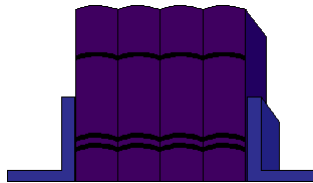


Agententechnologie

Ein Software-Agent ist ein Programm, das in einer vernetzten Umgebung selbständig Aufgaben durchführen kann.

Wie baut man Software auf, die nicht nur Aufträge ausführen kann, sondern auch eigene Entscheidungen trifft?



W. Brenner, R. Zarnekow, H. Wittig
Intelligente Software-Agenten
Springer, 1998

N.R. Jennings, M.J. Wooldridge (Hrsg.)
Agent Technology
Springer 1998

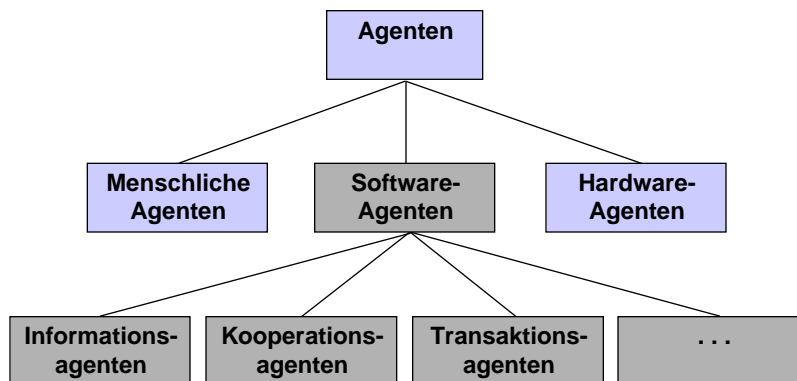
P. Maes
Designing Autonomous Agents
MIT/Elsevier, 1994

1

Kategorien von Agenten

Hier von Interesse: Software-Agenten

"Agent" ist Metapher für Eigenschaften menschlicher Agenten



2

Agent für Büroerwerb

Der Käufer eines Unternehmens überträgt Teilaufgaben beim Erwerb von Büroerwerb auf einen Software-Agenten:

- Anbieter von Büroerwerb finden
- Sortiment von Anbietern auf gewünschte Materialien hin überprüfen
- Angebote beschaffen
- Käufer über Angebote informieren

Erweiterte Aufgaben:

- Markt beobachten, Preisbewegungen registrieren
- Kaufzeitpunkte vorschlagen
- Lagerbestände im Unternehmen verfolgen
- Selbständig für Nachschub sorgen

3

Einige Agenten-Anwendungen

- Industrielle Anwendungen
 - Prozeßsteuerung
 - Produktion
 - Luftverkehrskontrolle
- Kommerzielle Anwendungen
 - Informations-Management
 - E-Commerce
 - Business Process Management
- Medizinische Anwendungen
 - Patientenüberwachung
 - Gesundheitsfürsorge
- Unterhaltung
 - Spiele
 - Interaktives Theater und Kino

4

Wann ist Agenten-Technologie nützlich?

Komplexe Systeme

Beispiele: Roboter, Produktionsautomatisierung, Katastrophenplanung

- Dekomposition in Agenten bietet Modularisierung und Abstraktion
- Metapher einer Gesellschaft kooperierender Agenten

Offene Systeme

Beispiele: Internet, Reservierungssystem, Auskunftssystem, E-Commerce

- Struktur eines offenen Systems ändert sich dynamisch
- Systemkomponenten sind heterogen
- Kooperation durch Aushandeln (Negotiation) statt Auftragsvergabe

Partnerschaftliche Systeme

Beispiele: Beratungssysteme, Handelspartner, Manager

- vage, unpräzise Auftragsbeschreibung
- proaktives Verhalten, eigene Vorschläge
- paßt sich an veränderte Aufgaben und veränderte Umgebung an

5

Grundelemente einer Agentenarchitektur

Agent muß über sein Umfeld Bescheid wissen

"Weltwissen", "Domänenwissen", "Weltmodell"



BELIEFS

Agent muß seine Möglichkeiten kennen

"Ressourcen", "Fähigkeiten", "Skills"



DESIRES

Agent muß seinen Zweck kennen

"Ziele", "Wünsche"



INTENTIONS

Agent muß Aktivitäten verfolgen

"Pläne", "Absichten"

6

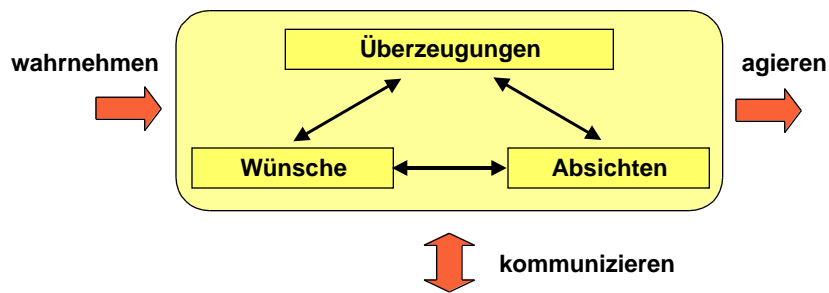
BDI-Architektur

Beliefs = Wissen, Überzeugungen

Desires = Wünsche, (langfristige) Ziele

Intentions = Absichten, Vorhaben

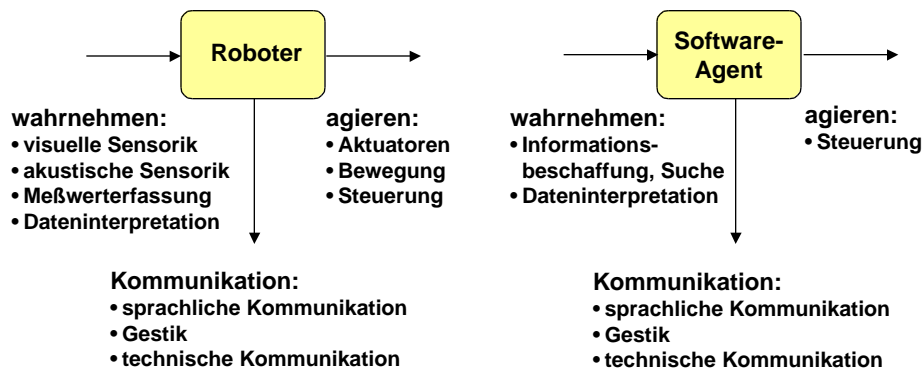
"deliberative agent"
(überlegender, abwägender Agent, BDI-Agent)



7

Roboter vs. Software-Agenten

- BDI-Architektur ist Metapher für Robotik und Agenten-Technologie
- Hauptunterschiede entstehen durch unterschiedliche Welten und unterschiedliche Interaktionsmöglichkeiten



8

Anforderungen an BDI-Agenten

Komponenten eines BDI-Agenten müssen (je nach Anwendung) komplexe Funktionalität unterstützen. Dafür gibt es teilweise umfangreiche Methodenschätze aus eigenständigen Forschungsbereichen.

BELIEFS - Verwaltung von heterogenem, veränderlichen Wissen über das Tätigkeitsfeld des Agenten

Objekt-orientierte Datenmodelle

DESIRES - heterogene Zielvorgaben mit veränderlicher Priorität

Wissensrepräsentation

INTENTIONS - Aktionen planen, die zur Erfüllung von Zielen dienen

Operations Research

KOMMUNIZIEREN - mit Menschen und kooperierenden Agenten in jeweils geeigneter Kommunikationsform

KI-Planungstechniken

WAHRNEHMEN - multimodale Informationen erfassen und interpretieren, eigenes Wissen ändern oder erweitern

Multiagentensysteme

AGIEREN - in das Umfeld eingreifen, zielorientierte Aktionen planmäßig durchführen

Bild- und Sprachverstehen

Robotik

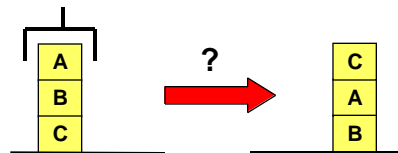
9

Planen

Planen ist zentrale Kompetenz eines BDI-Agenten:

Planen ist das Entwerfen einer Aktionsfolge, mit der eine Startsituation in eine gewünschte Zielsituation überführt werden kann

Veranschaulichung von Planungsproblemen anhand der Blockswelt:



Startsituation:
on (A, B), on (B,C), on (C, Table)

Elementare Aktion:
move (x, y)

Planungsverfahren sind Forschungsgegenstand in mehreren Disziplinen:

- Künstliche Intelligenz (KI)
- Operations Research
- Automatisierungstechnik

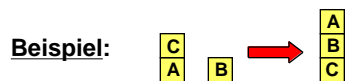
10

Planen als Suchen

Suchproblem wird definiert durch

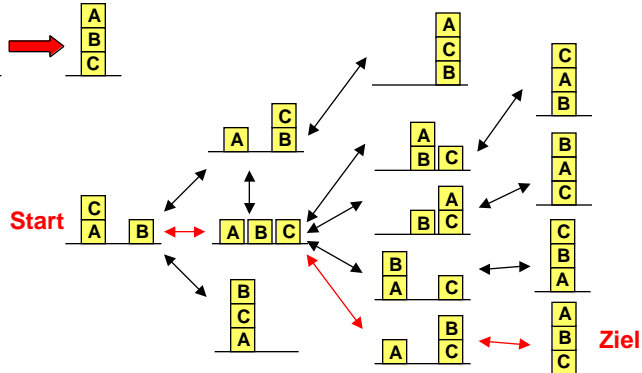
- Zustände (speziell: Startzustand, Zielzustände)
- Operatoren zur Zustandstransformation
- Suchstrategie

Vergleiche
Graphsuche
in Teil 1 von P3!



Suche ergibt Plan:

move (C, Table)
move (B, C)
move (A, B)



11

Planen im STRIPS-Paradigma

- STRIPS ist frühes Planungssystem der KI (Fikes und Nilsson 1971)
- Grundideen sind Bestandteil vieler moderner Planungssysteme

Hauptunterschiede zu reinen Suchverfahren:

1. Weltmodell ist Menge von Prädikaten
2. Anwendbarkeitsbedingungen und Wirkungen von Aktionen werden explizit angegeben:

PRECONDITION-Liste P
DELETE-Liste D
ADD-Liste A

3. Mittel-Zweck-Analyse (means-end-analysis) liefert geeigneten Operator

Beispiel

Beschreibung der Aktion move(x, y):

move(x, y)
P: clear(x), clear(y)
D: on(x, z), clear(y)
A: on(x, y), clear(z)

12

Probleme beim Planen

Qualifikationsproblem:

Unter welchen Vorbedingungen kann eine Aktion ausgeführt werden?

Rahmenproblem (frame problem):

Was sind die Effekte einer Aktion?

Beispiel: Reiseplanung

Operator "fahre_mit_eigenem_auto"
ist anwendbar wenn

- man eigenes Auto besitzt
- Auto nicht defekt ist
- Auto nicht gestohlen ist
- Autoschlüssel zur hand ist
- man autofahren kann
- Autofahren nicht verboten ist
- kein Erdbeben stattfindet
- das Auto nicht explodiert
- ...

Operator "fahre_mit_eigenem_auto"
hat Effekte:

- Ortsveränderung von Auto und Ladung
- Verbrauch von Benzin
- Ausstoß von Abgasen
- Abnutzung des Autos
- Abnutzung der Straße
- Behinderung von Verkehrsteilnehmern
- Lärmbelästigung
- Auslösen eines Erdbebens
- ...

13

Mensch-Agenten-Kommunikation

Bisher:

Mensch-Rechner-Kommunikation ist an Rechner angepaßt
(z.B. Kommandosprache, Parametereingabe unter Rechnerkontrolle)

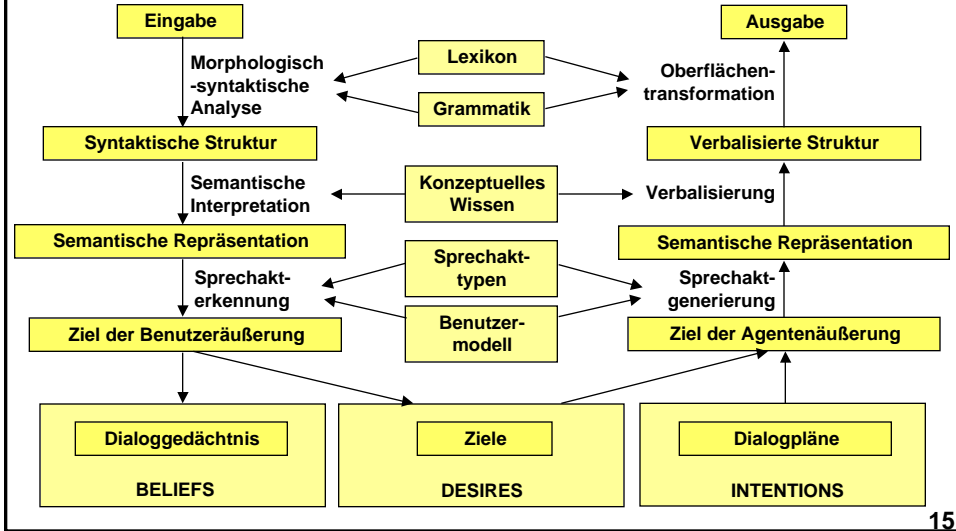
Ziel:

Mensch-Agenten-Kommunikation mit menschlichen Kommunikationsformen:

- Begriffssystem der natürlichen Sprache
 - flexibler Satzbau
 - Dialogkontext
 - Hintergrundwissen für Sprachverständnis
 - akustische Eingabe
 - Gestik
- } Sprachtechnologie

14

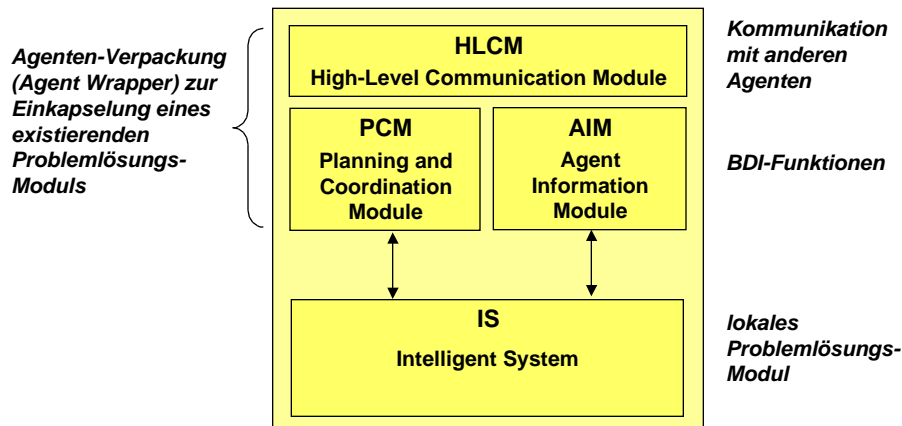
Sprachverarbeitung durch einen Agenten



15

ARCHON-Agentenarchitektur

Ergebnis eines Kooperationsprojektes von 13 europäischen Unternehmen
 Typisches Anwendungsszenario: Kooperation lokaler Expertensysteme bei der Störfallanalyse in Versorgungsnetzwerken (z.B. Stromversorgung)



16

Reaktive Agenten

- Agenten reagieren durch Verhaltensmodule unmittelbar auf ihre Umgebung
- kein internes Weltmodell

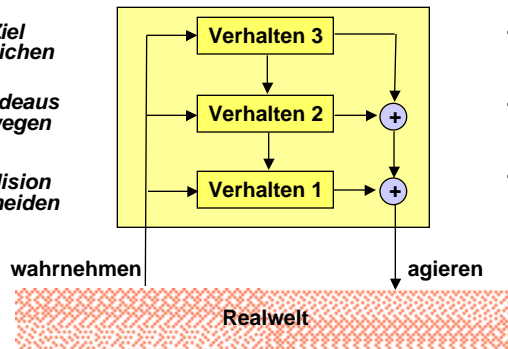
Subsumptionsarchitektur nach Brooks (1986):

Beispiel:

Ziel erreichen

geradeaus bewegen

Kollision vermeiden



- Jedes Verhalten ist (fast) eigenständiger Prozeß
- Verhalten i "subsumiert" (umfaßt) Verhalten $i-1$
- Realwelt übernimmt Rolle eines "Weltmodells"

17

Kooperierende Agenten

Wie können individuell motivierte, unabhängig agierende Software-Agenten zur Erreichung von Zielen kooperieren?

1. Agenten kooperieren durch komplementäre Aufgabenverteilung

Gemeinsames Ziel wird implizit erreicht, wenn jeder Agent seine Aufgabe ausführt.

Beispiel: Blackboard-System

2. Agenten verwenden Kooperationsmethoden

Kooperationsmethoden erlauben es, Multi-Agentenpläne zu schaffen und durchzuführen.

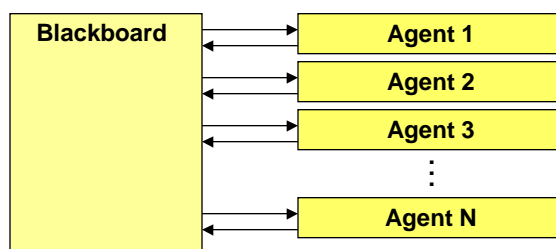
Kooperationsmethoden sind abstrakte, für eine breite Klasse von Anwendungen geeignete Schemata.

Beispiel: Kontraktnetz

18

Blackboard-System

- Kommunikation und Koordination über eine Kommunikationstafel (Blackboard)
- Agent liest Eingangsdaten vom Blackboard und schreibt Ergebnisse auf das Blackboard
- Agent wird tätig, wenn Eingangsdaten bereitstehen



19

Aufbau eines Blackboard-Agenten

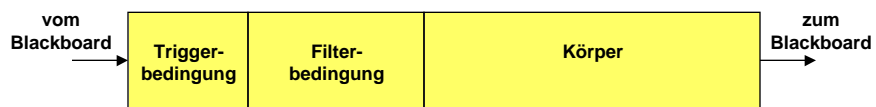
Ein Blackboard-Agent hat meist die Rolle eines Spezialisten:

- spezielles Wissen
- spezielle Fähigkeit
- ein einziges Ziel
- nur an seinen Eingabedaten interessiert

Ein Blackboard-Agent wird aktiv, wenn der Inhalt des Blackboards die Triggerbedingung des Agenten erfüllt (z.B. *Sind neue Daten in einer bestimmten Blackboard-Abteilung?*).

Ein getriggert Agent prüft das Blackboard mit seiner Filterbedingung (z.B. *Sind die neuen Daten für seine Aufgabe geeignet?*) und liest seine Eingabedaten.

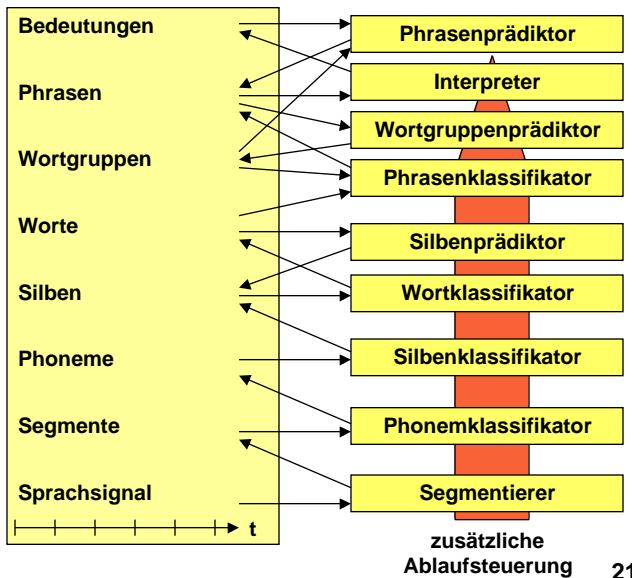
Im Körper des Agenten werden die Eingabedaten bearbeitet. Das Ergebnis wird auf dem Blackboard abgelegt.



20

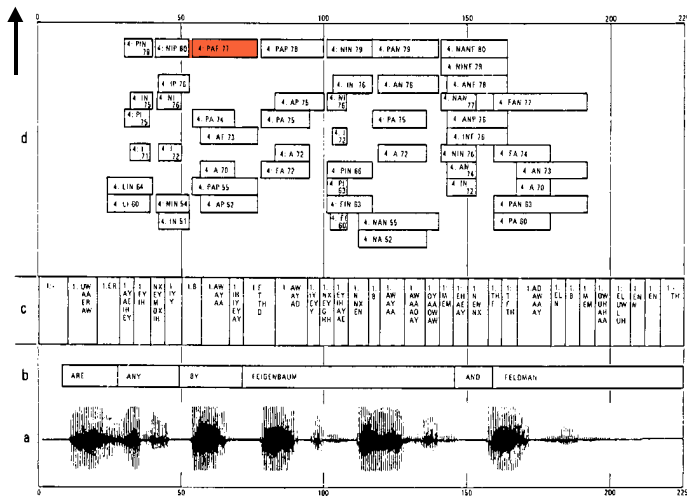
Sprachverstehen mit einem Blackboard-System

- Blackboard ist nach Repräsentationsebenen und entlang Zeitachse unterteilt
- Einträge sind Hypothesen über Abschnitte des Sprachsignals
- Ablaufsteuerung erforderlich, wenn Agenten gemeinsame Ressourcen (z.B. Prozessor) verwenden



Einträge in ein Blackboard (1)

untere Verarbeitungsebenen

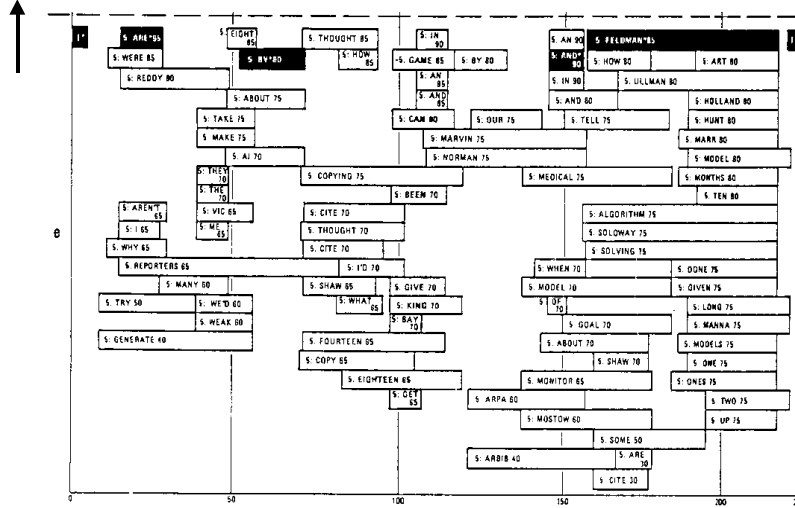


Bewertung (0 .. 100)

Hypothese
Schritt Nr.
4: PAF 77

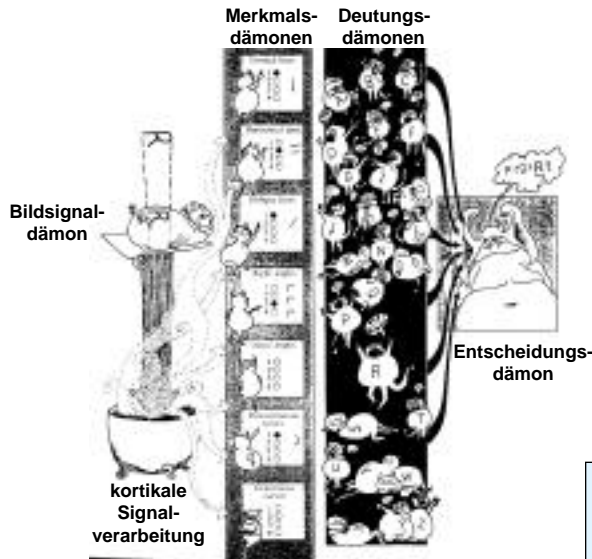
Einträge in ein Blackboard (2)

obere Verarbeitungsebenen



23

Dämonenmodell aus der Kognition



Zeichenerkennung in einer Multi-Agenten-Architektur:

- Agenten (Dämonen) als Spezialisten
- Kommunikation über gemeinsame Datenbereiche

aus:
P.H. Lindsay, D.A. Norman,
Human Information Processing
Academic Press, 1977

24

Nachrichtenaustausch zwischen Agenten

Kommunikationstheorie (Sprechakttheorie) bietet nützliche Abstraktionen für Nachrichtenaustausch:

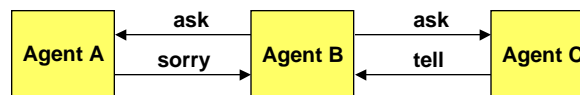
Nachrichtenaustausch = zielgerichtete Aktion

"Sprechakt" Akt des Nachrichtenaustausches

"Sprechakttyp" Unterscheidung von Sprechakten nach Art des Ziels (z.B. fragen, antworten, anweisen, informieren, ...)

Ein Kommunikationsprotokoll legt Regeln fest, nach denen Sprechakte in einem Dialog aufeinander folgen dürfen.

Beispiel:



25

Kommunikation mit KQML

KQML (Knowledge Query and Manipulation Language) ist verbreitete Sprache für Nachrichtenaustausch in Multi-Agentensystemen.

Struktur einer KQML-Nachricht:

```

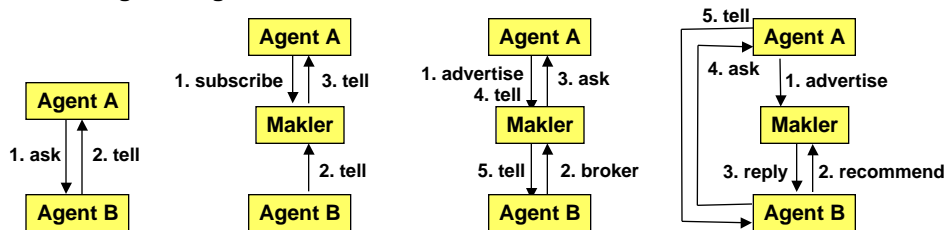
    (<Sprechakttyp>
     :content <Aussage/Sprechakt>
     :sender <Name>
     :receiver <Name>
     :language <text>
     :ontology <text>)
  
```

Beispiel:

```

    (ask-one
     :content (PRICE IBM ?price)
     :sender news-server
     :receiver stock-server
     :language LPROLOG
     :ontology NYSE-TICKS)
  
```

Wichtige Dialogstrukturen:



26

Kooperation zwischen Agenten

Sprechakte können die Kooperation zwischen Agenten zum Lösen von gemeinsamen Aufgaben strukturieren.

Sprechakttypen für Kooperationen:

propose
accept
reject
refine
modify
inform
query

Sprechaktobjekte für Kooperationen:

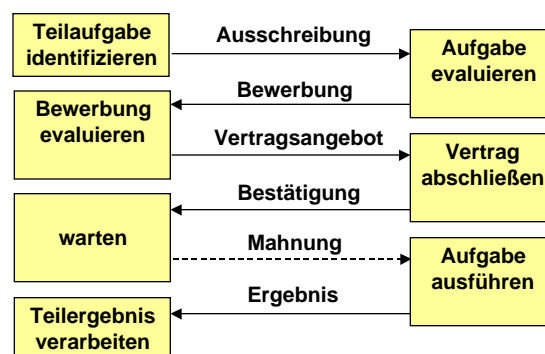
goal
goal breakdown
plan
commitment
task assignment
resource allocation
schedule

Kooperationsprotokolle legen geeignete Abläufe für den Austausch von (Kooperations-) Sprechakten fest.

27

Kontraktnetz-Protokoll

Schematisierter Verhandlungsablauf zwischen Manager und Agenten zur Vergabe von Teilaufgaben:



Jeder Agent kann seinerseits Teilaufgaben als Manager vergeben

28

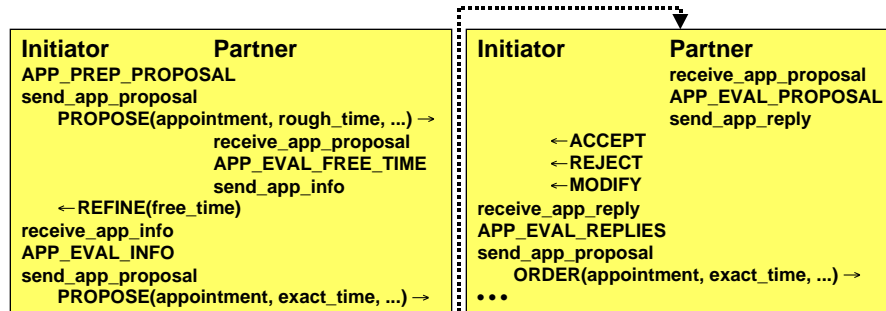
Terminkalender-Agenten

Klassische Anwendung der Agententechnologie:

Gemeinsamen Termin für mehrere beteiligte Menschen verabreden

- Jeder Beteiligte besitzt persönlichen Terminkalender-Agenten
- Verabredungen erfolgen durch Kooperation der Agenten

Aktionen eines Appointment Management Systems:

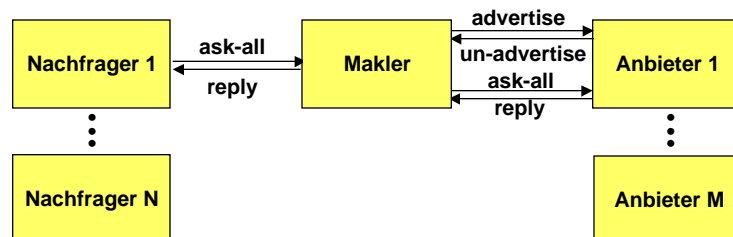


29

Makler-Protokolle

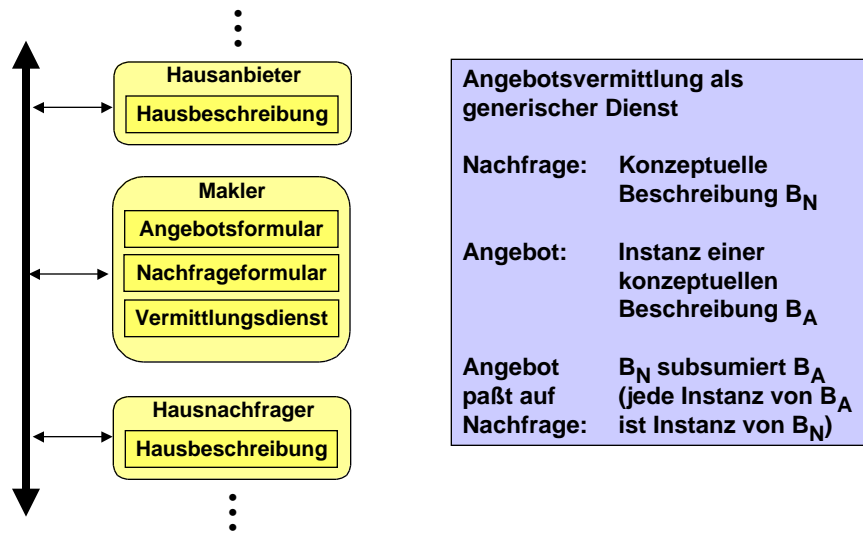
Ein Makler (Broker) ist Vermittler zwischen Nachfragern und Anbietern von Problemlösungen.

- Nachfrager (Requester) behandelt Makler wie Problemlöser
- Anbieter (Server) behandelt Makler wie Problemsteller



30

Maklersystem im Internet



31

Lernende Agenten

Beispiel: Maxims (Maes et al. 94)

- Agent lernt, wie ein Benutzer mit seiner Email umgeht
- Agent macht Vorschläge: Nachricht lesen, löschen, priorisieren, kategorisieren, archivieren, automatisch beantworten, ...

Agent kann mehrere Lernstrategien verfolgen:

- Lernen durch Imitieren des Benutzers
- Lernen durch Lehrbeispiele des Benutzers
- Lernen durch Benutzerkritik
- Lernen von anderen Agenten

Grundidee:

- Email-Aktionen des Benutzers anhand von Situations-Aktions-Beschreibungen speichern
- Situationen durch zahlreiche Merkmale beschreiben (Sender, Empfänger, Kurzüberschrift, Länge, Schlüsselworte, ...)
- Relevanz von Merkmalen lernen und gewichten
- Aktionen für neue Situationen durch Vergleich mit gespeicherten Situations-Aktions-Paaren ermitteln (→ Fallbasiertes Schließen)

32

Email-Aktionen lernen

Verstärkungslernen:

Relevanz von Merkmalswerten erhöhen, die zu derselben Aktion führen

Fallbasiertes Schließen:

Aktion desjenigen gespeicherten Falles F' vorschlagen, dessen Merkmale mit dem aktuellen Fall F am besten übereinstimmen:

Abstandsmaß $D(F, F') = \sum_i r_i \cdot d(m_i, m_i')$

Merkmal m_1 (Sender)	Relevanz r_1	Merkmal m_2 (Überschrift)	Relevanz r_2	Aktion
Cutter Corporation	7	E-Commerce Investment Strategies	3	delete unread
FBI-Dekanin	3	<u>FBR-Sitzung</u> 14.12.99	7	read
Cutter Corporation	7	Your Best Buy Ever	3	delete unread
FBI-Dekanin	3	<u>FBR-Sitzung</u> 2.2.00	7	read
FBI-Dekanin	4	FBI-Frauentreffen 9.1.00	6	delete unread
Ausschreibungsdienst	5	Mitteilung Nr. 42	5	archive
FBI-Planer	3	<u>FBR-Sitzung</u> 11.3.00	7	read
FBI-Planer	4	FBR-Sitzung 11.3.00 <u>fällt aus</u>	6	archive

33

Agententechnologie in Suchmaschinen

Arten von Suchmaschinen:

Einfache Suchmaschinen	Altavista, Yahoo, Webcrawler, Lycos, Excite, HotBot, Go, Witch (www.~.com oder www.~.de)
Meta-Suchmaschinen	MetaCrawler, SavvySearch, Metager
Personalisierte Suchmaschinen	Backweb, EntryPoint, Marimba, Netscape, Explorer

Agententechnologie spielt in Suchmaschinen eine wesentliche Rolle:

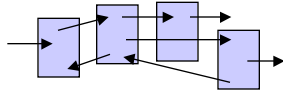
- Einfache Suchmaschinen verwenden unabhängige Agenten zur Suche nach neuen Internet-Seiten
- Meta-Suchmaschinen verwenden Agenten zur Beschaffung der Dokumente aus Suchergebnissen einfacher Suchmaschinen
- Personalisierte Suchmaschinen verwenden persönliche Agenten

34

Einfache Suchmaschinen

Arbeitsweise

1. Erfassen unbekannter Dokumente durch rekursives Verfolgen von bereits referenzierten Dokumenten



2. Indizieren der Informationen und Speicherung in einer Datenbank

Scheinkriterien

P2 Scheinkriterien

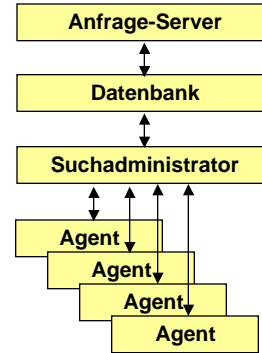
Es werden 8 Aufgabenblätter mit mindestens 24 Übungsaufgaben ausgegeben. Um einen Übungsschein zu erhalten, ist die persönliche Teilnahme an...
 URL: swt-www.informatik.uni-hamburg.de/Lehre/ss99/p2/Scheinkriterien.html
 Last modified on: 22-Mar-1999 - 4K bytes - in German

3. Bedienen von Suchanfragen (s. heutige Suchmaschinen)

z.B. automatische Übersetzung:

8 exercise sheets with at least 24 exercises are output. In order to receive a exercise banknote, 1. is the personal participation in at least 12 training group dates obligating, ...

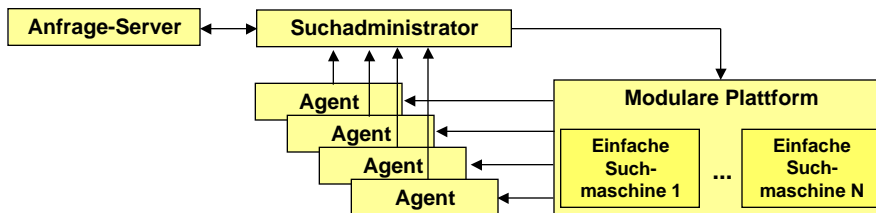
Architektur



35

Meta-Suchmaschinen

Lieferung qualitativ hochwertiger Ergebnisse durch parallele Abfrage der Datenbanken von einfachen Suchmaschinen und Aufbereitung der Einzelergebnisse.



Aufgaben des Suchadministrators

- Anfrage an einfache Suchmaschinen anpassen
- Suchergebnisse der einfachen Suchmaschinen analysieren und präsentieren

Aufgaben der Agenten

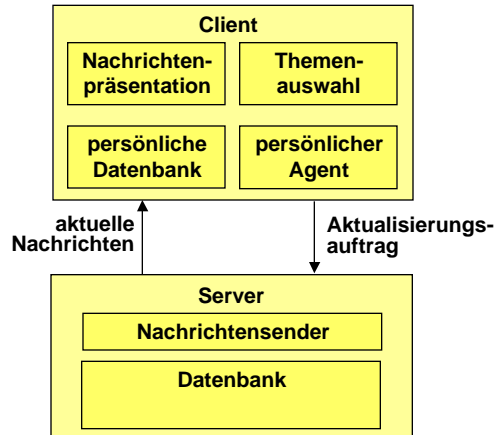
- Dokumente aus Suchergebnissen einfacher Suchmaschinen beschaffen

36

Personalisierter Nachrichtendienst

Beschaffung und Präsentation benutzerspezifischer Nachrichten im Abonnement (NewsWatcher)

- Auswahl personalisierter Nachrichtenkanäle
- Automatischer Informationsabgleich
- Push-Technologie (Server ist initiativ) oder Smart-Pull-Technologie (Client ist initiativ)



37

Mobile Agenten

Stationäre Agenten:

- Programm läuft auf einem Rechner
- Informationsaustausch durch Netzkommunikation

Mobile Agenten:

- Laufendes Programm kann Rechner wechseln
- Direkte Interaktion mit entfernten Agenten und Informationen

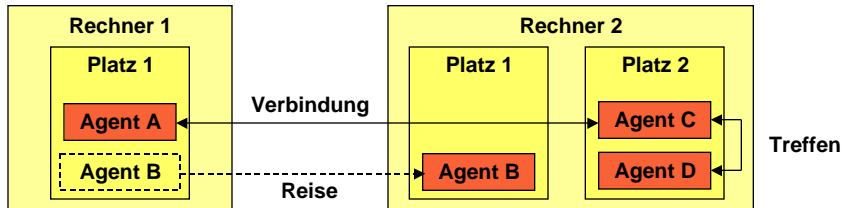
Aussuchen von Bildmaterial aus einem entfernten Archiv:

- Stationärer Agent muß (ggf. viele) Bilder aus Archiv zu lokalem Rechner übertragen, um aussuchen zu können
- Mobiler Agent überträgt sich zum Archiv, sucht dort aus und überträgt sich dann mit dem Bild zurück zum lokalen Rechner

Telescript Technologie von General Magic bietet kommerzielle Plattform für mobile Agenten

38

Telescript Konzepte



Platz	Einheit, die Dienstleistung anbietet
Agent	Funktionalität einer Anwendung, mobil oder stationär
Reise	Agent wechselt durch Go-Kommando zu einem anderen Platz, muß über Ticket mit Reiseparametern verfügen
Treffen	Interaktion zwischen Agenten an einem Platz durch Meet-Kommando, Agent muß Petition übergeben (z.B. Anfangszeit und Dauer des Treffens)
Verbindung	Kommunikation zwischen Agenten an verschiedenen Plätzen durch Connect-Kommando mit Angabe von Name und Platz eines Agenten
Autorität	Name-Kommando erfragt Autorität (Identität) eines Platzes oder Agenten
Permit	Berechtigung zum Nutzen von Ressourcen von Plätzen und Regionen, kann beschränkt erteilt werden. Agent stirbt bei Überschreitungen.

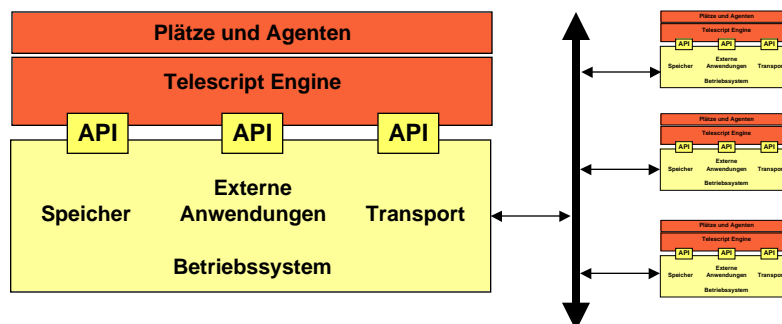
39

Telescript Engine

Die Telescript-Konzepte werden durch die Telescript-Engine realisiert.

APIs (Application Program Interfaces) nutzen Betriebssystem

- zum Retten von Plätzen und Agenten bei Störungen
- zum Anschluß externer C-Programme
- für die Verwendung der Kommunikation zum Transport von Agenten



40

Sicherheitskonzepte von Telescript

- **Laufzeitsicherheit (object runtime safety)**
Telescript-Engine sorgt für objekt-orientierte Kapselung, Laufzeit-Typenprüfung, Speichermanagement, Ausnahmenbehandlung
- **Prozeßsicherheit (process safety and security)**
Autoritäten- und Berechtigungskonzept und moderiertes Kommunikationsprotokoll schaffen spezielle Sicherheitsklassen (z.B. Copyright, Zugriffskontrolle)
- **Systemsicherheit (system safety and security)**
Kontrollierter Zugriff auf System-Ressourcen und Netzwerk-Ressourcen
- **Netzwerksicherheit (network security)**
Sicherheitsprüfungen beim Betreten von Regionen, verschlüsselte Nachrichtenübertragung

Telescript ist trotz sehr guter Konzepte bisher wenig verbreitet:

- proprietäres System
- Entwicklung plattformübergreifender Anwendungen problematisch

41

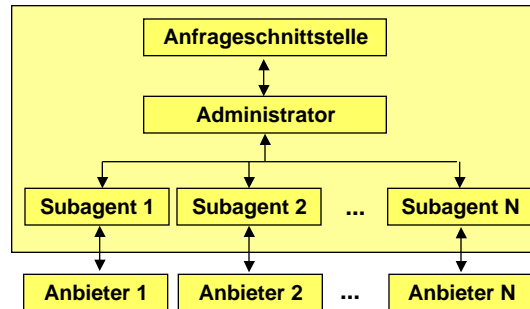
Electronic Commerce

Software-Agenten übernehmen die Rolle von Anbietern und Nachfragern

- **Einfache Kaufagenten** verschaffen Produktinformationen
 - Suche
 - Preisvergleich
- **Komplexe Kaufagenten** unterstützen gesamten Kaufvorgang
 - Suche
 - Preisvergleich
 - Zahlung
 - Lieferung
- **Agentenbasierte Marktplätze** umfassen Kauf- und Verkaufsagenten, Kreditagenten, Zahlungsagenten, Werbeagenten etc.
 - Suche
 - Werbung
 - Preisvergleich
 - Verhandlung
 - Kreditvergabe
 - Zahlung
 - Lieferung

42

Einfache Kaufagenten

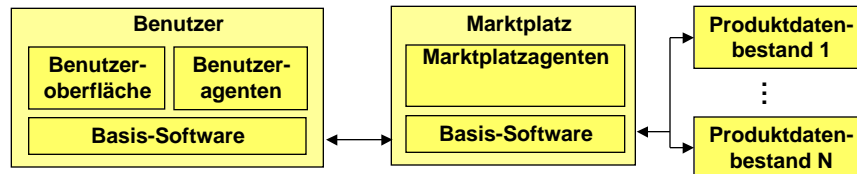


- **Anfrageschnittstelle**
Definition eines Produktwunsches und Präsentation der Ergebnisse
- **Subagent**
Führt Suchanfrage bei Anbieter durch und übermittelt Ergebnisse
- **Administrator**
Kennt Anbieter. Beauftragt Subagenten und aggregiert deren Ergebnisse.
- **Anbieter**
Stellt seine Produktdatenbestände zur Verfügung.

43

Agentenbasierte Marktplatz

Kooperation unabhängig agierender Agenten mit Verhandlungsfähigkeiten



Benutzeroberfläche

Zur Verständigung zwischen einem Benutzer und seinem Agenten. Eingabe notwendiger Parameter.

Benutzeragent

Unterstützung eines Benutzers beim Kauf oder Verkauf von Produkten. Kontakt mit dem Marktplatz über Basis-Software.

Marktplatzagent

Unterstützung von Anbietern beim Verkauf ihrer Produkte. Zugriff auf Produktdatenbestände. In der Regel ein Marktagent pro Anbieter.

44

Verständigung unabhängiger Agenten

Unabhängigkeit von Agenten hat 2 Seiten:

- unabhängige Entwicklung reduziert Komplexität
- unabhängiges Agieren erfordert Basis für Verständigung



Geringe Probleme

... bei ritualisierter (protokollarisch festgelegter) Kommunikation und Kooperation

Große Probleme

... bei Kommunikation über unabhängig entstandene Wissensbestände (z.B. Produktbeschreibungen, Spezialwissen)

=> Inhaltlicher Abgleich von sprachlichen Beschreibungen

=> Inhaltsbeschreibungssprache (z.B. Beschreibungslogik)

45

Zusammenfassung des 4. Teils

Wir haben Grundlagen der Agententechnologie kennengelernt:

Gründe für die Nützlichkeit von Agentenarchitekturen

(Komplexe Systeme, offene Systeme, partnerschaftliche Systeme)

BDI-Agenten, Blackboard-Agenten, Reaktive Agenten

(mit ihren Architekturen und besonderen Methoden)

Kommunikations- und Kooperationsprotokolle

(KQML, Kontraktnetz)

Agenten für besondere Anwendungsklassen

(Lernende Agenten, Suchmaschinen, Mobile Agenten, E-Commerce)

46

Zusammenfassung des 3. Teils

Wir haben Probleme der Nebenläufigkeit und ihre Lösungen kennengelernt:

Mehrbenutzerbetrieb von Datenbanken, verteilte Datenbanken

(Synchronisationsmethoden, Verklemmungsvermeidung, Transaktionsprotokolle)

Modellierung von nebenläufigen Prozessen

(Endliche Automaten, Petri-Netze)

Synchronisationsverfahren und -modelle

(Kritische Abschnitte, Schloßvariable, Semaphore, Pfadausdrücke)

Programmiertechnische Lösungen für Synchronisationsprobleme

(Threads, Nachrichtenaustausch, Fernaufruf von Prozeduren, Applets)

Echtzeitbetrieb

(Terminologie, Einplanung von Echtzeitprozessen)

47

Zusammenfassung des 2. Teils

Wir haben Methoden und Modelle für Organisation und Schnittstellen von Datenbeständen kennengelernt:

Datenbankentwurf

(Entity-Relationship-Modell, grundlegende Beziehungen)

Relationale Datenbanken

(Normalformen, Relationenalgebra, SQL)

Objektorientierte Datenbanksysteme (OODBS)

(OODBS-Elemente von Java, Vergleich mit semantischen Netzen)

Physische Datenorganisation

(Hashing, Interne Schlüssel, Blockorganisation, RAID)

Intelligente Datenbanken

(Deduktion mit Regeln, Datalog, Beschreibungslogiken)

48

Zusammenfassung des 1. Teiles

Wir haben wichtige Bausteine für die Modellierung von Realweltproblemen kennengelernt:

Dynamische Datenstrukturen mit zugehörigen Standardoperationen

(ADTs Listen, Mengen, Bäume, Graphen u.a.)

Dynamische Datentypen als Basis für wichtige problemorientierte Algorithmen

(Sortieren, Heuristische Suche, kürzester Pfad, MST, TSP u.a.)

Grundformen von Algorithmen

(Rekursion, Backtracking, Divide-and-Conquer, gierige Verfahren)

Komplexitätsklassen

(logarithmisch, linear, $n \log n$, quadratisch, polynomial, exponentiell u.a.)