

Intelligente Agenten und KI

Ulrich Furbach, Oliver Obst und Frieder Stolzenburg
Universität Koblenz-Landau, Fachbereich Informatik

9. April 2001

Zusammenfassung

Neuere KI-Textbücher stützen sich zur Begriffsdefinition von „Künstliche Intelligenz“ in der Regel massiv auf den Agentenbegriff. Autonomie, Körperhaftigkeit (embodiment), Reaktivität und Situiertheit in einem komplexen Kontext sind unmittelbar mit den Begriffen Agenten und KI verwoben. Hat man in der Vergangenheit eher versucht, einzelne Maschinen mit mächtigen Wissensverarbeitungsmechanismen auszustatten, wird aus heutiger Sicht auf die Interaktion mit der Umwelt und anderen Agenten gezielt. Natürlich muss auch hierbei Wissen repräsentiert und verarbeitet werden. Wir skizzieren die gängigsten Architekturprinzipien für Agenten und gehen unter anderem auf logikbasierte und BDI-Architekturen (BDI = Belief-Desire-Intention) näher ein. Außerdem werden zwei prinzipiell verschiedene Anwendungsmöglichkeiten an Beispielen beschrieben: Zum einen können Agenten benutzt werden, um Wissen zu beschaffen und aufzubereiten; die Verfügbarkeit von Wissen stellt zum anderen eine Voraussetzung für intelligentes Verhalten von autonomen Robotern dar. Dies wird veranschaulicht durch fußballspielende autonome Agenten im RoboCup und durch die Betrachtung eines Systems zur Informationsextraktion im Internet.

1 Die neue KI oder: Was sind Agenten?

So vielfältig wie die Literatur zum Thema *Agenten* ist, so vielfältig scheinen auch die Definitionen des Themas selbst zu sein: Brenner et al. [3] beispielsweise sehen Agenten als Programme, die dem Nutzer bei der Suche nach Informationen unterstützen und Aufgaben in einer vernetzten, digitalen Welt erledigen. Eine weit allgemeinere Definition findet man bei Russell und Norvig [8]: Ein Agent ist dort alles, was über *Sensoren* seine *Umgebung* wahrnimmt und diese wiederum durch *Aktionen* mittels seiner *Effektoren* beeinflusst. Legt man diese Begriffe nur weit genug aus, lässt sich von mechanischen Geräten über Computerprogramme bis hin zu Teams von autonomen Robotern alles als *Agent* klassifizieren. Nicht zuletzt wegen dieser Vielfalt wurden verschiedene Taxonomien entwickelt [10], die zum Teil aus der Verteilten Künstlichen Intelligenz übernommen wurden.

So können sich Agentensysteme beispielsweise hinsichtlich der Heterogenität der einzelnen Agenten (gleichartige gegenüber spezialisierten Agenten) oder hinsichtlich der Methoden bei

der Verteilung der Kontrolle unterscheiden. Agenten können als Team oder hierarchisch organisiert sein, sie können statische oder veränderliche Rollen haben und sie können miteinander im Wettbewerb stehen oder auch nicht. Andere Aspekte sind die Architekturen der Agenten oder die Architektur des Agentensystems. Es werden beispielsweise *reaktive* gegenüber *deliberativen* Agenten unterschieden. Mit reaktiven Agenten sind solche gemeint, die einen subsymbolischen Ansatz verfolgen. Sie reagieren mit Aktionen auf Reize, wodurch die Funktionalität des Agenten entsteht. Deliberative Agenten sind solche, die ihre Eingaben auf Symbole reduzieren und diese mit vorgegebenen Mechanismen bearbeiten. Zu den Merkmalen eines Agentensystems gehören beispielsweise Kommunikationsmöglichkeiten zwischen den Agenten und Protokolle.

2 Architekturen für Multiagenten-Systeme

Auch wenn es bis jetzt in der Literatur noch keine allgemein akzeptierte Definition des Agentenbegriffs gibt, so lassen sich doch eine Reihe von konkreten Architekturen für Multiagenten-Systeme unterscheiden. Im Folgenden betrachten wir Agenten zunächst einfach als Entitäten, die aufgrund von Wahrnehmungen und inneren Zuständen sich für bestimmte Aktionen entscheiden und diese dann auch tatsächlich in ihrer Umwelt ausführen. Die Ausführungen orientieren sich zum Teil an [13]. Zur Veranschaulichung für die verschiedenen Architekturen verwenden wir den Roboterfußball. Auf diese Anwendung werden wir noch genauer eingehen (in Abschnitt 3) – sowie später auf Agenten zur Informationsextraktion im Internet (in Abschnitt 4).

Logikbasierte Ansätze

Da wir in diesem Artikel uns in erster Linie auf intelligente Agenten für die Wissensverarbeitung konzentrieren wollen, sind zunächst einmal logikbasierte Ansätze interessant. Sie erlauben die symbolische Verarbeitung von Sensordaten, die ein Agent zur Verfügung hat, und eine kontrollierte Steuerung des Agenten durch explizite Regeln. Bei logikbasierten Agenten wird die Entscheidung, was als nächstes getan werden soll, realisiert durch logische Deduktion. Das Wissen (oder besser der Glauben) eines Agenten wird in einer Wissensbasis festgehalten in Form von logischen Propositionen: Regeln und Fakten.

Abbildung 1 zeigt schematisch und anhand des Fußball-Beispiels, wie die Entscheidungsfindung bei logikbasierten Agenten vorgestellt werden kann. Logikbasierte Agenten wählen nur Aktionen aus, die logisch aus der Datenbasis (DB) des Agenten geschlussfolgert werden können oder für die zumindest nicht ableitbar ist, dass sie nicht getan werden sollen. Das entspricht den beiden for-Schleifen in der Abbildung.

Die Hauptidee ist es also, dass ein Programmierer explizit Regeln spezifizieren kann, die besagen, unter welchen Umständen welche Aktion ausgeführt werden kann bzw. soll. Der große Vorteil dieser Vorgehensweise ist, dass das Verhalten von Agenten explizit kontrolliert und programmiert werden kann. Die Spezifikation eines Agenten ist ein Logikprogramm, das unmittelbar von einem Theorembeweiser auszuführen ist. Dieser muss natürlich mächtig genug sein, um mit allen vorkommenden Konstrukten umgehen zu können.

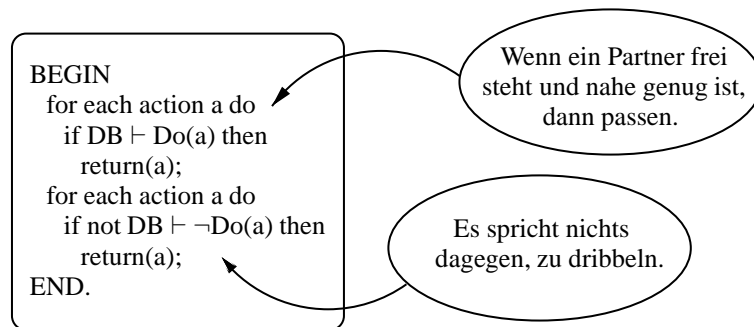


Abbildung 1: Auswahl von Aktionen bei logikbasierten Agenten

Subsumptionshierarchien

Kritiker des logikbasierten Ansatzes, der als klassischer KI-Ansatz verstanden werden kann, brachten mehrere Argumente dagegen hervor: Die Wahrnehmungen der Agenten seien subsymbolischer Natur. Die Übersetzung in eine symbolische Repräsentation erscheine daher umständlich. Außerdem könnten Theorembeweiser kein Echtzeitverhalten garantieren, sodass eine Interaktion mit der Umwelt schwierig ist.

Letzteres war aber gerade die Absicht der Embodiment-Bewegung in der KI. Agenten sollten reaktiver sein. Eine Möglichkeit, dies zu erreichen, ist die sogenannte Subsumptionshierarchie nach Brooks [4]. Die Grundannahme ist hier, dass intelligentes Verhalten durch das Zusammenwirken vieler kleiner Module entstehen kann und nicht durch eine wie auch immer geartete, komplizierte planende und kontrollierende Einheit. Verschiedene Module können parallel und eventuell mit unterschiedlichen Prioritäten arbeiten.

Abbildung 2 zeigt schematisch den Aufbau eines Moduls, das für den Torschuss verantwortlich ist. Die Ausgabe dieser Einheit kann überschrieben werden, wenn die Situation es erfordert, z.B. wenn es Abseits gibt. Weiter ist es möglich, dass Eingaben unterdrückt werden. Das erhöht die Flexibilität von Agenten. Eine explizite Kontrolle des Gesamtsystems, das aus vielen Modulen besteht, ist bei dieser Architektur schwierig zu bewirken. Auch lässt sich schwer sagen, was die Agenten eigentlich wissen, geschweige denn, wie sie es verarbeiten.

Belief-Desire-Intention

Die starke Reaktivität der gerade vorgestellten Agenten wird zum Problem, wenn Agenten langfristige Ziele verfolgen sollen. In diesem Zusammenhang war der philosophisch orientierte Ansatz von Bratman [2] sehr einflussreich. Er geht davon aus, dass Handeln nicht nur Reaktion ist auf äußere Wahrnehmungen und Reize. Agenten gehen nicht nur ihren unmittelbaren Bedürfnissen (*desires*) nach, sondern verfolgen langfristige Absichten (*intentions*). Dabei gehen sie von ihrer Sicht der Welt aus (*belief*). Abbildung 3 zeigt schematisch die BDI-Architektur und die funktionalen Abhängigkeiten zwischen den drei BDI-Komponenten mit Beispielen.

Dieser Ansatz fand bereits in Softwaresystemen Eingang und wurde formal weiterentwickelt.

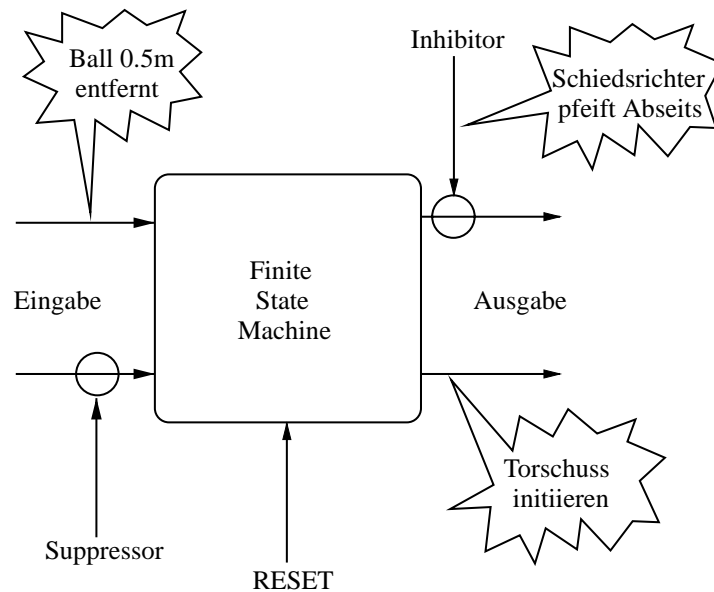


Abbildung 2: Reaktive Agenten durch Subsumptionshierarchien

Der Vorteil bei dieser Architektur ist, dass ein Agent über sein eigenes Handeln, seine Ziele und sein Wissen explizit Schlussfolgerungen anstellen kann. Diese Beziehungen werden meist mittels erweiterter Logiken, z.B. dynamische Logiken oder Modallogik, dargestellt [9]. Das macht es unter Umständen schwierig, BDI-Agenten (effizient) zu implementieren. Es gibt aber auch Versuche, BDI-Agenten direkt in Logik erster Stufe auszudrücken [7].

Schichten-Architekturen

Schichtenarchitekturen gehen davon aus, Wissen auf mehreren Abstraktionsebenen zu verarbeiten. Hier können auch verschiedene Grade von symbolischer und subsymbolischer Informationsverarbeitung kombiniert werden. Heutzutage ist es allgemein anerkannt, dass eine einzige Art der Wissensrepräsentation nicht ausreicht. Deshalb erscheinen aus heutiger Sicht die Unterschiede der bereits vorgestellten Ansätze nicht so unvereinbar, wie das erst angenommen wurde. Gerade die Einführung mehrerer Ebenen und Verarbeitung von Wissen unterschiedlicher Form macht diese Architekturen interessant aus Sicht der Wissensverarbeitung und Kognitionswissenschaft.

In Schichtenarchitekturen lassen sich Aspekte verschiedener Richtungen integrieren. Schichten können hierarchisch (horizontal) oder parallel (vertikal) angeordnet sein. Letzteres ermöglicht die angepasste Verarbeitung von Wissen jeder Form. So spielt auch in den hier betrachteten Anwendungen eine Mischung mehrere Wissensarten eine Rolle, z.B. werden für einen Torschuss genaue quantitative Daten benötigt, während die Erkennung einer Abseits-Falle einen mehr qualitativen Überblick der Agenten erfordert.

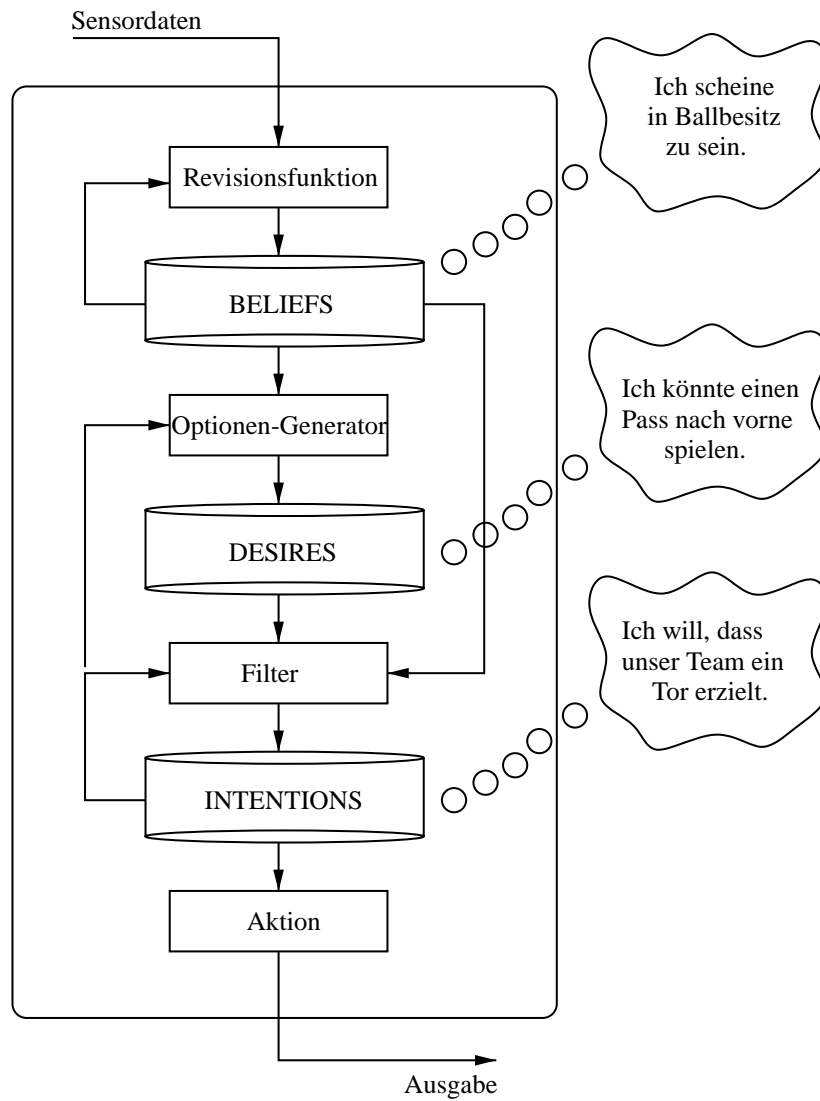


Abbildung 3: Ein BDI-Agent

3 Der RoboCup als Multiagenten-System

Eine Anwendung, in der Ansätze für Multiagenten-Systeme erforscht werden sollen, ist der RoboCup. Dabei wurden mehrere gemeinsame Plattformen geschaffen, die weltweit benutzt werden, um neue Ansätze zu entwickeln und diese mit anderen zu vergleichen. Im RoboCup gibt es mehrere Ligen, bei denen es darum geht, Robotern oder Computerprogrammen Fußball spielen beizubringen.

In der *Simulation League* wird das Spielfeld durch ein Programm simuliert (siehe Abbildung 4). Auf diesem müssen dann für jede Mannschaft elf Agenten in einem Spiel gegeneinander antreten. Die Agenten haben eine kleine Anzahl von Aktionen zur Auswahl, wie z.B. gegen den Ball treten, sich drehen oder beschleunigen. Die Eingaben, die ein solcher Agent bekommt, sind seine Sicht auf einen Ausschnitt des Spielfeldes. So wie die Sicht der Agenten eingeschränkt ist, so hat auch die Kommunikation der Agenten untereinander ihre Grenzen: nur kurze Zeichenketten können über das Spielfeld gerufen werden, die die Partner- und Gegner-Agenten in der näheren Umgebung hören können. Rufen mehrere Agenten gleichzeitig, entscheidet der Zufall, welche der Nachrichten ankommt. Dadurch, dass für alle Teilnehmer in einem solchen Spiel die Bedingungen gleich sind, lassen sich die erzielten Ergebnisse miteinander vergleichen. Allerdings sind neben der wissenschaftlichen Qualität des eigenen Ansatzes auch programmiertechnische Qualitäten entscheidend.



Abbildung 4: Spielsituation aus der *Simulation League*

Andere Ligen im RoboCup, in denen es auch um Fußball geht, sind die *Small Size League*, die *Middle Size League* und die *Sony Legged League*. Hier wird das Fußballfeld nicht mehr simuliert, es treten mehrere echte Roboter gegeneinander an (siehe auch Abbildung 5). Dadurch, dass hier zusätzliche Probleme mit der Hardware entstehen, aber auch weitere Aufgaben zu lösen sind, wie das Erkennen des Balls mit einer Kamera, sind diese Ligen – fußballtechnisch gesehen – noch weit vom Stand der Simulationsliga entfernt.

Die Fußball-Ligen im RoboCup bieten vieles von dem, was man sich für ein Multiagenten-System vorstellen kann: Die Agenten haben nur beschränkte Informationen über den momentanen Zustand ihrer Welt, beschränkte Kommunikationsmöglichkeiten und sie müssen möglichst

schnell reagieren. Einerseits müssen sie mit den Agenten des eigenen Teams kooperieren, andererseits stehen sie im Wettbewerb mit den Agenten des anderen Teams. Einen zusätzlichen Aspekt gibt es beim *RoboCup-Rescue*, bei dem es nicht um ein Spiel, sondern um das Retten von Personen nach einer Katastrophe, wie etwa einem Erdbeben geht. Hier kommt hinzu, dass neben den immer neuen Situationen auch die Umgebung unbekannt ist.

Für die Realisierung von RoboCup-Agenten kommen logikbasierte Agenten in Frage. So lassen sich Spielzüge leicht als logische Regeln formulieren. Eine Einbettung in eine Schichtenarchitektur ist sinnvoll. Diese Vorgehensweise wird in [6] verfolgt. Nichtsdestoweniger ist die Verwendung anderer Architekturen und Techniken durchaus gängig und von den Initiatoren des RoboCups sogar erwünscht.



Abbildung 5: *Middle Size League*-Roboter bei der Ladepause

4 Anwendungen in der Wissensverarbeitung

Die Schlagworte *allgegenwärtige Rechner*, *UMTS-basierte Multimedia-Dienste*, *E-Business* und viele andere mehr lassen ahnen, dass unser tägliches Leben immer mehr von Informationstechnik durchdrungen wird. Insbesondere die allgegenwärtige Verfügbarkeit von Wissen und das Zusammenwachsen verschiedenartigster Kommunikationsmöglichkeiten bietet einerseits spannende neue und hilfreiche Möglichkeiten, stellen aber andererseits immense Anforderungen an den Benutzer dar. Die Beherrschbarkeit dieser zunehmend komplexer werdender System ist nur denkbar, wenn möglichst autonome und intelligente Unterstützungssysteme entworfen werden.

Als Beispiel wollen wir hier die Informationsgewinnung im Internet heranziehen. Es existieren eine Vielzahl von äußerst leistungsfähigen Suchmaschinen, mit deren Hilfe Suchanfragen an das weltweite Internet gestellt werden können. In der Tat liefern diese Verfahren auch eindrucksvolle Listen von (mehr oder weniger) relevanten Webseiten, die der Benutzer nach den

gewünschten Inhalten durchsuchen muss. Diese Systeme liefern also eher Verweise auf Inhalte, anstatt diese Inhalte selbst. Der Benutzer muss sodann diesen Verweisen folgen, die Webseiten besuchen, um sich dann die entsprechenden Inhalte zu extrahieren.



Abbildung 6: Benutzeranfrage an einen Kleinanzeigen-Roboter

Hier können Agentensysteme hervorragend eingesetzt werden, um dem Benutzer sowohl die Suche, als auch die Extraktion der Inhalte aus den Webseiten abzunehmen. Im Beispiel aus Abbildung 6 sehen wir die Anfrage an einen Kleinanzeigen-Roboter (<http://www.uni-koblenz.de/home/bthomas/robot.html>), der es dem Benutzer gestattet, seine Suchanfrage auf einfache und natürliche Weise einzugeben. Hier ist z.B. spezifiziert, dass ein Schreibtisch unter 200 DM im Postleitzahlenbereich 02 gesucht wird. Durch Abschicken dieser Anfrage werden mehrere Software-Agenten aktiviert, die nun unabhängig voneinander vier verschiedene Kleinanzeigenanbieter im Internet besuchen. Die Agenten navigieren durch die jeweilige Seitenstruktur der Anzeigenanbieter und suchen jene Angebote, die auf die Anfrage passen. Als Ergebnisse werden dem Benutzer nun nicht die entsprechenden Links auf die einzelnen Webseiten geliefert; vielmehr extrahieren die Agenten die gesuchte Information aus diesen Seiten. Die Antwort auf unsere Beispielanfrage ist zu finden in Abbildung 7.

Dazu gehört einiges an Wissen über den Aufbau von Kleinanzeigen im Allgemeinen, aber auch über die spezielle Seitenstruktur der unterschiedlichen Anbieter. Natürlich darf diese Information nicht fest in die Systeme einprogrammiert werden, da die Agenten dann nicht auf Änderungen in

den Webseiten reagieren könnten und auch die Anzahl der zu besuchenden Anbieter nicht ohne Weiteres erweitert werden könnte.



Abbildung 7: Antworten des Kleinanzeigen-Roboter

Deshalb werden die Agenten mit Techniken des maschinellen Lernens in die Lage versetzt, während ihrer Arbeit zu lernen, wie die Struktur der Webseiten von Kleinanzeigenanbietern üblicherweise aussieht und wie daraus die relevante Information extrahiert werden kann. In [5] werden solche Lernverfahren diskutiert, in [11] wird die Extraktion von relevanter Information aus Webseiten diskutiert.

5 Schlussbemerkungen

In diesem Beitrag sind wir auf den Aufbau und die Verwendung von Multiagenten-Systemen in der KI eingegangen. Ausgehend von verschiedenen Architekturprinzipien für solche Systeme haben wir exemplarisch einige Anwendungsszenarien skizziert. Dabei war die Auswahl durchaus subjektiv; zum Schluss sei daher darauf hingewiesen, dass Agentensysteme in nahezu allen Bereichen der Informatik Anwendungen finden. Insbesondere im Bereich der Wirtschaftsinformatik (z.B. E-Commerce, E-Business) oder in Reiseunterstützungssystemen [1] finden sich zahlreiche Beispiele.

Literatur

- [1] Gerd Beuster, Bernd Thomas und Christian Wolff. Ubiquitous Web Information Agents. In *Proceedings of Workshop on Artificial Intelligence In Mobile Systems*, August 2000. European Conference on Artificial Intelligence (ECAI).
- [2] Michael E. Bratman. *Intention, Plans, and Practical Reason*. Harvard University Press, Cambridge, MA, London, England, 1987.
- [3] Walter Brenner, Rüdiger Zarnekow und Hartmut Wittig. *Intelligent Software Agents*. Springer-Verlag, Berlin, Heidelberg, New York, 1998.
- [4] Rodney A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23, 1986.
- [5] Gunter Grieser, Klaus P. Jantke, Steffen Lange und Bernd Thomas. A Unifying Approach to HTML Wrapper Representation and Learning. In *Proceedings of the Third International Conference on Discovery Science*, December 2000. Kyoto, Japan.
- [6] Jan Murray, Oliver Obst und Frieder Stolzenburg. Towards a Logical Approach for Soccer Agents Engineering. In Tucker Balch, Peter Stone und Gerhard Kraetzschmar (Hrsg.), *Proceedings of the 4th International Workshop on RoboCup*, S. 90–99, Melbourne, 2000.
- [7] Anand S. Rao. AgentsSpeak(L): BDI Agents speak out in a logical computable language. In Walter van de Velde und John W. Perrame (Hrsg.), *Agents Breaking Away – 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, LNAI 1038, S. 42–55, Berlin, Heidelberg, New York, 1996. Springer.
- [8] Stuart Russell und Peter Norvig. *Artificial Intelligence – A Modern Approach*. Prentice Hall, Englewood Cliffs, NJ, 1995.
- [9] Munindar P. Singh, Anand S. Rao und Michael P. Georgeff. Formal Methods in DAI: Logic-Based Representation and Reasoning. In Weiss [12], Kapitel 8, S. 331–376.
- [10] Peter Stone und Manuela Veloso. Multiagent Systems: A Survey from a Machine Learning Perspective. *Autonomous Robots*, 8(3):345–383, 2000.
- [11] Bernd Thomas. Token-Templates and Logic Programs for Intelligent Web Search. *Intelligent Information Systems*, 14(2/3):241–261, March-June 2000. Special Issue: Methodologies for Intelligent Information Systems.
- [12] Gerhard Weiss (Hrsg.). *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*. The MIT Press, Cambridge, MA, London, 1999.
- [13] Michael Wooldridge. Intelligent Agents. In Weiss [12], Kapitel 1, S. 27–77.