

High-Definition Multimedia

Grundlagen, Theorien, Architekturen
Agenten und Avatare

4. Februar 1999

<http://spellbreaker.org/~chrender/Agenten>
Zuletzt geändert am 6. Juli 2005.

Inhaltsverzeichnis

1	Geschichte und Ursprung	3
1.1	Ursprung der Agententechnologie	3
1.2	Erster Prototyp und Begriffsprägung	3
1.3	Zwei Entwicklungsrichtungen	3
2	Motivation	4
2.1	Verteiltes Rechnen	4
2.2	Verbessertes Interface	4
2.3	Gegenargumente	5
3	Grundlagen	5
3.1	Was ist ein Agent?	5
3.2	Eigenschaften von Agenten	6
3.3	Definition über Belief, Desire, Intention	7
3.4	Typologisierung von Agenten	10
3.5	Agenten als Vertreter	11
4	Theorien	11
4.1	Überblick	11
4.2	Moore – Knowledge and Action	11
4.3	Cohen and Levesque – Intention	11
4.4	Rao and Georgeff	11
4.5	Wooldridge	12
5	Agentenarchitekturen	12
5.1	Was sind Agentenarchitekturen	12
5.2	Reaktive Architekturen	12
5.3	Brook’s Subsumption Architecture	12
5.4	Planning Agents	13
5.5	Deliberative Agents	13
5.6	Hybride Architekturen	13
5.7	InteRRap Architecture	14
6	Die weitere Entwicklung	15
6.1	Forschende Organisationen	15
6.2	Zu lösende Probleme	15
6.3	Mögliche Agentenanwendungen	16

1 Geschichte und Ursprung

Um das Konzept eines Agenten besser begreifen zu können ist hier ein kleiner Ausflug in die Geschichte der Agenten aufgeführt.

1.1 Ursprung der Agententechnologie

Die Agenten kommen aus dem Bereich der künstlichen Intelligenz, genauer: aus dem Bereich der „Distributed Artificial Intelligence“ – DAI. Dieser Bereich ist in drei Teile aufgeteilt: die Multi-Agent-Systeme, das Distributed Problem Solving und die Parallel Artificial Intelligence. Die Software-Agenten stammen, wie man sich leicht denken kann, aus dem Bereich der „Multi-Agent-Systeme“.

Das Konzept eines Agenten kann bis zur Entstehung des DAI-Bereichs zurückverfolgt werden. 1970 formulierte Carl Hewitt das Modell eines „Actors“, eines interaktiven Objektes mit eigenem, internem Zustand, welcher auf Nachrichten von ähnlichem Objekten reagieren konnte. Ein solcher Actor ist ein computergestützter Agent, welcher eine Mail-Adresse hat und ein bestimmtes Verhalten aufweist. Actors kommunizieren mit Hilfe von Nachrichten and richten Ihre Aktionen nach diesen.

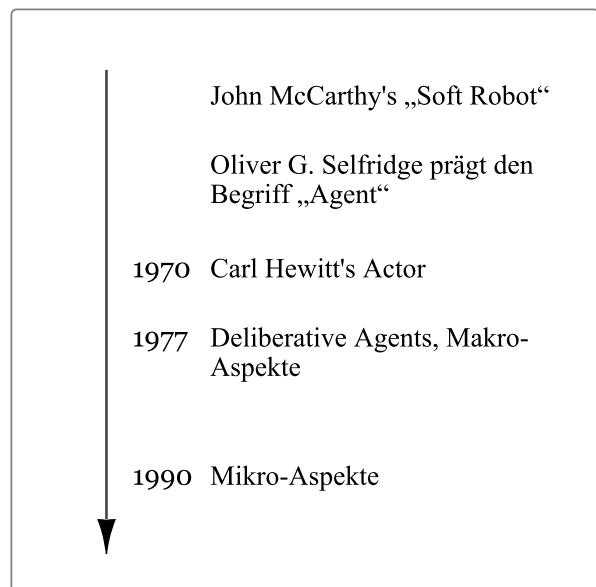
Obwohl Agenten eine sehr zentrale Rolle im Bereich der künstlichen Intelligenz spielen, hat man sich überraschenderweise bis in die späten Achtziger nur wenig praktische Gedanken über Agenten gemacht.

1.2 Erster Prototyp und Begriffsprägung

Der Ursprung der Agenten wird von Alan Kay, einem großen Befürworter der Agententechnologie, auf eine Idee von John McCarthy zurückgeführt. Der Begriff „Agent“ wurde von Oliver G. Selfridge einige Jahre später geprägt. Beide arbeiteten am Massachusetts Institute of Technology. Sie dachten an ein System, welches selbstständig eine gestellte Aufgabe lösen sollte, also ein „Soft Robot“, der im Gegensatz zum populären Bild vom Roboter nicht in der physikalischen, „realen“ Welt agiert, sondern im Computer.

1.3 Zwei Entwicklungsrichtungen

Die Entwicklung der Agenten lief seit ihrer Entstehung in zwei unterschiedliche Richtungen. Nach Ansicht von Hyacinth S. Nwana ([NWANA], Seite 2 und [BRADS], Seite 4) lief die erste Entwicklung ab etwa 1977 und beschäftigte sich mit den sog. deliberative Agents oder auch „smart agents“, welche mit symbolischen, internen Modellen arbeiten („collaborative agents“). Ein „deliberative agent“ ist



ein Agent, welcher ein explizites, symbolisches Modell der Welt besitzt, und in dem Entscheidungen via „symbolischer Argumentation“ gefällt werden. Bei Arbeiten in dieser Richtung konzentrierte man sich auf die Zusammenarbeit und Kommunikation zwischen einzelnen Agenten, die Aufgabenverteilung, Koordination und Konfliktlösung durch Argumentation. Diese „Makro-Aspekte“, stellen die Gesellschaft der Agenten über den einzelnen, wohingegen „Mikro-Aspekte“ sich hauptsächlich um das Individuum kümmern. Zusätzlich zu den „Makro-Aspekten“, um die sich die erste der Forschungsrichtungen bemühte, kümmerte man sich ebenso um Architekturen, Agententheorien und Sprachen.

Seit etwa 1990 hat sich ein weiterer Forschungszweig bemerkbar gemacht, bei dem sich die Menge der Agententypen stark erweitert hat. Hinzugekommen ist hauptsächlich der Lernfaktor, der bei den „collaborative Agents“ bisher nicht so stark ausgeprägt war.

2 Motivation

2.1 Verteiltes Rechnen

Computer arbeiten heute auf einer extrem einfachen Basis zusammen. Sie sind zwar in der Lage, Informationen auszutauschen – z.B. per WWW, FTP oder anderen Diensten – aber noch nicht auf einer solchen Ebene, daß die Systeme wirklich zusammenarbeiten. Zusammenarbeiten in dem Sinne, daß mehrere Systeme gemeinsam ein Problem lösen. Agenten können sehr gut eingesetzt werden, um eine intelligente Zusammenarbeit über das Netzwerk zu organisieren. So müssen nicht nur die Fähigkeiten aller Systeme verwaltet werden, sondern auch deren Aufgabenverwaltung, der Speicher und andere Ressourcen. Statt die Arbeit nun einen Agenten alleine machen zu lassen, der dann schnell zum leistungsmäßigen Flaschenhals eines solchen verteilten Systems wird, kann man für jedes System einen Agenten als Ressourcenverwalter einsetzen. Ein solches System bietet auch die Möglichkeit, Services auf einem höheren Abstraktionsniveau anzubieten oder anzufordern als bisher – es werden die Absichten des Anwenders dargelegt und keine spezielle Implementierung mehr angefordert. Durch die Möglichkeit, Aufgaben abstrakt zu beschreiben, muß ein Programm beim Datenzugriff nicht mehr über alle Details der Daten auf dem Zielsystem Bescheid wissen – in welchem Format die Daten vorliegen, welche Pfadangabe und welches Protokoll nötig ist und ähnliches.

2.2 Verbessertes Interface

In den letzten Jahren ist die Kommandozeile praktisch überall durch grafische Interfaces – meistens auf Basis von Icons – abgelöst worden. Diese sind hervorragend geeignet, um einfache, simple Aufgaben schnell und übersichtlich durchführen zu können. Sobald die Aufgaben an Komplexität gewinnen, wird die Anwendung eines solchen Systems jedoch schnell unangenehm. Immer wiederkehrende Aufgaben, so z.B. das Durchsuchen sehr großer Datenmengen, sollten besser automatisiert werden. Herkömmliche Interfaces sind auch nicht in der Lage, ständig wiederkehrende Aufgaben zu erkennen und selbst zu übernehmen, geschweige denn Muster in der Arbeit des Anwenders zu erkennen und so Aufgaben mit seiner „Handschrift“ ausführen zu können. Traditionelle Interfaces sind funktionsorientiert und können keine Rücksicht auf die aktuelle Aufgabe des Benutzers nehmen.

Forscher versuchen, die Absichten des Benutzers durch einen „Indirect Management Style of Interaktion“ in die Maschine zu übertragen. Das Ziel ist es, daß ein Benutzer dem System die

grundlegenden Richtlinien für die Aufgabe darlegt und sich nicht um die Details zu kümmern hat – der Rechner soll „menschlicher“ werden. Es sollen Anfragen folgender Art möglich sein:

- „Hat Dr. Wolff auf meine Nachricht vom Freitag geantwortet?“
- „Gibt es neue Pressemitteilungen über ein bestimmtes Ereignis?“

Agenten haben einige Vorteile zu bieten, die solche Arbeiten weiter erleichtern können: Sie können sehr schnell sehr große Datenmengen verarbeiten oder erst zu einer bestimmten Zeit damit beginnen – z.B. wenn die Telefongebühren gering sind oder der Anwender einen anderen Termin hat. Agenten können alternative Lösungsmöglichkeiten finden, da sie das Problem kennen und nicht einfach nur eine simple Anweisung ausführen. Da Agenten lernfähig sind, können sie sich an wiederkehrende Aufgaben gut anpassen.

2.3 Gegenargumente

Die oben gemachten Angabe verleiten dazu, Agenten als ein extrem leistungsfähiges Hilfsmittel anzusehen, die zur Lösung von Problemen sehr viel beitragen können. Ben Shneiderman, der sich mit der Entwicklung von Interfaces beschäftigt, sieht allerdings auch einige mögliche Probleme mit „intelligenten“ Interfaces ([BRADS], Seite 22):

- Qualitäten wie Vorhersagbarkeit und Kontrolle sind sehr wünschenswert. Wenn Maschinen intelligent oder anpassungsfähig werden, könnten diese Eigenschaften schnell verlorengehen.
- Wenn der Benutzer überzeugt ist, mit einem intelligenten System zu arbeiten, hat er große Probleme, mit eventuellen Fehlern zurechtzukommen.
- [...]

Shneiderman weist außerdem darauf hin, daß bisherige Versuche, Systeme mit echter Sprache, computerunterstützter Unterricht und intelligente sprechende Roboter eine Enttäuschung waren, und man deshalb in näherer Zukunft nicht zuviel von Agenten erwarten sollte.

3 Grundlagen

3.1 Was ist ein Agent?

Diese Frage zu beantworten ist sehr schwer. Es gibt keine Eigenschaften, die ein System unbedingt haben muß, um ein Agent zu sein. Die Liste der möglichen Eigenschaften ist sehr lang (3.1siehe unten). Dazu kommt, daß inzwischen sehr viele Software-Konstruktionen „Agenten“ genannt werden, obwohl sie die Anforderungen an ein solches Systems nur sehr mangelhaft erfüllen. Der Begriff eines Agenten ist inzwischen schon fast zu einem „noise term“ geworden, also einem Begriff, der oft mißbraucht wird. In diesem Abschnitt werden verschiedene Typen von Definitionen aufgezeigt, mit der Agenten heute beschrieben werden.

Wer den Begriff „Agent“ zum ersten Mal hört und statt einer komplexen Definition ein einfaches Beispiel sucht, findet hier einige kleine Beispiel für die verschiedensten Typen von Agenten, mit denen wir in Zukunft evtl. in Berührung kommen.

- **E-Mail-Agent**
Nachdem man sich am System angemeldet hat, wird man sofort vom persönlichem E-Mail-Agenten auf eine wichtige Nachricht aufmerksam gemacht. Durch Beobachtung hat der Agent mit der Zeit gelernt, welche Nachrichten der Benutzer als besonders wichtig betrachtet. Weiß dies der Agent noch nicht, so kann er evtl. bei anderen Agenten nachfragen, welche Nachrichten andere Nutzer als besonders wichtig betrachten. So können z.B. alle Agenten darauf hinweisen, daß Nachrichten vom Abteilungsleiter als besonders wichtig eingestuft werden. Da dies die Aussage vieler Agenten ist, kann unser persönlicher Agent daraus schließen, daß ein neuer Benutzer eine Nachricht vom Abteilungsleiter ebenfalls als wichtig betrachten wird.
- **Börsen-Agent**
Ein etwas fiktiveres Beispiel ist ein Agent, der an unserer Stelle an der Börse handeln kann. Das System bietet einerseits beratende Funktionen, so kann es z.B. den Markt und verschiedene Foren betrachten und daraus zu einem Schluß kommen, welche Anlagen momentan am vielversprechendsten sind, oder welche Aktien man möglichst schnell abstoßen sollte. Auf der anderen Seite wäre der Agent auch in der Lage, für uns zu handeln: So könnte der Agent automatisch Aktien verkaufen, wenn deren Wert unter eine bestimmten Wert sinkt. Prinzipiell wäre es sogar möglich, daß der Agent komplett selbstständig handelt.

3.2 Eigenschaften von Agenten

Die Definition über solche Beispiel ist natürlich alles andere als präzise. Eine bessere Möglichkeit, einen Agenten darzustellen, erhalten wir über die Liste der verschiedenen Eigenschaften, die ein Agentensystem besitzen kann ([BRADS], Seite 8).

Es folgt eine Liste von Eigenschaften, die man typischerweise mit Agenten in Verbindung bringt. Ein Agent muß – je nach Definition – keine dieser Eigenschaften zwingend besitzen, aber sie stellen eine gute erste Richtlinie dar um abzuschätzen zu können, was Agentensysteme leisten können.

- **Autonomie**
Ein Agent ist in der Lage selbsttätig zu agieren, vor allem aber hat er Kontrolle über seine Aktionen und Eigenschaften.
 - **Reaktivität**
Agenten sind in der Lage, ihre Welt wahrzunehmen – wobei die Welt die „echte“, physikalische Welt sein kann, es kann sich aber auch um eine Menge anderer Agenten, das Internet oder andere Welten handeln – und in einem zweitleitigen Rahmen direkt darauf zu reagieren. Reaktivität ist eine besondere Art der Autonomie, da der Agent nicht wirklich selbsttätig handelt, sondern durch eine Veränderung der Umgebung „angestoßen“ wird.
 - **„Pro-activeness“**
„Pro-Activeness“ oder Proaktivität bedeutet, daß Agenten nicht einfach nur auf Änderungen in ihrer Umgebung reagieren, sondern ebenso in der Lage sind, die Initiative zu ergreifen um ihr Ziel zu erreichen.
-

- **Mobilität**
Agenten können sich in einem Netzwerk selbstständig bewegen und zur Steigerung der Effektivität auch auf einem anderen System arbeiten. Hat unser Agent z.B. eine große Menge von Daten zu bewältigen, so ist es nicht sinnvoll, alle Daten auf das eigene System zu laden und sie von dort aus durchzuarbeiten. Statt dessen begibt sich der Agent direkt zum anderen System und arbeitet dort.
- **Soziale Fähigkeiten**
Agenten interagieren mit anderen Agenten (und eventuell auch Menschen) auch unterschiedliche Weise. Agenten sollen in der Lage sein, Informationen auszutauschen. So kann ein Agent z.B. auf das schon vorhandene Wissen anderer Agenten zurückgreifen.
- **Zusammenarbeit**
Agenten sollten zusammenarbeiten können um ein gemeinsames Ziel zu erreichen.
- **„Knowledge-Level“ Kommunikations-Fähigkeiten**
Die Fähigkeit mit Menschen und Agenten in einer Sprache zu kommunizieren, die eher der menschlichen Sprache entspricht anstatt einer am Programm orientierten. Durch diese Eigenschaft ist es möglich, Aufgaben in einer Weise zu beschreiben, die sich nicht an Implementierungsdetails sondern an der Aufgabe selbst orientiert. Durch diese Eigenschaft ist ein Agent in der Lage alternative Lösungswege zu finden, falls der evtl. vorgegebene einmal nicht zum gewünschten Ziel führt.
- **Fähigkeit zur Folgerung**
Ein Agent sollte basierend auf einer abstrakten Aufgabenstellung mit Hilfe von vorher angesammeltem Wissen und bevorzugten Methoden arbeiten können, um flexibler zu werden. Das bedeutet einfach, daß Agenten lernfähig sein sollen.
- **Kontinuität**
Der Agent behält seine Identität und sein Wissen über längere Zeit. Sehr nützlich natürlich in Verbindung mit der Lernfähigkeit, da der Agent einmal angesammeltes Wissen wiederverwenden soll.
- **Persönlichkeit**
Die Fähigkeit, bestimmte Eigenschaften und Zustände als für den Nutzer sichtbare Emotionen darzustellen. Der Agent kann evtl. als ein menschliches Gesicht dargestellt werden, welches die entsprechenden Emotionen darstellt. Diese Art von Interface ist für den Benutzer leichter verständlich.

Wie oben schon angedeutet, ist keine dieser Eigenschaften in einem Agenten zwingend notwendig. Ein System, welches nicht lernfähig ist, kann immer noch ein Agent sein. Damit stellt sich erneut die Frage nach einer sinnvollen Definition. Eine alternative Möglichkeit zur Beschreibung kommt aus dem Bereich der künstlichen Intelligenz. Es geht darum einen Agenten durch abstraktere Begriffe wie „Wunsch“ und „Glauben“ zu definieren.

3.3 Definition über Belief, Desire, Intention

Die Definition über „Belief“, „Desire“ und „Intention“ erklärt einen Agenten über Begriffe, die normalerweise für die Beschreibung von Menschen benutzt werden. Beschreibungen dieser Art sind für jeden klar verständlich:

- Jürgen glaubt, daß sein Tank bald leer ist.
- Jürgen fährt zur Tankstelle, weil er tanken will.

Die Meinungen, die mit dieser, unter den Menschen weit verbreiteten „Notation“ ausgedrückt werden, heißen Intentional Notions. Der Philosoph Daniel Dennet hat den Begriff Intentional System für die Begriffe Glaube, Wünsche und rationales Denken geprägt, um ein Wesen oder eine Persönlichkeit zu beschreiben. Dennet unterscheidet verschiedene Stufen eines Intentional Systems:

„Ein Intentional System erster Stufe hat Glauben und Wünsche (etc.), aber keinen Glauben oder Wünsche über Glaube und Wünsche. ... ein Intentional System zweiter Stufe ist weiter fortgeschritten; es hat Glauben und Wünsche (und keinen Zweifel über diesen internen Zustand) über Glauben und Wünsche (und andere ‚intentional states‘) – sowohl über seine eigenen als auch über die anderer.“

Diese Definition kann man auf eine so hohe Stufe bringen, wie man gerne möchte.

Für unserer Zwecke wollen wir Belief, Desire und Intention ein wenig genauer definieren:

- Belief bezeichnet das momentane Wissen des Agenten, stellt also seinen Glauben dar.
- Desires sind die Ziele des Agenten. Desires stellen die Motivation des Agenten dar Ziele zu erreichen, die ihm vom Benutzer vorgegeben wurden.
- Intention stellt eine Art momentaner Absichtserklärung dar, eine Art Kurzstreckenziel. Diese zweite Definition eines Zieles ist deswegen nötig, weil die ursprünglichen Ziele des Agenten aufgrund bestimmter Änderungen der Umgebung evtl. nicht mehr genau zu erfüllen sind. Um aber herauszufinden, ob sich bestimmte Änderungen in der Umgebung auf die Ziele unseres Agenten auswirken, müßte man nach jeder noch so kleinen Änderung das gesamte Wissen des Agenten konsultieren um herauszufinden, ob die Ziele eben noch erreichbar sind. Dies zieht einen sehr großen Arbeitsaufwand mit sich, der das zeitliche Verhalten extrem schlecht werden läßt. Man rettet sich mit folgender Annahme aus der Situation: Man nimmt an, daß sich nur große Änderungen der Umgebung auf unsere Ziele auswirken und muß dadurch entsprechend weniger überprüfen. Die Absichten zwischen diesen Überprüfungen stellen dann die Intentions des Agenten dar.

Die Definition dieser Begriffe ist deswegen so wichtig, weil die „deliberative Agents“, die einen klassischen Ansatz aus der künstlichen Intelligenz darstellen, mit Hilfe von Belief, Desire und Intention definiert werden. Man spricht von den sogenannten „BDI-Agenten“ (für weitere Informationen siehe [RAOG]). Die deliberative Agents stellen einen von vier grundlegenden Ansätzen dar, einen Agenten zu konstruieren (siehe auch 4.5Abschnitt Architekturen).

Es stellt sich nun die Frage, ob es sinnvoll und legitim ist, Software mit solchen Begriffen wie Glaube und Wünschen zu beschreiben. McCarthy sagt zu dieser Frage, daß es durchaus einen Sinn macht diese Begriffe zur Beschreibung eines Agenten zu verwenden, solange man daraus einen Nutzen zieht. Natürlich könnte man auch den Programmcode betrachten, um das Verhalten des Agenten so zu bestimmen. Ab einer gewissen Komplexität ist dies aber nicht mehr praktikabel – und ab dieser Grenze wird die neue Beschreibungsart sehr nützlich. Man kann diese Art der Beschreibung auch vorher einsetzen – das Beispiel der Beschreibung eines Lichtschalters als sehr kooperativer Agent wird häufig benutzt – aber es ist leicht zu

erkennen, daß uns die neue Form der Beschreibung hier keinen Nutzen bringt. Die Frage der Vollständigkeit einer Definition unserer Agenten über die intentional states läßt sich heute leider noch nicht beantworten. In seiner Master's thesis beschäftigte sich Shardlow mit dieser Frage und kam zu folgendem Schluß:

„Vielleicht ist ein Agent mehr, als sich mit Glaube und Wünschen beschreiben läßt, was aber auch immer es ist, es entzieht sich der momentanen Kenntnis der Wissenschaft“. (Übersetzung?)

Versucht man nun, das System der Intentional Notions in ein logisches System zu fassen ([WOOLD], Seite 7), tauchen schnell Probleme auf. Die Operatoren der klassischen Logik basieren auf Wahrheitswerten, die Intentional Notions jedoch nicht. So ist der Wahrheitswert des Satzes

Marc glaubt p

nicht allein von p abhängig: Marc glaubt p , egal ob p nun wahr ist oder nicht. Es werden also andere Formalismen benötigt. Die Menge an Techniken, die dieses Problem angehen ist jedoch relativ klein. Zwei Problematiken sind zu bearbeiten: Die syntaktische und die semantische.

Der bekannteste Ansatz für das semantische Problem besteht in der Verwendung der Semantik der möglichen Welten fassen ([WOOLD], Seite 12), um den Glauben eines Agenten zu charakterisieren. Dies wird am Beispiel eines Kartenspiels deutlich: Um herauszufinden, welche Karten ein bestimmter Mitspieler z.B. beim Poker hat, kann sich unser Agent nur am eigenen Blatt orientieren. Besitzt er z.B. Pik-As, so entspricht die Menge der Welten, in denen das Blatt des Gegners beschrieben wird, allen möglichen Kombinationen ohne das Pik-As. Dieses Modell hat den Vorteil, daß es der Wissensstruktur des Agenten keine Form aufzwingt, und daß die mathematische Theorie in Zusammenhang mit der Menge der möglichen Welten sehr interessant ist. Um diese Welten miteinander zu verknüpfen, wird die bisherige Sprache der Logik um zwei Begriffe erweitert: „nötigerweise“ und „möglichweise“. Diese Begriffe definieren, welche Welten von einer bestimmten aus erreichbar sind. Die Formel „nötigerweise p “ ist dann wahr, wenn p in jeder von unserer vorgegebenen Welt erreichbaren Welt wahr ist, „möglichweise p “ ist dann wahr, wenn p zumindest in einer erreichbaren Welt wahr ist.

Modale Logik, die auf dem obengenannten System aufbaut, besitzt leider zwei grundlegende Probleme, welche zusammen das berühmte „logical omniscience problem“ bilden. Zum einen haben wir die „necessiation rule“ – wenn p gültig ist, so ist auch „nötigerweise“ p gültig – die uns die Aussage liefert, daß unser Agent alle gültigen Formeln kennt, und daraus folgend auch alle möglichen Tautologien. Da es davon aber unendlich viele gibt, „weiß“ unser Agent unendlich viel, was nicht sehr einleuchtend klingt. Zum anderen sagt uns Axiom K – „nötigerweise“ $p \supseteq p$ – daß das Wissen des Agenten unter Implikation abgeschlossen ist, was zusammen mit der „necessiation rule“ bedeutet, daß das Wissen des Agenten unter logischer Konsequenz abgeschlossen ist, der Agent glaubt also alle logischen Konsequenzen seines Wissens, was ebenso nicht sehr verständlich ist: Falls ein Agent z.B. die Peano Axiome kennt, so muß er daraus folgernd auch Fermats „last theorem“ kennen, da dieses aus den Peano Axiomen folgt. Um dies zu beweisen, waren allerdings einige der führenden Köpfe des letzten Jahrhunderts nötig.

Wegen der Schwierigkeiten des „logical omniscience problems“ haben viele Forscher nach alternativer Möglichkeiten gesucht, einen Formalismus für den Glauben der Agenten zu finden. So gibt es z.B. die Möglichkeit, zwischen explizitem und implizitem Wissen zu unterscheiden – ein Agent kennt einige wenige explizite Wissensfragmente, und unterscheidet von diesen die daraus folgenden impliziten Folgerungen. Wie man sieht ist allein schon eine Theorie über die Repräsentation des Glaubens eines Agenten sehr schwer zu formulieren.

3.4 Typologisierung von Agenten

Hyacinth S. Nwana hat eine Typologie für Agenten entwickelt – eine Möglichkeit, Agenten in verschiedene Klassen einzuordnen ([NWANA], Seite 6). So ist es möglich, den sehr umfassenden Begriff „Agent“ etwas genauer zu unterteilen.

Zuerst einmal werden Agenten nach ihrer Mobilität eingeteilt: Wir unterscheiden zwischen statischen und mobilen Agenten. Zweitens klassifizieren wir Agenten als entweder „deliberative“ oder „reaktiv“. Agenten, die „deliberative“ sind, werden vom „deliberative thinking paradigm“ abgeleitet: Sie besitzen ein internes symbolisches Argumentationsmodell und versuchen ein Ziel durch Planung und Verhandlung mit anderen Agenten zu erreichen. Reaktive Agenten hingegen agieren durch Ereignis und Antwort – sie reagieren je nach Zustand der Umgebung in der sie sich befinden. Zum dritten werden Agenten nach verschiedenen Attributen eingeteilt, die ein Agent haben sollte. Das sind mindestens folgende drei: Autonomie, Lernfähigkeit und Kooperation. Wir benutzen diese Attribute um vier Agententypen zu definieren: Collaborative agents, Collaborative learning agents, Interface agents und Smart Agents.

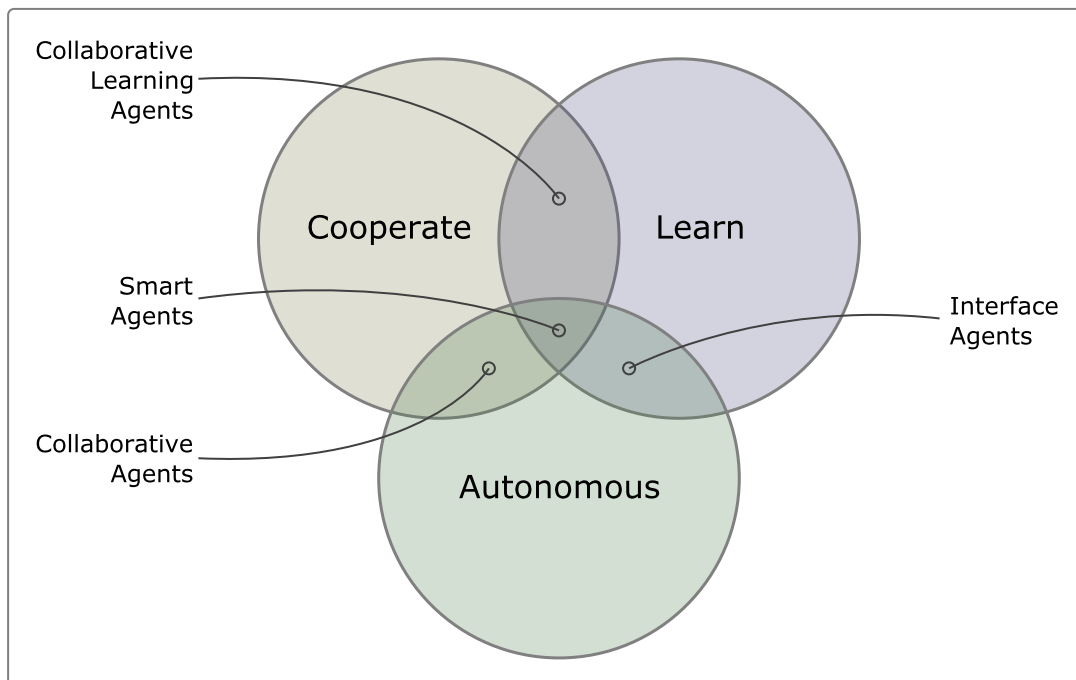


Abbildung 2: Die Agentenklassen als Mengendarstellung

Es ist zu beachten, daß die in Abbildung 2 definierte Einteilung nicht definitiv ist. Das bedeutet, daß ein Collaborative Agent durchaus Lernfähigkeiten haben kann, daß diese dann aber im Vergleich zu Autonomie und Kooperationsfähigkeit nicht besonders stark ausgeprägt sind. Wichtig ist außerdem, daß nach diesem Modell alle Systeme, die außerhalb einer „Schnittmenge“ liegen, keine Agenten sind – jeder Agent sollte über ein gewisses Maß an Kooperationsfähigkeit, Lernfähigkeit und Autonomie verfügen, aber dies ist momentan eher ein Wunsch als die Realität. Der vierte Punkt, nach dem Agenten unterschieden werden könnte, ist durch ihre Rollen gegeben, so z.B. World-Wide-Web-Agenten oder Informationsagenten. Zuletzt ist es auch möglich, daß ein Agent nach mehr als einer Philosophie entwickelt wurde, in diesem Fall handelt es sich um einen hybriden Agenten.

Es gibt noch einige andere Attribute, nach denen sich Agenten noch weiter einstufen lassen, etwa die Frage, wie vielseitig der Agent ist, wie hilfreich er ist, ob er nur wahrheitsgemäße Aussagen macht usw.

3.5 Agenten als Vertreter

Befragt man das American Heritage Dictionary nach dem Begriff „Agent“, wird ein Agent als „jemand der in Vertretung eines anderen handelt oder die Befugnis zum Handeln hat“, bzw. „Maßnahme durch die etwas getan oder verursacht wird, auch Instrument“ definiert. Vor allem die Bezeichnung „in Vertretung eines anderen handeln“ zeigt den zentralen Sinn des Agentendaseins, den Nicholas Negroponte auch mit dem Begriff „sister-in-law“ bezeichnet: Wir erwarten von jedem Agent, daß er in Vertretung eines anderen – meistens für den Anwender selbst – handelt. Egal, ob nun Post sortiert, interessante Texte aus dem Web geholt, Einblick in gespeicherte Korrespondenz genommen oder auch nur ein Hinweis auf einen rapide sinkenden Börsenkurs gegeben werden soll, alle diese Handlungen werden für einen bestimmten Benutzer ausgeführt, ob nun direkt oder indirekt. Der Agent weiß durch eine entsprechende Anpassung, was den Benutzer interessiert, und ist grade deswegen in der Lage, Arbeit für ihn zu verrichten. Der Agent ist in der Lage, den Benutzer zu vertreten, ihn zu „modellieren“.

4 Theorien

4.1 Überblick

Eine Agententheorie hat die Aufgabe, alle Teilaspekte eines Agenten in ein logisches System zu fassen. Da schon die einzelnen Teile beträchtliche Probleme aufwerfen, ist es nicht verwunderlich, daß bis heute noch keine allumfassende Agententheorie entwickelt worden ist. Das liegt unter anderem daran, daß eine Agententheorie alle Aspekte eines Agenten kombinieren muß – Glaube und Bewußtsein, Ziele und Wünsche, und evtl. noch einiges, was unbekannt ist. Außerdem müssen die dynamischen Aspekte des Agenten berücksichtigt werden.

Obwohl wir von einem vollständigen System noch weit entfernt sind, wurden schon erste signifikante Schritte in die richtige Richtung gemacht ([WOOLD], Seite 17).

4.2 Moore – Knowledge and Action

Moore beschäftigte sich mit der Frage, was ein Agent wissen muß, um zu handeln – der „pre-conditions for actions“. Er formalisierte das Modell der Fähigkeit und ein dynamisches System zur Modellierung von Aktionen. Dieser Formalismus erlaubt es einem Agenten, der nur wenig Informationen über ein Ziel hat, bestimmte Aktionen um mehr über das Ziel herauszufinden.

4.3 Cohen and Levesque – Intention

Eine der bestbekanntesten und einflußreichsten Theorien stammt von Cohen und Levesque. Obwohl sie ursprünglich eine Theorie über Absichten formulieren wollten, wird sie hauptsächlich in der Konfliktanalyse und Kooperation in Multiagentensystem verwendet.

4.4 Rao and Georgeff

Cohen und Levesque benutzten nur zwei geistige Eigenschaften für ihr Modell: Glaube und Ziele, weitere Eigenschaften, wie etwa Absichten, wurden über Glaube und Ziele definiert. Rao und Georgeff erarbeiteten ein Modell, welches auf drei geistigen Eigenschaften aufbaut: Glaube, Wünsche und Absichten. Ihr Formalismus basiert auf einem System, welches nach der Zeit strukturiert ist.

4.5 Wooldridge

Wooldridge entwickelte eine Familie von logischen Systemen, welche die Eigenschaften von Multiagentensystemen repräsentieren. Sein Ziel war allerdings kein allumfassendes Modell, sondern einige Formalismen zu schaffen, die in der Spezifikation und Verifikation realistischer Multiagentensysteme nützlich sind.

5 Agentenarchitekturen

5.1 Was sind Agentenarchitekturen

Pattie Maes definiert ([WOOLD], Seite 23) eine Agentenarchitektur folgendermaßen:

„[Eine] bestimmte Methode [Agenten] zu bauen. Sie spezifizieren wie ... der Agent in die Konstruktion aus modularen Komponenten und der Zusammenarbeit dieser Teile zerlegt werden kann. Zusammen bilden die Komponenten und ihre Interaktionen die Antwort auf die Frage, wie die sensorischen Daten und der momentane interne Zustand des Agenten seine Aktionen ... und den zukünftigen internen Zustand des Agenten bestimmen. Die Architektur umfasst Techniken und Algorithmen die diese Methodik unterstützen.“

Kaelbling beschreibt den Begriff als

„Bestimmte Software- (oder Hardware-) Module, die normalerweise mit Hilfe von Boxen und Pfeilen, welche den Datenfluß und die Kontrollmechanismen zwischen den Modulen widerspiegeln.“

5.2 Reaktive Architekturen

Reaktive Systeme sind sehr viel einfacher aufgebaut als Architekturen, die auf symbolischer künstlicher Intelligenz basieren. Ein reaktives System verzichtet auf jede Art eines symbolischen Modells der Welt und komplexe symbolische Argumentation. Neuere Theorien besagen, daß intelligentes Verhalten auch ohne explizite Repräsentation der Art, die symbolische KI benutzt, erzeugt werden kann. Intelligenz entsteht durch bestimmte, komplexe Systeme, sie ist das Ergebnis der Interaktion mit der Umgebung, keine eigne, isolierte Fähigkeit. Brooks, der diese Thesen aufstellte, untermauerte sie mit der Konstruktion einfacher Roboter, deren Leistung für symbolische KI-Systeme beachtlich wäre.

5.3 Brook's Subsumption Architecture

Eine der bekanntesten reaktiven Architekturen stellt Brook's subsumption architecture dar [NWANA], Seite 31. Die Idee besteht darin, ein System aus verschiedenen Schichten aufzubauen, die um die Kontrolle über das Gesamtsystem konkurrieren. Falls ein solcher Agent z.B. einen Roboter kontrolliert der seine Umgebung erforschen soll, so enthält die „oberste“ Schicht den „Gesamtfahrplan“ des Roboters, die unterste beschäftigt sich dagegen z.B. mit dem Ausweichen von Hindernissen.

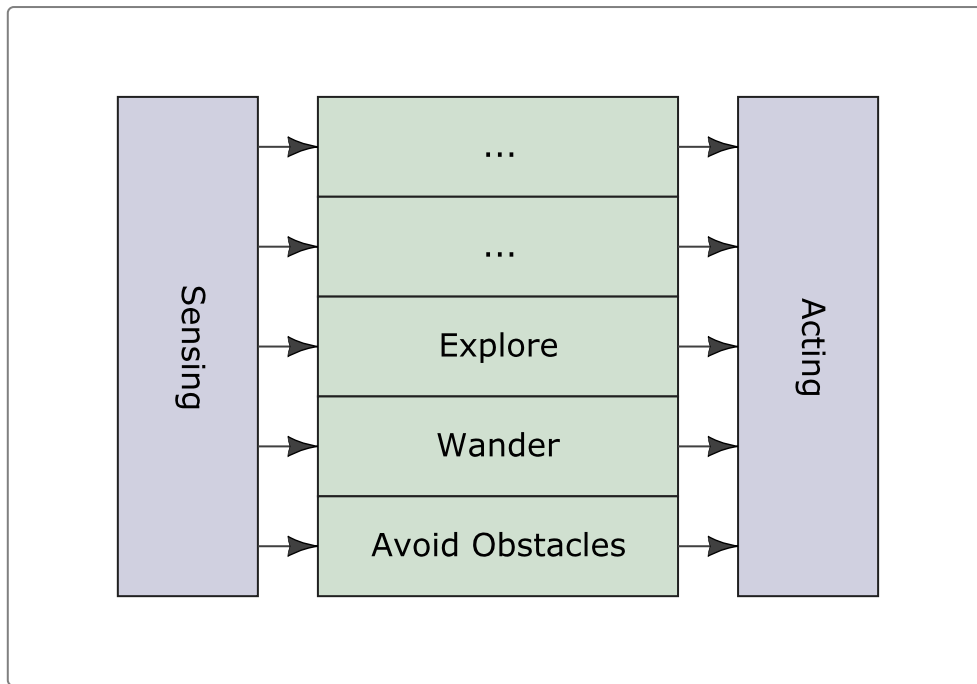


Abbildung 3: Brook's Subsumption Modell

5.4 Planning Agents

Planen bedeutet im Endeffekt automatisches Programmieren: Es wird eine Reihe von Handlungen erstellt, die, wenn sie ausgeführt werden, zum Erreichen eines bestimmten Zieles führen. Das bekannteste planende System namens STRIPS – von Fikes and Nilsson – baute auf einem sehr einfachen Algorithmus auf und hatte Schwierigkeiten schon Probleme mittlerer Komplexität zu lösen. Es wurden neue Gedanken in das Thema eingebracht und bereits verschiedene Versuche unternommen Agenten mit einer planenden Komponente als zentralem Bestandteil zu konstruieren. Allerdings entstand Mitte der achtziger Jahre eine Theorie, nach der planende Agenten sich als zweitmäßig ineffizient erweisen werden. Dies führte zu einem völlig neuen Ansatz: den reaktiven Architekturen.

5.5 Deliberative Agents

Der Begriff „deliberative agent“ ist gleichbedeutend mit einem sog. „sentential processing automaton“. Ein solcher Automat basiert auf physischen Symbolen, welche zusammengenommen Strukturen definieren und darauf Prozesse ablaufen lassen können. Wir definieren einen deliberative agent als einen Agenten, der ein symbolisches Modell der Welt besitzt und in dem Entscheidungen durch logische – und zumindest pseudo-logische Argumentation getroffen werden.

5.6 Hybride Architekturen

Viele Forscher sind der Meinung, daß keine der bisher genannten Methoden allein zu einem guten Ansatz führt. Sie vertreten den Standpunkt, daß die optimale Lösung eine Mischung aus den klassischen und den alternativen Ansätzen besteht.

Die einfachste Art, diese Methode anzuwenden, besteht in der Konstruktion eines Systems, welches aus zwei Teilen besteht: Eine „deliberative“ und eine reaktive Komponente. Der deliberative Teil plant und trifft Entscheidungen auf die schon bekannte Art, der reaktive Teil ist in der Lage, ohne komplexere Vorgehensweisen auf Änderungen in seiner Umgebung zu reagieren. Durch eine hierarchische Anordnung der einzelnen Komponenten kann man den Agenten an verschiedene Anforderungen anpassen. Es ist z.B. möglich, der reaktiven Komponente eine höhere Priorität zuzuordnen und damit eine höhere „Reaktionsgeschwindigkeit“ des Systems zu erzielen, wohingegen die planende Komponente mehr im Hintergrund läuft und Langzeitziele plant.

5.7 InteRRap Architecture

Die InteRRap Architektur [NWANA], Seite 34 ist ein Beispiel für eine Hybride Agentenstruktur. Abbildung 4 zeigt eine schematische Darstellung dieser Architektur. Wichtig für uns ist die Einteilung des Systems in drei Schichten: Die „Behaviour-based“ Layer, die „Local Planning Layer“ und die „Cooperative Planning Layer“. Um den Agenten möglichst leistungsfähig zu gestalten ist die Behaviour-based Layer reaktiv aufgebaut, da dies die „unterste“ Schicht ist, die am „nächsten“ an der Welt liegt. Die Implementierung als reaktive Architektur erlaubt ein schnelles Reagieren auf Änderungen in der Umgebung des Agenten. Dagegen ist die Cooperative Planning Layer „delibeartive“ aufgebaut, da auf dieser höheren Ebene keine schnelle Reaktionszeit mehr gefragt ist, und die Vorteile der deliberative Architektur gewinnbringend ausgespielt werden können.

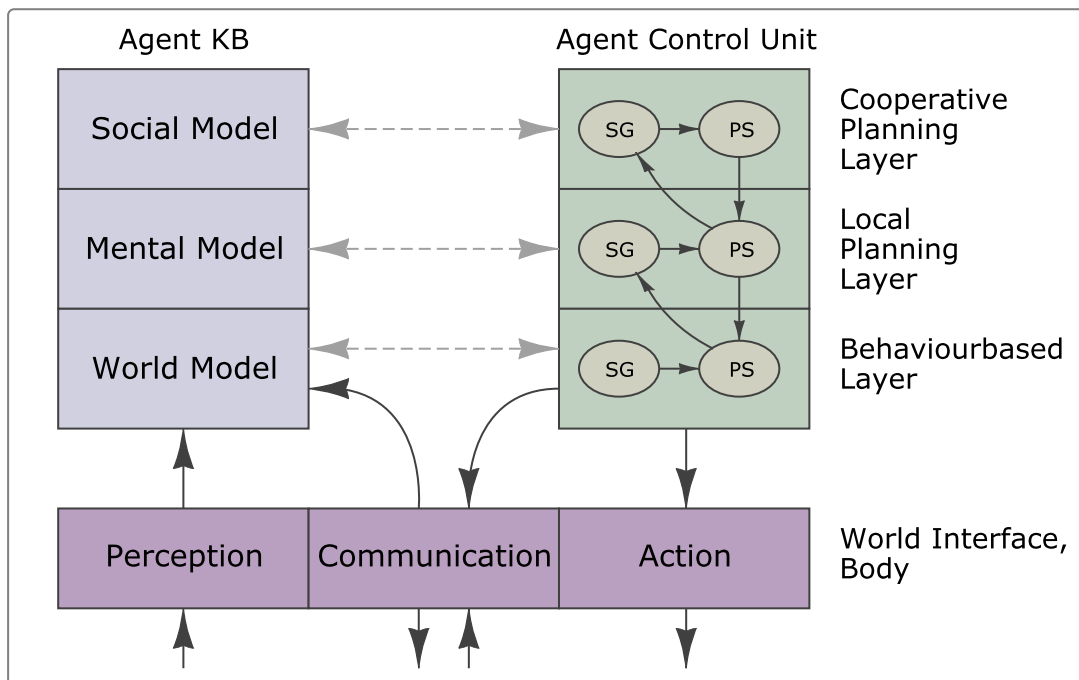


Abbildung 4: InteRRap Hybrid Architecture

6 Die weitere Entwicklung

6.1 Forschende Organisationen

Die Liste der sich mit Agenten beschäftigenden Organisationen ist ziemlich groß. Hier ein kleiner Auszug aus der Liste der größere Organisationen:

- Alcatel
- Apple
- AT&T
- BT
- Carnegie Mellon University (CMU)
- Daimler-Benz
- DEC
- Genral Magic
- HP
- IBM
- Lotus
- Mssachusettes Insitute of Technology (MIT)
- Microsoft
- Oracle
- Sharp
- University of London

6.2 Zu lösende Probleme

Neue Technologien bringen auch immer neue Probleme mit sich. Hier seien einige genannt, welche mit der besonderen Bedeutung und Vertretungsfunktion der Agenten zusammenhängen und deswegen besonders interessant sind:

- Privatsphäre
Vor allem wenn die Mobilität genutzt wird ergibt sich das Problem die Privatsphäre zu wahren. Da die Agenten Abfragen an Systeme stellen und mit allen releveanten Informationen des Benutzers unterwegs sind ist es durchaus möglich, daß nicht nur die Abfragen und Aktionen unseres Agenten protokolliert werden um so Informationen über den Benutzer zu erlangen, sondern daß evtl. auch Systeme, auf denen mobile Agenten laufen, so manipuliert sind, daß das Gedächtnis des Agenten vollständig ausgelesen wird.
-

- Verantwortlichkeit

Wir möchten gerne, daß ein Agent für uns handelt. So ist es in Anbetracht der Versuche mit „electronic cash“ gar nicht so abwegig, daß ein Agent den Benutzer falsch „versteht“, und z.B. statt nur Informationen über ein neues Auto zu organisieren, gleich ein neues Auto im Namen des Benutzers kauft. Wenn der Benutzer den Agenten entsprechend autorisiert hat, da der Agent z.B. auch in seinem Namen an der Börse handelt und deswegen die vollen Rechte besitzt, mag dieser Kauf absolut rechtskräftig sein.

- Rechtliche Probleme

Sollte ein Agent einmal eine Fehlfunktion aufweisen, und es entstehen daraus z.B. finanzielle Verluste – wer ist dafür verantwortlich zu machen? Der Benutzer, weil er eine Aufgabe vielleicht zu undeutlich beschrieben hat, oder der Hersteller, weil das Programm evtl. eine Fehlfunktion aufweist?

- Sonstiges

Viele andere Probleme sind zu lösen: Wie wird z.B. der Ressourcenverbrauch auf fremden Systemen geregelt, müssen Agenten „ehrlich“ gegenüber anderen sein und vieles mehr.

6.3 Mögliche Agentenanwendungen

Zwei Möglichkeiten die Agententechnologie anzuwenden, seien hier ganz kurz angerissen. Ein Konzept ist der „Remembrance-Agent“, ein System, welches auf einem tragbaren Computer läuft und dazu gedacht ist, daß man es immer bei sich trägt. Nach einer gewissen „Einarbeitungszeit“ ist das System in der Lage, Terminkalender, Notizblock und vieles andere zu ersetzen. Wenn das System mit den entsprechenden Sensoren ausgestattet ist, kann es einem auch durchaus Informationen der Art geben: „Mit der Person, die grade auf Dich zukommt, hattest Du am Mittwoch vergangener Woche ein Gespräch. Es ging um folgende Punkte...“. Eine andere Möglichkeit ist die Verknüpfung bestehender Flugsicherheitseinrichtungen. Da das Management der bestehenden Systeme auf diesem Gebiet momentan schon sehr komplex ist, erhofft man sich von der Agententechnologie einen Fortschritt. Falls z.B. ein System ausfällt, sollte die Lücke an Informationen die dadurch entsteht, von anderen Systemen ausgefüllt werden.

Literatur

- [WOOLD] Wooldridge, M. und Jennings, N.: „Intelligent Agents: Theory and Practice“, The Knowledge Engineering Review 10
<http://www.cs.umbc.edu/agents/introduction/ker.ps>
- [NWANA] Nwana, Hyacinth: „Software Agents: An Overview“, The Knowledge Engineering Review, Vol. 11, No 3
<http://agents.umbc.edu/introduction/ao.ps>
- [BRADS] Bradshaw, J.: „An Introduction to Software Agents“, AAAI Press/The MIT Press, ISBN: 0-262-52234-9
<http://aaai.org/Press/Books/Bradshaw/bradshaw.html>
- [RAOG] Rao A. und Georgeff M.: „BDI Agents: From Theory to Practice“, Tech. Rep. 56, Australian Artificial Intelligence Institute, Melbourne, Australia, Apr 1995
<http://agents.umbc.edu/introduction/rao.ps>
-