

February 2007

# Regelbasierte Koordinierung von agentengestützten Transportprozessen

Mathias Petsch

*Technische Universität Ilmenau*, mathias.petsch@tu-ilmenau.de

Dirk Pawlaszczyk

*Technische Universität Ilmenau*, dirk.pawlaszczyk@tu-ilmenau.de

Hagen Schorcht

*Technische Universität Ilmenau*, hagen.schorcht@tu-ilmenau.de

Follow this and additional works at: <http://aisel.aisnet.org/wi2007>

---

## Recommended Citation

Petsch, Mathias; Pawlaszczyk, Dirk; and Schorcht, Hagen, "Regelbasierte Koordinierung von agentengestützten Transportprozessen" (2007). *Wirtschaftsinformatik Proceedings 2007*. 77.

<http://aisel.aisnet.org/wi2007/77>

This material is brought to you by the Wirtschaftsinformatik at AIS Electronic Library (AISeL). It has been accepted for inclusion in Wirtschaftsinformatik Proceedings 2007 by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

In: Oberweis, Andreas, u.a. (Hg.) 2007. *eOrganisation: Service-, Prozess-, Market-Engineering*; 8. Internationale Tagung Wirtschaftsinformatik 2007. Karlsruhe: Universitätsverlag Karlsruhe

ISBN: 978-3-86644-094-4 (Band 1)

ISBN: 978-3-86644-095-1 (Band 2)

ISBN: 978-3-86644-093-7 (set)

© Universitätsverlag Karlsruhe 2007

# Regelbasierte Koordinierung von agentengestützten Transportprozessen

Mathias Petsch<sup>1</sup>, Dirk Pawlaszczyk<sup>2</sup>, Hagen Schorcht<sup>1</sup>

Institut für Wirtschaftsinformatik  
Fachgebiet Wirtschaftsinformatik für Dienstleistung<sup>1</sup>  
Fachgebiet Wirtschaftsinformatik für Industrie betriebe<sup>2</sup>  
Technische Universität Ilmenau  
98694 Ilmenau

{mathias.petsch, dirk.pawlaszczyk, hagen.schorcht}@tu-ilmenau.de

## Abstract

Die Planung und Gestaltung effizienter unternehmensübergreifender Transportnetze rückt zunehmend in den Fokus betriebswirtschaftlicher Betrachtungen. Für eine flexible überbetriebliche Koordination und informationstechnische Integration von Güterflüssen in Lieferketten wird dabei immer häufiger der Einsatz von Softwareagenten diskutiert. Zur Lösung des Koordinationsproblems innerhalb von Transportnetzen werden Allokationsmechanismen, beispielsweise Auktionsprotokolle, vorgeschlagen. Regeln kommt an dieser Stelle eine zentrale Bedeutung zu, da sie zur Lenkung des Verhaltens und als Ausdruck der Ziele von Akteuren dienen. Im vorliegenden Beitrag wird ein Ansatz zur regelbasierten Koordination agentengestützter Transportprozesse vorgestellt. Um eine Überwachung zwischenbetrieblicher Geschäftsprozesse zu ermöglichen, ist es notwendig technische und organisatorische Rahmenbedingungen auszuhandeln und verbindlich zu kodifizieren. Als Grundlage dient eine entsprechende Regelontologie, die zur Aushandlung gemeinsam abgestimmter Geschäftsregeln bzw. Prozessbedingungen genutzt werden kann und Rechte bzw. Pflichten der Agenten abbildet.

## 1 Einführung

Die Transportdisposition im Güterverkehr, genauer die Planung der Transportkapazitäten und die Feinabstimmung von Fahrplänen und Touren unterliegt naturgemäß einer hohen Dynamik.

Erschwerend kommt hinzu, dass Transportnetze grundsätzlich offene Systeme bilden, die durch Unsicherheiten und Informationsasymmetrien gekennzeichnet sind. Bei der Disposition von Transportaufträgen muss entsprechend häufig ad hoc - aus der Situation heraus - entschieden werden. Die raumzeitliche Verteilung und weitgehende Autonomie aller Akteure in solchen Netzwerken sowie die resultierende Koordinationskomplexität und die Notwendigkeit zur Gestaltung hoch skalierbarer Prozesse über das gesamte Netzwerk hinweg, erfordern gänzlich neue, weit über derzeit verfügbare Ansätze hinausgehende IT-Lösungen.

Das globale Verhalten derartiger Logistiksysteme ergibt sich aus der Interaktion einer Vielzahl von lokal agierenden, über nicht lineare Wechselwirkungen miteinander vernetzten Individuen. Die aus derart komplexen Beziehungsgefügen resultierenden Dynamiken können nur in Einzelfällen mathematisch exakt verstanden werden. Um dennoch mögliche Ordnungszustände in diesen Systemen analysieren und emergente Phänomene abbilden und kontrollieren zu können, werden zunehmend agentenbasierte Modelle genutzt. Die Subjekte des abzubildenden Objekts sind dabei durch Agenten, eine Form autonom agierender Softwareentitäten, innerhalb des Modells repräsentiert. Derartige Modelle finden insbesondere im Rahmen der Konzeption, Planung sowie Steuerung logistischer Systeme Anwendung ([GrGr05; DaHe04]). Mit dem Einsatz von Softwareagenten wird überdies eine höhere Flexibilität angestrebt, beispielsweise im Zusammenhang mit einer verbesserten Reaktionsfähigkeit auf Störeinflüsse innerhalb des Supply Chain Managements ([DaHe04; BBGL<sup>+</sup>04]).

Die Entitäten bzw. Akteure der softwaretechnisch abzubildenden Domäne werden durch technische und nichttechnische Agenten repräsentiert. So werden neben möglichen Entscheidungsträgern der beteiligten Organisationen (Auftraggeber, Disponenten, Empfänger) zumeist Transportmittel (LKW, Bahn), Verlade- und Umschlagplätze, mitunter selbst die zu befördernden Stückgüter durch Softwareagenten abgebildet.

Um die eigentliche Dynamik des Systems nachzubilden zu können, beispielsweise die gemeinsame Durchführung von Transportaufträgen, müssen Agenten in Form von Agentengruppen, auch als Multiagentensystem (MAS) bezeichnet, zusammenarbeiten. Diese Systeme werden für die Durchführung einer gemeinsamen Aufgabe gebildet und sind durch den stetigen Ein- und Austritt von Agenten in oder aus dem Verbund gekennzeichnet. Trifft beispielsweise ein Transportagent an einem Umschlagplatz ein, so muss er spätestens bei seiner Ankunft die Modalitäten bzgl. Entladung oder möglicher Anschlusstouren mit einem entsprechenden Disponenten klären. Bis zu seiner Weiterfahrt ist der Transportagent, genauso wie alle anderen

ankommenden Transportagenten somit Teil des Multiagentensystems, welches durch den Umschlagplatz repräsentiert wird.

Die überwiegende Mehrzahl bestehender Beiträge zum Thema agentenbasierter Planung und Steuerung von Transportnetzen fokussiert auf Koordinationsaspekte, die sich durch die Zusammenarbeit der unterschiedlichen Agenten ergeben, da diese das Systemverhalten und damit die Effizienz des Logistiksystems maßgeblich determinieren. Im Allgemeinen wird dies mit dem Einsatz entsprechender Allokationsmechanismen gleichgesetzt. Außer dem eigentlichen Verlade- und Transporttermin und der Zuweisung einer Route müssen die beteiligten Parteien gleichsam über verschiedene Rahmenbedingungen (Constraints) übereinkommen, deren Einhaltung überprüfbar sein muss. Dem Aspekt der Festschreibung dieser Constraints wird bisher wenig Beachtung geschenkt. Alle Mitglieder eines Multiagentensystems sind bestimmten Rechten und Pflichten unterworfen, die beispielsweise in der Ausführung von Aufgaben oder einer Beschränkung von Ressourcen bestehen können.

Regeln dienen hierbei zur Lenkung des Verhaltens und als Ausdruck der Ziele der Akteure. Sie haben auch die Funktion, andere Agenten auf ein gemeinsames Ziel auszurichten sowie Pflichten und Rechte zu vermitteln. Eventuelle Verletzungen der Regelmenge können systemintern durch Sicherheitsmechanismen, wie z.B. Reputationsmechanismen, überwacht und auftretende Verstöße durch Sanktionen geahndet werden. Regeln können somit das Verhalten des Agenten determinieren und koordinieren [TaWa02]. Durch das Festschreiben von verbindlichen Regeln können sich Agenten in offenen Systemen auf den verschiedensten Plattformen frei bewegen und mit „fremden“ Agenten kooperieren. Grundvoraussetzung hierfür bildet die Möglichkeit, Regeln automatisch zu prüfen, was eine geeignete maschinenverständliche Formalisierung voraussetzt.

In diesem Beitrag wird ein Konzept für die Aushandlung von Regeln betrachtet (Regelabgleich). Als Basis dient eine entsprechende Regelontologie, die als Grundlage für die Vereinbarung von Regeln fungiert. Nur wenn die Kooperationspartner diese kennen, verstehen und akzeptieren, ist eine erfolgreiche Zusammenarbeit möglich. Es werden verschiedene Regelarten unterschieden und mittels Object Constraint Language (OCL) formalisiert, mit dem Ziel, eine Überwachung agentenbasierter Geschäftsprozesse zu ermöglichen.

## 2 Grundlagen

### 2.1 Agentenbegriff

Agenten können als Softwareartefakte verstanden werden, die fähig sind, ihr Umfeld wahrzunehmen und zu verändern [RuNo03]. Sie können mit anderen Agenten Informationen austauschen, wobei ihr Lebenszyklus nicht direkt von dem eines anderen Agenten abhängt: „Agents are situated in an environment, act autonomously, and are able to sense and to react to changes“ [KnTi99]. Neben Ihrer Autonomie sind Softwareagenten häufig durch soziale Eigenschaften charakterisiert, was sich u.a. in der Fähigkeit widerspiegelt, benachbarte Systeme zu erkennen und mit diesen zu kommunizieren [WoJe95]. Multiagentensysteme bestehen wiederum aus Agenten, die zum Zweck der Lösung einer gestellten Aufgabe mit anderen im System vorhandenen Agenten kooperieren. Durch den Einsatz der Agententechnologie kann eine Verbesserung der betrieblichen Flexibilität und Reduzierung von softwarespezifischen Kosten realisiert werden (vgl. [CoJe96; Buss98; Paru00; Anth03]). Komplexe realwirtschaftliche Problemstellungen werden somit effizient lösbar. Der Einsatz der Agententechnologie erscheint insbesondere dann sinnvoll, wenn (1) der betrachtete Anwendungsbereich durch eine natürliche verteilte Struktur charakterisiert ist, die nicht erst künstlich aus einer zentralen Struktur erzeugt werden muss, (2) eine dynamische Umwelt vorliegt, in der strukturelle Änderungen berücksichtigt und flexibel auf Änderungen reagiert werden muss und (3) die abzubildenden Interaktionsbeziehungen derart umfangreich sind, dass eine komplexe Koordination zwischen den einzelnen Entitäten unverzichtbar ist [Müll97].

Eine inhärente Verteilung und dezentrale Steuerung über unterschiedliche wirtschaftlich oft selbständig handelnde autonome Organisationen hinweg prägen den Anwendungsbereich der Transportnetze ebenso wie ein hoher Interaktionsbedarf, der durch die Vielzahl an unterschiedlichen Randbedingungen und dem damit verbundenen hohen Kommunikations- bzw. Koordinationsaufwand bedingt ist. Agenten bieten hier eine geeignete Metapher, um Handlungsträger und deren Interaktionsbeziehungen innerhalb eines solchen Modells abzubilden. Im Vergleich zu konventionellen Ansätzen des Operations Researchs ist die Agententechnologie stärker an den praktischen Anforderungen der Transportdisposition ausgerichtet und bietet ein adäquates Modell für die Abbildung der Transportdomäne.

## 2.2 Regeln

In der Informatik wird häufig der Policy-Ansatz für die Formulierung von Rahmenbedingungen und sicherheitsrelevanten Einschränkungen verwendet (z.B. Java). Das Java Policy-Konzept zeichnet sich dadurch aus, dass im Gegensatz zu den meisten Sicherheitsmechanismen von Betriebssystemen, bei denen der Zugriff auf Ressourcen und Daten offen ist und durch den Nutzer beschränkt werden muss, Zugriffsbeschränkungen bestehen und explizit freigegeben werden müssen [FrMu96]. Dabei sind Policies (insbesondere Java Policies) darauf beschränkt, Ressourcenzugriffe zu limitieren, d.h. Freigaben auf Daten, externe Programme oder Ressourcen einzuschränken bzw. zu erlauben. Der vorgeschlagene Ansatz der Verwendung von Geschäftsregeln beschränkt sich jedoch nicht auf die alleinige Freigabe von Ressourcen. Es ist vielmehr möglich zusätzlich organisatorische Anforderungen bzw. Regelungen, wie Zugehörigkeiten, institutionelle Beziehungen, Aufgabenanforderungen oder Kommunikations- und Interaktionsprotokolle einzuschließen. Folglich werden nicht nur reine Zugriffsbeschränkungen definiert, sondern es können auch organisatorische und institutionelle Regelungen spezifiziert werden, die über das Policy-Konzept hinausgehen.

In der Organisationstheorie wird die Gesamtheit aller formalen Regeln zur Arbeitsteilung und Koordination als formale Organisationsstruktur bezeichnet [KiKu92, 18]. Diese Regeln dienen neben dem Aspekt der Leistungs- auch der Herrschaftssicherung, d.h. die Mitglieder müssen im Rahmen der vertraglich geregelten Mitgliedschaftsbedingungen auch vorgegebene "Verfahrensvorschriften und Regeln" [KiKu92, 18] akzeptieren, die letztendlich die Hierarchie bestimmen. In Multiagentensystemen kann es notwendig sein, dass Agenten während der Laufzeit das MAS wechseln, d.h. ein MAS verlassen, um sich aus Gründen der verfügbaren Ressourcen bzw. Problemlösungsmöglichkeiten einem anderen Multiagentensystem anzuschließen. An o.g. Szenarios wird deutlich, dass der Transportagent jeweils Mitglied des MAS Umschlagplatz wird, die sowohl über eigene technologische (beispielsweise verwendete Kommunikations- und Interaktionsprotokolle), als auch organisatorische Regeln (beispielsweise verantwortlicher Disponent ist folgender Agent, Touren werden nach einem gewissen Modus vergeben, bestimmte Umschlagplätze sind nur für gewisse Frachten zugelassen usw.) verfügen können. Folglich sind MAS in unterschiedliche organisatorische Kontexte eingebunden, bzw. unterstützen trotz aller spezifizierten und implementierten Standards unterschiedliche Technologien, Protokolle etc. Dabei kann es notwendig werden, dass Agenten sich den in dem entsprechenden MAS vorherrschenden Regeln (zur Verwendung bestimmter Protokolle, organisatorischer

Regelungen o.ä.) unterwerfen müssen. In der Folge bedeutet das, dass Agenten ebenso gegebene Verfahrensvorschriften und Regeln innerhalb eines Multiagentensystems akzeptieren und befolgen müssen, d.h. die Beziehungen des Agenten zu dieser Organisation (MAS) sind vertraglich zu regeln, ohne dass jedoch die Autonomie (Individualität) des Individuums Agent nachhaltig negativ beeinflusst wird. Die Gesellschaft kennt zu diesem Zweck unter verschiedene Vertragsformen, die den Grad der Mitgliedschaft mit den jeweiligen Rechten und Pflichten regeln (z.B. Arbeitsvertrag; Werkvertrag, etc.).

Organisatorische Regeln müssen jedoch nicht explizit vorgegeben werden, sondern können sich dynamisch durch Koordination der Organisationsmitglieder untereinander bilden bzw. Ergebnis eines kollektiven Lernprozesses sein [KiKu92, 21]. Laut Scherer [Sche99, 1] werden Organisationen als System von expliziten und impliziten Regeln beschrieben, die einem bestimmten Zweck dienen und Erwartungen kommunizieren, sich in einer bestimmten Art und Weise zu verhalten. Diese Regeln können die Individuen normalerweise nicht beeinflussen und unterwerfen oder entziehen sich ihnen. Dabei müssen die Regeln interpretiert und in verschiedene Handlungen umgesetzt werden [Kies99, 303]. Um eine einheitliche Übereinstimmung der Interpretation zu erhalten, kommunizieren die Organisationsmitglieder untereinander.

Im Softwareengineering werden seit einigen Jahren Geschäftsregeln (auch Business Rules) zur Beschreibung von Bedingungen und Restriktionen von Geschäftsprozessen verwendet. Hierbei liegt der Fokus vor allem auf der Darstellung von Strukturen, deren Abhängigkeiten sowie Möglichkeiten der Steuerung und Kontrolle von Geschäftsprozessen im Rahmen der Gestaltung von Informationssystemen. Geschäftsregeln können als Beschreibung von Vorgaben und Restriktionen von Geschäftsprozessen sowie von deren Zuständen und Abläufen verstanden werden [HeKn94, 1]. Gemäß der Definition der Business Rules Group zu Geschäftsregeln [BrGr05], können zwei Perspektiven eingenommen werden (siehe Tab. 1):

	Definition
Business perspective	“...a business rule is a directive, intended to influence or guide business behavior, in support of business policy that has been formulated in response to an opportunity, threat, strength, or weakness.”
I/S perspective	“...a business rule is a statement that defines or constrains some aspect of the business. It is intended to assert business structure, or to control or influence the behavior of the business.”

Tab1: Business Rules [BrGr2005]

Durch diesen Ansatz konnte erstmalig die lange diskutierte Zusammenführung von Geschäftslogik, Präsentation und Speicherung von Daten realisiert werden. Hierbei werden



Geschäftsregeln jedoch weniger als eine Erweiterung der Modellierungsmöglichkeiten verstanden. Sie dienen vielmehr der Beschreibung, Formalisierung und Spezifizierung der Regeln von Agenten und des Multiagentensystems selbst.

Sie werden folglich ähnlich dem Java Policy-Konzept verwendet, wenn auch die Business Rules in ihren Restriktionen als auch Freiheitsgraden über Policies hinausgehen. Im Zusammenhang mit der Modellierung von Agenten, basierend auf Business Rules, entwickelten Taveter und Wagner die Definition „Business Rules are statements that express (certain parts of) a business policy, such as defining business terms, defining deontic assignments (of powers, rights and duties), and defining or constraining the operations of an enterprise, in a declarative manner“ [TaWa02, 528]. Dabei weisen Regeln nach Herbst und Knolmayer [HeKn94, 2], angelehnt an die Datenbanktheorie, folgende Struktur auf:

- Ereignis: Wann soll eine Regel überprüft werden?
- Bedingung: Was soll überprüft werden?
- Aktion: Wie soll reagiert werden?

Entsprechend dieser Struktur werden die Regeln operationalisiert und in die Software (Agenten, MAS) implementiert. Es ist folglich eine Regelsprache notwendig, die solche Konstrukte unterstützt, die beschreiben lässt, wann eine solche Regel ausgeführt bzw. betrachtet werden muss (als Reaktion auf eine Aktion bzw. innerhalb einer vorgegebenen temporären Struktur), welche Bedingungen erfüllt sein müssen damit eine entsprechende Regel angewandt werden kann und wie letztlich auf die so entstandene Situation durch eine Aktion reagiert wird.

### **3 Regelbasierter Ansatz**

Im Folgenden wird ein Ansatz beschrieben, der Werkzeuge zur Beschreibung von Regeln zur Verfügung stellt und in einer Verhandlung gegebenenfalls die Regeln zwischen den beteiligten Entitäten anzupassen. Im Ergebnis der Verhandlung wird die Mitgliedschaft eines Agenten zu einem MAS über Regeln in einer Art Vertrag fixiert.

#### **3.1 Regelontologie**

Notwendige Voraussetzung für die eindeutige Beschreibung dynamischen Regelwissens sind Ontologien. Eine Ontologie ist „...an explicit specification of a conceptualization“ [Grub93, 1].

Durch die Konzeptualisierung wird eine vereinfachte Sichtweise auf die Welt geschaffen, wobei mit Hilfe eines definierten Vokabulars eine Domäne oder ein Thema spezifiziert wird. Eine Ontologie ist im Kontext der Verteilten Künstlichen Intelligenz folglich eine Wissensbasis, die durch eine Menge von Begriffen definiert ist, indem Entitäten der Diskurswelt (Klassen, Objekte etc.) mit durch Menschen lesbarem Text bzw. Erklärungen verbunden werden. Gemäß Hage und Verheij [HaVe99] besteht Regelwissen aus Konstrukten der universellen Ontologie, die als Tatbestände bezeichnet werden. Eine Regel verbindet demnach zumindest zwei Sachverhalte: die Vorbedingung und das Ereignis. Ontologien bilden ein entscheidendes Konstrukt bei der Modellierung von Regeln, um die Interpretation von Regelwissen der beteiligten Agenten und Systeme zu ermöglichen.

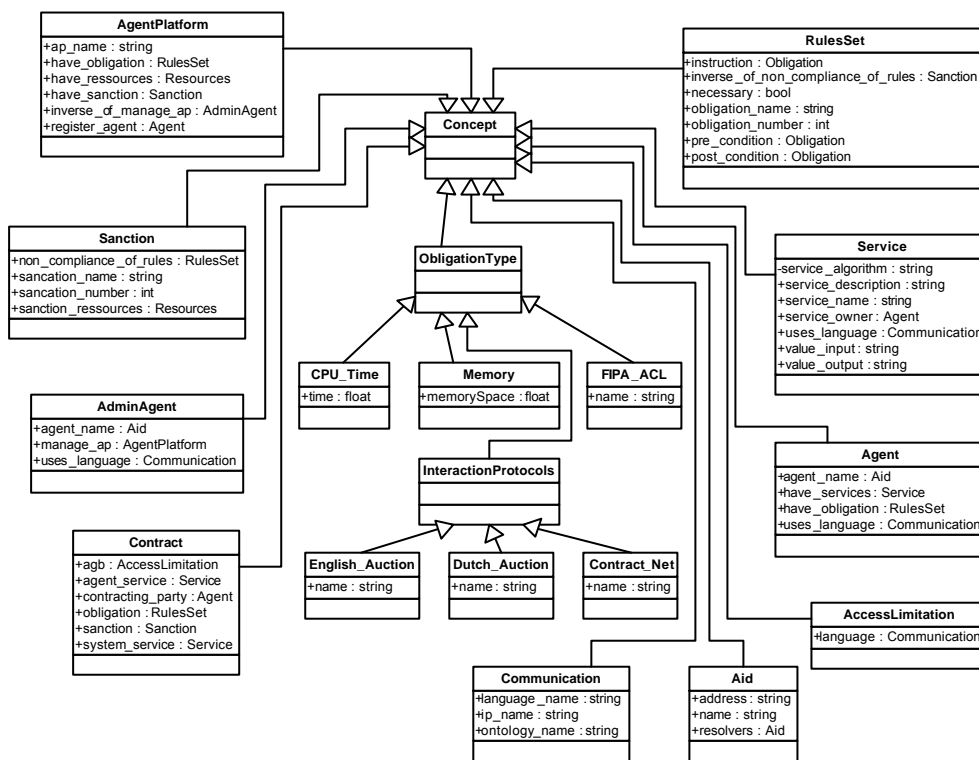


Abb. 1: Konzepte der Regelontologie

Um die semantische Korrektheit der regelbasierten Nachrichten zwischen den Agenten zu gewährleisten, wird eine Regelontologie eingeführt, die die wesentlichen Entitäten, Attribute und Aktionen zur Kommunikation und Interaktion über die Regelmenge definiert. Innerhalb der Ontologie wird zwischen Konzepten und Prädikaten unterschieden. Konzepte stellen dabei

Objekte dar, die die entsprechenden Daten der Kommunikation der Agenten beinhalten. In der praktischen Umsetzung (siehe Abb. 1) sind somit u.a. die Konzepte Contract (Vertrag zwischen dem Agenten und dem MAS), RulesSet (Menge der zu vereinbarenden Regeln), Obligation (die entsprechende Regel selbst) oder Sanction (Sanktion, welche bei einem Verstoß gegen eine Regel droht) definiert. Prädikate hingegen können als Methoden oder Funktionen betrachtet werden, die bestimmte Tätigkeiten eines Agenten beschreiben. So wurden u.a. die Prädikate searchForService (Suche nach einem Dienst, den ein potentiell zukünftiges Mitglied des MAS benötigt), askForRulesSet (Anfrage eines Agenten nach den Regeln eines Agenten), registerContract (Abschließen eines Vertrages zwischen Agent und System) oder leaveSystem (ein Agent beabsichtigt ein System zu verlassen) definiert.

Durch die oben beschriebenen Konzepte und Prädikate der Ontologie ist es nunmehr möglich, eine semantisch eindeutige Kommunikation zwischen Plattform und Agent über die Verhandlung einer Mitgliedschaft in einem Multiagentensystem zu gewährleisten. Die Ontologie ist generisch ausgelegt, beinhaltet folglich keine domänenspezifischen Elemente, die jedoch bei Bedarf noch hinzugefügt werden können. Die Ontologie bezieht sich allein auf die Regelverhandlung sowie Kontrolle der Einhaltung der Regeln und ist wenig oder nicht geeignet, anderweitige Problemklassen zu adressieren.

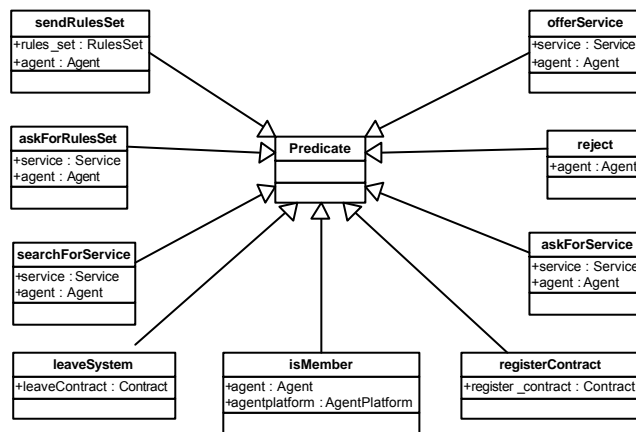


Abb. 2: Prädikate der Regelontologie

### 3.2 Regelmodell

Um eindeutige Regeln für Agenten und Multiagentensysteme modellieren und operationalisieren zu können, werden diese innerhalb eines Regelmodells abgebildet und definiert. Die Identifizierung und Modellierung von Regeln erfolgt mit Hilfe der Regelontologie, um die Eindeutigkeit der Regelkonstrukte zu gewährleisten. Die Formalisierung der Regeln erfolgt

durch die Object Constraint Language. OCL wurde durch die Object Management Group (OMG) zertifiziert und liegt derzeit in der Version 2.0 vor [OCL04]. Sie stellt eine Erweiterung der Modellierungssprache Unified Modeling Language (UML) dar. Notwendig wurde die Spracherweiterung, da herkömmliche Diagramme in UML keine Abhängigkeiten und Regeln abbilden können. OCL stellt eine Sprache dar, die neben der einfachen Verständlichkeit sehr eng an implementierungsnahen Methoden und Sprachen angesiedelt ist. Entsprechend codierte Regeln und Ausdrücke können daher ohne großen Anpassungsaufwand in objektorientierte Sprachen überführt werden. OCL stellt durch ihre bei der OMG erfolgte Spezifizierung eine verifizierte und in der Praxis erprobte Sprache dar.

Bei der Definition von Regeln kann zwischen domänenabhängigen Regeln, die in Abhängigkeit zur jeweils gegebenen Aufgabenstellung des Systems bzw. der Umwelt stehen und generischen Regeln, die keine Domänenspezifik besitzen und unabhängig von konkreten Aufgabenstellungen sind, unterschieden werden. Generische Regeln werden im Folgenden als statische Regeln bezeichnet, z.B.:

- Kommunikationsprotokolle im System (z.B. KQML, FIPA ACL);
- Koordinationsprotokolle (z.B. FIPA Dutch Auction, FIPA Contract Net);
- Inhaltsbeschreibungssprachen (z.B. Semantic Language (SL), KIF);
- Ontologien.

Domänenabhängige Regeln, die im Folgenden als dynamische Regeln bezeichnet werden, sind in Abhängigkeit von den gegebenen Aufgabenstellungen gestaltet, z.B.:

- Vertraulichkeit von Informationen;
- spezifische gesetzliche Bestimmung hinsichtlich Datenhaltung und Datenschutz;
- zu erbringender Anteil der Lösungskapazität eines Agenten in einem MAS und
- Anteil zu erbringender Kooperationsleistungen eines Agenten in einem System.

Weiterhin wird unterschieden zwischen Regeln, die unabdingbar (obligatorisch) und solchen, die optional für die Agenten und das Multiagentensystem sind (vgl. Tab. 2).

	Statisch	Dynamisch
Obligatorisch	Kommunikationssprache	Rollen, Ontologien
Optional	Interaktionsprotokolle	Zugriffsrechte, org. Regelungen

Tab. 2: Klassifizierung der Regeln von Multiagentensystemen

### 3.2.1 *Obligatorische statische Regeln*

Obligatorische statische Regeln definieren domänenunabhängige und zwingend notwendige Anforderungen, die das Funktionieren des Systems unmittelbar beeinflussen. Dazu gehören Anforderungen, die erfüllt sein müssen, um Kommunikation und Koordination in einem System überhaupt gewährleisten zu können. Diese werden für den Agenten und das MAS selbst festgelegt. Die Grundvoraussetzungen für die Interoperabilität von Agentensystemen sind innerhalb des FIPA-Standards definiert [FIPA06]. Dieser umfasst neben der Agentenkommunikationssprache ACL auch die Inhaltsbeschreibungssprachen (z.B. KIF und SLO). Obligatorische statische Regeln für Agenten im Kontext o.g. Szenarios können sein:

```
context agent inv:
  RulesSet.CommunicationLanguage = FIPA_ACL
  RulesSet.CommunicationLanguage.necessary = true
  RulesSet.ContentLanguage = FIPA_SLO
  RulesSet.ContentLanguage.necessary = true
```

Obligatorische statische Regeln für ein Multiagentensystem können sein:

```
context MAS inv:
  RulesSet.CommunicationLanguage = FIPA_ACL
  RulesSet.CommunicationLanguage.necessary = true
  RulesSet.ContentLanguage = FIPA_KIF
  RulesSet.ContentLanguage.necessary = true
```

Diese jeweils internen Regeln des Agenten sowie des MAS müssen hinsichtlich ihrer Vereinbarkeit überprüft werden. Bei obligatorischen Regeln ist eine Verhandlung über den Regelinhalt nicht möglich, d.h. eine Übereinstimmung ist zwingend notwendig, um die Grundvoraussetzungen für das Funktionieren des Systems zu schaffen.

Eine Überprüfung (oder in OCL die eigentliche Regel) würde wie folgt definiert werden:

```
MAS:: joinAgent(SystemRules: RulesSet, AgentRules: RulesSet) : boolean
  pre: SystemRules=AgentRules
  post: join=true
```

Dies würde einer trivialen Überprüfung bzw. einem Matching der Regeln entsprechen. Nur wenn die Regeln übereinstimmen, kann die Mitgliedschaft des Agenten bestätigt werden.

### 3.2.2 *Optionale statische Regeln*

Innerhalb optionaler statischer Regeln können vor allem Interaktionsprotokolle, wie das FIPA Kontrakt-Netz-Protokoll oder das FIPA Request-Protokoll definiert werden [FIPA06]. Interaktionsprotokolle sind zwar für die Kommunikation der Agenten nicht zwingend notwendig, sie sind aber unerlässlich um Koordination respektive Kooperation zwischen den Agenten zu gewährleisten. Gemäß dem eingeführten Szenario könnten optionale statische Regeln die unterschiedlichen Vergabekriterien von Transportaufträgen beinhalten, die beispielsweise über

Auktionen, oder aber über das FIPA Kontrakt-Netz Protokoll<sup>1</sup> durchgeführt werden können. Beim Abgleichen der Regelmenge des Agenten mit der des Multiagentensystems müsste das Eintrittskriterium so gestaltet werden, dass zumindest eines der durch den Agenten beherrschten Interaktionsprotokolle auch im System verfügbar ist und sich somit ein Transportagent überhaupt an der Vergabe von Transportaufträgen beteiligen kann. Eine OCL-konforme Formalisierung könnte wie folgt aussehen:

```
context agent inv:
  RulesSet.InteractionProtocol = FIPA_DUTCH_AUCTION;
  RulesSet.InteractionProtocol.necessary = false
RulesSet.InteractionProtocol = FIPA_ENGLISH_AUCTION;
  RulesSet.InteractionProtocol.necessary = false
RulesSet.InteractionProtocol = FIPA_CONTRACT_NET;
  RulesSet.InteractionProtocol.necessary = false
```

### 3.2.3 Obligatorische dynamische Regeln

Entsprechend dem Anwendungsszenario ist es erforderlich, domänenspezifische Regeln zu formulieren. So kann es geboten sein, spezielle Rollen innerhalb eines Multiagentensystems zu entwerfen (z.B. Agenten erfüllen spezielle Administrationsfunktionen oder fungieren als spezifische Schnittstellen zu existierenden IT-Systemen) oder aber Ontologien zu erstellen, die eine domänenspezifische Kommunikation unterstützen. Um ein inhaltliches Verständnis für die Kommunikation zu realisieren, kann als obligatorische Regel das Verwenden spezifischer Ontologien vorausgesetzt werden. Neben der vorgestellten Regelontologie, die generisch und somit domänenunabhängig verwendet werden kann, sind domänenspezifische Ontologien erforderlich, die die Begrifflichkeiten und Objekte der entsprechenden Domäne, im Beispiel eine Ontologie für logistische Sachverhalte (z.B. Agent.Enterprise Ontology<sup>2</sup>) definieren. Eine Formulierung der Regeln in OCL würde folgendermaßen gestaltet sein:

```
context agent inv:
  RulesSet.Ontology = RulesOntology;
  RulesSet.Ontology.necessary = true
RulesSet.Ontology = Agent.EnterpriseOntology;
  RulesSet.Ontology.necessary = true
```

Analog zu den obligatorischen statischen Regeln ist auch hier bei der Überprüfung der Regelmenge ein reines Matching, also Übereinstimmung der Regeln, erforderlich. Es wird also nur überprüft, ob die Regeln eines Agenten dem des Systems entsprechen bzw. ob der Agent bereit ist, falls er selbst nicht über die jeweiligen Regeln verfügt, die des Systems anzuerkennen.

---

<sup>1</sup> Innerhalb des FIPA Kontrakt-Netz Protokolls wird das Verfahren zur freihändigen Vergabe einer Aufgabe entsprechend der Fähigkeiten der verfügbaren Agenten spezifiziert.

<sup>2</sup> <http://www.agententerprise.net/>

### 3.2.4 Optionale dynamische Regeln

Optionale dynamische Regeln werden vorwiegend zur Formulierung organisatorischer Aspekte bzw. domänenspezifischer Anforderungen verwendet (z.B. Zugriffsrechte auf lokale Ressourcen wie Datenbanken, Speicherbereiche oder Dienste). Darüber hinaus können jedoch auch Regeln spezifiziert werden, die sich aus dem Anwendungskontext bzw. den organisatorischen Rahmenbedingungen ergeben können. Im Rahmen des Logistikszenarios ist es denkbar, dass die einzelnen Umschlagplätze bestehende Rahmenbedingungen definiert haben, die zum Beispiel den Zugriff auf lokale Ressourcen, wie Datenbanken, oder aber bei globalen Transportnetzen die jeweiligen gesetzlichen Bedingungen der entsprechenden Länder (beispielsweise Umgang mit Gefahrguttransporten, Ruhezeiten von Fahrern etc.) spezifizieren. Die Formulierung von eindeutigen, durch das System interpretierbaren Regeln erweist sich als verhältnismäßig komplex und lässt sich nur durch klar interpretierbare Ontologien erreichen. So könnte zum Beispiel eine Formulierung der Regeln in OCL wie folgt formuliert werden:

```
context MAS inv:
    RulesSet.Database.Login = "User";
    RulesSet.Database.Login.necessary = false
    RulesSet.Database.Password = "Guest";
    RulesSet.Database.Password.necessary = false
    RulesSet.DriverBreakAfter = "8h";
    RulesSet.necessary = true
```

Bei der Überprüfung wird nicht versucht, eine Übereinstimmung (Matching) der Regeln zwischen Agent und MAS zu erreichen, sondern die se müssen durch die jeweilig andere Entität akzeptiert und befolgt werden.

### 3.3 Regelabgleich/Mitgliedschaft

Der Anreiz für einen Agenten, Mitglied eines MAS zu werden, liegt entweder in den durch das System zur Verfügung gestellten Ressourcen, bzw. in Diensten, die durch Agenten des Systems bereitgestellt werden. Um Mitglied zu werden, muss der Agent mit dem System Regeln verhandeln, um die Interoperabilität und Koordinationsfähigkeit des Systems zu erhalten. Aus diesem Grund verfügen sowohl der Agent als auch das MAS über eine eigene Regelmenge, die neben technischen Spezifikationen (wie Protokolle und Sprachen) auch soziale und organisatorische Komponenten (Anforderungen an einen Agenten von Seiten seines Benutzers bzw. Anforderungen an das System in seinem sozialen Umfeld) beinhalten können.

Um einen Regelabgleich bzw. eine Verhandlung der Regelmenge durchführen zu können wird die Rete-Maschine [Forg82] von Jess (Java Expert System Shell [Jess06]) verwendet. Die

Regelmaschine von Jess besitzt eine Java-Schnittstelle, sodass Daten direkt aus dem Programm in Jess geschrieben und die Ergebnisse ausgelesen werden können.

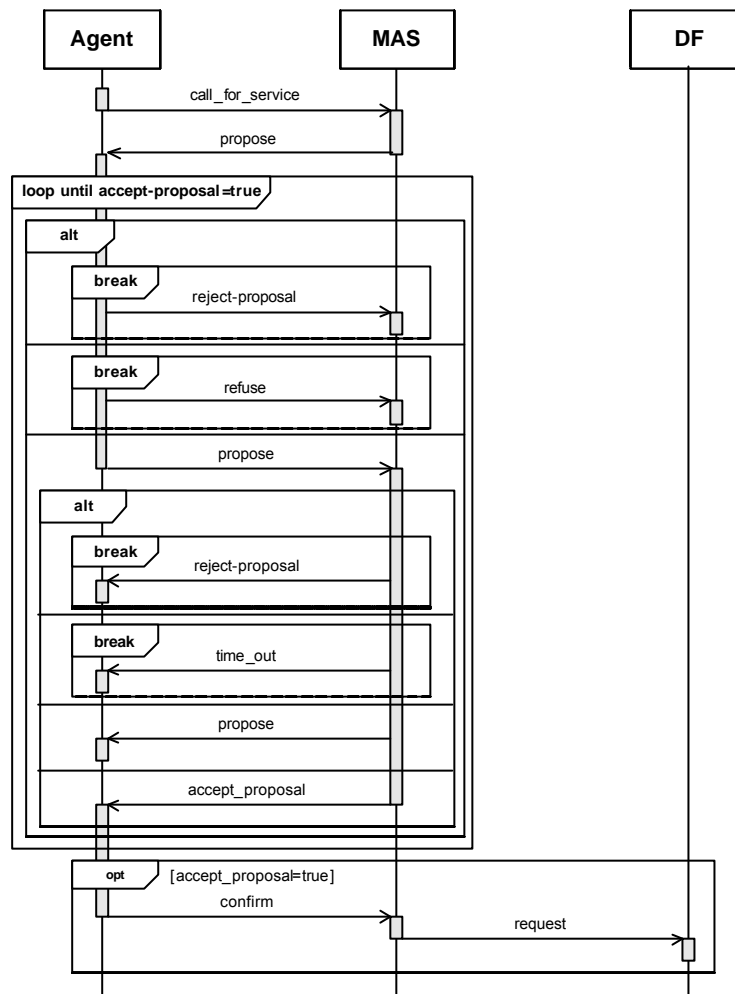


Abb. 3: Sequenzdiagramm des Interaktionsprotokolls zum Eintritt eines Agenten in ein MAS

Die Verhandlung zwischen dem System und dem Agent über eine mögliche Mitgliedschaft ist im Sequenzdiagramm (Abb. 3) dargestellt. Der Regelabgleich zwischen System und Agent ist im vorgestellten Prototyp gewichtungsabhängig realisiert. An einem verbesserten Algorithmus zum Regelabgleich auf Basis des Constraint Satisfaction Problem Verfahrens [RuNo03, 141], wird derzeit gearbeitet. Ist der Abgleich beendet, wird der Agent Mitglied des Systems, indem ein „Vertrag“ (Contract) durch das System und den Agenten vereinbart wird. Dieser Vertrag ist in einer zentralen Datenbasis gespeichert, um bei eventuell auftretenden späteren Verfehlungen darauf zurückgreifen zu können und gegebenenfalls Sanktionen zu verhängen, die von Ressourcenbeschränkungen bis hin zu einem Ausschluss eines Agenten aus einem System reichen



können. Für die Kontrolle der Einhaltung der Regelmenge durch den Agenten empfehlen sich Ansätze der s.g. Soft Security, wie zum Beispiel Reputationsmechanismen [Pado00].

#### 4 Prototyp

Der im Folgenden vorgestellte Prototyp ist das Ergebnis der Entwicklung einer agentenbasierten Transportnetzwerk-Simulation. Der Fokus hierbei liegt in der Untersuchung des dezentralen Koordinationskonzeptes in dem Bereich der Logistik. Im modellierten logistischen Netzwerk werden Stückgüter, Umschlagplätze und Fahrzeuge jeweils durch Softwareagenten repräsentiert. Das Stückgut erreicht einen Umschlagplatz an Bord eines Fahrzeuges und kann auf diesem beliebig umgruppiert werden. Die Entscheidung über die Beladung der Fahrzeuge erfolgt nach festgelegten Regeln durch direkte Verhandlungen zwischen den Stückgütern. Hierbei können Entscheidungen über den Weitertransport noch während der Fahrt auf den Fahrzeugen getroffen werden. Die Implementierung des Prototyps setzt auf eine weitgehende Trennung von Ablauf und Darstellung (Abb. 4).

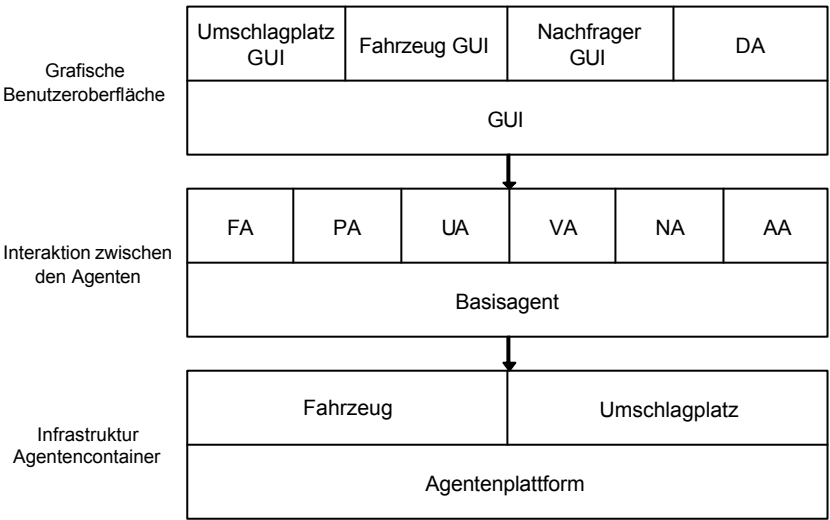


Abb. 4: Ebenen der Implementierung

Die unterste Ebene stellt die benötigte Infrastruktur und Basisdienste zur Verfügung. Auf der mittleren Ebene agieren und kommunizieren die Agenten untereinander. Hier sind die Funktionen und Eigenschaften der Agenten eingebunden. Graphical User Interfaces (GUI) für die einzelnen Agententypen werden auf der oberen Ebene repräsentiert.

Der Prototyp verfolgt die Strategie der dezentralen Koordination der logistischen Steuerelemente wie Fahrzeuge oder Stückgüter. Der Prototyp basiert auf dem *Java Agent Development Environment* (JADE) [JADE06]. JADE stellt eine Middleware auf Basis von Java dar und unterstützt die Entwicklung von Multiagentensystemen durch die Bereitstellung einer Agentenplattform den Rahmen für die Entwicklung von Agenten. Auf Grund der Konformität von JADE mit dem FIPA-Standard ist eine hohe Interoperabilität des Prototyps sichergestellt.

## **5 Zusammenfassung und Ausblick**

Unternehmensübergreifende Transportnetzwerke zeichnen sich durch eine zunehmende Komplexität und vor allem eine natürliche Verteiltheit der Ressourcen aus. Diese Charakteristiken lassen den Einsatz von Agenten zur Planung und Koordination der logistischen Prozesse als geeignet erscheinen. Dabei wurde der Ansatz der Verwendung von Regeln zur Beschreibung und Definition von Anforderungen und Rahmenbedingungen des logistischen Netzwerks, der zur Verfügung stehenden Plattformen und deren Akteuren bislang nur unzureichend betrachtet. Durch deren Definition ist es nun möglich, auch in einer weitgehend heterogenen Systemlandschaft, mit unterschiedlich verwendeten Protokollen, Interoperabilität und vor allem Koordinationsfähigkeit herzustellen. Dabei spielen nicht nur technische Aspekte eine Rolle, sondern es können auch organisatorische und soziale Sachverhalte definiert werden. Der Ansatz schließt damit die Lücke zwischen dem Ziel einer hohen Flexibilisierung logistischer Prozesse und dem Wunsch, die Einhaltung vereinbarter Geschäftsregeln zu überwachen. Die Formalisierung der Regeln erfolgte mittels der Object Constraint Language, die eine direkte Überführung in Programmcode erlaubt. Durch die aufgezeigte Regelontologie wird die semantische Klarheit der Kommunikation zur Vereinbarung von Regeln zwischen den Akteuren verbessert. Weiterhin wurde ein Ansatz zur Aushandlung von Regeln vorgestellt. Im dargestellten Prototyp sind die Ansätze implementiert.

Für die zukünftige Weiterentwicklung sind weitere domänenspezifische Ontologien geplant. Der Prozess der Verhandlung und des Abgleichs der Regeln wird flexibilisiert und vereinfacht. Abschließend kann festgestellt werden, dass das Regelkonzept in vorliegendem Papier zwar in Hinblick auf agentenbasierte Anwendungen entwickelt wurde, jedoch ebenso für weitere nicht agentenspezifische Systeme, wie zum Beispiel elektronische Märkte, denkbar ist.

## Literaturverzeichnis

- [Anth03] Anthes, G. H.: Agents of Change. [www.computerworld.com](http://www.computerworld.com), 2003.
- [BBGL+04] Benisch, M.; Greenwald, A.; Grypari, I.; Lederman, R.; Naroditskiy, V.; Tschantz, M.: Botticelli: A Supply Chain Management Agent. In: Proceedings of Third International Joint Conference on Autonomous Agents and Multi-Agent Systems, ACM Press, 2004.
- [BrGr05] Business Rules Group, <http://www.businessrulesgroup.org>, 2005.
- [Buss98] Bussmann, S.: Agent-Oriented Programming of Manufacturing Control Tasks. In: Proceedings of the 3rd International Conference on Multi-Agent Systems. IEEE CS Press, 1998, S. 57 - 63.
- [CoJe96] Cockburn, D.; Jennings, N.R.: Archon: A Distributed Artificial Intelligence System for Industrial Applications. In: O'Hare, G.M.P.; Jennings, N.R. (Hrsg.): Foundations of Distributed Artificial Intelligence. John Wiley, 1996, S. 319-344.
- [DaHe04] Davidsson, P.; Henesey, L et al: Agent-Based Approaches to Transport Logistics. AAMAS Workshop on Agents in Traffic and Transportation, 2004.
- [FIPA06] [www.fipa.org](http://www.fipa.org), 2006, Abruf am 10. Juli 2006.
- [Forg82] Forgy, C.: Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem. Artificial Intelligence 13(1982)19, S 17-37.
- [FrMu96] Fritzing, J.; Mueller, M.: Java Security. White Paper, Sun Microsystems, 1996.
- [GrGr05] Gaudina, V.; Grundspenkis, J.: Technologies and Multi-Agent System Architectures for Transportation and Logistics Support: An Overview. In: International Conference on Computer systems and Technologies – CompSysTech IIIA.6-1 – IIIA.6-2, 2005.
- [Grub93] Gruber, T. R.: A Translation Approach to Portable Ontology Specifications. In: Knowledge Acquisition 2 (1993) 5, S.199-220.
- [HaVe99] Hage, J., Verheij, B.: The law as a dynamic interconnected system of states of affairs – a legal top ontology. In: International Journal of Human - Computer Studies 6 (1999) 51, S. 1043-1078.
- [HeKn94] Herbst, H.; Knolmayer, G.: Arbeitsbericht Nr. 46 - Ansätze zur Klassifikation von Geschäftsregeln. Institut für Wirtschaftsinformatik, Universität Bern, 1994.
- [Jade06] Jade Agent Development Environment: <http://jade.cselt.it/>, 2006.

- [Jess06] Java Expert System Shell, 2006, <http://www.sandia.gov>, Abruf am 10. Juli 2006.
- [KiKu92] Kieser, A.; Kubicek, H.: Organisation. Walter de Gruyter, Berlin 1992.
- [Kies99] Kieser, A.: Konstruktivistische Ansätze. In: Kieser, A. (Hrsg.): Organisationstheorien. Verlag W. Kohlhammer, Stuttgart 1999, S. 287-318.
- [KnTi99] Knirsch, P.; Timm, I. J.: Adaptive Multiagent Systems Applied on Temporal Logistics Networks. In: Proceedings 4th International Symposium on Logistics, 1999.
- [Müll97] Müller, J.P.: Towards Agents Systems Engineering. International Journal on Data and Knowledge Engineering, In: Special Issue on Distributed Expertise 23. 1997.
- [OCL04] UML 2.0 OCL Specification. <http://www.omg.org/docs/ptc/03-10-14.pdf>, 2004, Abruf am 10. Juli 2006.
- [Pado00] Padovan, B.: Ein Vertrauens- und Reputationsmodell für Multi-Agenten Systeme. Dissertation, Albert-Ludwigs-Universität Freiburg, 2000.
- [Paru00] V.D. Parunak, H.: A Practitioners' Review of Industrial Agent Application. In: Autonomous Agents and Multi-Agent Systems 3 (2000) 4, S. 389-407.
- [PLSP04] Perugini, D; Lambert, D.; Sterling, L.; Pearce, A.: Agent-Based Global Transportation Scheduling in Military Logistics. In: Proceedings of Third International Joint Conference on Autonomous Agents and Multi-Agent Systems, ACM Press, 2004.
- [RuNo03] Russel, S.; Norvig, P.: Artificial Intelligence : a modern approach. 2. ed. Prentice-Hall, New Jersey 2003.
- [Sche99] Scherer, A.G.: Kritik der Organisation oder Organisation der Kritik? – Wissenschaftstheoretische Bemerkungen zum kritischen Umgang mit Organisationstheorien. In: Kieser, A. (Hrsg.): Organisationstheorien. Verlag W. Kohlhammer, Stuttgart, 1999, S. 1-37.
- [TaWa02] Taveter, K.; Wagner, G.: Agent-Oriented Enterprise Modeling Based on Business Rules. In: Arisawa, H.; Kambayashi, Y. (Hrsg.): Conceptual modeling for new information systems technologies. Springer-Verlag, Berlin 2002.
- [WoJe95] Wooldridge, M.; Jennings, N.R.: Agent Theories, Architectures, and Languages: a Survey. In Wooldridge, M.; Jennings, N.R. (Hrsg.): Intelligent Agents. Springer-Verlag, Berlin 1995, S. 1-22.