

Einführung in die Agenten-Technologie

Hans-Dieter Burkhard, Institut für Informatik, Humboldt Universität Berlin



Prof. Dr. sc. Hans-Dieter Burkhard leitet das Lehr- und Forschungsgebiet „Künstliche Intelligenz“ am Institut für Informatik der Berliner Humboldt Universität. Seine Arbeitsgebiete sind Verteilte KI, Agenten-Orientierte Techniken, Fall-basiertes Schließen, Informationssysteme und KI-Anwendungen in der Medizin.

Der Begriff des „Agenten“ markiert neue Anforderungen und Erwartungen an die Leistungen maschineller Intelligenz. Diese Anforderungen erfordern neue technologische Lösungen im Sinne einer agenten-orientierten Technologie.

Introduction to Agent Technologies

The notion of an „agent“ describes a changed view to programs with broad expectations of the users. The new view corresponds with new requirements for machine intelligence. These changes require new solutions by agent-oriented technologies.

1 Was ist ein Agent?

Der Begriff des „Agenten“ ist in den letzten Jahren sehr populär geworden. Er markiert heute eine neue äußere Sicht auf Programme, die technologisch noch lange nicht fest umrissen ist (zum Vergleich: Die objekt-orientierte Programmierung benötigte ca. 20 Jahre Entwicklungszeit). Mit der Vorstellung von „Agenten“ sind bestimmte Erwartungen der Nutzer von Computern an (künftige) Programme verbunden. Zwar gibt es den Begriff schon länger – etwa für Management-Systeme oder in der (Verteilten) Künstlichen Intelligenz – doch inzwischen wird der Begriff wesentlich umfassender benutzt. Ein „Agent“ bezeichnet in der Regel ein

eigenständiges Programm. Darüber hinaus werden in der Zusammenarbeit von Mensch und Maschine oft auch die menschlichen Akteure als „Agenten“ bezeichnet.

Es ist gegenwärtig nicht möglich, eine allgemein zutreffende Definition des Begriffs „Agent“ zu geben, obwohl zahlreiche Autoren sich darum bemühen. Ganze Artikel sind der vergleichenden Betrachtung solcher Definitionen gewidmet, wobei jede Definition lediglich eine bestimmte Sicht widerspiegelt [3]. Daß der Agent trotzdem eine solche Rolle in der aktuellen Computerwelt spielt, liegt an den bereits erwähnten Erwartungen: Offenbar sollen gewisse Eigenschaften, die Programme früher nicht hatten, besonders hervorgehoben werden. Diese Eigenschaften knüpfen an die Bedeutung des Wortes „Agent“ in der Umgangssprache an:

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

Tatsächlich wird mit den Agenten-Programmen genau eine solche eigenständige Aktivität assoziiert. Dabei muß man sich bewußt sein, daß es sich hier zunächst um ein *anschauliches Bild* handelt. Agenten-Programme sind mit bestimmter Funktionalität entworfen, und sie „handeln“ (bei korrekter Implementierung) nach den Vorgaben des Programmierers. Ihre Aktionen erinnern aber an das selbständige Handeln eines menschlichen Agenten (der im übrigen auch den Instruktionen seines Auftraggebers folgen soll). Auch die Benutzung von Begriffen wie „Handeln“, „Aufträge“, „Vorgaben“ usw. stammt aus dem menschlichen Bereich. Gerade solche, eigentlich an Personen gebundene Vorstellungen werden mit den Agenten-Programmen verknüpft. Nur so ist auch die allgemeine Übereinkunft zur Benutzung dieses (nicht definierten) Begriffs

zu erklären. Im übrigen endet die Parallele vielleicht gerade dort, wo der Mensch seine Vorschriften im geeigneten Moment auch verlassen darf: Von einem Computer-Agenten erwarten wir den absoluten „Dienst nach Vorschrift“.

Traditionelle Programme werden eher als passive Systeme gesehen: Auf eine bestimmte Eingabe wird eine bestimmte Ausgabe erwartet. Das heißt nicht, daß es bisher noch keine Programme gab, auf die das Bild des eigenständigen Handelns zutraf. Vielmehr werden solche Programme neuerdings auch als Agenten bezeichnet. Umgekehrt können natürlich auch Agenten-Programme eher traditionelle Züge aufweisen. So reagiert eine Suchmaschine („Informationsagent“) im Web auf eine Anfrage im Prinzip wie eine Datenbank: Alle passenden Einträge werden aufgelistet. Die aktive Seite der Suchmaschine besteht im selbständigen Durchsuchen von Web-Seiten und in der Strukturierung der gefundenen Informationen. Darüber hinaus werden künftige „intelligenter“ Suchmaschinen auch auf der Nutzerseite eigene Aktivitäten zur Unterstützung des Nutzers aufweisen müssen.

2 Aufgabenbereiche für Agenten

Ein „selbständig agierendes Programm“ kann seine Aufträge aus unterschiedlichen Quellen beziehen. Zunächst gibt es vielfältige Anwendungen, bei denen ein Agent die Aufträge *seines* menschlichen Nutzers ausführt und speziell diesem *seinem* Nutzer zur Verfügung steht. Dazu gehören „Assistenz-Programme“, die im Sinne eines persönlichen Assistenten Termine verwalten, Post (Email) bearbeiten, Dokumente archivieren usw. Es

gibt (möglicherweise migrierende) Agenten, die nach relevanten Informationen suchen und entfernt Aufträge ausführen (z.B. Hotelbuchung).

Aber auch für eher traditionelle Programme werden die sogenannten „intelligenten“/„nutzer-adaptiven“ Schnittstellen zunehmend als Agenten realisiert. In der einfachsten Form handelt es sich dabei um aufgewertete Hilfesysteme. In der Tendenz geht es aber um neue, an menschliche Erfordernisse angepasste Interaktionsformen zwischen Mensch und Maschine. Die Simulation von Schreibtisch-Oberflächen, Büro-Umgebungen oder Katalogen wird durch animierte Oberflächen verdrängt, in denen Agenten in virtuellen Räumen agieren und Aufträge übernehmen können: Verkaufsagenten können durch virtuelle Kaufhäuser führen, virtuelle Geräte erläutern oder virtuelle Touristik-Attraktionen vorstellen. Virtuelle Helpdesk-Agenten beraten an virtuellen Maschinen-Modellen, virtuelle Assistenten operieren in Workflow-Management-Systemen usw.

Kritisch ist bislang der Umgang mit komplexen Programmen für den sogenannten „normalen“ Nutzer, der über keine Programmierkenntnisse verfügt. Mit dem Bild des Agenten sind hier Erwartungen an Nutzerschnittstellen verbunden, die es erlauben, Aufträge an Programme wie an einen menschlichen Bearbeiter zu übertragen. Die bisherigen Schnittstellen erlauben die Manipulation des Computers, künftige Systeme sollen die Delegation von Aufgaben ermöglichen. Erforderlich ist aber stets eine präzise Vorstellung von den gewünschten Arbeitsabläufen (andernfalls ergibt sich ein „Eulenspiegeleffekt“, wenn der Rechner akkurat ausführt, was ihm aufgetragen wird).

Eine weitere Anwendungsform wurde oben bereits angesprochen: Hierbei handelt es sich um Agenten, die Aufträge *unterschiedlicher* menschlicher Nutzer ausführen und öffentlich zur Verfügung stehen. Dazu zählen heute vor allem die Informationsagenten (Suchmaschinen, Kataloge), die im weitesten Sinne Informationen anbieten über Produkte, Organisationen, Veranstaltungen, über Personen, wissenschaftli-

che Resultate, Musik, Bilder usw. Verbunden damit sind oft Verkaufsabsichten (Electronic Commerce), der Verkaufsagent soll ähnliche Funktionen ausfüllen wie sein menschliches Vorbild (beraten, alternative Angebote nennen ...). Der künftige Erfolg solcher Anwendungen hängt wesentlich von der auf menschliche Bedürfnisse eingestellten Kommunikation mit dem Computer ab. Gleichzeitig spielen zahlreiche Probleme eine Rolle, die bisher nur auf den Verkehr zwischen Menschen zutrafen (wirtschaftliche und rechtliche Aspekte, Sicherheitsfragen, Verantwortlichkeit). Verkaufsagenten lassen sich im übrigen auch unter dem Blickwinkel der Beauftragung durch *einen* Nutzer (den Verkäufer) betrachten.

Mit den obigen Szenarien wurde bereits ein weiterer wichtiger Gesichtspunkt angesprochen: Das Zusammentreffen von Agenten im Netz mit der Notwendigkeit zur Kooperation, zum Verhandeln und ggf. auch zum Lösen von Konflikten. Kooperation ist notwendig zwischen einem persönlichen Agenten mit dem Auftrag, eine Reise zu buchen, und dem Agenten eines Reiseanbieters: Sie müssen sich verständigen, ob eine passende Reise vorliegt. Danach beginnt eine Phase der Verhandlung zur Festlegung von Konditionen und Preisen, und es kommt möglicherweise zum Vertragsabschluß. Eine weitere Phase mit der Schlichtung aufgetretener Konflikte wird sicher nicht angestrebt, ist aber durchaus möglich. Gleichzeitig befinden sich die unterschiedlichen Anbieter von Reisen in Konkurrenz.

3 Offene Systeme

Heutzutage stehen bei Überlegungen zu Agenten vor allem Szenarien in Computernetzen im Vordergrund. Szenarien kooperierender bzw. in Konkurrenz stehender Agenten wurden aber auch schon früher betrachtet, etwa im Rahmen von Verteiltem Problemlösen oder von Multi-Agenten-Systemen in der Verteilten KI. Insbesondere der Gesichtspunkt der „offenen Systeme“ [5] hat dabei eine große Rolle

gespielt (gleichzeitig ist HEWITTS Begriff des „Actors“ eine wesentliche Quelle bei der Entstehung objekt-orientierter Technologien gewesen). Offene Systeme sind charakterisiert durch kontinuierliche Bereitschaft, Erweiterbarkeit, dezentrale Steuerung, asynchrone Arbeitsweise, inkonsistente Informationen, „Armlängen-Reichweite“.

Das Internet – aber auch andere vernetzte Systeme, sind Beispiele offener Systeme. Wichtig ist die ständige Bereitschaft in solchen Systemen. Sie erfordert bereits Programme, die eigenständig agieren. Die Systeme sollen erweiterbar sein: Traditionelle Software-Systeme erweisen sich schnell als Fessel für den Fortschritt. Dem Wachstum von Aufgaben entsprechend sollen neue Komponenten problemlos integriert werden können und bei laufendem Betrieb (etwa eines Krankenhaus-Informationen-Systems) muß der Austausch von Komponenten möglich sein. Dabei erweist sich jede globale Steuerung als Hemmschuh, und anstelle von Client/Server-Lösungen bietet sich die „Peer-to-peer“-Kommunikation und -Kooperation gleichberechtigter Agenten an.

Mit der zentralen Steuerung entfällt auch die Notwendigkeit zur Synchronisation. Nur die an einer Aufgabe beteiligten System-Komponenten müssen sich zeitweise aufeinander abstimmen. Informationen können lokal verwaltet werden, sie werden bei Bedarf abgefordert. Das kann dazu führen, daß Daten nicht systemweit auf dem neuesten Stand sind (was auch datenschutzrechtliche Gründe haben kann). Schließlich bedeutet „Armlängen-Reichweite“, daß nur die (inhaltlich) benachbarten Teilsysteme zusammenarbeiten müssen.

Die Entwicklung und Wartung komplexer Systeme ist nur mit geeigneten Technologien möglich. Hier spielen gegenwärtig vor allem objekt-orientierte Technologien eine Rolle. Es wird aber seit längerem diskutiert, ob das Bild des „Agenten“ als selbständig handelnder Einheit hilfreich für die Strukturierung großer heterogener Systeme eingesetzt werden kann. In diesem Sinne war „agenten-orientierte Programmierung“ als Fortsetzung von ob-

jekt-orientierter Programmierung angedacht [7]. Als offenes Problem galt dabei insbesondere die „Agentifizierung“, also die Beschreibung von System-Komponenten im Sinne von Agenten. Der Agent bietet eine Form der Datenabstraktion, die über andere Formen hinaus geht: Den einzelnen Komponenten können aktive Rollen zugewiesen werden, und Interaktionen sind intuitiv modellierbar.

Als Beispiel soll die Steuerung in einer flexiblen Fertigungsanlage betrachtet werden [8]. Die Umwelt ändert sich fortwährend durch neue Aufträge und den zeitweiligen Ausfall von Maschinen. Gesucht sind Planungsverfahren (Scheduling) für den optimalen Betrieb. Es geht weniger um die *einmalige* Berechnung eines optimalen Plans, sondern vielmehr um die ständige optimale Anpassung an veränderte Bedingungen. Dabei spielt Stabilität eine wichtige Rolle: Notwendige Änderungen (z.B. Verlagerung auf eine andere Maschine) sollen möglichst lokal abgefangen werden können, und die Umschichtungen sollen an weitere Bedingungen (Bediener, Transport) angepaßt werden.

In der Realität werden solche Probleme weitgehend durch Verhandlungen in „Armlängen“-Reichweite geklärt. Die Modellierung solcher Prozesse mit den tatsächlichen Beschränkungen durch ein monolithisches System ist problematisch und schwer handhabbar. Die Modellierung durch ein System kooperierender Agenten (für Maschinen, Werkstücke, Transporteinheiten) ist dagegen wesentlich intuitiver und deswegen unter softwaretechnischen Gesichtspunkten leichter zu beherrschen. Der erhoffte Nutzen agenten-orientierter Technologien ist hier die größere Transparenz der Programme.

4 Wesentliche Konzepte

Alle Definitionen des Begriffs „Agent“ beziehen sich zumindest auf das „selbständige Agieren“, auf die *Autonomie*: Auf Signale aus der Umwelt reagiert das System aufgrund eigenständiger Entscheidungen. Dagegen kann eingewendet

werden, daß die Kriterien für diese Entscheidungen durch den Programmierer vorgegeben sind. Das gleiche trifft aber auch auf menschliche Agenten zu: Ihr Auftrag und gewisse Hilfsmittel sind durch den Auftraggeber vorgegeben, über die konkrete Ausführung entscheidet der Agent aber allein zum gegebenen Zeitpunkt und in Abhängigkeit von den jeweiligen Umständen. Bei Programmen, die sich an ihre Aufgaben durch Lernen immer besser anpassen können, sind sogar nur noch Lernkriterien vorgegeben.

Bei komplexeren Entscheidungsverfahren kann das Programm mithilfe der Stärken des Computers zu Entscheidungen gelangen, die der Entwickler selbst nicht hätte finden können (man denke an Schachprogramme). Auf der anderen Seite gibt es ganz simple Systeme, die ebenfalls autonom agieren, zum Beispiel ein Thermostat. Komplexe Klima-Anlagen mit Anpassung an Wettervorhersagen und an Gewohnheiten der Hausbewohner usw. bieten allein vom Gesichtspunkt der Autonomie her keinen Unterschied zum Thermostaten. Unterschiedlich sind die Aufgabenbereiche und die Entscheidungsverfahren: Ein Thermostat arbeitet rein *reaktiv*, Sensorinformationen werden unmittelbar in Steuerungs-Aktionen umgesetzt, während die Klimaanlage komplizierte Entscheidungsvorgänge nachvollzieht und sich zusätzlich Wissen über das Verhalten der Menschen aneignet.

Es ist offensichtlich, daß komplexe Funktionalitäten und hohe Erwartungen an Agenten-Programme auch stets eigenständiges Handeln, also Autonomie erfordern. Das gleiche gilt für komplexe Steuerungssysteme von Anlagen und Robotern. Auch hier findet der Begriff des „Agenten“ zunehmend Verwendung.

Ein weiteres wichtiges Konzept ist die *Kooperation*, sowohl von Agenten untereinander als auch in bezug auf die menschlichen Nutzer. Mit der Kooperation eng verbunden ist die Kommunikation. Dabei spielen viele unterschiedliche „Schichten“ eine Rolle, die jeweils durch entsprechende Protokolle geregelt werden müssen. Neben den hard-

ware-nahen Protokollen gibt es Protokolle, die die inhaltliche Verständigung zwischen Agenten regeln (etwa unter der Berücksichtigung von Sprech-Akten). Auf noch höherer Ebene spielen Protokolle des „sozialen Zusammenlebens“ eine wesentliche Rolle (etwa zur Vermeidung und Klärung von Konflikten). Der *soziale Aspekt* wird häufig auch als wesentliches Konzept bei den vorgeschlagenen Definitionen des Begriffs „Agent“ gesehen.

Schließlich spielen *mentale Konzepte* eine wichtige Rolle. Der Begriff des „intelligenten“ Agenten ist zwar zuweilen nicht viel klarer als der Begriff der „intelligenten“ Waschmaschine, wenn inhaltlich nur die Abgrenzung von primitiverem Verhalten (Thermostat) angesprochen wird. Tatsächlich werden jedoch die Planungs- und Entscheidungsverfahren bei komplexen Agenten mit sehr spezifischen KI-Methoden modelliert. Hier gibt es eine Anlehnung an menschliche Entscheidungsprozesse, insbesondere spielt der Begriff der *Rationalität* eine wichtige Rolle. „Vernünftige“ Entscheidungen sollen mit vernünftigem Aufwand getroffen werden. Ziele sollen nur verfolgt werden, solange sie erreichbar sind.

Heutige Computerprogramme können in manchen Bereichen schneller zu guten Resultaten kommen als ein Mensch (zum Beispiel bei der Auswahl günstiger Zugverbindungen). Trotzdem wird ein solches Programm kaum noch als „intelligentes“ Programm bezeichnet. Die frühere Definition der „Künstlichen Intelligenz“ durch Leistungen, die beim Menschen als „intelligente“ Leistungen gelten würden, ist offenbar überholt. Dementsprechend ist auch der Begriff des „intelligenten Agenten“ schwer faßbar. Gleichzeitig stellen eher als primitiv angesehene Leistungen der menschlichen Intelligenz (wie das Überqueren einer belebten Straße in unbekannter Umgebung) noch echte Herausforderungen für künstliche Agenten dar.

Neben der Autonomie gilt vielfach die *Mobilität* als ein besonders wichtiges Konzept [11]. Das kann eine eher passive Mobilität betreffen: Ein Nutzer lädt ein Agenten-Programm zur Ausführung auf sei-

nen Rechner. Im Mittelpunkt der aktuellen Untersuchungen steht aber mehr die aktive Variante, in der ein Agent aus eigenem Entschluß auf einen anderen Rechner „migriert“, um dort einen Auftrag (z.B. Informationssuche) zu erfüllen.

5 Komplexität von Agenten

In der Verteilten Künstlichen Intelligenz werden seit langem unterschiedliche Agenten untersucht. Die Orientierung am menschlichen Vorbild führt hier zur Adaption von Begriffen, z.B. aus der Psychologie, Philosophie, Soziologie, Betriebswissenschaft [10]. Parallel zu dieser Entwicklung verläuft die Beschäftigung mit der Entstehung komplexen Verhaltens aus dem Zusammenwirken primitiver Agenten, gegenwärtig vielfach untersucht im Bereich „*artificial life*“.

Tatsächlich gibt es unterschiedliche Möglichkeiten zur Erzeugung von komplexen Verhaltensweisen:

- vollständige interne Modellierung der Umwelt durch komplexe Agenten,
- Reaktionen auf Umweltbedingungen durch einfachere Agenten und
- kollektives Verhalten einfachster Agenten.

Die Beschäftigung mit diesen Möglichkeiten ist auch für den Entwurf agenten-basierter Systeme lehrreich. Zunächst sollen die Unterschiede an einem Beispiel erklärt werden: Es ist ein Weg zu einem markanten Punkt in der Landschaft zu finden.

Der Agent mit der komplexen Umweltbeschreibung (z.B. mit einer Landkarte) kann sein inneres Modell inspizieren, einen Plan (eine Wegbeschreibung) generieren und entsprechend diesem Plan handeln (100 m geradeaus, dann nach links, weiter geradeaus bis ...).

Alternativ dazu genügen vergleichsweise einfache Verhaltensweisen: Der Agent muß die Richtung (Kirchturm) bestimmen können, in der das Ziel liegt. Er verfolgt diese Richtung soweit das möglich ist. Wenn Hindernisse auftreten, um-

geht er diese solange, bis er sich wieder geradlinig in Richtung auf das Ziel bewegen kann. Es kann sinnvoll sein, daß er sich mittels eines Würfels entscheidet, in welcher Richtung er ein Hindernis umläuft.

Schließlich kann eine Gruppe von Agenten ohne spezielle Fähigkeiten einfach loslaufen: Wenn sie dabei unterschiedliche Wege einschlagen (hier kann wieder ein Würfel für Unterschiede sorgen), werden einige irgendwann das Ziel erreichen.

Es ist denkbar, daß sich Agenten, die zunächst kein inneres Modell haben, während ihres Laufens bestimmte Dinge merken, sich also ein Modell erarbeiten. Alternativ dazu kann die Landschaft markiert werden („externes Gedächtnis“): Ameisen sind dadurch in der Lage, optimale Wege zwischen Futterplätzen und Nest anzulegen.

Vorzüge der einfacheren Strukturen liegen auch in ihrer Robustheit gegenüber Änderungen in der Umwelt. Allgemein läßt sich sagen, daß die Aufteilung auf mehr Agenten zu einfacheren Agenten-Strukturen führt, dafür aber mehr Aufwand für die Interaktion entsteht. Dieser Mehraufwand bleibt vergleichsweise gering, wenn sich die Aufteilung an „natürlichen“ Schnittstellen vollzieht [9]. Auch hier steht im Zentrum die intuitive Modellierung: Wenn Dinge in natürlicher Weise beschrieben werden, lassen sich konzeptionelle Fehler leichter vermeiden. Wie in menschlichen Gemeinschaften geht es darum, Entscheidungen am richtigen Ort treffen zu lassen.

6 Verhaltensweisen und Architekturen

Grundsätzlich lassen sich zumindest drei Komponenten eines Agenten identifizieren:

- Sensor-Komponente,
- Entscheidungskomponente und
- Aktions-Komponente.

In dieser einfachen Form ist zum Beispiel die Kommunikation ein Teil der Sensorik (Empfang von Nachrichten) und der Aktorik (Sen-

den von Nachrichten). Grundlage für die Aktionen ist eine Menge von Fähigkeiten, die etwa als Bibliothek ausführbarer Prozeduren (Pläne, Skripten) vorliegt.

Entsprechend den drei Komponenten lassen sich drei Phasen der Arbeit eines Agenten identifizieren („observation – thought – action“ – Zyklus):

- Sensorische Aktivität (Aufnahme, Verarbeitung),
- Interne Entscheidungsprozesse (Auswahl, Planung),
- Effektorische Aktivität (Aktionen, kommunikative Äußerungen).

Eine weitgehend parallele Ausführung dieser Schritte ist vor allem eine Frage der Synchronisation.

Es wird zwischen zwei Arbeitsformen von Agenten unterschieden, wobei die Auffassungen zuweilen etwas divergieren [12]:

- reaktives Verhalten,
- reflexives oder deliberatives Verhalten.

Bei reaktivem Verhalten stellen die Aktionen direkte Reaktionen auf Reize aus der Umgebung dar (evtl. noch in Abhängigkeit von einem inneren Zustand). Bei reflexivem/deliberativem Verhalten wird dagegen ein Entscheidungs- und Planungsprozeß durchlaufen, bei dem Aktionen vorausschauend bewertet und ganze Handlungsfolgen erzeugt werden können.

Einfache Agenten sind rein reaktiv (Thermostat), komplexere Agenten können dagegen sowohl reaktives als auch reflexives Verhalten zeigen. So kann durch einen Entscheidungsprozeß zunächst ein Handlungsziel ausgewählt werden. Zum Erreichen dieses Ziels wird dann ein Plan (eine Aktionsfolge) bestimmt, der seinerseits reaktiv abgearbeitet wird.

Die verfügbaren „Fähigkeiten“ bestimmen die Handlungen des Agenten. Dabei kann es sich um einfache Aktionen oder um ganze Aktionsfolgen (Pläne) handeln. Bei rein reaktivem Verhalten ist einem Reiz aus der Umwelt einfach eine auszuführende Fähigkeit zugeordnet. Softwaretechnisch handelt es sich dabei zum Beispiel um eine Prozedur oder ein Script, die noch

parametrisierbar sein können. Die Umweltreize wirken als Trigger für diese Prozeduren, es erfolgt eine ereignisgesteuerte Verarbeitung.

Bei komplexeren Agenten steht vor der Entscheidung für die Anwendung einer Fähigkeit noch die genaue Analyse der Umweltbedingungen und der möglichen Resultate. Dazu können auch das Zusammenfügen von Fähigkeiten zu einem längeren Plan und die Kooperation mit anderen Agenten gehören. So könnte ein Agent den Auftrag „Konvertierung eines Textes“ auf folgende Weise bewerkstelligen: Zunächst stellt er fest, welcher Agent (zum Beispiel auf einem Macintosh) über ein entsprechendes Konvertierungsprogramm verfügt. Dann überstellt er diesem Agenten den Text mit dem Auftrag zur Konvertierung. Schließlich übernimmt er den konvertierten Text in einer zuvor vereinbarten Weise. Falls die dazu notwendige Aktionen-Folge noch nicht als Plan vorliegt, könnte sie im Laufe eines Planungsprozesses erzeugt werden (und danach möglicherweise auch in eine Plan-Bibliothek aufgenommen werden).

7 Technologische Anforderungen

Software-Technologien sollen die Kluft überbrücken, die zwischen der inhaltlichen Beschreibung von Problemen einerseits und ihrer maschinensprachlichen Kodierung andererseits bestehen. Das vom Menschen bearbeitete Programm soll sich möglichst eng an die inhaltliche Beschreibung halten können. Weiterhin gehören zu einer Software-Technologie umfangreiche Hilfsmittel, die eine effiziente Arbeit in der jeweiligen Programmier-Umgebung erlauben.

Die Entwicklung solcher Umgebungen ist aufwendig, und die Umstellung auf eine neue Technologie bereitet zahlreiche Probleme. So ist es zu erklären, daß veraltete Technologien noch lange weiter verfolgt werden müssen. Gleichwohl erzwingt die wachsende Kluft zwischen den Erfordernissen neuer An-

wendungen einerseits und den Fesseln einer unzureichenden Technologie andererseits letztlich die Einführung und Stabilisierung neuer technologischer Lösungen.

Agenten-orientierte Technologien wurden zunächst nur als mögliche Erweiterungen von objekt-orientierten Technologien betrachtet [7]. Inzwischen wird aber immer klarer, daß mit dem Begriff des „Agenten“ völlig neue Erwartungen an die Nutzung von Computern geknüpft sind. Die Interaktion mit Agenten-Programmen soll auf eine Weise geschehen, die sich an die Interaktion zwischen Menschen anlehnt. Die Programme sollen Aufträge selbständig ausführen wie auch ein menschlicher Agent. Insgesamt handelt es sich um das massenhafte Erstellen geeigneter Software. Prinzipiell ist die Verwendung herkömmlicher Technologien möglich (und zunächst auch notwendig), auf die Dauer ergeben sich daraus aber ernsthafte Produktivitäts- und Zuverlässigkeitsprobleme.

Künftige agenten-orientierte Technologien müssen Lösungen aus unterschiedlichen Bereichen integrieren (und teilweise auch noch entwickeln). Dazu gehören unter anderem Techniken aus den Bereichen Verteilte Systeme, Computernetze, Management, objekt-orientierte Programmierung, Künstliche Intelligenz, Datensicherheit, Datenschutz. Zu klären und gegebenenfalls technologisch abzusichern sind z.B. juristische, wirtschaftliche und soziale Gesichtspunkte. Für Agenten-Architekturen und für die Gestaltung der Interaktionen zwischen Agenten und Nutzer sind Beiträge aus Psychologie, Philosophie und Organisationswissenschaften erforderlich. Wenn die mit einem Agenten verknüpften bildhaften Vorstellungen mehr Transparenz in die Struktur von aktiv agierenden Programmen bringen sollen, müssen die verwendeten Bilder korrekt sein. Das bedeutet, daß die Verwendung mentaler Begriffe wie „Wissen“, „Ziele“, „Wünsche“, „Aufträge“, „Absichten“, „Fähigkeiten“, „Verpflichtungen“ in sich konsistent und mit allgemeinen Vorstellungen verträglich sein müssen.

Es ist fraglich, ob alle notwendigen oder wünschenswerten Aspekte letztlich in eine geschlossene Technologie münden werden. Dafür sind wahrscheinlich zu viele unterschiedliche Anwendungsarten zu berücksichtigen. Vielmehr könnte es Lösungen für einzelne Anwendungsbereiche bzw. einzelne Problemkreise geben.

Unter solchen Lösungen kann auch eine agenten-orientierte Programmiersprache sein, die bisherige objekt-orientierte Methoden aufgreift und – z.B. in Form von Bibliotheken – weiter entwickelt. Tatsächlich setzen viele Agenten-Implementationen auf objekt-orientierten Methoden auf. Dabei bieten aktuelle Entwicklungen objekt-orientierter Methoden (JAVA) bereits grundlegende Unterstützung für Rechner-Unabhängigkeit, Migration und Kommunikation über Rechengrenzen hinweg. Die Lösung dieser vor allem netzwerkbezogenen Fragen war wichtig für die stürmische Entwicklung im Bereich der Agenten.

Interessanterweise bietet die agenten-orientierte Technologie auch die Lösung für die Integration unterschiedlicher Programmierkonzepte und für die Einbindung alter Lösungen in neue Systeme an. Das alte Programm erhält eine „Umhüllung“ (engl. „wrapper“), die nach außen als ein Agent in der neuen Umgebung erscheint und intern die Interaktion mit dem unveränderten alten Programm realisiert.

Zahlreiche Probleme sind noch in der Form von Protokollen auf höheren Ebenen zu lösen. Gefordert werden immer wieder standardisierte Kommunikationssprachen und -protokolle. Die Klassifikation von Kommunikationsakten orientiert sich meist an der Sprechaktheorie und ist Bestandteil entsprechender Protokolle. Als Sprache ist KQML [2] weitgehend akzeptiert.

Ebenfalls in der Form von Protokollen ist das soziale Zusammenwirken von Agenten zu organisieren. In den Arbeiten aus der Gruppe um J. Rosenschein [6] wurden Vorschläge für Protokolle zur Gewährleistung bestimmter Verhaltensweisen ausgehend von spieltheoretischen Untersuchungen entwickelt.

Die Künstliche Intelligenz hat Fragen der (intelligenten) Problemlösung intensiv untersucht und kann damit viele nützliche Konzepte für die Architektur rationaler Agenten liefern. Die Architekturprinzipien wurden zunächst im Rahmen multi-modaler Logiken entwickelt. Sie sind inzwischen soweit verstanden, daß sie eine hervorragende Basis für die Strukturierung komplexer Agenten bieten (insbesondere auch für die Implementierung mit objekt-orientierten Methoden, – Beispiel dafür ist die Entwicklungszeit von nur wenigen Monaten für die Programme der Mannschaft „AT Humboldt“, den Sieger der 1. Weltmeisterschaft im virtuellen Computer-Fußball des RoboCups 1997 [1]).

Aus der Verteilten KI kommen bewährte Konzepte für die Kommunikation und die Kooperation von Agenten. Dazu gehören Blackboard-Architekturen und Contract-Net-Protokolle. Wie bei einzelnen Programmen bieten agenten-orientierte Begriffe konzeptionelle Vorteile: Schon die Strukturierung komplexer Systeme in der Begriffswelt von Multi-Agenten-Systemen kann wesentliche Verbesserungen bringen.

Es gibt bereits zahlreiche Erfahrungen aus prototypischen Entwicklungen, speziell im Bereich von Entwicklungsumgebungen für vor-strukturierte Agenten bzw. Multi-Agenten-Systemen. Inzwischen gibt es auch bereits kommerzielle Produkte wie dMARS [4].

Bei der Entwicklung von agenten-orientierten Techniken lassen sich gegenwärtig zwei Hauptrichtungen identifizieren:

- Anreicherung von Programmiersprachen um agenten-orientierte

Konzepte, meist in der Form von Bibliotheken. Dabei ist eine Konzentration auf JAVA zu beobachten. Die Plattform-Unabhängigkeit ist für das Agieren in verteilten heterogenen Systemen eine wesentliche Eigenschaft.

- Generische Agenten und generische Multi-Agenten-Systeme. Hier werden bestimmte Architektur-Konzepte so umgesetzt, daß der Nutzer seine eigenen Komponenten für Sensorik, Entscheidungsfindung und Aktorik spezifizieren kann.

8 Zusammenfassung

Wir stehen an einem Umbruch bei der Nutzung von Informationstechniken, der in allen seinen Konsequenzen noch nicht abschätzbar ist. Mit den weltweiten Netzen, den Möglichkeiten selbständig aktiver Agenten-Programme und einer menschenfreundlicheren Interaktion zwischen Mensch und Maschine sind neue Erwartungen und Anforderungen entstanden. Diese Erwartungen können heute befriedigt werden. Sie erzeugen einen gewaltigen Bedarf an Software-Produkten. Die Erschließung agenten-orientierter Technologien ist im Augenblick von dieser Entwicklung überholt worden, die Untersuchung und Einführung weiterer Werkzeuge und Methoden ist deshalb eine dringende Aufgabe. Aber schon die Strukturierung von Programmen nach Agenten-Gesichtspunkten („agenten-orientiertes Spezifizieren“) bietet auch im Rahmen existierender Technologien wesentliche Vorteile.

Literatur

- [1] Burkhard, H.D.; Hannebauer, M.; Wendler, J.: AT Humboldt – Development, Practice and Theory. Proc. Workshop „RoboCup 97“, 351–366 (erscheint in der LNAI-Serie des Springer-Verlags).
- [2] Finin, F.; Labrou, Y.; Mayfield, J.: KQML as an Agent Communication Language. In: Bradshaw, J. (Hrsg.): Software Agents. MIT Press, Cambridge, 1997.
- [3] Franklin, S.; Graesser, A.: Is it an Agent, or just a Program: A Taxonomy for Autonomous Agents. <http://www.mscl.memphis.edu/franklin/AgentProg.html> In Proc. 3. Int. Workshop on Agent Theories, Architecture, and Languages (ATAL), Budapest 1996, 193–206.
- [4] Georgeff, M.P.; Kinny, D.N.: Modeling and Design of Multi-Agent Systems. In: Müller, J.P.; Wooldridge, M.J.; Jennings, N.R. (Hrsg.): Intelligent Agents III, LNAI 1193, 1–20, Springer-Verlag, 1997.
- [5] Hewitt, C.: Offices Are Open Systems. In Bond, A.; Gasser, L. (Hrsg.): Readings in Distributed Artificial Intelligence. Morgan Kaufmann, Los Angeles, CA, 1988., 210–215.
- [6] Rosenschein, J.S.; Zlotkin, G.: A Domain Theory for Task Oriented Negotiation. MIT-Press, 1994.
- [7] Shoham, Y.: Agent-oriented Programming. In: Artificial Intelligence (60), 1993, 51–92.
- [8] Bussmann, S.: Autonome und Kooperative Produktionssysteme. In diesem Heft.
- [9] Fischer, K.: TeleTruck: Ein Online-Dispositionssysteme für Speditionen. In diesem Heft.
- [10] Kirn, S.; Gasser, L.: Organizational Approaches to Cooperation in Multi-Agent Systems. In diesem Heft.
- [11] Mattern, F.: Mobile Agenten. In diesem Heft.
- [12] Müller, J.: Kontrollarchitekturen für autonome, kooperierende Agenten. In diesem Heft.

Prof. Dr. sc. Hans-Dieter Burkhard
 Inst. für Informatik, Humboldt Universität Berlin,
 Axel-Springer-Str. 54a, D-10099 Berlin,
 Email: hdb@informatik.hu-berlin.de
<http://www.ki.informatik.hu-berlin.de>