flexiblen Umgebung, die zu einem großen Teil aus anderen autonomen Agenten besteht. Zusammenfassend kann gesagt werden:

„MAS corresponds to a research agenda that has focused on getting certain internal properties in a system of agents whose individual properties can vary. Thus, MAS has been concerned with how agents with individual preferences will interact in particular environments such that each will consent to act in a way that leads to desired global properties. MAS asks how, for a particular environment, can certain collective system properties be realized if the properties of agents can vary uncontrollable.“ (Durfee/Rosenschein 1994, S.8).

Ein wesentlicher Bestandteil der Forschung ist die Emergenz von Verhalten oder Funktionalität im Gesamtsystem.

„Emergence offers a bridge between the necessity of complex and adaptive behavior at a macro level (the one of an observer) and the mechanisms of multiple situation-based actions at a micro level (the one of the agents)“ (Chantemargue/Hirsbrunner 1999, S. 1).

Das Verhalten eines Systems kann als emergent bezeichnet werden, wenn es nur durch deskriptive Kategorisierung beschrieben werden kann, die nicht notwendig ist, um das Verhalten der konstituierenden Komponenten zu beschreiben (vgl. Forrest 1990). Aus emergentem Verhalten folgt emergente Funktionalität, wenn das Verhalten zur Stabilisierung des Systems beiträgt und das System sich darauf aufbauend weiterentwickeln kann (vgl. Steels/Brooks 1994).

Um die eigennützigen Ziele der Agenten unter einen Hut zu bringen, muß eine Koordination durch Verhandlungsprozesse stattfinden. Dafür wird angenommen, dass die Agenten an einer Teilnahme am Marktmechanismus interessiert sind, d.h. dass sie durch Nichtteilnahme schlechtergestellt wären als durch das Eingehen von Konzessionen im Rahmen dieser Verhandlungsprozesse und daher führt der Wunsch des Einzelnen nach Verbesserung der eigenen Position zur Teilnahme an der marktlichen Koordination. Multi-Agenten-Systeme entsprechen in ihren grundlegenden Konzepten durchaus ökonomisch orientierten Domänen mit vielen, lateral kooperierenden Wirtschaftssubjekten. Beispiele für solche Domänen, in denen Agenten versuchen, ihren eigenen Nutzen zu maximieren, sind (Kraus 1997, S.82) elektronische Märkte, bei dem die Agenten unterschiedliche Unternehmen repräsentieren und in deren

Namen kaufen und verkaufen, (z.B. Chavez/Maes 1996, Fischer et al. 1996, Tsvetovaty/Gini 1996, Eymann et al. 1998a). Wegen seiner realen Implikationen interessant ist ein Luftverkehrsszenario, in dem Flugzeuge unterschiedlichen Fluggesellschaften gehören, sich mit anderen Flugzeugen die begrenzten Ressourcen des gleichen Flughafens teilen und ein Mechanismus gesucht wird, der denjenigen Flugzeugen mit dem wenigsten Treibstoff an Bord Priorität gibt (Ljunberg/Lucas 1992).

Im Sinne des Schichtenmodells der Telematik werden in den folgenden beiden Abschnitten die notwendigen Kommunikationskonzepte (3.4) und Koordinationskonzepte (3.5) dargestellt, die zur Erreichung der Koordination eines MAS in den Agenten implementiert werden müssen.

3.4 Kommunikation zwischen Agenten

Agenten besitzen Interaktionsfähigkeiten, um mit anderen Agenten Informationen austauschen zu können. Kommunikation ist eine notwendige Fähigkeit, um Koordination zu erreichen. Da der Unterschied zwischen Objekten und Agenten vor allem in der autonomen Steuerung des eigenen Verhaltens liegt, sind Methodenaufrufe (Kommunikation durch Übergabe von Variablenwerten), bei denen kooperatives Verhalten durch ein anderes Objekt erzwungen wird, bei Agenten praktisch ausgeschlossen²⁵. Als Alternative bieten sich nachrichtenbasierte Verfahren an, die Informationen entweder durch direkte Kommunikation, über spezielle Facilitator Agenten, oder über Broadcasting an andere Agenten weitergeben. Ein datenbankbasiertes Verfahren ist das Blackboard-Prinzip, bei dem Information in eine zentrale Datenbank hineingeschrieben und aus ihr gelesen wird.

Broadcast Architekturen (Foner 1999) übertragen sämtliche Nachrichten an alle Agenten, wodurch eine hohe Flexibilität erreicht wird. Agenten können in eine solche Umgebung eintreten und diese wieder verlassen, ohne sich auf der Anwendungsebene ein- oder austragen zu müssen. Trotzdem kann davon ausgegangen werden, dass sämtliche Informationen ankommen, eine fehlerfreie technische Übertragung vorausgesetzt. Bei allen Agenten wird damit ein gleicher Wissensstand erreicht, so dass eine homogene Entscheidungsgrundlage für deren Aktionen existiert. Diese Eigenschaft macht Broadcasting für solche Anwendungsdomänen notwendig, bei denen die Verfolgung eines Gesamtzieles im Vordergrund steht.

25 Im später beschriebenen Multi-Agenten-System AVALANCHE wird jedoch genau über solche Methodenaufrufe kommuniziert. Dieser Widerspruch löst sich dadurch auf, dass dabei lediglich eine Nachricht als String übergeben wird, auf deren Verarbeitung der aufrufende Agent keinen Einfluß hat. Ein direkter Methodenaufruf (z.B. `berechneAngebotspreis()`), der die Autonomie des Agenten verletzt, findet jedoch auf keinen Fall statt.

Blackboardsysteme (Corkill 1991) bestehen aus einer Datenbank (*Repository*), in die Agenten Nachrichten schreiben, erzielte Ergebnisse speichern und Information auslesen. Die Agenten lesen entweder (1) das Blackboard in kurzen Zeitabständen aus, um neue Nachrichten zu erfahren, (2) das Blackboard schickt diese weiter (Subscriber-Modell), oder (3) spezielle Filteragenten reagieren auf neue Nachrichten (sog. generative Kommunikation) und senden Benachrichtigungen.

In vielen Implementationen, bei denen Agenten auf unterschiedlichen Abstraktionsstufen arbeiten, ist das Blackboard in derartige Ebenen unterteilt. Dadurch können Ergebnisse aus einer untergeordneten Ebene durch das Blackboard synthetisiert und auf der höheren Ebene dargestellt werden, ebenso wie Ziele höherer Ebenen heruntergebrochen werden, um niedrigere Agenten anzuleiten. Es ist nicht verwunderlich, dass Blackboard-Architekturen eng mit DPS-Konzepten verknüpft sind. In Domänen, in denen es wichtig ist, dass alle Agenten unbedingt gleichzeitig die gleichen Informationen besitzen (wie beim Broadcast) und gleichzeitig eine allgemeingültige Kategorisierung der Informationen stattfindet, ist ein Blackboard unverzichtbar. Die Agenten können so ihre Aktionen implizit aufeinander abstimmen und damit die Verfolgung eines Gesamtzieles sicherstellen. Anwendungsdomänen, bei denen eine Hierarchie vorgegeben ist und ohne weitere Koordination schnell gehandelt werden muß, finden sich z.B. im militärischen Bereich, im Rettungsdienst und in der operativen Fertigungssteuerung.

In Anwendungsdomänen wie z.B. elektronischen Märkten, in denen es weder eine vordefinierte Hierarchie zwischen den Agenten noch ein von allen geteiltes Gesamtziel gibt, ist eine derartige öffentliche Kommunikation nicht erwünscht. Eine Blackboard-Architektur auf einem elektronischen Markt entspräche einer Katalog- oder Auktions-Website, in der für alle Nachfrager undifferenziert gleiche Preise festgelegt wären. Die Stärken eines agentenbasierten Ansatzes, bei dem bilateral zwischen den Agenten individuelle Preise ausgehandelt werden können, werden durch ein Blackboard negiert. Die Veröffentlichung sämtlicher Informationen mittels Broadcast verletzt die Privatheit der Verhandlung zwischen zwei Agenten. Hier bietet sich eher die direkte Kommunikation oder die Nutzung eines Facilitator-Ansatzes an.

Bei der **direkten Kommunikation** instantiiieren die Agenten Kommunikationskanäle zwischen sich und nutzen dafür die Dienste des Computernetzwerks. Um diese Kanäle aufzubauen, benötigen sie die direkten Netzwerkadressen anderer Agenten. Diese können entweder als Broadcast an alle Agenten verteilt werden, wie beim AgentNameServer in *JATLite* (Jeon et al. 2000), oder es gibt ein zentrales Objekt (Agent Directory, FIPA 1997), welches diese Adressen auf Anfrage mitteilt. Die Agenten müssen sich dafür selbstständig in diesem Register eintragen. Direkte Kommunikation ist vor allem dann sinnvoll, wenn die Agenten nicht mobil sind, oder Informationen immer nur zwischen wenigen Agenten übertragen werden

müssen. Als Nachteil der direkten Kommunikation wird die mangelnde Skalierbarkeit angeführt, d.h. der Verwaltungsaufwand steigt überproportional zur Menge der Agenten in dem System.

Sogenannte **Federated Systems** (Genesereth/Ketchpel 1994) lösen dieses Problem, indem sie neben den eigentlichen Problemlösungsagenten spezielle Facilitatoren (oder Mediatoren) einführen. Eine Gruppe von Problemlösern sowie ein einzelner Facilitator bilden eine sogenannte Federation, in der nur der Facilitator nach außen hin mit anderen Gruppen kommuniziert. Die Problemlöser führen die Aufgaben durch, die ihnen vom Facilitator zugewiesen werden und geben einen Teil ihrer Autonomie auf. Im FIPA Standard (1997) ist ein „Domain Facilitator“ vorgesehen, der jeweils für eine Domain (im technischen Sinne, z.B. ein lokaler Host) neben der Kommunikation nach außen auch das Agent Directory führt und damit die innere Kommunikation zwischen den Problemlösern unterstützt.

Im Kontext der elektronischen Märkte können sowohl direkte Kommunikation als auch ein Facilitator-Ansatz Anwendung finden. Letzter ist allerdings nur sinnvoll, wenn zum einen der Facilitator selbst als vertrauenswürdige Partei eingestuft werden kann, d.h. wenn er nicht die kommunizierten Informationen von Agenten zu deren Schaden ausnutzt. Dies ist technisch ohne weiteres möglich und erfordert eine organisatorische Festlegung, bevor ein Agent den Markt betritt. Zum anderen hat der Facilitator auch die Möglichkeit, Nachrichten zu interpretieren und nach seiner Entscheidung Transaktionspartner miteinander in Kontakt zu bringen. Diese Fähigkeit wird in physischen Märkten als „business matching“ bezeichnet und stellt eine Domäne von Handelsintermediären dar. Da in der vorliegenden Arbeit Dezentralität in der Koordination, sowie Privatheit und Autonomie der Agenten im Vordergrund steht, wird konsequenterweise in der Kommunikationsebene auf die teilweise zentrale Lösung mit einem Facilitator verzichtet.

Kommunikation mit einem anderen Agenten bedeutet Interaktion. Im Umfeld der Agententechnologie wird Interaktion definiert als kollektive Aktion, bei der ein Agent eine Aktion ausführt oder eine Entscheidung trifft, die durch die Anwesenheit oder das Wissen eines anderen Agenten beeinflusst wird (vgl. Gasser 1998). In der Regel geht es um die Interpretation der Semantik der durch die Kommunikationsebene übermittelten Informationen. Für die Kooperation in einem System verteilter Agenten wird eine Menge an Nachrichten (eine Sprache) benötigt, die agenten-unabhängige Semantik implementiert. In der Agententechnologie werden für diesen Zweck zwei alternative Implementierungen der Sprechakttheorie (Searle 1997) eingesetzt, die „Knowledge Query and Manipulation Language“ (KQML) und die „Agent Communication Language“ (ACL). In der Sprechakttheorie stellen die ausgetauschten Nachrichten

nicht nur Informationen dar, sondern sind gleichzeitig auch Aktionen des Senders gegenüber dem Empfänger.

KQML (Labrou/Finin 1997) definiert ein Protokoll zum Austausch von Informationen (Kommunikationsebene) und eine Sprache zum Austausch von Wissen (Kooperationsebene). KQML spezifiziert sowohl das Nachrichtenformat als auch das Message-Handling Protokoll, um den Wissensaustausch (Knowledge Sharing) zwischen Agenten in Echtzeit zu unterstützen. Es kann als Sprache benutzt werden, damit ein Anwendungsprogramm mit einem intelligenten System interagiere, oder zwei oder mehr intelligente Systeme Wissen austauschen können, um eine kooperative Problemlösung zu erzielen. KQML baut auf einer erweiterbaren Menge von codierten Performativen auf, die erlaubte Operationen definieren, mit denen die Agenten auf das gespeicherte Wissen und die Ziele der jeweils anderen Agenten zugreifen können. Performative sind Sprechakte, die von einem Agenten auf einen anderen Agenten hin ausgeübt werden und bereits eine Aktion darstellen, im Unterschied zu sogenannten Konstativen, die lediglich wahre oder falsche Informationen darstellen:

*„By using them in utterances the speaker not only **says** something, but also **does** something, the world changes by saying the utterance with this kind of verb, e. g. I christen this ship Queen Mary“ (Levinson 1993, S.241)*

Diese Performative stellen ein Substrat dar, auf dessen Basis höher entwickelte Modelle der Agenteninteraktion aufgebaut werden können, wie z.B. Kontraktnetze und Auktionsverfahren. Zusätzlich stellt KQML eine fundamentale Architektur zur Wissensvermittlung (knowledge sharing) bereit, indem eine spezielle Klasse von Agenten, die “communication facilitators” existiert, die die Interaktionen der anderen Agenten koordinieren. Die Konzepte, die dem Design von KQML zugrundeliegen, werden durch experimentelle Prototypen in Anwendungsdomänen wie „Concurrent Engineering“, „Intelligent Design“ und „Intelligent Planning and Scheduling“ erforscht.

Auch FIPA’s Agent Communication Language (**ACL**) (FIPA 1997) basiert auf der Sprechaktheorie und definiert eine Menge von Performativen, mit denen Agenten interagieren und Information austauschen können. Der Mechanismus des Ausübens eines Sprechaktes entspricht dem Senden einer Nachricht, in der dieser Sprechakt codiert ist. Zusammen mit dem Wissen über die Anwendungsdomäne ermöglicht der Sprechakt dem Empfänger, die Bedeutung des Nachrichteninhalts zu erfassen. Die Rollen des Initiators und des Empfängers des Sprechakts sind gleichzusetzen mit dem Sender und dem Empfänger der Nachricht. In der ACL wird ein Kern an Performativen definiert, auf dessen Basis verschiedene Sprechaktgruppen aufbauen, die zwischen Generalisierung, Ausdrucksstärke und Einfachheit balancieren.

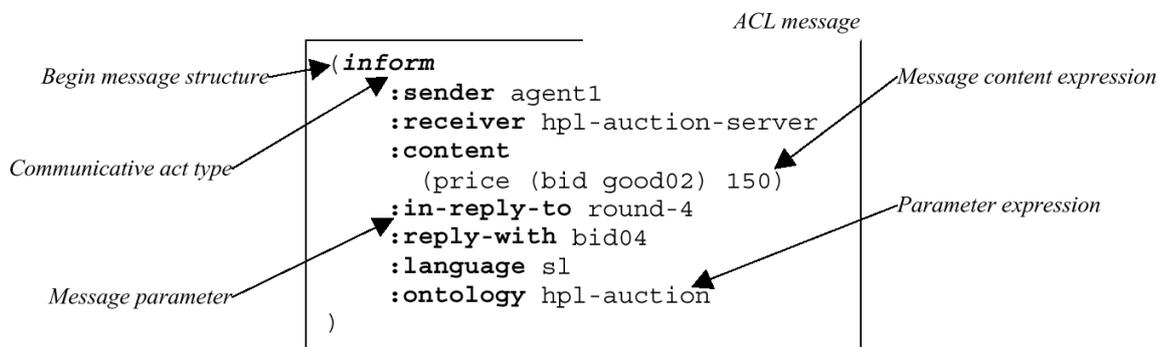


Abb. 13: Aufbau einer ACL Nachricht (aus FIPA 1997, S. 11)

Das Performativ läßt Rückschlüsse auf den Zustand des internen Modells des Senderagenten und die vom Sender erwarteten Konsequenzen auf den Zustand des internen Modells des Empfängeragenten zu. Da beide Agenten autonom sind, gibt es keine Garantie, dass die erwarteten Konsequenzen tatsächlich eintreffen. Ein Beispiel (FIPA 1997) im Sinne des BDI-Konzepts ist der Glaube von Agent *agent1*, dass sich beide Agenten besser stellen, wenn das Gut *good02* für 150 Geldeinheiten den Besitzer wechselt und seine Intention, dass Agent *agent2* das auch glauben sollte (siehe Abb. 13). Agent *agent1* wird *agent2* von seinem Glauben an die Wahrheit dieser Proposition informieren (durch den in der ACL definierten Sprechakt “inform”). Agent *agent2* wird, durch die Interpretation der Semantik der “inform”-Nachricht, von Agent *agent1*’s Zustand bezüglich dieser Proposition wissen. Ob jedoch *agent2* diesen Glauben übernimmt, ist alleine *agent2*’s Entscheidung. In diesem Sinne wird sämtliche Inter-Agenten-Kommunikation durch Message Passing durchgeführt.

Die Behandlung eines Sprechaktes innerhalb eines Dialogs sollte der anderer Aktionen entsprechen; ein Sprechakt ist nur eine derjenigen Aktionen, die ein Agent durchführen kann und zwischen deren Ausführung er sich u.U. entscheiden muss. Der Begriff der Nachricht (Message) wird in der ACL Spezifikation auf zwei Arten verwendet, die im Schichtenmodell der Telematik den Konzepten der technischen Übertragung und der nicht-technischen Kommunikationsschicht entsprechen. Auf der ersteren Ebene bezeichnet eine Nachricht die Datenstruktur, die vom Nachrichtenübertragungsdienst gewählt wird, um Daten von einem Agenten zum anderen zu befördern. Auf der nicht-technischen Schicht ist eine Nachricht ein Synonym für einen Sprechakt.

In diesem Abschnitt wurden die unterschiedlichen Möglichkeiten zur Implementierung von Kommunikation zwischen Agenten vorgestellt. Das hier verfolgte Ziel einer dezentralen marktlichen Koordination kann am besten durch eine ebenso dezentrale direkte Kommunika-

tion der Agenten unterstützt werden. Die kommunizierten Nachrichten selbst werden wie bei anderen Agentenprojekten durch Implementationskonzepte der Sprechakttheorie dargestellt. Hierbei sind die beiden Varianten KQML oder ACL gegenseitig austauschbar. Aufgrund der besseren Dokumentation und aktiven Weiterentwicklung von ACL wird der MAS-Prototyp in Kapitel 4 aber mittels dieser Variante implementiert.

3.5 Automatisiertes Verhandeln zwischen Agenten

In einem elektronischen Markt interagieren zwei oder mehr Software-Agenten, indem über das Austauschen von Kommunikationsakten eine Verhandlung durchgeführt wird. Verhandlungen zwischen Agenten können je nach Implementation auf zwei grundsätzlich unterschiedliche Arten durchgeführt werden (Pruitt 1981, S. 91):

- *“Concession exchange, in which the parties move toward one another on a single dimension or swap concessions on different dimensions.*
- *Problem-solving discussions, in which the parties share information about goals and priorities in search of an option that will satisfy both parties needs, that is, an integrative agreement.”*

Der grundlegende Unterschied zwischen beiden Alternativen besteht in der Handhabung der Verhandlungsdimensionen. Im Allgemeinen zeichnen sich Marktsituationen durch das Vorhandensein einer gemeinsamen Nutzenwertskala (Geld) aus. Damit besteht eine Dimension, um (Preis-)Konzessionen anzubieten und zu interpretieren. In einer Marktsituation muß grundsätzlich davon ausgegangen werden, dass der Abbruch eines Verhandlungsvorganges ein Ziel eines der Verhandlungspartner sein kann und dass die Verhandlung nicht notwendigerweise zu einem Ergebnis führen muß.

Der Einsatz von Problemlösungsdiskussion bietet sich hingegen an, wenn die Kontrahenten sich erst über solche Dimensionen einigen müssen. Dabei wird von einer Kooperativität beider Parteien in Bezug auf die Erreichung eines Zieles ausgegangen. Die Verhandlung hat eher den Zweck, aus einer Anzahl alternativer Pfade den besten zu finden und sich auf diese Lösung gemeinsam zu einigen. Ohne das Grundverständnis, dass die Beteiligten an einer Lösung interessiert sind, bleibt aber nur der Konzessionsaustausch als Mechanismus übrig.

Aufgrund der in dieser Arbeit getroffenen Einschränkung auf Marktsituationen kann das einfachere Konzessionsverfahren als ausreichend angesehen werden. In diesem Fall ist das Ziel der Verhandlung, über eine schrittweise Annäherung an die gegnerischen Preisvorstellungen ein gemeinsames Ergebnis zu erzielen, üblicherweise die Durchführung einer Kauf- bzw. Ver-

kaufstransaktion zu einem für beide Seiten akzeptablen Preis. Der einzelne Agent benötigt für diese Aufgabe (Preist 1998, Kraus 1998):

1. eine Definition der Güter oder Dienste, die er handeln soll,
2. eine quantitative und qualitative Definition des Zieles, welches sein Prinzipal aus der Aushandlung zu erreichen wünscht und
3. eine Verhandlungsstrategie, die mindestens genauso effektiv ist wie diejenige einer qualifizierten Menschen in der gleichen Situation.

Welche Alternativen es für die Ausgestaltung dieser Eigenschaften gibt, wird in den folgenden Unterabschnitten vorgestellt.

3.5.1 Die Handelsgüter

De Paula et al. (2000) unterscheiden bei der Spezifikation der zu handelnden Güter zwischen den Produktattributen und den Verhandlungsattributen (deal attributes). Die Produktattribute sind nicht verhandelbar und beschreiben das Gut. Die Verhandlungsattribute sind für die aktuelle Verhandlung spezifisch und umfassen jene Variablen, bei denen von den Verhandlungspartner überhaupt Konzessionen eingegangen werden können.

Die ersten Aushandlungen, die von automatisierten Agenten getätigt werden, werden für Wirtschaftsgüter durchgeführt, bei denen der Preis das einzige Verhandlungsattribut ist, z.B. bestimmte Rohöl-, Elektrizität- und Weizenarten (Sierra 2000, S. 10). Bei zunehmender Komplexität ist es wichtig, eine formale Repräsentation zu entwickeln, die ein Produkt spezifiziert und die es auch von ähnlichen Produkten anderer Hersteller differenziert. Ein Käufer wird möglicherweise für ein Produkt höherer Qualität oder für Zusatzleistungen bereit sein, mehr zu bezahlen. Solche Abwägungen werden bei Menschen häufig aus dem Bauch heraus getroffen. Es ist sehr schwierig, die exakten Kriterien hinter solchen Entscheidungen zu benennen, um einen automatisierten Agenten für sich arbeiten lassen zu können. Die Spezifikation der Güter hängt jedoch eng mit den Spezifikationsmöglichkeiten des Zieles in den Agenten zusammen.

Falls sich sämtliche Attribute des Gutes durch den Preis kodieren lassen, spricht man von einem Wirtschaftsgut (*commodity*). Entweder ist das Gut so homogen, dass ausser dem Preis keine Unterscheidung zwischen zwei unterschiedlichen Angeboten gemacht werden kann, oder andere Attribute können in den Preis einberechnet werden.

Falls das Gut neben dem Preis weitere Attribute aufweist, die für den Verhandlungsprozeß wichtig sind, müssen diese ebenfalls benannt und durch die Verhandlungspartner in ihrer Wichtigkeit bewertet werden können. Ein Beispiel, das beide Fälle abdecken kann, ist die

Bestellung eines Produktes mit einer unterschiedlichen Lieferzeit, z.B. per Flugzeug oder per Schiff. Falls die Lieferzeit in der Verhandlung überhaupt keine Rolle spielt, lassen sich die Güter alleine nach der Summe aus Produktpreis und Transportpreis bewerten. Falls der Lieferzeit eine Bedeutung durch den Käufer zugemessen wird, muß dieser die beiden Attribute in ihrer Wichtigkeit bewerten und nötigenfalls jenes Produkt erwerben, welches zwar teurer ist, aber schneller geliefert werden kann.

3.5.2 Das Ziel der Agenten

Die Frage nach der Spezifikation des Zieles lässt sich unterteilen nach der Frage, welche Ziele durch den Agenten verfolgt werden und wie die Zielerreichung gemessen werden kann.

Welche Ziele durch die einzelnen Agenten verfolgt werden, entscheidet in zentralen Systemen üblicherweise der Systemdesigner. In offenen Netzen ist diese Möglichkeit nicht gegeben, im Gegensatz muß davon ausgegangen werden, dass jeder Beteiligte ein unterschiedliches Ziel verfolgt:

“The agents that we are interested in looking at are heterogeneous, self-motivated agents. [...] In particular, the personal assistants do not act benevolently unless it’s in their interest to do so.” (Rosenschein/Zlotkin 1994b, S. 30).

Grundlegende Annahme für die Notwendigkeit von Verhandlungen sind die Heterogenität und individualistische Zielverfolgung der einzelnen Beteiligten an einem Gesamtsystem. Diese Annahmen sind der Normalfall in nicht-technischen Systemen und der Ausgangspunkt für jede Behandlung der Koordinationsproblematik.

In der Informatik ist erst mit der Möglichkeit der Kommunikation und Kollaboration zwischen Informationssystemen die Frage aufgekommen, wie geeignete Systemarchitekturen aussehen müssen, in denen sich die individualistischen Teilnehmer zur Verfolgung eines gemeinsamen Ziels koordinieren:

“The question is, how will these agents be cooperating with each other, competing with each other, and negotiating with each other?” (Rosenschein/Zlotkin 1994b, S. 29).

Einen ersten Schritt bietet die Betrachtung existierender Ansätze zur Koordination in sozialwissenschaftlichen und naturwissenschaftlichen Bereichen. Das Ziel, formalisierte Gestaltungshinweise für automatisierte Verhandlungen abzuleiten, wird von Rosenschein/Zlotkin konsequenterweise verstanden *“[...] as a kind of social engineering for communities of machines” (Rosenschein/Zlotkin 1994b, S. 30).*

Gibt es ein systemweites Ziel des Mehragentensystems, dessen Erreichung bei der Gestaltung formalisierter Verhandlungsprotokolle im Vordergrund steht? Rosenschein/Zlotkin (1994b, S.31) führen an, dass als *“acceptable doctrine of general goodness”* das Pareto-Optimum betrachtet werden kann, d.h. dass nach dem Erreichen eines bestimmten Verhandlungsergebnisses kein Agent sich verbessern kann, ohne dass ein anderer Agent schlechter gestellt wird. Ein Ergebnis X ist Pareto-Optimal, sofern die beteiligten Agenten kein anderes Ergebnis Y finden können (welches sie durch gemeinsame Anstrengungen erreichen können), so dass jeder Agent in Y einen höheren Nutzen hat als in X oder wenigstens in Y niemand schlechter dasteht und gleichzeitig einer besser dasteht. In diesem Fall ist Ergebnis Y Pareto-überlegen gegenüber Ergebnis X. Falls ein Ergebnis X also nicht Pareto-optimal ist, können die beteiligten Agenten aus sich heraus zu einer anderen Verteilung Y wechseln, d.h. ihre Kooperationsprozesse so lange fortsetzen, bis dieses Ergebnis Y erreicht ist. Diese Betrachtungsweise wird auch von Sandholm (1994, S. 13ff.) aufgegriffen, der neben dem Pareto-Optimum noch weitere Kriterien zur Auswahl stellt. Aus der Spieltheorie z.B. kann das Nash-Gleichgewicht herangezogen werden, welches auftritt, wenn kein Agent einen Anreiz hat, von seiner einmal gewählten Strategie abzuweichen.

Welcher Maßstab für die Erfolgsmessung der Zielverfolgung herangezogen wird, ergibt sich größtenteils aus der Beschreibung der Güter. Beim Handeln mit Wirtschaftsgütern steht von vornherein die Berechnung in Geldeinheiten fest. Ein Pareto-Optimum ergibt sich als Zustand, in dem kein Agent seinen monetären Gewinn erhöhen kann, ohne einem anderen Agenten dabei Geld wegzunehmen. Der Nutzen (engl. utility) eines Agenten kann der Gewinn pro Transaktion sein, der Gewinn über einen bestimmten Zeitraum, der Umsatz, oder andere Werte. Als ökonomisch handelnde Subjekte können im Prinzip sämtliche Erfolgskriterien herangezogen werden, die auch in der Betriebswirtschaftslehre gelten. Sollen mehrere Nutzenattribute in unterschiedlicher Gewichtung verfolgt werden, müssen Ansätze zur quantitativen oder qualitativen Analyse von Entscheidungsproblemen herangezogen werden.

Multi-Objective Decision Analysis (Guttman/Maes 1998, S. 8ff., Keeney/Raiffa 1976) untersucht die quantitative Analyse von Entscheidungen, die mehrere, voneinander abhängige Ziele verfolgen und von einem einzelnen Entscheider getroffen werden. Es wird zwischen der Untersuchung von Unsicherheiten und der Präferenzanalyse unterschieden. Das übliche Vorgehen der *Multi-Attribute Utility Theory* (MAUT) erstellt eine Nutzenfunktion, indem die Attribute einer Entscheidungssituation relativ zueinander gewichtet werden. In einer Marktumgebung wird dieses Vorgehen im einfachsten Fall zu einem Abtragen von Güterpreis gegenüber Lieferzeit führen. Eine qualitative Analyse von Multi-Attribut-Entscheidungen kann als *Constraint Satisfaction Problem* (CSP) (Tsang 1993) dargestellt werden. In CSP werden Variablen, deren Domänen und eine Menge von Einschränkungen (*constraints*) definiert, welche

Beziehungen zwischen den Variablen definieren und die Werte einschränken, die diesen Variablen gleichzeitig zugewiesen werden können. Die Aufgabe einer sogenannten CSP-Engine ist es dann, jeder Variablen einen Wert zuzuweisen und gleichzeitig alle Einschränkungen zu befriedigen. Es kann noch zwischen solchen Einschränkungen unterschieden werden, die unbedingt erfüllt werden müssen (*hard constraints*, z.B. „das Produkt darf nicht teurer sein als € 2000“) und solchen, die eine Wichtigkeit ausdrücken, die erfüllt werden kann, aber nicht muß (*soft constraints*, z.B. „Lieferzeit ist mir wichtiger als Preis“). Dadurch kann eine Vielzahl von Entscheidungsproblemen als CSP artikuliert werden. Eine Variante ist *Distributed CSP* (DCSP), bei dem Variablen und Einschränkungen über mehrere Agenten verteilt sind (vgl. Yookoo/Durfee 1992). Diese Technik kann z.B. in einer Verhandlungssituation benutzt werden, um die Bedürfnisse sowohl von Anbieter wie auch von Konsument in einer integrativen Verhandlung gleichzeitig zu befriedigen.

Für relativ einfache Märkte, wie in der vorliegenden Arbeit angenommen, werden derart komplexe Verfahren nicht benötigt. Die Einschränkung der gehandelten Wirtschaftsgüter auf *Commodities* lässt als Ziel der Agenten jeweils die in Geldeinheiten gemessene Nutzenmaximierung zu, d.h. die größtmögliche Differenz zwischen Indifferenz- und Transaktionspreis (s.u.) bei gleichzeitiger Maximierung der Transaktionsfrequenz. Wie in der Betriebswirtschaft üblich bedeutet dies ein Abwägen zwischen Gewinn und Umsatz, für das es keine allgemeine Lösung gibt, sondern von jedem Agenten (und dessen Prinzipal) anders formuliert werden wird. Mit zunehmender Komplexität der Märkte wird weiterhin die Einbeziehung von Verfahren der MAUT oder CSP bei der Konstruktion von handelnden Agenten notwendig werden. Funktionsweise und Ergebnis des marktlichen Koordinationsmechanismus bleiben von diesen Überlegungen unberührt, so dass mit der einfachstmöglichen Zieldefinition begonnen werden kann.

3.5.3 Ansätze für Verhandlungsstrategien

Neben der Definition der Güter und der Ziele der Verhandlung bedarf es drittens der Festlegung einer Vorgehensweise, um das Ziel der Verhandlung zu erreichen, d.h. einer Verhandlungsstrategie. Die Ausgangssituation stellt sich grafisch wie in der eindimensionalen Abbildung 14 gezeigt dar.

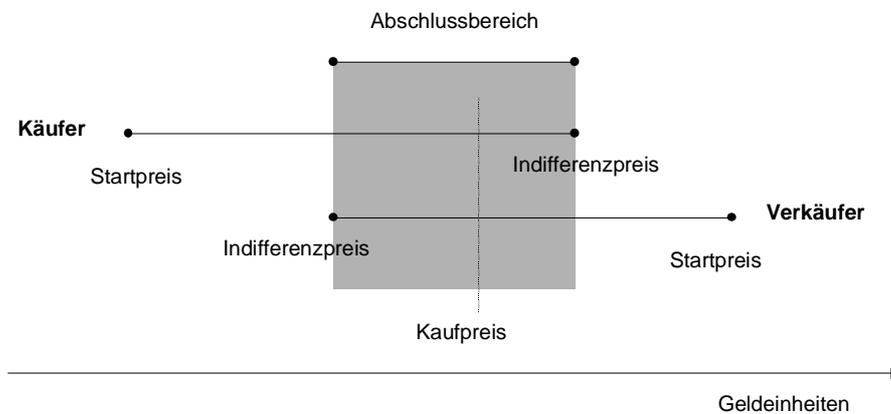


Abb. 14: Bilateraler Verhandlungsprozeß (nach de Paula et al. 2000, S. 3)

Die unterschiedlichen Punkte stellen die in einer Verhandlung wesentlichen und in Geldeinheiten bewertbaren Meilensteine dar. Im Allgemeinen gibt jeder Prinzipal seinem Agenten einen internen Indifferenzpreis oder Limitpreis vor, der seiner eigenen Wertschätzung des Gutes entspricht. Der Indifferenzpreis stellt für den kaufenden Agenten ein Maximum dar, bis zu dem er bieten würde. Der Nutzen, den der Agent aus dem Kauf zieht, ist in der Darstellung proportional zu der Differenz zwischen dem Kaufpreis und dem Indifferenzpreis²⁶. Analog dazu gibt es einen Indifferenzpreis als Minimum, unter dem der verkaufende Agent keinen Nutzen aus der Transaktion ziehen würde. In jeder Verhandlungssituation gibt es ausserdem einen Startpreis, mit dem beide Agenten die Verhandlung beginnen. Im Zeitablauf einer Verhandlung beginnen Käufer und Verkäufer außen an ihrem Startpreis und bewegen sich durch Eingehen unilateraler Konzessionen auf die Mitte zu. Der grau unterlegte Bereich in der Mitte zwischen beiden Indifferenzpreisen wird als Abschlußbereich (engl. „deal range“) bezeichnet. Falls dieser Bereich leer ist, d.h. sich die Grenzen, innerhalb derer eine Transaktion möglich ist, nicht überschneiden, so kann keine Transaktion stattfinden. Um von den Startpreisen im Zeitablauf den Abschlußbereich zu erreichen, können unterschiedliche Verhandlungsverfahren eingesetzt werden. Grundsätzlich sind folgende Vorgehensweisen möglich:

1. Anbieter und/oder Nachfrager nennen jeweils ihren Startpreis, ohne anschließend Konzessionen einzugehen, d.h. sich preislich aufeinander zuzubewegen. Ein solches Vorgehen kann als „friß-oder-stirb“-Strategie bezeichnet werden. Diese Taktik geht auf, wenn der Startpreis

²⁶ Eine Grenznutzenbetrachtung, bei der zwei Agenten bei gleichem Geldgewinn unterschiedliche Nutzen erreichen, wird hier und im folgenden unterlassen. Aus Vereinfachungsgründen in der Darstellung ist der Nutzen unabhängig vom jeweiligen Agent immer linear proportional zum gewonnenen oder verlorenen Geldbetrag.

eines der Partner innerhalb des Abschlußbereiches liegt. Ein Beispiel für eine derartige „Verhandlung“ ist der Einsatz von Katalogen.

2. Der Verkäufer geht unilaterale Konzessionen ein, bis er in den Abschlußbereich und damit unter den Indifferenzpreis des Käufers gerät. Diese Vorgehensweise des „Abwärtsbietens“ zeigt z.B. das Verfahren der holländischen Auktion.
3. Der Käufer geht unilaterale Konzessionen ein, bis er den Abschlußbereich des Verkäufers erreicht. Dieses „Aufwärtsbieten“ charakterisiert das Verfahren der bekannten englischen Auktion.
4. Schließlich gibt es die Möglichkeit, dass sich beide Agenten gleichzeitig aufeinander zu bewegen. Diese monotone Abfolge von Konzessionen (Rosenschein/Zlotkin 1994b) auf beiden Seiten kennzeichnet eine sogenannte Doppelauktion (Friedman 1993).

In der ACL (FIPA 1997) sind verschiedene Verfahren, die eine Aufeinanderfolge von Kommunikationsakten (d.h. ein Kooperationsprotokoll) strukturieren, bereits formalisiert. Dazu gehören z.B. die englische und die holländische Auktion.

Zwei Fragestellungen müssen für die Realisierung einer Verhandlungsstrategie in den Agenten beantwortet werden (De Paula et al. 2000, S.3):

1. Welches sind die Handlungsalternativen der Agenten in einer Verhandlungsrunde? In einer normalen Verhandlung hat jeder Agent grundsätzlich drei Möglichkeiten, auf ein Gebot zu reagieren. Er kann es entweder annehmen, ein Gegengebot machen, oder die Verhandlung abbrechen. Gebote können aber auch alternative Produktvorschläge oder zeitliche Grenzen (Ultimatum), die den Verhandlungsdruck erhöhen, umfassen, wodurch die Dimensionalität der Entscheidungssituation gesteigert wird.
2. Wie entscheiden die Agenten, welche Alternative in einer Runde gewählt wird? Die Entscheidungssituation, welcher von den o.a. Pfaden eingeschlagen wird, muß sowohl vergangene Erfahrungen als auch prognostizierte Ergebnisse einbeziehen. Dabei können drei Ansätze unterschieden werden (Preist 1998):
 - Der **regelbasierte Ansatz**: eine Verhandlungsstrategie ist im Agenten festgeschrieben, indem ein Regelsatz angibt, wie der Agent verhandeln soll und unter welchen Umständen ein Angebot akzeptiert werden soll (Kreifelt/von Martial 1991, Barbuceanu/Fox 1996, Mouaddib 1997, Sierra et al. 1997).

- Der **spieltheoretische Ansatz**: Eine Verhandlungsstrategie ergibt sich aus der spieltheoretischen Analyse eines Verhandlungsproblem. Hierbei wird vorausgesetzt, dass genügend Informationen vorliegen, um zu validen Analyseergebnissen zu kommen (Rosenstein/Zlotkin 1994, Vulkan/Jennings 1998, Luce/Raiffa 1957, Kraus/Wildenfield 1991).
- Der **adaptive Ansatz**: Einfache Agenten passen ihre Strategie an, indem sie das Verhalten des Marktplatzes und ihre eigene Leistung in Beziehung zueinander setzen (Sycara 1989, Bussmann/Müller 1992, Sathi/Fox 1989, Durfee/Montgomery 1990, Tesfatsion 1997, Cliff/Bruten 1998).

Diese unterschiedlichen Ansätze sollen im folgenden kurz vorgestellt und ihre Vor- und Nachteile diskutiert werden.

3.5.3.1 Regelbasierte Ansätze

Regelbasierte Ansätze gehen davon aus, dass die Umgebungssituation bekannt und statisch ist. Dem Agenten wird durch seinen Prinzipal ein abgeschlossenes Regelsystem für sein internes Modell übermittelt, mittels dessen er alle vorstellbaren Entscheidungssituation innerhalb dieser Umgebung bewältigen kann. Im Kontext einer Verhandlungssituation bedeutet das, dass die möglichen Züge des Gegners bekannt sind und für jede daraus entstehende Situation eine Bewertung a priori möglich ist.

Sofern die marktliche Koordination über feststehende Kataloge durchgeführt wird, ist eine derartige Verhandlungsstrategie sehr einfach zu realisieren. Die notwendige „friß-oder-stirb“-Entscheidung ist eine Regel, die sich auch nach dem n -ten Besuch eines Kataloges nicht verändert, wenn dieser in Sortiment und Preisen statisch ist. Ein Lernen des Agenten oder eine Anpassung der Konsumentenstrategie verändert die Entscheidungssituation nicht. Solche einseitigen Mechanismen, zu denen neben Katalogen auch die holländische Auktion gehört, sind mit Regeln leicht beherrschbar. Ganz anders verhält es sich bei beidseitigen Mechanismen, z.B. der Doppelauktion. Ein regelbasierter Ansatz müsste hier sämtliche Entscheidungsalternativen des Gegenübers einbeziehen, die rekursiv jedoch von den eigenen Entscheidungen abhängen. Dieser Rekursionsbaum macht eine exakte Entscheidung unmöglich und verlangt nach der heuristischen Festlegung eines Mindestinkrements zum Abbruch der Rekursion.

Für eine hochdynamische Umgebung, wie einen dezentralisierten elektronischen Markt, ist der regelbasierte Ansatz insgesamt zu wenig adaptiv. Es gibt aber Teilbereiche, in denen feste Regeln bereits Ausdruck einer existierenden Norm sind und keine Adaptivität (und der damit verbundene Implementationsaufwand) notwendig erscheint. Beispielsweise kann als eine feste

Regel definiert werden, keine Verhandlung weiterzuverfolgen, bei welcher der im Gebot des Gegners geäußerte Initialpreis doppelt so hoch ist wie der aus der Vergangenheit perzipierte Marktpreis (Wucherregel). Diese Regel ist intuitiv verständlich, hält aber einer theoretischen Betrachtung nicht stand (warum doppelt so hoch und nicht das Anderthalbfache?).

Keine Adaption zu implementieren und z.B. die Entscheidungen der Agenten auf statischen Regeln beruhen zu lassen, ist nur dann eine Option, wenn sich die Umwelt des Agenten in keiner Weise verändert. In den hier zugrundegelegten Szenarien offener Systeme führt dieses Vorgehen in kürzester Zeit zu einem Missverhältnis zwischen dem tatsächlichen Umweltzustand und der Perzeption der Umwelt durch den Agenten, was letztendlich zu falschen Prognosen über den Nutzen von alternativen Handlungsentscheidungen und zu immer schlechteren Ergebnissen aus den Aktionen des Agenten führt.

3.5.3.2 Spieltheoretische Ansätze

Der Einsatz von spieltheoretischen Konzepten (Rosenschein/Zlotkin 1994b, Harsanyi 1977) geht von der Betrachtung einer Marktsituation als Spiel zwischen einem Anbieter und einem Nachfrager aus, bei dem beide um die Verteilung des Nutzens aus der Transaktion konkurrieren. Das interne Modell des Agenten besitzt neben wenigen festen Regeln ein Entscheidungskalkül, welches das Verhalten des Verhandlungspartners explizit einbezieht. Dadurch wird es für die Agenten sinnvoll, Erkenntnisse über das strategische Verhalten des Gegenübers in die Beurteilung des Verhandlungsangebotes einfließen zu lassen. In Märkten handelt es sich um sogenannte wertorientierte Domänen (*worth-oriented domains*), die sich durch eine in Geldeinheiten abstufbare Bewertung alternativer Handlungsmöglichkeiten auszeichnen. Für die Agenten ist es möglich, Konzessionen einzugehen, d.h. Alternativen zu verschiedenen Preisen anzubieten oder nachzufragen. Die Nutzung von spieltheoretischen Konzepten zur Implementation in handelnden Agenten verlangt nach der Beachtung einiger Einschränkungen, die im Spannungsfeld zwischen theoretischer Analyse und robuster Implementation in einem realistischen Anwendungsumfeld liegen. Grundsätzlich wird ein *“product-maximizing negotiation protocol, such as in Nash bargaining theory”* (Rosenschein/Zlotkin 1994b, S. 37) angewendet. Dabei muß lediglich beachtet werden, dass *“it really doesn’t matter which one they use as long as it’s symmetric; that is, it maximizes the product of the utilities.”* (ebda.) Die Einschränkungen sind im einzelnen:

1. Der Zeithorizont der Agenten muß prinzipiell offen sein, d.h. es muß eine unendlich große Anzahl zukünftiger Transaktionen geben. Sollte ein Zeitpunkt entdeckt werden können, ab dem die Nichtkooperation eines Agenten nicht mehr sanktioniert werden kann, entsteht ein sogenanntes Endspielproblem – alle Agenten mit dem gleichen Entscheidungsproblem wer-

den ab diesem Zeitpunkt keine Konzessionen mehr eingehen, so dass überhaupt keine Kooperation bzw. Transaktion mehr stattfindet. Ohne Kooperation gibt es aber keine Koordination und damit auch keinen Markt.

2. Die Strategie muß der jeweiligen Situation und der prognostizierten Zukunft angepasst werden können. Erst durch eine solche Reflektionsschleife wird ein Strategiewechsel möglich und der Agent stellt einen ernstzunehmenden Verhandlungspartner da. Ein Agent, der erkanntermaßen immer kooperiert, kann schnell ausgebeutet werden. Ein Agent, der immer defektiert, wird bald keine Verhandlungspartner mehr finden. Erst die Fähigkeit, zwischen beiden Extremen zu wechseln, ermöglicht die strategische Ausnutzung von Verhandlungssituationen.
3. Ein hoher Diskontparameter sorgt dafür, dass künftigen Ergebnissen eine höhere Bedeutung zugewiesen wird als vergangenen Ergebnissen. Diese Einstellung reflektiert die Lernfähigkeit potentieller Partner und die Prognose, dass sich die Gesamtsituation (Preise und Mengen) im Markt ständig verändert.
4. Die Kooperationsrente, d.h. die Erlöse aus Transaktionen mit einer kooperativen Strategie, darf nicht geringer sein als der Ausbeutungsgewinn, d.h. die Erlöse aus Transaktionen mit Defektion²⁷. Dieses Verhältnis kann der einzelne Agent nicht selbst bestimmen, sondern es ergibt sich aus der Marktsituation im Zeitablauf. In der physischen Welt wird diese Differenz durch das Eingreifen zentraler Instanzen oder dezentral durch soziale Vereinbarungen und Normen aufrechterhalten. Ähnliche Institutionen müssen analog für agentenbasierte elektronische Märkte vorgesehen werden.
5. Die Spieltheorie geht davon aus, dass die Menge der Handlungsoptionen eines Partners bekannt ist und die Wahrscheinlichkeit der Auswahl einer Option von berechenbaren Faktoren abhängt. In realen Anwendungsszenarien wird die Wahrscheinlichkeit der Wahl einer Strategie durch das Gegenüber nicht genau bestimmt werden können; noch nicht einmal die Zahl seiner Optionen muß bekannt sein. Ein Agent muß in diesem Fall Erwartungswerte zuweisen, die z.B. aus vergangenen Transaktionen gewonnen werden können. Je nach Komplexität der Agenten sind auch neuartige Strategien möglich, die spieltheoretische Überlegungen sprengen, in dem neue Handlungsoptionen generiert werden, die durch das Gegenüber nicht antizipiert worden sind. Je komplexer die Verhandlungsstrategien der Agenten sind und je näher diese dem Verhalten des menschlichen Prinzipals entsprechen, um so komplizierter wird es, die Strategie des Gegenübers zu modellieren und daraus eigene Entscheidungen abzuleiten.

27 Diese Bedingung betrifft alle Situationen, die sich als Gefangenendilemma beschreiben lassen (vgl. Axelrod 1985).

6. Spieltheoretische Überlegungen bedingen ebenfalls die Rationalität des Handelns der Partner. Es muß ausgeschlossen werden können, dass Agenten zufällig und ohne Grund ihre Strategien ändern, da die eigenen Verhaltensberechnungen auf der Annahme rationalen Handelns des Gegenübers beruhen.

Kraus (1997, S. 83ff.) untersucht Situationen, in denen eine kleine Anzahl eigennütziger Agenten Ressourcen teilt oder von der Aufteilung von Aufgaben profitiert. Bei ihrem Ansatz sind die Verhandlungsaktionen der einzelnen Agenten Züge in einem nicht-kooperativen Spiel und die Rationalitätsannahme bezüglich der Teilnehmer wird durch das Nash-Gleichgewicht ausgedrückt. Für die praktische Anwendung von spieltheoretischen Ansätzen in der Implementation von Agenten nennt Kraus (1997, S. 84) fünf Probleme, die für eine erfolgreiche Adaption gelöst werden müssen:

1. Auswählen eines Modells des strategischen Handelns, das für das spezifische VKI-Problem anwendbar ist. Ein Beispiel ist Rubinsteins Modell alternierender Angebote (Rubinstein 1982, Osborne/Rubinstein 1990). Dadurch wird beispielsweise ein zeitliches Fortschreiten der Verhandlung mit in die Strategien einbezogen. In Fällen, in denen ein früher Zeitpunkt der Transaktion bzw. eine kurze Verhandlungszeit zur Nutzensteigerung beiträgt, lassen sich keine Modelle anwenden, die von Gleichgewichtszustand zu Gleichgewichtszustand springen und dazwischen eine zeitlich unbegrenzte Berechnung der einzelnen Aktion zulassen.
2. Zusammenführen des DAI-Szenarios mit den spieltheoretischen Definitionen des gewählten Modells. Hier muss festgelegt werden, welches Objekt der realen Welt dem jeweiligen Objekt des Modells entspricht. Rubinsteins Modell erfordert die Festlegung von Spielern, Vereinbarungen, dem Verhandlungsprotokoll und den Strategien der Spieler. Je nach Szenario entsprechen diese Rollen unterschiedlichen Trägern.
3. Identifizieren von Gleichgewichtsstrategien. Die Spieltheorie ist immer verknüpft mit der Vorstellung eines stabilen Gleichgewichts. Die Strategien, die von der Spieltheorieforschung vorgeschlagen werden, führen zu einem (Nash-)Gleichgewicht bzw. Pareto-Optimum.
4. Entwickeln niedrig-komplexer Techniken, um nach passenden Strategien zu suchen. Ranglisten von Strategien können vorgegeben werden, wenn z.B. die Startaufstellung an Spielern und Ressourcen bekannt ist. Ist dies nicht der Fall, muss es dem Agenten möglich sein, zur Laufzeit nach veränderbaren Kriterien eine Rangfolge der Strategien bestimmen zu können.
5. Bereitstellen von Nutzenfunktionen. Die Anwendung spieltheoretischer Konzepte geht unbedingt vom Vorhandensein einer Nutzenfunktion im Agenten aus, die ihm zur Laufzeit bekannt ist und im Allgemeinen vom Prinzipal beim Design des Agenten mitgegeben wird.

Die Möglichkeit, dass eine eigene Nutzenfunktion nicht (oder nicht vollständig) bekannt ist, wird nicht in Betracht gezogen, da sie zur Nichtanwendbarkeit der Spieltheorie in diesem Kontext führt.

Folgendes Fazit läßt sich für die Anwendung spieltheoretischer Ansätze ziehen:

„[They are] suitable for environments with a relatively small number of agents. But, in very large agent-communities, these negotiation methods are typically too computationally complex and time-consuming. Furthermore, direct communication connections between all of the agents may be impossible or too costly to establish.”
(Kraus 1997, S.86).

In Fällen, die durch eine relativ geringe Anzahl verschiedener Agenten (höchstens 20) und eine vordefinierte Festlegung von Regeln und Protokollen gekennzeichnet sind, lassen sich spieltheoretische Mechanismen anwenden (Kraus 1997, S. 82). Für das dieser Arbeit zugrundegelegte Szenario eines großzahligen, offenen Marktes erscheinen spieltheoretische Mechanismen als Grundlage der Verhandlungsstrategie nicht geeignet. Offene Systeme, in denen zwar die Kommunikation definiert ist, aber eine große Anzahl von Agenten mit nur geringen Einwirkungsmöglichkeiten seitens einer steuernden Instanz miteinander interagiert, sind eher durch die besser skalierenden adaptiven Verhandlungsansätze abzubilden, die im folgenden Abschnitt beschrieben werden.

3.5.3.3 Adaptive Ansätze

Regelbasierte Ansätze gehen davon aus, prinzipiell sämtliche Entscheidungen im vorhinein abbilden zu können. Spieltheoretische Ansätze basieren auf der fallweisen Bewertung einer Entscheidungssituation. Beide Ansätze hängen von der den Agenten verfügbaren Information ab, die für eine erfolgreiche Analyse als vollständig angenommen wird. Im Gegensatz dazu gehen adaptive Ansätze von vornherein von unvollständiger Information über Umwelt und Gegner, sowie daraus folgenden fehlerhaften Entscheidungen aus:

“We cannot possibly hope to enumerate the set of reasons an agent might have for preferring one set of answers over another, and we should not try to do so.” (Vidal/Durfee 1996, S.4.).

Wirtschaftspsychologische Ansätze, die versuchen, mögliche Strategien menschlicher Verhandlungspartner zu identifizieren und Gegenstrategien vorzuschlagen können z.B. in populärwissenschaftlichen Büchern wie den bei Kraus (1997) genannten (z.B. Druckman 1977, Hall 1993) gefunden werden. Deren Ratschläge sind allerdings formal nicht ausreichend spezifizierbar, um sie in allgemeingültige Methoden für die Anwendung in Multi-Agenten-Systemen

umsetzen zu können. Diese Modelle wurden für die Entwicklung von Verhandlungsheuristiken (Kraus/Lehmann 1995) oder als Grundlage für die Entwicklung eines logischen Modells der Verhandlung (Kraus et al. 1993) genutzt.

Ein einfaches Beispiel eines adaptiven Ansatzes zeigt Preist (1998, S. 4f.). In ihm werden heuristische Regeln mit einer einfachen Lernregel kombiniert. Jeder Agent ist willens, zu einem Preis zu handeln, der unterhalb (Käufer) oder oberhalb (Verkäufer) seines Limitpreises liegt. Seine autonome Entscheidung liegt darin, einen Preis zum Kaufen oder Verkaufen festzulegen. Als Marktmechanismus wird eine rundenbasierte kontinuierliche Doppelauktion (CDA) verwendet, in der alle Agenten die aktuellen Gebote kennen. Die Regeln sind (Preist 1998, S. 4):

```
„For BUYERS;
If the highest bid is below the lowest offer, then target just above the
highest bid.
If the highest bid is equal or greater than the lowest offer, then tar-
get the lowest offer.
For SELLERS;
If the highest bid is below the lowest offer, then target justbelow the
lowest offer.
If the highest bid is equal or greater than the lowest offer, then tar-
get the highest bid.“
```

Die Anwendung dieser Regeln führt zu unterschiedlichen Resultaten in der Realisierung des eigenen Nutzens. Die Heuristiken bestimmen, welchen Gebotspreis der Agent als Ziel anpeilt und eine Lernrate mit dem Wert zwischen 0 und 1 bestimmt, wie weit sich der Agent von seiner gegenwärtigen Wertschätzung auf den Zielwert zubewegt. Die Widrow-Huff-Lernregel (Rumelhart et al. 1986), die auch in Neuronalen Netzen verwendet wird, passt diesen Parameter im Zeitablauf an. Preist (1998) erhält als Ergebnis stabile Gleichgewichtspreise für die gehandelten Güter in seinem Mehragentensystem, ähnlich der in Abbildung 12 dargestellten Ergebnisse der Implementation von Cliff (1997).

Diese Implementation ist exemplarisch für die Realisierung adaptiver Strategien. Eine Anzahl numerischer oder boolescher Variablen determiniert die Strategie und die Variablenwerte werden durch ein Lernverfahren verändert. Unter den in Agenten einsetzbaren Lernverfahren finden sich Neuronale Netze (Fausett 1994), Q-Learning (Sandholm/Crites 1995), Classifier Systems (Holland 1992), Bayes'sche Netze (Neal 1996) und Evolutionäre Algorithmen (Goldberg 1994). Es sollte aber darauf hingewiesen werden, dass das Ziel des Lernverfahrens in der schnellen Anpassung an sich verändernde Umweltbedingungen liegt. Je komplexer die Implementation des Lernverfahrens, desto länger dauert die Berechnung und desto größer ist die Differenz zwischen der aktuellen und der beurteilten Umweltsituation, für die die Strategie

angepasst wurde. Wenn in den einzelnen Agenten individuelle Lernverfahren zum Einsatz kommen, dann ergeben sich MAS notwendigerweise als komplexe, dynamische Systeme, in denen regelbasierte und spieltheoretische Strategien zu statisch sind, um auf Dauer Erfolg versprechen zu können²⁸.

In der in Kapitel 4 vorgestellten Implementierung wird ein einfaches Modell der Preisanpassung verwendet, welches sich an Preists (1998) und Cliffs (1997) Implementationen sowie an Brenners Variation-Imitation-Decision-Modell (Brenner 1999) anlehnt. Zusätzlich werden von Smith/Taylor (1998) vorgeschlagene Elemente eines dezentralen evolutionären Algorithmus implementiert. Aus den Ergebnissen von Preist und Cliff wird die grundsätzliche Funktionsfähigkeit und Anwendbarkeit dieses einfachen Preisanpassungsmodelles für die marktliche Koordination abgeleitet.

3.5.3.4 Welche Variablen bestimmen die Strategie?

In einem offenen elektronischen Markt soll davon ausgegangen werden, dass die Verhandlungsstrategie der Agenten einzeln durch ihre jeweiligen menschlichen Besitzer (Marktteilnehmer) definiert werden muß. Die Parameter, die die Strategie der Agenten bestimmen, stellen eine Formalisierung der Informationen dar, die ein Mensch als Entscheidungsgrundlage in einer gleichen Situation nutzen würde. Die Aufgabe hier ist, diejenigen Variablen zu identifizieren, die Verhandlungsentscheidungen von Menschen zugrunde liegen und eine adäquate Abbildung zur Implementation in Software-Agenten zu liefern.

Pruitt (1981) untersucht die Verhandlungsführung zwischen menschlichen Agenten im Alltag und in politischen und wirtschaftlichen Szenarien. Die dort zusammengeführten Überlegungen sind als Basis für die Beschäftigung mit der Verhandlungsführung zwischen Software-Agenten zu sehen. Hinter jedem Software-Agenten steht nach wie vor ein natürlicher und juristischer Prinzipal, dessen Eigeninteresse das Verhalten des Agenten bestimmt. Die Verhandlungen zwischen den Agenten sind "Stellvertreterkriege", die sich durch eine eingeschränkte Syntax der formalisierten Kommunikation unterscheiden. Die Ziele, Handlungsoptionen und strategischen Entscheidungen der SW-Agenten sind aber von denen ihrer menschlichen Prinzipale abgeleitet. Deswegen gelten die Überlegungen für das Verhalten menschlicher Verhandlungspartner im Grossen und Ganzen unverändert für Software-Agenten weiter. Die Notwendigkeit der Formalisierung und Implementation von strategischen Optionen schränkt diese jedoch ein.

28 Desweiteren wird sich auf Systemebene ebenfalls eine adaptive Weiterentwicklung ergeben, die als Markoff-Modell (vgl. Honerkamp 1990) beschrieben werden kann und zu einer Koevolution der Agentenstrategien (vgl. Vriend 1999) führt.

Das *Strategic Choice* Model geht davon aus, dass Verhandlungspartner in alternierenden Aktionen ihre Positionen überdenken und Vorschläge zur Konfliktlösung machen. Jeder Verhandlungspartner wählt zu jedem Handlungszeitpunkt (d.h. der Partner ist am Zug) zwischen fünf grundsätzlichen Strategien aus (Pruitt 1981, S. 15):

- **Unilaterale Konzession:** Der am Zug befindliche Partner vermindert einseitig die Distanz zwischen den Positionen der beiden Parteien.
- **Kompetitives Verhalten:** Der am Zug befindliche Partner bleibt bei seiner Position und versucht, den Partner durch Druck (Argumente, Drohungen etc.) zur Aufgabe seiner Position zu bewegen.
- **"Koordinatives" (Pruitt) Verhalten:** Beide Partner kollaborieren und versuchen durch gemeinsame Handlungen die Kontroverse aufzulösen. Beispiele können sein ein Kompromißvorschlag, die Einberufung einer spannungsvermindernden Initiative oder die Kooperation mit einer dritten Partei, die die Kontroverse auflöst. In diesem Fall geht die *"goal/expectation hypothesis"* (Pruitt/Kimmel 1977) davon aus, dass beide Partner zum einen diese Koordination wollen und zum anderen dem jeweiligen Gegenüber in dieser Hinsicht vertrauen. Gegenseitiges Vertrauen ist besonders wichtig, da *"coordinative behavior always lays one open to the possibility of being exploited. Such exploitation seems less likely to materialize if the other party also seems motivated to achieve coordination"* (Pruitt 1981, S.16).
- **Nichtstun:** Der am Zug befindliche Partner führt die Verhandlung nicht weiter und macht kein Gegenangebot. Dies unterscheidet sich vom kompetitiven Verhalten, bei dem das letzte Gebot noch einmal bekräftigt wird. Typischerweise führt dieses Vorgehen nach Überschreiten einer Zeitschranke zu einer Reaktion des Verhandlungspartners. Die Wahl dieser Alternative impliziert, dass der Partner nicht notwendigerweise an einer gemeinsamen Zielerreichung in der Verhandlung interessiert ist, sich diese Möglichkeit aber noch offenlässt.
- **Abbruch der Verhandlung:** Der am Zug befindliche Partner zieht sich einseitig durch eine eindeutige Willensäußerung (z.B. das Übermitteln eines Sprechaktes *refuse* bzw. *reject*) von der Verhandlungssituation zurück. Der Konflikt, der ursprünglich zu der Verhandlung geführt hat, besteht ohne Änderung weiter. Wie beim Nichtstun zeigt die Wahl dieser Strategie, dass auch die Nichterreichung des Verhandlungszieles von Beginn an im Kalkül des Verhandlungspartners lag.

Um Verhandlungen zwischen menschlichen Teilnehmern weiter zu formalisieren, führt Pruitt das Konzept des *"Demand level"* und der *"Concession rate"* ein (Pruitt 1981, S.19).

Der **Demand level** ist der Nutzenzuwachs, den der Verhandlungspartner bei dem gegenwärtig bestehenden Angebot erreichen würden. In betriebswirtschaftlichen Situationen stimmt der Nutzenzuwachs für einen Verkäufer mit seinem Gewinn überein, für einen Nachfrager mit der Ersparnis, d.h. der Differenz zwischen Kaufpreis und persönlichem Wert. Eine **Concession** ist die Veränderung des *Demand Level* in Richtung des Vorschlages des Verhandlungspartners. Damit vermindert sich gleichzeitig der eigene Nutzen. Die **Concession rate** ist die Geschwindigkeit, mit der der *Demand Level* über die Zeit abnimmt. Wie Pruitt selber einschränkt, sind diese Definitionen *"unproblematical, when only one issue or underlying value is being considered, as in a simple wage or price negotiation"* (Pruitt 1981, S. 19). Ein Verhandlungspartner kann in diesem Fall jede Verhandlungsalternative (den gebotenen Preis) in Hinblick auf ihren Wert für ihn selbst überprüfen. Da andere (z.B. politische oder militärstrategische) Fragestellungen im Rahmen dieser Arbeit nicht betrachtet werden sollen, ist die vorliegende Definition ausreichend.

Dagegen sind die folgenden Ergebnisse von höchstem Interesse für die in dieser Arbeit beschriebene Implementation:

"One ist that lower initial demands and faster concessions on the part of one or both parties make agreement more likely and more rapidly reached. This can be logically deduced because lower demands mean that the parties are less likely to become discouraged and break off negotiation and that their demands will coincide at an earlier point in time" (Pruitt 1981, S.20).

Pruitt zitiert mehrere empirische Untersuchungen (Bartos 1974, Benton et al. 1972, Hamner 1974, Harnett/Vincelette 1978), die dieses Ergebnis stützen. Ein interessanter Nebeneffekt in menschlichen Verhandlungssituationen ist, dass ein zu schnelles Entgegenkommen die Zielerreichung verlangsamt: *"very fast movement encourages the other party to expect further large concessions, discouraging the other from making its own concessions"* (Pruitt 1981, S.20).

Zum anderen beschreibt er, dass *"a second conclusion is that, if agreement is reached, a bargainer who makes larger initial demands and smaller concessions will achieve a larger outcome (and the other party a correspondingly smaller outcome)"* (Pruitt 1981, S.20), ebenfalls unterstützt durch verschiedene empirische Studien (Bartos 1974, Benton et al. 1972, Chertkoff/Conley 1967).

Diese zwei Ergebnisse führen zu einem dritten Ergebnis, *"that there will be an inverted U-shaped relationship between a bargainer's typical demand and his or her average outcome over a series of negotiations"* (Pruitt 1981, S. 21). Verhandlungspartner, die mit einem geringen *Demand Level* in eine Verhandlung einsteigen, werden schneller ein Ergebnis erreichen,

nur einen geringen Nutzen erwirtschaften. Diejenigen, die mit hohen Erwartungen (und einem ebensolchen *Demand Level*) beginnen, laufen Gefahr, zu keiner Einigung zu gelangen und haben mit dieser Strategie ebenfalls über die Zeit nur einen geringen Nutzen. Verschiedene bei Pruitt zitierte empirische Studien (Benton et al. 1972, Hamner 1974, Harnett/Vincelette 1978) stützen dieses Ergebnis.

Aus diesen Ergebnissen folgen mehrere Anforderungen an die Implementation von Verhandlungsstrategien in Software-Agenten:

Die Höhe des *Demand Levels*, einer *Concession* und der *Concession Rate* müssen als Teil der Verhandlungsstrategie durch den Prinzipal initial vorgegeben werden können. Diese Variablen können sich über die Lebensdauer des Agenten ändern, sie sind jedoch ursächlich für die wahrgenommene Strategie des Agenten. Eine starke Abweichung in diesen drei Variablen von den Vorgaben des Prinzipals führt zu grundsätzlich anderem Verhalten des Agenten und damit zu (positiv als auch negativ) veränderten Nutzenrealisierungen.

Es gibt keine absolut "richtigen" Werte dieser Variablen. Der Nutzen, der aus einer bestimmten Kombination dieser Werte erwächst, bestimmt sich ausschließlich in Bezug zu der Strategie der (potentiellen) Verhandlungspartner. Ein Lernen dieser Werte, d.h. eine zeitliche Veränderung der Variablen, ist notwendig, um in Zukunft einen höheren Nutzen zu erzielen. In allen drei Variablen können unabhängig voneinander zu hohe oder zu niedrige Werte zu geringeren Nutzenrealisierungen führen. In der in dieser Arbeit beschriebenen Verhandlungssituation von Software-Agenten in elektronischen Märkten, das sei an dieser Stelle vorweggenommen, konnten diese Ergebnisse repliziert werden.

Die Werte dieser drei Variablen werden in natürlichen Umgebungen nach Pruitt (1981) durch folgende Parameter determiniert:

- Erwartungen über das ultimative Gebot des Gegenübers – Verhandlungspartner wählen ihre eigenen Gebote oder Konzessionen nach dem erwarteten Verhalten des Gegenübers: "*The farther the other is expected to concede, the more will be demanded and the less will be conceded*" (Pruitt 1981, S. 21). Dieser Parameter selbst wird durch komplexe psychologische Vorgänge beeinflusst.
- Positions- und Imageverlust – Hohe Gebote und langsame Konzessionen sind meist motiviert durch Befürchtungen über möglichen Positions- oder Imageverlust. Positionsverlust ist dabei die Aufgabe einer erwünschten Alternative, Imageverlust ist "*the development in other people's eyes of an impression that the bargainer lacks firmness (i.e. is ready to make substantial concession)*" (Pruitt 1981, S. 23).

- Grenze und Höhe der Erwartungen – Die Grenze der Erwartungen beschreibt die unterste Position, unterhalb derer keine Konzession mehr gemacht wird. Die Höhe der Erwartungen beschreibt hingegen die Obergrenze, auf die vor Beginn der Verhandlung zugestrebt wird.
- Zeitdruck kann aus unterschiedlichen Gründen hervorgerufen werden. Einerseits kann die Fortsetzung der Verhandlung Geld kosten (Opportunitätskosten aus Nichtbeschäftigung mit anderen Alternativen, Kosten von Intermediären, Qualitätsverlust des Verhandlungsgegenstands z.B. bei verderblichen Gütern). Andererseits stellt die Konfliktaustragung selbst eine Belastung auch zukünftiger Kooperationsmöglichkeiten dar und bindet Ressourcen.

Ausgehend von den obigen Überlegungen, welche Parameter die Verhandlungsstrategie eines Menschen bestimmen, können diese in Form einer adaptiven Strategie in den Agenten implementiert werden. Um ausgehend von wenigen Parametern ein komplexes, nicht zu einfach zu prognostizierendes Verhalten zu realisieren, kann z.B. die Form eines nicht-deterministischen endlichen Automaten gewählt werden. Dabei basiert die Strategie auf Wahrscheinlichkeiten, die anders als bei Regeln nicht eine Strategie fest vorgegeben, sondern zwischen mehreren Alternativen gewichtet zufällig "auswählen". Ob ein Agent in diesem Fall z.B. eine Konzession eingeht oder nicht, entscheidet sich stochastisch anhand einer Wahrscheinlichkeit p . Wenn eine Entscheidung über eine Konzession ansteht, wird sich dieser Agent neu und für seine Umwelt im Einzelfall nicht vorhersehbar entscheiden. Auf lange Sicht ist er berechenbar, denn stochastisch wird er sich p annähern, so dass ex post durchaus eine Aussage über seine "Strategie" möglich ist. Mit diesem Implementationstrick ist es möglich, einen einfachen endlichen Automaten zu erstellen, der auf wenigen Parametern basiert und der einem komplexen deterministischen Automaten mit vielen Parametern äquivalent ist. Die Strategie des einzelnen Agenten kann auf stochastischen Parametern wie der Wahrscheinlichkeit, eine unilaterale Konzession einzugehen, oder der Wahrscheinlichkeit, mit den vorliegenden Angeboten zufrieden zu sein und kein weiteres Angebot eines anderen Anbieters anzufordern (und damit dem Zeitdruck nachzugeben), basieren.

Warum Agenten überhaupt Konzessionen eingehen sollten, begründet Pruitt folgendermaßen:

"A bargainer's demand level can be thought of as the level of benefit to the self associated with the current offer or demand. A concession is a change of offer in the supposed direction of the other party's interests that reduces the level of benefit sought. Concession rate is the speed at which demand level declines over time. [...] Concessions ordinarily result from the belief that they will hasten agreement, prevent the other party from leaving the negotiation, or encourage the other to make reciprocal

concessions" (Pruitt 1981, S. 19).

Eine Konzession einzugehen vermindert in erster Linie den eigenen Nutzen und ist von daher grundsätzlich nur durch die Einbeziehung eines zeitlichen Verlaufs sinnvoll zu begründen. Falls die Verhandlung über einen längeren Zeitraum andauert, besteht die Gefahr eines Verhandlungsabbruchs durch den Partner; in diesem Fall gibt es keine Nutzensteigerung für irgendeinen Teilnehmer.

3.5.3.5 Adaption der Strategie mit evolutionären Algorithmen

Evolutionäre Algorithmen können auf dieser Grundlage eingesetzt werden, um nach solchen Parameterwerten zu suchen, die zur Verbesserung des Erfolgs eines Agenten führen (Holland et al. 1991) und nicht von der Ausgangsausstattung abhängen. Jeder Agent repräsentiert zu Anfang eine relativ zufällige Verteilung der Parameterwerte. Im Verlauf der Marktteilnahme wird er die Zusammenstellung dieser Parameter verändern müssen, um trotz der gleichzeitigen Veränderung der anderen Agenten seinen Nutzen aus Transaktionen zumindest in gleicher Höhe zu gewährleisten. In diesem Sinne findet eine (ständige) Optimierung der Strategieparameter statt.

Evolutionäre Algorithmen (EA) stellen eine Klasse von Optimierungsverfahren dar, zu denen Genetische Algorithmen, Genetische Programmierung und Evolutionsstrategien gehören (Back et al. 1997, Schöneburg et al. 1994). Der Einsatz genetischer Algorithmen als Instrument ökonomischer Optimierung wird z.B. in der Forschungsrichtung der evolutiven Ökonomik (Witt 1993, Dosi/Nelson 1994) propagiert. Im vorliegenden Fall könnte man auf die Idee kommen, dass EA eine Möglichkeit bieten, das wirtschaftliche Koordinationsergebnis des Gesamtsystems, bestehend aus den teilnehmenden Agenten, zu optimieren. Der Erfolg einer Agentenpopulation als Ganzes kann in technischer (Rechen- oder Kommunikationseffizienz) oder ökonomischer Hinsicht (Paretoeffizienz oder soziale Wohlfahrt) ermittelt werden (Sandholm 1996). Eine Betrachtung eines Gesamtzieles kann nur dort stattfinden, wo das MAS als Ganzes ein Ziel erreichen soll. Dies setzt die Entwicklung und Anwendung des MAS durch eine koherente organisatorische Einheit, mit dem Ziel, ein bestimmtes, bekanntes Problem zu lösen, voraus. In einem offenen System wie dem hier betrachteten, in dem unterschiedliche Entwickler von Agenten jeweils individuelle Ziele formulieren, fehlen diese Voraussetzungen. Das kleinste gemeinsame Ziel, auf das sich alle Entwickler verständigen können, ist die Aufrechterhaltung der Koordinationsleistung über die Laufzeit des MAS (dies kann auch als "unendlich lang" aufgefasst werden), sowie die individuelle Nutzenmaximierung aus der Kooperation mit anderen Agenten. Ein Streben z.B. nach optimaler Wohlfahrt, d.h. die Maximierung der Nutzensumme aller Agenten, würde ein kooperatives Grundverständnis aller

Beteiligten voraussetzen und damit aktives benevolentes Handeln. Diese Annahme wurde jedoch bereits ausgeschlossen. Gerade bei genetischen Algorithmen wird immer von einer zentralen Optimierung ausgegangen (vgl. Chattoe 1998), ebenso bei den weiter oben genannten Nicht-EA-Verfahren Neuronaler Netze und Bayes'scher Netze.

Stattdessen muß ein Adaptionproblem in Betrachtung gezogen werden, welches Kearney et al. (2000) wie folgt charakterisieren:

„Let us consider a set of similar agents, each of which selfishly seeks to exploit effectively some limited set of resources. Imagine that each agent's behaviour is dictated by a set of (discrete) features, but assume an agent has no particular intelligence about how those features relate to its own success. Given this rather severe set of constraints, one must consider what an agent might do to improve its performance“ (Kearney et al. 2000, S.3).

Tatsächlich trifft dieses Problem auf das Szenario des offenen, elektronischen Marktplatzes zu. Jeder Agent versucht, aus einer begrenzten Geldmenge einen möglichst grossen Anteil zu erhalten. Sein Verhalten wird durch die diskreten Parameter gesteuert, die zu seiner Strategie führen. Das Ergebnis seiner Strategie wird in hohem Maß von den ihm unbekanntem Strategien anderer Agenten beeinflusst. Da es keine Steuerungsinstanz des Marktplatzes gibt, können zentrale Verfahren nicht eingesetzt werden. Im folgenden soll eine adaptive Vorgehensweise für MAS eingeführt werden, die eine strikt dezentrale Implementation eines Genetischen Algorithmus darstellt.

Dabei vergleicht jeder Agent ständig seine eigene Performance mit seiner Umwelt und passt seine Strategie selbständig den erfolgreicherem Agenten an. Bei Genetischen Algorithmen (Goldberg 1989, Mitchell 1994) werden vier Schritte unterschieden, die eine Adaption durchführen: Evaluation, Reproduktion, Rekombination und Mutation. Genetische Algorithmen, die ganze Populationen optimieren, implementieren diese Schritte meist in einer zentralen Instanz. Die einzelnen Phasen entsprechen denjenigen der Genetischen Algorithmen, werden jedoch auf Basis eines alternativen EA-Konzeptes von Smith/Taylor (1998, S.1f.) implementiert, das im folgenden als **STDEA**²⁹ bezeichnet wird:

"To design an agent-based EC system, one must turn the typical GA software design on its head. In common GA software, a centralized GA program stores the GA individuals as data structures, and imposes the genetic operations that generate successive populations. In natural genetics, the equivalent of the centralized GA does not exist.

29 STDEA = **S**miths und **T**aylor's **d**ezentraler **e**volutionärer **A**lgorithmus.

Instead, the evolutionary effects emerge from individual agents. This makes designing an agent-based EC system a matter straight forward analogy."

Der EA wirkt auf die gesamte Population ein, aber er verändert nur Variablen in jeweils einem Agenten. Grafisch kann diese Vorgehensweise wie in Abbildung 15 dargestellt werden. Die grundlegenden Ideen von Smith/Taylor (1998) lassen sich am besten in einer Mischung biologischer und wirtschaftswissenschaftlicher Terminologie beschreiben, die derjenigen der evolutionären Ökonomie nicht unähnlich ist (vgl. Vriend 1994, Chattoe 1998). Die "Gene" des Agenten stellen die Strategieparameter dar, der *Genotyp* jedes Agenten ist seine Kombination der Variablenwerte. Diese Werte sind für den Agenten konstant, er kann sie selbst nicht verändern. Die Strategie, die der Agent zeigt und die von anderen Agenten beobachtet werden kann, ist im biologischen Sinne der *Phänotyp*. Der Genotyp (Parameter) beeinflusst den Phänotyp (Strategie), ein formaler Rückschluß aus dem Phänotyp auf den Genotyp ist aufgrund der hier vorgenommenen Implementation der Strategie als nichtdeterministischer endlicher Automat jedoch nicht möglich.

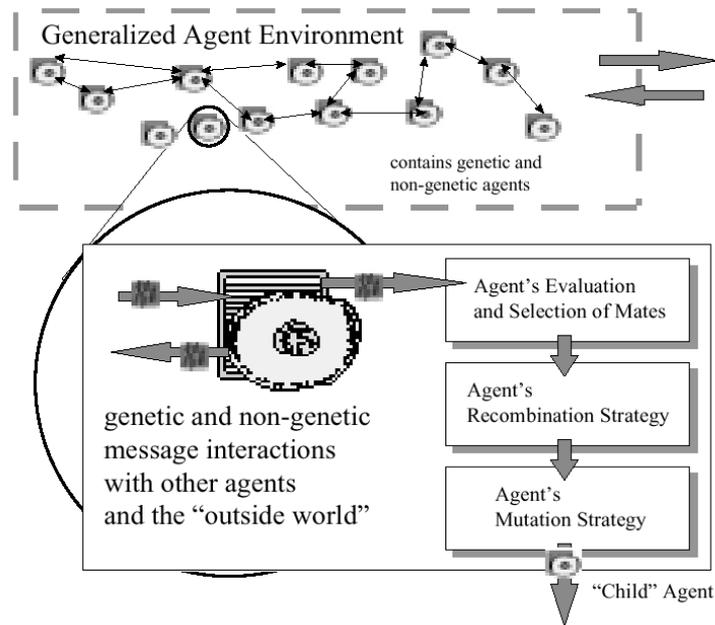


Abb. 15: Prinzip eines agentenbasierten, dezentralen evolutionären Algorithmus (Smith/Taylor 1998, Fig. 2)

In der **Evaluations- und Auswahlphase** des STDEA beurteilt jeder Agent andere Agenten anhand des für ihn ersichtlichen Nutzenzuwachses (der *Fitness*). Die Fitness eines Agenten ergibt sich in einem Markt aus der Höhe des Einkommens, welches der Agent aus Verhand-

lungssituation mit anderen Agenten, die zu erfolgreichen Transaktionen führen, erzielen kann. Um sich mit anderen Agenten auf dem Marktplatz vergleichen zu können und ihre Strategie den erfolgreichen Agenten anzupassen, müssen die Agenten Informationen über die Fitness von Agenten erhalten, die mit Dritten verhandeln und ihre eigenen Daten und Präferenzen ansonsten streng geheim halten. In STDEA werden nach jeder Transaktion Informationen über die beteiligten Agenten an andere weitergegeben. Jeder Agent sendet nach einer erfolgreich abgeschlossenen Transaktion ein sogenanntes *Plumage*-Objekt ab, das seine Fitness sowie seinen Genotyp enthält. Ein *Plumage* kann als eine Menge an beobachtbarer Information interpretiert werden, die im Moment der Transaktion als Schnappschuß über einen der beteiligten Agenten verfügbar wird. In der betriebswirtschaftlichen Realität können offizielle Veröffentlichungen eines Marktteilnehmers wie Börsenmitteilungen und Geschäftsberichte oder inoffizielle Informationen wie Marktgerüchte und Insiderinformationen als derartige *Plumages* betrachtet werden. Jeder Agent wählt zu einem festzulegenden Zeitpunkt anhand der ihm vorliegenden *Plumages* dasjenige mit dem höchsten Fitness (-Versprechen) aus und entnimmt diesen den (besten) Genotyp.

Der Schritt der **Rekombination** wird durch das Verknüpfen dieses besten Genotyps mit dem eigenen Genotyp erreicht. Dabei wird das entsprechende „cross-over“-Verfahren der genetischen Algorithmen genutzt. Letztendlich ergibt sich daraus eine neue Parameterkombination für die Agentenstrategie, die von beiden Eltern-Genotypen abweicht.

Die letzte Phase der **Mutation** verändert den erzeugten Genotyp erneut. Die Attributwerte werden in geringer Weise verändert, was eine nicht exakte Umweltperzeption simuliert. Im Marktmodell stellt diese Unschärfe im Ergebnis die Unwägbarkeiten in der Exaktheit der übermittelten Informationen, der Informationsaufnahme, der Interpretation und des Rekombinationsvorganges dar.

In der Regel wird aus dem so erhaltenen neuen Genotyp ein neuer Agent des gleichen Typs generiert, es wäre jedoch ebenfalls möglich, den Genotyp des Agenten durch den modifizierten zu ersetzen, wie Smith et al. (2000) es vorschlagen:

„An evolutionary analogy can be made without explicit consideration of "birth" and "death" for agents, since an agent perturbing itself can be viewed as producing a slightly mutated child, then allowing the child to replace the parent" (Smith et al. 2000, S. 3).

Wenn diese Variante nicht gewählt wird, dann kann das plötzliche Auftreten zusätzlicher Agenten, die erfolgreiche Agenten imitieren, mikroökonomisch als folgendes Phänomen verstanden werden: offensichtlicher Erfolg eines Unternehmens auf einem Markt führt zu neuen Unternehmen, die beim Markteintritt das beobachtbare erfolgreiche Handeln adaptieren

(Varian 1993). Im Sinne des Marktmodells stellt die neue Parameterkombination eine Marktstrategie dar, welche versucht, zwei vorhandene erfolgreiche Strategien zu kombinieren und damit einen besseren Nutzen zu generieren als jede der Elternstrategien alleine. Im Zusammenspiel des Erzeugens von Kombinationen erfolgreicher Strategien und des Herausnehmens erfolgloser Strategien ergibt sich eine Veränderung der Zusammensetzung der Population und damit auch ein ständig wechselnder Vergleichsmaßstab für die Agenten.

Dieses dynamische Vorgehen führt, obwohl dezentralisiert, zu einer den zentralen evolutionären Algorithmen vergleichbaren Optimierung des Gesamtsystems. Da explizit der Fitnessvergleichswert dem durchschnittlichen Gewinn entspricht, wird implizit die optimale Wohlfahrt als Summe der einzelnen Nutzenzuwächse maximiert. Durch die Mutationsphase entstehen neue Agenten, die unterschiedlich zu den bereits existenten Agenten sind und die die evolutionäre Chance nutzen können, das System zu ihrem Vorteil zu verändern. Auf diese Weise kann in Experimenten mit elektronischen Märkten die Auswirkung veränderter Produktionsfunktionen oder anderer, sonst als konstant betrachteter Parameter der Agenten oder der Marktstruktur, untersucht werden.

3.6 Zusammenfassung der Anforderungen an die Implementierung

Für das Szenario eines offenen, elektronischen Marktes, auf dem sich autonome Agenten unterschiedlicher Prinzipale treffen, ist das Paradigma des Multi-Agenten-Systems nach diesen Ausführungen wesentlich besser geeignet als dasjenige des Distributed Problem Solving. Anhand des Schichtenmodells lassen sich die folgenden Ergebnisse zur Implementation eines elektronischen Marktes mit der technischen Architektur des Multi-Agenten-Systems zusammenfassen:

- Auf der *technischen Ebene* sind die einzelnen Komponenten autonome Softwareobjekte (Agenten), die über ein offenes Netz miteinander kommunizieren und interagieren.
- Die *Kommunikation* zwischen den Agenten wird über Sprechakte abgewickelt. Ihre Implementation orientiert sich an verfügbaren Standards zur Kommunikation zwischen Agenten (z.B. ACL). Die Agenten stehen in direkter Kommunikation miteinander und benötigen keine Blackboard- oder Broadcast-Architektur. Sie verwenden direktes Message Passing oder Facilitatoren und greifen auf Verzeichnisdienste zu. Die versendeten Nachrichten stellen Sprechaktprimitive dar, deren Bedeutung auf der Kooperationschicht definiert ist.

- Die *Kooperation* zwischen den Agenten drückt sich in der Verhandlung über den Kauf oder Verkauf von Wirtschaftsgütern und dem Eingehen von Konzessionen aus. Derartige Transaktionen werden durchgeführt, wenn damit eine Nutzenerhöhung verbunden ist. Das Ziel der Agenten ist alleine die jeweilige individuelle Nutzenmaximierung, welche eine adaptive Strategie erfordert, die sich im Zeitablauf und je nach Umwelteinwirkungen über ein Lernverfahren verändert. Jeder Agent besitzt eine Strategie, die Aktionswahl und Entscheidungen bestimmt. Die Qualität der Entscheidungen bestimmt sich für den Agenten aus Rückmeldungen seiner Umwelt, die für ihn nicht prognostizierbar sind. Die Strategie wird durch ständige Rückkoppelung adaptiv unter Verwendung eines dezentralen evolutionären Algorithmus an eine sich verändernde Umwelt angepasst. In der Kooperationsschicht jedes Agenten wird ein allgemeines „Monotonic Concession“ Verhandlungsprotokoll implementiert, welches die Aufeinanderfolge von Sprechakten regelt. Käufer und Verkäufer tauschen Gebote aus und bewegen sich strategisch von beiden Seiten über das Eingehen von Konzessionen auf einen Kompromißpreis zu.
- Die *Koordination* zwischen den Agenten kommt schließlich durch häufige Interaktionen und das Austauschen von Preissignalen zustande. In der Koordinationsschicht muß von der individualistischen Eigennutzverfolgung der Agenten ausgegangen werden. Der Markt stellt ein Multi-Agenten-System dar, in dem die einzelnen Agenten das Gesamtverhalten des Systems ausschließlich durch ihre Interaktion und ihre Nutzenverfolgung beeinflussen. Eine Kooperation zwischen zwei Agenten wird nur dort begonnen, wo

beide Agenten einen eigenen Nutzenzuwachs prognostizieren. Dadurch werden die in den Güterpreisen gekapselten Informationen über vor- und nachgelagerte Wertschöpfungsstufen in dem verteilten System weitergegeben.

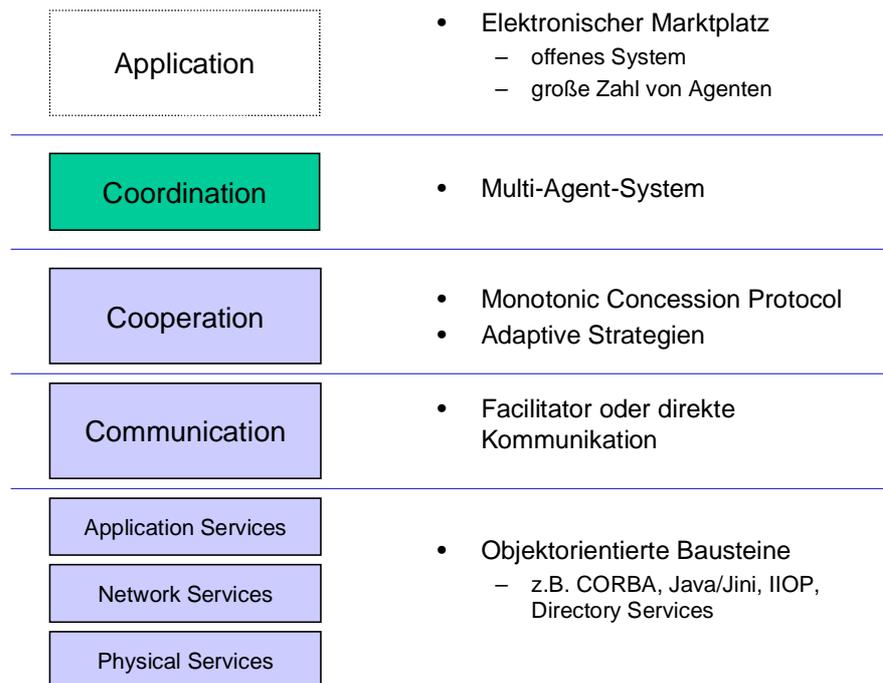


Abb. 16: Agentenbasierte Realisierung eines elektronischen Marktes

Damit sind die Anforderungen spezifiziert, die an die Realisierung eines agentenbasierten elektronischen Marktes gestellt werden können. Im folgenden Kapitel wird das Multi-Agenten-System AVALANCHE als beispielhafte Implementation in seinem Aufbau vorgestellt.

4 Realisierung marktlicher Koordination in einem MAS

In diesem Kapitel wird der Aufbau des experimentellen MAS AVALANCHE beschrieben. Die Beschreibung der Realisation wird begleitet von einer Darstellung der Gestaltungsoptionen und den Begründungen, warum auf den unterschiedlichen Ebenen des Schichtenmodells jeweils eine bestimmte Implementation realisiert wurde. Das *Agent-Based Value Chain Experiment* (AVALANCHE) stellt eine aus handelnden Agenten und Marktplätzen bestehende Software-Umgebung dar. Die Anzahl an Agenten und Marktplätzen ist konzeptionell und technisch nicht limitiert. In den durchgeführten Experimenten wurden 1 bis 5 Marktplätze aufgebaut, auf denen zwischen 50 bis 500 Agenten handelten. AVALANCHE wird seit 1997 an der Universität Freiburg entwickelt³⁰.

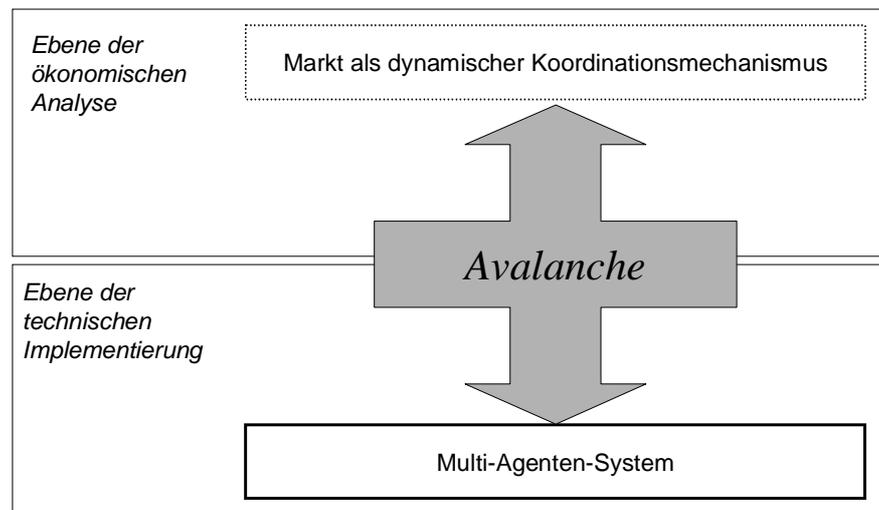


Abb. 17: AVALANCHE zwischen Markt und Multi-Agenten-System

Die Erkenntnisse der vorhergehenden Kapitel dienen als Grundlage, um mit AVALANCHE, wie in Abb. 17 schematisch dargestellt, eine Brücke zwischen der Ebene der ökonomischen Analyse und der Ebene der technischen Implementierung schlagen zu können.

³⁰ Es wurde seither in internationalen Publikationen und auf Konferenzen vorgestellt (z.B. Eymann et al. 1998a, 1998b, Eymann/Padovan 1999, Eymann/Kearney 2000).

Aus Kapitel 2 wird der dezentrale Ansatz zur marktlichen Koordination übernommen, der auf dem Katallaxieansatz von Hayek basiert und direkte Verhandlungen zwischen den Marktteilnehmern als Ausgangspunkt der Koordination versteht. Die Gestaltungsmöglichkeiten der VKI, insbesondere der Multi-Agenten-Systeme, die in Kapitel 3 beschrieben sind, werden wie in Abbildung 16 zusammengeführt.

Der Aufbau von AVALANCHE, der in diesem Kapitel 4 beschrieben wird, dient als exemplarische Implementation, die wesentliche allgemeine Elemente eines generellen **dezentralen Koordinationsmechanismus für Multi-Agenten-Systeme** vorwegnimmt. Die konkreten Klassen und Instanzen werden auf ein spezielles Modell der Koordination einer Wertschöpfungskette eingeschränkt. In Kapitel 5 zeigt AVALANCHE in seiner Eigenschaft als **Werkzeug zur dynamischen Untersuchung dezentraler Marktkoordination**, wie die Änderung bestimmter lokaler Parameter der Agentenstrategie die marktliche Koordination über das gesamte System hinweg beeinflusst.

In diesem Kapitel wird zuerst das Modell der Wertschöpfungskette vorgestellt, welche ein einfaches Anwendungsbeispiel nachbildet, in dem sich Software-Agenten durch Teilnahme an elektronischen Märkten koordinieren. In der Architektur von AVALANCHE erfordert diese Wertschöpfungskette die Definition von Agenten, Marktplätzen und einer Überwachungsinstanz für die Experimente (Kapitel 4.2). Der Aufbau dieser einzelnen Komponenten wird in den folgenden Abschnitten beschrieben, wobei ein Schwerpunkt auf der Gestaltung der Kooperationsebene der Agenten liegt (Kapitel 4.5.3).

4.1 Referenzszenario der Koordination einer Wertschöpfungskette

Das Szenario soll eine Wertschöpfungskette der Holzverarbeitenden Industrie modellhaft abbilden, in der Software-Agenten miteinander um Rohstoffe, Zwischen- und Endprodukte verhandeln. Idealtypisch besteht eine Wertschöpfungskette aus mehreren verbundenen organisatorischen Einheiten (Porter et al. 1985), die Inputgüter (Rohstoffe) beziehen und Outputgüter (Produkte) verkaufen. Diese Entitäten verarbeiten Rohstoffe unter Zuhilfenahme von Produktionsfaktoren und erhöhen damit den Wert der Produkte. Die Produkte dienen wiederum als Inputgüter für die nächste Stufe der Wertschöpfungskette, bis das Endprodukt den Konsumenten erreicht. Die Kette reicht von primitivsten Rohstoffen (Holz, Öl, Gas) bis zu wesentlich komplexeren Endprodukten (Fahrzeugen, Gebäuden). Die entstehende Struktur gleicht, über alle Produktbeziehungen gesehen, eher einem Netz als einer linearen Kette, jedoch kann jede Rohstoff-Endprodukt-Relation als linearer Zusammenhang beschrieben werden, so wie in der

folgenden Abbildung 18 gezeigt. Zwischen einem Rohstoff (Baum) und einem Endprodukt (Tisch) ergibt sich eine ununterbrochene Kette aufeinanderfolgender Produktionstechnologien. Die zugrunde gelegte Wertschöpfungskette kann abstrakt als ein von Menschen gebildetes komplexes System beschrieben werden, mit einer Anzahl an Entitäten (Organisationseinheiten) die mittels lokaler Regeln (einfache Regeln zum Kauf und Verkauf von Gütern) zur Kommunikation und Kooperation interagieren. Die Wertschöpfungskette stellt damit zuerst nicht mehr da als *"a collection of autonomous, problem-solving agents which interact when they have interdependencies"* (Jennings et al. 1996), eine hierarchielose Kollektion, in der keinerlei Unter- oder Überordnungen definiert werden.

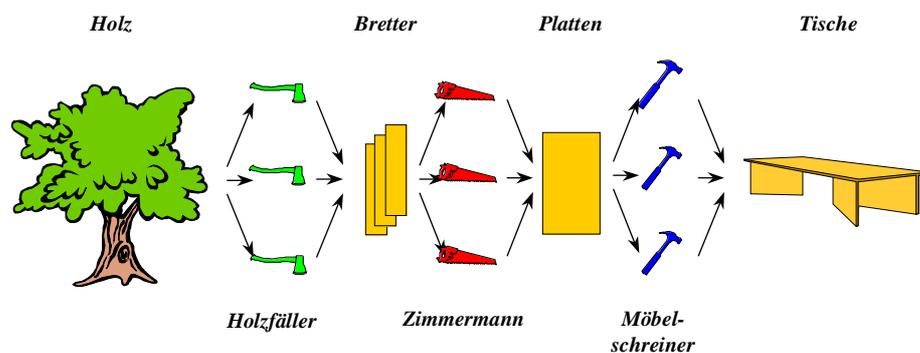


Abb. 18: Szenario der Wertschöpfungskette in AVALANCHE³¹

Die Koordination von Wertschöpfungsketten ist eine Domäne des *Supply Chain Management* (SCM), wobei im betriebswirtschaftlichen Kontext zwischen zentraler und dezentraler Koordination unterschieden wird (Hahn 2000, S. 16). Bei zentraler Koordination existiert eine Koordinationsinstanz mit Steuerungs- und Sanktionsgewalt gegenüber den Kettengliedern. Dezentrale Systeme ohne eine solche Instanz, besonders im unternehmensinternen SCM-Bereich der Produktionsplanung und -steuerung, beherrschen hingegen "tendenziell eine hohe Umweltdynamik" und sind "für die Abbildung der internen Wertschöpfungskette [...] die Grundvoraussetzung" (Wildemann 2000, S. 70). Das hier beschriebene und implementierte Szenario geht von einer strikt dezentralen Betrachtungs- und Steuerungsweise von Wertschöpfungsketten

³¹ Diese modelhafte Wertschöpfungskette zeigt bildhaft einen fiktiven Produktionsprozess der holzverarbeitenden Industrie. Es ist dem Autor bewußt, dass in der Realität Zimmerleute nicht Teil dieses Wertschöpfungsprozesses sind und auch die Bezeichnung der Güter und Aufgabenträger keinen Anspruch auf Realitätstreue erhebt.

aus. Die Agenten folgen ausschließlich ihren lokalen Handelsregeln, ohne dass eine zentrale Koordinationsinstanz auf sie einwirkt. Unterscheiden lassen sich die Agenten nach ihren Produktionstechnologien. Ein Typ von Agenten fällt Bäume (*Holzfäller*, engl. *Lumberjacks*), während der in der Kette folgende Typ die Bäume in Bretter zersägt (*Zimmerleute*, engl. *Carpenter*). Aus Brettern werden schließlich von *Möbelschreibern* (engl. *Cabinetmaker*) Tische produziert. Die Tische werden als Endprodukte konsumiert und stehen im System anschließend nicht mehr als handelbare Güter zur Verfügung. Die unterschiedlichen Agenten werden durch unterschiedliche Symbole in der Abbildung 18 dargestellt.

Im Zeitablauf ist es wahrscheinlich, dass es nicht nur zur Bildung einer einzelnen Wertschöpfungskette kommt, sondern dass sich mehrere parallele, aber auch verflochtene Ketten bilden. Durch die Produktionstechnologien ist der jeweilige Typ des Abnehmers oder Lieferanten für einen Agenten festgelegt. Unter den einzelnen Agenten eines Typs besteht hingegen Konkurrenz. Zwischen welchen Agenten genau eine Transaktion durchgeführt wird, ist das Ergebnis des Koordinationsmechanismus und nicht von vornherein festgelegt. Vor diesem Hintergrund werden die einzelnen Glieder (Agenten) der Kette als "low-level primitives" in Sinne der Complex Adaptive Systems betrachtet (Holland et al. 1991). Die Entstehung eines Endproduktes (Tisch) aus Rohstoffen (Bäumen) erfolgt als emergente Eigenschaft im Zeitablauf der Aushandlungsprozesse. Weder kennt der *Holzfäller* das endgültige Produkt, das aus seinem gefällten Baum entsteht, noch weiß der *Möbelschreiber*, von welchem Baum seine Bretter stammen. Trotzdem produzieren alle Agenten zusammen einen hoffentlich qualitativ hochwertigen Tisch – lediglich durch bilaterale, für alle nutzenverbessernde Kooperation, die sequentiell den Stufen der Wertschöpfungskette folgt. Da das Szenario nur einen Ausschnitt der Wirklichkeit darstellt, müssen die Eingangspreise und Eingangsmengen, durch die die Rohstoffe (Bäume) zu Beginn der Wertschöpfungskette in die Marktumgebung eingeführt werden, durch den Experimentator festgelegt werden. Gleiches gilt umgekehrt für die Abnahmepreise und -mengen der Endprodukte (Tische).

Zur Verdeutlichung wird im folgenden ein gesamter Verhandlungsablauf mit zwei Agenten, einem Verzeichnisdienst und den zwischen den Agenten kommunizierten Nachrichten beispielhaft dargestellt.

Die beiden Agenten Anna (Lumberjack) und Benno (Carpenter) befinden sich auf einem ansonsten als leer vorgestellten Markt. Anna hat ein Brett für € 20 zu verkaufen, Benno möchte ein Brett für € 20 kaufen.

Anna schickt als erste Aktion eine Nachricht an den Verzeichnisdienst, in der ihr Angebot (Brett) und ihre ID (Anna) eingetragen sind, ohne jedoch einen Preis zu nennen.

Der Verzeichnisdienst listet ihre ID dann in seiner Rubrik Brett/Verkäufe auf.

Benno schickt danach eine Nachricht an den Verzeichnisdienst, welche Einträge in der Rubrik Brett/Verkäufe vorhanden sind. Der Verzeichnisdienst übergibt ihm einen Vektor mit allen IDs in dieser Rubrik. In diesem Fall hat der Vektor nur einen Eintrag, die ID von Anna. Benno schickt eine Nachricht ("cfp") an Anna, in der er sie um ihr Gebot bittet. Anna antwortet mit einer Nachricht ("propose-offer"), in der sie € 20 als Stückpreis für ein Brett fordert.

Da dieses Gebot dem internen Reservepreis von Benno bereits entspricht, ist es nicht notwendig, weiter in das in Abschnitt 4.5.8 im Detail beschriebene Verhandlungsverfahren einzusteigen. Benno wird Anna eine Nachricht zurücksenden, in der er ihr Gebot akzeptiert ("accept-offer"). Anna bucht die Produkteinheiten von ihrem Lagerbestand ab, den Kaufpreis ihrem Geldkonto zu und schickt eine Bestätigungsmitteilung ("commit"). Nach deren Erhalt führt auch Benno die entsprechenden Waren- und Geldebuchungen durch.

In der nächsten Phase wird nur Benno betrachtet. Da Benno keine fertigen Endprodukte am Lager hat, jedoch genügend Rohstoffe, um diese herzustellen, wird er eine Produktionsphase einleiten. Nachdem diese beendet ist, wird er seine ID und sein Angebot (Platte) an den Verzeichnisdienst schicken, um für potentielle Abnehmer erreichbar zu sein.

Im folgenden soll zuerst der technische Aufbau der Agenten gezeigt werden, anschließend derjenige der relevanten Umgebung, die ausser den Agenten vor allem aus der Marktplattform besteht. Daran schließt sich die detaillierte Beschreibung des dynamischen Ablaufs der marktlichen Koordination, dargestellt anhand des Schichtenmodells der Telematik, an.

4.2 Architektur eines agentenbasierten Marktes

Die Struktur eines agentenbasierten Marktes kann unterteilt werden in den organisatorischen Aufbau, der die einzelnen Objekte, ihre Methoden und Variablen und ihre Beziehungen definiert, sowie den dynamischen Ablauf, der sich in die Kommunikation und die Kooperation zwischen den Agenten einteilen lässt. Explizite Koordinationsdienste der Marktplätze oder des Gesamtsystems, wie z.B. der Verzeichnisdienst oder Analysewerkzeuge, werden gesondert dargestellt. Insbesondere wird bei diesen darauf eingegangen werden müssen, inwiefern sie für

die marktliche Koordinationsleistung notwendig sind, oder lediglich der Beobachtung und Auswertung dienen, ohne am Ergebnis der Koordination etwas zu ändern. Die graphische Darstellung erfolgt in der auf der Beschreibungssprache UML (Unified Modelling Language) (Burkhardt 1999) basierenden Erweiterung *Agent UML* (AUML, Odell et al. 2000).

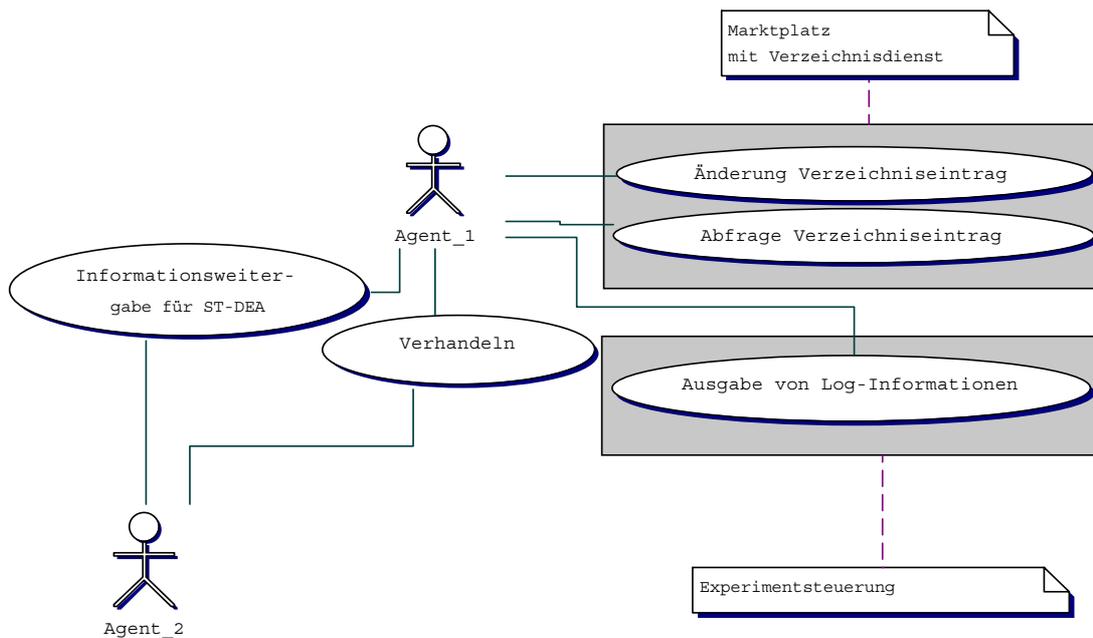


Abb. 19: UML-Anwendungsfalldiagramm von Avalanche

In Abb. 19 lassen sich 3 Komponenten unterscheiden: die verhandelnden *Agenten*, die in AVALANCHE in der Klasse *AvTradeAgent* implementiert werden, der *Marktplatz* selbst (*AvLocationAgent*) und die *Experimentsteuerung* (*AvInfoServer*)³².

- Kernstück eines agentenbasierten Marktes sind die **Agenten**. In den Agenten sind die Handelsprotokolle und die Kommunikationsfähigkeiten definiert. Jeder Agent kann mit anderen Agenten direkt und bilateral über das Versenden von Nachrichten kommunizieren, ebenso wie mit anderen Objekten des Marktplatzes, auf dem sich der Agent zur Zeit befindet.

³² Um die Zuordnung der Begriffe zu erleichtern, wird in diesem und dem folgenden Abschnitt folgende Zeichenformatierung eingeführt. Feststehende Begriffe, die einen bestimmten Objekttyp oder Methode agentenbasierter Märkte beschreiben, werden in *kursiv* dargestellt, z.B. *Holzfüller*. Begriffe, die für die Implementation von Avalanche spezifisch sind, werden dagegen in nichtproportionaler Schriftart dargestellt, z.B. *AvTradeAgent*.

- Auf jedem **Marktplatz** muß es mindestens ein weiteres Objekt geben, welches die Kommunikation der Agenten unterstützt. Um zu wissen, welche Agenten mit welchen Methoden ansprechbar sind, wird ein *Verzeichnisdienst* (engl. *directory service*) benötigt. Dadurch erhalten die Agenten eine technische Referenz auf ein anderes Objekt, mit dem sie anschließend einen Kommunikationskanal aufbauen können. Der Verzeichnisdienst wird von den Software-Agenten benutzt, um sich auf dem Marktplatz an- und abzumelden, so dass andere Agenten ihnen Gebote senden können. Der Verzeichnisdienst beeinflusst weder die Strategie der einzelnen Software-Agenten noch Art und Umfang ihrer Kommunikation. Die spezielle Implementation hat keinen Einfluß auf das Ergebnis der Koordination. Weitere Objekte, die z.B. Auktionsdienste oder Notarfunktionen anbieten, sind möglich, aber nicht Bestandteil dieser Arbeit.
- Aufgabe der **Experimentsteuerung** ist schließlich die Zeitnahme, das Führen von Protokolldateien, die Bildschirmanzeige sowie das Starten und Beenden der Experimente. Diese experimenttechnischen Funktionen sollten die Koordinationsergebnisse ebenfalls nicht beeinflussen können.

Die Implementation von AVALANCHE wurde in der objektorientierten Programmiersprache JAVA³³ (Version 1.3) durchgeführt. Neben den Diensten von JAVA selbst wird eine *Object Request Broker* (ORB)-Klassenbibliothek namens VOYAGER³⁴ verwendet (Objectspace 1997), welche die Erstellung von Agenten unterstützt. Die in AVALANCHE eingesetzten Agenten sind vollständig voneinander unabhängige Java-Prozesse.

4.3 Strukturierung der Datentypen

Bei der objekt-orientierten Programmierung (OOP) werden Objekte ganzheitlich beschrieben, d.h. die Festlegung der Datenstrukturen und der mit den Daten ausgeführten Aktionen erfolgt in einem. Die wichtigsten Vorteile der objekt-orientierten Programmierung sind: die Aufspaltung von komplexen Software-Systemen in kleine, einfache, in sich geschlossene Einzelteile, einfache und klar verständliche Schnittstellen zwischen den einzelnen Komponenten, weitgehende Vermeidung von Programmierfehlern beim Zusammenspiel zwischen den Komponenten und geringer Programmieraufwand durch die Wiederverwendung von Elementen. Der Begriff der Klasse (*class*) beschreibt eine allgemeine Kategorie und stellt eine abstrakte Realisation eines Datentyps dar, z.B. ein Verhandlungsgebot. Als Objekt (*object*) bzw. Instanz

33 siehe Glossareintrag „Java“.

34 siehe Glossareintrag „Voyager“.

(*instance*) bezeichnet man eine konkrete Ausprägung eines Objekts, also ein Stück aus der Menge der Objekte dieser Klasse (Partl 2000).

Um die Komplexität des Gesamtsystems zu vereinfachen, werden die Klassen in Pakete (sogenannte *Packages*) unterteilt, die in sich abgeschlossen sind und einen gemeinsamen Aspekt der Gesamtimplementation behandeln. In JAVA versteht man unter einem *Package* eine Menge von zusammengehörenden Klassen, ähnlich wie bei einer Programmbibliothek. Die Pakete von AVALANCHE sind mit den in ihnen definierten Klassen grafisch im Anhang dargestellt.

- Das Paket `de.unifreiburg.iig.avalanche.agents` gruppiert die im Rahmen dieser Arbeit wichtigsten Klassen. Die einzelnen handelnden Wirtschaftssubjekte, die zur Erzielung einer Koordination miteinander kommunizieren und kooperieren, werden durch die in diesem Paket zusammengefassten Klassen formal beschrieben.
- Das Paket `de.unifreiburg.iig.avalanche.evolution` umfaßt Klassen, die ein spezielles Lernverfahren der Agenten im Sinne evolutionärer Algorithmen realisieren. Im wesentlichen handelt es sich um die Modifikation eines Konzeptes von Smith/Taylor (1998).
- Die Agenten kooperieren und kommunizieren miteinander über die Klassen des Pakets `de.unifreiburg.iig.avalanche.negotiation`. Diese transportieren die Information und geben durch ihren Aufbau die Interaktionsmöglichkeiten vor.
- Das Paket `de.unifreiburg.iig.avalanche.location` enthält die Klassen zur Darstellung des Marktplatzes. Diese können das Koordinationsergebnis durch Anbieten von Diensten beeinflussen. In der gegenwärtigen Implementation sind nur minimale Dienste, z.B. die Schnittstelle zum Verzeichnisdienst, vorhanden. Im folgenden werden die Begriffe "Lokation" und "Marktplatz" synonym verwendet.
- Im Paket `de.unifreiburg.iig.avalanche.generator` werden solche Klassen zusammengefasst, die für den Start des Gesamtsystems, einzelner Marktplätze oder einzelner Agenten benötigt werden. Dieses und die folgenden beiden Pakete gehören logisch zur Experimentsteuerung. Auf den Betrieb der Marktplätze oder die Koordinationsleistung haben diese Klassen keinen Einfluß und ihre Implementation ist in hohem Maße durch die speziellen Gegebenheiten von AVALANCHE beeinflusst.
- Das Paket `de.unifreiburg.iig.avalanche.infoserver` gruppiert diejenigen Klassen, welche für die Beobachtung und Analyse der dynamischen Entwicklung des Gesamtsystems genutzt werden. Die meisten Klassen dienen der graphischen oder

tabellarischen Darstellung der gesammelten Daten, oder dem Speichern der Logdateien für die Ex-Post-Analyse. Sie haben ebenfalls keinen Einfluss auf die marktliche Koordination.

- Das Paket `de.unifreiburg.iig.avalanche.avanabot` (AVALANCHE Analyse Bot) ist für die Ex-Post-Analyse der gesammelten Daten eines Experimentierlaufes bestimmt. Seine Ausgestaltung ist nicht Teil dieser Arbeit. Da es mit den durch die Experimentsteuerung generierten Logdateien arbeitet, ist es während des Marktbetriebs nicht aktiv und hat keinen Einfluß auf das Koordinationsergebnis.

4.4 Datentypen und Methoden der Agentenklassen

Im Paket `de.unifreiburg.iig.avalanche.agents` sind die wesentlichen Klassen der Agenten-Implementation zusammengefasst. Ausgehend von der zentralen Klasse `AvAgent` werden weitere Klassen mit speziellen Fähigkeiten abgeleitet³⁵. Diese werden im folgenden beschrieben. Eine begleitende grafische Darstellung [findet sich in Abb. 50](#).

Die Klasse `AvAgent` dient der Sammlung verbindender Variablen und Methoden aller abgeleiteten Klassen, um diese systemunabhängig (z.B. von `VOYAGER`) zu gestalten. Jeder `AvAgent` zeichnet sich durch einen eindeutigen Namen (`agentName`) sowie eine Typenzuordnung (`agentType`) aus. Der aktuelle Marktplatz, auf der sich der Agent befindet (`location`), sowie der Zeiger auf die Experimentsteuerung (`infoServer`), sind ebenfalls eindeutig. Jeder Agent kann mit anderen Agenten über Nachrichten kommunizieren, ebenso wie mit dem Marktplatz, auf dem sich der Agent zur Zeit befindet.

Die Implementation der Kommunikation zwischen den Agenten wurde einem Beispiel von Bigus/Bigus (1999, S. 193ff.) entnommen. Jeder Agent erzeugt Nachrichtenereignisse (*Events*), die bei den Empfängeragenten (*EventListener*) eine Reaktion des aufgerufenen Agenten praktisch erzwingen. Die benachrichtigten Agenten können auf dieses Ereignis reagieren. Die Ereignisse selbst bestehen aus einem `AvAgentMessage`-Objekt, welches einen Sprechakt codiert und die Nachricht überträgt. Durch die auf diese Weise erzwungene Annahme der Nachricht ist keine Entscheidung über eine Verarbeitung im Sinne des Senders erfolgt; die Nachricht kann abgelehnt werden und die Autonomie des Agenten bleibt gewahrt. Andererseits wird die Zustellung einer Nachricht durch dieses Vorgehen gewährleistet.

35 Die Klassen `IAvAgent`, `IAvTradeAgent`, `IAvConsumerAgent` und `IAvProducerAgent` sind spezielle Klassen von `Voyager` und besitzen keine eigene Funktionalität.

In der Klasse `AvAgent` finden ansonsten nur wenige Berechnungsvorgänge statt. Über die Methoden `generateRandom()` können Zufallszahlen zwischen einer spezifizierten unteren und oberen Grenze erzeugt werden. Die Methoden `RandomNumberIsHigherThan()` bzw. `RandomNumberIsLowerThan()` greifen auf `generateRandom()` zurück. Sie geben einen binären Wert aus, der aussagt, ob eine zufällig gezogene Zahl größer oder kleiner als ein übergebener Wert aus dem gleichen [Zahlenraum](#) ist. Diese Methoden werden später zur Beeinflussung der Handlungsstrategie der abgeleiteten Klasse `AvTradeAgent` genutzt.

4.5 Datentypen und Methoden der handelnden Agenten

Die zentrale Klasse der gesamten AVALANCHE-Implementation ist `AvTradeAgent`. Sie ist aus `AvAgent` abgeleitet und repräsentiert die handelnden Wirtschaftssubjekte.

Jeder `AvTradeAgent` wird mit einem `AvTradeAgentProperties`-Objekt initialisiert, welches charakterisierende Variablen für die Stammdaten des Agenten enthält, z.B. den Namen und Typ, die Art der Input- und Outputfaktoren sowie diejenigen Parameterwerte, die das strategische Verhalten des Agenten bestimmen. Die `AvTradeAgent`-Klasse beschreibt die abstrakten Handlungsmöglichkeiten und definiert die Methoden, während in den initialisierenden `AvTradeAgentProperties` die Variablen als individuelle Eigenschaften jedes Agenten kodiert sind, die im konkreten Fall zu durchaus unterschiedlichen Aktionen führen können.

Die Methoden der Klasse `AvTradeAgent` lassen sich in 3 große Gruppen einteilen:

1. Die Einkaufs- und Verkaufstätigkeiten der Agenten benötigen Methoden der Verhandlung, der Kommunikation sowie zur Zielbildung bzw. -veränderung. Die Methode `work()` stellt die wesentliche Steuerung dar und wird im folgenden genauer beschrieben. Aus dieser Methode heraus werden jeweils die Methoden `trySelling()`, `tryProduction()` und `tryBuying()` aufgerufen, die durch Senden von Nachrichten versuchen, eine Verhandlung zu initiieren. Die Methode `processMessage()` verarbeitet eingehende Nachrichten, von denen einige einfache Aktionen anstossen (wie Zu- oder Abbuchung von Gütern), andere das interne Modell verändern und z.B. eine Entscheidung über Art und Umfang der Verhandlungsführung erfordern. Die Entscheidungen werden in der Methode `negotiate()` gesammelt bearbeitet. Durch diese Trennung ist es möglich, weitere Klassen aus `AvTradeAgent` abzuleiten, die ein anderes Verhandlungsverhalten aufweisen, indem `negotiate()` überschrieben wird.
2. Zweitens benötigen die Agenten einfachste buchhalterische Methoden zur Beobachtung und Steuerung von Geld- und Warenvermögen. Die Warenbuchhaltung wird durch die

Methoden `addItemToInventory()`, `removeItemFromInventory()` sowie die Abfragemethode `amountOfItemInInventory()` realisiert. Die Finanzbuchhaltung berechnet über die Methode `capital()` das Eigenkapital des Agenten als Summe aus Bargeld und bewerteten Lagergütern, führt über die Methode `payCostsOfLife()` Gebühren an den Marktplatz ab und aktualisiert im Sinne eines Controlling die aktuellen Gewinn- (`updateProfit()`) und Umsatzdaten (`updateSales()`).

3. Zuletzt werden noch Methoden benötigt, die den evolutionären Algorithmus STDEA implementieren. Im `AvTradeAgent` werden in der Auswahlphase die erhaltenen *Plumages* mittels der Methode `estimateFitness()` untersucht und durch `getMate()` der beste *Genotyp* ausgewählt. Die eigenen *Plumages* werden hingegen mittels der Methode `createPlumage()` erzeugt und nach einer erfolgreichen Transaktion an die anderen Agenten verschickt.

Anhand der Abbildung 20 soll die sogenannte Hauptschleife des Agenten in der Methode `work()` dargestellt werden. Diese Endlosschleife wird bei der Erzeugung des Agenten initialisiert und erst beim "Tod" des Agenten wieder verlassen.

Im dargestellten Aktivitätendiagramm lassen sich verschiedene Phasen unterscheiden, die durch Konditionalentscheidungen miteinander verbunden sind und in der dargestellten Reihenfolge durchlaufen werden. Die Rangfolge der Phasen wurde anhand der Nähe zur Möglichkeit der Einkommensgenerierung gebildet. In diesem Sinne ist die vordringlichste Aufgabe der Verkauf von Produkten durch den Agenten. Nachrangig zum Verkauf ist die Produktion von Gütern angelegt, noch geringer ist der Einkauf von Rohstoffen, da diese Phase gleichzeitig einen Abfluß an liquiden Mitteln darstellt. Im Fall, dass keine dieser Phasen erfolgreich abgeschlossen werden kann, wird der Agent einen Wechsel des Marktplatzes in Betracht ziehen, in der *ceteris paribus* Annahme, dass sich auch im nächsten Schleifendurchlauf keine Änderung an der gegenwärtigen ergebnislosen Situation ergeben wird.

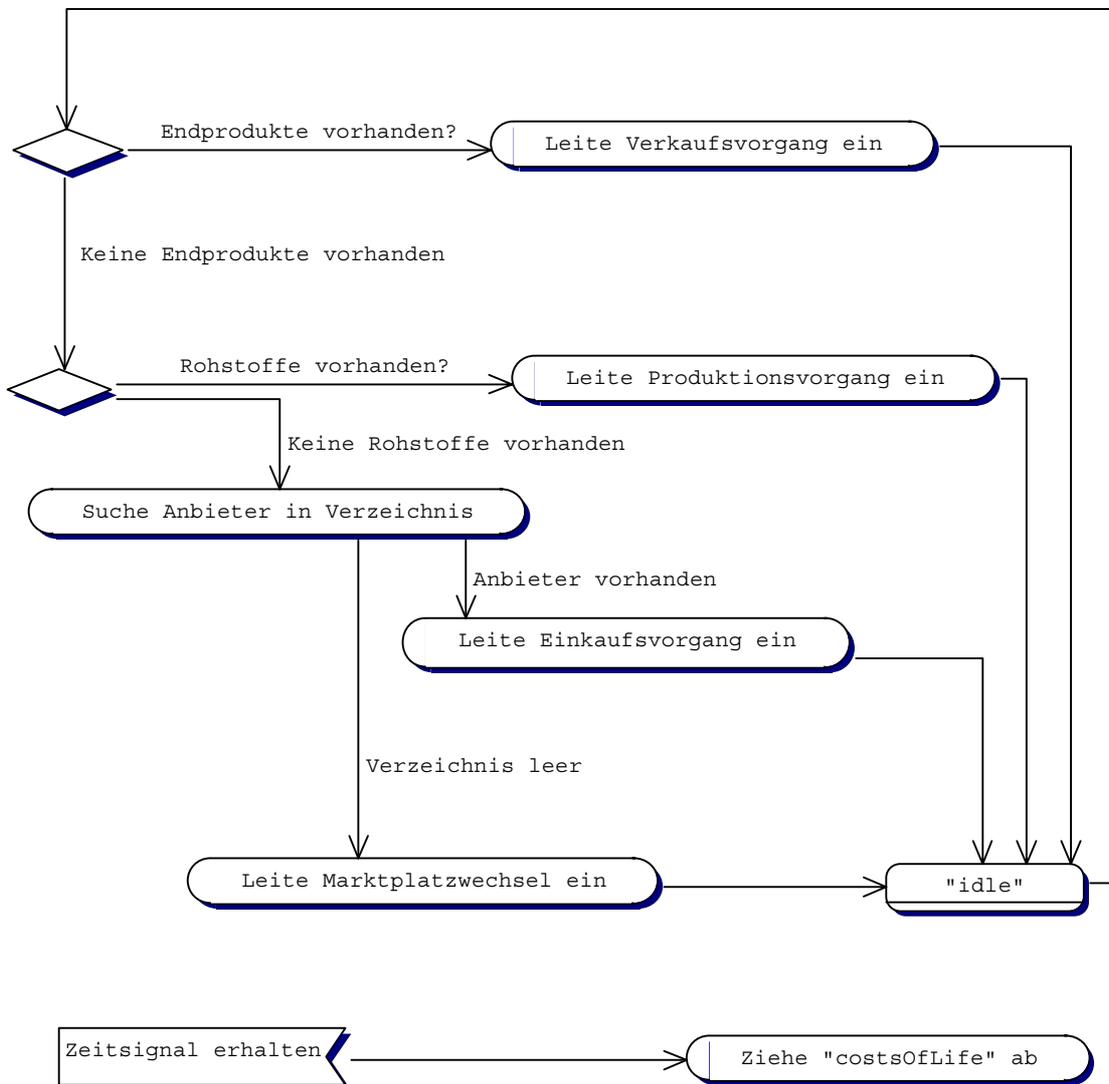


Abb. 20: UML-Aktivitätsdiagramm für AvTradeAgent

Jeder AvTradeAgent versucht, sein Ziel der Gewinnmaximierung zu verfolgen und die folgenden Aktionen in der in Abbildung 20 gezeigten Reihenfolge durchzuführen.

Falls der Agent ein oder mehrere Endprodukte besitzt, führt er eine **Angebotsphase** aus, um seine Endprodukte (`outputFactors`) zu verkaufen. Durch diese Tätigkeit erhält der Agent Geld, welches ihm einerseits das weitere Überleben und andererseits den Einkauf von Rohstoffen ermöglicht. Jeder Handelsplatz besitzt einen Verzeichnisdienst, in dem ein Anbieter seine eigene Identität (als Referenzadresse) und die Art des angebotenen Gutes einträgt. Der Angebotspreis wird nicht mitgeteilt (sonst hätte das Verzeichnis gleichzeitig Katalogfunktion),

sondern wird durch den Anbieter erst auf direkte Anfrage eines Nachfragers errechnet und nur diesem Nachfrager mitgeteilt. Dadurch wird vordergründig die Möglichkeit ausgeschaltet, dass sich Anbieter an bereits bekannten und veröffentlichten Angebotspreisen orientieren, so dass eine wesentliche Grundlage für Preiskriege (vgl. Kephart et al. 1998) nicht existiert.

Der aktuelle Angebotspreis für das Endprodukt des Agenten wird intern gespeichert und ist weder für den Marktplatzbetreiber noch für andere Agenten direkt einsehbar. Seine Berechnung orientiert sich zum einen am Kostenpreis als Mindestverkaufspreis. Der Kostenpreis errechnet sich aus dem Einkaufspreis sowie den Produktionskosten. Zum anderen berechnet der Agent einen spekulativen Preis auf Grundlage der letzten von ihm erzielten Verkaufspreise für das gleiche Gut. Auf die höhere beider Alternativen wird eine Gewinnspanne aufgeschlagen. Findet der Anbieter einen Nachfrager, der seine Produkte zum geforderten Preis abnimmt, wird die Verkaufstransaktion durchgeführt. Falls kein Nachfrager auftritt, wird jeweils nach einer voreingestellten Zeit eine unilaterale Konzession eingegangen, d.h. der Verkaufspreis zu Lasten der Gewinnspanne gesenkt.

Besitzt der Agent keine Endprodukte, kann die Angebotsphase nicht ausgeführt werden. In diesem Fall wird überprüft, ob genügend Rohstoffe vorhanden sind, um Endprodukte herzustellen. Erst durch die Produktion wird ein Verkauf möglich; eine andere Alternative, mit Rohstoffen Geld zu verdienen, wäre spekulatives An- und Verkaufen, was in AVALANCHE nicht implementiert ist. In der simulierten **Produktionsphase** werden entsprechend der Technologiefunktion Rohstoffeinheiten vom Lager ab- und Fertigprodukteinheiten zugebucht. Außerdem wird die in der Angebotsphase verwendete Kostenfunktion berechnet. In einer realen Implementation würde ein Agent ständig seitens seines Prinzipals angezeigt bekommen, ob Endprodukte als Ergebnis des realen Produktionsprozesses verfügbar sind.

Falls nicht genügend Rohstoffe vorhanden sind, wird der Agent als **Nachfrager** für diese Güter auftreten. Ohne Rohstoffe gibt es wenige Aktionen, die der Agent ausführen kann. Gelingt es dem Agenten nicht, über einen längeren Zeitraum Rohstoffe zu kaufen, kann sein Vermögen auf Null sinken und der Agent dadurch gezwungen werden, aus dem System zu verschwinden. Um einen Anbieter zu finden, orientiert er sich am Verzeichnisdienst des lokalen Handelsplatzes. Derzeit wählt er aus den möglichen Anbietern einen zufällig aus. Mit diesem beginnt er einen Verhandlungsprozeß, der nicht notwendigerweise zu einem Abschluß führen muß. Ein zu hohes Startgebot des Partners, mangelnde Kompromißbereitschaft oder eine zu lange Dauer des Verhandlungsprozesses läßt ihn eine Verhandlung u.U. schnell abbrechen. Durch die häufige Aufnahme und Abbruch von Verhandlungen mit unterschiedlichen Anbietern verschafft sich der Nachfrager einen Marktüberblick, der ihn seine Gebote so setzen läßt, dass eine Verhandlung zur Transaktion führt.

Schließlich zahlt der Agent am Ende der Schleife über die Methode `payCostsOfLife()` einen prozentualen Anteil seines Eigenkapitals an den Marktplatz. Dieser Aspekt des Experiments spiegelt eine Realisierung wider, bei der der Marktplatz für die Inanspruchnahme von Diensten (z.B. Verzeichnisdienst) und Ressourcen (z.B. CPU, Speicherplatz, Bandbreite) von den ihn nutzenden Agenten bezahlt wird. Zwei Alternativen sind vorstellbar: zum einen kann der Marktplatz einen prozentualen Anteil an den Transaktionen als Gebühr einbehalten, zum anderen kann er eine nach Leistungsfähigkeit gestaffelte konstante Benutzungsgebühr fordern, die nicht von den getätigten Transaktionen abhängt. In dieser Implementation wurde die zweite Variante gewählt, da zum einen die Schaffung eines Marktüberblicks bereits Ressourcen in Anspruch nimmt und zum anderen eine Strategie des Nichtstuns nicht als Meßlatte für den Erfolg der Verhandlungsstrategien in den Agenten dienen sollte. Falls der Agent das verlangte Geld nicht zahlen kann, muß er den Markt verlassen (und wird aus dem Experiment herausgenommen). Durch den direkten Aufruf der Methode `payCostsOfLife()` wird die Autonomie der Agenten stark verletzt. Konsistenter wäre eine nachrichtenbasierte Implementation mit einem FIPA-request-Protokoll (FIPA 1999, S. 62). Am Effekt "wer nicht zahlen kann, muß gehen" würde diese Variante nichts ändern.

Ist keines der Ziele Verkaufen, Produzieren oder Kaufen auf dem aktuellen Marktplatz erreichbar, versucht der Agent, auf einen anderen **Marktplatz** zu **wechseln**. Er hofft, dort bessere Marktbedingungen vorzufinden, um eine der o.a. Aktionen durchführen zu können.

Diese Hauptschleife ist für alle handelnden Agenten gleich, unabhängig von ihrer Position innerhalb der Wertschöpfungskette. Jeweils am Anfang und am Ende der Kette gibt es in AVALANCHE aus Simulationsgründen besondere Wirtschaftssubjekte, die entweder nur den Verkauf oder den Einkauf eines Agenten implementierten.

Die Klasse `AvConsumerAgent` überschreibt einige Methoden des `AvTradeAgent`, um statt eines kaufenden und verkaufenden Wirtschaftssubjektes eine reine Konsumentenseite abzubilden. Ein *Konsument* kann keine Endprodukte verkaufen und braucht auch nichts zu produzieren. Außerdem besitzt er immer genügend Geld (`cash`), kann also beliebig viele Güter einkaufen. Die *Konsumenten* simulieren allerdings mit einer Preisobergrenze einen von außerhalb der Softwareumgebung herrührenden `Marktpreis`, der nicht durch die Dynamik des Systems beeinflussbar ist. Durch eine zu den `AvTradeAgents` identische Implementation einer Einkaufsstrategie üben sie einen Preisdruck auf die vorhergehende Wertschöpfungsstufe (die *Möbelschreiner*) aus.

Die Klasse `AvProducerAgent` überschreibt einige Methoden des `AvTradeAgent`, um einen reinen Produzenten zu repräsentieren. Dieser benötigt keine Inputfaktoren zu Produktion und bildet daher keine Einkaufsseite ab. Hingegen ist die Absatzstrategie identisch zu den

AvTradeAgents, so dass der Produzent einen ständigen Preisdruck auf die nächsthöhere Wertschöpfungsstufe (die *Holzfäller*) ausübt. Um die Ressourcen für die Agenten in der Wertschöpfungskette knapp zu halten und eine Aussage über die Qualität der Ressourcenallokation in AVALANCHE zu ermöglichen, produziert der AvProducerAgent Güter in gleichen Zeitabständen, die durch einen Zähler (productionTimeCounter) determiniert werden. Dieser Zähler wird durch die Methode payCostsOfLife() hochgezählt, die durch die Handelsplätze bei allen Agenten angestossen wird.

4.5.1 Datentypen des individuellen Verhaltens

Die Individualisierung des jeweiligen Agentenobjekt leistet die Klasse AvTradeAgentProperties. Sie bestimmt die Heterogenität in der Population, die durch die Freiheitsgrade in der Initialisierung der Parameterwerte dieser Klasse gesteuert werden kann. Sie besteht aus einer Menge an individualisierenden Variablen, besitzt aber keine eigenen Methoden. Neben dem Namen (agentName) und dem Typ (agentType) des Agenten handelt es sich hierbei vor allem um Variablen, die das wirtschaftliche Handeln und das Lernen beeinflussen.

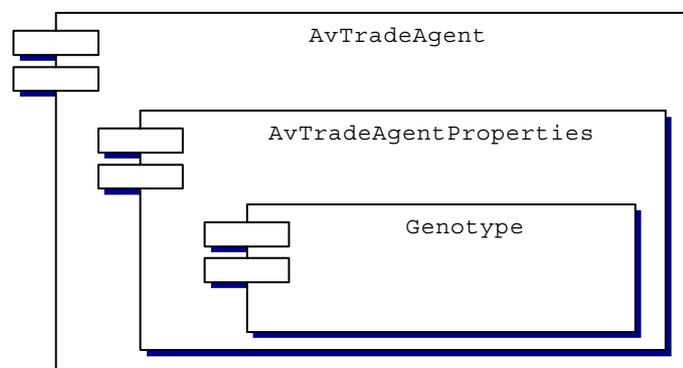


Abb. 21: UML-Komponentendiagramm für AvTradeAgent

Eine ökonomische Individualisierung kann z.B. über unterschiedliche Bargeldbestände (cash bzw. initialCash), unterschiedliche Marktplatzgebühren bzw. Transaktionskosten (costs of life), die jeweiligen Maximal- und Minimalpreise für Einkauf und Verkauf, die Art der Input- und Outputfaktoren selbst oder die Warenausstattung (initialStock) geschehen. In den in Kapitel 5 dargestellten Experimentläufen sind diese Eigenschaften über alle gestarteten Agenten gleich.

Abbildung 21 zeigt, dass die Klasse der `AvTradeAgentProperties` wiederum eine Klasse enthält, den `Genotype`. Der `Genotype` ist in seiner Funktion von Smith/Taylor (1998) entnommen und besteht aus lediglich sechs Variablen, welche die Strategie des Agenten wesentlich determinieren. Die Experimente, die in Kapitel 5 gezeigt werden, beschäftigen sich ausschließlich mit Veränderungen in den Werten einzelner Parameter bzw. unterschiedlichen Parameterkombinationen.

4.5.2 Kommunikation zwischen den Agenten

Die zwischen den einzelnen Agentenklassen verschickten Nachrichten lassen sich in vier Protokolle unterteilen, die unterschiedliche semantische Bedeutung haben:

- Das Protokoll "*trade*" beschreibt die Aufeinanderfolge von Sprechakten (Searle 1970), die der Durchführung einer Verhandlung zwischen zwei Agenten dienen. Die Kommunikationsprimitive implementieren den Austausch von Geboten und Gegengeboten, sowie eine bestätigte Annahme eines Transaktionsvorschlags.
- Das Protokoll "*recommend-seller*" dient der Kommunikation zwischen Agent und Marktplatz und entspricht im wesentlichen einer Verzeichnisabfrage bezüglich der aktiven Anbieter eines bestimmten Gutes. Der Marktplatz wird von einem Käufer um eine Liste der Anbieter eines Gutes gefragt, oder vom Verkäufer um eine Liste der Nachfrager nach einem Gut.
- Das Protokoll "*move*" dient ebenfalls der Kommunikation zwischen Agent und Lokation. Der Marktplatz wird von einem Agenten um eine Referenz auf einen benachbarten Markt gebeten, zu dem sich der Agent autonom bewegen wird. Da die in dieser Arbeit beschriebenen Experimente jeweils nur auf einem Marktplatz ausgeführt wurden, wird auf die Beschreibung der aus der Mobilität herrührenden Konsequenzen an dieser Stelle verzichtet.
- Das Protokoll "*reproduction*" schließlich umfaßt solche Nachrichten, die zur Implementation des verwendeten Lernverfahrens, eines dezentralen evolutionären Algorithmus, eingesetzt werden. Es steuert die Interaktion zwischen zwei Agenten in Bezug auf den in AVALANCHE eingesetzten evolutionären Algorithmus STDEA.

Für die Implementation der Interaktion zwischen den Agenten, insbesondere der Kommunikation und der Verhandlung, werden Objekte der in dem Paket "`*.negotiation`" zusammengefaßten Klassen benötigt:

Die grundsätzliche Klasse für die Nachricht stellt bei AVALANCHE die Klasse `AvAgentMessage`. Sämtliche Nachrichten zwischen Agenten werden in Objekten dieser Klasse verpackt

und direkt zwischen den Agenten ohne die Einschaltung eines Intermediärs ausgetauscht. Der Mechanismus, der beim Eintreffen einer Nachricht den Agenten auf dieses Ereignis reagieren lässt, wird durch die Klassen `AvAgentEvent` und `AvAgentEventListener` implementiert, die technische Funktion haben und die Kommunikation selbst nicht beeinflussen.

Die semantische Bedeutung der Nachrichtentypen orientiert sich an den Standardisierungsvorschlägen der *FIPA* (Foundation for Intelligent Physical Agents) für eine "*Agent Communication Language*" (ACL) (FIPA 1997). Für die Nutzung in *AVALANCHE* wurde nur eine Untermenge der dort vorgeschlagenen Nachrichtentypen implementiert. Das genutzte Verhandlungsprotokoll orientiert sich am *Iterated-Contract-Net-Protocol* der ACL (FIPA 1997), die ihrerseits in den hier verwendeten Elementen auf Kontraktnetzen (Smith 1980) basiert. Der syntaktische Aufbau der Nachrichten in der Klasse `AvAgentMessage` orientiert sich an diesen Vorschlägen, mit spezifischen Unterschieden, die in Tabelle 2 dargestellt sind..

Tabelle 2: Vergleich zwischen AVALANCHE Implementation und ACL Spezifikation

AVALANCHE Implementation	ACL Spezifikation	Bedeutung
String <i>performative</i> ^a	Benennung des "communicative act"	In dieser Variable steht der Nachrichtentyp, z.B. propose, reject, accept-offer.
String <i>content</i>	:content	Diese Variable enthält die eigentliche Information der Nachricht, bei einem "propose" z.B. wird hier das Gut und der verlangte Preis kodiert.
String <i>inReplyTo</i>	:in-reply-to	Hier wird eine Referenz auf die vorausgegangene Nachricht übermittelt, um die Verhandlungssequenz verfolgen zu können.
String <i>language</i>	:language	Diese Variable wird nicht genutzt. In der ACL Spezifikation ist dieses Feld für die Festlegung der Codierung der Variable "content" vorgesehen.

Tabelle 2: Vergleich zwischen AVALANCHE Implementation und ACL Spezifikation

<i>String protocol</i>	:protocol	Diese Variable wird zur Einordnung der empfangenen "communicative acts" verwendet. In AVALANCHE gibt es 4 Protokolle, die dem Handel, der Bewegung mobiler Agenten, der Suche nach Anbietern und der Reproduktion der Agenten dienen.
<i>String receiver</i>	:receiver	Variable zur eindeutigen Identifikation des Empfängers.
<i>String replyWith</i>	:reply-with	Ungenutzte Variable. In der ACL Spezifikation ist dieses Feld als Vorschlag des Senders für die Festlegung des "in-reply-to" vorgesehen.
<i>String sender</i>	:sender	Variable zur eindeutigen Identifikation des Senders.
<i>Object payload</i>		Diese Variable kann je nach Nachrichtentyp unterschiedliche Bedeutung haben. In ihr können vor allem solche Informationen übermittelt werden, die nicht in String-Objekten codiert werden können.

a.Statt der Möglichkeit, jeden Nachrichtentyp als eigene Klasse abzuleiten, wurde in Avalanche eine Variable performative gewählt, die in der Klasse AvAgentMessage die semantische Bedeutung der überlieferten Nachricht bestimmt

Auch die zwischen den AvTradeAgents und AvLocationAgents ausgetauschten Nachrichten werden in der Klasse AvAgentMessage kodiert. Dabei handelt es sich vor allem um Informationsanforderungen an den Verzeichnisdienst im Protokoll "*recommend-seller*". Einerseits suchen Agenten nach fremden Handelspartnern, andererseits muß bei Bedarf der eigene Eintrag geändert werden (siehe Kapitel 4.6). Im Falle der Bewegung des Agenten auf eine andere Lokation muß über das Protokoll "*move*" eine Information über die existierenden Nachbarlokationen erfragt werden. Der Verzeichnisdienst stellt eine Möglichkeit zur Koordination zwischen den Agenten dar und wird in einem späteren Kapitel als notwendiges Element eines elektronischen Marktplatzes dargestellt. Die im weiteren Verlauf des Kapitels dargestellten Quellcodeauszüge der Versendung von Nachrichten sind wie in den beiden folgenden Beispielen dargestellt. Die Schaffung eines Nachrichtenobjekts erfolgt durch:

```
public AvAgentMessage(String Performative, String Content,  
    String InReplyTo, String Language, String Protocol,  
    String Receiver, String ReplyWith, String Sender,  
    Object Payload) {  
    (...)}  
}
```

Ein Beispiel aus der Verhandlung zwischen den Agenten ist z.B.:

```
AvAgentMessage answer =  
    new AvAgentMessage("cfp", pending.offer.item,  
        null, null, "trade",  
        sellerAgent, pending.offer.id,  
        agentName, this);
```

Hierdurch wird eine Nachricht mit dem Performativ „*cfp*“ erzeugt, eine Aufforderung an einen Verkäufer zur Abgabe eines Gebotes. Inhalt der Nachricht ist die eindeutige Bezeichnung des gewünschten Gutes. Da es sich um die Aufnahme einer neuen Verhandlung handelt, bezieht sich die Nachricht auf keine Vorgängernachricht. Die Nachricht gehört zum Interaktionsprotokoll „*trade*“ und ist in den Kontext einer Verhandlung einzuordnen. Empfänger der Nachricht ist ein vorher ausgewählter Verkäuferagent. Um mögliche Antworten einordnen zu können, erzeugt der Agent ein Kennziffer für diesen Verhandlungsvorgang `pending.offer.id`. Als Sender gibt er seinen eigenen Namen an und als `Payload` schickt er einen Zeiger auf seinen eigenen Prozess mit, was einen Implementationstrick zur schnellen Kommunikation darstellt.

4.5.3 Kooperation, Interaktion und Verhandlungen

Die zweckgebundene Kommunikation zwischen unterschiedlichen Marktteilnehmern (`AvTradeAgents`) wird durch die Notwendigkeit zur bilateralen Kooperation erzwungen. Im Fokus des hier Beschriebenen handelt es sich um die Kooperation zwischen den handelnden Subjekten, die den Zweck verfolgen, eine Markttransaktion jeweils erfolgreich abzuschließen. Der Erfolg einer Transaktion wird von den Agenten in Geldeinheiten gemessen. Eine einzelne Transaktion ist dann erfolgreich, wenn der einkaufende Agent unter seinem Reservepreis bleibt, oder umgekehrt der verkaufende Agent einen höheren Erlös erzielt. Diese Ziele werden im Allgemeinen antagonistisch sein und damit handelt es sich in dem in *AVALANCHE* implementierten Markt um die klassische Situation einer "Kooperation zwischen Egoisten" (Moulin 1995, S. 5). Pro Transaktion kann die Kooperationssituation als spieltheoretisches Problem dargestellt werden, in der kooperatives Verhalten aus Sicht eines Agenten als unilaterale Konzession des Gegenübers übersetzt wird.

Über einen längeren Zeitraum ist der Erfolg an der gesamten Höhe des Gewinns zu messen, d.h. der Summe der aus den einzelnen Transaktionen erzielten Erlöse. Dieses langfristige Ziel ist dem kurzfristigen Transaktionserlös übergeordnet und führt zu Strategien, die gleichermaßen Gewinn als auch Umsatz (Transaktionen pro Zeitraum) einbeziehen müssen. Es ist nicht notwendig, jede Transaktion zu „gewinnen“, sondern die Umgebung derart zu beeinflussen, dass möglichst viele Transaktionen innerhalb eines gegebenen Zeitraums zufriedenstellend beendet werden.

Ein Agent, der im Namen seines Besitzers handeln soll, benötigt (Preist 1998, Kraus 1998):

- eine Spezifikation der Güter oder Dienste, die er handeln soll,
- eine Vorstellung des Zieles, welches sein Besitzer aus der Aushandlung zu erreichen wünscht und
- eine Verhandlungsstrategie, die mindestens genauso effektiv ist wie diejenige eines qualifizierten Menschen in der gleichen Situation..

Diese Eigenschaften werden in ihrer für AVALANCHE gewählten Implementation im folgenden vorgestellt.

4.5.4 Güter

De Paula et al. (2000) unterscheiden bei der Spezifikation der zu handelnden Güter zwischen den Produktattributen und den Verhandlungsattributen (*deal attributes*). Die Produktattribute sind nicht verhandelbar und beschreiben das Gut. Die Verhandlungsattribute sind für die aktuelle Verhandlung spezifisch und umfassen jene Variablen, bei denen von den Verhandlungspartner überhaupt Konzessionen eingegangen werden können. In der derzeitigen Version von AVALANCHE sind die Güter als Wirtschaftsgüter (*commodities*) ausgezeichnet, das einzige Verhandlungsattribut ist der Preis. Die Erweiterung auf weitere Verhandlungsattribute ist durch die Architektur nicht ausgeschlossen, wird aber zur Darstellung der grundsätzlichen Funktionsweise des Koordinationsmechanismus nicht benötigt.

Die Produktattribute stellen die Variablen der Klasse `AvFactor` dar. Zum gegenwärtigen Zeitpunkt enthält diese Klasse nur die Variable `factorType`, die eine eindeutig unterscheidbare Güterart kodiert. In realistischeren Implementationen wird diese Klasse weitere Beschreibungselemente umfassen, um zusätzliche Gütereigenschaften (wie Qualität, Substitutionsmöglichkeiten, Maße und Gewichte) darzustellen. Diese werden in Multi-Attribut-Verhandlungen zusätzlich zum Güterpreis in die Präferenzbildung der handelnden Agenten einfließen.

Die Verhandlungsattribute, d.h. die aktuellen Preise sowie die Historie der Gebote, wird in AVALANCHE in einem `AvOffer`-Objekt gespeichert. Die Initialisierung von `AvOffer` erfolgt mit einem `AvTradeAgent`-Objekt als dem Anbieter des Angebots, einem `AvFactor`-Objekt als Angebotsgegenstand, sowie zwei ganzzahligen Variablen, die Preis und Menge angeben. Wird nur der Anbieter angegeben, wird ein "leeres" Angebot erzeugt (Preis und Menge sind 0). Ein Ausschnitt aus dem Quellcode von `AvOffer` zeigt diese Variablen.

```
public class AvOffer implements java.io.Serializable,
    java.lang.Cloneable {
    public String owner;           // AvTradeAgent.getName()
    public String item = null;    // AvFactor.getName()
    public int    amount = 0;
    public double price = 0;
    public String id;             // the unique offer id
```

4.5.5 Zielbildung der Agenten

In der vorliegenden Implementation von AVALANCHE ist das Ziel der Agenten die Maximierung des eigenen Nutzens im Zeitablauf. Da die Agenten sowohl einkaufen als auch verkaufen können, bedeutet dies letztlich auf der Einkaufsseite die Vergrößerung der Distanz zwischen Indifferenz- und Einkaufspreis, auf der Verkaufsseite die Distanzausweitung zwischen Verkaufs- und Indifferenzpreis.

Dazu wird die Variable `del_jump`, die im Genotype kodiert ist, eingesetzt. Sie entscheidet über die prozentuale Veränderung des Verhandlungsstartpreises in aufeinanderfolgenden, erfolgreichen Transaktionen. Je höher die Variable ist, desto stärker ist das Streben nach Vergrößerung der Preisdistanzen. Eine detaillierte Schilderung des Mechanismus folgt im nächsten Abschnitt. Diese einfache Zielbildung, die in Geldeinheiten gemessen werden kann, ergibt sich aus der Einfachheit der Produkte, die als Wirtschaftsgüter definiert sind³⁶. Der Grad der Zielerreichung wird den Agenten nicht "bewußt" und damit gibt es auch kein Optimum der Zielerreichung. Der Versuch, den eigenen Nutzen zu maximieren, ist weder in der Höhe noch in der Zeitdauer begrenzt und stellt einen nicht endenden Prozeß dar.

Das stetige Streben der Agenten, besser als andere zu werden, schlägt sich in dem gewählten Lernverfahren nieder. Die Suche nach solchen Parameterkombinationen, die besonders gute Nutzenergebnisse versprechen, führt dazu, dass sich Agenten mit anderen vergleichen und

³⁶ Ansätze zur Optimierung mehrerer Zielattribute, wie Multi-Attribute-Utility-Theory oder Constraint Satisfaction (Guttman et al. 1998) brauchen in diesem Stadium noch nicht angewendet werden, bieten sich für eine Erweiterung an.

deren Parameterwerte (soweit diese bekannt sind) mit ihren eigenen Parameterwerten kombinieren. In dieser Hinsicht führt die Ausgestaltung des Zieles in AVALANCHE in direkter Linie zur Implementation des dezentralen evolutionären Algorithmus STDEA als Lernverfahren.

4.5.6 Verhandlungsstrategie

Die Verhandlungsstrategie dient dazu, das Nutzenziel zu erreichen. Sie spezifiziert die Handlungen, die dem Agenten zur Erreichung des Zieles offenstehen und wählt aufgrund von Informationen über die Umwelt aus alternativen Handlungen die im Sinne der Zielerreichung beste aus. Die Verhandlungsstrategie wird durch 6 Parameter definiert: *Acquisitiveness*, *deltaChange*, *deltaJump*, *Satisfaction*, *WeightMemory* und *Reputation*. Deren Auswirkungen auf die Strategie werden in diesem Abschnitt vorgestellt.

Der erste dieser Parameter, *Acquisitiveness* bzw. `p_acquisitiveness`, bestimmt die Wahrscheinlichkeit, eine unilaterale Konzession einzugehen, d.h. während einer Verhandlung, ob der Agent beim folgenden Angebot seinen Preis verändert. Die tatsächliche Entscheidung wird durch eine stochastische Probe bestimmt. Der Wert des Parameters determiniert nicht die Handlung, sondern gibt eine Wahrscheinlichkeit an. Der Wertebereich des Parameters liegt zwischen 0 und 1. Ein Wert von 0.7 bedeutet eine 70%ige Wahrscheinlichkeit, dass der Agent eine kompetitive Strategie verfolgt, seinen Preis also nicht anpasst. Tritt der Agent als Verkäufer auf, so wird er, sofern der Aufruf der Methode `randomNumberIsHigherThan(p_acquisitiveness)` den booleschen Wert "*false*" liefert, seinen Preis in einer bestimmten Höhe senken, wie im folgenden Quellcodeauszug aus der Methode `negotiate()` der Klasse `AvTradeAgent` gezeigt³⁷. Tritt der Agent als Käufer auf, so wird er sein Angebot erhöhen. Ein Agent mit einem *Acquisitiveness*-Wert von 1 wird seinen ursprünglichen Preis niemals ändern. Ein Agent mit einem *Acquisitiveness*-Wert von 0 wird bei jeder Entscheidung eine unilaterale Konzession eingehen, d.h. sein Gebot in Richtung auf das Gebot des anderen Agenten ändern.

```
// check whether we will make a different offer
    if (randomNumberIsHigherThan(p_acquisitiveness)) {
// if I overcome my greed...
        sellActPrice = sellActPrice - current.stepSize;
// ...then lower price by step size
```

37 Alle Quellcodeauszüge in diesem Abschnitt sind dieser Methode `negotiate()` entnommen, sofern keine andere Quelle genannt wird.

Die Höhe der Konzession (`stepSize`) wird durch den zweiten Parameter `del_change` (*DeltaChange*) festgelegt. Die Festlegung einer absoluten Höhe der Konzession nimmt keine Rücksicht auf die Höhe des geforderten Preises, so dass in AVALANCHE eine prozentuale Berechnung implementiert wurde. Durch eine direkte Angabe der Konzession als Prozentanteil des Gebots wird der Verkäufer benachteiligt, da dieser in einer Verhandlung von einer höheren absoluten Zahl ausgeht. Ein Beispiel mag dies verdeutlichen:

Anna möchte ein Gut für € 80 verkaufen, Benno bietet aber nur € 60. Falls sich beide zu einer Konzession von 20% entscheiden, so bietet Anna in der nächsten Runde € 16 weniger (was ihrer Nutzensenkung entspricht), Benno aber nur € 12 mehr. Der Nutzenzuwachs wird auf diese Weise nicht symmetrisch sein, sondern immer zu Gunsten von Benno ausgehen³⁸.

Die Marktpreise würden durch ein solches Vorgehen im Zeitablauf im Gesamtsystem gesenkt, was ein falsches Bild über die Koordinationsergebnisse erzeugen würde. In AVALANCHE wurde ein prozentualer Anteil der Differenz zwischen den Startgebotsen beider Parteien ausgewählt. Ein Wert von `del_change=0.25` bedeutet in diesem Fall das Erreichen des gegnerischen Preisangebots in 4 Verhandlungsrunden (sofern der Gegner nicht ebenfalls entgegenkommt, sondern bei seinem Preisangebot verharrt). Der folgende Quellcodeauszug wird bei beiden Agenten zu Beginn der Verhandlung einmal durchlaufen.

```
if (current.stepSize == 0) {  
    //we need to determine the step size for this negotiation  
    current.stepSize = (sellActPrice - offer.price) * del_change; }
```

Damit das Eingehen einer Konzession Sinn macht, muß es umgekehrt eine drohende Wahrscheinlichkeit geben, dass der Partner die Verhandlung verläßt. Diese Wahrscheinlichkeit wird in dem Parameter `p_satisfaction` abgebildet. Falls die Verhandlungsrunden nicht zur Zufriedenheit des Agenten verlaufen (d.h. dass der Transaktionspartner seinen Preis nicht ändert und damit ein kompetitives Verhalten zeigt), wird durch stochastische Prüfung gegen den Parameter `p_satisfaction` eine Entscheidung über den Abbruch der Verhandlungen (d.h. ein Senden der Nachricht *reject-offer* anstelle eines Gegenangebots *propose*) getroffen. Ein Wert von `p_satisfaction=0.75` bedeutet eine 75% Chance eines Verhandlungsabbruchs im Moment der Prüfung. Bei jeder neuen Verhandlungsrunde kann jeder der beiden

38 Unter der Annahme, dass gleiche Geldbeträge identische Nutzenveränderungen bei allen Agenten bedeuten.

Partner diesen Abbruch einleiten – je kürzer die Verhandlung dauert, desto wahrscheinlicher ist der erfolgreiche Abschluß der Transaktion. Durch einfache statistische Berechnung lässt sich zeigen, dass die Wahrscheinlichkeit eines Verhandlungsabbruchs mit zunehmender Verhandlungsdauer steigt: die Abbruchwahrscheinlichkeit nach der n -ten Runde ergibt sich als $(1 - \text{Satisfaction})^n$.

```
if (current.lastPriceOffer >= offer.price) {
    // if other party doesn't raise price offer
    if (randomNumberIsHigherThan(p_satisfaction)) {
        // then check against our satisfaction and
        // reject eventually
```

Bisher sind drei Parameter identifiziert worden, *Acquisitiveness*, *DeltaChange* und *Satisfaction*. Ein weiterer Parameter folgt aus dem stetigen Streben der Agenten nach Nutzenmaximierung. Es wird davon ausgegangen, dass ein Agent (und sein Besitzer) mit dem Erreichten niemals zufrieden ist. Um bei der nächsten Verhandlung ein höheres Ergebnis zu erzielen, hat der Agent zwei Möglichkeiten. Entweder vermindert er seine Konzessionsrate, durch Erhöhen des $p_acquisitiveness$ -Wertes. Oder er beginnt mit einem höheren Ausgangswert (als Verkäufer), damit er bei der gleichen Konzessionsrate auf einem höheren Endpreis landet. In AVALANCHE wird davon ausgegangen, dass die Grundanlagen der Strategie (die durch die hier vorgestellten Parameter beschrieben werden), durch den Agenten selbst ebenso wenig veränderbar sind, wie ein Mensch seine genetischen Anlagen und die Grundelemente seines Verhaltens verändern kann. Um dieser Diskussion aus dem Weg gehen und Genotyp (Veranlagungen) und Phänotyp (Verhalten) sauber zu trennen, wird die zweite Möglichkeit implementiert, die Erhöhung des Startnutzens. Rechnerisch führt dieses Vorgehen zu einer Erhöhung des Startpreises für Verkäufer (berechnet als Summe aus Indifferenzpreis und erwünschtem Nutzenzuwachs) bzw. zu einer Verminderung des Startgebots für Käufer (berechnet als Indifferenzpreis abzüglich erwünschtem Nutzenzuwachs). Der Parameter del_jump modifiziert in AVALANCHE den jeweiligen Startwert bei Neuverhandlungen. Das Verhandlungsergebnis der letzten Transaktion wird gespeichert. Durch Multiplikation mit einem prozentualen Aufschlag del_jump errechnet sich der Ausgangswert für die nächste Transaktion. Ein Beispiel für $del_jump=0.15$: ein Agent hat beim letzten Mal ein Gut für € 100 verkauft. Beim nächsten Mal wird er mit € 115 in die Verhandlungen einsteigen.

```
new starting price for negotiations is agreement price plus
del_jump%
    lastSellPrice = agreement;
    sellMaxPrice = agreement * (1.0 + del_jump);
```

Bei der Vorstellung der bisherigen Parameter war bereits die Rede von der Lernfunktion der Agenten in Bezug auf Marktpreise. Allein damit ist den Agenten eine Bewertung eines ihnen vorliegenden Angebots möglich. Findet eine solche Bewertung nicht statt, wird der Agent bei jedem Gebot in eine Verhandlung einsteigen und diese ernsthaft betreiben. Dadurch wird spekulativen Angeboten Tür und Tor geöffnet. Ein Beispiel:

Anna kann € 1000 für ein Gut fordern, Benno mit € 10 dagegenhalten und beide werden verhandeln, bis sie sich irgendwann bei ca. € 505 treffen. Nur die Kenntnis des üblichen Marktpreises kann verhindern, dass einer von beiden Agenten schwer benachteiligt wird. Falls der Marktpreis für das Gut bei € 900 liegt, hat Benno ein tolles Schnäppchen gemacht, ein Marktpreis von € 20 lässt den Vorwurf der Wucherei an Anna zu.

Dieser Überlegung liegt das Vorhandensein sogenannter "common value" Gütern zugrunde, d.h. der Preis des Gutes kann an Marktpreisen gemessen werden, da es im Prinzip unendlich weiterverkauft werden kann (Sandholm 1996b, S. 1). Je mehr Information ein Agent über die unterschiedlichen Preise erlangt, die für ein bestimmtes Gut verlangt werden, desto besser kann er sich ein Bild über den Markt machen und seine eigene Preissetzung (oder die Bewertung der Preissetzung anderer) daran ausrichten.

Um feststellen zu können, ob der vom Transaktionspartner im allerersten Angebot angegebene Preis überhaupt in einen Bereich fällt, der eine erfolgreiche Verhandlungslösung zulässt, prüft der Käufer diesen vor Aufnahme von Verhandlungen. Er vergleicht den erhaltenen Preis mit denjenigen Preisangeboten, die er in der Vergangenheit von anderen Agenten erhalten hat, unabhängig davon, ob diese Angebote später zu Abschlüssen geführt haben oder nicht. Insbesondere kann er Wuchergebote, wie oben beschrieben, von Beginn an ablehnen, wie im folgenden Codeauszug gezeigt. In AVALANCHE werden Gebote, die offensichtlich über das Doppelte des üblichen Marktpreises hinausgehen, von vornherein als unseriös betrachtet und ein einseitiger Verhandlungsabbruch eingeleitet. Bei Geboten, die zwischen dem Marktpreis und dieser Grenze liegen, wird der Agent als unsicher betrachtet und prüft erneut gegen den bereits vorgestellten Parameter *p_satisfaction*, ob die Verhandlung gleich (sozusagen in der null-ten Runde) beendet wird.

```
if (offer.price >= memory) {
// ...then use memory to check for too high prices
    if ( randomNumberIsHigherThan(p_satisfaction) ) {
        reject = true;
    }
}
```

```

if (memory != 0 && offer.price > 2 * memory) {
    reject = true;
}

```

In welcher Weise die Preisinformationen zum Bilden eines Marktpreises verarbeitet werden, muß in den Agenten selbst formalisiert werden, da prinzipiell kein zentrales Informationsmittel (z.B. eine Datenbank oder ein Auktionator) zur Verfügung steht. In AVALANCHE wird jede Preisinformation, die einem Agenten zur Kenntnis gelangt (durch erhaltene Gebote) mittels exponentieller Glättung 1. Ordnung mit den bestehenden Preisinformation verknüpft und in der Variablen `memory` (Marktpreis) gespeichert. Die grundsätzliche Formalisierung der Wertbildung sieht folgendermaßen aus:

$$\text{Marktpreis}_t = \text{AktuellerGebotspreis} \times \alpha + \text{Marktpreis}_{t-1} \times (1 - \alpha). \quad (3)$$

Das Verhältnis der Gewichtung aktueller Preisinformationen gegenüber historischen Preisinformationen übt einen großen Einfluß auf die Strategie des Agenten aus. Je höher der Gewichtungsfaktor α (in der AVALANCHE-Implementation als `w_memory` bezeichnet) ist, desto stärker wird der aktuelle Wert einbezogen und desto schneller wird sich der Wert des Marktpreises der aktuellen Marktlage anpassen. Andererseits ist der Agent in diesem Fall wesentlich schneller durch kurzfristige Preisschwankungen auf dem Markt beeinflussbar und wird diese Entwicklung im folgenden durch sein aus dem Parameter abgeleitetes Verhalten vermutlich verstärken. Ein "richtiger" Wert für diesen Gewichtungsfaktor kann a priori nicht festgelegt werden, sondern ergibt sich aus dem Zusammenspiel der Agenten in der marktlichen Koordination. Dieser Parameter `w_memory` wird ebenfalls beim Start des Agenten als verhaltenseigen festgelegt. Er geht direkt anschließend an das vorherige Codebeispiel in folgender Weise in das Lernen der Marktpreise ein. Allerdings werden auch Wucherpreise in die Formel einbezogen, damit eine tatsächliche rasche Veränderung der Marktgegebenheiten trotzdem nachvollzogen werden kann.

```

memory = (1-w_memory) * memory + w_memory * offer.price;
// update our memory of initial prices

```

Für Simulationszwecke in Bezug auf unterschiedlich kooperatives Verhalten der Agenten wird ein weiterer Wert `p_reputation` eingeführt, der die Wahrscheinlichkeit angibt, vereinbarte Transaktionen korrekt abzuwickeln. Für die Implikationen, die sich aus in diesem Sinne unterschiedlichem strategischem Verhalten der Agenten in offenen elektronischen Märkten ergeben, sei auf Padovan (2000) verwiesen. Die in der vorliegenden Arbeit betrachteten Agenten sind kooperativ, im dem Sinne, dass der Güter- und Werteübergang immer korrekt und ohne Änderung zu den ausgehandelten Ergebnissen durchgeführt wird.

Mit den Parametern `p_acquisitiveness`, `del_change`, `del_jump`, `p_satisfaction` und `w_memory` kann das Verhalten eines Agenten vollständig beschrieben werden. In Verbindung mit den Sprechakten des Verhandlungsprotokolls, die im übernächsten Abschnitt beschrieben werden, ergibt sich ein nicht-deterministischer endlicher Automat. Die Parameter werden durch die Agenten selbst nicht verändert, eine Adaption ergibt sich aber durch die Implementation des evolutionären Algorithmus STDEA.

4.5.7 Realisierung der Adaptivität

Die grundlegenden Ideen des STDEA von Smith/Taylor (1998) wurden in Kapitel 3.5.3.5 dargestellt. In diesem Abschnitt geht es um die Implementation des Konzeptes in AVALANCHE. Die "Gene" des Agenten stellen die Strategieparameter dar, der Genotyp jedes Agenten ist seine Kombination der Variablenwerte `p_acquisitiveness`, `p_satisfaction`, `del_change`, `del_jump`, `p_reputation` und `w_memory`. Diese Werte sind für den Agenten konstant, er kann sie selbst nicht verändern. Das Experiment beginnt mit einer in Mittelwert und Varianz festgelegten Zufallsverteilung der Parameter auf die Agentenpopulation. Insgesamt ergibt sich eine heterogene Verteilung von Handelsstrategien, da der Genotyp jedes Agenten einzigartig ist.

Die Fitness jeder Handelsstrategie ergibt sich in AVALANCHE aus der Höhe des Einkommens, welches der Agent aus Verhandlungssituation mit anderen Agenten, die zu erfolgreichen Transaktionen führen, erzielen kann. Dieses Einkommen wird jeweils zum Vermögen des Agenten hinzuaddiert – solange alle Agenten mit dem gleichen Vermögen starten (wie im Experiment durchgeführt), lässt sich an diesem das Einkommensergebnis über die Zeit ablesen. Alle Kauf- und Verkaufshandlungen haben unmittelbaren Einfluß auf den Kapitalbestand – einerseits werden die erworbenen Güter aus ihm bezahlt, andererseits führt der Verkauf eines Gutes zu einem Kapitalzufluß. Durch die Einführung eines festen Ausgabebetrags `costsOfLife`, der in gleichen Zeitabständen durch den Marktplatz, auf dem sich der Agent befindet, eingezogen wird, ist jeder Agent gezwungen, durch erfolgreiche Transaktionen im Zeitablauf sein Vermögen zu mehren. Der Ausgabebetrag kann als Kapitalsteuer oder konstante Benutzungsgebühr verstanden werden und führt dazu, dass eine Strategie des Nichtstuns nicht als erfolgreich bestehen kann, sondern die Agenten Transaktionen durchführen müssen. Ein Agent, dessen Vermögen auf Null sinkt, wird vom Marktplatz entfernt (im Experiment gelöscht, in einer realen Umgebung zu seinem Besitzer zurückgeschickt) – seine Strategie steht dem System nicht mehr zur Verfügung.

Jeder Agent sendet nach einer erfolgreich abgeschlossenen Transaktion ein sogenanntes "plumage"-Objekt ab, das seine Fitness sowie seinen Genotyp enthält. Die Plumages werden

durch den `AvLocationAgent` einem zufälligen Agenten des gleichen Agententyps wie der Sender zugestellt. Die Nachricht *"inform"* stellt das einzige Performativ des Protokolls *"reproduction"* dar.

```
AvAgentMessage msg =
    new AvAgentMessage("inform", "plumage", null,
        null, "reproduction", null,
        null, agentName,
        createPlumage());
```

Die Fitness des Agenten wird als exponentiell geglätteter Durchschnitt des Nutzenzuwachses aus den Transaktionen berechnet. Der Genotyp selbst ist ein Vektor mit den o.a. sechs Strategieparametern.

```
final Plumage createPlumage() {
    return new Plumage(
        agentName, tribe, estimateFitness(), getGenes());
```

Hat der `AVALANCHE`-Agent eine höhere Anzahl an Plumages erhalten, als in dem Parameter `courterThreshold` vorgegeben wurde, wählt er aus diesen denjenigen Genotyp mit dem höchsten Fitness (-Versprechen) aus und führt über die Methoden

```
Genotype g = getGenes().cross(p.getSenderGenes(),
    randomGenerator);
g.mutate(randomGenerator);
```

die Phasen der Rekombination und der Mutation durch. In der gegenwärtigen Implementation wird aus dem so erhaltenen neuen Genotyp ein neuer Agent des gleichen Agententyps generiert. Die Implementierung benötigt dazu die in Abbildung 52 dargestellten Klassen. Sie sind von Smith/Taylor (1998) übernommen und leicht modifiziert worden.

4.5.8 Das Verhandlungsprotokoll

Das in `AVALANCHE` eingesetzte Verhandlungsprotokoll dient der Interaktion zwischen den Agenten. Es definiert einen Prozeß, in dem bestimmte Sprechakte aufeinanderfolgen. In Kenntnis vollständiger Marktinformation kann das Verhandlungsprotokoll äußerst kurz und trivial sein:

"There are all sorts of examples of different protocols, different rules, that will bring the

agents to an agreement that maximizes the product of their utilities. You can even have a one-step protocol; if they know everything, they can compute the agreement point, which will be the point they jump to." (Rosenschein/Zlotkin 1994b, S. 38).

Bei dem vorliegenden dezentralen Konzept wird der Zustand vollkommener Information nie erreicht; beide Agenten einer Verhandlung nähern sich dem "*agreement point*" in iterativer Weise über Angebot und Gegenangebot, bis ein Kompromißpreis erreicht wird. Dieses Vorgehen kann als monotonen Konzessionsprotokoll ("*monotonic concession protocol*", Rosenschein/Zlotkin 1994a, S. 40) bezeichnet werden.

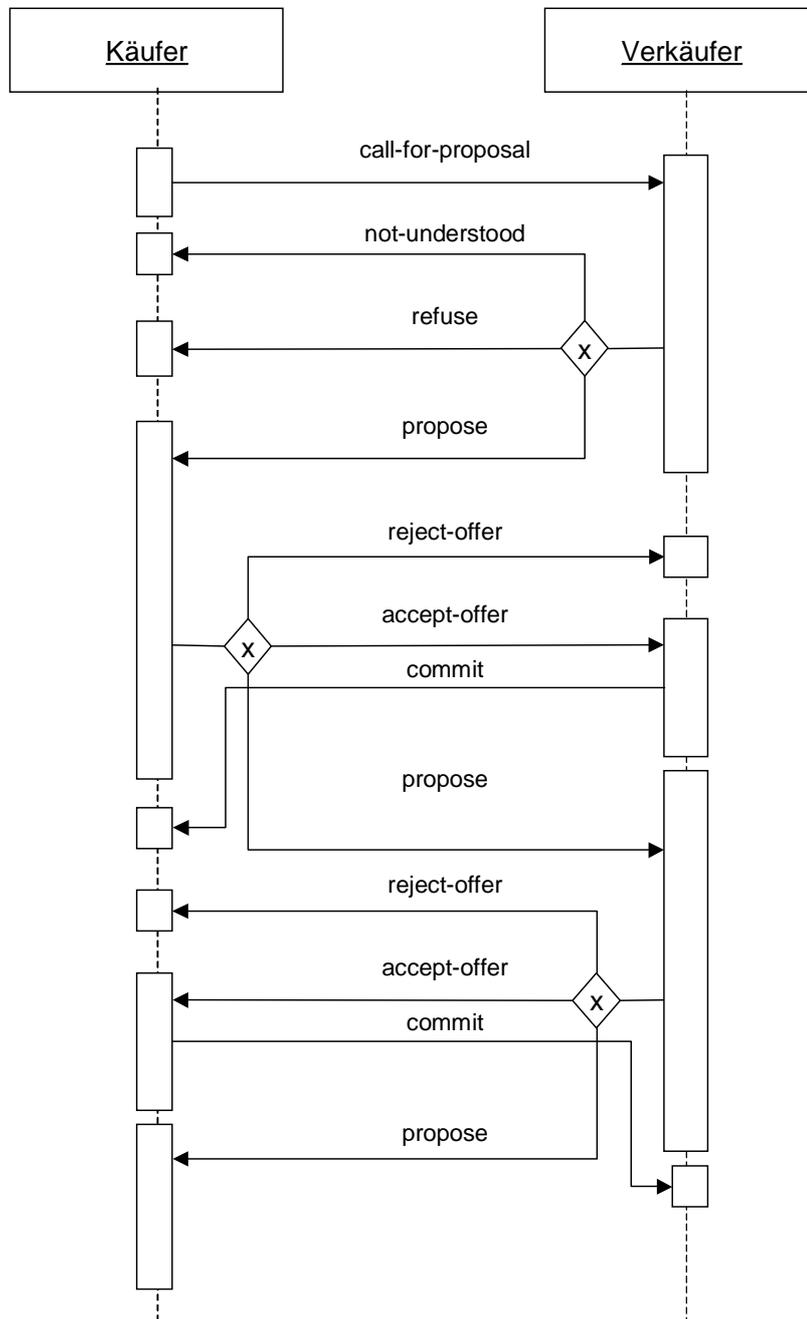


Abb. 22: AUML-Sequenzdiagramm der Verhandlung zwischen Agenten

AVALANCHE unterscheidet verschiedene Kommunikationsprotokolle, von denen nur das Protokoll "trade" die Strukturierung der Verhandlung und damit das Koordinationsergebnis beeinflusst. Die innerhalb des Protokolls "trade" versendeten Nachrichtentypen werden bilateral

zwischen zwei Instanzen der handelnden Agenten (d.h. von `AvTradeAgent` oder daraus abgeleiteten Klassen) verschickt. Ziel der Kommunikation ist die Durchführung einer ökonomischen Aushandlungsphase mit Angebot und Gegenangebot bis zum Erreichen einer Vereinbarung über Güter- und Wertaustausch. Das UML-Sequenzdiagramm für dieses Protokoll ist in der folgenden Abb. 22 dargestellt.

Dieses Interaktionsprotokoll führt zu unterschiedlichen Zuständen im internen Modell des Agenten, die in Abb. 23 exemplarisch als nicht-deterministischer endlicher Automat für einen Käuferagenten dargestellt sind.

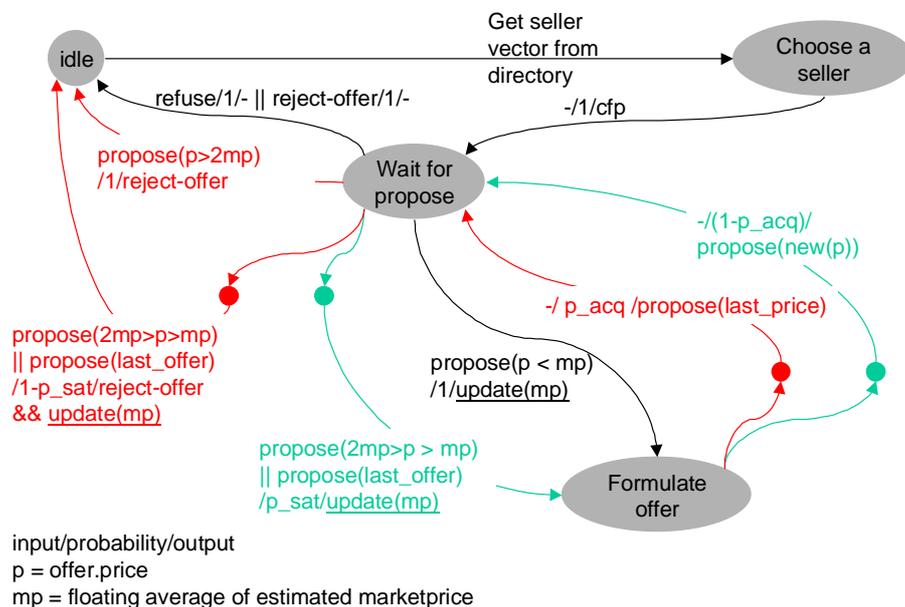


Abb. 23: Zustandsdiagramm eines Agenten in der Käuferrolle

Auf Grundlage dieser beiden Abbildungen lässt sich im folgenden ein detaillierter Ablauf einer Verhandlung schildern. Aus der Kombination des Verhandlungsprotokolls "trade" und der Strategieparameter ergibt sich der in Abbildung 23 dargestellte nicht-deterministische endliche Automat (engl. *non-deterministic finite state machine*" bzw. abgekürzt *non-det FSM*) dargestellt werden, dessen Zustände den Verhandlungszuständen entsprechen. Das Eingabealphabet des FSM besteht aus der Menge an Nachrichten, die von den Teilnehmern versendet werden kann, als Tupel `<Sender, Nachricht>`. Das Ausgabealphabet ist die Potenzmenge derjenigen Nachrichten, die an die Teilnehmer versendet werden können. In der Abbildung 23 ist jeder Zustandsübergang durch ein Tupel `<Eingabe/Wahrscheinlichkeit/Ausgabe>` gekennzeichnet.

zeichnet, wobei verschiedene Eingabe- und Ausgabekonditionen durch logische Oder (||) bzw. Und (&&) verknüpft sein können.

Die Darstellung zeigt einen Käuferagenten, der entsprechende Verkäufer ist im wesentlichen spiegelbildlich zu sehen. Die Abbildung zeigt vier Zustände, einen Ruhezustand (*idle*), die Auswahl eines Verkäufers (*choose seller*), die Formulierung und Absendung eines eigenen Proposals (*formulate offer*) und das Warten auf die Reaktion des Verhandlungspartners (*wait for propose*).

Der Käuferagent beginnt üblicherweise im Zustand "*idle*". Bei Auftreten eines Kaufwunsches wird er zuerst (entweder über eine "*recommend-seller*"-Nachricht an den `AvLocationAgent` oder eine andere Verzeichnisimplementation) Informationen über die verfügbaren Anbieter gesammelt. Aus diesen wählt er zufällig einen Transaktionspartner aus. Dieser Auswahlprozess unterscheidet einen 0.-Ebene-Agenten (Vidal/Durfee 1998) von einem 1.-Ebene-Agenten, welcher gezielt auf Grundlage historischer Erfahrungen einen Transaktionspartner ansprechen würde.

Das Verhandlungsprotokoll beginnt mit dem Abschicken einer Nachricht vom Typ "*cfp*" (*call for proposal*). Diese wird vom Käuferagenten direkt an den Verkäuferagenten geschickt. Die Nachricht enthält lediglich Informationen über die Art des nachgefragten Gutes, stellt aber noch kein Gebot dar, da kein Preis übermittelt wird:

```
AvAgentMessage answer =
    new AvAgentMessage("cfp", pending.offer.item,
        null, null, "trade",
        sellerAgent, pending.offer.id,
        agentName, this);
```

Der Verkäuferagent sendet als Antwort ein "*propose*" mit genaueren Angaben zu seinem Angebot, insbesondere seinen Startpreis. Er geht hier immer in Vorleistung gegenüber dem Käuferagenten, der zu diesem Zeitpunkt einen Informationsvorsprung hat (er kennt beide Ausgangspreise, der Verkäufer nur seinen). Sein Angebot speichert der Agent in einem `AvOffer`-Objekt, um bei einer Antwort des Verhandlungspartners dessen Angaben mit seinen eigenen vergleichen zu können. Das "*content*"-Feld enthält die notwendigen Daten in einer vordefinierten Struktur, die aus Vereinfachungsgründen nicht die von der FIPA vorgesehene Content Language, die sich an KIF (Knowledge Interchange Format) (Genesereth 1991) anlehnt, implementiert, sondern ein eigenes einfaches Freitextformat darstellt.

```
AvAgentMessage answer =
    new AvAgentMessage("propose",
        item + " " + current.offer.id
        + " " + current.offer.price,
        msg.replyWith, null, "trade",
        msg.sender, current.offer.id,
        agentName, this);
```

Falls der Partner eine "refuse"- oder "reject"-Nachricht schickt, wird die Verhandlung sofort abgebrochen. Der Agent kehrt unverrichteter Dinge in den "idle"-Zustand zurück und beginnt den Kreislauf von vorn. Die semantische Bedeutung des Sprechaktprimitivs "refuse" ist "the action of refusing to perform a given action, and explaining the reason for the refusal" (FIPA 1997, S. 32), d.h. dass eine Verhandlung überhaupt nicht aufgenommen wird und keine Preisinformation an den fragenden Agenten übermittelt wird. In der derzeitigen Version von AVALANCHE steht kein strategisches Interesse hinter dem Absenden einer "refuse"-Nachricht, sondern sie ist allein ein Zeichen dafür, dass das Gut bereits verkauft wurde und die Information des Verzeichnisdienstes veraltet war.

```
AvAgentMessage answer =
    new AvAgentMessage("refuse", msg.content,
        msg.performative, null,
        "trade", msg.sender, null,
        agentName, this);
```

Im Falle eines erhaltenen "propose" mit einem Wucherangebot (höher als der doppelte Marktpreis), bricht der Agent mittels des Aussendens einer "reject-offer" Nachricht von sich aus die Verhandlung ab. Falls das Angebot des anderen Agenten jedoch zwischen 100% und 200% des Marktpreises liegt, oder falls der Partner seinen Preis im Vergleich zum letzten Mal nicht verändert hat, so entscheidet eine stochastische Probe gegen den `p_satisfaction`-Parameter, welche Handlung gewählt wird. Falls eine Ablehnung erfolgt (mit der Wahrscheinlichkeit $1 - p_satisfaction$), so sendet der Agent ein *reject-offer*, passt aber seinen Marktpreis entsprechend an. Anderenfalls passt er erst einmal nur den Marktpreis an und wechselt in den Zustand der Angebotsformulierung (*formulate offer*). Er testet dort gegen seinen eigenen *Acquisitiveness*-Wert, ob er selbst seinen Preis senken wird. Mit der Wahrscheinlichkeit `p_acquisitiveness` wird er den Wert seines letzten Angebotes beibehalten, anderenfalls wird er wie oben angeführt den Gebotspreis um einen bestimmten Satz erhöhen. Nach Absen-

dung der *"propose"*-Nachricht mit einem von beiden Preisen wartet der Agent erneut auf ein Gegenangebot des Verhandlungspartners.

Falls der Verkäuferagent sich entscheidet, die Verhandlung weiterzuführen, wird seine *"propose"*-Nachricht zuerst durch den Käufer verarbeitet. Abhängig von der Preissetzung des Verkäufers (siehe oben bei der Vorstellung des Parameters `p_satisfaction`) wird der Käuferagent seinerseits mit einem *"propose"* antworten oder die Verhandlung mit einem *"reject-offer"* abbrechen. Im Inhalt des *"reject-offer"* wird die Nachricht, auf die geantwortet wird, wiederholt. Daraufhin werden beide Agenten auf den Zustand „idle“ zurückgesetzt und können sich auf dem Markt neu orientieren.

```
AvAgentMessage answer =
    new AvAgentMessage("reject-offer", msg.content,
        msg.replyWith, null, "trade",
        msg.sender, null, agentName,
        this);
```

Im Verhandlungsfall folgen auf ein *"propose"* wechselseitig weitere *"proposes"*, in denen sich Käufer und Verkäufer in ihren Preisvorstellungen langsam einander annähern. Schließlich sendet einer von beiden ein *"accept-offer"*, mit dem die Annahme des Angebots signalisiert wird.

```
AvAgentMessage answer =
    new AvAgentMessage("accept-offer",
        msg.content, msg.replyWith,
        null, "trade", msg.sender,
        current.offer.id, agentName, this);
```

Zur Sicherstellung einer korrekten Abwicklung des Güter- und Zahlungsflusses muß die Nachricht *"accept-offer"* durch den anderen Agenten bestätigt werden. Der Verhandlungspartner antwortet mit einem *"commit"* – jetzt erst ist die Transaktion abgeschlossen.

```
AvAgentMessage answer =
    new AvAgentMessage("commit", msg.content,
        msg.replyWith, null,
        "trade", msg.sender,
        current.offer.id,
        agentName, this);
```

Zur internen Verarbeitung des aktuellen Zustandes ist hinzuzufügen, dass die Information über den gegenwärtigen Stand der Verhandlung von den Agenten in einem `BasicNegotiation`-Objekt hinterlegt wird, welches wie folgt definiert ist:

```
public class BasicNegotiation implements java.io.Serializable {
    public AvOffer offer;        // current offer for this item
    public AvOffer prevOffer;    // previous offer for this item
    public double lastPriceOffer =
        0.0;                    // last offer of trade partner, used to see if
                                // there's a change in her offer
    public double stepSize =
        0.0;                    // the step size we decided to use in this
                                // negotiation
    public int iteration;        // # of back and forth offers
}
```

Diese Information ist notwendig, um die Historie der Angebote der Gegenseite aufzunehmen und den Fortschritt bei Erhalt eines neuen Gebotes bewerten zu können. Die Variablen sind relativ selbsterklärend, `stepSize` wurde bereits weiter oben in diesem Abschnitt erläutert, `iteration` dient der Auswertung des Interaktionsverhaltens der Agenten.

4.6 Klassen und Methoden des Marktplatzes

Marktplätze sind prinzipiell ORB-Prozesse von VOYAGER (siehe Glossar). Sie werden zu Beginn jedes Experiments gestartet und nach Beendigung des Experiments wieder heruntergefahren. Jeder ORB-Prozess ist durch die Angabe der IP-Adresse des Computers (z.B. 132.230.16.32) und eine eindeutige Portadresse (z.B. 7777) eindeutig und unterscheidbar ansprechbar (z.B. 132.230.16.32:7777). Auf jedem Marktplatz gibt es eine einzige Instanz der Klasse `AvLocationAgent`. Dieses Objekt stellt für die Agenten einen Verzeichnisdienst sowie Hilfsmethoden zur Unterstützung der Interaktion bereit, ohne in Ergebnisse der ökonomischen Koordination einzugreifen.

Um ein Netzwerk von Lokationen aufbauen zu können, stellen die `AvLocationAgents` die Methoden `link()` und `unlink()` bereit, die von der Experimentsteuerung während des Aufbaus der Umgebung genutzt werden. Prinzipiell ist es möglich, über den Aufruf dieser Methoden zur Laufzeit ein dynamisches, sich ständig änderndes Netzwerk zu erzeugen.

Der Verzeichnisdienst des `AvLocationAgent`-Objekts wird von den `AvTradeAgent`-Objekten benutzt, um sich auf dem Marktplatz an- und abzumelden (über die Methoden

`register()` und `deregister()`), so dass andere Agenten ihnen Nachrichten senden können. Weitere Hilfsdienste, die der `AvLocationAgent` bietet, sind

- die Zeitnahmemethode `tick()`, die in gleichen Zeitabständen die `payCostsOfLife()`-Methode der eingetragenen `AvTradeAgents` aufruft.
- der Vektor `newAgentQueue`, in dem durch den evolutionären Algorithmus ST-DEA erzeugte `AvTradeAgents` so lange gespeichert werden, bis die Summe der von den `AvTradeAgents` erhaltenen Zahlungen aus der Methode `payCostsOfLife()` einem festgelegten Startkapital (`endowment`) entspricht. Anschließend tritt der neue `AvTradeAgent` in den Marktplatz ein.
- die Methode `shutdown()`, die allen `AvTradeAgents` das Abschalten des Marktplatzes signalisiert und ihnen die Möglichkeit gibt, diese vorher zu verlassen.

Der Verzeichnisdienst der `AvLocationAgent`-Objekte ist mittels eigener Methoden implementiert. `AvTradeAgents` senden Nachrichten an den `AvLocationAgent`, um sich in das Verzeichnis einzutragen, den eigenen Eintrag zu verändern, oder nach möglichen Verhandlungspartnern zu suchen.

Ein Agent, der neu auf einem Marktplatz eintritt, kündigt mittels der bestätigungslosen Nachricht *advertise* gegenüber dem `AvLocationAgent` an, dass er einen Bestand an Endprodukten hat, den er verkaufen möchte.

```
AvAgentMessage msg =
    new AvAgentMessage("advertise",
        outputFactor.type, null,
        null, "trade", null,
        outputFactor.type,
        agentName);
```

Der Käuferagent sendet hingegen eine Bedarfsmeldung an den `AvLocationAgent` des Marktplatzes und wird daraufhin ebenfalls in das Verzeichnis eingetragen.

```
AvAgentMessage msg =
    new AvAgentMessage("check",
        outputFactor.type, null,
        null, "trade", null,
        outputFactor.type,
        agentName);
```

Diese passive Eintragung in das Verzeichnis wird begleitet durch ein aktives Suchen nach Käufern und Verkäufern von allen Agenten. Die Aufeinanderfolge der Nachrichten für die

Abfrage eines Käufers ist in Abbildung 24 als Sequenzdiagramm gezeigt, die Abfrage eines Verkäufers stellt sich analog dazu dar.

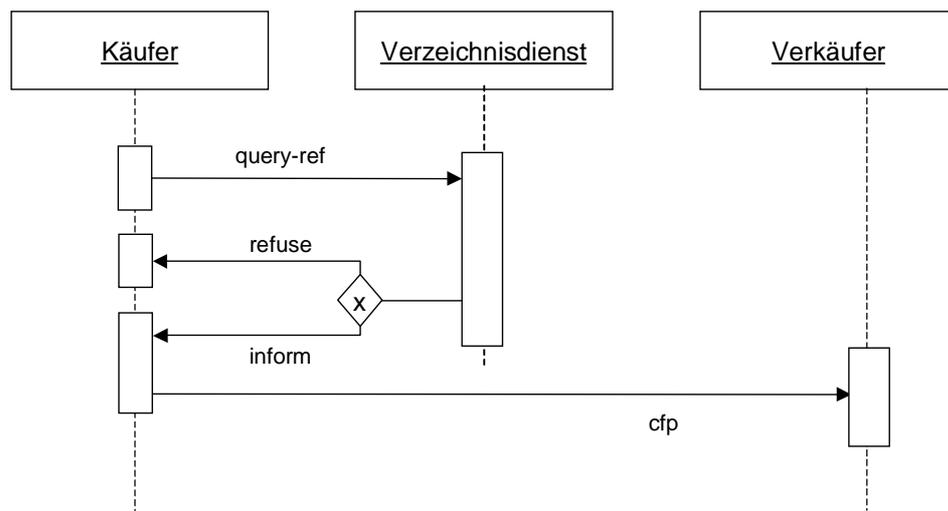


Abb. 24: AUML-Sequenzdiagramm der Verzeichnisdienstabfrage

Auf eine Anfrage „*query-ref*“ eines Agenten sendet der *AvLocationAgent* ihm die Liste der gerade registrierten Agenten, die diesen Gütertyp je nach Anfrageart entweder suchen oder anbieten. Daraufhin wird der Käuferagent zufällig oder gezielt Agenten aus dieser Liste mittels einer "*cfp*"-Nachricht ansprechen und einen neuen Verhandlungsversuch, wie in Kapitel 4.5.8 beschrieben, starten.

An dieser Stelle sei in Bezug auf die Kommunikation zwischen *AvTradeAgent* und Verzeichnisdienst festgestellt, dass die Implementierung mit einem Intermediär (*AvLocationAgent*) nicht zwingend notwendig ist. Einerseits wird das hier verfolgte strikt dezentrale Implementationskonzept verletzt, andererseits wird der Marktplatzbetreiber selbst nicht als autonomer Agent behandelt, da er keine freie Entscheidungsgewalt besitzt, ob er seine Informationen weitergeben soll. Ein direktes Hineinschreiben und Auslesen eines öffentlich verfügbaren Verzeichnisdienstes wird für zukünftige Implementierungen angestrebt.

4.7 Datentypen und Methoden der Experimentsteuerung

Die Klassen der Pakete "**.generator*" und "**.infoserver*" dienen der Steuerung und Auswertung von AVALANCHE als experimentellem MAS..

Die Klassen, die im Paket `"*.generator"` zusammengefaßt sind, dienen dem Start des Experiments. Zum Starten der Experimentierumgebung wird an die Klasse `G1` (für "Generator 1") eine Menge von Kommandozeilen-Parametern übergeben, welche den Aufbau eines bestimmten Experimentierlaufs bestimmen. Die Klasse `G1` starten nacheinander ein `AvInfoServer`-Objekt, ein oder mehrere Marktplatz-Objekte (`AvLocationAgent`) und alle Agenten-Objekte. Nähere Informationen zu den Kommandozeilenparametern finden sich im Anhang. Ein beispielhafter Aufruf des Generators sieht etwa so aus:

```
java de.unifreiburg.iig.avalanche.generator.G1 -n 50 -t 600 -g
0.9 0.25 0.15 0.75 0.2 1 -f test
```

In diesem Fall wird ein Experiment mit 50 Agenten pro Typ gestartet, insgesamt also 250 Agenten. Die Laufzeit soll 600 Sekunden betragen. Alle Agenten werden mit dem gleichen Parameterset für die Verhandlungsstrategie gestartet (siehe Kapitel 4.5.6). Dieses Vorgehen führt zu einer homogener Agentenpopulation. Alle Logdateien besitzen einen Dateinamen, der wie `test*.*` aufgebaut ist (z.B. `test.xls` für die zentrale Logdatei). Wenn, wie hier, nichts anderes angegeben wird, wurde genau ein `AvLocationAgent`, auf der voreingestellten Adresse `localhost:8000` gestartet. Nachdem alles gestartet ist, schläft der Prozeß die als Laufzeit festgelegte Zeitdauer. Nach dem Aufwachen ruft der `G1`-Prozeß im Infoserver-Objekt die Methode `shutdown()` auf, welche dann kaskadierend für die saubere Beendigung aller Marktplatz- und Agentenprozesse sorgt.

Während des Experiments geben die Agenten über Objekte der Klassen des Pakets `"*.info-server"` Informationen über ihren aktuellen Zustand aus. Dafür sind folgende Varianten vorgesehen:

- An verschiedene Logdateien werden Datensätze im Falle der Neuentstehung (Geburt) eines `AvTradeAgent`-Objektes, der Entfernung aus der Experimentalumgebung (Tod) und der erfolgreichen Durchführung einer Handelstransaktion angehängt. Diese Dateien werden nach Beendigung des Experiments ausgewertet.
- Während der Laufzeit können bestimmte aktuelle Daten (z.B. erzielte Einkaufs- und Verkaufspreise, Bargeldbestand, Lagerbestand an Input- und Outputgütern) der `AvTradeAgent`-Objekte tabellarisch in einem Statusfenster angezeigt werden.
- Schließlich ist es möglich, während der Laufzeit des Experiments Fehlermeldung über die Kommandozeile ausgeben zu lassen.

Die von den `AvTradeAgents` dem `AvInfoServer` zugesandten Informationen sind in der `AVALANCHE`-eigenen Klasse `AvInfo` kodiert. Diese stellt einen Vektor dar, dessen einzelne Felder den Spalten des Statusfensters entsprechen: `agentName`, Barvermögen (`cash`), Lagerbestand an Inputgütern (`inputUnits`) und Outputgütern (`outputUnits`), den letz-

ten Transaktionspartner (`partner`), den aktuellen (`location`) sowie den vorherigen (`lastLocation`) Marktplatz, sowie den letzten Einkaufs- (`inputPrice`) und Verkaufspreis (`outputPrice`).

Eine weitere Aufgabe der Experimentsteuerung ist die Zeitnahme. In einem Objekt der Klasse `Clock` läuft während des Experiments eine Stoppuhr, welche alle 10 Sekunden die `tick()` – Methode der Agenten aufruft. Beim Schreiben von Informationen in die Logdateien wird durch dieses Objekt jeweils eine Zeitmarke hinzugefügt, damit der zeitliche Ablauf konsistent dargestellt bleibt.

Da die Klassen der Experimentsteuerung zur Darstellung der Koordinationsleistung dienen, diese aber selbst nicht beeinflussen, werden sie im Haupttext dieser Arbeit nicht weiter beschrieben. Notwendige technischen Informationen dazu finden sich aber im Anhang.

4.8 Vorläufige Ergebnisse der Realisierung

Die Funktionsfähigkeit der Realisierung von `AVALANCHE` lässt sich auf den beiden Ebenen der Software-Agenten und der Experimentierumgebung beurteilen.

Die technologische Basis der Software-Agenten mit der Programmiersprache `JAVA` und einem `CORBA`-ähnlichen Verzeichnisdienst ist für die Entwicklung als ausreichend zu beurteilen. Untersuchungen zum Laufzeitverhalten der Agenten ergaben, dass fast die Hälfte der Rechenzeit für die Kommunikation aufgewendet wird, insbesondere die Generierung und Interpretation der Zeichenfolgen (`Strings`) der Sprechakte, aber auch Anfragen an den Verzeichnisdienst. Die versuchsweise Implementation einer `Public-Key-Infrastruktur` zur Verschlüsselung und Signierung von Nachrichten führte zu einem um den Faktor 10-20 schlechteren Laufzeitverhalten. Aufgrund des experimentellen Aspektes von `Avalanche` können diese technischen Probleme im Folgenden vernachlässigt werden. Für die Entwicklung realistischer agentenbasierter Märkte zeigen sie aber Notwendigkeiten der softwaretechnischen Weiterentwicklung auf.

In Bezug auf die Experimentierumgebung lässt sich festhalten, dass aufgrund der nicht-deterministischen Ausgestaltung der Agenten nicht bei jedem Experiment die gleichen Entwicklungen eintreten. Aus technischen Gründen der Prototypimplementation passiert es zum einen leider häufiger, dass einzelne Prozesse (Agenten) abstürzen oder den Fortgang des Experiments blockieren. In wirtschaftlicher Hinsicht kommt es zweitens z.B. häufiger vor, dass die Holzfäller-Agenteninstanzen nicht genügend Gewinn (in absoluten Geldeinheiten) erwirtschaften und durch die Dynamik des Experiments relativ schnell ohne ausreichendes Bargeld dastehen (und dadurch das Experiment abbricht). Ein starker Unsicherheitsfaktor ist die absolute Anzahl an handelnden Agenten. Da diese durch die Konzepte eines offenen Marktplatzes und die

Arbeitsweise des evolutionären Algorithmus nicht begrenzt ist, sind bei gleicher Ausgangssituation jeweils eine unterschiedliche Anzahl an Agenten im Markt tätig. Damit ergibt sich gleichzeitig auch ein Meßproblem, welches die Ermittlung von auf die Größe der Population basierenden bzw. Pro-Kopf-Kennzahlen erschwert. Andererseits zeigen die im nächsten Kapitel dargestellten Experimente z.B. bei der Güterpreisentwicklung ähnliche Muster, die unabhängig von der Zahl der Agenten in vergleichbarer Weise auftreten.

Inwiefern eine marktliche Koordinationsleistung überhaupt erreicht wird, kann durch die Feststellung der technischen Funktionsfähigkeit des MAS nicht beurteilt werden. Aus diesem Grund beschäftigt sich das folgende Kapitel mit der Frage, ob das dynamische Verhalten des MAS überhaupt ein koordiniertes Handeln der Agenten zeigt und welche Parameter die Koordination beeinflussen.

5 AVALANCHE als Experimentierumgebung für marktliche Koordination

In diesem Kapitel wird AVALANCHE als Mittel verwendet, um bestimmte Aspekte des Koordinationsverhaltens agentenbasierter elektronischer Märkte in experimenteller Weise zu erforschen. Das dynamische Verhalten eines realistischen Prototyps, verglichen mit der Koordinationsleistung von Menschen in Märkten, ist dabei von besonderem Interesse, wie z.B. Kephart et al. (2000) in einem vergleichbaren Projekt betonen:

„While [software agents] may be more expert at certain narrowly-defined tasks, they are likely be less generally capable and flexible than we are. Before unleashing them on the world's economy, it is essential that we consider their economic incentives and behaviors very seriously, and we must use every available means to anticipate their collective interactions with one another and with us” (Kephart et al. 2000, S.4).

In AVALANCHE werden mittels eines Generatorprogramms, welches im Anhang dieser Arbeit beschrieben wird, nacheinander ein AvInfoServer-Objekt als Experimentiersteuerung, ein AvLocationAgent als Marktplatzbetreiber und Verzeichnisdienst, sowie insgesamt zwischen 50 und 250 Agenten der Klassen AvTradeAgent, AvProducerAgent und AvConsumerAgent gestartet. Durch das Setzen von Kommandozeilenargumenten beim Aufruf lässt sich die Startverteilung der Strategieparameter steuern. Das Kapitel beginnt mit relativ einfachen Parameterkombinationen und führt darauf aufbauend komplexere Experimente durch.

Die folgenden Darstellungen zeigen bestimmte Muster, die durch wiederholtes Experimentieren mit den gleichen Ausgangswerten immer in ähnlicher Form auftreten. Die absoluten Preishöhen oder der Zeitpunkt des Eintritts einer Entwicklung hat keine Aussage, da sich diese Werte bei gleichen Startparametern jedesmal ändern. Von Bestand und damit einer gewissen Interpretationsfähigkeit sind relative Werte, z.B. die Preisrelationen der Güter zueinander.

Im folgenden wird von einfachen Experimenten ausgegangen und die Komplexität der Fragestellungen und damit auch die Interpretation der Muster schrittweise erhöht. Es beginnt mit einer Population homogener Agenten, die exakt gleiche Strategien aufweisen und diese auch nicht verändern. Im nächsten Schritt (5.2) wird gezeigt, welche Auswirkungen eine identische Veränderung der Konzessionsbereitschaft in allen Agenten produziert. Davon ausgehend wird gezeigt (5.3), wie unterschiedliche Konzessionswahrscheinlichkeiten pro Typ die Güterpreise und die Gewinner der Agenten beeinflussen.

Bis zu diesem Punkt findet noch keine adaptive Anpassung der Strategien in den Agenten statt. In Kapitel 5.4 werden die Auswirkungen der Koevolution auf den Markt gezeigt, wenn zusätzlich zu dem in 5.3 dargestellten Problem die Anwendung des evolutionären Algorithmus STDEA die Zusammensetzung der Agentenpopulation verändert. Kapitel 5.5 zeigt erneut Koevolution, dieses Mal von einer zufälligen Verteilung der Startpreise ausgehend. In Tabelle 3 wird die zunehmend komplexer werdende Aufeinanderfolge der Experimente dargestellt.

Tabelle 3: Experimentfolge

Kapitelnummer und Grafik	Konzessionsbereitschaft (p_acq)	Konzessionshöhe (del_chg)	Zufriedenheit (p_sat)	Adaption durch Evolutionären Algorithmus	Startpreise
5.1	homogen über die gesamte Population	homogen über die gesamte Population	homogen über die gesamte Population	nein	vorgegeben
5.2	homogen über die gesamte Population	homogen über die gesamte Population	homogen über die gesamte Population	nein	vorgegeben
5.3	homogen innerhalb eines Agententyps	homogen über die gesamte Population	homogen über die gesamte Population	nein	vorgegeben
5.4	Zufallsverteilung	homogen über die gesamte Population	homogen über die gesamte Population	ja	vorgegeben
5.5	homogen über die gesamte Population	homogen über die gesamte Population	homogen über die gesamte Population	nein	zufällig
5.6, Abb. 39	homogen über die gesamte Population	Zufallsverteilung	homogen über die gesamte Population	ja	vorgegeben
5.6, Abb. 42 und 43	Zufallsverteilung	Zufallsverteilung	homogen über die gesamte Population	ja	vorgegeben
5.6, Abb. 44	Zufallsverteilung	Zufallsverteilung	homogen über die gesamte Population	ja	zufällig
5.6, Abb. 45	Zufallsverteilung	Zufallsverteilung	Zufallsverteilung	ja	zufällig

5.1 Veränderungen der Güterpreise

In diesem Abschnitt soll zuerst als Referenzpunkt die marktliche Koordination dargestellt werden, die sich aus dem Zusammenspiel homogener Agenten ergibt. Dabei sind sämtliche AvTradeAgent-Objekte mit dem gleichen Genotype-Objekt ausgestattet.

Es wurden 50 Agenten pro Typ gestartet, d.h. 250 Agenten insgesamt. Die Laufzeit wurde auf 600 Sekunden festgelegt. Diese Einstellungen ergeben auf einem normalen Pentium-PC ausreichende Daten für eine Interpretation und minimieren gleichzeitig technische Probleme der Implementation.

Folgende Strategieparameter wurden fest vorgegeben: `p_acquisitiveness = 0.50`, `del_change = 0.25`, `del_jump = 0.15`, `p_satisfaction = 0.75`, `w_memory = 0.2`, und `p_reputation = 1`. Alle Agenten sind in ihrer Strategie homogen, es gibt keine Variierung der Werte in der Population. Es gibt keine Lagerbestände in den Agenten, Güter kommen also nur über die Produzenten in den Markt hinein. Die Startpreise sind in den Agenten vorgegeben, so dass die ersten Transaktionen auch auf dieser Ebene stattfinden. Abbildung 25 zeigt deutlich, dass sich an dieser Voreinstellung auch nichts ändert, d.h. dass die Güterpreise sich weder nach oben noch nach unten verändern. Anhand dieser einfachen Darstellung, die noch keine großen Schlüsse auf die Dynamik des Koordinationsmechanismus zulässt, soll zuerst die Darstellung dieser Grafiken erläutert werden.

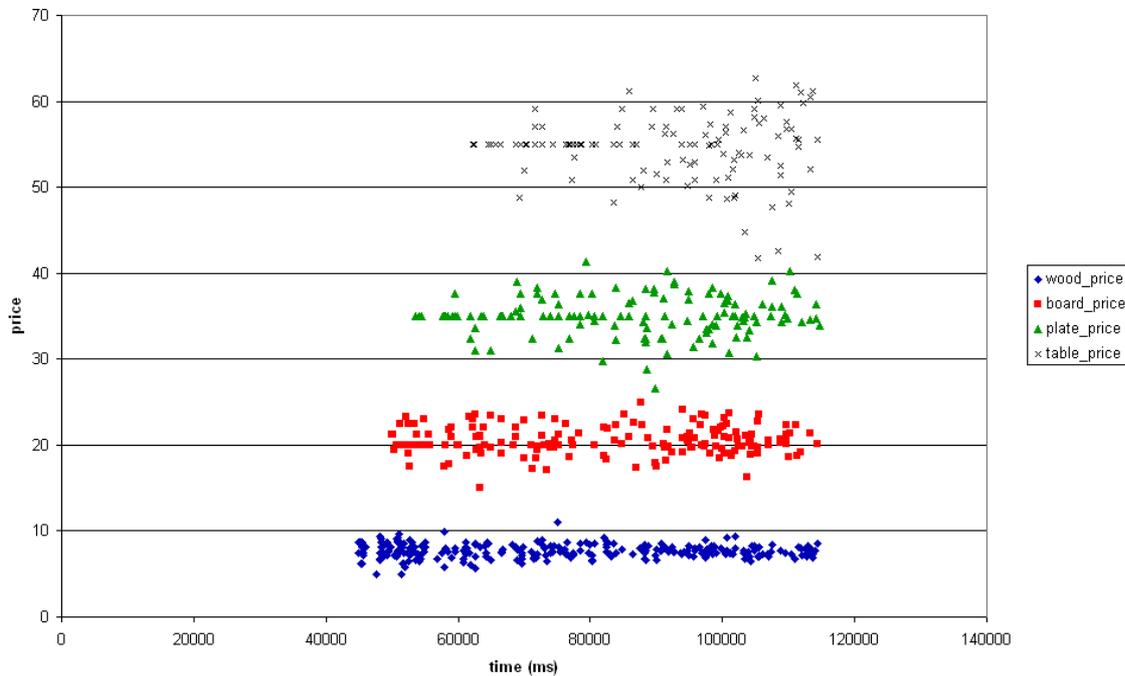


Abb. 25: Startphase mit festgelegten Preisen

Auf der Abszisse der Grafik 25 ist der zeitliche Ablauf des Experiments in Millisekunden abgetragen, auf der Ordinate ist eine Wertskala angezeigt, die die Güterpreise in Geldeinheiten misst. Jeder Punkt in der Grafik markiert eine abgeschlossene Transaktion, die durch den Verkäuferagenten in einem AvInfo-Objekt an den Infoserver kommuniziert und dort in einer Logdatei vermerkt wird. Der Typ des verkauften Gutes wird durch unterschiedliche Punktfarben dargestellt, die hier als Graustufen gedruckt werden. Die Legende zeigt unterschiedliche Symbole, die von oben nach unten die Wertschöpfungskette anhand der Güter zeigt (Holz: *wood*, Bretter: *board*, Platten: *plate*, Tische: *table*). In der Grafik ist diese Reihenfolge meist umgedreht, da *wood* für den geringsten Preis verkauft wird und *table* für den höchsten. Um die Darstellung handhabbar zu machen, wurde der höchste Preis, zu dem die Konsumenten-Agenten kaufen, auf 120 Geldeinheiten festgelegt. In einigen späteren Grafiken ergibt sich dort eine sichtbare Grenze, hier spielt diese Vereinbarung noch keine Rolle. Die Darstellung des zeitlichen Verlaufs beginnt nicht an der Nullmarke, da die Stoppuhr beim Starten der Experimentsteuerung beginnt und das Senden des Startsignals an die Agenten später erfolgt.

An der Grafik 26 zeigt sich klar das an sich nicht selbstverständliche Zustandekommen einer Koordinationsleistung. Ausgehend von einer Grundausstattung an Rohprodukten (Bäumen) werden über vier Stationen einer Wertschöpfungskette Endprodukte (Tische) hergestellt, ohne

dass sich ein chaotischer Verlauf in den Kurven zeigt. Zwar zeigt sich keine Konvergenz eines Güterpreises hin auf einen einzigen Gleichgewichtspreis, aber die dargestellten Transaktionspreise bleiben innerhalb eines sehr engen Bandes, vergleichbar mit den Ergebnissen der Koordination von Cliff (1997) bzw. Preist (1998), die in Kapitel 3 gezeigt wurden.

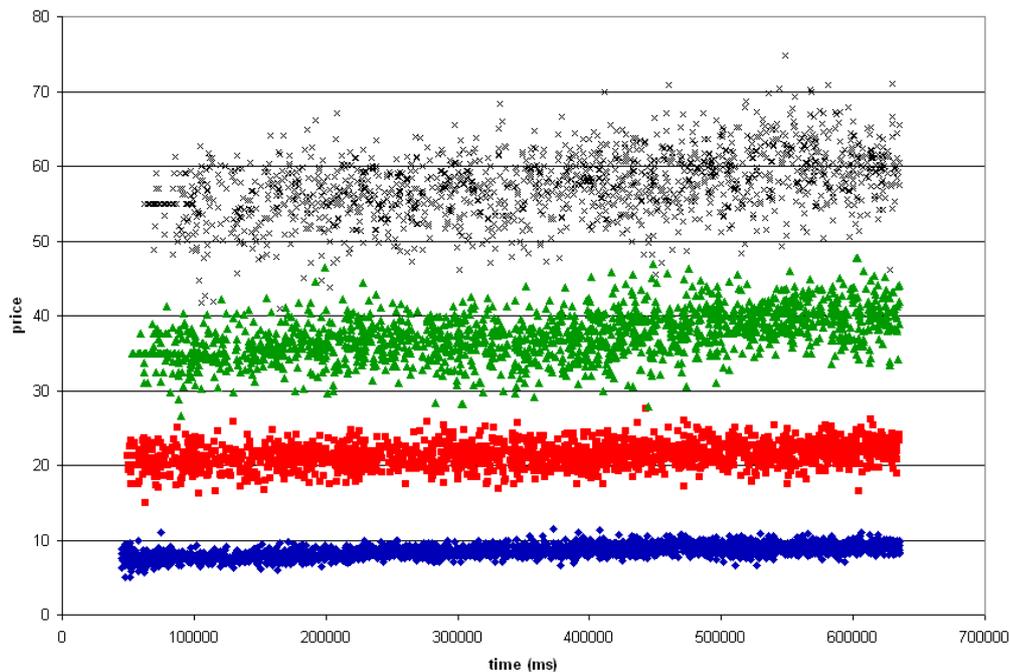


Abb. 26: Experimentverlauf bei festgelegten Startpreisen und homogener Strategieverteilung

Über den weiteren Verlauf des Experiments zeigt sich keine grundsätzlich verschiedene Bewegung der Güterpreise. Die einzelnen Güterpreiskurven bleiben im Verlauf des Experiments auf etwa der gleichen (Ausgangs-)höhe. Dabei ist die absolute Höhe des Startpreises nicht relevant und auch die Anzahl der Agenten insgesamt ist unerheblich. Da alle Agenten eine stochastisch gleiche Wahrscheinlichkeit besitzen, in der Verhandlung Konzessionen einzugehen, ist es keinem Agenten möglich, sich einen preislichen Vorteil herauszuarbeiten. Die Preise bleiben auf dem zu Beginn willkürlich gegebenen Startpreis, womit auch die Gewinne der Agenten im Zeitablauf gleichbleiben, wie in Abbildung 27 gezeigt.

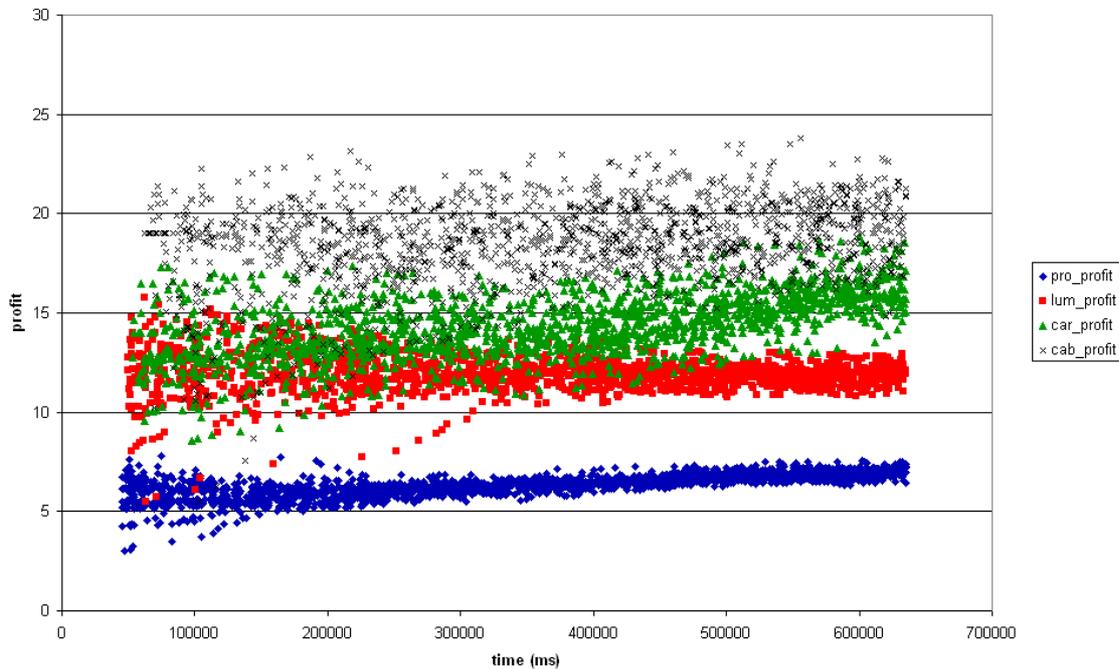


Abb. 27: Gewinnentwicklung bei festgelegten Startpreisen

Die Abbildung 27 ist zu der vorhergehenden Preisentwicklungsdarstellung etwas verändert zu lesen. Die Abszisse zeigt in gleicher Weise den zeitlichen Verlauf des Experiments in Millisekunden. Auf der Ordinate sind in Geldeinheiten die Gewinne der Agenten pro Agententyp abgetragen. Es fällt auf, dass sich die unterschiedlichen Wertschöpfungsstufen in der Höhe ihrer absoluten Gewinne pro Transaktion unterscheiden und zwar säuberlich nach ihrer Position in der Wertschöpfungskette getrennt. Dies kann dadurch erklärt werden, dass die Agenten sich in ihrem Gewinnstreben prozentual an den Marktpreisen ihrer Güter orientieren und dadurch solche Agenten, die mit geringwertigen Gütern handeln (Holz, Bretter) gegenüber den anderen Agenten (Platten, Tische) auch geringere absolute Gewinne einfahren.

Als Ergebnis kann festgehalten werden, dass der agentenbasierte dezentrale Marktplatz von AVALANCHE ähnliche Koordinationsergebnisse zeigt wie die zentrale Doppelauktion, die bei Cliff (1997) bzw. Preist (1998) eingesetzt wurde und die ihrerseits auf Marktexperimenten mit menschlichen Teilnehmern von Smith (1962) beruhen. Der Effekt der marktlichen Koordination ist miteinander vergleichbar, obwohl der Mechanismus grundsätzlich verschieden ist. Diese Erkenntnis dient gleichzeitig zur Einordnung der folgenden Experimente, die auf dieser Basis aufbauen.

5.2 Anzahl der Transaktionen

Im Zusammenspiel der bisher vorgestellten Parameter sollte sich in der dynamischen Koordination ein der menschlichen Koordination vergleichbares Ergebnis einstellen. In Experimenten mit menschlichen Teilnehmern kann die Entwicklung beobachtet werden, dass eine höhere Konzessionsrate zu häufigeren Transaktionen führt:

"One [logical conclusion] is that lower initial demands and faster concessions on the part of one or both parties make agreement more likely and more rapidly reached. This can be logically deduced because lower demands mean that the parties are less likely to become discouraged and break off negotiation and that their demands will coincide at an earlier point in time." (Pruitt 1981, S.20).

Diese Situation kann in AVALANCHE nachgestellt werden, indem lediglich der *Acquisitiveness*-Parameter verändert wird. Die Anzahl der Transaktionen wird wesentlich durch das Verhältnis zwischen $p_{\text{acquisitiveness}}$ (kurz: p_{acq}) und $p_{\text{satisfaction}}$ (kurz: p_{sat}) bestimmt. Ein hoher p_{acq} -Wert verflacht die gedachte Steigung der Konzessionskurve, so dass ein Zugehen auf den Verhandlungspartner unwahrscheinlicher wird. Damit dauert die einzelne Verhandlung tendenziell länger, da mehr Runden bis zum Erreichen eines gemeinsamen Kompromisspreises durchschritten werden müssen. Je mehr Runden durchlaufen werden, desto höher ist die Wahrscheinlichkeit, aufgrund eines hohen p_{sat} -Wertes aus der Verhandlung auszusteigen. Bei einem p_{acq} -Wert von 0.9 wie in Abbildung sind auffällig wenige Transaktionen zu sehen, da die Agenten nur mit einer Wahrscheinlichkeit von 10% Konzessionen eingehen. Die Verhandlungen dauern wesentlich länger und werden auch häufiger aufgrund von Unzufriedenheit abgebrochen. Ein p_{acq} -Wert von 1.0 führt zu überhaupt keinen Transaktionen. Der in Abbildung 28 gezeigte Verlauf führt zum Abbruch des Experiments, da nach 49 Minuten alle Holzfäller-Agenten (*Lumberjacks*), mangels Transaktionsgewinnen bei andauerndem Abzug der Marktgebühren (`costsOfLife`), durch Verlust sämtlichen Barvermögens vom Marktplatz entfernt wurden. Insgesamt haben in diesem Experimentierlauf nur 366 Transaktionen stattgefunden.

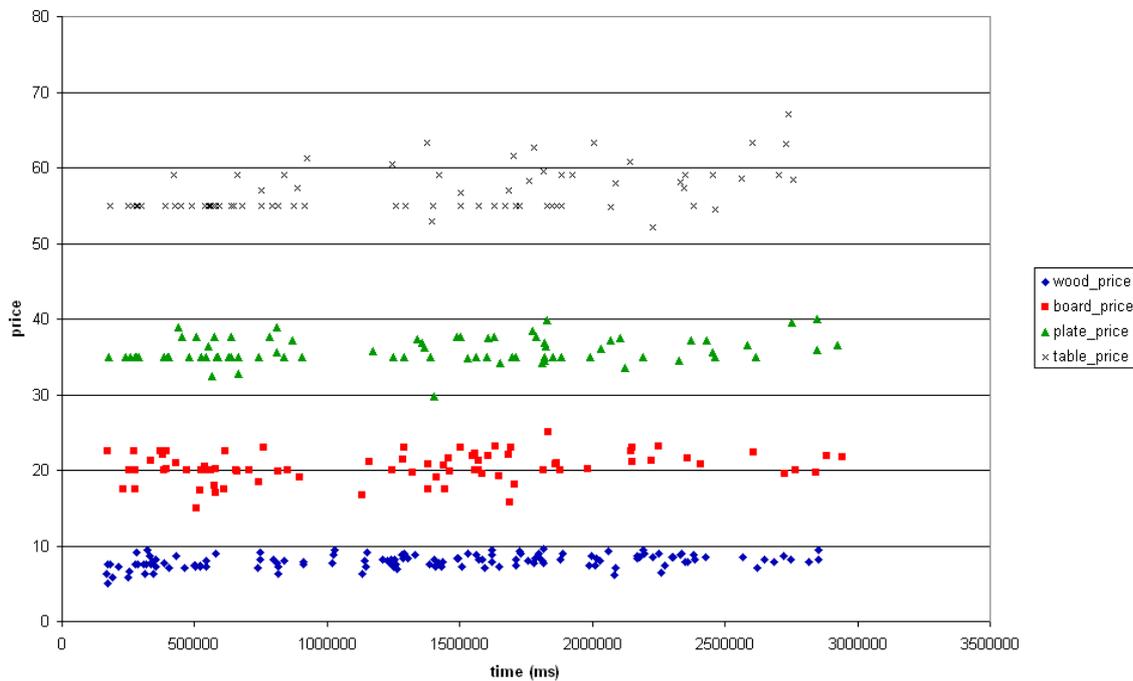


Abb. 28: Preisentwicklung bei einer *Acquisitiveness*-Vorgabe von 0.9

Ganz anders ist es, wenn die *Acquisitiveness* auf Werte von 0.5, wie in Abbildung 26 gezeigt, gesetzt werden. Die gesteigerte Konzessionierungsfreude der Agenten führt zu einer wesentlich höheren Zahl an Transaktionen im Zeitablauf (in dem Beispiel sind es 7115 Transaktionen), aber auch zu einem Verschwimmen der Preisebenen. Dies ist darauf zurückzuführen, dass sich die Agenten schnell in den Preisvorstellungen annähern, aber nicht gleichschnell. Damit ist eine Analogie zu dem Verhalten menschlicher Verhandlungspartner erreicht, die ebenfalls bei höherer Konzessionsfreude häufigere Transaktionen erreichen (vgl. Pruitt 1981, S. 20).

5.3 Gewinnentwicklung bei unterschiedlicher Konzessionsrate

Eine weitere Analogie ergibt sich aus dem Zusammenspiel von `del_jump` und `p_acquisitiveness`. In menschlichen Verhandlungssituationen ist empirisch folgendes Verhalten feststellbar:

"A second conclusion is that, if agreement is reached, a bargainer who makes larger initial demands and smaller concessions will achieve a larger outcome (and the other

part a correspondingly smaller outcome)" (Pruitt 1981, S. 20).

Um dieses Verhalten in AVALANCHE nachzubilden, wird das Experiment so verändert, dass die Agenten mit unterschiedlichen Strategien starten. In diesem Beispiel wird eine Initialisierungsdatei angelegt, die den Zimmermann-Agenten (*Carpenter*) einen p_{acq} -Wert von 0.6 gibt, allen anderen Agenten einen p_{acq} -Wert von 0.4. Die übrigen Parametereinstellungen bleiben gleich dem vorherigen Beispiel. Wie aus Abbildung 29 zu erkennen ist, schaffen es die Zimmermann-Agenten schnell, sich eine hohe Gewinnspanne zu sichern, drängen die anderen Agententypen nach oben und unten ab und vermindern auf diese Weise deren Möglichkeit, bessere Preise und höhere Gewinne realisieren zu können.

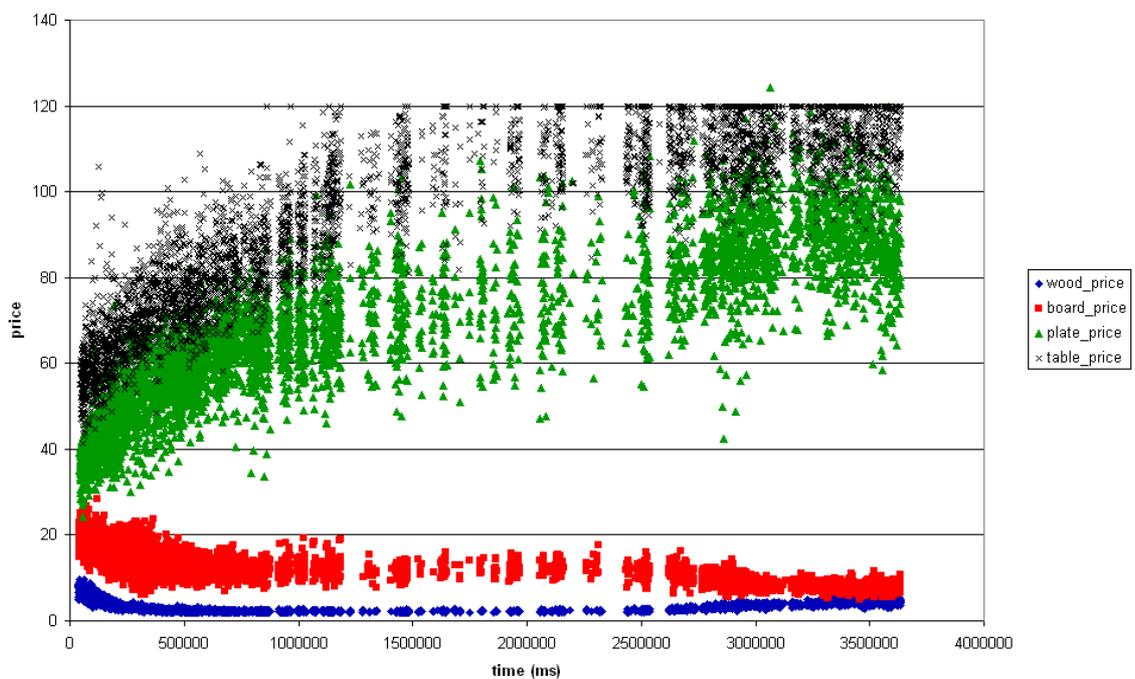


Abb. 29: Preisentwicklung mit hartnäckigen Zimmermann-Agenten³⁹

Diese Entwicklung tritt deswegen ein, weil z.B. die Holzfäller- und die Möbelschreiner-Agenten in höherem Maße als die Zimmermann-Agenten Konzessionen eingehen. In jeder Verhandlung zwischen einem Zimmermann und einem anderen Agenten wird der Kompromißpreis näher am Startpreis des Zimmermanns liegen als an dem des anderen Agenten. Damit schaffen

³⁹ Die deutlich sichtbaren Streifen sind Artefakte des Experiments, da sich Avalanche den Speicher und CPU des Computers noch mit weiteren Anwendungen teilt. In den leeren Zeiträumen findet kein Handel statt, die Prozesse ruhen und fahren nach dieser technischen Pause mit der Berechnung an der gleichen Stelle fort. Die Aussage der Experimente wird dadurch nicht beeinträchtigt.

es die Zimmermänner insgesamt, ihre Gewinnsituation entscheidend zu verbessern (vgl. Abbildung 30).

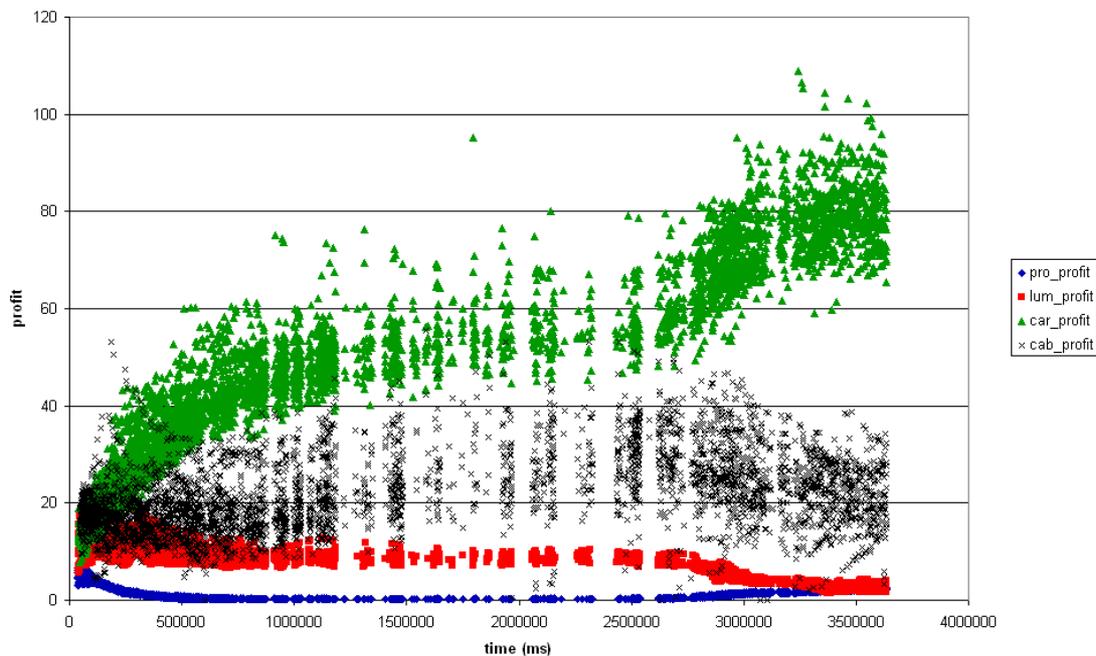


Abb. 30: Gewinnentwicklung bei hartnäckigen Zimmermannagenten

Deutlich sichtbar ist, dass nach einer noch relativ ausgewogenen Gewinnsituation zu Anfang alle anderen Agententypen auf Gewinne je Transaktion von unter 10 Geldeinheiten gedrückt werden. Konsequenterweise bricht das Experiment nach ca. 15 Minuten ab, da sämtliche Holzfäller oder Möbelschreiner in der Zwischenzeit ihr Vermögen vollständig verloren haben. Der empirische Befund von Pruitt aus menschlichen Verhandlungen, dass eine höhere Konzessionsrate zu niedrigeren Gewinnen führt, kann hier in der Tendenz bestätigt werden. Offensichtlich scheint in diesem Fall eine geringe Konzessionsneigung eine dominante Strategie im Sinne der Spieltheorie zu sein, sofern die anderen Agenten eher Konzessionen eingehen. Falls alle Agenten auf den Gedanken kommen, ihre Konzessionsrate zu verringern, ergibt sich im Endeffekt das traurige Bild von Abbildung 28. Dabei gelangen die Agenten nur selten zu einer Transaktion und letztendlich müssen alle Agenten mangels ausreichendem Einkommen den Markt verlassen; die Koordination als solche bricht zusammen.

5.4 Anpassung von Strategieparametern durch STDEA

Gibt es eine absolut "beste" Konzessionsrate, die nicht von den Umweltbedingungen und insbesondere den anderen Agenten abhängt? Um diese Fragestellung untersuchen zu können, wird das Optimierungs- und Lernverfahren des evolutionären Algorithmus von Smith/Taylor (1998), STDEA, angewendet.

Die Startparameter des Experimentierlaufs werden so gesetzt, dass der *Acquisitiveness*-Parameter mit einer bestimmten Verteilung auf die im übrigen homogenen Agenten verteilt wird. 50 Agenten jedes Typs werden gestartet, p_{acq} wird als kontinuierliche Zufallsvariable mit einem Mittelwert von 0.5 und einer Breite der Gleichverteilung von 0.9 erzeugt, die Werte liegen also über die Population verteilt mit gleicher Wahrscheinlichkeit innerhalb des Intervalls [0.05;0.95]. Durch den EA ergibt sich die Ausbildung eines bestimmten *Acquisitiveness*-Wertes im Mittel aller Agenten eines Typs, welche die Richtung der Optimierung anzeigt. Dies kann anhand der Abbildung 31 verdeutlicht werden. Jede Kurve zeigt die Entwicklung des arithmetischen Mittels des *Acquisitiveness*-Parameters jeweils eines Agententyps (*Lumberjack*, *Carpenter*, *Cabinetmaker*).

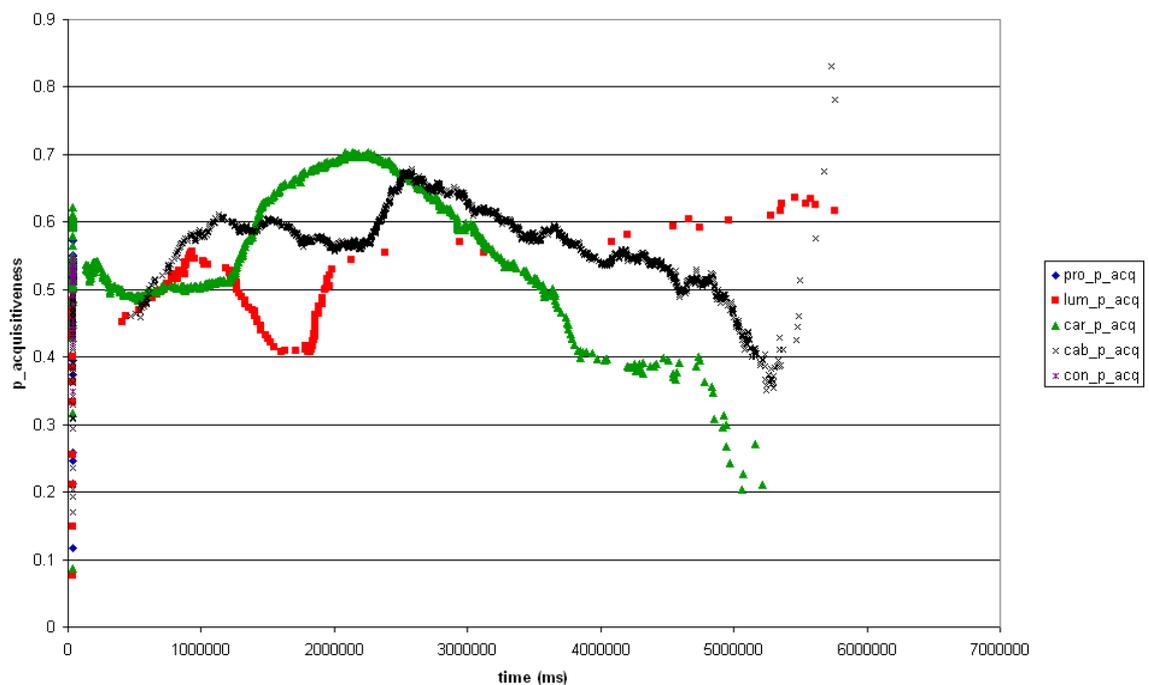


Abb. 31: Entwicklung des *Acquisitiveness*-Parameters durch den evolutionären Algorithmus ($p_{acq} = 0.5$)

An der Ordinate und in der Legende sind sogar fünf verschiedene Typen an Agenten zu erkennen, die drei genannten sowie die Produzent- und die Konsument-Agenten. Die Produzent- und Konsument-Agenten bleiben bei der Startverteilung ihrer Parameter stehen und entwickeln keine neuen Agenten; ihre Anzahl und Strategie bleibt über den gesamten Zeitablauf des Experiments gleich. Lediglich die inneren Agenten (*Lum*, *Car* und *Cab*) entwickeln sich weiter. Auch in der Abbildung 31 zeigt die Abszisse den zeitlichen Verlauf des Experiments in Millisekunden. Jeder Punkt der Grafik zeigt die "Geburt" eines neuen Agenten. Dessen aus der Anwendung des EA generierte *Acquisitiveness*-Wert verändert den Mittelwert des gleichen Parameters der existierenden Population dieses Typs. Ganz links an der Startlinie zeigt sich die Startausstattung der Population. An der farblichen Verteilung ist zu sehen, dass diese sich etwa über das Intervall $[0.1;0.8]$ erstreckt. Im Verlauf des Experiments werden keine neuen Produzenten und Konsumenten erschaffen, so dass nur 3 Kurven im weiteren zu sehen sind. Nach etwa 100 Sekunden sieht man eine annähernd gleiche Entwicklung aller Agententypen. Nach 200 Sekunden zeigen sich aber deutliche Unterschiede, die aber nicht von Dauer sind. Über den Verlauf des Experiments schwingen die Kurven hoch und runter, es ist eine Koevolution zu verzeichnen. Dies wird in Abbildung 32 noch deutlicher. In diesem Fall ist der *Acquisitiveness*-Wert etwas höher ($p_{acq} = 0.6$), d.h. die Konzessionsneigung ist geringer und es werden weniger Transaktionen durchgeführt.

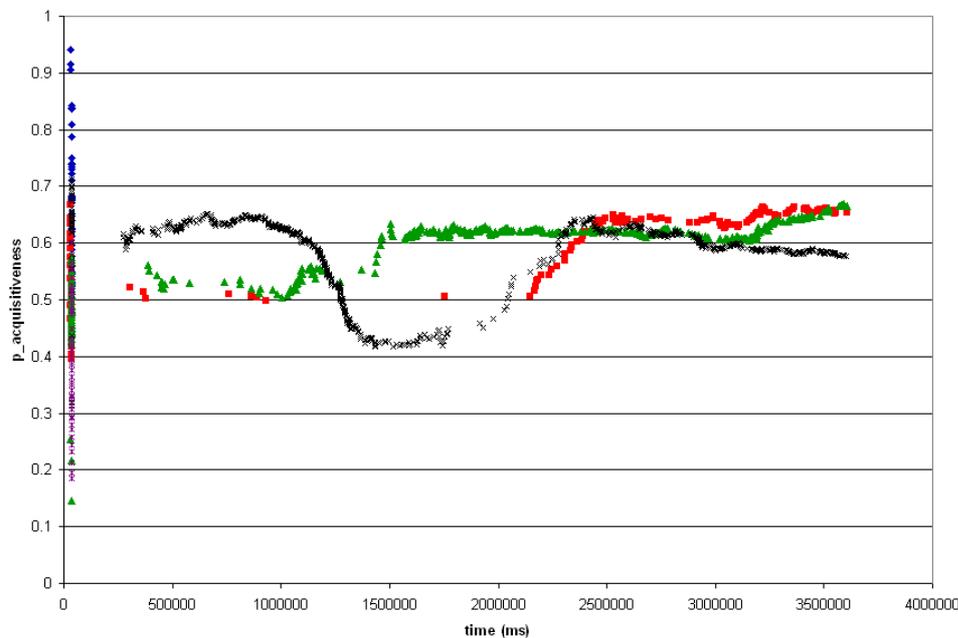


Abb. 32: Entwicklung des *Acquisitiveness*-Parameters durch den evolutionären Algorithmus ($p_{acq} = 0.6$)

Auch hier beginnen die einzelnen Kurven auf etwa dem gleichen Wert zwischen 0.5 und 0.6, es zeigt sich aber eine stetigere Entwicklung des *Acquisitiveness*-Parameters bis zu einem gemeinsamen Wert bei etwa 0.65 für alle Typen. Bis dahin ist aber besonders die Entwicklung der Kurve der Möbelschreiner bemerkenswert. Beginnend mit einer im Mittel höheren *Acquisitiveness* durch die zufällige Anfangsverteilung schwenkt dieser Wert deutlich unter die Kurve der anderen Agenten, um sich zum Ende zu erholen. Dies ist Ausdruck einer Koevolution dieses Agententyps, der zuerst hohe Gewinne einfahren kann. Dabei sind jedoch diejenigen Agenten am erfolgreichsten, die im Verhältnis zum Durchschnitt dieses Typs eine geringe *Acquisitiveness* aufweisen, da sie häufiger zum Abschluss gelangen. Dadurch senkt sich die Kurve zunächst. Wenn die *Acquisitiveness* der Möbelschreiner jedoch geringer ist als die der anderen Agententypen, so werden sie von diesen ausgebeutet. In diesem Fall sind diejenigen Möbelschreiner im Vorteil, die eine relativ höhere *Acquisitiveness* als ihre „Artgenossen“ zeigen. Auf diese Weise pendeln sich die *Acquisitiveness*-Kurven aller Agententypen auf einem annähernd gleichen Wert ein.

Eine Abwandlung dieses Experiments stellt der folgende Versuch dar (Abbildung 33). In diesem Fall wurden lediglich die Zimmermann-Agenten mit einer geringeren Wahrscheinlichkeit, Preiskonzessionen zu machen, initialisiert, d.h. mit einem hohen *Acquisitiveness*-Wert von $p_{acq} = 0.6$, während alle anderen Agenten bei $p_{acq} = 0.5$ initialisiert wurden. In der ersten Hälfte des Experimentierlaufs zeigt sich die erwartete Entwicklung, dass die Zimmermänner zu Ungunsten der anderen Agententypen ihren Nutzenzuwachs verbessern können.

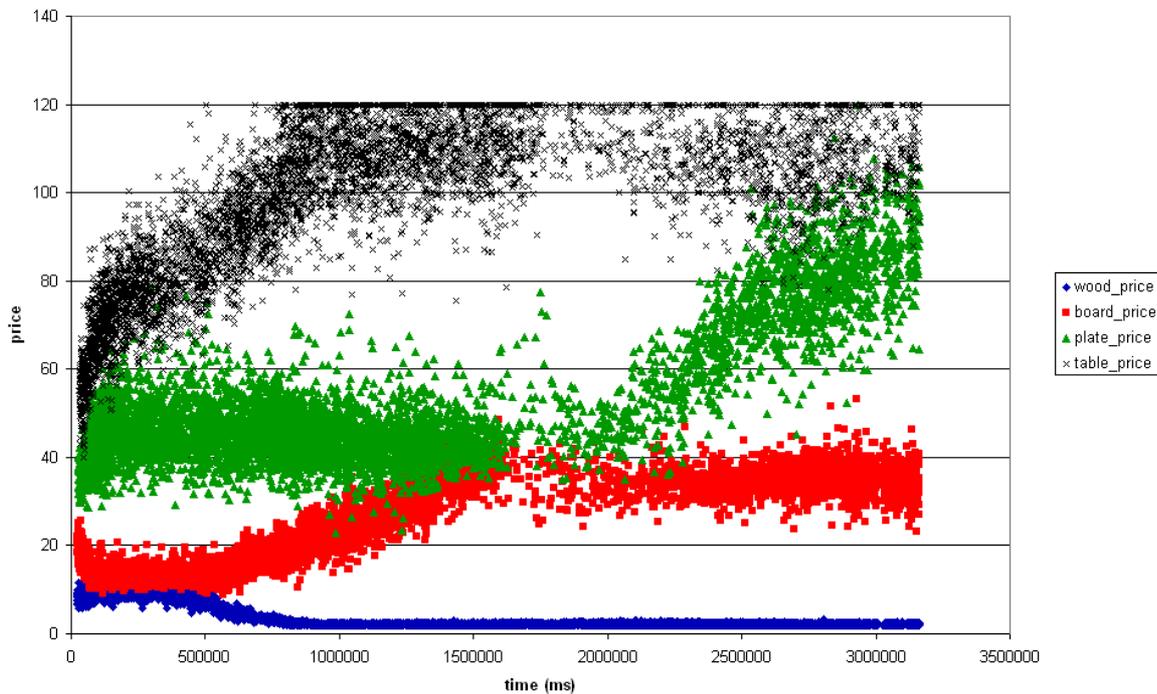


Abb. 33: Einschwingen der Preise bei Koevolution

Durch den STDEA treten im folgenden solche Holzfäller- und Möbelschreiner-Agenten auf, die ihrerseits eine hohe *Acquisitiveness* besitzen. Die Funktionsweise eines evolutionären Algorithmus legt generell eine Korrelation eines hohen *Acquisitiveness*-Wertes mit einem hohen Nutzenwert nahe. Zur gleichen Zeit treten zunehmend solche Zimmermann-Agenten dazu, die einen niedrigeren *Acquisitiveness*-Wert besitzen als die zu Anfang gestarteten. Auch diese Entwicklung ist verständlich, da Agenten mit einer höheren Konzessionsneigung höhere Umsätze tätigen. Im Effekt bewegen sich alle p_acq -Kurven in Abbildung 34 aufeinander zu. In dem hier gezeigten Experimentierlauf mündete diese Entwicklung tatsächlich in einem Zurückschwingen der Preiskurven auf die Anfangswerte, nachdem sowohl Holzfäller als auch Möbelschreiner ihren Druck auf die Preise erhöhen konnten. Diese Entwicklung ist ein klassisches Beispiel für eine Koevolution von Agenten innerhalb einer Population.

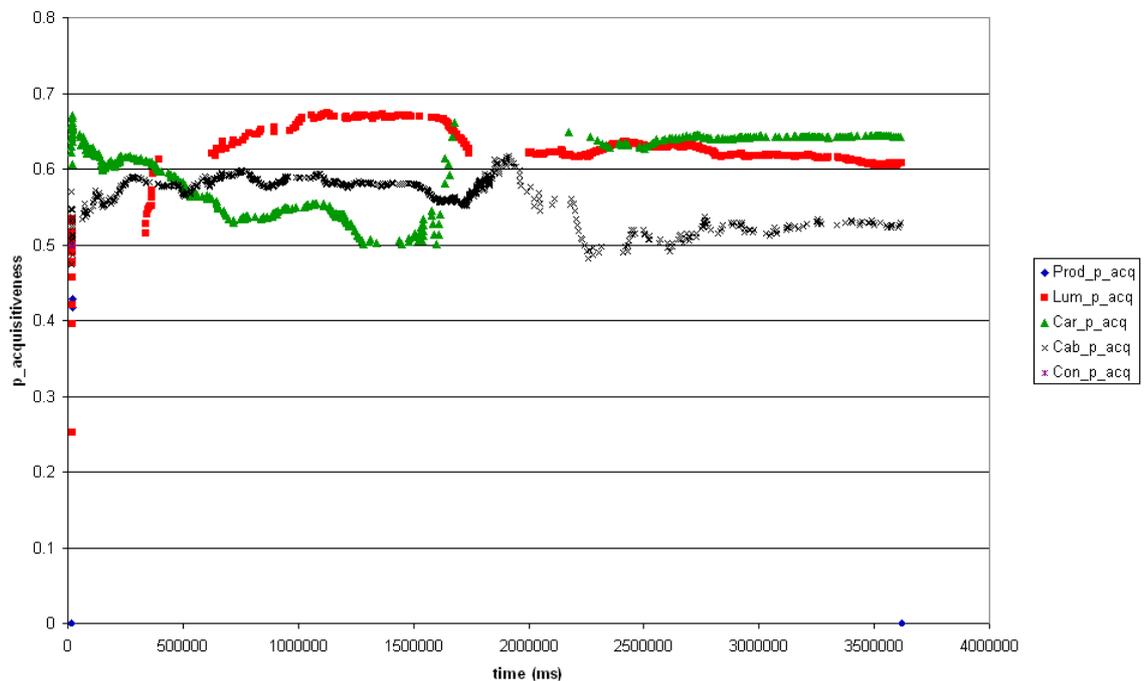


Abb. 34: Koevolution des *Acquisitiveness*-Parameters

Eine dritte Erkenntnis von Pruitt (1981, S. 21) aus den empirischen Untersuchungen lautete:

"Bargainers who demand little will usually reach agreements but achieve low profits. Those who demand a lot, will fail to reach agreement and hence also do poorly. With moderate demands, bargainers will often reach agreement at a good level of profit."

Dieses Verhalten wird offensichtlich auch in dem in diesem Abschnitt gezeigten Experimenten nachvollzogen. Jene Agententypen, die eine im Vergleich zu den anderen Agententypen zu hohe *Acquisitiveness* zeigen, orientieren sich im Zeitablauf an diesem niedrigeren Wert. In gleicher Weise führt der STDEA bei zu niedrigen *Acquisitiveness*-Werten im Zeitablauf zu einem Anwachsen, bis sich der Wert bei allen Agententypen auf ungefähr dem gleichen Wert einpendelt. Als Ergebnis lässt sich festhalten, dass ein der menschlichen Verhandlungsführung vergleichbarer Effekt marktlicher Koordination durch die in AVALANCHE codierte Art der Agentenstrategie erreicht werden kann.

5.5 Erreichen von Preisebenen aus zufälligen Startpreisen

Lässt man die Agenten mit dem gleichen Genotyp, aber vollkommen zufällig gewählten Startpreisen beginnen, so ergibt sich zu Anfang des Experiments ein chaotisches Bild zufälliger

Transaktionspreise, welches mit fortschreitendem Lernen der Agenten durch Selbstorganisation in die voneinander abgrenzbaren Preisebenen des vorherigen Bildes mündet. Die zufälligen Startpreise der Agenten werden durch folgenden Programmausschnitt berechnet:

```
//generate two random prices, higher one will become sell-level,  
//lower is buy-level  
double buy_level = random(par_probDistribution) * 120;  
double sell_level = random(par_probDistribution) * 120;  
if (sell_level < buy_level) {  
    double tempprice = sell_level;  
    sell_level = buy_level;  
    buy_level = tempprice;}  
prop.buyMinPrice = buy_level;  
prop.buyMaxPrice =  
    buy_level + random(par_probDistribution) * 10;  
prop.sellMinPrice = sell_level;  
prop.sellMaxPrice =  
    sell_level + random(par_probDistribution) * 10;
```

Der im folgenden gezeigte Experimentverlauf ist symptomatisch für folgende Ausgangssituation. Es wurden 20 Agenten pro Typ gestartet, d.h. 100 Agenten insgesamt. Die Laufzeit wurde auf 1800 Sekunden festgelegt. Folgende Strategieparameter wurden fest vorgegeben, alle Agenten sind in ihrer Strategie homogen und es gibt keine Verteilung der Werte: $p_{\text{acquisitiveness}} = 0.75$, $\text{del_change} = 0.25$, $\text{del_jump} = 0.15$, $p_{\text{satisfaction}} = 0.75$, $w_{\text{memory}} = 0.2$, und $p_{\text{reputation}} = 1$. Es gibt keine Lagerbestände in den Agenten, Güter kommen also nur über die Produzenten in den Markt hinein.

Das Experiment beginnt mit zufälligen Preisen, wie in Abbildung 35 gezeigt. Da mangels Güterausstattung zuerst nur die Produzenten mit den Holzfällern handeln können, überwiegen in der Grafik die Holz-Transaktionen. Deren Preise sind zu Anfang zufällig zwischen 10 und 120 Geldeinheiten verteilt, bereits hier ist aber eine Entwicklung auf einen Preiskorridor hin erkennbar.

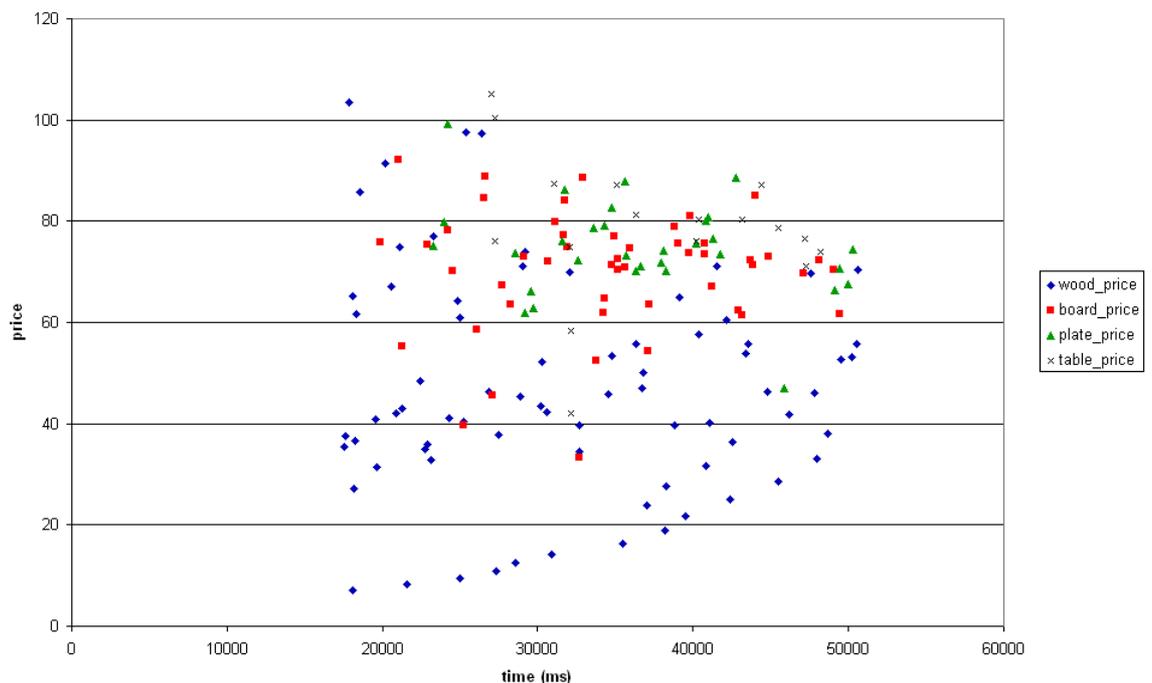


Abb. 35: Experiment mit zufälliger Preisausstattung (173 Transaktionen)

Die Entwicklung am oberen Rand lässt sich rasch durch die Wucherpreisregelung erklären. Holzfäller, die aus abgebrochenen Verhandlungen bereits Preisinformationen zwischen 10 und 40 Geldeinheiten gesammelt haben, werden hohe Preise zwischen 60 und 120 als Wucherpreise ansehen und Verhandlungen mit diesen Produzenten nicht aufnehmen. Am unteren Rand ist es den Produzenten möglich, die Preise schrittweise anzuheben, da diese im Vergleich zu den hohen Preise sehr günstig sind und es noch keinen preislichen Widerstand durch Referenzpreise auf der nächsten Wertschöpfungsstufe (Bretter) gibt. Auch die Bretter starten etwas später im Experiment mit einer zufälligen Verteilung, deren Mitte aber höher liegt als beim Holz - auch dies ist nachvollziehbar, da die Holzfäller zu sehr unterschiedlichen Preisen eingekauft haben, aber als Verkaufspreisuntergrenze für Bretter ihren Kaufpreis für Holz ansetzen. Ähnliches folgt zu einem späteren Zeitpunkt bei den ersten Platten- und Tisch-Transaktionen. Lässt man das Experiment weiterlaufen, ergibt sich die erst die in Abbildung 36 und dann die in Abbildung 37 gezeigte Entwicklung. Zuerst fällt auf, dass sich nach Gütern sortierte Preisebenen ergeben. Das zufällige Ergebnis des Anfangs weicht einer durch Selbstorganisation erreichten Aufteilung der Preise in nach Gütern unterscheidbaren Ebenen, die den Bildern aus Kapitel 5.1 ähneln.

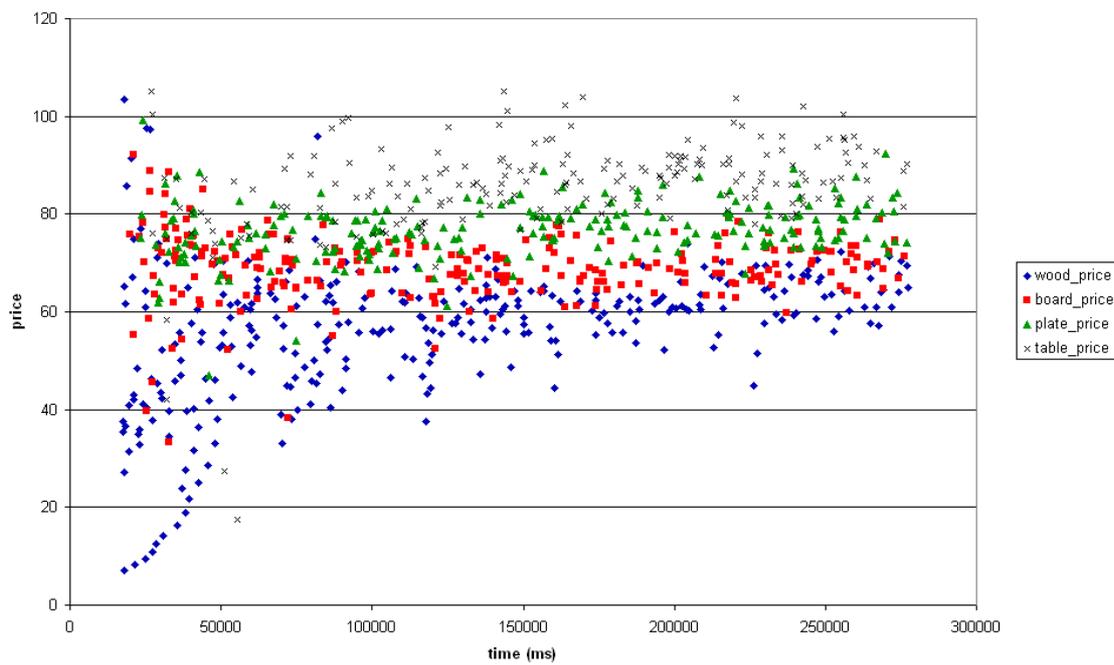


Abb. 36: Experiment mit zufälliger Preisausstattung (973 Transaktionen)

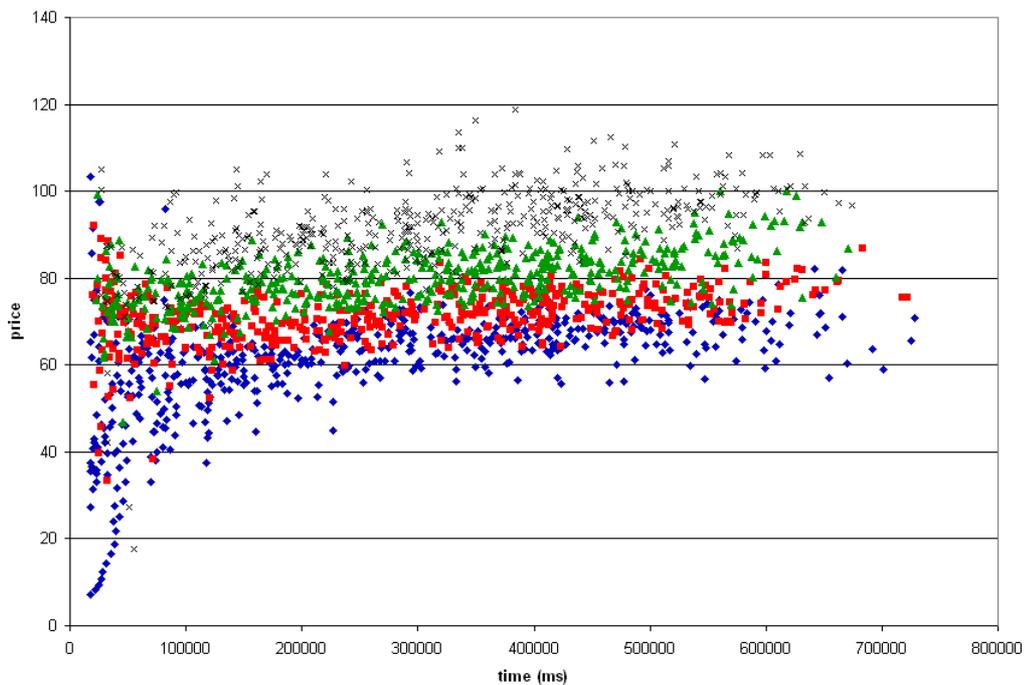


Abb. 37: Experiment mit zufälliger Preisausstattung (1973 Transaktionen)

Im Ergebnis zeigt sich eine deutliche Selbstorganisation, die auf nicht extern vorgegebene Marktpreise zuläuft, ohne dass ein Auktionator diese berechnet hat. Von besonderem Interesse ist die Entwicklung der Gewinnsituation der einzelnen Teilnehmer. Da alle Agenten homogen sind, könnte man wie in den vorherigen Beispielen annehmen, dass die Startdisposition einzelne Agenten bevorzugt und sich diese Verteilung aufgrund der fehlenden evolutorischen Weiterentwicklung nicht ändert. Tatsächlich findet aber auch hier ein Anpassungsprozeß statt, wie in Abbildung 38 gezeigt werden kann.

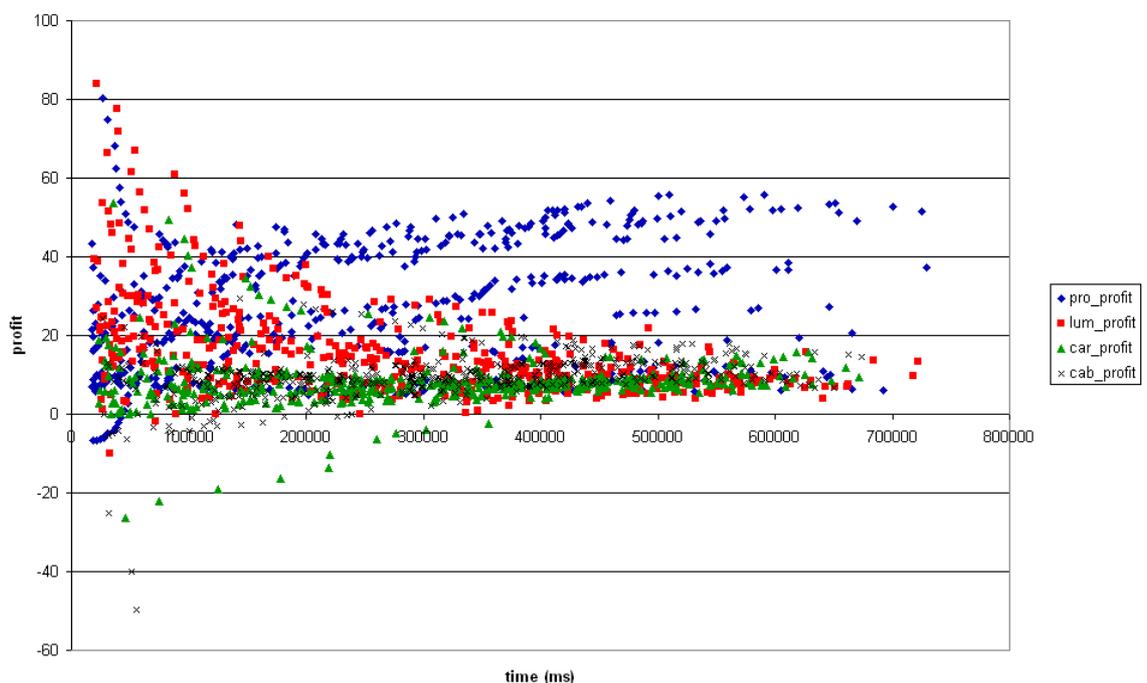


Abb. 38: Gewinnentwicklung bei zufälliger Preisausstattung

Während gerade zu Beginn aufgrund der fehlenden Referenzpreise Zimmermänner und Möbelschreiner Verluste hinnehmen müssen, können im Zeitverlauf Produzenten und Holzfäller die resultierenden Gewinne nicht halten. Die Produzenten profitieren stark von der Holzpreisentwicklung, da sie ja keine Beschaffungskosten haben und verzerren damit die Darstellung. Dagegen können die in der Wertschöpfungskette folgenden Agenten keine Vorteile gegenüber den Anderen erwirtschaften. Dies liegt zu einem großen Teil an der homogenen Strategiewerksausstattung, die eine Verbesserung der Gewinnsituation nicht ermöglicht. Im Ergebnis finden sich fast gleiche Gewinne pro Transaktion bei allen Agenten.

5.6 Veränderungen der Konzessionshöhe

Bis hierher wurde der *Acquisitiveness*-Parameter isoliert betrachtet. Welche Schlüsse lassen sich aus der Variierung der übrigen Parameter ziehen? In dem folgenden Experiment wurde der *Acquisitiveness*-Parameter bei $p_{acq} = 0.6$ belassen, jedoch die Konzessionshöhe variiert (Parameter *del_change* auf 20 Agenten gleichverteilt im Intervall $[0.05;0.65]$). Während in den vorhergehenden Experimenten die Wahrscheinlichkeit, dass eine unilaterale Konzession eingegangen wurde, variierte, geht es im Folgenden um die Höhe dieser Konzession als Prozentanteil der Spreizung zwischen dem ersten Gebot jeweils von Käufer und Verkäufer. Eine Konzessionshöhe *del_change* von 0.5 bedeutet also eine unilaterale Konzession, so dass das neue Gebot genau im arithmetischen Mittel zwischen den beiden Geboten liegt. Um ein häufigeres Entstehen von Agenten zu ermöglichen, wurde die Anzahl der betrachteten *Plumages* im STDEA-Mechanismus von 25 auf 10 verringert (vgl. Kapitel 3.5.3.5). In diesem Fall wählen die Agenten ihren *crossover*-Partner aus einer geringeren Zahl von Alternativen, führen aber die Fortplanzungsmethode häufiger aus. Dadurch wird eine im Sinne des EA schnelle Generationenfolge und auch Veränderung der Zusammensetzung der gesamten Agentenpopulation erreicht. Ein erster Experimentierlauf führt zu folgender Entwicklung des *del_change* Parameters:

:

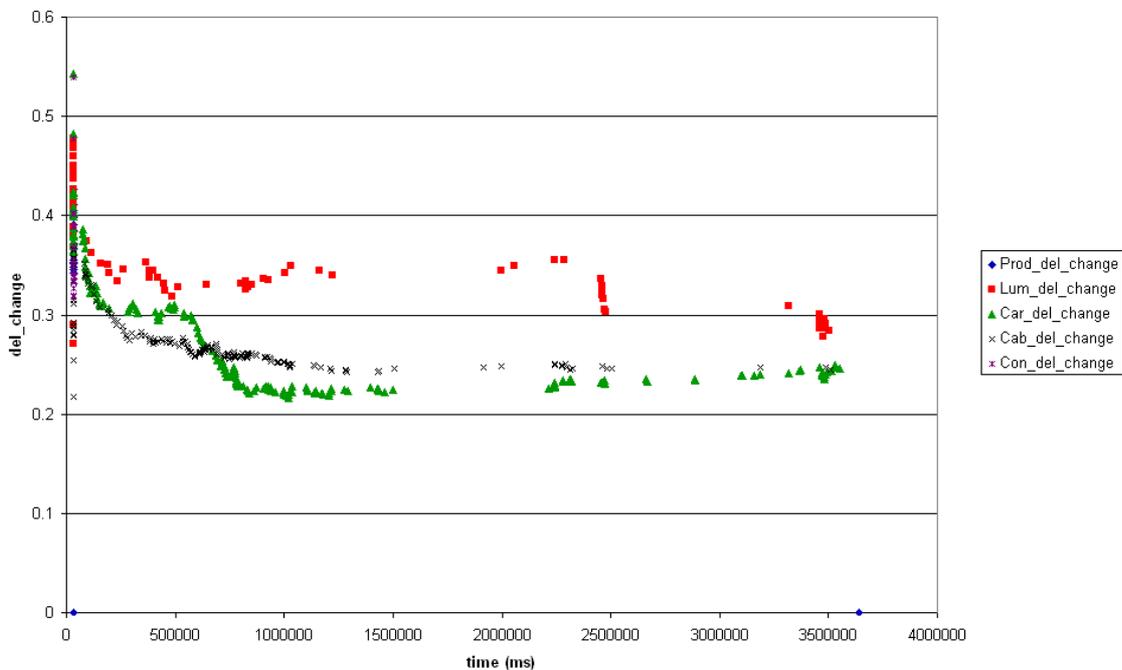


Abb. 39: Evolution der Konzessionshöhe

In Abbildung 39 ist keine besondere Entwicklung auszumachen. Alle Agententypen entwickeln sich in einer Koevolution auf einen niedrigeren `del_change` Wert, der etwa bei 0.25 liegt, also bei Konzessionen beider Partner in 2 Gebotsrunden zu einem Kompromiss führt.

Betrachtet man die Preisentwicklung (Abbildung 40), so ergibt sich in etwa ein ähnliches Bild wie in Abbildung 26. Die einzelnen Preisebene sind voneinander mit geringen Überlappungszonen getrennt. Die größere Varianz der Preise ist teilweise auf den niedrigen *Acquisitiveness*-Wert zurückzuführen. Was auffällt, ist die Ausnutzung der gesamten möglichen Preisbandbreite von 1 bis 120, die in dieser Form bei den vorhergehenden Experimenten nicht zustande kam. Dies weist auf eine größere Vielfalt erfolgreicher Agentenstrategien hin.

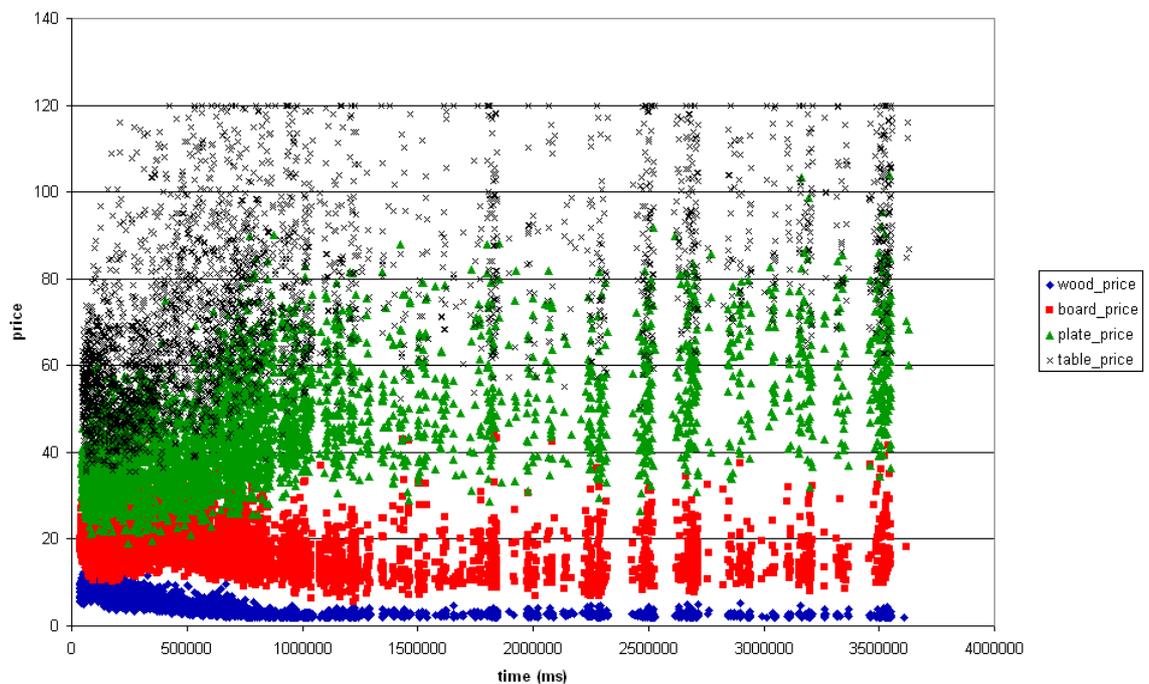


Abb. 40: Preisentwicklung bei Evolution der Konzessionshöhe

Mit den gleichen Startparametern für `del_change` kann sich auch ein vollkommen unterschiedliches Bild ergeben, wie im folgenden gezeigt wird (Abbildung 41). Die Ausgangssituation entspricht exakt derjenigen der vorhergehenden beiden Abbildungen, der Parameter `del_change` entwickelt sich aber auf einen deutlichen niedrigeren Wert hin, der den Verlauf des Experiments entscheidend in eine andere Richtung bewegt.

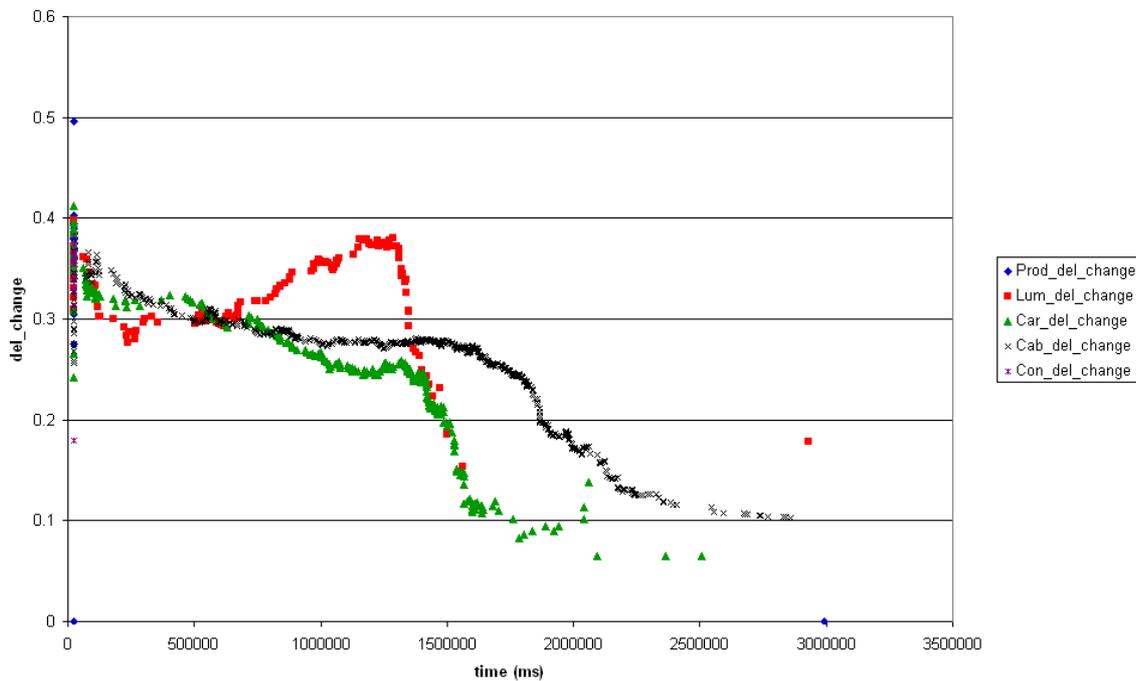


Abb. 41: Evolution der Konzessionshöhe (2. Versuch)

Was an dieser Abbildung 41 auffällt, ist erneut die Koevolution des Parameters, d.h. die gegenseitige Verfolgung der Kurven. Trotz des Verfalls des Parameters, der schließlich zum Abbruch des Experiments durch „Bankrott“ aller Agenten führt, findet erneut eine Koevolution statt. Diese Entwicklung läßt darauf schließen, dass es sich kein Agent leisten kann, höhere Konzessionen als seine Gegenspieler einzugehen, auch wenn analytisch in keinem dieser Experimente deutlich wird, wie genau dieser Parameter zur Nutzenmaximierung der Agenten beiträgt. Im Ergebnis können die Zimmermänner aufgrund ihrer geringeren Konzessionshöhen relativ hohe Gewinne pro Transaktion einfahren, bevor das Experiment abgebrochen wird.

Das nächste Experiment (Abbildung 42) befasst sich mit der Fragestellung, welche Entwicklung sich aus der Variation sowohl der Konzessionsneigung p_{acq} als auch der Konzessionshöhe del_change ergibt. Es wurde eine Gleichverteilung von p_{acq} zwischen 0.3 und 0.9 (0.6 ± 0.3) und von del_change wieder von 0.05 bis 0.65 (0.35 ± 0.3) vorgelegt. Die sich daraus ergebende Preiskurve in Abbildung 42 zeigt ein schnelles Auffüllen des gesamten zur Verfügung stehenden Preisraumes zwischen 0 und 120 Geldeinheiten, genau wie bei Abbildung 40.

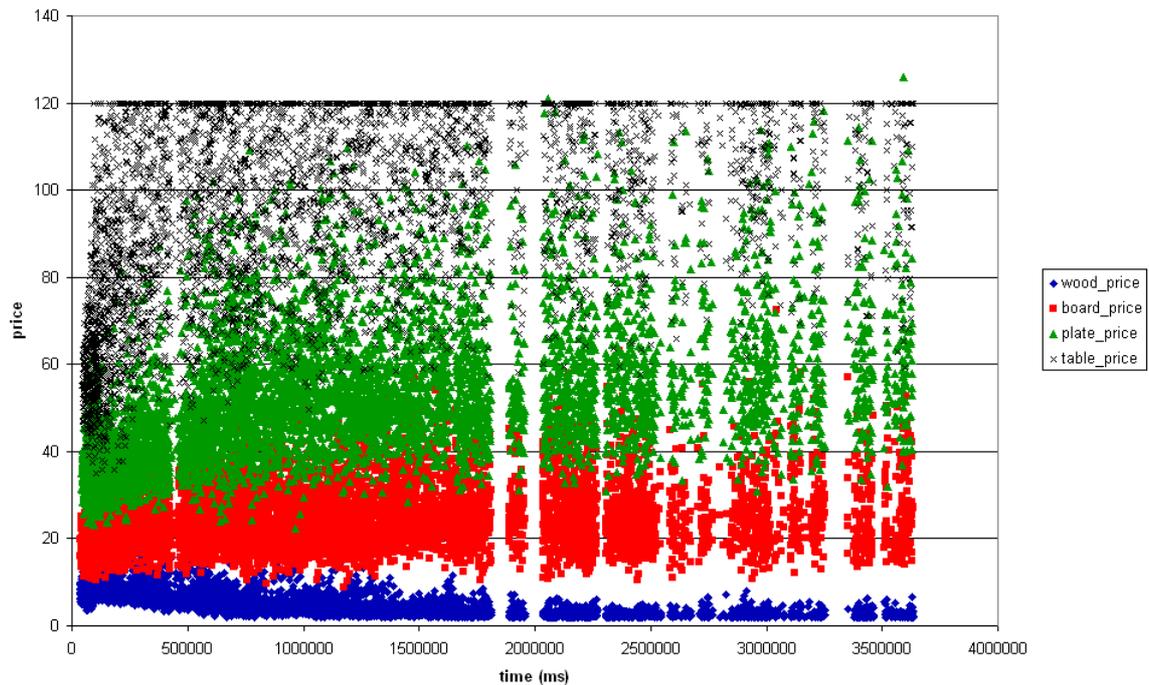


Abb. 42: Preisentwicklung bei simultaner Evolution von p_{acq} und del_change

In Bezug auf die Begriffswelt evolutionärer Algorithmen könnte man diese Entwicklung als Besetzen aller möglichen Nischen interpretieren. In diesem Experiment existieren wesentlich unterschiedlichere Variationen der Handelsstrategie als in den ersten Experimenten dieses Kapitels, bei denen sich feste Preisbänder ergaben. Hier scheinen sich verschiedene Alternativen der Realisierung der Wertschöpfungskette nebeneinander zu zeigen. Welche Entwicklung zeigen jetzt die beiden Parameter p_{acq} und del_change über die Laufzeit des Experiments? Wenn das Ergebnis der o.a. Preisentwicklung "anything goes" bedeutet, d.h. eine Interpretation zulässt, die jedem Agenten unabhängig von seiner Parameterausstattung ausreichenden Gewinn zumisst, um das Experiment zu überleben, würden sich dann nicht diese Parameter vollkommen chaotisch entwickeln? Ein Blick auf die Abbildung 43 zeigt ein in diesem Sinn überraschendes Ergebnis.

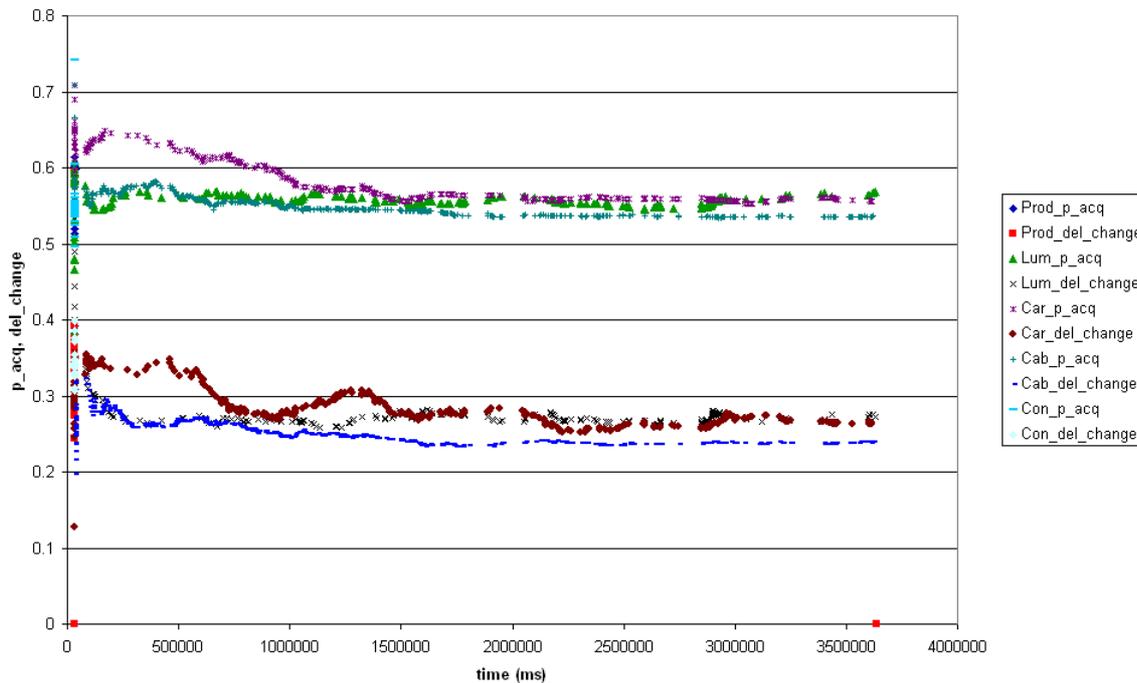


Abb. 43: Koevolution der Parameter p_{acq} und del_change

Offensichtlich gibt es eine stabile Parameterkonstellation bei $p_{acq} = 0.55$ und $del_change = 0.25$, die alle Agenten gleichermaßen magisch anzieht. Diese Kombination läßt sich mit den zur Verfügung stehenden Mitteln nicht analytisch begründen. Es kann aber festgehalten werden, dass die Entwicklung des Marktes nicht auf chaotischem oder nichtdeterministischem Verhalten beruht, sondern deutliche selbstorganisatorische Züge zeigt.

Erweitert man dieses Experiment, indem man mit vollkommen zufälligen Indifferenzpreisen in den Agenten beginnt, so ergibt sich in der Preisentwicklung der Abbildung 44 eine ähnliche Darstellung wie in den Abbildungen 37 und 42. Die einzelnen Preisebenen verteilen sich erneut symmetrisch entlang der Wertschöpfungskette. Anders als in der Abbildung 37 zeigen die einzelnen Preisebenen eine wesentlich höhere Varianz und auch hier wird wieder fast der gesamte Preisbereich von 0 bis 120 Geldeinheiten "besiedelt."

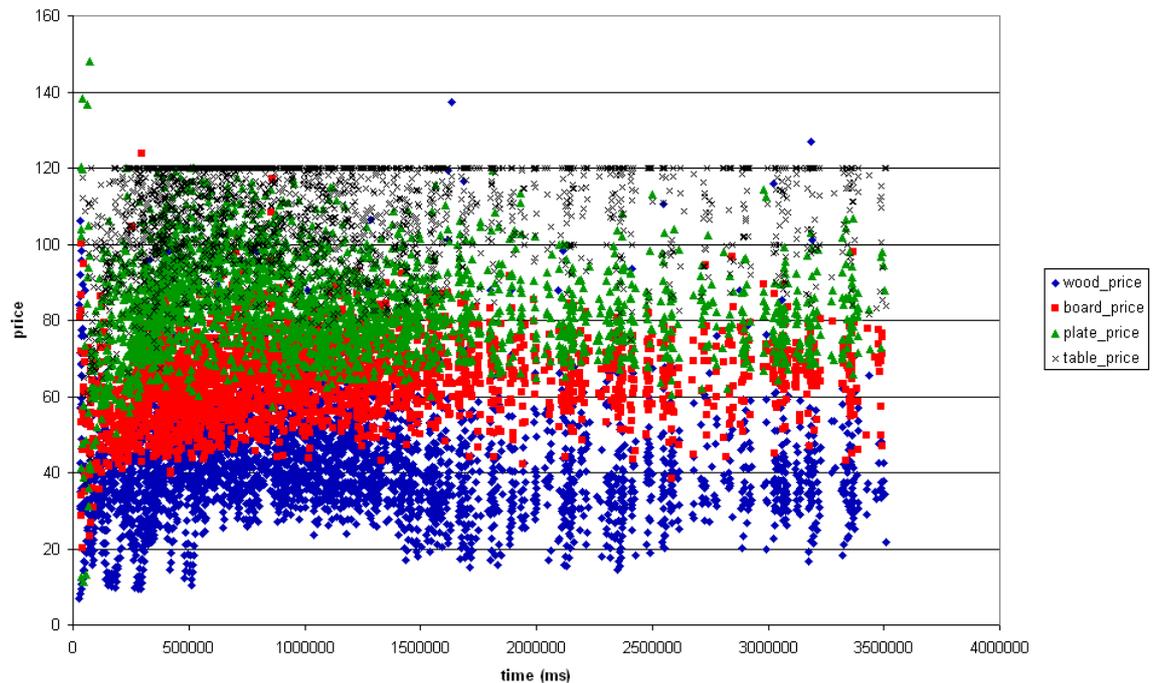


Abb. 44: Preisentwicklung bei zufälligen Startpreisen und Variation von p_acq und del_change

In diesem Experiment wurde neben der zufälligen Preisausstattung auch noch die Zufriedenheit (*satisfaction*) variiert, mit gleichen Mittelwerten und Verteilung wie die *Acquisitiveness*. Die Zufriedenheit eines Agenten gibt die Wahrscheinlichkeit an, mit der eine Verhandlung nach Erhalt eines Gebotes fortgesetzt wird. Je höher dieser Wert ist, desto länger können Verhandlungen im Mittel dauern, bis eine Einigung erreicht wird. In Abbildung 45 wird gezeigt, wie sich alle drei Parameter Konzessionswahrscheinlichkeit (*Acquisitiveness*, p_acq), Konzessionshöhe (*DeltaChange*, del_change) und Zufriedenheit (*Satisfaction*, p_sat) für die drei Agententypen Holzfäller, Zimmermann und Möbelschreiner im Zeitablauf des Experiments verändern.

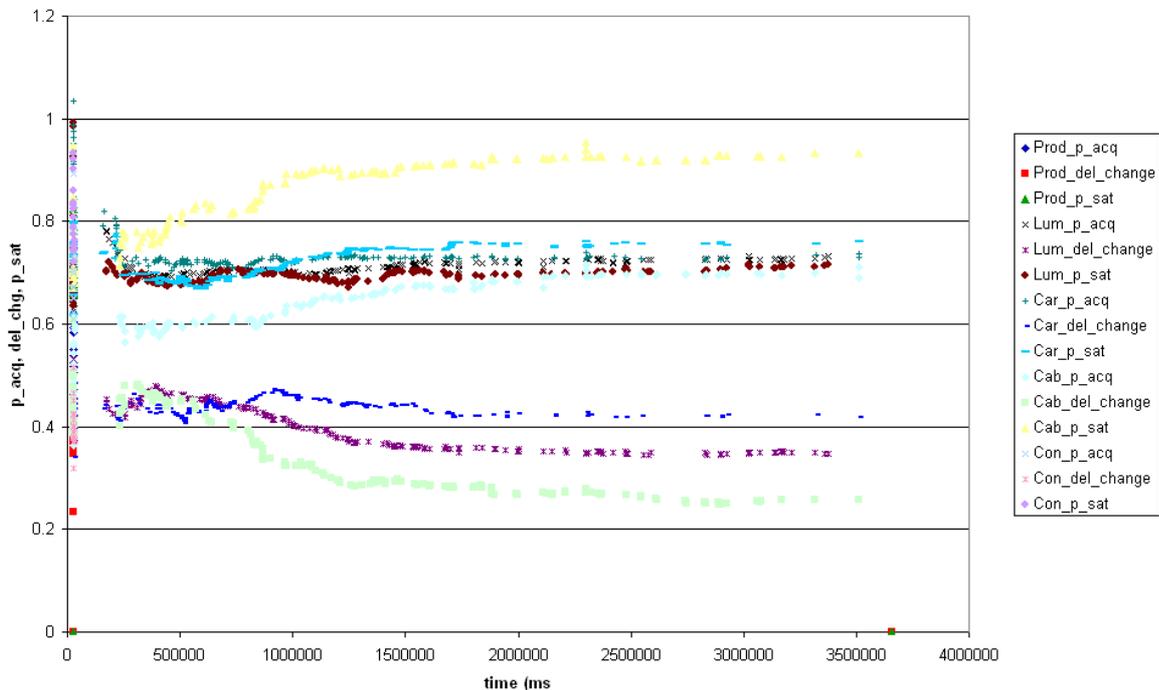


Abb. 45: Koevolution von p_{acq} , del_change und p_{sat}

Die Konzessionswahrscheinlichkeit und die Zufriedenheit entwickeln sich sichtlich parallel zueinander, was durchaus ein Nebeneffekt der Strategieformalisierung sein kann - je schneller Konzessionen aufeinanderfolgen, desto schneller ist eine Verhandlung abgeschlossen, wodurch der Einfluß der Zufriedenheit auf die Verhandlungsführung geringer ist. Die Konzessionshöhe entwickelt sich separat bei allen drei Agententypen von einem höheren Startlevel aus auf einen Bereich zwischen 0.25 und 0.4.

In der Kombination zufälliger Preise und der Variation dreier Parameter zeigt dieses Experiment eine Art der Selbstorganisation der Agenten im Markt. Durch die hohe Variablenzahl im Verhältnis zu den ersten Experimenten, bei denen jeweils nur ein Parameter variiert wurde, ergibt sich auch ein Problem mit der Interpretation der komplexen Ergebnisse.

Die emergente Entwicklung von Preisen und Gewinnen sowie die Veränderung hin zu nutzenmaximierenden Strategien in den ansonsten sehr einfach gehaltenen Agenten, die weder ein symbolisches internes Modell noch komplizierte Lernkonzepte implementieren, zeigt letztendlich die Leistungsfähigkeit des marktlichen Preismechanismus. Die Aussagefähigkeit, inwiefern eine bestimmte Entwicklung auf eine Veränderung eines isolierten Parameters zurückzuführen ist, nimmt bei zunehmender Variablenzahl schnell ab.

Nach Ansicht des Verfassers zeigt AVALANCHE stattdessen, wie aus der Interaktion einfacher Agenten äußerst komplexe, ökonomisch interpretierbare Muster entstehen, deren Entwicklung weder einprogrammiert sind, noch durch die Analyse des Verhaltens eines einzelnen Agenten ex ante prognostiziert werden können. Es soll an dieser Stelle noch einmal darauf hingewiesen werden, dass für jede in diesem Kapitel gezeigte Darstellung der Ergebnisse mehrere Experimentierläufe notwendig waren, von denen viele auch ohne Aussage endeten. Hier zeigt sich nicht nur der nicht-deterministische Aspekt der Agentenstrategie, sondern auch die Unsicherheit in Bezug auf Angebot und Nachfrage, d.h. letztendlich die "konstitutionelle Unsicherheit" im Sinne von Hayek. Zwischen den hier vorgestellten Forschungsergebnissen und einer stabilen Koordination, die in Produktionsumgebungen einsetzbar ist, liegen noch einige technische (z.B. Stabilität der JAVA-Prozesse) und ökonomische Schritte (z.B. Institutionen zur ökonomischen Stabilisierung).

6 Ergebnisse und Ausblick

In den vorhergehenden Kapiteln wurde gezeigt, in welcher Weise ökonomische Konzepte eines dezentralen, selbstorganisierenden Marktes für die Koordination von Multi-Agenten-Systemen genutzt und wie umgekehrt Elektronische Märkte als Multi-Agenten-Systeme implementiert werden können.

Das Multi-Agenten-System AVALANCHE ist eine prototypische Implementation eines dezentralen marktlichen Koordinationsmechanismus und zeigt dessen Leistungsfähigkeit anhand der Koordination einer Wertschöpfungskette. Die Koordination auf den Marktplätzen und über die Wertschöpfungskette insgesamt ergibt eine "spontane Ordnung" von Angebot und Nachfrage, obwohl auf der Systemebene keine steuernden Regeln in dieser Hinsicht definiert wurden. In diesem Sinne stellt AVALANCHE eine Realisierung von Hayeks Katallaxiebegriff dar, die eine marktliche Ordnung als emergente Eigenschaft der individuellen Nutzenmaximierung begreift. Die Abbildung des Vielkomponentensystems "Markt" auf ein Multi-Agenten-System, bestehend aus interagierenden, autonom handelnden Softwareobjekten, kann auf mehreren Wegen geschehen. Die Aufgabe besteht dabei in der Verbindung der technischen Realisierung eines Multi-Agenten-Systems (eine Domäne des Forschungsgebietes der Verteilten Künstlichen Intelligenz) und dem Verständnis des ökonomischen Phänomens des Marktes (eine Domäne der Ökonomik bzw. Volkswirtschaftslehre). Zwei unterschiedliche Wege sind in Abbildung 46 einander gegenübergestellt.

Es kann unterschieden werden zwischen der Ebene der ökonomischen Analyse und derjenigen der technischen Implementierung. In der ökonomischen Analyse wird das Phänomen des Marktes in seiner Wirkung als dynamischer Koordinationsmechanismus betrachtet. Zwei Beschreibungsmöglichkeiten, die den Koordinationsmechanismus formalisieren, sind zum einen der Walras'sche Auktionator, zum anderen die Katallaxie von Hayek. Während der Auktionator formal und mathematisch definiert ist, gibt es über die Funktionsweise der Katallaxie nur grundsätzliche, aber nicht detailliert ausgearbeitete Ansichten.

Auf der Ebene der technischen Implementierung sind Multi-Agenten-Systeme bereits in der Forschung und ersten Anwendungen zu beobachten. Koordination in Multi-Agenten-Systemen kann prinzipiell auf eine ökonomische Weise interpretiert und implementiert werden. Wellman (1995) hat diese Verbindung über seine Implementierung des Walras'schen Auktionators geschaffen und sich einerseits der formalen Beschreibung von Walras (1847) bzw. Samuelson (1974), andererseits einer existierenden MAS-Technologie bedient. AVALANCHE basiert auf dem eher vage beschriebenen dezentralen Preismechanismus, den Hayek als Katal-

laxie definiert und bringt davon ausgehend eine selbstorganisatorische Koordination zustande. Im Ergebnis wird das Problem der Koordination von beiden Ansätzen in ähnlicher Weise gelöst. Die Frage, welchen Weg man gehen will, entzündet sich im wesentlichen an der ökonomischen Interpretation der Funktionsweise des Marktes.

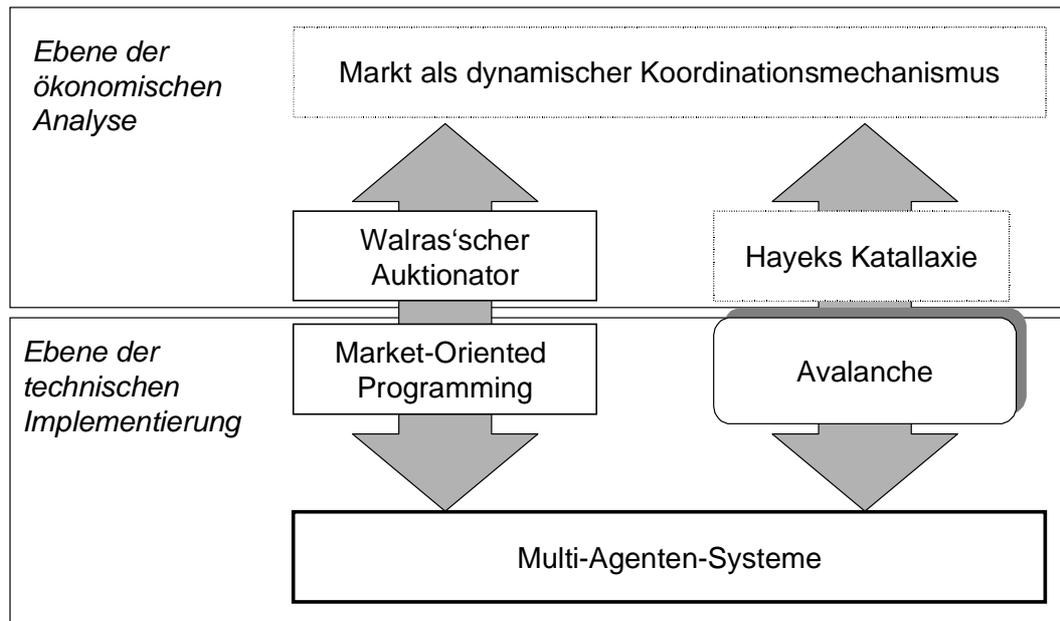


Abb. 46: Vom MAS zum Markt: Konzepte und Implementierungen

Der linke Pfeil der Abb. 46 stellt das Verständnis des Phänomens "Markt" der neoklassischen Ökonomik bzw. der allgemeinen Gleichgewichtstheorie dar. Hier steht mit dem Walras'schen Auktionator eine detaillierte mathematische Beschreibung eines Koordinationsmechanismus zur Verfügung. Durch die Umsetzung dieser Beschreibung in eine Software-Implementierung lassen sich Multi-Agenten-Systeme koordinieren, wie es Wellman mit seinem Konzept des Market-Oriented Programming zeigt. Dieser Weg birgt keine Überraschungen, da die Konzepte hinreichend bekannt und analysiert sind. Es sei darauf hingewiesen, dass der Walras'sche Auktionator (sowie seine Implementierung in MAS) zwar den Effekt eines Marktes simuliert, jedoch nicht notwendigerweise mit dem in der Realität vorzufindenden Marktmechanismus übereinstimmt (Kearney/Merlat 1999, S. 149). Für die technische Gestaltung bedeutet dies aber auch, dass die Annahmen, unter denen der Walras'sche Auktionator seine Ergebnisse hervorbringt (vgl. Kapitel 2.4.1), in der Implementierung eingehalten werden müssen, notfalls durch Ausschluss bestimmter Charakteristika des Koordinationsproblems. Damit muß sich Market-Oriented Programming dennoch dem Vorwurf einer Vorgehensweise aussetzen, die Wooldridge/Jennings (1999, S. 21) als "shoehorning" bezeichnen und die in der Anpassung

des Problems mündet, um unbedingt einen agentenbasierten Lösungsansatz darauf anwenden zu können.

Alternativ zeigt der Pfeil auf der rechten Seite von Abb. 46 eine andere Verbindung zwischen Implementierung und ökonomischem Verständnis des "Marktes" über das Katallaxiekonzept von Hayek. Das in dieser Arbeit vorgestellte MAS AVALANCHE stellt eine prototypische Implementierung in diesem Sinne dar. Da die Ökonomie in ihrer zweihundertjährigen Forschungsgeschichte keine alternative Formalisierung zustande gebracht hat, die den gleichen Detaillierungsgrad aufweist wie der Walras'sche Auktionator, stellt AVALANCHE für die beiden Disziplinen Ökonomie und (Wirtschafts-)Informatik einen Erkenntnisgegenstand dar.

6.1 Thematische Einordnung der Ergebnisse

Die Ergebnisse der vorliegenden Arbeit lassen sich unter zwei Gesichtspunkten diskutieren. Ziel der Arbeit war die Entwicklung eines Koordinationsmechanismus, welcher in elektronischen Märkten, bestehend aus von vielen menschlichen Anwendern erstellten und wechselnden Software-Agenten, die individualistisches und informationsstrategisches Verhalten zeigen, es den einzelnen Agenten ermöglicht, planmäßig eine nutzenverbessernde Ressourcenallokation anzustreben.

Die für AVALANCHE gewählte Realisierung des Koordinationsmechanismus kann zum einen im Sinne der **Wirtschaftsinformatik** als Gestaltungsoption für die Koordination von Systemen eigennütziger Agenten verstanden werden. Die hier vorgestellte Implementation eines elektronischen Marktplatzes, auf dem Software-Agenten miteinander handeln können, stellt einen experimentellen Prototyp dar. Dieser zeigt, wie die technische Realisierung derartiger Marktplätze aussehen kann und bietet innovative Lösungen für theoretische sowie technische Probleme, die deren Automatisierung und Implementation bisher behindert haben. Damit reiht sich AVALANCHE in andere Forschungsarbeiten zu agentenbasierten elektronischen Märkten ein, wie FISHMARKET (Gimenez et al. 1998), die auf Market-Oriented Programming basierenden Projekte AUCTIONBOT (Wurman et al. 1998) und HOMEBOT (Ygge 1998), oder der agentenbasierte Finanzmarkt in AMTRAS (Gomber 1999). Diese Projekte realisieren ohne Ausnahme marktliche Koordination mit Hilfe eines Auktionators. AVALANCHE stellt für diesen Anwendungszweck eine grundlegend verschiedene Gestaltungsoption dar.

Andererseits ist AVALANCHE nicht nur ein weiterer elektronischer Marktplatz, sondern durch die Abstraktion der Koordinationschicht vom Anwendungskontext auch zur experimentellen Untersuchung und Simulation unterschiedlicher Koordinationsprobleme in den **Wirtschaftswissenschaften** einsetzbar. Die Kombination ökonomischer Modelle autonomen Verhaltens

mit einer Computertechnologie, die autonomes Verhalten implementiert, stellt ein fruchtbares Forschungsfeld dar (vgl. auch Wellman 1995, Vulkan/Jennings 1998, Kearney/Merlat 1999), welches als *Agent-based computational economics* (ACE) bezeichnet wird (Tesfatsion 1997). Dabei untersuchen Ökonomen selbst-organisatorische Fähigkeiten dezentraler Marktökonomien, um Erkenntnisse für das Verständnis ökonomischer Mechanismen und Effekte zu gewinnen. ACE-Modelle sind im Computer als virtuelle ökonomische Welten implementiert. Sie werden, vergleichbar mit der Entwicklung von Zellkulturen in einem Labor, über den Zeitablauf in ihrem Wachstum untersucht. Im Prinzip sind, sofern die Startbedingungen einmal gesetzt sind, alle folgenden Ereignissen in diesem virtuellen ökonomischen Welten (Lane 1993) initiiert und getrieben von Agent-Agent- sowie Agent-Umgebungs-Interaktionen. Eingriffe von außen durch den Modellierer (z.B. Berechnung und Verfügbarmachen eines Gleichgewichtspreises) sind nicht erlaubt (Tesfatsion 1997). Der Modellierer gibt stattdessen Interaktionswege und Normen vor, die z.B. aus dem Forschungsgebiet des „Social Modelling“ herkommen, welches die emergente Entwicklung von Institutionen (z.B. soziale Normen) untersucht (vgl. Conte/Castelfranchi 1995).

Für Konzepte agentenbasierter Marktplätze stellt AVALANCHE eine experimentelle ACE-Testumgebung dar, in der verschiedene Implementationen der Umgebung bzw. der Agenten einander gegenübergestellt werden können (vgl. Eymann/Padovan 1999). Die technische Implementation des Marktplatzes, des Verzeichnisdienstes sowie die meisten Elemente der Softwareagenten werden nicht verändert. Die Veränderungen beziehen sich auf die organisatorische Anordnung von Marktplätzen und deren Vernetzung bzw. auf unterschiedliche Verhandlungsstrategien und das ökonomische Rational in den Agenten. In diesem Sinne lassen sich mit AVALANCHE Zustände simulieren, die in der Zukunft technisch realisierbar sind, jedoch nicht zwangsläufig eintreten müssen. Diese Zustände können experimentell verglichen werden, indem *ceteris paribus* nur wenige Parameter verändert werden. Die sechs Parameter des Genotyps lassen sich z.B. als Simulation sehr komplexer, deterministischer Verhandlungsstrategien auffassen⁴⁰. Die unterschiedliche Ausgestaltung einzelner Parameter führt zu unterscheidbaren Strategien, mit deren Hilfe „Intelligente“ Agenten simuliert werden können, die autonome Entscheidungen durch sehr komplexe Mechanismen treffen.

Dadurch wird es möglich, generelle Fragestellungen, die mit der Koordination von Vielkomponentensystemen zusammenhängen, in experimenteller Weise zu betrachten. Der Vorteil

40 Der nicht-deterministische Automat, der durch den Genotyp bestimmt wird, stellt die Verhandlungsstrategie eines Agenten dar. Er kann durch bekannte Verfahren (Hopcroft/Ullman 1994, S. 19ff.) in einen wesentlich komplexeren deterministischen Automaten überführt werden, so dass durchaus eine Vergleichbarkeit zu einer komplexen Strategievorgabe durch einen menschlichen Prinzipal besteht.

einer Simulation auf Basis eines Multi-Agenten-Systems ist zweifellos die Verbindung zwischen Mikroverhalten der Agenten und Makroverhalten des Systems:

“Thus mathematical models are used at the macro level whereas multi-agent simulation models are used to cross the micro-macro bridge by letting global configuration emerge from the local interactions.” (Ferber 1994, S. 16)

Als Werkzeug sind die MAS vergleichbar mit Ansätzen der Synergetik (Haken 1990) oder der System Dynamics (Karnopp et al. 1990). In den Augen anderer Wissenschaftler gehen sie über deren Möglichkeiten hinaus, wie Kauffman (1995, S. 264) mit dem synonymen Begriff der "patches" beschreibt:

"In fact, I suspect that analogues of patches, systems having various kinds of local autonomy, may be a fundamental mechanism underlying adaptive evolution in ecosystems, economic systems, and cultural systems. If so, the logic of patches may suggest new tools in design problems. Moreover, it may suggest new tools in the management of complex organisations and in the evolution of complex institutions worldwide."

Diese grundlegende Klasse an Systemen, die als "Complex Adaptive Systems" bezeichnet werden, erzeugt globale Koordinationsmechanismen durch die Modellierung lokaler Interaktionsprotokolle und baut auf dem Wettbewerb zwischen Individuen auf. Ein Beispiel ist die Koordination von Leistungserstellungsprozessen durch den Einsatz autonomer Agenten, die mittels ökonomischer Protokolle kooperieren und kommunizieren.

“By executing these models on a computer we gain a double advantage: (i) An experimental format allowing free exploration of system dynamics, with complete control of all conditions; and (ii) an opportunity to check the various unfolding behaviors for plausibility, a kind of 'reality check'. Whether or not agents in such worlds behave in an optimal manner, the very act of contemplating such systems will lead to important questions and answers” (Holland/Miller 1991, S. 370).

In diesem Sinne besteht zwischen dem Entwurf von markt-basierten Multi-Agenten-Systemen und ihrer ökonomischen Analyse eine enge Verbindung (vgl. Eymann/Kearney 2000). Die Gestaltung von Interaktionen zwischen den einzelnen Agenten, die Güter oder Dienstleistungen wechselseitig ein- oder verkaufen, ist ein Schwerpunkt der ökonomischen Forschung. Aus Forschungen des "Economic Mechanism Design" (Varian 1995) ist beispielsweise die Vickrey-Auktion entstanden, deren hervorragendstes Merkmal die Förderung ehrlichen Verhaltens (Äusserung wahrer Präferenzen als dominierende Strategie) darstellt. In ähnlicher Weise haben

Spieltheorie und Soziologie wertvolle Beiträge zur Analyse und Gestaltung von Interaktionen zwischen menschlichen Teilnehmern geleistet. In der weiteren Entwicklung des Agent-mediated Electronic Commerce (AmEC) sind viele Projekte entstanden, die versuchen, Konzepte aus der Interaktion zwischen menschlichen Agenten auf Software-Agenten zu übertragen. Von einer praktischen Sicht, welche über akademische und kleinzahlige Projekte hinausgeht, lässt sich sagen dass für die meisten dieser Projekte die folgende Aussage gilt:

“Most theoretical models from economics are not valid for multiple encounter types of interaction, which is a more than probable scenario for agent-mediated electronic commerce” (Sierra 2000, S.9).

6.2 AVALANCHE - ein Prototyp für ein Katallaktisches Informationssystem

Vom ökonomischen Standpunkt aus ist AVALANCHE eine der ersten Koordinationsimplementationen eines Marktes, die explizit nicht auf den Walras'schen Auktionator rekurriert. Es stellt einen Gestaltungsansatz für ein nicht-neoklassisches Marktdesign dar und bietet einen Analyseansatz für das Funktionieren ökonomischer Koordination. An der Schnittstelle zwischen Ökonomie und Informatik findet sich dadurch ein neues, abgrenzbares Feld zukünftiger Forschung. Zur Bezeichnung derjenigen Klasse der Informationssysteme, die sich durch einen dezentralen Koordinationsmechanismus in der in dieser Arbeit vorgestellten Weise auszeichnen, wird hier der Begriff der *Katallaktischen Informationssysteme* vorgeschlagen. *Katallaktische Informationssysteme* lassen sich durch die beiden Eigenschaften der ökonomischen Dezentralisation und der zunehmenden Komplexität charakterisieren, die sie von existierenden Informationssystemen und Auktionatorsystemen unterscheidet.

1. **Ökonomische Dezentralisation:** Koordination und Entscheidung sind vollständig dezentralisiert. Es gibt keine zentrale Entscheidungsinstanz, die den Teilnehmern hierarchisch übergeordnet ist. Die Kohärenz der Einzelaktionen wird durch ökonomische Verfahren sichergestellt. Die autonomen Softwareobjekte (Agenten) verändern ihr internes Modell mittels lokaler, adaptiver Entscheidungsregeln. Ihre Sensorik orientiert sich dabei an der Perzeption von Preisverhältnissen; ihre Aktionen bestehen in der Signalisierung von Preisänderungen (als Zeichen der Wertschätzung einer Ressource) an ihre Umwelt. Trotz dieser nur eingeschränkten Handlungsmöglichkeiten zeigt das Gesamtsystem ein emergentes Verhalten, welches komplexe Aufgaben lösen kann.
2. **Zunehmende Komplexität:** Die Komplexität des Systems wird zur Laufzeit erhöht und nicht bereits durch das Design vorgegeben. Die Zahl der Agenten, wie auch die Komplexi-

tät ihres internen Modells, kann zu Beginn der Existenz eines Multi-Agenten-Systems gering sein. Über die Zeit wird sich die Zahl der Agenten aufgrund von Größeneffekten, Verbund- und Netzeffekten verändern. Die Komplexität des internen Modells der Agenten wird durch maschinelles Lernen zunehmen. Ein agenten-basierter Lernansatz lässt die autonomen Objekte des Informationssystems zwischen alternativen Handlungsmöglichkeiten und -strategien auswählen, diese verbessern oder neue entwickeln. Ein ökonomischer Erfolg in der Koordination eines Agenten führt zur Anpassung der Strategien anderer Agenten durch Veränderung ihrer ökonomischen Entscheidungsvariablen und Verfahren über dezentrale, modifizierte evolutionäre Algorithmen (Smith/Taylor 1998, Stender 1994, Holland/Miller 1991). Diese Vorgehensweise eines Markoff-Modells ermöglicht es, mit einer nichtoptimalen Population anzufangen, in der jeder Agent eine mehr oder weniger zufällige Verteilung von Parameterwerten repräsentiert und in der das System erst im Zeitablauf ökonomische Stärke und Lösungsfähigkeit entwickelt. Aufgrund der dynamischen und nicht-linearen Progression des evolutionären Algorithmus können sogar solche Agenten entstehen und sich als besser in Bezug auf die Anpassung ihrer Parameterkombination an die Umgebung erweisen, die in der Ausgangspopulation nicht vertreten waren und diese spezielle evolutionäre Nische unbesetzt ließen (Kauffman 1995).

Nach Ansicht des Verfassers stellt AVALANCHE ein prototypisches und noch rudimentär ausgeprägtes *Katallaktisches Informationssystem* dar. Die gerade angeführten Designcharakteristika sind in AVALANCHE wiederzufinden: autonome, ökonomisch entscheidende Agenten sind der zentrale technische Baustein des Systems und die Komplexität des Systems als Ganzes wird durch das inhärente Lernen der Agenten bewältigt und verändert, wie durch die Experimente gezeigt werden konnte.

6.2.1 Herausforderungen an den praktischen Einsatz von MAS

Autonome Agenten und Multi-Agenten-Systeme stellen ein lebhaftes und rapide expandierendes Feld informatischer Forschung und Entwicklung dar. Die einbezogenen Konzepte stammen aus unterschiedlichen Bereichen wie Distributed Computing, Objektorientierte Systeme, Software-Entwicklung, Künstliche Intelligenz, Ökonomie, Soziologie und Organisationswissenschaften. Trotz des offensichtlichen Potentials bleiben eine Reihe von Forschungs- und Entwicklungsfragen offen (Jennings et al. 1998). Der Einsatz von Multi-Agenten-Systemen in praktischen Anwendungen steht in vielerlei Hinsicht am Anfang. In diesem Abschnitt sollen schlaglichtartig die wesentlichen technologiespezifischen Probleme beleuchtet werden, die einem praktischen Einsatz von solchen MAS wie AVALANCHE derzeit noch entgegenstehen.

- **Software Engineering:** Das konzeptuelle Design von Multi-Agenten-Systemen für praktische Anwendungen hat bisher kein anerkanntes Vorgehensmodell hervorgebracht. Jennings et al. (1998, S.299) sprechen in Bezug auf das heutige Vorgehen bei dem Software Engineering eines MAS von *"ad hoc manner — either by borrowing a methodology (typically an object-oriented one) and trying to shoe-horn it to the multi-agent context or by working without a methodology and designing the system based on intuition and past experience"*. Wooldridge/Jennings (1999) haben eine Reihe von Fallen (Pitfalls) identifiziert, in die ein unerfahrener Entwickler eines Multi-Agenten-Systems hereintappen könnte. In einer praktischen Anwendung wird z.B. die Notwendigkeit des Einbindens existierender IT-Systeme wesentlich höher gewertet als bei universitären, kleinzahligen Forschungsprojekten. Daher werden "Agent Wrapper" notwendig, die existierende Interfaces und Methoden kapseln, um eine Weiterentwicklung der Technologie zu ermöglichen und gleichzeitig einen Investitionsschutz für existierende Systeme darzustellen.
- **Evaluation der Technologie:** Auch die Evaluation von MAS steht in vielen Fällen noch aus. Ob MAS tatsächlich einen Mehrwert zu konventionellen IT-Systemen, Expertensystemen oder DPS-Ansätzen darstellt, ist für die wenigsten Anwendungen untersucht. Das objektivste Kriterium dafür ist sicherlich der Markt für IT-Systeme, d.h. die Frage, ob Menschen für den Einsatz von MAS in der Industrie bezahlen werden (Nwana/Ndumu 1999). Diese Frage kann zum gegenwärtigen Stand der Technologie noch nicht beantwortet werden. Als Anekdote sei das Beispiel der deutschen Softwarefirma "living systems"⁴¹ angeführt, die erfolgreich Auktions-MAS entwickelt und verkauft (z.B. eBay Deutschland, AMTRAS, oder die Energiebörse EEX), obwohl (oder gerade weil) die Agentenbasierung ihrer Systeme nicht als Verkaufsargument herausgestellt wird.
- **Sicherheit:** Weitergehende Probleme werden im Bereich des Vertrauens von Anwendern in die Sicherheit des E-Commerce gesehen. Insbesondere Vertraulichkeit, Integrität, Authentifizierung und Unabstreitbarkeit, sichere Zahlung und Lieferung werden als Herausforderungen genannt. Die Betrachtung von technischen Risiken in MAS konzentriert sich meist auf die Frage, inwiefern Agenten und/oder Agentenumgebungen (Server) vor technischen Angriffen wie Datenflussanalysen, Abhören und Speicherung von Daten, Maskeraden oder dem Einsatz subversiver Agenten geschützt werden können (Foner 1997, Chess 1998). Zur Abwehr derartiger Angriffe existieren technische Konzepte wie die Verschlüsselung von Kommunikationswegen oder von Agenten-Code (bei mobilen Agenten) (FIPA 1998).

41 <http://www.livingsystems.de>

-
- **Reputation:** Auch die a priori unbekannte Reputation anderer Agenten ist ein wichtiges Thema, da Agenten in offenen Märkten über den Umgang mit ihnen unbekanntem Agenten mindestens genauso viel Bescheid wissen müssen, wie über die tatsächliche Durchführung von Kauf- und Verkaufsvorgängen. Die Durchführung betriebswirtschaftlicher Transaktionen wird trotz funktionierender technischer Sicherheitsmaßnahmen potentiell wesentlich mehr durch die Möglichkeit beeinträchtigt, dass andere Agenten ihre wahren Präferenzen nicht offenbaren und aus strategischen Gründen falsche und/oder unvollständige Informationen vermitteln. In kleinen, geschlossenen Systemen kann dieses Risiko vernachlässigt werden, in offenen Systemen mit ex ante unbekanntem Agenten jedoch nicht (vgl. Durfee/Rosenschein 1994). Vielversprechende Ansätze zur risikominimierenden dezentralen Kooperation sind durch eine schrittweise Aushandlung von Transaktionen zwischen individualistisch handelnden Agenten gekennzeichnet. Ansätze für schrittweise Kontraktbildung (levelled commitment), die in Kontraktnetz-Protokolle eingefügt werden, sind für DPS-Agentenkonzepte bereits vorgeschlagen worden (Sandholm 1996). Für den von Menschen betriebenen sicheren Electronic Commerce liegen Aushandlungsprotokolle und Erfahrungen mit ihnen vor (Damker et al. 1997). Beide Ansätze gehen davon aus, dass die Agenten (Marktteilnehmer) stets ihre Präferenzen in formalisierter Weise korrekt offenbaren und die Grundlage für einen Interessenausgleich bieten. Ein “Verbergen” von Absichten wird in der Spieltheorie und der betriebswirtschaftlichen Entscheidungslehre behandelt und muss im vorliegenden Fall in formaler Weise für MAS bewertet und eingebunden werden. Ein Mechanismus zur Verfolgung der Reputation unbekannter und möglicherweise unkooperativer Agenten wird von Padovan (2000) vorgestellt.
 - **Vertrauen in den eigenen Agenten:** Entscheidungsaufgaben in elektronischen Märkten werden von menschlichen Anwendern an Agenten delegiert, damit diese sie autonom ausführen. Dabei gehen wir als Menschen zu einem gewissen Grad davon aus, dass diese Aufgaben nach unseren Vorgaben erfüllt werden. Delegation bedeutet aber auch, die Aufgabendurchführung einer anderen Entität mit unterschiedlichen Fähigkeiten, Erfahrungen und Agenda zu übertragen. Das Risiko einer fehlerhaften Handlungsfolge durch den Agenten muß gegenüber dem Vertrauen abgewogen werden, dass der Agent das Richtige tun wird (vgl. Blumberg/Galyean 1995). Diese Entscheidung basiert sowohl auf unserem internen Modell des Handlungskalküls des Agenten sowie auf der Domäne und der durch diese bestimmten möglichen Schadenshöhe.
 - **Kontrolle des Systemverhaltens:** Auf der Systemebene können trotz spezifikationsgemäßem Verhalten im Einzelnen unerwünschte Systemzustände eintreten. Als größtes Risiko kann der Systemzusammenbruch angesehen werden, d.h. der vollständige Verlust

der Unterstützungsleistung bzw. Koordinationsleistung des Systems. Abweichendes Systemverhalten kann endogen durch eine fehlerhafte Konfiguration der Agenten oder durch technisch korrektes, jedoch nicht-linear dynamisches Verhalten der einzelnen Agenten (Parunak et al. 1998) begründet sein. In diesem Fall müssen Maßnahmen zur Systemstabilisation bzw. zur Entfernung der "fehlerhaften" Agenten getroffen werden, soweit diese identifiziert werden können. Das benötigte Regelwerk muß wiederum dem Anwendungsumfeld entnommen werden, da nur in diesem ein normatives Urteil über das Bedrohungsmaß der "Abweichung" gefällt werden kann (bspw. über die "Fehlerhaftigkeit" einer Verhandlungsstrategie). Anders verhält es sich bei Zusammenbrüchen der Netzinfrastruktur oder Ausfällen einzelner Netzteilnehmer, die auf tiefere technische Ebenen zurückzuführen sind. Hier wird davon ausgegangen, dass die Infrastruktur sowohl bei den Netzen als auch bei der eingesetzten Hard- und Software ausfallfrei funktioniert. Dies wird in der Informatik üblicherweise durch Assessment- und Zertifizierungsverfahren sichergestellt (Rannenberg 1999).

6.2.2 Zukünftiger Einsatz von MAS im digitalen Wirtschaften

In der weiteren Entwicklung des Einsatzes von Agenten und MAS speziell in elektronischen Märkten und im Agent-mediated Electronic Commerce hat die Special Interest Group on Agent-mediated Electronic Commerce (SIG-AMEC) der europäischen Forschungsinitiative AgentLink folgende Herausforderungen zusammengestellt, die bei Sierra (2000, S. 8) dargestellt werden:

"The short term development of the field will move a step forward and involve moving into real trading, that is, negotiating deals and making purchases. In order to facilitate this development the researchers and developers are facing a series of challenges:

- Relate ideal and actual agent behaviour with respect to aspects such as rationality, autonomy, situatedness and optimality.*
- Relate roles of agents and their inherent qualities: identity, delegability, liability, and reliability (be they software representatives or mediators).*
- Generate ontologies, interaction standards and social conventions: market institutions; insurance, financial and certification instruments; acceptable interactions and trading conventions.*
- Generate new products, services and practices: market-specific agent shells vs. more generic trading tools; payment and contracting methods; risk-assessment and*

coverage; quality, prestige and performance certification."

Diese Herausforderungen zeigen gleichzeitig die Entwicklungsmöglichkeiten für Avalanche auf, um den Prototyp vielleicht einmal als Softwareplattform für realistische Anwendungen des digitalen Wirtschaftens einsetzen zu können. Ein Schwerpunkt muß auf den Fragen der rechtlichen und technischen Zuverlässigkeit liegen, sowie der Einhaltung von sozialen Konventionen, akzeptierbarem Interaktionsverhalten und Handelsvereinbarungen. Ein Beispiel stellt die Entwicklung eines sozialen Reputationsmechanismus für offene, elektronische Märkte dar (Padovan et al. 2001). Individualistische Agenten verfolgen, per definitionem, einen Handlungsweg, der ihren eigenen Nutzen maximiert. In einer Gemeinschaft individualistischer Agenten ist es erwünscht, dass das globale Systemverhalten akzeptierbar ist, gerade weil oder obwohl jeder der Agenten seinen Nutzen maximiert (Binmore 1992). Folgende Probleme müssen gelöst werden:

1. Agenten können eine gemeinsam genutzte Ressource, wie ein Kommunikationsnetzwerk, so übermäßig in Anspruch nehmen, dass die Ressource und damit jegliche Koordination, die auf ihr basiert, zusammenbricht. Dieses Problem wird die *"tragedy of the commons"* (Hardin 1968) genannt. Die Tragödie ergibt sich daraus, dass jeder Teilnehmer im Versuch, wenigstens noch ein bisschen Anteil an der Ressource zu erlangen, einen immer höheren Anteil für sich reklamiert und damit den Konkurrenzdruck immer weiter erhöht. In ökonomischen Systemen wird dieses Problem durch Internalisierung und Abbildung der Ressource auf einen eigenen Agenten, der Preis- oder Steuermechanismen anwenden kann, gelöst.
2. Eine Gemeinschaft individualistischer Agenten kann komplexes oszillatorisches oder chaotisches Verhalten zeigen (Huberman/Hogg 1988, Thomas/Sycara 1998). Komplexes Verhalten kann bereits in sehr einfachen Computersystemen beobachtet werden. Experimentelle Resultate zeigen, dass unvollständiges Wissen der Agenten oszillatorisches Verhalten unterdrückt, gleichzeitig aber die Effizienz des Systems senkt (Kephart et al. 1998). Derart emergentes Verhalten ist aber nicht notwendigerweise negativ. Richtig umgesetzt, können auch positive Eigenschaften wie Selbststabilisierung und erhöhte Flexibilität aus dem Systemverhalten folgen.
3. Das letzte Problem schließlich, dem sich eigennützige Systeme gegenübersehen, ist es, dass Agenten zu lügen oder täuschen versuchen, um ihren individuellen Nutzen zu vergrößern. Dieses Vorgehen kann einen negativen Effekt auf die gesamte Gesellschaft der Agenten haben (Kephart et al. 1998). Forschung über marktbasierende Modellierung konzentriert sich z.B. auf die implizite Emergenz „fairer“ Preise durch verteilte Markteffekte oder Marktmechanismen, die schrittweise Kontrakte vorsehen (Sandholm/Lesser 1998).

Sollten diese Probleme gelöst werden können, werden autonome Software-Agenten mikroökonomische Einheiten darstellen, welche aus der selbständigen Verfolgung ihrer Nutzenmaximierung heraus ein komplexes und adaptives Netzwerk von Güter- und Geldflüssen bilden. Kephart et al. (2000, S. 24) beschreiben schrittweise, wie sich die heutige Realität agentenbasierten Wirtschaftens in die Richtung der am Anfang dieser Arbeit beschriebenen Vision verändern wird:

1. Heutige⁴² Software-Agenten treffen keineswegs selbständig ökonomische Entscheidungen. Ihre Autonomie ist beschränkt und sie interagieren in großem Maße mit ihrem menschlichen Prinzipal. Beispiele für diese Agenten sind „shopbots“, welche nach vorgegebenen Regeln Preise aus verschiedenen Katalogen heraussuchen, vergleichen und die Ergebnisse dem Anwender präsentieren. Eine Kaufentscheidung wird von ihnen nicht getroffen, ebensowenig wie eine eigenständige Erweiterung des Suchraumes.
2. In einem ersten Schritt werden weitere Phasen des Electronic Commerce automatisiert, um Transaktionskosten der menschlichen Teilnehmer der GII zu minimieren. Phasenmodelle von Marktprozessen (z.B. das Consumer Buying Behavior Model) unterscheiden zwischen Need Identification, Product Brokering, Merchant Brokering, Negotiation, Purchase and Delivery sowie Service and Evaluation. Je mehr dieser Phasen durch Software-Agenten abgedeckt werden, umso niedriger sind die Transaktionskosten für die Menschen. Gleichzeitig steigt jedoch die Komplexität des Agentendesigns.
3. Die zur Abwicklung der ihnen zugewiesenen Teilaufgaben des Marktprozesses benötigten Informationen können aufgrund dieser Komplexität nicht mehr alleine durch den menschlichen Anwender bereitgestellt werden. Software-Agenten werden auch untereinander kommunizieren müssen, um vorhandene Informationen nutzen zu können. Kommunikationsstandards, die diese Aufgabe unterstützen, sind bereits in Entwicklung (vgl. FIPA 1997). Im Vordergrund steht die Frage der Wissensrepräsentation und -übermittlung, wie es im Agentenkontext das "*Knowledge Interchange Format*" zu standardisieren versucht (Genesereth 1991).
4. Die quantitativ und auch qualitativ zunehmende Interaktion zwischen den Software-Agenten führt zum Auftreten von Intermediären, ebenfalls in Form von Software-Agenten. Diese fungieren z.B. als Vermittler, Clearing-Stelle oder Händler. Die Dienste, die diese Software-Agenten anbieten, stellen einen ökonomischen Nutzen für andere Software-Agenten dar – die Intermediär-Agenten werden für die Inanspruchnahme ihrer Dienste bezahlt und generieren Einkommen für ihre menschlichen Prinzipale.

42 Zum Zeitpunkt der Verfassung dieser Arbeit im Herbst 2000.

Schließlich ergibt sich in gewisser Weise ein Abbild der menschen-basierten Ökonomie in der „Information Economy“. Beide Sphären lassen sich nicht voneinander trennen. Die Software-Agenten der „Information Economy“ verfolgen Ziele, die ihnen von ihren menschlichen Prinzipalen vorgeben werden und die diese aus ihrer menschlichen Sphäre beziehen. Die von den Software-Agenten erzielten Ergebnisse fließen in den Nutzen der Prinzipale ein und haben Rückwirkungen auf den Zustand der menschlichen Sphäre:

„Building economic behavior into the agents themselves offers a twofold promise: their myriad interactions and conflicts will be governed by the same economic principles that have lent coherency to the activity of billions of humans; and they will be able to operate in the same economic space as humans, to the benefit of both species” (Kephart et al. 2000, S. 24).

Abkürzungsverzeichnis

Abb.	Abbildung
ACE	Agent-Based Computational Economics
ACM	Association of Computing Machinery
AI	Artificial Intelligence
ALife	Artificial Life
AmEC	Agent-mediated Electronic Commerce
Aufl.	Auflage
Bd.	Band
bearb.	bearbeitet
bspw.	beispielsweise
ca.	circa
CGE	Computable General Equilibrium
CIM	Computer Integrated Manufacturing
CSCW	Computer Supported Cooperative Work
DAI	Distributed Artificial Intelligence
DCE	Distributed Computing Environment
d.h.	dass heißt
Diss.	Dissertation
DPS	Distributed Problem Solving
dt.	in deutscher Sprache
durchges.	durchgesehen
ebda.	ebenda
Ed.	Editor
Eds.	Editors
e.g.	exempli gratia
engl.	in englischer Sprache
erw.	erweitert
et al.	et alias

etc.	et cetera
f.	folgend
ff.	fortfolgend
FB	Fachbereich
FMS	Flexible Manufacturing System
GI	Gesellschaft für Informatik
Habil.	Habilitation
Hrsg.	Herausgeber
ISO	International Standards Organization
IT	Informationstechnik
IuK	Informations- und Kommunikationstechnik
Jg.	Jahrgang
korr.	korrigiert
LAN	Local Area Network
MAS	Multi-Agent Systems
MDCN	Market-Driven Contract Net
mglw.	möglicherweise
MIT	Massachusetts Institute of Technology
MOP	Market-Oriented Programming
neubearb.	neubearbeitet
No.	Number
Nr.	Nummer
o.a.	oben angegeben
Ph.D.	Philosophical Doctor
PC	Personal Computer
Proc.	Proceedings
rev.	revidierte
S.	Seite
Sp.	Spalte
Tab.	Tabelle

u.a.	und andere
überarb.	überarbeitet
Univ.	Universität
u.U.	unter Umständen
vgl.	vergleiche
VKI	Verteilte Künstliche Intelligenz
Vol.	Volume
vollst.	vollständig
vs.	versus
z.B.	zum Beispiel

Literaturverzeichnis

- Adar, E., Huberman, B.A. (2000): Free Riding on Gnutella. *First Monday*, Vol. 5., No. 10. Okt. 2000. http://www.firstmonday.dk/issues/issue5_10/adar/index.html
- Albayrak, S., Garijo, F.J. (Eds.) (1998): *Intelligent Agents for Telecommunication Applications*. LNCS 1437. Springer-Verlag: Berlin, Germany, 1998.
- American Heritage (1996): *The American Heritage® Dictionary of the English Language, Third Edition* Copyright © 1996, 1992 by Houghton Mifflin Company
- Anez, J.C. (1999): *Achieving Synergy: VOYAGER ORB 3.0 from Objectspace*. Java Report Online, SIGS Publications, December 1999. http://www.objectspace.com/press_room/javareport-dec99.asp
- Arrow, K.J., Hahn, F.H. (1971): *General Competitive Analysis*. Amsterdam: North Holland, 1971.
- Austin, C., Pawlan, M. (2000): *Advanced Programming for the Java 2 Platform*. Addison Wesley, 2000, bzw. <http://developer.java.sun.com/developer/onlineTraining/Programming/JDCBook/index.html>
- Axelrod, R. (1995): *Die Evolution der Kooperation*, 3. Aufl., München: Oldenbourg 1995.
- Back, T., Fogel, D. B., and Michalewicz, Z. (1997). *The Handbook of Evolutionary Computation*. Oxford University Press, 1997.
- Baker, A.D. (1995): *Metaphor or Reality: A Case Study where Agents Bid with Actual Costs to Schedule a Factory*. In: Clearwater, S. (Ed.): *Market based control: a paradigm for distributed resource allocation*. World Scientific, Singapore, 1995, S. 184-223.
- Barbuceanu, M., Fox, M.S. (1996): *The Design of a Coordination Language for Multi-Agent Systems*. In J.P. Muller, M.J. Wooldridge, and N.R. Jennings, editors, *Intelligent Agents III. Proceedings of Third International Workshop on Agent Theories, Architectures, and Languages (ATAL'96)*, number 1193 in LNAI. Springer Verlag, August 1996.
- Barbuceanu, M., Fox, M.S. (1997): *Integrating Communicative Action, Conversations and Decision Theory to Coordinate Agents*. In *Agent'97 Conference Proceedings*, Marina del Rey, California, February 1997.
- Barthelmann, K.G. (2000): *Vorlesungsunterlagen Internetprogrammierung und Java 2*. Universität Mainz, 2000. <http://www.Informatik.Uni-Mainz.DE/~barthel/Java/Java2.html>
- Bartos, O.J. (1974): *Process and outcome in negotiation*. New York: Columbia University Press, 1974.
- Bauer, B., Müller, J.P., Odell, J. (2000): *An extension of UML by protocols for multiagent interaction*. Proceedings of the 5th International Conference on Multi-Agent Systems (ICMAS'98), Boston, 2000. http://www.jamesodell.com/ICMAS_2000-US.pdf

- Benton, A.A., Kelley, H.H., Liebling, B. (1972): Effects of extremity of offers and concession-rate on the outcomes of bargaining. *Journal of Personality and Social Psychology*, 1972, Vol. 23, S. 73-83.
- Bigus, J.P., Bigus, J. (1998): *Constructing intelligent agents with Java: a programmer's guide to smarter applications*. New York: Wiley & Sons, 1997.
- Binmore, K.(1992): *Fun and Games. A Text on Game Theory*. D. C. Heath and Company: Lexington, MA, 1992.
- Bleicher, K. (1991): Kooperation als Teil des organisatorischen Kooperationsprozesses, in: Wunderer, R. (Hrsg.): *Kooperation: Gestaltungsprinzipien und Steuerung der Zusammenarbeit zwischen Organisationseinheiten*, Poeschel, Stuttgart 1991, S. 143-157
- Blumberg, B., Galyean, T. (1995): *Do the Right Things...Oh Not That!*, Workshop Notes of the AAAI '95 Spring Symposium on Interactive Story Systems, Stanford University, California, March 1995. <http://bruce.www.media.mit.edu/people/bruce/aaai.ps>
- Bradshaw, J.M.(Ed.) (1997): *Software Agents*. Menlo Park, CA: AAAI Press, 1997.
- Brenner, T. (1999): *Modelling Learning in Economics*. Cheltenham: Edward Elgar, 1999.
- Brooks, R.A. (1986). A robust layered control system for a mobile robot. *IEEE Robotics and Automation*, RA(2): S. 14-23, 1986.
- Budimir, M., Gomber, P. (1999): Dynamische Marktmodelle im elektronischen Wertpapierhandel. In: Scheer, A. W., Nüttgens, M. (Hrsg.): *Electronic Business Engineering, Tagungsband zur Wirtschaftsinformatik'99*. Physica-Verlag, Heidelberg, 1999, S. 251-269.
- Burkhardt, Rainer (1999): *UML - Unified Modeling Language : objektorientierte Modellierung für die Praxis. - 2. aktualis. Aufl. - Bonn: Addison-Wesley, 1999.*
- Burmeister, B., Haddadi, A., Matylis, G. (1997): Applications of multi-agent systems in traffic and transportation. *IEEE Transactions on Software Engineering*, 144(1), S. 51-60, February 1997.
- Bussmann, S., Müller, J. (1992): A Negotiation Framework for Co-operating Agents. In S. M. Deen, editor, *Proceedings of CKBS-SIG*, pages 1-17. Dake Centre, University of Keele, 1992.
- Conte, R., Castelfranchi, C. (1995): Understanding the effects of norms in social groups through simulation. In Gilbert, G.N., Conte, R. (Eds.): *Artificial societies: the computer simulation of social life*. London, UCL Press 1995.
- Chaib-draa, B (1995): Industrial applications of Distributed AI. *Communications of the ACM*, 38(11), S. 47-53, 1995. <http://iris.ift.ulaval.ca/publications/chaib/6.ps.gz>
- Chantemargue, F., Hirsbrunner, B. (1999): A Collective Robotics Application Based on Emergence and Self-Organization, in: *Proceedings of Fifth International Conference for Young Computer Scientists (ICYCS'99)*, 1999. <http://www-iiuf.unifr.ch/pai/axe/Amoc-Papers/icycs99.ps>.
- Chattoe, E. (1998): Just How (Un)realistic are Evolutionary Algorithms as Representations of

- Social Processes?, in: *Journal of Artificial Societies and Social Simulation*, Vol.1, Nr. 3. 1998, <http://www.soc.surrey.ac.uk/JASSS/1/3/2.html>
- Chavez, A., Maes, P. (1996): *Kasbah: an agent marketplace for buying and selling goods*. In *Proceedings of the first international Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*, London, U.K., April 1996.
- Cheng, J.Q., Wellman, M.P. (1998): *The WALRAS algorithm: A convergent distributed implementation of general equilibrium outcomes*. *Computational Economics*, Vol. 12 S. 1-24, 1998.
- Chertkoff, J.M., Conley, M. (1967): *Opening offer and frequency of concession as bargaining strategies*. *Journal of Personality and Social Psychology*, 1967, Vol. 7, S. 181-185.
- Chess, D. M.: *Security Issues in Mobile Code Systems*, in: Vigna G.: *Mobile Agents and Security*, LNCS 1419, S. 1-14, Springer Verlag, Berlin, Heidelberg, 1998
- Clearwater, S. (Ed.) (1995): *Market based control: a paradigm for distributed resource allocation*. World Scientific, Singapore, 1995.
- Cliff, D. (1997): *Minimal-Intelligence Agents for Bargaining Behaviors in Market-Based Environments*. Technical Report Nr. 91, Hewlett-Packard Research Labs, Bristol, 1997. <http://www.hpl.hp.com/techreports/97/HPL-97-91.html>.
- Cliff, D., Bruten, J. (1998): *Less than Human: Simple adaptive trading agents for CDA markets*. In: *Proceedings of the IFAC Symposium on Computation in Economics, Finance, and Engineering: Economic Systems (CEFES'98)*, Cambridge, UK, June 29-July 1, 1998. <http://www.hpl.hp.com/techreports/97/HPL-97-155.pdf>
- Conry, S.E., Meyer, R.A., Lesser, V.R. (1986): *Multistage Negotiation in Distributed Planning*. Technical report, COINS, University of Massachusetts, Amherst Boston, 1986.
- Corkill, D.D. (1991): *Blackboard Systems*, *AI Expert*, Vol. 6 No. 9, S.40-47, September, 1991.
- Corkill, D.D., Lesser, V. R. (1986): *The Use of Meta-Level Control for Coordination in a Distributed Problem Solving Network*. *Proc. Int. Joint Conf. on AI*, Karlsruhe, Germany, S. 748-756.
- Cournot, A.A. (1838): *Recherches sur les principes mathématiques de la théorie des richesses*. Paris: Hachette, 1838.
- Crutchfield, J.P. (1994): *The calculi of emergence: computation, dynamics and induction*. In: *Physica D*, 75 (1994), S. 11-54
- Damker, H.; Federrath, H.; Reichenbach, M.; Bertsch, A. (1997): *Persönliches Erreichbarkeitsmanagement*, in: Müller, G.; Pfitzmann, A. (Hrsg.): *Mehrseitige Sicherheit in der Kommunikationstechnik: Verfahren, Komponenten, Integration*, Addison-Wesley, München, 1997.
- De Paula, G.E., Ramos, F.S., Ramalho, G.L. (2000): *Bilateral Negotiation Model for Agent Mediated Electronic Commerce*. In: Dignum, F. (Ed.): *Proc. Workshop on Agent Mediated Electronic Commerce III*, 4th Intl. Conf. on Autonomous Agents, Barcelona, Spain, June 3 - June 7, 2000.

- Deutsche Börse 2000 (Hrsg.) , Xetra Marktmodell Aktien, Release 4.0, <http://www.xetra.de>
- Dosi, G., Nelson, R.R. (1994): An Introduction to Evolutionary Theories in Economics. *Journal of Evolutionary Economics*, Vol. 4, S. 153-172.
- Druckman, D. (1977): *Negotiations*. Beverly Hills: Sage, 1977.
- Durfee, E.H., Rosenschein, J.S. (1994): Distributed Problem Solving and Multi-Agent Systems: Comparisons and Examples, in: *Proceedings of the 13th International Distributed Artificial Intelligence Workshop*, S. 94-104, 1994. <ftp://ftp.eecs.umich.edu/people/durfee/daiw94-dr.ps.Z>
- Durfee, E.H. (1988): *Coordination of Distributed Problem Solvers*, Kluwer Academic Press, Boston 1988.
- Durfee, E.H. (1999). In Weiss, G. (Ed.): *Multiagent Systems*, MIT Press, 1999, S. 121-164.
- Durfee, E.H., Lesser, V.R. (1989): Negotiating task decomposition and allocation using partial global planning. In: Gasser, L., Huhns, M., (Eds.): *Distributed Artificial Intelligence Volume II*, pages 229–244. Pitman Publishing: London and Morgan Kaufmann: San Mateo, CA, 1989.
- Durfee, E.H., Lesser, V.R., Corkill, D.D. (1987). Coherent Cooperation among Communicating Problem Solvers. *IEEE Transactions on Computing*, 36(11):1275-1291, 1987.
- Durfee, E.H., Montgomery, T.A. (1990): A Hierarchical Protocol for Coordinating Multi-agent Behaviour. In *Proceedings of the 8th National Conference on Artificial Intelligence*, pages 86-93, Boston, MA, 1990.
- Effelsberg, W., Fleischmann, A. (1986): Das ISO-Referenzmodell für offene Systeme und seine sieben Schichten, in: *Informatik-Spektrum Nr. 9*, 1986, S. 280–299.
- Eymann, T., Kearney, P.J. (2000): Market-based decentralised process management using multi-agent systems. In: *AgentLink Newsletter No. 5*, May 2000, pp. 14-16.
- Eymann, T., Padovan, B. (1999): Eine Multi-Agenten-Simulation zur ökonomischen Analyse der dezentralen Koordination von Wertschöpfungsketten, in: Scheer, A.-W., Nüttgens, M. (Hrsg.): *Electronic Business Engineering, Konferenzband zur 4. Internationalen Tagung Wirtschaftsinformatik (WI'99)*, 3.-5. März 1999, Saarbrücken. Physica Verlag: Heidelberg 1999, S. 625-642.
- Eymann, T., Padovan, B., Schoder, D. (1998a): The Living Value Chain - Coordinating Business Processes with Artificial Life Agents, in: *Proceedings of the Third International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agents (PAAM'98)*, March 23-25, London 1998, S. 111-122.
- Eymann, T., Padovan, B., Schoder, D. (1998b): Simulating Value Chain Coordination with Artificial Life Agents, in: Demazeau, Yves (Ed.): *Proceedings of the 3rd International Conference on Multi-Agent Systems (ICMAS'98)*, July 2-8, Paris 1998, IEEE Computer Society Press: Los Alamitos, CA, 1998, S. 423- 424.
- Eymann, T., Padovan, B., Schoder, D. (2000): The Catallaxy as a new paradigm for the Design of Information Systems. In: *Proceedings of the 16th IFIP World Computer Congress*,

- Conference on Intelligent Information Processing, Beijing/ PR China, August 21-25, 2000.
- Fausett, L.V. (1994): Fundamentals of Neural Networks. Prentice Hall, 1994.
- Fenster, M., Kraus, S., Rosenschein, J.S. (1995): Coordination without communication: experimental validation of focal point techniques. Proc. ICMAS95, S. 102-108
- Ferber, Jacques (1994): Simulating with reactive agents, in: Hillebrand, E.; Stender, J. (Eds.): Many-Agent Simulation and Artificial Life, Amsterdam: IOS-Press, 1994, S. 8-30.
- Ferguson, D.F., Nickolaou, C., Sairamesh, J., Yemini, Y. (1995): Economic Models for Allocating Resources in Computer Systems. In: Clearwater, S. (Ed.): Market-Based Control: A Paradigm for Distributed Resource Allocation. Singapore: World Scientific, 1996, S. 156-183.
- Finin, T., Labrou, Y., Mayfield, J. (1997): KQML as an agent communication language. in: Bradshaw, J. (Ed.), ``Software Agents'', MIT Press, Cambridge, 1997. <http://www.cs.umbc.edu/kqml/papers/kqmlacl.pdf>
- FIPA - Foundation for Intelligent Physical Agents, Agent Communication Technical Committee (1997): Agent Communication Language. FIPA '97 Draft Specification Part 2. <http://www.fipa.org>
- FIPA - Foundation for Intelligent Physical Agents (1998). FIPA 98 Specification, Part 10, Version 1.0, Agent Security Management, Genf, 1998
- Fischer, K., Müller, J.P., Heimig, I., Scheer, A.W. (1996): Intelligent agents in virtual enterprises. Proceedings of PAAM-96.
- Fischer, K., Müller, J.P., Pischel, M. (1996): Cooperative Transportation Scheduling: An Application Domain for DAI, Journal of Applied Artificial Intelligence. Special Issue on Intelligent Agents, 10/1, 1996. <ftp://ftp.dfki.uni-sb.de/pub/MAGSY/Papers/ICMAS95.ps.gz>
- Fischer, K., Müller, J.P., Pischel, M., Schier, D. (1995): A model for cooperative transportation scheduling. Proceedings of ICMAS-95, S. 109-116.
- Foner, L.N. (1997): A Security Architecture for Multi-Agent Matchmaking, Proceedings of the 2nd International Conference on Multi-Agent Systems (ICMAS '97), Kyoto, Japan 1997.
- Foner, L.N. (1999): Political Artifacts and Personal Privacy: The Yenta Multi-Agent Distributed Matchmaking System. PhD Thesis, MIT 1999. <http://foner.www.media.mit.edu/people/foner/PhD-Thesis/Dissertation/>
- Foster, J. (2000): Competitive selection, self-organisation and Joseph A. Schumpeter, in: Journal of Evolutionary Economics, Vol. 10, S.311-328. Heidelberg: Springer, 2000
- Foster, J., Wild, P. (1994): Self-Organisation and Synergetics in Economics. The Re-Vitalisation of Economic Science?, in: ESGEE (Ed.): Papers on Economics & Evolution, . Freiburg: 1994
- Franklin, S., Graesser, A. (1996): Is It an Agent or Just a Program? A Taxonomy for Autono-

- mous Agents, in: Proceedings of the Third International Workshop on Agent Theories, Architectures and Languages. New York: Springer 1996.
- Friedman, D. (1991): Evolutionary Games in Economics, in: *Econometrica*, Vol. 59, S. 637-666
- Friedman, D.P. (1993): The Double Auction Market. Santa Fe Institute studies in sciences of complexity, Vol. 14. Reading, Mass.: Addison-Wesley, 1993.
- Gagliano, R.A., Mitchem, P.A. (1995): Valuation of Network Computing Resources. In: Clearwater, S. (Ed.): Market based control: a paradigm for distributed resource allocation. World Scientific, Singapore, 1995, S. 28-52.
- Gasser, L. (1991): Social Conceptions of knowledge and action: DAI foundations and open systems semantics. *Artificial Intelligence* 47 (1991), S. 107-138. Abgedruckt in: Huhns, M.N., Singh, M.P. (1998): Readings in Agents. Morgan Kaufmann, San Francisco, 1998, S. 389-404.
- Genesereth, M.R. (1991): "Knowledge Interchange Format", Proceedings of the Second International Conference on the Principles of Knowledge Representation and Reasoning, Allen, J., Fikes, R., Sandewall, E. (eds), Morgan Kaufman Publishers, 1991, pp 238-249.
- Genesereth M.R., Nilsson N.J. (1987): Logical Foundations of Artificial Intelligence. Morgan Kaufman, Los Altos, 1987.
- Genesereth, M.R., Ketchpel, S.P. (1994): Software Agents. In: *Communications of the ACM*, Vol. 37, S. 48-53, July 1994.
- Georgeff, M. (1983): Communication and Interaction in Multi-agent Planning. In Proceedings of the 1983 National Conference on Artificial Intelligence, S. 125-129, 1983.
- Gibney, M.J., Jennings, N.R., Vriend, N.J., Griffiths, J. M. (1999): "Market-Based Call Routing in Telecommunications Networks using Adaptive Pricing and Real Bidding" Proc. 3rd Int. Workshop on Multi-Agent Systems and Telecommunications (IATA-99), Stockholm, Sweden, S. 50-65. <http://www.ecs.soton.ac.uk/~nrj/download-files/iata99.pdf>
- Gilbert, D., Aparicio, M., Atkinson, B., Brady, S., Ciccarino, J., Grosf B., O'Connor, P., Osisek, D., Pritko, S., Spagna, R., Wilson, L. (1995): IBM Intelligent Agent Strategy, UNICOM Seminar on Agent Software. IBM Corporation 1995.
- Gimenez, E., Godo L., Rodríguez-Aguilar, J.A., Garcia P. (1998): Designing Bidding Strategies for Trading Agents in Electronic Auctions. In Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS-98), S. 136-143.
- Gode, D. K., Sunder, S. (1993). Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. *Journal of Political Economy*, Vol. 101 No.1, S. 119-137.
- Goldberg, D. (1989): Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley, Reading MA, 1989.

- Gomber, P., Schmidt, C., Weinhardt, C. (1998): Auctions in Electronic Commerce - Efficiency versus Computational Tractability. In: Proceedings of ICEC '98, . Seoul, Korea 1998.
- Gravelle, H., Rees, R. (1992): Microeconomics. 2. ed. London: Longman, 1992.
- Gutmann, J.-S., Hatzack, W., Herrmann, I., Nebel, B. , Rittinger, F. , Topor, A., Weigel, T. (2000): The CS Freiburg Team: Playing Robotic Soccer Based on an Explicit World Model, The AI Magazine, 21(1), S. 37-46, 2000. <ftp://ftp.informatik.uni-freiburg.de/documents/papers/ki/gutmann-et-al-aim00.pdf>
- Guttman, R., Moukas, A., Maes, P. (1998): Agent Mediated Electronic Commerce: A Survey. Knowledge Engineering Review, June 1998. <http://ecommerce.media.mit.edu/papers/ker98.pdf>
- Hahn, D. (2000): Problemfelder des Supply Chain Management. In: Wildemann, H. (Hrsg.): Supply Chain Management, München: TCW Transfer-Centrum-Verlag, 2000, S. 9-20.
- Haken, H. (1990): Synergetik: eine Einführung. 3., erw. Aufl. Berlin; Heidelberg: Springer, 1990.
- Hall, L. (Ed.) (1993): Negotiation: Strategies for Mutual Gain. Beverly Hills: Sage 1993.
- Hamner, W.C. (1974): Effects of bargaining strategy and pressure to reach agreement in a stalemated negotiation. Journal of Personality and Social Psychology, 1974, Vol. 30, S. 458-467.
- Hardin, G. (1968): The tragedy of commons. Science, 162:1243, 1968.
- Harnett, D.L., Vincelette, J.P. (1978): Strategic influences on bargaining effectiveness. In: Sauermann, H. (Ed.): Contributions to experimental economics (Vol. 7). Tübingen: Mohr, 1978.
- Harsanyi, J.C. (1977): Rational Behavior and Bargaining Equilibrium in Games and Social Situations. Cambridge: Cambridge University Press, 1977.
- Hayek, F. A. (1988): The fatal conceit: the errors of socialism. Chicago: Univ. of Chicago Press, 1988.
- Hayek, F.A. (1945): The Use of Knowledge in Society, in: American Economic Review, XXXV, No. 4, September 1945, S.519-530. 1945, <http://www.virtualschool.edu/mon/Economics/Hayek/UseOfKnowledge.html>.
- Hayer-Roth, B. (1985): A Blackboard Architecture for Control. Artificial Intelligence, 25:251-321, 1985.
- Hobbes, Th. (1651/1980): Leviathan. München: Reclam, 1980. Nachdruck der Ausgabe von 1651.
- Hogg, T., Huberman, B.A. (1991): Controlling chaos in distributed systems. IEEE Transactions on Systems, Man, and Cybernetics, 21:1325, 1991. <http://www.parc.xerox.com/istl/groups/iea/www/controllingChaos.html>
- Holland, J. (1992): Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, 2nd Ed., MIT Press/

- Bradford Books, Cambridge, MA, 1992.
- Holland, J.H., Miller, J. (1991): Artificial Adaptive Agents in Economic Theory, in: American Economic Review, Papers and Proceedings, Vol. 81, No. 2, 1991, S. 365-370.
- Holland, John H. (1992): Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control and artificial intelligence, 2nd print, Cambridge: MIT Press, 1992.
- Holler, M.J., Illing, G. (1996): Einführung in die Spieltheorie, 3. Auflage, Heidelberg: Springer 1996.
- Honerkamp, J. (1990): Stochastische dynamische Systeme: Konzepte, numerische Methoden, Datenanalysen. Weinheim: VCH, 1990.
- Hopcroft, J.E., Ullman, J. (1994): Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie. 3. korrigierte Aufl. - Bonn : Addison-Wesley, 1994.
- Hoppmann, E. (1999): Unwissenheit, Wirtschaftsordnung und Staatsgewalt, in: Vanberg, V. (Hrsg.): Freiheit, Wettbewerb und Wirtschaftsordnung, Freiburg: Haufe Verlag, 1999, S. 135-169.
- Huberman, B. A., Lukose, R. M. (1997): Social Dilemmas and Internet Congestion. In: Science Magazine, Vol. 277, 1997.
- Huberman, B.A. (Ed.) (1988): The Ecology of Computation, Amsterdam, 1988. North-Holland.
- Huberman, B.A., Hogg, T. (1988): The behavior of computational ecologies. In: Huberman, B.A. (Ed.): The Ecology of Computation. North-Holland, Amsterdam, 1988.
- Huberman, B.A., Lukose, R.M., Hogg, T. (1996): An economics approach to hard computational problems. Science, No. 275, S. 51-54, 1996.
- Huhns, M., Singh, M.P. (1994): CKBS-94 Tutorial. In Distributed Artificial Intelligence for Information Systems. Dake Centre, University of Keele, 1994
- Huhns, M.N., Stephens, L.M. (1999): Multiagent Systems and Societies of Agents. In: Weiss, G. (Ed.): Multiagent Systems, MIT Press, 1999, S. 79-120.
- Hummel, Thomas (1996): Informationstechnische Unterstützung lateraler Kooperation - Empirische Studie und Erklärungsansatz, Wiesbaden: Deutscher Universitäts Verlag, 1996.
- Jennings, N.R., Wooldridge, M. (1998): Applying agent technology. In: Jennings, N. R., Wooldridge, M., (Eds.): Agent Technology: Foundations, Applications, and Markets, S. 3-28. Springer-Verlag: Berlin, Germany, 1998. <ftp://ftp.elec.qmw.ac.uk/pub/isag/distributed-ai/publications/agt-technology.pdf>
- Jennings, N.R. (1996): Coordination Techniques for Distributed Artificial Intelligence. In G.M.P. O'Hare and N.R. Jennings, editors, Foundations of Distributed Artificial. John Wiley and Sons, 1996.
- Jennings, N.R., Faratin, P., Johnson, M.J., O'Brien, P., Wiegand, M.E (1996): Using Intelligent

- Agents to Manage Business Processes. In: Proceedings of PAAM96 (London 1996), S. 345-360. <ftp://ftp.elec.qmw.ac.uk/pub/isag/distributed-ai/publications/BCS96.ps.gz>
- Jennings, N.R., Sycara, K., Wooldridge, M. (1998): A Roadmap of Agent Research and Development. *Autonomous Agents and Multi-Agent Systems Journal*, No. 1, 1998, S. 275–306, Kluwer Academic Publishers. <ftp://ftp.elec.qmw.ac.uk/pub/isag/distributed-ai/publications/roadmap.pdf>
- Jeon, H., Petrie, C., Cutkosky, M. R.: JATLite: A Java Agent Infrastructure with Message Routing, *IEEE Internet Computing*, Mar/Apr 2000.
- Kalakota, R., Whinston, A.B. (1996): *Frontiers of Electronic Commerce*. Reading: Addison-Wesley, 1996.
- Karnopp, D.C., Margolis, D.L., Rosenberg, R.C. (1990): *System dynamics: a unified approach*. 2. Ed. New York, NY : Wiley, 1990.
- Kauffman, S. (1995): *At Home in the Universe*, New York: Oxford University Press, 1995.
- Kauffman, S. (1991): *Origins of Order*. Oxford: Oxford University Press, 1991.
- Kearney, P., Sehmi, A., Smith, R. (1994): Emergent Behaviour in a Multi-agent Economics Simulation. In A. G. Cohn (Ed.): *Proceedings of the 11th European Conference on Artificial Intelligence*. John Wiley, 1994.
- Keeney, R., Raiffa, H. (1976): *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. Wiley & Sons, 1976.
- Kephart, J.O., Hanson, J.E., Levine, D.W.: Grosz, B.N., Sairamesh, J., Segal, R.B., White, S.R. (1998): Dynamics of an information-filtering economy. in: Klusch, M., Weiß, G. (Ed.): *Cooperative Information Agents II*. LNAI No. 1435. Heidelberg: Springer, 1998. http://www.ibm.com/iac/papers/cia98/cia98_public.html
- Kephart, J.O., Hanson, J.E., Greenwald, A.R. (2000): Dynamic Pricing by Software Agents. *Computer Networks*, 2000 (to appear). <http://www.research.ibm.com/infoecon/paps/html/rudin/rudin.html>
- Kieser, A., Kubicek, H. (1992): *Organisation*, 3. völlig neu bearb. Aufl., Walter de Gruyter, Berlin, New York 1992
- Kirman, A. (1998): Self-Organization and Evolution in Economics. In: Schweitzer F., Silverberg G.: *Evolution und Selbstorganisation in der Ökonomie*. Berlin: Duncker und Humblot, 1998. S. 13-46.
- Klein, M. (1997): *Coordination Science: Challenges and Directions*, in: Conen, W., Neumann, G. (Ed.): *Coordination Technology for Collaborative Applications*, S.161-176. Heidelberg: Springer, 1997
- Klimecki, R. (1984): *Laterale Kooperation - Grundlagen eines Analysemodells horizontaler Arbeitsbeziehungen in funktionalen Systemen*, Diss., Essen und Bern 1984
- Kraus, S. (1997): Negotiation and cooperation in multi-agent environments, *Artificial Intelligence* 94 (1997), S. 79-97. <http://www.cs.biu.ac.il:8080/~sarit/Articles/ct.ps>

- Kraus, S., Lehmann, D. (1995): Designing and building a negotiating automated agent. *Computing Intelligence*, Vol. 11 (1995), S. 132-171.
- Kraus, S., Nirkhe, N., Sycara, K.P. (1993): Reaching agreements through argumentation: a logical model. *Proceedings of DAI-93, LIT 1993*.
- Kraus, S., Wildenfield, J. (1991): *The Function of Time in Cooperative Negotiations: Preliminary Report*. Technical report, Department of Computer Science, University of Maryland, 1991.
- Kreifelt, T., von Martial, F. (1991): A negotiation Framework for Autonomous Agents. In Y. Demazeau and J. P. Mueller, editors, *Proceedings of Decentralized AI2*. Elsevier Science, 1991.
- Kumar, M., Feldman, S.I. (1998): Business Negotiations on the Internet, in: IBM Institute for Advanced Commerce (Ed.): Technical Report. March 11, 1998. <http://www.ibm.com/iac/reports-technical/reports-bus-neg-internet.html>.
- Kurose, J.F., Schwartz, M., Yemini, Y. (1985): A microeconomic approach to decentralized optimization of channel access policies in multiaccess networks. *Proceedings of the 5th International Conference on Distributed Computing Systems*, Denver, 13.-17. Mai 1985.
- Kuwabara, K., Ishida, T., Nishibe, Y., Suda, T. (1995): An equilibratory market-based approach for distributed resource allocation and its applications to communication network control. In Clearwater, S. (Ed.): *Market-Based Control: A Paradigm for Distributed Resource Allocation*. Singapore: World Scientific, 1996, S. 53-73
- Labrou, Y., Finin, T. (1997): A Proposal for a new KQML Specification. University of Maryland, Computer Science and Electrical Engineering Dept., Technical Report CS-97-03, February 1997. <http://www.cs.umbc.edu/kqml/>.
- Lane, D. A. (1993): Artificial worlds and economics (Part II), in: *Journal of Evolutionary Economics*, S. 177-197. Springer 1993.
- Laßmann, A. (1992): *Organisatorische Koordination: Konzepte und Prinzipien zur Einordnung von Teilaufgaben*, Diss., Univ. Köln, Gabler, Wiesbaden 1992
- Lee, L., Nwana, H.S., Jennings, N.R. (1997): Co-ordination in Multi-Agent Systems. In H. S. Nwana and N. Azarmi, editors, *Software Agents and Soft Computing*, number 1198 in LNAI. Springer Verlag, 1997.
- Lesser, V.R., Corkill, D.D. (1981): Functionally accurate, cooperative distributed systems. *IEEE Transactions on Systems, Man and Cybernetics*. SMC-11 (1), S. 81-96, 1981.
- Levinson, S.C. (1983): *Pragmatics*. Cambridge: Cambridge Univ. Press, 1983.
- Lieberman, H. (1995): Letizia: An Agent That Assists Web Browsing, *Proceedings of the International Joint Conference on Artificial Intelligence*, Montreal, August 1995. <http://lieber.www.media.mit.edu/people/lieber/Lieberary/Letizia/Letizia-AAAI/Letizia.html>
- Lieberman, H. (1997): Autonomous Interface Agents, *Proceedings of the ACM Conference on Computers and Human Interface, CHI-97*, Atlanta, Georgia, March 1997. <http://lieber>

- ber.www.media.mit.edu/people/lieber/Lieberary/Letizia/AIA/AIA.html
- Ljunberg, M., Lucas, A. (1992): The OASIS air traffic management system. In Proceedings of the Second Pacific Rim International Conference on AI (PRICAI-92), Seoul, Korea, 1992.
- Loasby, B.J. (2000): Market institutions and economic evolution, in: *Journal of Evolutionary Economics*, Vol. 10, S.297-309. Heidelberg: Springer, 2000
- Luce, R.D., Raiffa, H. (1957): *Games and Decisions*. John Wiley and Sons, 1957.
- Maes, P. (1993): Behavior-Based Artificial Intelligence. In Proceedings of the Fifteenth Annual Meeting of the Cognitive Science Society, pages 74-83, Hillsdale, NJ, 1993. Lawrence Erlbaum.
- Maes, P. (1994a): Agents that reduce work and information overload. *Communications of the ACM*, Vol. 37 (1994), No. 7, S. 30-40.
- Maes, P. (1994b): Modeling Adaptive Autonomous Agents. In: Langton, C.G. (Ed.): *Artificial Life Journal*, Vol. 1, No. 1 & 2, S. 135-162, MIT Press, Cambridge, 1994.
- Malone, T.W. (1987): Modelling Coordination in Organizations and Markets. *Management Science* 33, S. 1317-1332.
- Malone, T.W., Crowston, K. (1990): What is Coordination Theory and How Can It Help Design Cooperative Work Systems, in: *CSCW 90 Proceedings*, October 1990, S. 357-370
- Malone, T.W., Crowston, K.. (1994): The Interdisciplinary Study of Coordination, in: *ACM Computing Surveys*, Vol. 26, No. 1, March 1994, S. 87-119
- Malone, T.W., Yates, J., Benjamin, R.I. (1987). Electronic Markets and Electronic Hierarchies: Effects of New Information Technologies on Market Structures and Corporate Strategies. In: *Communications of the ACM*, 30 (1987), S. 484-497.
- Maturana, H. Varela, F.J. (1987): *The Tree of Knowledge: The biological roots of human understanding*. New Science Library, Boston, MA, 1987.
- Ménard, C. (1995): Markets as institutions versus organisations as markets? Disentangling some fundamental concepts. *Journal of Economic Behavior and Organization*, Vol. 28 (1995), S. 161-182.
- Miller, J.H. (1996): *Evolving Information Processing Organizations*. <http://zia.hss.cmu.edu/econ/misc/evolorg.html>
- Miller, M.S., Drexler, K.E. (1988): Markets and computation: agoric open systems. In: Huberman, B. A. (Ed.): *The Ecology of Computation*, Amsterdam, 1988. North-Holland. <http://www.agorics.com/agorpapers.html>
- Minsky, M. (1981): *The Society of Mind*. MIT Press, 1981.
- Mirowski, P., Somefun, K. (1998): Markets as evolving computational entities, in: *Journal of Evolutionary Economics* (1998) , No. 8, S. 329-356
- Mitchell, M., Forrest, S. (1994): Genetic Algorithms and Artificial Life, in: *Artificial Life 1*

- (1994), S. 267-289.
- Montgomery, T.A., Durfee, E.H. (1993): Search Reduction in Hierarchical Distributed Problem Solving. *Group Decision and Negotiation*, Vol. 2, S. 301-317, 1993.
- Mouaddib, A.I. (1997): Progressive Negotiation For Time-Constrained Autonomous Agents. In *Agent'97 Conference Proceedings*, Marina del Rey, California, February 1997.
- Moulin, H. (1995): *Cooperative microeconomics - a game-theoretic introduction*. London: Prentice Hall Harvester Wheatsheaf, 1995.
- Müller, G. (1992): Strategic Directions towards Multimedia Systems, *Arbeitsberichte des Instituts für Informatik und Gesellschaft (Freiburger Hefte)* 1992.
- Müller, G., Kohl, U., Schoder, D. (1997): *Unternehmenskommunikation: Telematiksysteme für vernetzte Unternehmen*. Bonn: Addison-Wesley-Longman, 1997.
- Müller, G., Rannenber, K. (1999): Security for Global Communications. In: Müller, G., Rannenber, K. (Eds.): *Multilateral Security in Communications, Volume 3: Technology, Infrastructure, Economy*. München: Addison Wesley Longman, 1999, S. 11-20.
- Nash, J.F. (1950): The bargaining problem. *Econometrica*, Vol. 18, S. 155-62.
- Neal, R.M. (1996): *Bayesian Learning for Neural Networks*. Lecture Notes in Statistics, Vol 118. Heidelberg: Springer Verlag, 1996.
- Norrie, M.C., Wunderli, M. (1995): Modelling in Coordination Systems. *Intl. Journal of Cooperative Information Systems*. 1995 Vol. 4. Nr. 2 & 3, S. 189-211.
- North, D.C. (1998): *Institutionen, institutioneller Wandel und Wirtschaftsleistung*. Nachdruck. Tübingen : Mohr, 1998.
- Nozick, R. (1974): *Anarchy, state and utopia*. New York: Basic Books, 1974.
- Nwana, H.S. (1996): Software Agents: An Overview. *The Knowledge Engineering Review*, No. 11:3, 1996, S. 205-244. <http://www.labs.bt.com/projects/agents/publish/papers/agentreview.htm>
- Nwana, H.S., Ndumu, D.T. (1999): A Perspective on Software Agents Research, in: *Knowledge Engineering Review*, January 1999. <http://www.labs.bt.com/projects/agents/publish/papers/hn-dn-ker99.zip>
- Odell, J., Parunak, H.V.D., Bauer, B. (2000): Extending UML for Agents. In: Gerd Wagner, Yves Lesperance, and Eric Yu (Eds.): *Proc. of the Agent-Oriented Information Systems Workshop at the 17th National Conference on Artificial Intelligence*. Austin, TX, pp. 3-17, accepted paper, AOIS Workshop at AAAI 2000. <http://www.jamesodell.com/ExtendingUML.pdf>
- O'Driscoll, G.P., Rizzo, M.J. (1985): *The economics of time and ignorance*. Oxford/New York: Blackwell, 1985.
- Objectspace (1997): *ObjectSpace VOYAGER Technical Overview*, White Paper. Ed.: ObjectSpace. <http://www.objectspace.com/voyager/VOYAGER-Tech-Overview.pdf>, 1997.
- Objectspace (1998): *Objectspace VOYAGER Quick Start Guide*. Version 2.0 Beta 2. <http://>

- www.objectspace.com/voyager/prodVOYAGER.asp, 1998.
- Osborne, M.J., Rubinstein, A. (1990): *Bargaining and Markets*. San Diego: Academic Press, 1990.
- Padovan, B., Sackmann, S., Eymann, T. (2001): A prototype for an agent based secure electronic marketplace including reputation tracking mechanisms. In Sprague, R.H. (Ed.): *Proceedings of the 34th Hawaii International Conference on System Sciences*, January 3-6, 2001 (erscheint).
- Padovan, B. (2000): *Ein Reputationsmechanismus für offene Multi-Agenten-Systeme*. Dissertation, Freiburg, erscheint 2000.
- Page, S.E. (1999): *Computational Models from A to Z*, Keynote Lecture, SwarmFest 1999, Anderson Business School, UCLA, March 26, 1999. <http://www.santafe.edu/projects/swarm/swarmfest99/keynote.html>
- Partl, H. (2000): *Java-Einführung*, Juli 2000. <http://www.boku.ac.at/javaeinf/jein.html>.
- Parunak, H. V. D. (1998): *Industrial and practical applications of DAI*. In: Weiß, G. (Ed.): *Multi-Agent Systems*. The MIT Press: Cambridge, MA, 1998. <http://www.erim.org/~van/apps98.pdf>
- Parunak, H. V. D.; Savit, R.; Riolo, R.L. (1998): *Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and Users' Guide*, in: Sichman, J. S., Conte, R., Gilbert, N. (Ed.): *Multi-Agent Systems and Agent-Based Simulations*, LNAI 1534, S.10-25, Heidelberg: Springer, 1998
- Parunak, H.V.D. (1989): *Manufacturing Experiences with the Contract Net*. In L. Gasser and M. Huhns, editors, *Distributed Artificial Intelligence 2*. Morgan Kaufmann, 1989.
- Petrie, C.J. (1996): *Agent-Based Engineering, the Web, and Intelligence*. *IEEE Expert*, Vol. 11, No. 6, S. 24-29, Dezember 1996. <http://cdr.stanford.edu/NextLink/Expert.html>
- Porter, M. E., Millar, V. E. (1985): *How information gives you competitive advantage*. In: *Harvard Business Review*, July-August 1985, S.149-160. Cambridge 1985.
- Porto, V. W. (1997). *Evolutionary Programming*. In: Back, T., Fogel, D. B., and Michalewicz, Z. (1997). *The Handbook of Evolutionary Computation*. Oxford: University Press, 1997, S. B1.4: 1-10.
- Preist, C. (1998): *Economic Agents for Automated Trading*. HP Technical Reports HPL-98-77. Hewlett Packard Laboratories, Bristol, 1998. <http://www.hpl.hp.com/techreports/98/HPL-98-77.html>
- Pruitt, D.G. (1981): *Negotiation Behavior*. New York: Academic Press, 1981.
- Rannenbergh, K. (1999): *What can IT-Security Certification do for Multilateral Security?*, in: Müller, G.; Rannenbergh, K. (Hrsg.): *Multilateral Security in Communications - Technology, Infrastructure, Economy*, Addison-Wesley, München, Reading MA, u.a.O., 1999
- Rao, A., Georgeff, M. (1995): *"BDI agents: From theory to practice,"* In: *ICMAS-95: First International Conference on Multi-Agent Systems*: July 12-14, 1995, San Francisco,

- California: Proceedings. Menlo Park, Calif.: AAAI Press; Cambridge, Mass.: MIT Press, S. 312-319.
- Rasmusson, L., Janson, S. (1999): Agents, self-interest and electronic markets. *Knowledge Engineering Review*, Vol. 14 (2), 1999, S. 143-150.
- Rasmusson, L., Rasmusson, A., Janson, S. (1997): Using Agents to Secure the Internet Marketplace - Reactive Security and Social Control, Proceedings of the 2nd International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agents (PAAM'97), London 1997.
- Ropohl, G. (1999): *Allgemeine Technologie: eine Systemtheorie der Technik*. 2. Aufl., München: Hanser, 1999.
- Rosenschein, J.S., Zlotkin, G. (1994a): *Rules of Encounter*. Cambridge: MIT Press, 1994.
- Rosenschein, J.S., Zlotkin, G. (1994b): Designing Conventions for Automated Negotiation. In: *AI Magazine*, Fall 1994, S. 29-46.
- Ross, S. (1973): The Economic Theory of Agency: The Principal's Problem. *American Economic Review* 63, May 1973, S. 134-139.
- Rötzer, F. (2000): E-Commerce-Websites lahmgelegt. *Telepolis aktuell*, 09.02.2000, <http://www.heise.de/tp/deutsch/inhalt/te/5766/1.html>
- Rubinstein, A. (1982): Perfect equilibrium in a bargaining model. *Econometrica* 50 (1982), S. 97-109.
- Rudolph, G. (1997). Evolution Strategies. In Back, T., Fogel, D. B., and Michalewicz, Z. (1997). *The Handbook of Evolutionary Computation*. Oxford: University Press, 1997, S. B1.3:1-6.
- Rumelhart, D., Hinton, G., Williams, R. (1986): Learning internal representations by error propagation. In: Rumelhart, D., McClelland, J.: *Parallel Distributed Processing*, Vol. 1: Foundations. MIT Press, 1986.
- Russel, S.J., Norvig, P. (1995): *Artificial Intelligence. A Modern Approach*. Englewood Cliffs: Prentice-Hall, 1995.
- Samuelson, P. (1947): *Foundations of Economic Analysis*. Cambridge: Harvard Univ. Press, 1947.
- Sandholm, T.W. (1996a): Negotiation among self-interested computationally limited agents. Dissertation, Amherst MA, 1996. <http://siesta.cs.wustl.edu/~sandholm/dissertation.ps>
- Sandholm, T.W. (1996b): Limitations of the Vickrey Auction in Computational Multiagent Systems. Proceedings of the Second International Conference on Multiagent Systems (ICMAS-96), Keihanna Plaza, Kyoto, Japan, December, S. 299-306.
- Sandholm, T.W. (1999): An Algorithm for Optimal Winner Determination in Combinatorial Auctions. International Joint Conference on Artificial Intelligence (IJCAI), S. 542-547, Stockholm, Sweden.
- Sandholm, T.W., Crites, R.H. (1995): On Multiagent Q-Learning in a Semi-competitive

- Domain. IJCAI-95 Workshop on Adaptation and Learning in Multiagent Systems, S. 71-77, 1995.
- Sandholm, T.W., Lesser, V.R. (1997): Coalitions among computationally bounded agents. *Artificial Intelligence*, Vol. 94 (1997), S. 99.-137.
- Sathi, A. Fox, M. (1989): Constraint-directed Negotiation of Resource Allocations. In L. Gasser and M. Huhns, editors, *Distributed Artificial Intelligence 2*. Morgan Kaufmann, 1989.
- Schelling, T.C. (1960): *The strategy of conflict*. Cambridge: Harvard Univ. Press, 1960.
- Schiefloe, P.M., Syvertsen, T.G. (1997): Coordination in Knowledge-Intensive Organizations, in: Conen, W., Neumann, G. (Ed.): *Coordination Technology for Collaborative Applications*, S.9-24. Heidelberg: Springer, 1997
- Schmid, B., Lindemann, M.A. (1998): Elements of a Reference Model for Electronic Markets. *Proceedings of the 31st Annual Hawaii International Conference on Systems Science (HICCS'98)*, Vol. IV, pp. 193-201, Hawaii, January 6-9 1998. http://www.netacademy.org/netacademy/publications.nsf/all_pk/466
- Schoder, D., Eymann, T. (2000): Mobile Agents - the Real Challenges, in *Communications of the ACM*, June 2000, Vol. 43, No. 6, S. 111-112.
- Schoder, D., Hummel, T., Müller, G. (1997): Interdisziplinäre Modelle für Entwurf und Einsatz telematischer Systeme - Grundsätzliche Probleme diskutiert am Beispiel des "Freiburger Schichtenmodells der Telematik", in: Jarke, M./Pasedach, K./Pohl, K.: *Informatik '97: Informatik als Innovationsmotor*, Aachen, 24.-26. September 1997 (GI-Jahrestagung), S. 277-286
- Schöneburg, E., Heinzmann, F., Feddersen, S. (1994): *Genetische Algorithmen und Evolutionsstrategien*. Bonn: Addison Wesley, 1994.
- Schumacher, Michael, Chantemargue, Fabrice, Hirsbrunner, Beat (April 26-28, 1999): The STL++ Coordination Language: a Base for Implementing Distributed Multi-Agent Applications, in: *Proceedings of Third International Conference on Coordination Models and Languages*, . Amsterdam: April 26-28, 1999 <http://www-iiuf.unifr.ch/pai/axe/AmocPapers/coord99.ps>.
- Searle, J.R. (1970): *Speech Acts: An Essay in the Philosophy of Language*, Cambridge University Press, January 1970.
- Shoham, Y. (1997): An Overview of Agent-oriented Programming, in: Bradshaw, J.M. (Ed.): *Software Agents*. Menlo Park, CA: AAAI Press, 1997, S. 271-290.
- Sierra, C., Faratin, P., Jennings N.R. (1997): A service oriented negotiation model between autonomous agents. *Proc. of Modelling Autonomous Agents in a Multi Agent World (MAAMAW)*, 1997. <ftp://ftp.elec.qmw.ac.uk/pub/isag/distributed-ai/publications/SierraFaratinJennings.ps.Z>
- Sierra, C. (2000): *Agent-mediated Electronic Commerce: Scientific and Technological Roadmap*. <http://www.iiia.csic.es/AMEC/Roadmap.ps>

- Smith, A. (1784): An inquiry into the nature and causes of the wealth of nations. 1. publ., selected edition. Oxford : Oxford Univ. Press, 1993 (Nachdruck der Erstausgabe von 1784).
- Smith, R.E., Taylor, N. (1998): A Framework for Evolutionary Computation in Agent-Based Systems. In: Looney, C., Castaing, J.: Proceedings of the 1998 International Conference on Intelligent Systems, S. 221-224. <http://www.ics.uwe.ac.uk/~rsmith/fecabs.pdf>
- Smith, R.G. (1980): The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. In: IEEE Transactions on Computers, Vol. 29, 1980, S. 1104-1113.
- Smith, V.L. (1962): An experimental study of competitive market behavior. Journal of Political Economy, Vol. 70, S. 111-137, 1962.
- Smith, V.L. (1992): Papers in Experimental Economics. Cambridge: Cambridge University Press, 1992.
- Steels, L., Brooks, R. (1994): The artificial life route to artificial intelligence. Building Embodied, Situated Agents. Lawrence Erlbaum: Hillsdale, 1994.
- Stender, J. (1994): Many-Agent Simulation and Genetic Algorithms – Adaptation Mechanisms in Economic Models, in: Hillebrand, E., Stender J. (Ed.): Many-Agent Simulation and Artificial Life, Amsterdam: IOS-Press, 1994.
- Strauß, R.E. (1996): Determinanten und Dynamik lernender Organisationen - eine empirische und modellgestützte Analyse, Gabler Verlag 1996
- Sycara, K. (1989): Multi-agent Compromise via Negotiation. In L. Gasser and M. Huhns, editors, Distributed Artificial Intelligence 2. Morgan Kaufmann, 1989.
- Tesfatsion, L. (1997): How Economists can get alive. in: Arthur, W.B., and Durlauf, S., and Lane, D. (Eds.): The Economy as a Complex Evolving System, II. Sciences of Complexity, Proceedings Vol. XXVII, Santa Fe Institute Studies, Addison-Wesley, Redwood City CA, 1997. <http://www.econ.iastate.edu/tesfatsi/rgetroot.ps>
- Thomas, J., Sycara, K. (1998): Heterogeneity, Stability and Efficiency in Distributed Systems. In: Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS-98), Paris, France, July, 1998.
- Tsang, E. (1993): Foundations of Constraint Satisfaction. Academic Press, 1993.
- Tsvetovatyy, M.B., Gini, M. (1996): Toward a virtual marketplace: architectures and strategies. Proc. PAAM-96, London 1996, S. 597-614.
- Tygar, D. (1997): Atomicity in electronic commerce. In: Denning, P., Denning, D. (Eds.): Internet Beseiged. ACM Press and Addison-Wesley, 1997, pages 389 - 406.
- Turner, R.M. (1993): The tragedy of the commons and distributed AI systems. Proc. 12th International Workshop on Distributed Artificial Intelligence, S. 379-390, Hidden Valley, PA 1993.
- Varela, F.J., Thompson, E., Rosch E. (1991): The embodied mind: Cognitive science and human experience. MIT Press, Cambridge, MA, 1991.

- Varian, H.R. (1991): Grundzüge der Mikroökonomik. 2. Aufl. München: Oldenbourg Verlag, 1991.
- Varian, H.R. (1994): Mikroökonomie. 3. Aufl. München: Oldenbourg Verlag, 1994.
- Varian, H.R. (1995): Mechanism Design for Computerized Agents. Talk presented at the Usenix Workshop on Electronic Commerce, July 11-12, 1995, New York. <http://www.sims.berkeley.edu/~hal/Papers/mechanism-design.pdf>
- Vidal, J.M., Durfee, E.H. (1996): "Building Agent Models in Economic Societies of Agents." AAAI-96 Workshop on Agent Modeling, Portland, Oregon, July 1996. <ftp://ftp.eecs.umich.edu/people/durfee/aaai96ws-am-vd.ps.Z>
- Von Böventer, E.; Illing, G.: Einführung in die Mikroökonomie. 8. Aufl. München: Oldenbourg Verlag, 1995.
- Vriend, N.J. (1999): Was Hayek an Ace? Working Paper No. 403, Queen Mary and Westfield College, University of London, 1999. <http://www.qmw.ac.uk/~ugte173/abs.hayek.html>
- Vulkan, N. (1999): Economic Implications of Agent Technology and E-commerce. The Economic Journal (February 1999).
- Vulkan, N., Jennings, N.R. (1999): Efficient Mechanisms for the Supply of Services in Multi-Agent Environments. Proc of 1st Int Conf. on Information and Computation Economies, Charlestown, South Carolina, 1-10. <ftp://ftp.elec.qmw.ac.uk/pub/isag/distributed-ai/publications/ice98.ps>
- Walras, L. (1874): *Éléments d'économie politique pure: ou théorie de la richesse sociale*. Paris: Economica, 1988. Nachdruck von Lausanne: L. Corbaz, 1874.
- Wegener, A. (1998): Entwicklung einer datenbankgestützten Java-Anwendung zur Missionsvorbereitung in der Raumfahrt. Diplomarbeit, Fachhochschule Bremen, 1998. http://www.fbe.hs-bremen.de/dokumente/diplomdoku/JavaDB/Diplom/Diplom/html/5_5.htm#5_5_1
- Weiss, G. (Ed.) (1999): Multiagent Systems, MIT Press, 1999.
- Wellman, M.P. (1993): A market-oriented programming environment and its application to distributed multicommodity flow problems. *Journal of Artificial Intelligence Research*, Vol. 1., S. 1-22, 1993.
- Wellman, M.P. (1995): The economic approach to artificial intelligence. *ACM Computing Surveys*, Vol. 27., No. 3, September 1995.
- Wellman, M.P. (1996): Market-Oriented Programming: Some Early Lessons, in: Clearwater (Ed.): *Market-Based Control: A Paradigm for Distributed Resource Allocation*. World Scientific, 1996. <ftp://ftp.eecs.umich.edu/people/wellman/mbc95.ps.Z>
- Wellman, M.P., Wurman, P.R. (1997): Market-Aware Agents for a Multiagent World, in: Boman, Magnus, Van de Velde, Walter (Ed.): *Multi-Agent Rationality: Proceedings of Eight European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-97)*, Springer Notes in Artificial Intelligence. Heidelberg: Springer, 1997
- Werkman, K.J. (1990): Knowledge-based Model of Negotiation using Shareable Perspectives.

- In Proceedings of the 10th Int Workshop on DAI, 1990.
- Wildemann, H. (2000): Von Just-In-Time zu Supply Chain Management. In: Wildemann, H. (Hrsg.): Supply Chain Management, München: TCW Transfer-Centrum-Verlag, 2000, S. 49-86.
- Witt, U. (1983): Marktprozesse: neoklassische versus evolutorische Theorie der Preis- und Mengendynamik. Königstein/Ts. : Athenaeum, 1980.
- Wooldridge, M.J (1992): The Logical Modelling of Computational Multi-Agent Systems. PhD thesis, UMIST, Manchester, October 1992. <http://www.csc.liv.ac.uk/~mjw/pubs/thesis.ps.gz>
- Wooldridge, M.J. (1997): Agent-based software engineering. IEE Proceedings on Software Engineering, No. 144, S. 26-37, 1997.
- Wooldridge, M.J. (1999): Intelligent Agents. In Weiss, G. (Ed.): Multiagent Systems, MIT Press, 1999. <http://www.csc.liv.ac.uk/~mjw/pubs/mas99.ps.gz>
- Wooldridge, M.J., Jennings, N.R (1994): Towards a Theory of Cooperative Problem Solving. In Proceedings of the Sixth European Workshop on Modelling Autonomous Agents in Multi-Agent Worlds (MAAMAW-94). <http://www.csc.liv.ac.uk/~mjw/pubs/maamaw94.ps.gz>
- Wooldridge, M.J., Jennings, N.R. (1995): Intelligent Agents: Theory and Practice. In Knowledge Engineering Review, 10 (2), 1995. <http://www.csc.liv.ac.uk/~mjw/pubs/ker95.ps.gz>
- Wooldridge, M.J., Jennings, N.R. (1999): Software Engineering with Agents: Pitfalls and Praterfalls. In IEEE Internet Computing, May/June 1999, Vol. 3, No. 3, S. 20-27. <http://www.computer.org/internet/ic1999/w3020abs.htm>
- Wurman, P.R., Wellman, M.P., Walsh, W.E. (1998): The Michigan Internet AuctionBot: A configurable auction server for human and software agents. Proc. Second International Conference on Autonomous Agents, pages 301-308, May 1998.
- Yamaki, H., Wellman, M.P., Ishida, T. (1996): A market-based approach to allocating QoS for multimedia services. In Proc. ICMAS-96, S. 385-392, Kyoto, Japan 1996.
- Ygge, F. (1998): Market-Oriented Programming and its Application to Power Load Management, Ph.D. Thesis. ISBN 91-628-3055-4, CODEN LUNFD6/(NFCS-1012)/1-224/ (1998). Lund University, 1998.
- Yookoo, M., Durfee, E. (1992): Distributed Constraint Satisfaction for Formalizing Distributed Problem Solving, Proceedings of the 12th IEEE International Conference on Distributed Computing Systems, 1992.

Autorenindex

A

American Heritage	15
Anez	252
Arrow	48, 63
Axelrod	118

B

Baker	70, 71, 72
Barbuceanu	115
Bleicher	15
Blumberg	211
Böventer	30
Bradshaw	77, 93
Brenner	122
Brooks	86
Bruten	55, 87, 116
Budimir	58, 59
Burkhardt	140
Bussmann	95, 116

C

Castelfranchi	4, 206
Chavez	104
Cheng	63, 64
Chess	210
Clearwater	20
Cliff	55, 87, 116
Conry	95
Conte	4, 206
Corkill	96, 98, 105
Crites	121
Crowston	28, 32, 40
Crutchfield	79

D

Damker	211
De Paula	110, 114, 115, 154
Deutsche Börse	31, 57
Distributed Planning	98
Distributed Vehicle Monitoring	98
Dosi	127
Drexler	20, 54, 55, 56
Druckman	120
Durfee	59, 83, 84, 86, 93, 95, 96, 97, 98, 99, 100, 101, 102, 103, 113, 116, 120, 211

E

Eymann	104
--------------	-----

F

Fankhauser	21, 42, 43, 44
Fausett	121
Feldman	57, 60
Ferber	7, 85, 86, 207
Ferguson	37, 38
Finin	107
FIPA	85, 105, 107, 108, 115, 148, 151, 167, 210, 214
Fischer	104
Foner	104, 210
Foster	65
Fox	115, 116
Franklin	77
Friedman	115
Functionally-Accurate Cooperation	98

G

Gagliano	36
Galyean	211
Gasser	92, 106
Genesereth	83, 98, 106, 166, 214
Georgeff	85, 95
Gilbert	77
Gimenez	57, 205
Gini	104

Gode	55, 87
Goldberg	121, 128
Gomber	58, 59, 205
Gravelle	49
Guttman	12, 57, 112

H

Hahn	48, 63
Haken	207
Hall	120
Hardin	213
Harsanyi	117
Hayek	31, 65, 67, 68, 69
Hayer-Roth	95
Hobbes	5
Hogg	213
Holland	79, 121, 127, 138, 207, 209
Hoppmann	65, 66
Huberman	20, 213
Huhns	93, 95
Hummel	14, 15, 16, 19

I

Illing	30
--------------	----

J

Janson	22, 28, 37, 45, 46, 102
Jennings	7, 60, 78, 91, 94, 100, 116, 137, 206, 209, 210
Jeon	105

K

Karnopp	207
Kauffman	207, 209
Kearney	60, 62, 95, 206
Keeney	112
Kephart	1, 91, 147, 175, 213, 215
Ketchpel	106
Kieser	30

Kimmel	123
Klein	15, 28, 29
Klimecki	15
Kraus	6, 103, 110, 116, 119, 120, 121, 154
Kreifelt	115
Kubicek	30
Kumar	57, 60
Kurose	32
Kuwabara	72, 73

L

Labrou	107
Lane	4, 206
Laßmann	15
Lee	94
Lehmann	121
Lesser	96, 98, 213
Levinson	107
Ljunberg	104
Loasby	69
Lucas	104
Luce	42, 116

M

Maes	77, 104, 112
Malone	28, 32, 40, 96
Martial	115
Maturana	86
Ménard	30
Merlat	60, 62, 206
Miller	20, 27, 54, 55, 56, 207, 209
Minsky	56
Mitchell	128
Mitchem	36
Montgomery	97, 116
Mouaddib	115
Moulin	31, 32, 33, 34, 35, 153
Müller	13, 95, 116

N

Nash	42
Ndumu	210
Neal	121
Nelson	127
Nilsson	83
Norrie	41
Norvig	80, 81
nson	101
Nwana	77, 94, 210

O

O'Driscoll	75
Objectspace	141
Osborne	119

P

Padovan	160, 211
Partial Order Planner	98
Parunak	95, 212
Porter	136
Preist	12, 38, 110, 115, 121, 154
Pruitt	15, 39, 109, 122, 123, 124, 125, 127, 181, 182, 183, 189

R

Raiffa	42, 112, 116
Rannenberg	212
Rao	85
Rasmusson	22, 28, 37, 45, 46, 101, 102
Rees	49
Result Sharing	98
Rizzo	75
Rosenschein	21, 59, 93, 94, 96, 98, 99, 100, 101, 102, 103, 111, 112, 116, 117, 163, 211
Ross	2
Rötzer	62
Rubinstein	119
Rumelhart	121
Russel	80, 81

S

Samuelson	49, 203
Sandholm	22, 56, 59, 112, 121, 127, 159, 211, 213
Sathi	116
Schelling	39
Schoder	15
Schumacher	94
Shoham	77
Sierra	12, 110, 115, 208, 212
Singh	95
Smith	30, 95, 128, 129, 130, 142, 151, 161, 162, 209
Steels	78
Stender	209
Stephens	93
Strauss	29
Sunder	55, 87
Sycara	95, 116, 213

T

Taylor	128, 129, 142, 161, 162, 209
Tesch	21, 42, 43, 44
Tesfatsion	116, 206
Thomas	213
Tilley	97
Tsang	112
Tsvetovatyy	104
Turner	53
Tygar	21

V

Varela	86
Varian	21, 33, 38, 48, 50, 59, 63, 131, 207
Vidal	59, 83, 84, 86, 120
Vulkan	59, 60, 116, 206

W

Walras	48, 203
Weinhardt	59
Weiss	5

Wellman	20, 21, 22, 23, 39, 50, 51, 52, 53, 63, 64, 203, 206
Werkman	95
Wild	65
Wildenfield	116
Witt	31, 127
Wooldridge	77, 78, 79, 80, 83, 85, 210
Wunderli	41
Wurman	21, 50, 57, 205

Y

Yamaki	53
Ygge	53, 205
Yookoo	113

Z

Zlotkin	21, 59, 94, 101, 111, 112, 116, 117, 163
---------------	--

Stichwortindex

A

ACL 107
Agent Communication Language 151
Agent UML 140
Agent-mediated Electronic Commerce 57
Arbitrationsschema 42
Arbitrator 42
Autonomie 77, 78

B

BDI-Architekturen 85
Behavior-Based AI 86
Blackboardsysteme 105
Bottom-Up AI 86
Broadcast Architekturen 104

C

Complex Adaptive Systems 138
Concession 124
Concession rate 123
Constraint Satisfaction Problem 112
Continuous Double Auction 87

D

Deliberativer Agent 83
Demand level 124
Distributed Problem Solving 94

E

Economic Mechanism Design 27
Edgeworth-Prozeß 63
Emergenz 103
Enaction 86

Evolutionäre Algorithmen 127

F

Facilitator 106

fair-share-Prinzip 34

Federated Systems 106

G

Genetische Algorithmen 128

Goal/expectation hypothesis 123

greedy lies 43

H

Hahn-Prozeß 63

I

Indifferenzpreis 114

informed lies 43

Iterated-Contract-Net-Protocol 151

J

JATLite 105

K

k.-Ebene-Konzept 84

KIF 166

Knowledge Interchange Format 214

Kontraktnetz 97

Kontraktnetz-Protokoll 95

KQML 107

L

Limitpreis 114

M

Market-Driven Contract Net 70

Marktmodell 57

Multi-Agent Planungstechniken 95

Multi-Agent Systems 94

Multi-Attribute Utility Theory 112

Multi-Objective Decision Analysis 112

N

Nash-Gleichgewicht 112

no-envy-Test 34

Non-Tâtonnement-Prozesse 63

O

Organizational Structuring 95

P

Pareto-Optimum 112

Performative 107

Proaktivität 79

Produktattribute 110

R

Reaktive Agenten 85

Reaktivität 79

S

Sozialverhalten 79

State-Oriented Domain 37

Strategic Choice Model 123

Subsumptionsarchitektur 86

T

Task Sharing 97

Task-Oriented Domain 37

U

Unsichtbare Hand 30

V

Verhandlungsattribute 110

Verhandlungstechniken 95

Verteilte Künstliche Intelligenz 93

Verzeichnisdienst 141

W

Worth-Oriented Domain 38

Worth-oriented domain 117

Z

Zero Intelligence Agenten 87

Glossar

1. *CORBA*

CORBA (Common Object Request Broker Architecture) verfolgt das Ziel Objekte miteinander zu verbinden, die sich möglicherweise auf verschiedenen Rechnern befinden. Das Konzept ist sprachunabhängig, d.h. es erlaubt die Kommunikation von Objekten miteinander, die in unterschiedlichen Programmiersprachen geschrieben wurden. Die Schnittstellen werden in einer speziellen Interface Definition Language (IDL) beschrieben. Die Abbildung der Datentypen auf die Gegebenheiten einer konkreten Programmiersprache ist jeweils genormt (Binding oder Mapping), aber nicht immer vollkommen (die Datentypen lehnen sich an C++ an). Objekte werden zwar registriert, jedoch nicht an einem festen Rechner. Sie erhalten einen weltweit eindeutigen (hierarchisch strukturierten, vergleiche DNS) Namen, über den auch ihre Maschine auffindig gemacht werden kann. Auf jeder Maschine läuft ein Object Request Broker (ORB). Die jeweiligen ORBs kommunizieren miteinander, um Dienste bereitzustellen (z.B. das Finden von Objekten) und Daten auszutauschen. Untereinander benutzen sie das Internet Inter-ORB Protocol (IIOP) (Barthelmann 2000).

2. *JAVA*

Java wurde von der Firma Sun entwickelt und erstmals am 23. Mai 1995 als neue, objekt-orientierte, einfache und plattformunabhängige Programmiersprache vorgestellt. Sun besitzt weiterhin das Schutzrecht auf den Namen Java, die Sprache ist aber für alle Computersysteme verfügbar (im Allgemeinen kostenlos) und ist seit 1997 auf dem Weg, eine offizielle ISO-Norm zu werden. Java geht auf die Sprache Oak zurück, die 1991 von Bill Joy, James Gosling und Mike Sheridan im Green-Projekt entwickelt wurde, mit dem Ziel, eine einfache und plattformunabhängige Programmiersprache zu schaffen, mit der nicht nur normale Computer wie Unix-Workstations, PCs und Apple programmiert werden können, sondern auch die in Haushalts- oder Industriegegeräten eingebauten Micro-Computer, wie z.B. in Waschmaschinen und Videorekordern, Autos und Verkehrsampeln, Kreditkarten und Sicherheitssystemen und vor allem auch in TV-Settop-Boxes für "intelligente" Fernsehapparate. Die wichtigsten Eigenschaften von Java sind Plattformunabhängigkeit, Objektorientierung mit einer Syntax ähnlich wie C++ und einem Schwerpunkt auf der Gewährleistung technischer Sicherheit von Internet-Anwendungen (Partl 2000).

3. *RMI*

Remote Methode Invocation (RMI). Mit RMI ist es möglich, ein verteiltes System zu entwickeln, das sich auf die Adreßräume unterschiedlicher Rechner aufteilt, mit der Möglichkeit Daten über ein Netzwerk auszutauschen. Das direkte Arbeiten mit Sockets wird hier durch Remote Procedure Calls (RPC) ersetzt. RPC erlaubt das Aufrufen von Prozeduren oder das Starten von Prozessen auf entfernten Rechnern über eine Netzwerkverbindung. Aus Sicht des Entwicklers sieht die Kommunikation mit einem anderen Rechner unter Einsatz von RMI wie ein lokaler Funktionsaufruf aus. Die Parameter eines solchen Aufrufs werden über das Netzwerk zu einem anderen Rechner übertragen. Dort wird die Funktion oder Prozedur ausgeführt und das Ergebnis schließlich zum Aufrufer zurückgeschickt. Anhand dieser Mechanismen ist es möglich, ein Client / Server-System zu entwickeln, wie es zuvor mit Sockets beschrieben wurde. Das Versenden z.B. einer Nachrichten für die Anmeldung an eine Datenbank kann durch einen normalen Funktionsaufruf ausgeführt werden. Der Rückgabewert dieser Funktion kann die Bestätigung des Servers enthalten, dass die Datenbankverbindung steht. Die Verbindung zwischen einem Client- und einem Server-Objekt bildet ein Interface. In diesem Interface (bestehend aus einer Java-Klasse) müssen alle Methoden deklariert sein, die über den Aufruf eines Clients auf einem Server-Objekt ausgeführt werden sollen. Die Implementierungen dieser Methoden befinden sich in dem Server-Objekt. (Wegener 1998).

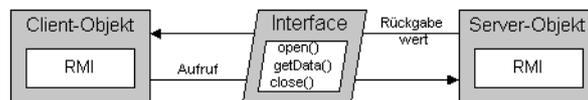


Abb. 47: Prinzipdarstellung der Remote Method Invocation (RMI)

4. *VOYAGER*

VOYAGER ist ein Object Request Broker, der sowohl CORBA als auch Sun's RMI Implementation unterstützt. Proxies für entfernte Objekte werden dynamisch erzeugt, so dass keine Stubdateien a priori generiert werden müssen. Das Produkt nutzt die Java Virtual Machine (JVM), um zur Laufzeit Klassen zu laden, die mobile Objekte und autonome Agenten darstellen können. VOYAGER selbst eignet sich besonders gut für die Implementation heterogener verteilter Systeme, da verschiedene existierende Standardisierungen einbezogen sind (Anez 1999). Das Feature der Kommunikation zwischen Agenten unterschiedlicher Objekttechnologie (z.B. CORBA, RMI und COM) macht es möglich, von der technischen Gestaltung der Kommunikation zwischen Objekten weitestgehend zu abstrahieren. Ein genereller Naming Service macht es möglich, unterschiedliche Verzeichnisdienste transparent zu nutzen. Das Versenden von

Nachrichten abstrahiert von einer lokalen oder entfernten Implementation des Empfängerobjekts. VOYAGER verwendet einen sogenannten *Namespace* als hierarchischen und verteilten Verzeichnisdienst. Referenzen auf benannte Objekte werden durch Übergeben des Hostnamen, des IP-Ports und der Objektbezeichnung an die Methode `Namespace.lookup()` ermittelt.

Da kein zentrales Programm existiert, welches die Agenten ständig kontrolliert, benötigt man zur Aufnahme der Kommunikation die eindeutige Adresse des anderen Agenten. VOYAGER verwendet einen sogenannten *Namespace* als hierarchischen Verzeichnisdienst. Referenzen auf benannte Objekte werden durch Übergeben des Hostnamen, des IP-Ports und der Objektbezeichnung an eine Methode `Namespace.lookup()` ermittelt.

```
IAVLocationAgent market =  
    (IAVLocationAgent) Namespace.lookup("//host:8000/L0");
```

Die Referenz auf dieses Objekt ist jedoch nicht von der Klasse des Objekts, sondern stellt eine sogenannte Stummelklasse (engl. *stub*) dar, d.h. eine Interfaceklasse auf dem lokalen Rechner. Ein Interface enthält keinen Code, sondern definiert eine Menge an Methodenaufrufen (method signatures), die durch die Klasse, die dieses Interface implementiert, mit Code realisiert werden müssen. Eine Interface-Variable (im obigen Beispiel `market` vom Typ `IAVLocationAgent`) kann auf jedes Objekt zeigen, dessen Klasse dieses Interface implementiert. Die Namenskonvention von VOYAGER ist es, die Interfaces mit "I" zu beginnen. `IAVLocationAgent` ist also eine Interfaceklasse von `AvLocationAgent`. Ein entferntes (remote) Objekt wird in VOYAGER durch ein spezielles lokales Proxy-Objekt repräsentiert, welches das gleiche Interface implementiert wie das entfernte Objekt. Eine Interface-Variable zeigt auf den lokalen Proxy, während die Verarbeitung durch das entfernte Objekt durchgeführt wird. Um ein Objekt auf einem bestimmten Port zu starten, wird die `Factory.create()` Methode verwendet. Diese erstellt den Proxy in dynamischer Form und gibt dessen Adresse an das aufrufende Programm zurück (s.u.). Die Zusammenhänge zwischen Proxy, lokaler Referenz und entferntem Objekt zeigt Abbildung 48.

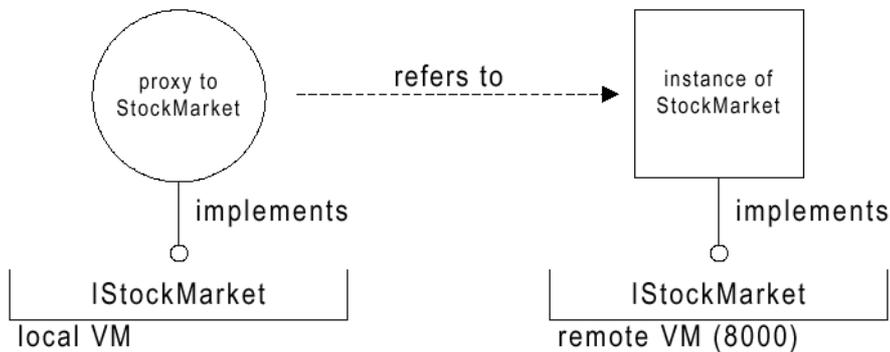


Abb. 48: Zugriff auf entfernte Objekte in VOYAGER (aus Objectspace 1998, S. 9)

Objekte müssen jedoch nicht statisch an einen Host gebunden sein, sondern können auch zwischen Rechnern bewegt werden. "Mobile Objekte" bezwecken eine Verringerung der Netzlast und Erhöhung der Verarbeitungsgeschwindigkeit, indem Objekte auf den gleichen Rechner bewegt werden und dort lokal miteinander kommunizieren. Ein Beispiel dafür zeigt folgender Code-Ausschnitt, bei dem ein Marktplatz vom Typ StockMarket von einem Server zu einem Anderen bewegt wird:

```

IStockMarket market = (IStockMarket) Namespace.lookup("NASDAQ");
Mobility.of(market).moveTo("//another_server:7000");
  
```

Eine spezielle Art von Mobilität sind "Mobile Agenten", die sich technisch von Mobilien Objekten durch ihre autonome Weiterbewegung unterscheiden – Mobile Objekte werden von einem anderen Programm bewegt, während Mobile Agenten sich selbst bewegen. In AVALANCHE sind die Agenten als Mobile Agenten implementiert, damit sie sich bei Bedarf von einem Marktplatz zum anderen bewegen können. In diesem Fall tragen sie ihre sämtlichen Methoden und Variablen mit sich, um am Zielort selbständig mit der Verarbeitung fortfahren zu können. Die notwendigen Codebestandteile sehen im Beispiel von VOYAGER folgendermaßen aus. In diesem Beispiel wird ein Objekt `broker` der Klasse `AvTradeAgent` erzeugt und als autonom markiert (so dass es nicht von der Java Garbage Collection entfernt wird, obwohl keine Verweise vom startenden Thread auf dieses Objekt existieren). Die Einstellung des ResourceLoaders dient der Festlegung desjenigen Ortes, an dem sich die Klassendateien der Implementation befinden. Dadurch kann darauf verzichtet werden, alle Klassen auf jedem Rechner lokal vorhalten zu müssen. Der ResourceLoader zeigt auf jenen Server und IP-Port, von dem die Klasse geladen wird. Der Agent wird in der letzten Zeile zu einem bekannten Server marke-

thost geschickt, auf dem er nach Ankunft mit der Verarbeitung der Methode work() beginnt.

```
IAvTradeAgent broker = Factory.create("AvTradeAgent");
IAgent agent = Agent.of(broker);
agent.setAutonomous(true);
agent.setResourceLoader(
    new URLResourceLoader("http://this-
host:8000"));
agent.moveTo("//markethost", "work");
```

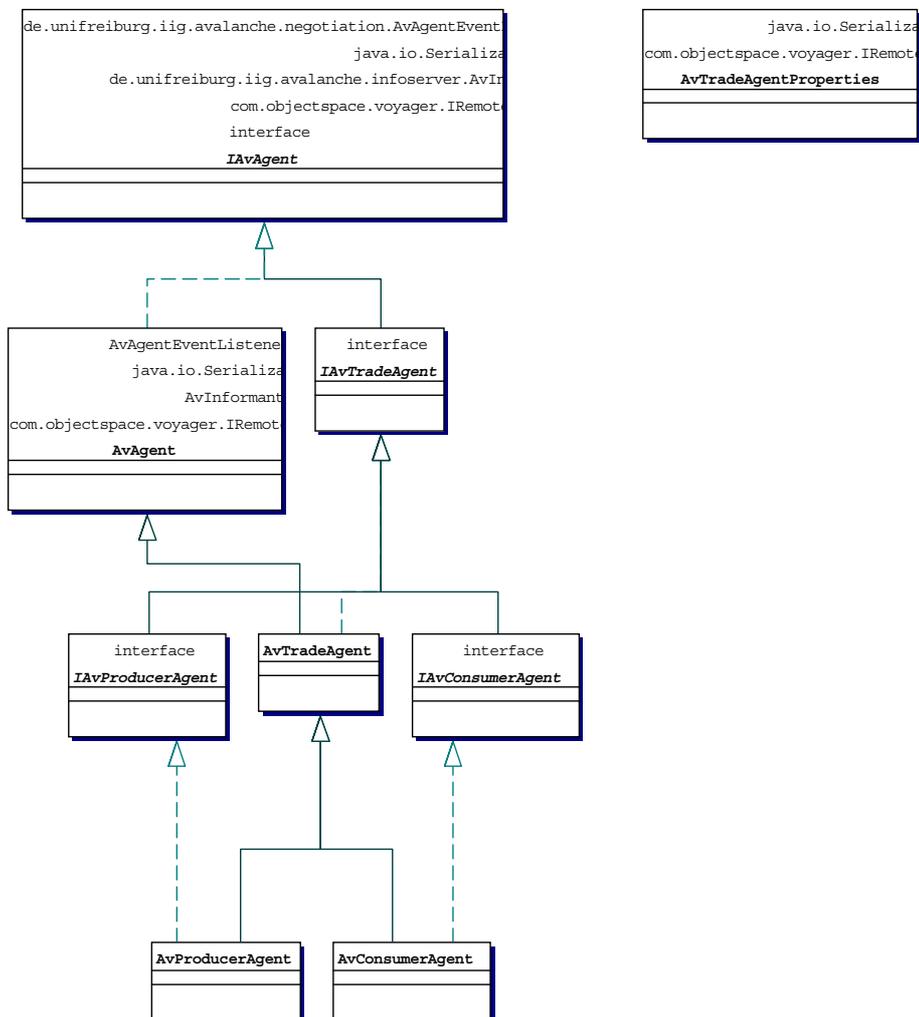



Abb. 50: UML-Klassendiagramm der handelnden Agenten

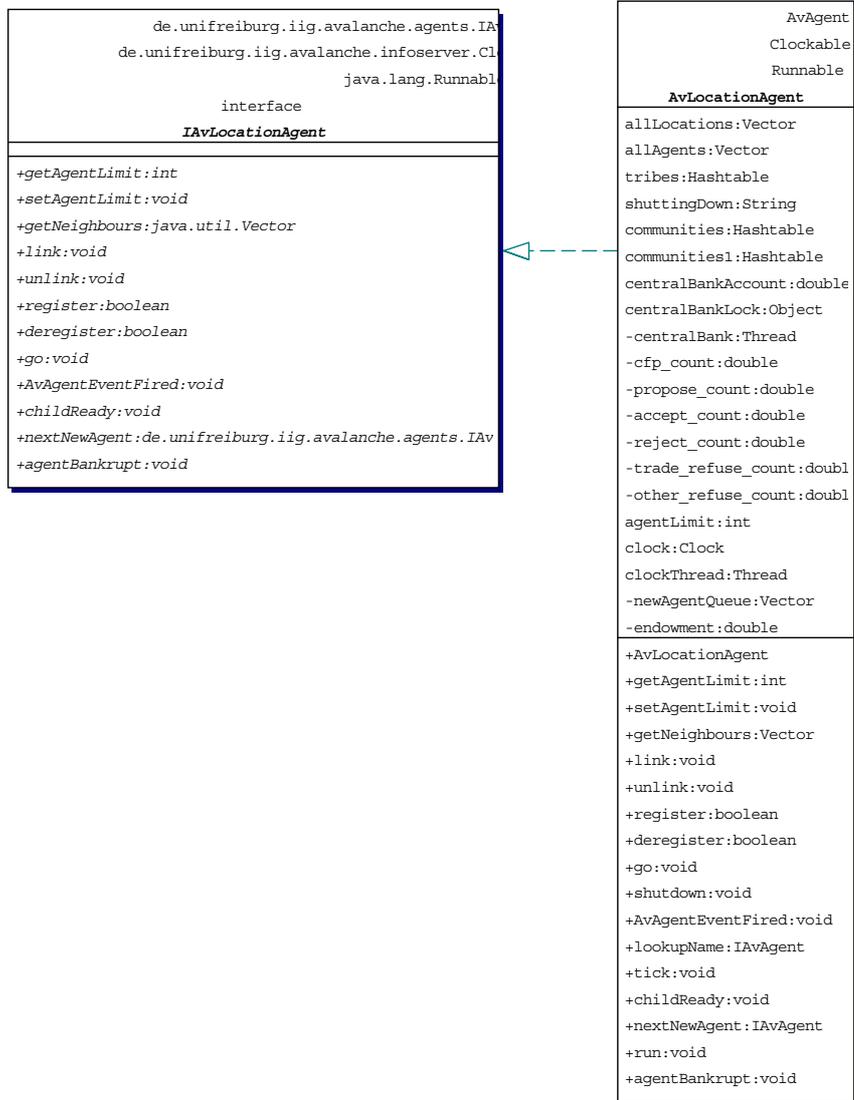


Abb. 51: UML-Klassendiagramm des Marktplatzes

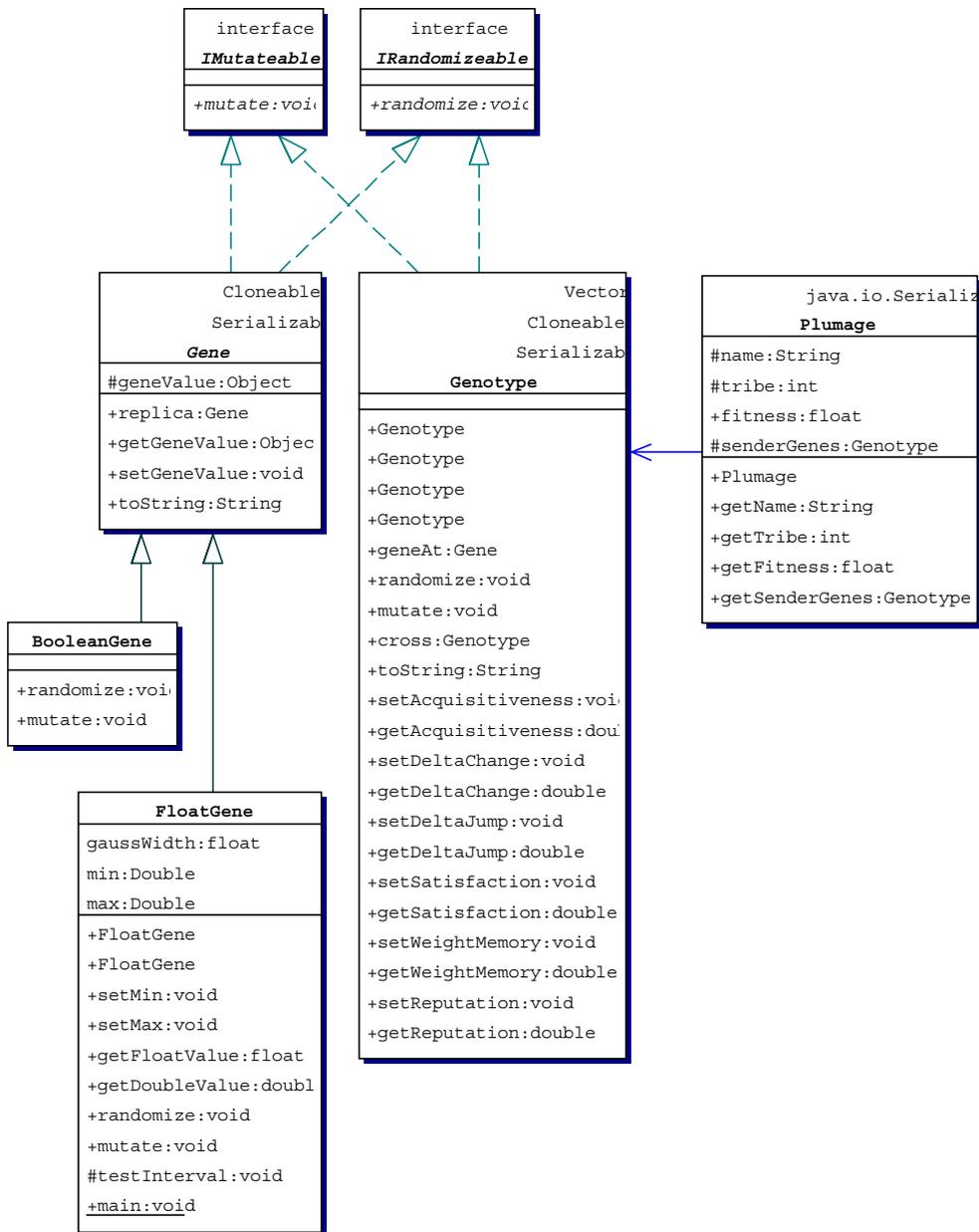


Abb. 52: UML-Klassendiagramm des evolutionären Algorithmus STDEA



Abb. 53: UML-Klassendiagramm der Inter-Agenten-Kommunikation

