0 Inhalt

1 Motivation

Täglich entstehen neue WWW-Seiten, wie z.B. Tageszeitungen, Informationen von Institutionen

Probleme:
Eine gezielte Suche nach bestimmten Informationen, da eine systematische Suche sehr schwierig ist (Suchmaschinen als Standardmechnamismus).

Menschlicher Faktor:
Langeweile/Kosten beim Warten auf Antwort, bzw. rigoros WWW Seiten zu lesen

Organisatorischer Faktor:
Es gibt keinen Standard für WWW-Seiten und die Sprache des Web (HTML) ist unstrukturiert.

1 Motivation

Lösung:
Ein Sekretär, der die Interessen des Anwenders versteht, der Uninteressantes versteckt und Interessantes hervorhebt.

Ansatz:
Ein Agent als Sekretär.

2 Begriffserklärung

Der Begriff Agent im informations-technologischen Umfeld.

2.1 schwacher Begriff

Ein Agent ist ein Hard- besser Softwaresystem, das:

- autonom arbeiten kann.
- die Möglichkeit der Kommunikation/Kooperation mit anderen Agenten (und Anwendern) besitzt.
- seine Umgebung wahrnimmt, auf diese reagiert und sie verändert.
- zielgerichtetes Verhalten aufweist.

Lernende Agenten im WWW Baetz / Burkamp

2.2 starker Begriff

Dem Softwaresystem 'Agent' werden zusätzlich menschliche Eigenschaften zugeschrieben:

- mentale Begriffe (Glauben/Überzeugung, Wunsch, ...)
- Rationalität
- Wahrhaftigkeit
- Anpassungsfähigkeit/Lernen

Lernende Agenten im WWW Baetz / Burkamp

3 Entwicklung der A.Theorie

Hierbei steht die Frage im Mittelpunkt, wie Theoretiker Agenten konzeptionieren.

3.1 Semantik

Der Architektur, der Sprache und den Werkzeugen (Tools) muß eine Semantik zugeordnet werden.

Ohne diese kann man keine exakten Vorhersagen über Geschehen und Abläufe treffen. (Endanwender brauchen die Semantik nicht lesen oder verstehen, aber ohne diese kann es keine Sprachweiterentwicklung geben).

3.2 Agenten mit Ziel

Der Ausgangspunkt dieser Theorie liegt im starken Agentenbegriff.

Menschliches Verhalten wird vorhergesagt und für andere leicht verständlich durch die Zuschreibung von Einstellungen erklärt (wie Glauben und Wollen vgl. Janine-Regenschirm-Bsp.)

3.2 Agenten mit Ziel

Der Philosoph D. Dennett unterscheidet zwei Stufen zielgerichteten Handelns:

• Ein ‘first-order’ System
hat Annahmen und Wünsche,
aber nicht über Annahmen und Wünsche.

• Ein ‘second-order’ System
hat auch Annahmen und Wünsche über Annahmen und Wünsche auch anderer Agenten.

3.2 Agenten mit Ziel

Einer Maschine menschliche Eigenschaften zuzuschreiben ist durchaus sinnvoll wenn es hilft, die Struktur der Maschine oder ihr Verhalten zu verstehen (vor allem, wenn die Struktur nur unvollständig bekannt ist).

Prinzipiell kann jedes Objekt auf diese Weise beschrieben werden. (Lichtschalterbsp.)

Bei einfachen Systemen hat dieses Vorgehen offensichtlich wenig Sinn.

3.2 Agenten mit Ziel

Je Komplexer die zu beschreibenden Systeme werden, desto schwieriger wird eine rein mechanische Erkärung, falls sie überhaupt noch Möglich ist (Computersysteme).

Die Erstellung einer Beschreibung mit menschlichen Eigenschaften ist als Abstraktionshilfe sehr gut geeignet

3.3 Theorie der Einstellungen

unser Ziel:
Erschaffen von Computersystemen in mentalen Begriffen

Dazu ist es nötig eine lenkbare Untermenge von Einstellungen und ein Interaktionsmodell zu erzeugen, um Systemverhalten zu erschaffen.

Als Basis dienen zwei Kategorien:
- informative Einstellungen - Annahmen und Wissen
- Vor-Einstellungen - Wünsche, Intentionen, Verpflichtungen, Auswahl

Jede Zielzugriffs-Welt repräsentiert ein Modell der Welt, wie sie nach erreichen des Ziels des Agenten aussehen könnte.

3.4 Dynamik der Agenten

Ein wichtiger Aspekt ist die Dynamik der Agenten.

- Wie sind die Voreinstellungen des Agenten verbunden?

- Wie ändert sich der kognitive Status des Agenten mit der Zeit?

- Wie führen Informationen und Voreinstellungen zu Aktionen?

3.4.1 Intention

[Der Ansatz nach Cohen und Levesque]

Folgende Eigenschaften müssen vorhanden sein, um die Theorie der Intentionen zu erfüllen:

• Intentionen stellen Probleme für Agenten dar, die Wege finden müssen diese zu lösen.

• Intentionen liefern einen Filter um andere Intentionen anzunehmen, die nicht im Konflikt mit den bestehenden Intentionen stehen.

• Agenten verfolgen den Erfolg ihrer Intentionen, und Veranlassen einen neuen Versuch, falls der vorherige fehlschlug.

• Agenten glauben, daß ihre Intention durchführbar ist.

3.4.1 Intention

• Agenten glauben nicht, daß sie ihre Aufgabe nicht erfüllen können.

• Unter gewissen Umständen glauben Agenten, daß sie ihr Ziel nicht erreichen.

• Agenten brauchen nicht alle Seiteneffekte,die bei der Erfüllung ihrer Intentionen auftreten, zu beabsichtigen.

4 Agententypisierung

Wie werden Agenten klassifiziert?

Eine Möglichkeit ist die Unterscheidung anhand der Mobilität der Agenten. Damit ergibt sich folgende Unterteilung:

- **mobile Agenten**
- **fixierte Agenten**

(4.1 Begriffseinführung)
... zuvor führen wir jedoch diverse Begriffe und Fragen ein.

- **Hostsicherheit**
- **Agentensicherheit**
- **Heterogenität des Host**

4.1.1 Hostsicherheit

Welche Risiken “bedrohen” den Host?

- **Sabotageakte** durch Agenten mit “böser” Absicht.
- **Ressourcen-Überanspruchung** durch “gierige” Agenten
- **Seiteneffekte** “defekter” Agenten
- Zeigertypen unterstützende Programmiersprachen bergen die Gefahr der **Korrumpierung** des Speicherraums
- Manche Aktionen sind unter normalen Umständen harmlos, können bei **Fehlnutzung** jedoch schnell gefährlich werden. (z.B. eMail)

4.1.1 Hostsicherheit

Welche Schutzmöglichkeiten gibt es?

• Viele Agenten-Sprachen unterstützen **keine Zeigertypen**.

• UNIX-ähnliche **Zugriffsrechte**

• **Sichere Bibliotheken** die den Dateizugang, die Speicherung, den Prozessraum und vieles mehr zur Verfügung stellen müssen.

• In manchen Agenten-Sprachen kann die **Ressourcennutzung** des Agenten **limitiert** werden. (z.B. TELESCRIPT)

•Einige bieten die Möglichkeit der **Überprüfung der Agentensicherheit** bei dessen Erhalt.

•Konzept der "**Gummizelle**": Die Agenten arbeiten auf einem physikalisch abgetrennten Co-Prozessor in Quaratäne

•Der Host muß "abgestürtzte" Programme **kontrolliert behandeln** können.

•Einführung "**befreundeter**" Agenten.

4.1.2 Agentensicherheit

Welche Sicherheitsaspekte?

• Agenten haben ein Recht auf **Privatsphäre**

•Es besteht die Möglichkeit das die Absicht des Agenten **ausgespäht** und gegen den Auftraggeber verwandt wird.

•Der Agent kann ohne Wissen des Benutzers **modifiziert**, ja sogar **sabotiert** werden.

•Der Daten- und **Programmtransfer** kann über konventionelle kryptische Verfahren sichergestellt werden.

•Um eine **Verfälschung** zu verhindern können digitale Wasserzeichen und/oder Checkcodes verwandt werden.

4.1.3 Heterogenität des Host

Was tun mit diversen Maschinentypen?

Da es viele **verschiede Hardwareplattformen** gibt (PC,SPARC ...) müssen Umgebungen geschaffen werden in denen es möglich ist Agenten unabhängig von der Hardware-plattform auszuführen. ==>

• Benutzung von **Interpretersprachen** oder die Nutzung eines **virtuellen Maschine-Konzeptes** (da Compilersprachen eine Reduktion auf Maschinen-Code durchführen)

• **Dynamische Link-Bibliotheken** müssen in verschiedenen Umgebungen gemeinsame Schnittstellen anbieten. (Ansonsten würden die Agenten bei ohnehin schon geringer Leitungs-bandbreite zu groß)

4.2 Mobile Agenten

Welche Typen mobiler Agenten gibt es?

Es läßt sich eine Typisierung aufgrund des Auslösers und der Art der Mobilität vornehmen.

• **Selbstständige** mobile Agenten

• auf **Anforderung** mobile Agenten

• **Aktive**-Post Agenten

4.2.1 selbstständig mobile A.

Charakterisierung...

unter selbstständig mobilen Agenten verstehen wir Agenten die **selbst** in der Lage sind über ihren Zielort und die dort zu erledigende Arbeit zu **entscheiden**. (z.B. Wieviel eCash diese ausgeben können.)

Solche Agenten werden im allgemeinen mit einer speziellen Sprache programmiert die eine 'Go'-Anweisung anbieten. (z.B. Telescript)

Schlüssel-Begriffe wie **Orte** und **Agenten** kennzeichnen diese Technologie. Agenten können sich innerhalb dieser virtuellen Welt bewegen und Plätze einnehmen. Zu diesem Zweck führt jeder Agent ein Ticket mit **Reiseinformationen** mit sich.

Agenten können miteinander **kommunizieren** und sich **treffen**.

4.2.2 Mobil auf Anforderung

Charakterisierung...

Die Idee hierbei ist, das der Host nur ausdrücklich **angeforderte** Agenten ausführen muß.

Das meist bekannte Bsp. sind in HTML eingebettete JAVA-Applets.

Diese Applets werden mit der Seite neben den Grafiken und dem Text heruntergeladen und dann auf der Maschine des Benutzers ausgeführt.

JAVA selbst ist eine C++ ähnliche Programmiersprache die **keine Zeigertypen** aufweist.

4.2.3 Aktive-Post Agenten

Charakterisierung...

Die Idee hierbei ist, den Agenten sozusagen **Huckepack mit einer eMail** zu versenden.

Bei Erhalt der eMail wird die Dateinanlage **entpackt** und der enpackte Agent wird auf der Zielmaschine durch einen **restriktiven Interpreter** ausgeführt. (Aktive eMail)

Das weitverbreiteste Beispiel für diese Art von Agenten ist die **MIME** Erweiterung der eMail in Verbindung mit **sicheren TCL Skripten**.

4.2.3.1 MIME mit TCL/TK

Beschreibung...

• TCL: die Tool Control Language ist eine Kommandosprache.

• TK: ist ein auf X-Windows basierendes Toolkit, daß die Erzeugung von graphischen Oberflächen sowie Interprozesskommunikation ermöglicht.

==> Die Kombination TCL/TK ist ein einfach zu benutzendes GUI Entwicklungswerkzeug mit interessanten Eigenschaften im Bezug auf Agenten.

•TCL ist eine Interpretersprache

•TCL ist leicht erweiterbar

•TCL/TK kann als Code eingebettet werden

4.2.3.1 MIME mit TCL/TK

Warum sind die Eigenschaften so interessant ?

•TCL ist eine Interpretersprache

==> Plattformunabhängigkeit

•TCL ist leicht erweiterbar

==> Weiterentwickelbarkeit, anpassbar

•TCL/TK kann als Code eingebettet werden

==> Austauschbarkeit.

Also können TCL-Skripten als eine Art Agenten aufgefaßt werden.

Das zentrale Problem war die Sicherheit.
(Die erweiterten Anforderungen führten zu Safe-TCL)

4.3 Fixierte Agenten

Nicht zuletzt aufgrund der Sicherheitsprobleme scheinen mobile Agenten nicht der Weisheit letzter Schluß zu sein.

Eine andere Möglichkeit sind die fixierten Agenten die Programme im traditionellen Sinne sind.

Welche Typen von fixierten A. gibt es?

• Client-seitige Agenten

• Server-seitige Agenten

4.3.1 Client-seitige Agenten

Charakterisierung...

Bei Client-seitigen Agenten bleibt der Agent an die **lokale Maschine** des Benutzers gebunden.

Der Agent fungiert als **Vermittler** zwischen dem Benutzer und dem Internet.

Die **Bewertung** der Informationen aus dem Netz erfolgt **durch das eigene** intelligente **Programm**. (Bsp. MAXIMS, NEWT, WebWatcher)

Vorteile:

• keine Sicherheitsprobleme

• keine Verletzung der Privatsphäre

Nachteile:

• hohes Netzwerk-Verkehr

• Limitierung des Agenten

4.3.2 Server-seitige Agenten

Charakterisierung...

Bei Server-seitigen Agenten ist die "Intelligenz" **an den Server gebunden**.

Die **Bewertung** der Informationen aus dem Netz erfolgt **durch den Anbieter.**
(Bsp. LYCOS, INFO-SEEK, any PERL-Skript)

Vorteile:

• Informationen werden der Informationsquelle zugeordnet

• Aufbau von Benutzermodellen möglich --> Verbesserter Service

• bessere Netzwernutzung

Nachteile:

• Sicherheitsprobleme/Verletzung der Privatsphäre wird durch das Erstellen von Benutzerprofilen möglich

5. Agentenarchitektur

Wir möchten Agenten konstruieren, die alle vorher beschriebenen Eigenschaften aufweisen können. Agentenarchitektur umfaßt daher folgende Themen

• Wie macht man das?

• Welche Soft- und Hardware Strukturen sind dafür geeignet

• Was ist eine angemessener Einsatz bei Aufteilung/Vermischung der Konzepte?

5.1 Schließende Agenten

Der klassische Ansatz zur Konstruktion von Agenten ist es diese als einen bestimmten Teil eines wissensbasierten System zu sehen und alle schon bekannten Konzepte und Methoden dieser Systeme zu übernehmen.

Dieses Paradigma ist als **symbolische KI** bekannt. Die Architektur eines **"beratenden" Agenten** setzt sich dabei aus zwei Komponenten zusammen:

• Der Agent enthält ein explizit repräsentiertes symbolisches **Modell der realen Welt**.

• Er trifft Entscheidungen auf Grundlage **symbolischer Schlüsse**.

5.1 Schließende Agenten

Soll ein Agent in dieser Struktur konstruiert werden, gilt es zwei Probleme zu lösen:

• **Translation**: Wie übersetzt man die reale Welt in akkurate, adäquate symbolische Beschreibungen und schafft dies innerhalb kurzer Zeit.

• **Repräsentation**: Wie können Informationen symbolisch repräsentiert werden und wie konstruiert man einen Agenten, der in Lage ist aus diesen Informationen 'just in time' zu schließen.

Keines der Probleme ist bisher auch nur annähernd zufriedenstellend gelöst.

Einige Probleme modaler Logik scheinen **höchstgradig unentscheidbar** zu sein.

5.1.1 Planende Agenten

Ein Spezialfall der schließenden Agenten sind die planenden Agenten.

Planung ist eigentlich eine Form der automatischen Programmierung:
"Das Design eines Verfahrens, das angewandt ein bestimmtes Ziel erreicht."

Ein **planender Agent** besteht aus:

• einer symbolischen **Beschreibung der Welt**

• einer symbolischen **Bescheibung eines Ziel-Status**

• einer Menge von **Aktionsbeschreibungen** die aus Beschreibungen der mit der Aktion verknüpften **vorherigen und nachfolgenden Bedingungen** bestehen.

Die Algorithmen zur Lösung eines Problemes basieren in erster Linie auf Vergleiche des aktuellen Status mit dem Zielstatus nach Ausführung diverser Aktionen.

5.2 Reaktive Architekturen

Aufgrund der wahrscheinlich unlösbaren Probleme der schließenden Agenten wurden andere Konzepte verfolgt.

Ein reaktiver Agent ist ein Agent dessen Architektur keinerlei zentrale symbolische Logik-Modelle enthält und daher auch nicht auf komplexe logische Schlüsse zurückgreifen muß.

5.2 Reaktive Architekturen

Zentrale Thesen der reaktiven Architekturen:

• Intelligentes Verhalten ist ohne die explizite Repräsentation der Welt und komplexe Schlüsse im Sinne der symbolischen KI möglich. [BROOKS]

• Intelligenz ist eine **entstehende** Eigenschaft bestimmter komplexer Systeme [BROOKS]

• Intelligentes Verhalten entsteht durch **Interaktion** des Agenten mit seiner Umwelt [BROOKS]

Grundlegende Beobachtungen sind:

• Die meisten Aktivitäten sind **Routine ohne neue Schlüsse**. [AGRE & CHAPMANN]

• Die meisten Aufgaben lassen sich durch ein **variierte Routine-Lösung** bewältigen. [AGRE & CHAPMANN]

5.3 Situative Automaten

Die Strategie der situativen Automaten ist es, einen Agenten in einer **regel-ähnlichen Sprache** zu spezifizieren und diese Beschreibung in eine **Digitale-Maschine** zu kompilieren die die Spezifikation erfüllt.

Diese Maschine arbeitet in **beweisbaren Zeitgrenzen.**

Jedoch:

Je ausdrucksvoller die Beschreibungssprache wird, desto schwieriger wird die Kompilierung. Einige Forschungsergebnisse sagen aus, daß die **Kompilierung ab einer gewissen Komplexitätstiefe der Sprache nicht durchzuführen ist**.

5.4 Hybride Architektur

Ein hybrider Architekturansatz konstruiert den Agenten aus zwei (oder mehr) Teilen:

• einem beratenden Teil

• einem reaktiven Teil

Der beratende Teil enthält ein symbolisches Welt-Modell mit dem Pläne entwickelt und Entscheidungen getroffen werden.

Der reaktive Teil ist in der Lage ohne komplexe Schlüsse auf Ereignisse zu reagieren. (Häufig besitzt die reaktive Komponente Vorrang vor der Beratenden)

Die hybriden Agenten werden hierarchisch in einem Schichten-Modell aufgebaut. Höheren Schichten werden Informationen höherer Komplexität zugeordnet.

Das größte Problem dieser Architektur ist die Interaktion zwischen den Schichten und Teilen.

5.5 Touring Maschinen

Agenten in der Architektur der Touring-Maschinen bestehen aus drei Subsystemen:

• Einem Wahrnehmungssystem

• Einem Handlungssystem

• Einem Kontrollsystem

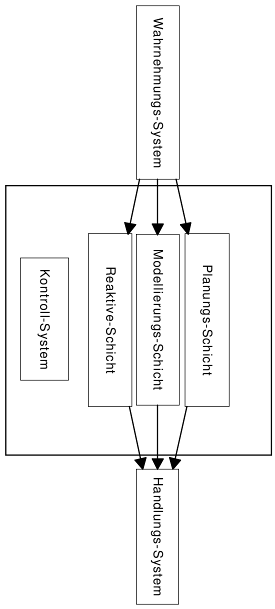
Das Kontrollsystem setzt sich aus drei Kontroll-Schichten zusammen:

• einer reaktiven Schicht

• einer planenden Schicht

• einer modellierenden Schicht

5.5 Touring Maschinen



Lernende Agenten im WWW Baetz / Burkamp

5.5 Touring Maschinen

Die **reaktive Schicht**
wird als eine Menge situativer-Regeln implementiert.

Die **planende Schicht**
erzeugt Pläne und wählt Aktionen zur Erreichung des Ziels aus.

Die **modellierende Schicht**
enthält symbolische Repräsentationen des "Bewußtsein" des Agenten im Bezug zu seiner Umgebung.

Alle drei Schichten kommunizieren miteinander und sind in einen Kontroll-Rahmen eingebettet.