

# Intelligente Software-Agenten

Reinhard Kuntz

Seminar Kognitionswissenschaft und Intelligente Agenten

09.05.05

Das Gebiet der intelligenten Software-Agenten ist momentan sehr populär. Deshalb erweist es sich als sehr dynamisch und schwer abzugrenzen. Das große Interesse an diesem Gebiet begründet sich hauptsächlich darin, dass viele unterschiedliche Aspekte der Informatik insbesondere der KI zusammengeführt werden. Außerdem beinhaltet das Gebiet der intelligenten Software-Agenten beispielsweise die Wiederaufnahme der frühen Idee des allgemeinen Problemlösers aus der KI (General Problem Solver, Newell/Shaw/Simon 1957). Im Bereich des Software Engineering werden zum Teil Software-Agenten auch als natürliche Weiterentwicklung der objektorientierten Programmierung betrachtet. Das agentenorientierte Software Engineering (AOSE) sieht in Agenten ein auf das objektorientierte Paradigma aufsetzendes neues Programmierparadigma [Weiß]. Genauso breit wie die hinter dem Begriff des Software-Agenten stehenden Gedankenmodelle sind auch die Einsatzmöglichkeiten gefächert. Der Einsatz von (intelligenten) Software-Agenten wird in den Bereichen eCommerce, Informationsrecherche und Simulation, sowie für diverse Routineaufgaben verwendet [Wikipedia]. Im Bereich der Simulation sind insbesondere Multiagentensysteme von Interesse. Dies zeigt sich beispielsweise bei der Spieltheorie und bei Wirtschaftsraumsimulationen. In jüngerer Zeit lässt sich auch im Bereich der Fertigung die Verwendung von Multiagentensystemen für die Planung und Durchführung beobachten, wobei insbesondere die automatische Reaktionsfähigkeit der Systeme bei unvorhersehbaren Ereignissen von Interesse ist.

In dieser Seminararbeit soll ein einführender Überblick über das Gebiet der intelligenten Software-Agenten gegeben werden. Nach einem schrittwei-

sen Definitionsversuch des Begriffs 'intelligente Software-Agenten' werden der Begriff 'Umwelt' (bezogen auf Agenten), sowie weitere Grundlagen vorgestellt. Darauf folgend wird auf Agentenarchitekturen eingegangen. Nach einer Betrachtung einiger Anwendungsbeispiele wird diese Arbeit abgerundet durch Erörtern von Chancen und Risiken intelligenter Software-Agenten.

## 1 Abgrenzung / Definition

Zu Beginn stellt sich die Frage, was eigentlich ein intelligenter Software-Agent ist. Dies zu beantworten erweist sich als schwierig. Die Definitionen in der Fachliteratur gehen erheblich auseinander. Hinzu kommt, dass seitens der Softwareindustrie, derzeit der Trend besteht zu Marketinggründen jedes 'neuartige' Programm als Agent zu bezeichnen [Kwast, 1999]. In diesem Abschnitt soll deshalb versucht werden ausgehend vom Begriff des Agenten hin zum Begriff des intelligenten Software-Agenten eine beschreibende Definition zu bilden. Zu beachten ist dabei, dass diese Vorgehensweise eine schrittweise Spezialisierung darstellt.

### 1.1 Agent

Schon der Begriff des Agenten wird unterschiedlich verstanden. In [Russel/Norvig, 2004] findet sich die folgende recht abstrakte Definition:

„Als Agent kann alles angesehen werden, was seine Umwelt durch Sensoren (sensors) wahrnimmt und durch Effektoren (effectors) beeinflusst.“

Eine eher umgangssprachliche Interpretation

des Begriffs Agent hingegen findet sich in [Burkard, 2002]:

”Ein Agent ist jemand, der in fremdem Auftrag selbständig handelt.”

Betrachtet man diese Definition etwas allgemeiner kommt man zur folgenden Definition für Agenten [Gilgen, 1995]:

”Eine Einheit, die autorisiert anstelle einer anderen handelt, entweder als Repräsentant ... oder als ein Instrument, durch das eine führende Intelligenz ein Ergebnis erreicht.”

Werden diese drei aufgeführten Definitionen zusammengeführt, sind die wichtigsten Eigenschaften des Agenten-Konzept klar. Es beinhaltet das Wahrnehmen und das Beeinflussen der Umwelt (siehe 2.1) durch eine Einheit, die selbständig, aber im Auftrag einer anderen Einheit agiert, bis ein oder mehrere Aufträge erfüllt sind. Dies stellt im Normalfall ein langfristiges Arbeiten dar.

Das Konzept eines Agenten ist ein Werkzeug zur Analyse von Systemen. Es stellt keine absolute Charakterisierung zur Unterteilung der Welt in Agenten und nicht Agenten dar [Russel/Norvig, 2004]. Ein Agent kann demnach eine Maschine (z.B. Roboter), ein Mensch, aber auch ein Computerprogramm sein. In [Russel/Norvig, 2004] wird zwischen einem Agentenprogramm und einer Agentenarchitektur unterschieden, welche zusammen einen Agenten ergeben. Auf den Menschen bezogen stellt somit der Körper die Agentenarchitektur dar und das Denken, Schlussfolgern, usw. das Agentenprogramm.

## 1.2 Software-Agent

Im Gegensatz zum allgemeinen Agentenkonzept enthält ein Software-Agent keine physischen Teile. Ein Software-Agent ist ausschließlich ein Computerprogramm. Hierbei werden die Computer bzw. Computernetzwerke nicht als Agentenarchitektur, sondern als Umwelt betrachtet. Meist sogar eine durch anderweitige Software - Middleware-Systeme oder Agentenplattformen (siehe 6) - künstlich gebildete, weiter abstrahierte Sicht oder

sogar eine künstlich erstellte digitale Welt. Beispielsweise ist das Internet die Umwelt für einen Information-Retrieval-Agenten.

Der zentrale Punkt bei der Differenzierung von Software-Agenten gegenüber ’normaler’ Software sind neuartige Erwartungen an diese Programme [Burkard, 2002]. Allerdings wird in Fachkreisen immer noch darüber diskutiert, welche Eigenschaften mindestens erfüllt sein müssen, damit ein Software-Agent vorliegt [Gilgen, 1995]. In der Fachliteratur werden eine ganze Reihe von Dimensionen zur Beschreibung von Software-Agenten unterschieden. Diese müssen jedoch nicht alle erfüllt werden, damit ein Software-Agent vorliegt. (Eine komplette Auflistung der Dimensionen findet sich unter [Burkard, 2002](Kap.:24.2.2)) Die im Folgenden aufgeführten Eigenschaften sind die, die meist als die mindestens zu erfüllenden angesehen werden [Burkard, 2002], [Wikipedia].

- **Eigenständigkeit (Autonomie):**

Software-Agenten unterliegen keiner direkten Steuerung oder Kontrolle eines anderen Software-Agenten oder eines Nutzers. Sie handeln im Sinne des Auftraggebers, aber eigenständig.

- **Interaktion mit einer Umwelt:**

Software-Agenten nehmen Informationen aus ihrer Umwelt auf. Das beinhaltet ihren Auftrag, die Situation und die Folgen ihrer Handlungen, um diese überprüfen zu können.

- **Andauernde Verfügbarkeit / Aktivität:**

Software-Agenten sind über längere Zeit ansprechbar und nehmen Aufträge von Nutzern oder anderen Software-Agenten entgegen. Ihre Aktivitäten zur Verfolgung ihrer Ziele bzw. zur Erfüllung ihrer Aufträge können sich ebenfalls über längere Zeit hinziehen.

- **Zielgerichtetheit:**

Agenten verfolgen Ziele oder Aufträge, die unter Umständen ganze Handlungsfolgen erfordern und gegebenenfalls über einen längeren Zeitraum gehen.

Diese Sicht entspricht auch der bei AOSE. So kapseln Objekte lediglich ihre Identität ("wer"), ihren Zustand ("was") und ihr Verhalten ("wie"). Software-Agenten kapseln hingegen zusätzlich Freiheitsgrade in ihrer Aktionsauswahl und Interaktion ("wann", "warum", "mit wem" und "ob überhaupt"). [Weiß]

### 1.3 Intelligenter Software-Agent

Der Begriff der Intelligenz ist schwer zu definieren im Zusammenhang mit Software-Agenten. Er beinhaltet mit Sicherheit den Begriff der Rationalität. Nach [Russel/Norvig, 2004] ist ein rationaler Agent "[...] ein Agent der das Richtige tut." Hierbei ist eine Aktion "richtig", wenn sie den Agenten veranlasst so "erfolgreich wie möglich" zu sein. Er also einem Erfolgskriterium Rechnung trägt. Rationalität sollte nicht mit Optimalität verwechselt werden. Optimal kann nur dann von einem Agenten gehandelt werden, wenn er allwissend ist. Rationalität beinhaltet also ein Abwägen zwischen möglichst umfassender Analyse einer Situation vor dem Handeln und schnellem Handeln [Burkard, 2002]. Rationalität alleine ist noch nicht ausreichend. Lernfähigkeit, nutzen von Erfahrung und selbständige Anpassungsfähigkeit müssen außerdem gegeben sein, damit von Intelligenz gesprochen werden kann. Intelligente Software-Agenten zeichnen sich also neben rationalem Handeln durch Wissen, Lernfähigkeit, Schlussfolgern und der Möglichkeit zur Verhaltensänderung aus.[Wikipedia]

## 2 Grundlagen

Nachdem im letzten Abschnitt der Begriff der intelligenten Software-Agenten abgegrenzt wurde, wird in diesem, nach einigen Erläuterungen zum Begriff der Umwelt, auf den grundlegenden Aufbau von Software-Agenten eingegangen.

### 2.1 Umwelt

Wenn im Zusammenhang von Agenten von Umwelt gesprochen wird ist immer ein für den Agen-

ten relevanter Ausschnitt der Welt auf einer für ihn relevanten Abstraktionsebene gemeint. Für Software-Agenten ist stets ein Ausschnitt aus einer "virtuellen" Welt gemeint, wie beispielsweise das Internet, oder das World Wide Web. Die Umwelt kann dabei auch andere Software-Agenten beinhalten. Entweder direkt, durch Nachrichten (Anfragen/Aufträge) von ihnen oder auch indirekt, durch von ihnen hervorgerufene Veränderungen der Umwelt.

Die Umwelt hat einen starken Einfluss auf die Gestaltung und die Zuverlässigkeit eines Agenten. Einerseits betrifft dies seine Sensorik und seine interne Umwelt-Modellierung. Andererseits betrifft es seine Aktorik und Entscheidungsfindung [Burkard, 2002]. Bei einem Roboter kann beispielsweise eine Aktion "greife Gegenstand" sein. Bei einem Software-Agent besteht eine Aktion lediglich aus einer Nachricht. Diese Nachricht kann verschiedene semantische Bedeutungen in der virtuellen Umwelt haben.

In der Fachliteratur wird versucht mittels weniger Dimensionen die unendliche Anzahl möglicher Umwelt-Ausprägungen zu charakterisieren. Hier soll nun in Anlehnung an [Burkard, 2002] auf einige im Zusammenhang der Software-Agenten wichtige Dimensionen eingegangen werden.

- **Abgeschlossenheit der Umwelt**

Kann die Umwelt als feste Gesamtheit von festliegenden Situationen betrachtet werden, so ist sie abgeschlossen. Ist ein Agent beispielsweise für den An- und Verkauf von Aktien zuständig, so kann seine Umwelt nicht als abgeschlossen betrachtet werden, da neue Firmen an die Börse gehen könnten.

- **Vollständigkeit/Zugänglichkeit und Zuverlässigkeit der sensorischen Informationen**

Selbst wenn die prinzipielle Möglichkeit besteht alle (relevanten) Informationen zu erlangen, kann das unter Umständen zu aufwändig oder auch schlicht nicht erlaubt sein; z.B. wegen Datenschutzrichtlinien. Man spricht von "effektiver Zugänglichkeit", wenn alle für den Agenten wichtigen Informationen zur Entscheidungsfindung durch die Sensorik erfasst

werden. Diese Informationsselektion bei der Beobachtung und Verarbeitung sind ein wichtiger Aspekt der Agenten-Technik.

Hinzu kommt, dass sensorische Informationen unsicher sein können. Das ist durch unsichere Erfassung oder Übertragung mit unter sogar durch absichtliche Fälschung durch andere Agenten in der Umwelt möglich. Auch können Informationen an sich unscharf und dadurch unzuverlässig sein.

- **Determiniertheit der Umwelt**

Ist der nächste Zustand der Umwelt vollständig vom vorhergehenden Zustand und der vom Agenten ausgewählten Aktion abhängig spricht man von einer deterministischen Umwelt. Ist das nicht gegeben, liegt eine stochastische Umwelt vor. Nur zum Teil beobachtbare (durch Sensorik wahrnehmbare) Umwelt kann für einen Agenten stochastisch erscheinen. Eine Umgebung, welche bis auf die Aktionen anderer Agenten deterministisch ist, wird als strategisch bezeichnet.

- **Episodische Umwelt**

In einer episodischen Umwelt ist die Erfahrung des Agenten nur für eine atomare Episode relevant. Jede nachfolgende Episode ist unabhängig von jeder vorhergehenden. Das gegenteilige extrem ist die sequentielle Umgebung (im Sinne von fortwährend/andauernd). Hier können alle Entscheidungen alle nachfolgenden Entscheidungen beeinflussen. Jede noch so kleine Aktion hat also Langzeitwirkungen.

- **Dynamik der Umwelt**

Die Umwelt eines Agenten kann sich während seiner Entscheidungsfindung ändern. In einer solchen dynamischen Umwelt wird der Agent "ständig gefragt", was er tun will. Entscheidet er eine Zeit lang nichts, ist das als ob er nichts tun will. Ist die Umgebung eines Agenten hingegen statisch müssen keine Beobachtungen der Umwelt während der Entscheidungsfindung durchgeführt werden.

- **Granularität, Skalierungen, Wertebereiche**

Für einen Agenten können verschiedenste Skalen (Zeit, Raum, ...) in einer Umgebung relevant sein. Hierbei sind diskrete Skalen, insbesondere endliche, relativ einfach darzustellen. Schwierigkeiten machen hingegen vor allem kontinuierliche Wertebereiche.

Neben den einzeln aufgeführten Dimensionen gibt es einige weitere Eigenschaften. Beispielsweise Sicherheitsaspekte, die Zuverlässigkeit der Resultate von Handlungen eines Agenten und die zeitlichen Anforderungen an Entscheidungen der Agenten (echtzeit Systeme). Eine weitere recht prägnante Eigenschaft ist die Frage ob ein Einzelagenten- oder ein Multiagentenumgebung vorliegt.

## 2.2 Grundlegende Strukturen

Aus den Eigenschaften von intelligenten Software-Agenten ergibt sich, dass sie zyklisch arbeiten. Es werden drei Phasen unterschieden. Die Informationsaufnahme, die Phase der Wissensverarbeitung und Entscheidung, sowie die Phase der Aktionsausführung [Burkard, 2002].

Es gibt zunächst die von der Sensorik tatsächlich erkannten Reize  $r$  aus einer Menge erkennbarer Reize  $R$ . Die eingegangenen Reize  $r$  werden durch eine Funktion

$$\text{wahrnehmen} : R \rightarrow W$$

zu einer Wahrnehmung  $w \in W$  (Wahrnehmungen) aufbereitet. Es schließt sich dann ein interner Schlußfolgerungsprozeß an. Ganz allgemein gesprochen wird der interne Zustand  $z \in Z$  (interne Zustände) der Agenten verändert. Der neue Zustand jedes Zyklus  $z_{neu}$  hängt sowohl vom vorhergehenden Zustand  $z_{alt}$  als auch von der Wahrnehmung  $w$  ab. Der Schlußfolgerungsprozeß wird also von einer Funktion

$$\text{denken} : Z \times W \rightarrow Z$$

beschrieben. Anhand des Zustandes  $z$  führt der Agent schließlich Aktionen  $a \in A$  (mögliche Aktionen) gemäß der Funktion

$$\text{agieren} : Z \rightarrow A$$

aus. In vereinfachter Form lässt sich ein Agent also wie folgt (in Pseudocode) darstellen:

```
wiederhole für immer {  
  w := wahrnehmen(r);  
  z_neu := denken(z_alt, w);  
  a := agieren(z_neu);  
}
```

Die 3 Phasen dieses "wahrnehmen-denken-agieren-Zyklus" können unterschiedlich komplex implementiert sein. Ein realistischer nicht so stark vereinfachter intelligenter Software-Agent enthält die folgenden Komponenten [Burkard, 2002]:

- Umweltkopplung
- Steuerung und Management
- Fähigkeiten
- Umweltmodell
- Entscheidungskomponenten

Die beschriebenen 3 Phasen stehen mit diesen Komponenten in engem Zusammenhang, sind ihnen aber nicht fest zugeordnet.

Die Komponente "Umweltkopplung" wird dabei von den Funktionen *wahrnehmen* und *agieren* realisiert. In dieser Komponente und in der Komponente "Steuerung und Management" werden meist sogenannte Middlewarefunktionen bereitgestellt. Dies sind für den Programmablauf notwendige Funktionen, wie beispielsweise die Verwaltung nebenläufigen Arbeitens oder die Unterstützung für die Migration (Mobilität) der Agenten. Die eigentliche Intelligenz des Agenten befindet sich in der Funktion *denken*. Hier finden die KI-Methoden mittels der Komponenten "Fähigkeiten", "Umweltmodell" und "Entscheidungskomponenten" Anwendung. Meist werden hierfür die Funktion *denken* und der interne Zustand  $z$  weiter aufgliedert.

## 3 Architekturen

Nachdem der Grundlegende Aufbau von intelligenten Software Agenten vorgestellt wurde, soll nun auf Agentenarchitekturen eingegangen werde. Unter einer Agentenarchitektur wird die Art und Weise, wie die Definition und Verwaltung des Agentenverhaltens erfolgt, verstanden [Wikipedia]. Prinzipiell herrscht bei Agentenarchitekturen eine große Vielfalt. Es gibt jedoch eine weitgehend anerkannte Einteilung in zwei Bereiche [Wikipedia]. Die reaktiven und die deliberativen Agentenarchitekturen.

### 3.1 Reaktive Architektur

Reaktive oder auch Reflex-Agenten genannt, agieren direkt auf ihre Wahrnehmungen bzw. ihre Wahrnehmung in Verbindung mit ihrem internen Zustand. Bei reaktiven Agenten besteht der interne Zustand lediglich aus Annahmen  $b \in B$  (beliefs) über die Umwelt. Dieses interne Umweltmodell beinhaltet Annahmen darüber, wie die Umwelt unabhängig vom Agenten funktioniert und welche Auswirkung Aktionen des Agenten in der Umwelt haben. Dieses Modell muss ständig aktualisiert werden, beeinflusst von der Wahrnehmung des Agenten und internen Schlussfolgerungen. Dadurch ist gewährleistet, dass das Modell der Welt weitestmöglich gültig ist.

Die Phase der Wissensverarbeitung und Entscheidung (siehe 2.2) begrenzt sich daher auf den Abgleich der aktuellen Wahrnehmung  $w$  bzw. dem aktuellen Umweltmodellzustand  $b$  mit Bedingungs-Aktion-Regeln. Es wird also anstelle des Aufrufes

```
z_neu := denken(z_alt, w);
```

die Aktualisierung des Weltmodells vorgenommen mit

```
b_neu := aktualis(b_alt, w);
```

und anschließend durch den Abgleich des Weltmodellzustandes mit Bedingungs-Aktion-Regeln ein Aktionsplan  $p$  erstellt. Dieser wird im Anschluß durch die Funktion *agieren* ausgeführt, wie dargestellt.

```

p := abgleichen(b_neu);
a := agieren(p);

```

### 3.2 Deliberative / BDI Architektur

Die komplexeren, aber eher dem Begriff der intelligenten Software-Agenten entsprechenden Agenten, aus der Klasse der deliberativen Agentenarchitekturen verwenden zusätzlich zum internen Weltmodell weiteres explizites Wissen und verfügen über ein komplexeres Zustandsmodell. Hierdurch werden zielgerichtete Handlungen über längere Zeit ermöglicht.

Die BDI-Architektur ist der wohl bekannteste Vertreter der deliberativen Agentenarchitekturen. Deshalb soll sie in dieser Arbeit beispielhaft für die deliberativen Agentenarchitekturen betrachtet werden.

BDI steht für Belief, Desire und Intention, die drei Hauptbestandteile dieser Architektur [Wikipedia (3)]. Die Agenten werden also im Vergleich zu den reaktiven Architekturen zusätzlich mit Wissen über erstrebenswerte Zustände oder Wünsche (desire) und Absichten (intension) ausgestattet [Burkard, 2002]. Ein BDI-Agent lässt sich wie folgt in Pseudocode darstellen:

```

0 wiederhole für immer {
1   w := wahrnehmen(r);
2   b_neu := aktualis(b_alt, w);
3   d_neu :=
   ermitteln(d_alt, b_neu);
4   i_neu :=
   filtern(i_alt, d_neu, b_neu);
5   p_neu :=
   mza(p_alt, i_neu, b_neu);
6   a := agieren(p_neu);
7 }

```

Wie schon bei reaktiven Agenten wird in Zeile 2 das interne Weltbild aktualisiert. Mit Hilfe des Weltbildes (*b\_neu*) werden dann in der ersten Stufe des Deliberationsprozesses wünschenswerte Zustände (*d\_neu*) ausgewählt (3). Diese können untereinander oder zu bestehenden Absichten im Konflikt stehen. Die Funktion *ermitteln* ist

bei BDI-Agenten sehr wichtig, da der Agent ohne (neue) Ziele (*d\_neu*) zu keinen weiteren Aktionen veranlasst würde. Das zielorientierte Verhalten ermöglicht es außerdem, im Falle einer fehlgeschlagenen Aktion, das ausgewählte Ziel weiter zu verfolgen, z.B. durch einen alternativen Lösungsansatz oder einen späteren erneuten Versuch [Wikipedia (3)]. Die zweite Stufe des Deliberationsprozesses besteht aus dem Auswählen von weiter zu verfolgenden Absichten, aus den in der Vergangenheit schon ausgewählten Absichten und aus den Wünschen ableitbaren Absichten. Hierbei muss verhindert werden, dass Absichten aufgenommen werden, die für den Agenten nicht ausführbar sind (Realismus Forderung) [Burkard, 2002]. Aus diesem Grund ist die entsprechende Funktion auch mit *filtern* benannt (4).

Nach dem Deliberationsprozess folgt nun die sogenannte Mittel-Ziel-Argumentation (means-ends-reasoning) [Burkard, 2002]. Dabei ist wichtig, welche Fähigkeiten zur Verfügung stehen und inwieweit diese mit den ausgewählten Absichten (*i\_neu*)(5) übereinstimmen. Hierfür wird meist eine Plan- bzw. Fähigkeitendatenbank verwendet, aus der hierarchisch organisierte Pläne ausgewählt werden können, die den Agenten seinem Ziel näherbringen [Wikipedia (3)]. Die Implementierung der Funktion *mza* kann von einem einfachen Auswahlprozess bis hin zu einem komplexen Planungsverfahren jegliche Komplexität annehmen [Burkard, 2002]. Der ausgewählte Plan wird, einen Zyklus abschließend, noch durch die Funktion *agieren* ausgeführt.

## 4 Anwendungsbeispiele

Nachdem das Konzept und einige Architekturen von intelligenten Software-Agenten vorgestellt wurden soll nun in diesem Abschnitt ein Blick in die Praxis geworfen werden. Zu diesem Zweck sind einige Beispiele aufgeführt, die Aufschluss darüber geben sollen, wo und wie (intelligente) Software-Agenten bereits verwendet werden. Die Entscheidung, ob die Software-Agenten in diesen Anwendungen als Intelligent anzusehen sind soll dabei offen gelassen werden. Unter den Beispielen sind

einige Einzelagenten- sowie Multiagentensysteme vertreten, um einen möglichst breiten Überblick zu ermöglichen, auch wenn auf Multiagentensysteme an anderer Stelle in dieser Arbeit nicht näher eingegangen wird.

#### 4.1 Einzelagentensysteme

- Webcrawler

Ein Webcrawler (auch Spider oder Robot genannt) ist ein Computerprogramm, das automatisch das World Wide Web durchsucht und Webseiten analysiert. Webcrawler werden vor allem von Suchmaschinen eingesetzt. Weitere Anwendungen sind beispielsweise das Sammeln von Mailadressen oder anderweitige Informationen bis hin zur Informationsextraktion. Auch zum Data Mining und zur Untersuchung des Internets im Allgemeinen (Webometrie) eingesetzt. Webcrawler sind eine spezielle Art von Bots und werden auch als Spinnen bezeichnet, da sie sich wie diese, in einem Netz fortbewegen [Wikipedia (3)].

- Remembrance Agent

Dieser Agent unterstützt die Arbeit am Computer durch Erinnerungshilfen. Der Agent beobachtet ständig das Arbeiten am Computer und versucht Muster und Zusammenhänge zu erkennen. Mittels dieser Beobachtungen versucht der Agent dem Nutzer Hinweise zu geben, welches Dokument er als nächstes verwenden möchte bzw. welches er gerade sucht [Kwast, 1999].

- Information Management – Maxims

Maxims ist ein sogenannter Mailfilter-Agent. Er priorisiert und löscht E-Mails, leitet sie bei Bedarf automatisch weiter und sortiert und archiviert sie. Um diese Arbeiten im Sinne des Benutzers erledigen zu können beobachtet der Agent den Benutzer eine Zeit lang und lernt aus jeder Aktion des Benutzers. Maximus erstellt dafür intern immer Vorhersagen über die nächste Handlung des Benutzers. Sobald seine Vorhersagen zu einem gewissen Grad richtig sind fängt er an autonom zu handeln

und dem Benutzer Vorschläge für die Sortierung, die Priorisierung, usw. zu unterbreiten [Kwast, 1999].

#### 4.2 Multiagentensysteme

- Prozeßkontrolle – ARCHON-Plattform

Die ARCHON-Plattform wurde als eines der ersten Multiagentensysteme auf der Welt getestet und eingesetzt. Beispielsweise wird in Spanien die ARCHON-Plattform zum Management von Schaltwegen für Elektrizität eingesetzt. Auch findet die Plattform zur Kontrolle und Steuerung in Teilchenbeschleunigern Anwendung [Kwast, 1999].

- Luftverkehrkontrolle – OASIS

OASIS wird unter anderem im Flughafen von Sydney getestet und eingesetzt. Dabei werden sowohl Flugzeuge, als auch verschiedene Luftverkehrskontrollsysteme durch Agenten repräsentiert. Sobald ein Flugzeug in den Luftraum von Sydney gelangt, wird ein Agent mit den Informationen und Zielen des realen Flugzeugs erzeugt. Auf diese Weise lässt sich der Wunsch einer Flugzeugbesatzung auf einer bestimmten Rollbahn zu landen berücksichtigen. Auch 'simuliert' das System die Verhandlungen zur Lande- und Start-Freigabe. Hierfür kommunizieren die Flugzeugagenten mit den Flugverkehrskontrollagenten des Systems [Kwast, 1999].

### 5 Ausblick

Die Agententechnologie ist mit Sicherheit bedeutend und hat in den letzten Jahren große Fortschritte gemacht. Insbesondere ist der junge und sich schnell entwickelnde Bereich AOSE entstanden. Das große Interesse begründet sich darin, dass der agentenorientierte Ansatz sich den ständig wachsenden Anforderungen an Software (Offenheit, Verteiltheit, Dynamik, Heterogenität, ...) stellt [io-port]. Unterstützend wirkt dabei, dass dieser Ansatz mit anderen Ansätzen (z.B. Design Pattern) konsistent ist. Gebremst wird das AOSE aller-

dings noch dadurch, dass es kaum Entwicklungs-umgebungen gibt, die das Konzept umsetzen und deshalb immer noch mit objektorientierten Methoden Software-Agenten entwickelt werden müssen.

Dieser Sachverhalt wird sich sicherlich recht schnell ändern. Diese These erscheint, wegen vieler Vorteile des Konzepts, wie beispielsweise der Natürlichkeit des Ansatzes, aber vor allem wegen einem seitens der Industrie stark forcierten Standardisierungsprozess, sehr plausibel [io-port]. Sicher ist der agentenorientierte Ansatz kein Allheilmittel. Das Agentenmodell ist in manchen Fällen höchst wahrscheinlich reichlich überdimensioniert. Auch bestehen noch Realisierungsprobleme in Bereichen, wo Echtzeitantworten garantiert sein müssen oder in denen Deadlocks oder Livelocks garantiert nicht auftreten dürfen [Kwast, 1999].

Bei Planungsaufgaben in sehr dynamischem Umfeld ist momentan das größte Potential von intelligenten Software-Agenten zu sehen. So lässt sich hier durch 'simulieren von Verhandlungen' sehr schnell eine möglichst minimale und trotzdem ausreichende Umplanung bei entsprechenden Ereignissen erreichen. Was in diesem Anwendungsfeld ausgenutzt wird, ist zugleich die größte Schwäche von Intelligenten Software-Agenten. In fast jedem realistischen Softwareagentensystem ist ausreichend globales Wissen nicht erreichbar, was bedeutet, dass der Agent global gesehen 'nur' suboptimale Entscheidungen treffen kann [Kwast, 1999]. Das erschwert die Bildung von Vertrauen in intelligente Software-Agenten, was für den großflächigen Erfolg die wichtigste Voraussetzung ist. Dieses Vertrauen ist in vielen Bereichen noch nicht gegeben, was sich beispielsweise darin widerspiegelt, dass nicht geklärt ist, wer im Allgemeinen für das Handeln eines intelligenten Software-Agenten verantwortlich ist.

Außerdem tauchen im Zusammenhang mit intelligenten Software-Agenten immer wieder einige mehr oder weniger berechtigte Bedenken und Befürchtungen auf. Diese haben in der Vergangenheit mitunter Hollywood und diverse Schriftsteller beschäftigt. Dabei bleibt der Punkt offen, ob Fortschritte bei der 'Intelligenz' von intelligenten Software-Agenten irgendwann dazu führen, dass

diese eigene Interessen verfolgen und nicht mehr im Sinne des Nutzers arbeiten.

## 6 Anhang

### 6.1 Agentenplattformen

Es gibt zahlreiche Implementierungen von Agentenplattformen. Diese haben meist einen speziellen wissenschaftlichen Fokus. Beispielsweise intelligentes Verhalten, Sicherheit oder effiziente Migration. Einige wichtige Plattformen (übernommen von [Wikipedia]) sind:

- D'Agents (<http://agent.cs.dartmouth.edu/>)
- GNUBrain (GPL)
- JADE (<http://sharon.csel.it/projects/jade>)
- JIAC (<http://www.jiac.de/>)
- SeMoA - Secure Mobile Agents (<http://www.semoa.org/>)
- Tracy (<http://www.mobile-agents.org/>)

### 6.2 BDI-Agenten Frameworks

Frameworks zur Erstellung von BDI-Agenten (übernommen von [Wikipedia (2)]):

- Jadex (<http://vsis-www.informatik.uni-hamburg.de/projects/jadex/>) (Open Source-Aufsatz für Jade (<http://jade.tilab.com/>))
- JAM (<http://marcush.net/IRS/>) (kommerziell, kostenfrei für nichtkommerziellen Einsatz)
- JACK (<http://www.agent-software.com>) (kommerziell)
- AgentBuilder (<http://www.agent-software.com>) (kommerziell)



## Literatur

- [Burkard, 2002] Hans-Dieter Burkhard - Kap 24 Software-Agenten : Günther Görz, Claus-Rainer Rollinger, Josef Schneeberger (Hrsg.), *Handbuch der Künstlichen Intelligenz*, 3. Auflage, Oldenbourg 2002
- [Gilgen, 1995] Robert Gilgen *Ein System zur Automatisierung von Softwareprozessen*, Diplomarbeit, Technische Universität Berlin, 1995
- [io-port] io-port.net – informatics online (Informatik-Lexicon), *Agentenorientiertes Software Engineering (AOSE)*, (25.04.2005) [http://www.io-port.net/ioport2004/content/e45/e383/e443/e445/index\\_ger.html](http://www.io-port.net/ioport2004/content/e45/e383/e443/e445/index_ger.html)
- [Kwast, 1999] Torsten Kwast, *Intelligente Softwareagenten*, Seminararbeit WS 1999, (25.04.2005) <http://www.fh-wedel.de/~si/ws99/Ausarbeitung/agent/agent0.htm>
- [Russel/Norvig, 2004] Stuart Russel, Peter Norvig. *Künstliche Intelligenz - Ein Moderner Ansatz*, 2. Auflage, Pearson Education Deutschland, 2004
- [Weiß] Gerhard Weiß *Agentenorientiertes Software Engineering*, Paper, Technische Universität München
- [Wikipedia] aus Wikipedia, der freien Enzyklopädie *Software-Agent*, (25.04.2005) <http://de.wikipedia.org/wiki/Software-Agent>
- [Wikipedia (2)] aus Wikipedia, der freien Enzyklopädie *BDI-Agenten-Architekturen*, (25.04.2005) <http://de.wikipedia.org/wiki/BDI-Agenten-Architekturen>
- [Wikipedia (3)] aus Wikipedia, der freien Enzyklopädie *Webcrawler*, (02.05.2005) <http://de.wikipedia.org/wiki/Webcrawler>