



Konzepte der AI

Intelligente Agentensysteme

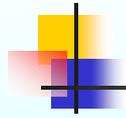
DBAI
IBD
DBAI

Robert Baumgartner



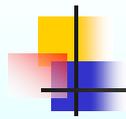
Inhalt

- Einleitung und Überblick
- Kollaborierende Agenten
- Interface Agenten
- Mobile Agenten
- Reaktive Agenten
- Hybridagenten / Heterogene Agentensysteme



Was ist ein Agent?

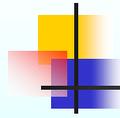
- keine Einigkeit über allgemeine Definition
- viele Arten
physikalische Welt (Roboter), in Computernetzwerken (Softbots)
- für bestimmte Aufgaben (Taskbots)
haben Tasklevelskills und auch Communicationskills, sowie ein gewisses Wissen auf das diese beiden Komponenten zugreifen



Definition von Agenten

laut Woolridge und Jennings:

- **Autonomie** Handlungen selbst entscheidbar
- **Sozialfähigkeit** Agent Communication Language
- **Reaktionsfähigkeit** Wahrnehmung der Umgebung
- **Aktionsfähigkeit** Handlungen aus eigener Initiative, vorausblickendes Verhalten
- **Strenger Agentenbegriff** menschenähnliche Intelligenz, Wissen, Überzeugungen, Absichten, Verpflichtungen



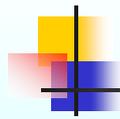
Definition von Agenten (2)

nach E. Horn:

Ein Software-Agent ist ein Computerprogramm, das **autonom** und **zielgerichtet** im Namen einer/anstelle einer Organisation arbeitet. Ein Agent hat ein eigenes Ausführungsprofil und kann während seines Lebenszyklus Aufgaben auf eigene Initiative (und in Kooperation mit anderen Agenten) ausführen. Ein Agent hat eine Eigenschaft die festlegt, wer zu seiner Benutzung autorisiert ist.

nach D. Dennett:

Ein Agent ist ein Objekt dessen Eigenschaften sich als intentionale Standpunkte und Haltungen zu den Problemstellungen darstellen lassen (zwei Kategorien: Verhältnis zu Informationen, Handlungsbeeinflussende Haltungen)



Typologie von Agenten

- **Klassifikation nach Ausführungsort**
Desktop, Internet, Intranet, ...
- **Klassifikation nach Anwendungstyp**
Informationssuche, Filter, Trainingsassistenten
- **Klassifikation nach Architekturprinzipien**
Lernende Agenten, Neurale Agenten, Mobile Agenten



Typologie von Agenten (2)

- statisch vs. mobil
- reaktiv vs. deliberativ (symb. Modell der Realität)
- autonom, kooperativ/konkurrierend, lernend
 - kooperativ und lernend: kollaborierende lernende Agenten
 - autonom und lernend: Interface-Agenten
 - kooperativ und autonom: kollaborierende Agenten



Relevante Gebiete

- **Computationale Intelligenz**
Wissensbasierte Systeme, Neurale Netze, Intentionale Systeme, Reasoning Theory
- **Software Engineering**
Objects, Image and Speech Processing, High Level Event Inferencing, On-Line Monitoring
- **Human Interface**
Intelligent Tutoring, Interactive Experiments, Cognitive Engineering, User Modeling

Deliberative Agenten

Wissensbasierte Agenten werden auf drei Ebenen beschrieben:

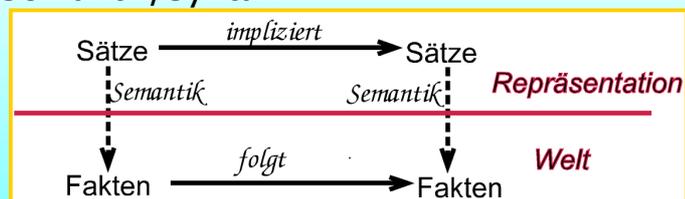
- abstrakte Wissensebene (zu sagen was er weiß)
- logische Ebene (Beschreibung mit logischen Sätzen)
- Implementationsebene (wie die logischen Sätze in der Systemarchitektur implementiert sind)

Wissensrepräsentationssprache soll einfachen Zugang zur Wissensbeschreibung geben (deklarativer Ansatz des Systementwurfs)

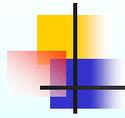
Wumpus-Spielwelt

Deliberative Agenten (2)

- Logik (Aussagenlogik, Prädikatenlogik,...; Prolog)
- Semantik/Syntax

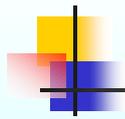


- Situationskalkül (um Änderungen zu beschreiben; in Präd.logik)
- Zielbasierter Agent (Suche, Planen, Inferenz)



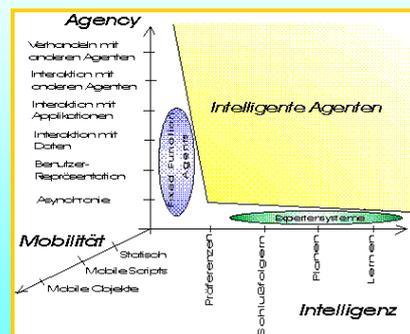
Klassifikation von Software-Agenten

- Kollaborierende/konkurrierende Agenten
- Interface Agenten
- Mobile Agenten
- Reaktive Agenten
- Hybride Agenten
- Heterogene Agentensysteme
- Informationsagenten (siehe Teil 2)



Klassifikation von Software-Agenten (2)

- Regelsysteme
- Expertensystemansatz
- Selbstanpassungsansatz
maschinelles Lernen, adaptives Lernen,
Data Mining, nichtsymbolisches Lernen
(Statistische Analyse, Neurale Netze),...





Kollaborierende Agenten

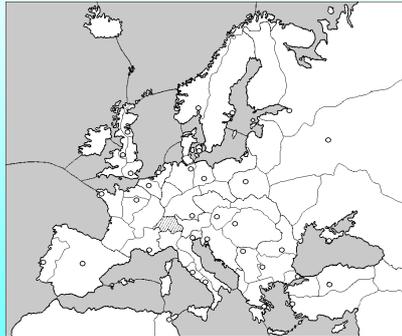
- Agentenbegriff ursprünglich in Multi-Agenten-Systemen geprägt
- Carl Hewitts concurrent actor model, 1977
Ein Actor ist ein computationaler Agent mit einer Mail-Adresse und einem bestimmten Verhalten - er kommuniziert über Botschaften mit anderen Agenten und simultan werden Aktionen durchgeführt.
- Jeder Agent ist autonom und es gibt Verhandlungen/Gespräche zwischen Agenten zur Abstimmung aufeinander
- Verhandlungsmechanismen müssen entwickelt werden



Distributed AI

- Das Ensemble der Agenten erreicht Dinge die ein einzelner Agent nicht erreichen könnte
- Anstreben eines gewissen maximalen Wertes durch Verhandlung bzw. Informationsaustausch
- Gebiete der DAI
 - Distributed Problem Solving
gemeinsames Erarbeiten (Bsp. Air Traffic Control, Roboterfußball)
 - Concurring Multi-Agent Systems
konkurrierende Agenten mit notwendiger temporärer Koalitionsbildung (Bsp. Diplomacy)

Diplomacy als MAS



- Sieben Mächte Europas kämpfen um die Vormachtstellung
- Ein Sieg kann nur durch zeitweilige Bündnisse realisiert werden
- Spiel mit unvollständiger Information, simultane Bewegung, kein Zufallselement
- Ansätze von Kraus und Loeb: *lokale Agenten zur Aufgabenteilung, Verhandlungsalgorithmen, Bewertungen etc.*

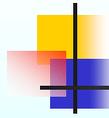
Robotic Soccer

- kollaborierende Hardware-Agenten
- Probleme wie Selbstlokalisierung und Weltmodellierung
- Robocup: <http://www.robocup.org>
- SC Freiburg: <http://www.informatik.uni-freiburg.de/~robocup>



[Robocup-Video 1](#) ↩

[Robocup-Video 2](#) ↩



Probleme und Methoden der DAI

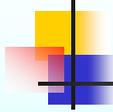
- **Resource Allocation Problem**
Die Frage wie und von wem die vorhandenen Ressourcen genutzt werden
- **Task Distribution Problem**
Die Frage wie vorhandene Aufgaben auf verschiedene Agenten aufgeteilt werden

- **Methoden:** blackboard (bzgl. Kommunik.), contract-net (bzgl. Koordination), partial global planning
- **Beispiele:**
 - Verhandlungen zweier voll informierter Agenten
 - Modell der alternierenden Angebote
 - Verwendung des Nash-Gleichgewichtes (Spieltheorie)
 - Diverse Grundregeln denen Agent folgt (Bsp. er bevorzugt eine frühere Abmachung einer späteren)



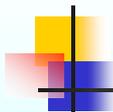
Techniken der DAI (nach Kraus)

Multi-Entitäten-Techniken	in DAI für:
Spieltheorie: Strategische Verhandlungsmodelle	<i>Verhandlung für Aufgabenteilung und Ressourcenzuordnung in MA</i>
Spieltheorie: Theorien der Koalitionsbildung	<i>Koalitionsbildung in MA</i>
Spieltheorie: Principle-Agents Modelle	<i>Vertragsbildungen in MA</i>
Physik: Klassische Mechanik (Vielteilchensysteme)	<i>Zielerfüllung in großen DPS-Umgebungen</i>
Operations Research: Set Partition&Coalition Problem	<i>Koalitionsbildung in DPS</i>
Operations Research: Queuing Networks	<i>Aufgabenzuordnung in DPS</i>
Behavioral Sciences: Verhandlungsleitfäden	<i>Diplomatische Verhandlungen</i>
Behavioral Sciences / Logik: Persuasion Modelle	<i>Argumentation</i>
Beh. Sciences / Logik / Entscheidungstheorie: Fokalfpunkte	<i>Kooperation ohne Kommunikation</i>
Philosophie/Logik	<i>Kollaborative Pläne</i>



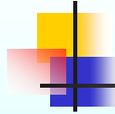
KQML

- Knowledge Query and Manipulation Language
Kann als Agent Communication Language (ACL) zur kooperativen Problemlösung benutzt werden
- Kommunikationsprotokoll für wissensbasierte Systeme
Wissen und Information zur Laufzeit austauschbar
- unabhängig von Transportmechanismen
(wie TCP etc.)
- schichtorientiert



KIF

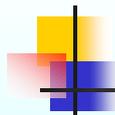
- Knowledge Interchange Format
- Eine ACL ist ein KQML-Ausdruck in den ein KIF-Satz eingebettet ist
- KIF ist Präfixversion der Prädikatenlogik
- kontextunabhängig
- Metawissen damit ausdrückbar
wie: John interessiert sich für die Temperatur der Stadt, nicht den Satz über die Temperatur der Stadt; (*interested joe '(temperature, wien, 1.1.2000)'*)



KQML/KIF

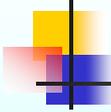
Drei Schichten:

- **Content Layer** – *Ein KIF Satz*
- **Message Layer** –
Infos über Content Language und Welt, Deklarationen
- **Communication Layer** –
Sender, Empfänger, Kommunikationsmodus



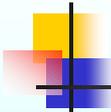
ZEUS Agent Building Toolkit

- verwendet visuelle Programmierung
- verwendet Standardtechnologie
(Java, Swing, TCP/IP, KQML,...)
- **Komponenten**
Agent Component Library, Agent Building Tools, Visualisation Tools
- <http://www.labs.bt.com/projects/agents/zeus/>



Bestandteile eines ZEUS-Agenten

- **Definitionslayer**
Reasoning Fähigkeiten, Ziele, Wünsche, Wissen, Einstellungen
- **Organisationslayer**
Beziehung zu und Wissen über andere Agenten
- **Koordinationslayer**
soziale Entität, verwendet Koordinations- und Verhandlungstechniken, Kommunik.protokolle



Interface-Agenten

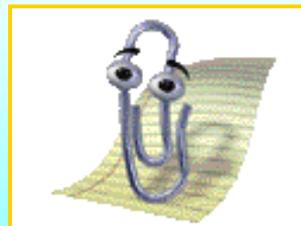
- **Assistent eines Users**
benötigt keine Agentenkommunikationssprache
- **OS-Agenten, Applikationsagenten**
zur aktiven Userhilfe
- Ein **lernender Interface-Agent** ist ein Agent der machine-learning Techniken dazu verwendet um ein pseudo-intelligentes Interface für Aktionen einzurichten
- **Beispiele:** Memory Aids, Kalender-Agenten, Filter, Guide/Hilfestellungen, Ecommerce-Interface-Agenten

Interface-Agenten: Lernen

- durch Beobachten und Imitieren des Users
- durch negatives und positives Userfeedback
- durch das Erhalten expliziter User-Anweisungen
- evtl. Fragen anderer Agenten
(Kommunikation auf Fragen beschränkt)

Der Office Agent

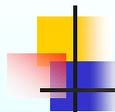
- **autonom**
arbeitet ohne direkte Nutzeranweisung im Hintergrund; kann seinen internen Zustand nur tw. kontrollieren
- **soziales Verhalten**
nur Kommunikation mit User
- **Reaktivität**
beobachtet Userverhalten, kommuniziert durch Userschnittstelle, reagiert auf Userfehler, Tips zur besseren Nutzung
- **aber kein vorausschauendes Verhalten**





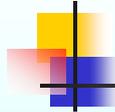
Mobile Agenten

- Mobilität ist weder hinreichend noch notwendig für Agentenstatus
- Wissen ist nicht lokal vorhanden
- Der Request erhält den auszuführenden Code
- Ort wird zu Designkriterium
- Agent entscheidet über Zielort
- Bsp: Postscriptdateien
hier wird der Code an den Drucker geschickt und direkt dort ausgeführt



Remote Procedure Calling vs. Remote Programming

- RPC – Ein Computer kann Prozeduren auf einem anderen Computer aufrufen (Protokoll)
Bsp. Löschen von Files auf Server die mindestens zwei Monate alt sind: Ein Useraufruf ist für Daten nötig, werden n Dateien gelöscht, so muß der User **(2n+1)** Botschaften erhalten bzw. senden
- Remote Programming – die auszuführenden Prozeduren können vom Agenten mitgenommen werden
benötigt im obigen Bsp. nur **eine** Botschaft die den Agenten von einem zum anderen Computer transportiert. Der Agent entscheidet on-site welche Dateien gelöscht werden sollen
- Computer als Ansammlung von "Plätzen" für mobile Agenten



Mobile Agenten (2)

- **Vorteile**
 - bessere Lastverteilung
 - schnellere Ausführungszeit
 - nimmt eigenes Wissen mit
- **Remote Evaluation**
nur Code wird verschickt, Sender ist Initiator, Code wird am Ziel ausgeführt (Bsp.: PS)
- **Code on Demand**
nur Code wird verschickt, Empfänger ist Initiator (Bsp. Applets)

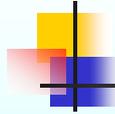


Sprachen und Plattformen

- Telescript (General Magic)
- Nachfolger Odyssee
- ObjectSpace Voyager
- IBM Aglets
- Mitsubishi Concordia
- ...

*Konzept von Places für logische Strukturierung
eines verteilten Systems wird verwendet (Travel, Meetings, Connections,...)*

Agent Orientated Programming: Programmierung in intentionalen Begriffen (Shoham)



Reaktiv vs. Deliberativ

- **Deliberativ**

ausgehen von einem wissensbasierten System (symbolische AI) ergibt deliberative Architektur. Der Reasoning-Prozeß findet in einem Modell der Realität statt. Haben Beliefs, Desires, Goals, Intentions, Plans.

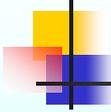
- **Reaktiv**

kein zentrales symbolisches Modell der Realität in dem agiert wird. Vorteil: niedriger Rechenaufwand; Systeme sind dennoch in manchen Fällen gut verwendbar



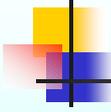
Reaktive Agenten

- Subsumption Architektur: Hierarchie von Verhaltensregeln
- Stimulus-Response zu momentanem Umgebungszustand
- sehr einfache Struktur, aber komplexe Verhaltensmuster wenn ein Ensemble von Agenten betrachtet wird
- Emergente Funktionalität (kein a priori-Plan, es ergibt sich aus vielen Agenten ein intelligenter Agent)
- Aufgabenteilung: Sammlung von autonomen Modulen für bestimmte Aufgaben (minimale Kommunikation nötig)
- operieren auf rohen Sensordaten; nicht in symbolischer Repräsentation



Bsp.: Brooke's Subsumptionarchitektur

- Eine Menge von Modulen basierend auf erweiterten endlichen Automaten
- Diese agieren wenn das Inputsignal einen bestimmten Wert überschreitet (wie inhibitorische und exhibitatorische Potential)
- Die Module sind in Layer gruppiert; jedes Layer hat vorgegebenes Verhalten
- ermöglicht gewisses Selbstorganisationsverhalten
- Situated Automata: neuerer Ansatz basierend auf einem modallogischen Formalismus



Hybridagenten / Heterogene Agentensysteme

- **Hybridagenten**
Kombination aus mehreren Agenttypen; z.B. Agent basierend auf der reaktiven und kollaborativ (deliberativen) Philosophie – bringt Geschwindigkeit und gleichzeitig gutes Langzeitverhalten; z.B. mobiler Agent mit User-Interface
- **Heterogene Agentensysteme**
Agentensystem aus verschiedenen Typen von Agenten; benötigt ACL (KQML, KIF) und Kommunikationsarchitektur