

Autonome Software

Wilfried Brauer
Technische Universität München
brauer@in.tum.de

Abstract: Die neuen Konzepte der Datenverarbeitung wie Peer-to-Peer, Grid, Service-oriented, Pervasive, Ubiquitous, Organic, Agent-based oder Autonomic Computing sowie Ambient Intelligence erfordern explizit oder implizit die Verwendung von autonomer Software – insbesondere als Mittel zur Reduzierung von Handhabungs- oder Administrations-Komplexität von Anwendungs-Softwaresystemen. Die enorme Bedeutung, die den genannten Konzepten für die Weiterentwicklung der Informatiknutzung zugesprochen wird, macht es nötig zu fragen, was denn Autonomie von Software eigentlich bedeuten kann und leisten soll und wie sie erzeugt werden könnte. Im Folgenden wird – nach einleitenden Bemerkungen – auf eine Reihe von (an meinem Lehrstuhl entstandenen) Publikationen mit neuen konstruktiven Ansätzen zur Beantwortung dieser Fragen hingewiesen.

1 Einleitung

Der Begriff Softwareautonomie meint eine Vielfalt von anthropomorphen Vorstellungen, die als Anregungen zur Gewinnung oder Hilfen für die Beschreibung neuer Konzepte und Methoden zum Entwurf von Software verwendet werden – wie das in der Informatik (und insbesondere in der KI) schon immer erfolgreich geschah.

Es geht dabei *erstens* um Komplexitätseingrenzung: Um Verringerung der Schwierigkeiten bei der Benutzung ebenso wie der Administration von komplexer Software, Reduktion der Außenbeziehungen von sonst schwer beherrschbaren Teilsystemen, Einkapselung von fehleranfälligen Teilsystemen, Aufteilung von Aufgaben und Delegation von Teilaufgaben an mehrere Leistungserbringer zur Erhöhung der Leistung, der Flexibilität oder der Robustheit des Gesamtsystems oder zur ‚Privatisierung‘ von Risiken, um Änderung der Entscheidungsbefugnisse und Verantwortlichkeiten sowie deren Verteilung durch Verlagerung aus einer globalen (oberen) Ebene auf eine (untere) Ebene lokaler Teilsysteme u.s.w. – also um die Weiterentwicklung der traditionellen Informatikmethoden im Hinblick auf den Ausbau der oben genannten neuen Konzepte.

Zweitens geht es insbesondere um das Ziel, dass zur Entwurfszeit eines Softwaresystems nicht mehr dessen gesamtes Verhalten bis ins kleinste Detail exakt festgelegt werden muss. D.h. man will den Teilsystemen (Objekten, Modulen, Komponenten, ‚Services‘, Diensten, Agenten) gewisse Unabhängigkeit, Eigenständigkeit, Eigenverantwortung, Selbstständigkeit, Entscheidungsfreiheit, Handlungsbefugnisse geben und sie befähigen zu Selbstorganisation, Selbstmanagement, Selbstbeobachtung, Selbstprüfung, Selbstevaluierung (hin-

sichtlich Performanz, Zuverlässigkeit oder Sicherheit), Selbsthilfe, Selbstreparatur, Selbstschutz (vor störenden, möglicherweise gefährlich werdenden Einflüssen, Eingriffen von außen), zu Eigeninitiative, Adaptivität, Lernen, Flexibilität, Mobilität – jeweils unter Beachtung von Rahmenbedingungen (Normen, Verboten, Geboten) und Leistungsanforderungen – wobei zugelassen sein kann, dass sich Leistungsanforderungen oder Rahmenbedingungen (durch externe Einflüsse oder durch Handlungen oder Entscheidungen der Teilsysteme) ändern. Um diese Anforderungen erfüllen zu können, benötigen die Teilsysteme Wissen, Interessen, Antriebe, Motivationen, Intentionen, Ziele sowie Ressourcen und Fähigkeiten zu rationalem Schlussfolgern und Handeln. Und sie müssen die ‚Freiheit‘ haben, sich selbst zu aktivieren, d.h. sie dürfen auch, ohne erst durch Aufrufe von außen aufgefordert zu werden, selbständig aktiv werden.

Schließlich – *drittens* – sollen die Teilsysteme, die ich von nun an Agenten nennen werde, mit anderen Agenten in so genannten Multiagentensystemen nebenläufig aktiv sein (sich als Ganzes selbst koordinierend durch Kooperation oder Konkurrenz) oder auf innere Veränderungen (Ausfall von Teilen des Systems, Auftreten von Fehlern) bzw. externe Einflüsse (Änderung von Umweltbedingungen) gemeinschaftlich durch Änderung von Organisation und Verhalten reagieren können – und dazu müssen sie miteinander (mehr oder weniger frei) kommunizieren und sich (mehr oder weniger gut) verstehen können. Außerdem sollten sie lernen können, das künftige Verhalten anderer Agenten einzuschätzen, etwa um sich aufeinander verlassen (einander vertrauen) oder vor anderen schützen zu können.

2 Autonome Agenten und Multiagentensysteme

Die Konzepte der Agenten und der Multiagentensysteme (MASE) sind im letzten Vierteljahrhundert im Gebiet der Verteilten Künstlichen Intelligenz geschaffen worden (die Wurzeln reichen zurück bis in die Mitte des vorigen Jahrhunderts) – eine Vielzahl von Agentenkonzepten sowie von Entwürfen, Implementationen und konkreten Anwendungen von MASen wurde seitdem entwickelt (siehe z.B. [Wei99]). Seit Anfang der 1990er Jahre entstand das Gebiet der agentenorientierten Softwareentwicklung (siehe z.B. [WJ05]). Überlegungen zur Autonomie von Agenten spielten bisher noch keine große Rolle – ein wichtiger Anstoß für die genauere Untersuchung der Möglichkeit von (regulierbarer) Softwareautonomie war neben der Entwicklung der Computer- und Kommunikationstechnologie (insbesondere des Internets) die (unbemannte) Raumfahrt.

Es gibt zwar schon eine Reihe von speziellen Ansätzen, Autonomie von Software etwas genauer zu definieren und Aspekte von Autonomie in verschiedenen Anwendungen zu verwenden (siehe z.B. [NRW04]), aber ein Bereichs- und Anwendungs-unabhängiges allgemeines Rahmenwerk, das hilft, Autonomie-Aspekte in Anwendungen zu analysieren und dann zu spezifizieren – um sie konstruktiv bei der Softwareentwicklung einzusetzen – existiert noch nicht. In [RW05] wurde ein erster Versuch hierzu unternommen – dabei wurden neben den statischen (strukturellen) auch dynamische (sich zur Laufzeit ändernde) Aspekte berücksichtigt. Zur Veranschaulichung dient ein künstliches Anwendungsbeispiel, nämlich die Verlinkung von Webseiten durch Verhandlungen zwischen Webseiten-

Agenten, um die Zahl der für sie ‚nützlichen‘ Besuche auf den eigenen Webseiten zu maximieren.

Wesentliche Beiträge zur Weiterentwicklung von Theorie und Technologie der Softwareagenten und entstanden in dem gerade zu Ende gegangenen DFG-Schwerpunktprogramm Sozionik, das von den Soziologen Thomas Malsch, TU Hamburg-Harburg (Koordinator) und Werner Rammert, FU Berlin sowie H.J. Müller, TU Freiberg (Verteilte KI) und mir 1998 initiiert wurde und Ende 1999 mit einer Reihe interdisziplinärer Forschungsgruppen startete. Unser Teilprojekt (gemeinsam mit T. Malsch) hatte das Thema „Konfliktbewältigung und Strukturwandel: Gesellschaftstheorie als Bauanleitung für lernfähige Mehragentensysteme“. Das übergeordnete Ziel des Projekts war es, soziologisch fundierte Modelle und Architekturen für sehr große, skalierbare Multiagentensysteme zu entwickeln und zu untersuchen. Zu den geplanten inhaltlichen Schwerpunkten zählten dabei insbesondere Lernen und Konflikt-handhabung in Multiagentensystemen sowie das Engineering von solchen Systemen bzw. von agentenorientierter Software generell. Die Erreichung des übergeordneten Zieles wurde parallel in zwei miteinander eng kooperierenden Teilprojekten verfolgt – mit den Mitarbeitern Nickles (Informatik) und Lorentzen (Soziologie) bzw. Rovatsos (Informatik) und Paetow (Soziologie). Die beiden Teilprojekte (NL und RP) sollten auf sehr unterschiedlichen soziologischen Gesellschaftstheorien aufbauen: NL auf der sehr allgemeinen, abstrakten und eigentlich kontraintuitiv erscheinenden Systemtheorie N. Luhmanns, RP auf dem philosophischen (amerikanischen) Pragmatismus, dem symbolischen Interaktionismus und insbesondere der Handlungstheorie von H. Joas. Wir vermuteten, dass Luhmanns Theorie nur sehr schwer auf die Informatik anwendbar sein würde, weil Luhmann den Akteur und mit ihm den Handlungsbegriff aus seiner soziologischen Theorie eliminiert und durch den Kommunikationsbegriff ersetzt hatte. Die Handlungstheorie hingegen schien relativ direkt anwendbar zu sein. Das Gegenteil war der Fall. Herr Rovatsos musste erst die Mikrosoziologie von E. Goffman und dessen Konzepte des ‚frame‘ und des ‚framing‘ entdecken, ehe er soziologische Theorie in neue Konzepte und Methoden für die Informatik umsetzen konnte.

Luhmanns konstruktivistische Systemtheorie lieferte folgende Resultate für die Informatik:

- die Entwicklung eines neuartigen empirischen Kommunikationsmodells, basierend auf der Datenstruktur der evolvierenden, probabilistischen Erwartungsnetze (Expectation Networks – annotierte gerichtete Graphen) [NRBW04];
- auf diesen Netzen aufbauend die Spezifikation einer völlig neuartigen Semantik für Agenten-Kommunikationssprachen (empirische Semantik) sowie zugehöriger Algorithmen für das Erlernen von Kommunikationsstrukturen [NRW05b]. Diese für MASE mit echt autonomen Agenten einzig universelle Kommunikationssemantik besteht aus den von einem Beobachter erwartbaren Auswirkungen in Form nachfolgender Ereignisse, sie ist ohne Rückgriff auf (in offenen Systemen sowieso nicht vorhandenes) Wissen über ‚mentale‘ Zustände einzelner Agenten erlernbar;
- die Entwicklung einer Architektur für Multiagentensysteme, die von gängigen Multiagentenarchitekturen radikal abweicht (Mirror, HoloMAS) und Entwicklung eines

wissensbasierten Systems für die Speicherung und Abfrage von Kommunikationsstrukturen [NW05]; die Kommunikationsstrukturen abstrahieren (wie in Luhmanns Theorie) vom Verhalten einzelner Agenten, weil andernfalls die Komplexität sozialer Interaktion nicht handhabbar wäre;

- die Entwicklung einer Methode für die Konstruktion offener Multiagentensysteme, die den Entwurf und die Analyse solcher Systeme auf der Ebene sozialer Erwartungsstrukturen als soziale Systeme erlaubt (EXPAND); sie wurde inzwischen verbessert zu EOM [NRW05a]; und
- eine mehrdimensionale Beschreibung von Semantic-Web bzw. Peer-to-Peer Ressourcen durch das Zusammenwirken mehrerer Agenten mit heterogenen Meinungen [NW03], [NW04]; hieraus entwickelte sich ein neues (mittlerweile von der DFG gefördertes) Vorhaben zum Thema „Kommunikative Erzeugung, Darstellung und Anwendungen Offener Ontologien und Offener Wissensbasen“ [FNW⁺05], [NF04].

Die interaktionistische Soziologie und insbesondere Goffmans Konzepte der Rahmen (frames) und Rahmungen (framings) lieferten – zusammen mit dem zu Beginn des Projekts gemeinsam entwickelten Konzept der empirischen Semantik – insbesondere folgendes [Rov04]:

- ein völlig neuartiges abstraktes Konzept (und eine formale Beschreibung) einer Architektur für sozial intelligente Agenten (InFFrA), die sich eignet sowohl für die Konstruktion von Agenten in abgeschlossenen Systemen, insbesondere aber auch für Agenten, die in offenen Systemen agieren sollen. Grundlegend ist die Idee, dass das Verhalten eines Agenten entsteht durch schrittweise Anwendung von situationsspezifischen Mustern für Interaktionen 'von Angesicht zu Angesicht', welche den Agenten per Konstruktion mitgegeben und durch seine eigenen Erfahrungen an neue Verhältnisse angepasst oder sogar gänzlich neu erlernt werden können [RP05, PR03],
- (computationelle) Interaktionsrahmen als eine neue komplexe Datenstruktur für Interaktionsmuster, d.h. für die Repräsentation von Wissen, das für rationales Verhalten in gewissen Interaktionssituationen nützlich oder notwendig ist, und auf der die Schlussfolgerungskomponente eines Agenten arbeitet. Gespeichert wird dabei nicht nur das, was als Voraussetzung für die Anwendung des Rahmens in einer konkreten Situation zu berücksichtigen und durch Beobachtung der Umgebung zu verifizieren ist, sondern auch einiges von dem, was durch ‚Introspektion‘ des Agenten von ihm selbst über seinen eigenen Status erkannt werden kann – inklusive aus Erfahrung gewonnene Vermutungen über das Wissen, das andere Agenten über den betreffenden Rahmen gewonnen zu haben scheinen. Ferner werden Angaben über die bei und nach Anwendung des Rahmen möglichen Folge-Interaktionen bereitgehalten [FR05, Rov01],
- ein computationelles, formales Modell des Prozesses der Rahmung, in dem insbesondere spezifiziert wird, wie ein Agent in konkreten Situationen Rahmen auswählt, wie Rahmen adaptiert und neue Rahmen erlernt werden und das neue Möglichkeiten

der Optimierung des Verhaltens von Agenten ermöglicht (durch Berücksichtigung der Wirkung des eigenen Verhaltens auf die eigenen kommunikativen Erwartungen) [FR05, RNW04, FRW05, RRFW05]

- eine Implementierung einer skalierbaren Testumgebung (**LIESON**) für das Szenario der Webseiten-Verlinkung. Agenten vertreten hier unterschiedliche Websites und können mit Beurteilungen (in Form numerischer Kantengewichte) versehene Links zu anderen Seiten legen, löschen, oder deren Gewichte verändern. In dem globalen Link-Netzwerk können auf der Basis eines unterstellten ‚Besucher‘-Verhaltens Popularitäten der Websites (d.h. der Agenten) bestimmt werden; jeder Agent strebt danach, die eigene Popularität sowie die favorisierter Websites Dritter zu maximieren. Dies ist ein hochinteressantes Anwendungsszenario für intelligent verhandelnde Agenten, da die Agenten praktisch nie vollständiges Wissen über das sich ständig wandelnde Link-Netzwerk erlangen können, unterschiedliche (nicht notwendigerweise kompatible) Motive haben und ihre eigene ‚Macht‘ (d.h. Popularität) optimal nutzen müssen, um andere Agenten zu vorteilhaften Link-Modifikationen zu bewegen. **LIESON** ist ein vollständig Java-basiertes System, das sowohl massiv verteilte als auch zentralisierte, nebenläufige und sequentielle Multi-Agenten-Simulationen zulässt. Durch die Anbindung an die generische Agentenplattform JADE kann das System – auf beliebig viele Rechner verteilt – die berechnungsmäßige Last komplexer Agenten problemlos bewältigen [Rov04].

Über die aus dem Sozionik-Schwerpunktprogramm hervorgegangenen Arbeiten hinaus entstanden und entstehen eine Reihe weiterer Arbeiten, z.T. in anwendungsorientierten Folgeprojekten – mehrere davon in Zusammenarbeit mit verschiedenen Firmen. Hier sei nur noch auf zwei Publikationen zur Spezifikation von Agenten-Autonomie hingewiesen, die beide von den soziologischen Konzepten der Rollen (Beschreibungen erwünschter Verhaltensweisen), Normen (Verbote, Gebote, Erlaubnisse) und Sanktionen (Bestrafungen und Belohnungen) ausgehen.

In [WDRN04, WNRF05] wird eine allgemeine Methode sowie eine spezielle, auf deontischer Logik basierende, formale Sprache (RNS2) zur Spezifikation sowohl von normkonformem wie normabweichendem Rollenverhalten und den zugehörigen Sanktionen vorgestellt, die es u.A. erlaubt, Autonomie und Kooperationsverhalten von Agenten so zu spezifizieren, dass schon zur Entwurfszeit eines MASs potentielle Kooperationskonflikte automatisch detektiert werden können. Wichtig dabei ist, dass auch potentielle dynamische Änderungen von Normen und Sanktionen im MAS modelliert werden können.

In [WFNR05] wird eine Variante von RNS2 zur Autonomie-Spezifikation (ASL) präsentiert, bei der auf die explizite Spezifikation von Normveränderungen verzichtet wird (da es diese sehr schwer machen, zukünftiges Systemverhalten vorherzusehen bzw. Verhaltensstabilität zu garantieren), bei der aber präzise unterschieden wird zwischen der Aktivierung/Deaktivierung von Normen einerseits (z.B. Aktivierung der Verpflichtung zur Lieferung einer Ware) und Handlungsaufforderungen andererseits (z.B. der Aufforderung zu liefern). ASL ermöglicht die automatische Detektion und Behandlung von Normkonflikten, so dass entweder zur Entwurfszeit oder durch passende Maßnahmen zur Laufzeit Konflikte gelöst werden können.

3 Schlussbemerkung

Es ist allgemein bekannt, dass zu große Handhabungskomplexität von Softwaresystemen sowohl bei Benutzern als auch (und sogar) bei Systemadministratoren zu ernsthaftem *Fehlverhalten* führt, das weitreichende negative Auswirkungen haben kann. Die anfangs erwähnten neuen Konzepte der Datenverarbeitung haben deshalb zum Ziel, die Handhabungskomplexität von Softwaresystemen zu senken – insbesondere mit Hilfe des Konzepts der Autonomie. Es ist insbesondere im Bereich der Verteilten KI und speziell der MASE schon sehr viel zum besseren Verständnis und zur effektiven Nutzung von Softwareautonomie beigetragen worden. Das reicht aber noch längst nicht dafür aus, dass sich Softwareautonomie in der Praxis bewährt und durchsetzt. Neben beträchtlicher Weiterentwicklung der vorhandenen und Neukonzeption weiterer Methoden und Verfahren zur exakten, formalen Spezifikation und automatischen Analyse, Verifikation und Simulation von Autonomie-Aspekten in der Entwicklungsphase sind neue Konzepte zur Laufzeitüberwachung des Verhaltens von autonomen Systemen erforderlich. Ein interessanter Ansatz hierzu (mit dem wir uns jetzt beschäftigen wollen) ist der des 'computational trust' [RHJ04]: Agenten müssen zu anderen Agenten, Verfahren (z. B. Auktionen) oder Institutionen (etwa Marktplätzen) 'Vertrauen' entwickeln und den jeweiligen Grad ihres Vertrauens einschätzen können – nur wenn ausreichendes Vertrauen vorhanden ist, wird kooperiert. Hoher computational trust sollte auch das menschliche Vertrauen in das benutzte System erhöhen können.

Literatur

- [FNW⁺05] Tina Fröhner, Matthias Nickles, Gerhard Weiss, Wilfried Brauer und Rolf Franken. Integration of Ontologies and Knowledge from Distributed Autonomous Sources. *Künstliche Intelligenz (KI), Special Issue on Knowledge Representation, Reasoning and Management*, 1/05:18–23, 2005.
- [FR05] Felix Fischer und Michael Rovatsos. An Empirical Semantics Approach to Reasoning About Communication. *Engineering Applications of Artificial Intelligence, Special Issue on Best Papers of CIA 2004*, 2005. To appear.
- [FRW05] Felix Fischer, Michael Rovatsos und Gerhard Weiß. Acquiring and Adapting Probabilistic Models of Agent Conversation. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, Utrecht, The Netherlands, 2005. ACM Press. To appear.
- [NF04] Matthias Nickles und Tina Froehner. Social Reification for the Semantic Web. Research Report FKI-244-01, AI/Cognition Group, Institut für Informatik, Technische Universität München, 2004.
- [NRBW04] Matthias Nickles, Michael Rovatsos, Wilfried Brauer und Gerhard Weiß. Towards a unified model of sociality in multiagent systems. *International Journal of Computer and Information Science*, 5(2):73–88, 2004.
- [NRW04] Matthias Nickles, Michael Rovatsos und Gerhard Weiß, Hrsg. *Agents and computational autonomy. Potential, risks, and solutions*, Jgg. 2969 (Hot Topics) of *Lecture Notes in Artificial Intelligence*, Berlin, Germany, 2004. Springer.

- [NRW05a] Matthias Nickles, Michael Rovatsos und Gerhard Weiss. Expectation-Oriented Modeling. *Engineering Applications of Artificial Intelligence*, 18, 2005. To appear.
- [NRW05b] Matthias Nickles, Michael Rovatsos und Gerhard Weiss. Formulating Agent Communication Semantics and Pragmatics as Behavioral Expectations. In Frank Dignum, Rogier van Eijk und Marc-Philippe Huget, Hrsg., *Agent Communication Languages II*, Lecture Notes in Artificial Intelligence, Berlin, Germany, 2005. Springer.
- [NW03] Mathias Nickles und Gerhard Weiss. A framework for the social description of resources in open environments. In *Proceedings of the 7th International Workshop on Co-operative Information Agents (CIA)*, Jgg. 2782 of *Lecture Notes in Computer Science*, Seiten 206–221. Springer, 2003.
- [NW04] Matthias Nickles und Gerhard Weiss. Agent-based social assessment of shared resources. In *Second International Workshop on Agents and Peer-to-Peer Computing (AP2PC)*, 2004.
- [NW05] Matthias Nickles und Gerhard Weiß. Multiagent systems without agents: Mirror-holons for the compilation and enactment of communication structures. In K. Fischer und M. Florian, Hrsg., *Socionics*, Lecture Notes in Artificial Intelligence. Springer, 2005. To appear.
- [PR03] Kai Paetow und Michael Rovatsos. Grundlagen einer interaktionistischen Sozionik. Bericht Research Report RR-8, Department of Technology Assessment, Technische Universität Hamburg, 2003.
- [RHJ04] Sarvapali D. Ramchurn, Dong Huynh und Nicholas R. Jennings. Trust in Multi-Agent Systems. *The Knowledge Engineering Review*, 19(1):1–25, 2004.
- [RNW04] Michael Rovatsos, Matthias Nickles und Gerhard Weiß. An empirical model of communication in multiagent systems. In F. Dignum, Hrsg., *Agent communication languages*, Lecture Notes in Computer Science, Vol. 2922, Seiten 18–36. Springer, 2004.
- [Rov01] Michael Rovatsos. Interaction Frames for Artificial Agents. Research Report FKI-244-01, AI/Cognition Group, Institut für Informatik, Technische Universität München, 2001.
- [Rov04] Michael Rovatsos. *Computational Interaction Frames*. Dissertation, Institut für Informatik, Technische Universität München, 2004.
- [RP05] Michael Rovatsos und Kai Paetow. On the Organisation of Social Experience: Scaling up Social Cognition. In Klaus Fischer und Michael Florian, Hrsg., *Socionics: Its Contributions to the Scalability of Complex Social Systems*, LNCS. Springer, Berlin, Germany, 2005. To appear.
- [RRFW05] Michael Rovatsos, Iyad Rahwan, Felix Fischer und Gerhard Weiss. Adaptive Strategies for Practical Argument-Based Negotiation. In *Proceedings of the 2nd International Workshop on Argumentation in Multi-Agent Systems (ArgMAS)*, Utrecht, The Netherlands, 2005. To appear.
- [RW05] Michael Rovatsos und Gerhard Weiss. Autonomous Software. In In S.K. Chang, Hrsg., *Advances in Software Engineering & Knowledge Engineering*. World Scientific Publishing, 2005. To appear.
- [WDRN04] Gerhard Weiss, Johann Duscher, Michael Rovatsos und Matthias Nickles. Spezifikation von Softwareautonomie. *Künstliche Intelligenz*, 4/04:44–50, 2004.

- [Wei99] Gerhard Weiß, Hrsg. *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*. The MIT Press, Cambridge, MA, 1999.
- [WFNR05] Gerhard Weiss, Felix Fischer, Matthias Nickles und Michael Rovatsos. Operational Modelling of Agent Autonomy: Theoretical Aspects and a Formal Language. In *Proceedings of the 6th International Workshop on Agent-Oriented Software Engineering (AOSE)*, Utrecht, The Netherlands, 2005. To appear.
- [WJ05] Gerhard Weiß und Ralf Jakob. *Agentenorientierte Softwareentwicklung – Methoden und Tools*. Xpert.press. Springer, 2005.
- [WNR05] Gerhard Weiss, Matthias Nickles, Michael Rovatsos und Felix Fischer. Specifying the Intertwining of Coordination and Autonomy in Agent-Based Systems. *Journal of Network and Computer Applications*, 28, 2005. To appear.