Summer School Agententechnologie

Konzepte, Systeme und Anwendungen

Veranstaltungsnummer: 18.363 SS 04

Ort: F-535

Zeit: 10.-20.09.2004 10 – 18

Veranstalter:

Lars Braubach, Winfried Lamersdorf, Daniel Moldt, Alexander Pokahr und Heiko Rölke

NAME@informatik.uni-hamburg.de

Universität Hamburg, Fachbereich Informatik

Vogt-Kölln-Straße 30, D-22527 Hamburg

Übersicht

- Organisatorisches
- Themeneinführung
  - Motivation
  - Agentendefinitionen und -eigenschaften
  - Abgrenzung
- Agentenorientierte Softwareentwicklung
- Modellierung
  - Umsetzung von Agenteneigenschaften
  - Entwurf einer Agentenarchitektur
- Ausblick

Organisatorisches

Das Projektseminar erstreckt sich über sieben Wochentage.

Organisatorisches

Das Projektseminar erstreckt sich über sieben Wochentage.
Für die erfolgreiche Teilnahme sind
die Anwesenheit,
die engagierte Beteiligung,
die Erledigung der Aufgaben an den einzelnen Veranstaltungstagen,
die Erstellung einer kurzen Ausarbeitung zu einem im Laufe des Projektseminars festgelegten Themas und
die Abschlußpräsentation der Ergebnisse am Montag
Voraussetzung.

Organisatorisches

Das Projektseminar erstreckt sich über sieben Wochentage.
Für die erfolgreiche Teilnahme sind
die Anwesenheit,
die engagierte Beteiligung,
die Erledigung der Aufgaben an den einzelnen Veranstaltungstagen,
die Erstellung einer kurzen Ausarbeitung zu einem im Laufe des Projektseminars festgelegten Themas und
die Abschlußpräsentation der Ergebnisse am Montag
Voraussetzung.
Die jeweils vereinbarten Termine sind möglichst einzuhalten.

Organisatorisches

Das Projektseminar erstreckt sich über sieben Wochentage.
Für die erfolgreiche Teilnahme sind
die Anwesenheit,
die engagierte Beteiligung,
die Erledigung der Aufgaben an den einzelnen Veranstaltungstagen,
die Erstellung einer kurzen Ausarbeitung zu einem im Laufe des Projektseminars festgelegten Themas und
die Abschlußpräsentation der Ergebnisse am Montag
Voraussetzung.
Die jeweils vereinbarten Termine sind möglichst einzuhalten.
Fragen?

Motivation

Computer arbeiten Schritt für Schritt genau das ab, was man (der Programmierer und/oder Anwender) ihnen sagt.
Dieses imperative Denk- und Vorgehensmuster führt immer dann zu Problemen, wenn unvorhergesehene Eingaben auftreten.

Motivation

Computer arbeiten Schritt für Schritt genau das ab, was man (der Programmierer und/oder Anwender) ihnen sagt.
Dieses imperative Denk- und Vorgehensmuster führt immer dann zu Problemen, wenn unvorhergesehene Eingaben auftreten.
In vielen Anwendungsfällen ist eine direkte Überwachung nicht möglich bzw. nicht wünschenswert.

Motivation

Computer arbeiten Schritt für Schritt genau das ab, was man (der Programmierer und/oder Anwender) ihnen sagt.
Dieses imperative Denk- und Vorgehensmuster führt immer dann zu Problemen, wenn unvorhergesehene Eingaben auftreten.
In vielen Anwendungsfällen ist eine direkte Überwachung nicht möglich bzw. nicht wünschenswert.
Die Komplexität der Welt mit vielen (oft einfachen) interagierenden Einheiten (in dynamischen Netzwerkverbindungen) ist nur schwer in monolithischen/konventionellen Programmen einzufangen.

Motivation

Computer arbeiten Schritt für Schritt genau das ab, was man (der Programmierer und/oder Anwender) ihnen sagt.
Dieses imperative Denk- und Vorgehensmuster führt immer dann zu Problemen, wenn unvorhergesehene Eingaben auftreten.
In vielen Anwendungsfällen ist eine direkte Überwachung nicht möglich bzw. nicht wünschenswert.
Die Komplexität der Welt mit vielen (oft einfachen) interagierenden Einheiten (in dynamischen Netzwerkverbindungen) ist nur schwer in monolithischen/konventionellen Programmen einzufangen.
Lösungsvorschlag: Unabhängige Softwareeinheiten
d.h.: *Agenten* und eine adaptive Architektur

Anforderungen

- Schnell und gravierend ändernde Umwelt
- Unvorhersehbare Eingaben
- rasche Reaktionen, eigene Entscheidungen
- Versagen/Ausfallen als ständige Option
- Adaption an dauerhafte Veränderungen

Intuitive Agenteneigenschaften

Welche Eigenschaften sind nötig, um die oben genannten Anforderungen zu erfüllen?

Intuitive Agenteneigenschaften

Welche Eigenschaften sind nötig, um die oben genannten Anforderungen zu erfüllen?

Sind die Anforderungen adäquat mit vorhandenen Techniken zu meistern?

Was ist ein Agent?

Definition 1:

Klassischer Ansatz: *keine* Definition

Agent = KI-System

Nilsson: „I will consider a progression of AI systems or ’agents’, each...“

(Sicht bis Anfang/Mitte 90’er Jahre)
Agent ist Hilfsmittel der (V)KI: Agentenmetapher hilft bei der Erstellung „intelligenter“ Systeme.

Was ist ein Agent?

Definition 2:

> „An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future.“ (Franklin/Graesser 1995)

90’er: Agenten und Agentensysteme werden eigenständiger Forschungsgegenstand: Eigenschaften, Architekturen, Sprachen, ...

Was ist ein Agent?

Defi nition 2/II:

[An autonomous agent is] a hardware or (more usually) software-based computer system that enjoys the following properties:

- autonomy
- social ability
- reactivity
- pro-activeness

Schwache Defi nition; *starke* impliziert mentale Kategorien, die normalerweise Menschen zugeschrieben werden: Glauben, Wissen, Ziele...

(Jennings/Wooldridge 1995)

Was ist ein Agent?

Definition 3:

Agenten als Erweiterung der Objektorientierung

„An agent is an entity (object) whose state is viewed as consisting of mental components such as beliefs, capabilities, choices, and commitments.“
(Shoham 1997)

Was ist ein Agent?

Definition 4:

> „An object is just something with abilities and attributes and has no further defining characteristics. An agent is an object that is useful to (typical another) agent where this usefulness is defined in terms of satisfying that agent's goals“
> (Luck/Griffiths/d'Iverno 1996)

Mitte bis Ende der 90'er: Agenten als Spezialisierung von Objekten

Was ist ein Agent?

Letzten Jahre: Agenten und Agentensysteme geraten in den Blick der „Mainstream“-Informatik: Agentenorientierte Softwareentwicklung (AOSE)

Agenten als universelles Softwarewerkzeug,
Agenten sind „in Mode“

- Eine wichtige Literaturquelle:

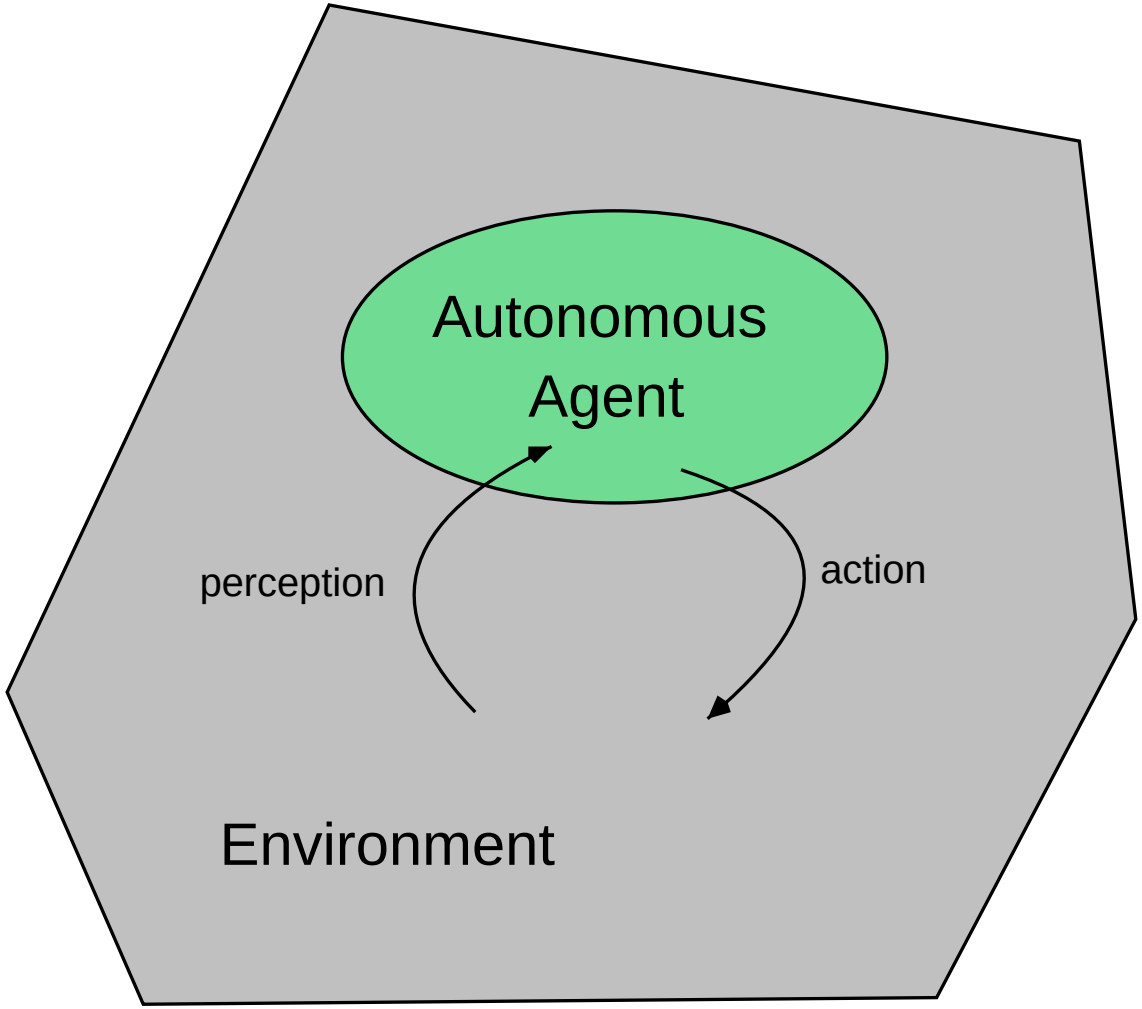
Artikel aus
Artificial Intelligence 117 (2000) 277-296

Autor:
Nicholas R. Jennings

Titel:
On agent-based software engineering

Was ist ein Agent?

Graphische Veranschaulichung:



Sammlung von Agenteneigenschaften

- Autonomie
- Mobilität
- Zielorientierung
- Kommunikation
- Koordinierung: Kooperation/Konkurrenz
- Intelligenz
- Lernfähigkeit
- Adaption
- Reaktivität/Proaktivität
- ...

Abgrenzung Agenten

Objekte:

- Zustand gekapselt
- Methoden
- Nachrichtenkommunikation

Abgrenzung Agenten

Objekte:

- Zustand gekapselt
- Methoden
- Nachrichtenkommunikation

nicht abgedeckt:

- Zugriffskontrolle (bel. fein)
- Systeme ohne „gemeinsames Ziel“
- flexible Reaktionen
- dauerhafte Aktivität (eigener Kontrollthread)

Abgrenzung Agenten 2

Expertensystem:

- keine Kapselung
- eingeschränkte Umwelt / Interaktion
- Einzelsysteme
- keine Autonomie / eigene Aktivität

Definition Agent

kleinster gemeinsamer Nenner:

„An agent is (a computer system) situated in an environment and capable of flexible autonomous action within that environment.“

(nach Jennings/Wooldridge)

Definition Agent

kleinster gemeinsamer Nenner:

„An agent is (a computer system) situated in an environment and capable of flexible autonomous action within that environment.“

(nach Jennings/Wooldridge)

- klare Umgebungsabgrenzung (Kapselung)

Definition Agent

kleinster gemeinsamer Nenner:

„An agent is (a computer system) situated in an environment and capable of flexible autonomous action within that environment.“

(nach Jennings/Wooldridge)

- klare Umgebungsabgrenzung (Kapselung)
- Flexibilität (Adaption, Intelligenz)

Definition Agent

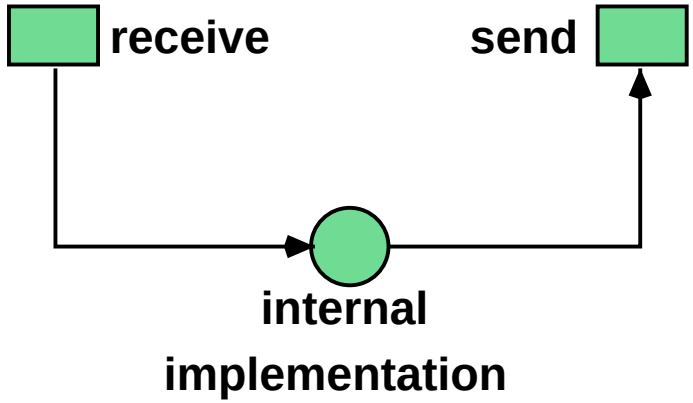
kleinster gemeinsamer Nenner:

„An agent is (a computer system) situated in an environment and capable of flexible autonomous action within that environment.“

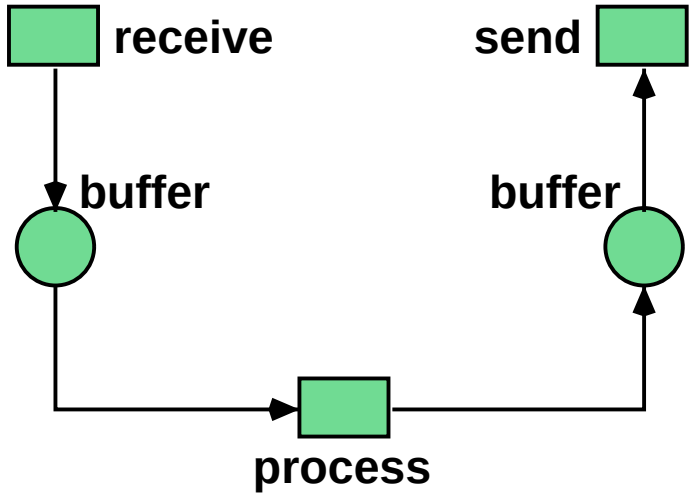
(nach Jennings/Wooldridge)

- klare Umgebungsabgrenzung (Kapselung)
- Flexibilität (Adaption, Intelligenz)
- Autonomie

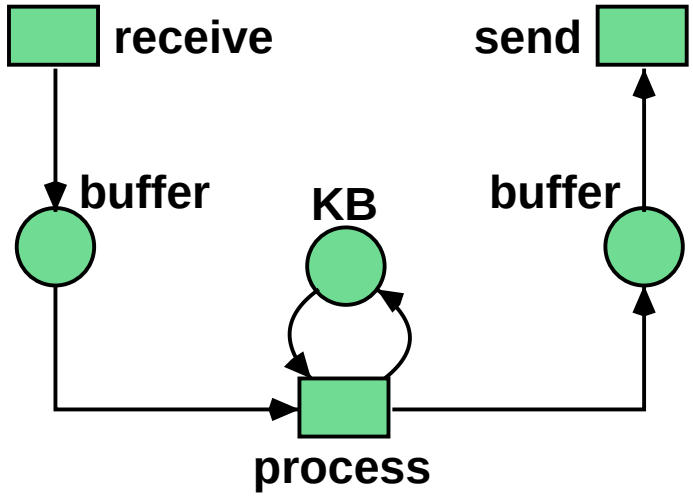
Modellierung von grundlegenden Agenteneigenschaften



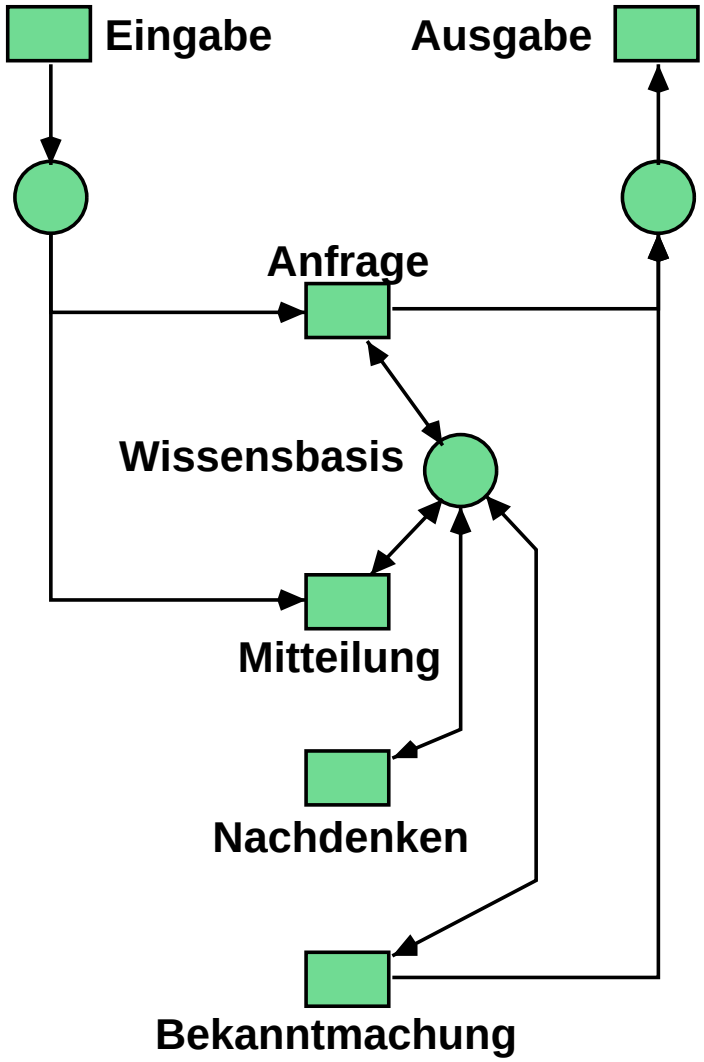
Modellierung von grundlegenden Agenteneigenschaften



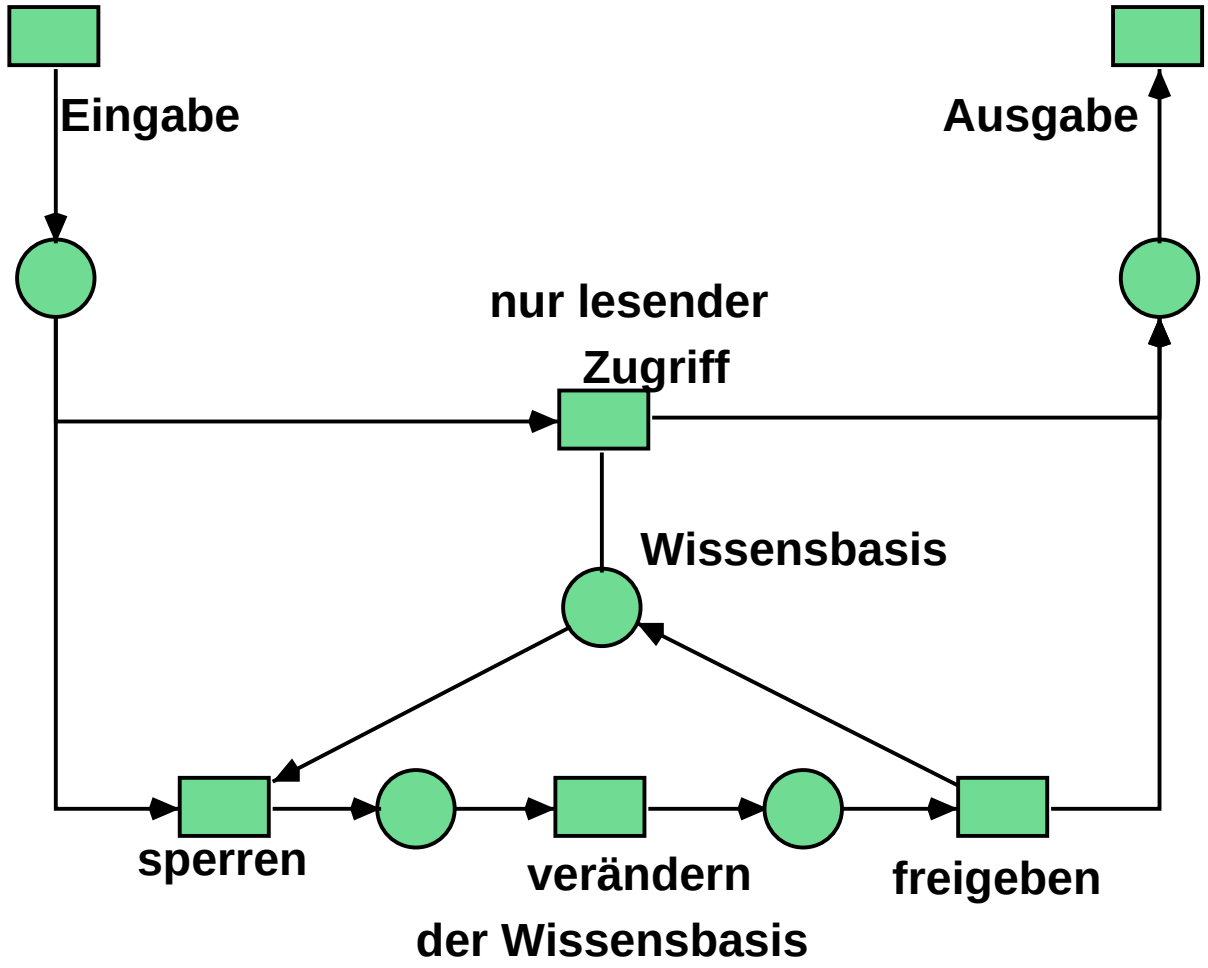
Modellierung von grundlegenden Agenteneigenschaften



Verfeinertes Agentenmodell



Verfeinertes Agentenmodell



Systeme von Agenten

(1) Die Grundidee hinter Agenten – unabhängige (Software-) Einheit mit eigenständigem Verhalten – ist zwar unabhängig von der Stückzahl, praktisch alle Anwendungen bauen aber auf Systemen auf, die aus mehreren bis zahlreichen Agenten bestehen. (Agentensysteme, Multiagentensysteme (MAS))

Systeme von Agenten

(1) Die Grundidee hinter Agenten – unabhängige (Software-) Einheit mit eigenständigem Verhalten – ist zwar unabhängig von der Stückzahl, praktisch alle Anwendungen bauen aber auf Systemen auf, die aus mehreren bis zahlreichen Agenten bestehen. (Agentensysteme, Multiagentensysteme (MAS))

(2) Ein wichtiger Entwicklungsgrundsatz ist die Aufteilung der Funktionalität in zahlreiche einfache Einheiten, die Komplexität des Gesamtsystems erwächst dann aus der Interaktion dieser Einheiten. (MAS, Agentengesellschaften)

Systeme von Agenten

Multiagentensystem (Makrosystem) erwächst aus der Zusammenfassung und Interaktion zahlreicher Agenten (Mikrosysteme) in einer Umgebung.

Sichtweisen

Systeme von Agenten

Multiagentensystem (Makrosystem) erwächst aus der Zusammenfassung und Interaktion zahlreicher Agenten (Mikrosysteme) in einer Umgebung.

Sichtweisen

Ferber (1999) gibt eine rein technische Definition aus Umgebung, Objekten, Agenten und deren Beziehungen.

Systeme von Agenten

Multiagentensystem (Makrosystem) erwächst aus der Zusammenfassung und Interaktion zahlreicher Agenten (Mikrosysteme) in einer Umgebung.

Sichtweisen

> System von Agenten besitzt einen „Mehrwert“ gegenüber der Summe der Einzelagenten. (Nwana, 1999)

Systeme von Agenten

Multiagentensystem (Makrosystem) erwächst aus der Zusammenfassung und Interaktion zahlreicher Agenten (Mikrosysteme) in einer Umgebung.

Sichtweisen

> Zusammenfassung der Agenten geschieht mit spezieller Absicht und einem Ziel.
> (Schumacher, 2001)

Eigenschaften MAS

- Jeder Agent besitzt nur unvollständiges Wissen.
- Keine globale Kontrollinstanz
- Daten und Funktionalität liegen verteilt vor.
- Offenheit und "verteilter Entwurf"

- Unabhängige und nebenläufige Ausführung der Agenten.
- Kopplung durch Unterbau zur Kommunikation
- (zahlreiche techn. Detaileigenschaften)

Vorteile MAS

ggü. Einzelagenten/Einzelsystemen:

- „Natürliche“ Umsetzung verteilter Systeme (Daten, Funktionalität)
- Dynamische Struktur, offene Systeme
- Robustheit
- Erweiterbarkeit
- Skalierbarkeit
- Wiederverwendbarkeit
- „Intelligenz nur durch Interaktion“ (nach Brooks)

Probleme MAS

- Aufwand technischer Unterbau
- Interaktionen evtl. schwer zu beherrschen/überblicken
- Design der Einzelagenten: Reaktivität – Deliberation
- Koordination sollte/muss sichergestellt sein
- u.v.a.m.

Probleme MAS

Anmerkung:

Das Hauptproblem liegt in der Systemarchitektur, die auch die Art der Kommunikation der einzelnen Agenten festlegt. Hier gibt es über Standards (FIPA) zahlreiche Bemühungen. Diese reichen jedoch nicht aus! Die Problematik des *Semantic Web* beim Internet greift in gleicher Weise auch bei MAS!

Eine Lösung sind Ontologien.
(Siehe Vortrag nächste Sitzung)

Agentenorientierte Softwareentwicklung

- Wie kommt man von Agenten zu Agentensystemen?

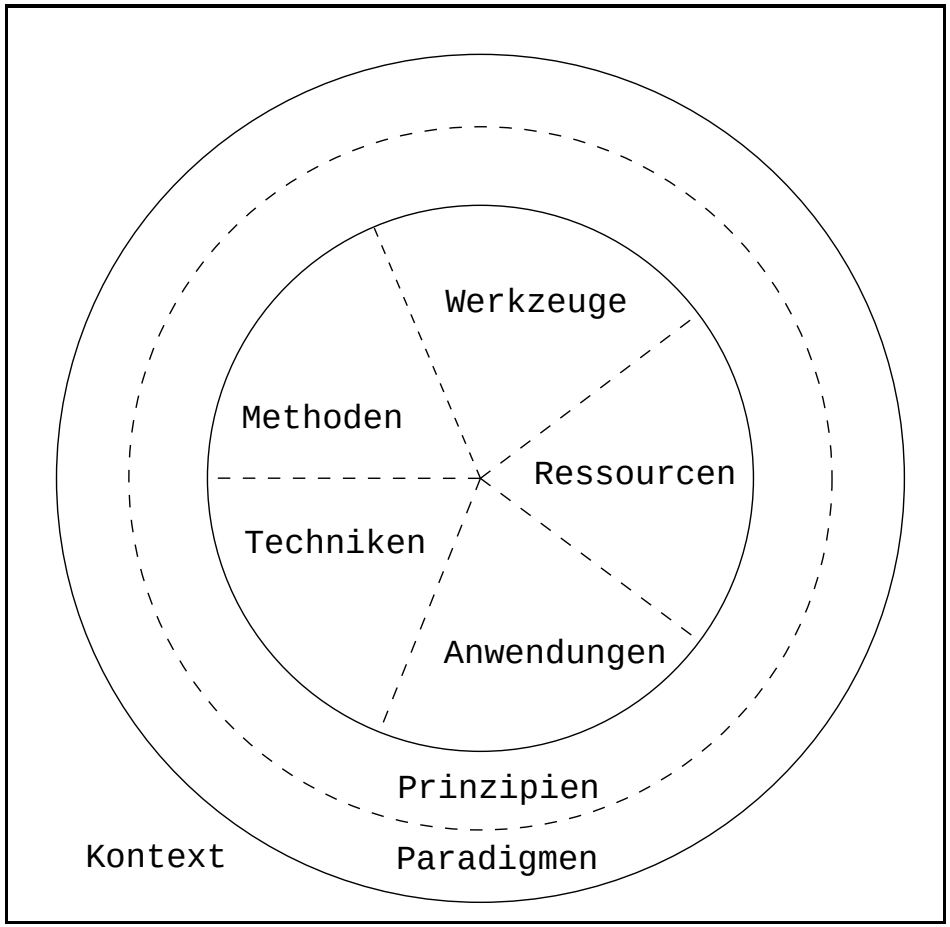
Agentenorientierte Softwareentwicklung

- Wie kommt man von Agenten zu Agentensystemen?
- Wie entwickelt man Agentensysteme?

Agentenorientierte Softwareentwicklung

- Wie kommt man von Agenten zu Agentensystemen?
- Wie entwickelt man Agentensysteme?
- Was sind wichtige Facetten eines Ansatzes?

Ansatz



Ein Ansatz mit dem Rahmen und seinen Facetten

Ein Ansatz ist die systematische Betrachtung bei Aufgabenstellungen der Systemspezifi kation mit Beschränkung auf die softwaretechnischen Aspekte.

Relevanter Ansatz des Projektes

Im Kontext dieses Projektseminars:

Relevanter Ansatz des Projektes

Im Kontext dieses Projektseminars:
Paradigma:

- Agentenorientierung
- Ausbau der Künstlichen Intelligenz und der Objektorientierung

Relevanter Ansatz des Projektes

Im Kontext dieses Projektseminars:
Paradigma:

- Agentenorientierung
- Ausbau der Künstlichen Intelligenz und der Objektorientierung

Prinzip:

- Problem adäquate Zerlegung

Relevanter Ansatz des Projektes

Anwendung:
Prey / Predator (Beute / Jäger) Beispiel

- Einzelne Akteure (Beute + Jäger)
- Geschlossene Welt
- Jäger versuchen Beute zu stellen
- Beute versucht zu entkommen
- eingeschränkter Sichtbereich der Akteure, aber Kommunikation möglich
- Technische Basis: Java
- Individuelle Strategien

Zielsetzung im Projektseminar:
Erproben mit verschiedenen Werkzeugen

Living creatures | Highscore

Rank	Type	Name	Age	Poi
1		Clever	56	
2		Lenny	56	
3		Heinz	56	
4		Duffy	56	
5		Charlton	56	
6		Harald	56	
7		Louis	56	
8		Dumbo	56	

Environment Control

Round number: 5

Round time [millis]: 1

Autosave highscore [millis, -1 for off] 5

Food rate [every n ticks] 5

Save highscore

Relevanter Ansatz des Projektes

Ressourcen sind konstant.
Methode und Techniken varieren mit den Werkzeugen:

- Sesam
- Jess
- Jadex
- Mulan

Technische Umgebung

Technische Werkzeuge, die im Projekt verwendet werden:

- Entwicklungsumgebungen:
  PCs mit jeweiligen Editoren, Compilern etc.
- Java
- Agentenplattformen

Aspekte der AOSE

Drei Bereiche sind besonders bei der Softwareentwicklung zu beachten:

- Die Umgebung eines Multiagentensystems: Welchen Zweck soll das MAS erfüllen?

Aspekte der AOSE

Drei Bereiche sind besonders bei der Softwareentwicklung zu beachten:

- Die Umgebung eines Multiagentensystems: Welchen Zweck soll das MAS erfüllen?
- Die Interaktion zwischen den Agenten Wie ist die Systemarchitektur?

Aspekte der AOSE

Drei Bereiche sind besonders bei der Softwareentwicklung zu beachten:

- Die Umgebung eines Multiagentensystems:
  Welchen Zweck soll das MAS erfüllen?
- Die Interaktion zwischen den Agenten
  Wie ist die Systemarchitektur?
- Der Aufbau der Agenten
  Was können die Agenten?

Aspekte der AOSE

Rahmen eines jeden Projektes sind:

- der Zweck des Ergebnisses
- die verfügbaren Ressourcen
  (Zeit, Geld, Material, Werkzeuge, Personen etc.)
- der verwendete Ansatz

Aspekte der AOSE

Die Facetten des AOSE werden in den nächsten Tag im Zusammenhang mit den jeweiligen Werkzeugen unterschiedlich gewichtet und ausgefüllt. Alle Teilnehmer werden daher gebeten sich der jeweiligen “Umgebung” bewusst zu werden und gezielt einen Ansatz zu verfolgen, insbesondere:

- Paradigma
- Techniken
- Werkzeuge
- Methoden

Aspekte der AOSE

Klippen bei der Arbeit im Projektseminar:

- Wissen über die Techniken
  - Java
  - Protege
  - (A)UML
  - Referenznetze
- Wissen über die Entwicklungsumgebung
- Wissen über den Ansatz allgemein

Aspekte der AOSE

Technischer Background wird im nächsten Vortrag angegangen. Gegenstand: Jade, Ontologien und deren Nutzung

Organisatorisches

Heute:

- Kurze Pause
- Vortrag: Jade und Ontologien
  Aufgabe
- Mittagspause
- Aufgaben
- Pause
- Aufgaben

Organisatorisches

Montag:

- Beginn: 10h15(!) mit dem ersten Gastvortrag

  SeSAm

Organisatorisches

?

Exkurs: Sprechakttheorie

Sprache besteht aus Aktionen (Anforderung, Zugeständnis, Einwilligung, ...). Eine sprachliche Äußerung beinhaltet drei Arten von Aktionen bzw. löst diese aus:

- lokutionärer (lokutiver) Akt: die (Laut-) Äußerung selbst

Exkurs: Sprechakttheorie

Sprache besteht aus Aktionen (Anforderung, Zugeständnis, Einwilligung, ...). Eine sprachliche Äußerung beinhaltet drei Arten von Aktionen bzw. löst diese aus:

- lokutionärer (lokutiver) Akt: die (Laut-) Äußerung selbst
- illokutionärer (illokutiver) Akt: intendierte Bedeutung/Handlung, Zustandsänderung (vom Sprecher ausgehend)

Exkurs: Sprechakttheorie

Sprache besteht aus Aktionen (Anforderung, Zugeständnis, Einwilligung, ...). Eine sprachliche Äußerung beinhaltet drei Arten von Aktionen bzw. löst diese aus:

- lokutionärer (lokutiver) Akt: die (Laut-) Äußerung selbst
- illokutionärer (illokutiver) Akt: intendierte Bedeutung/Handlung, Zustandsänderung (vom Sprecher ausgehend)
- perlokutionärer (perlokutiver) Akt: Auswirkung auf den Angesprochenen (ausgelöste Handlung)

Exkurs: Sprechakttheorie

Ausruf: „Sie sind verhaftet!“ – lokutionärer Akt

Exkurs: Sprechakttheorie

Ausruf: „Sie sind verhaftet!“ – lokutionärer Akt

Festnahme – illokutionärer Akt

Exkurs: Sprechakttheorie

Ausruf: „Sie sind verhaftet!“ – lokutionärer Akt

Festnahme – illokutionärer Akt

Flucht – perlokutionärer Akt

Exkurs: Sprechakttheorie

Klassische Einteilung:

- assertive: behaupten, informieren
- directive: anweisen
- promissive: zusichern
- expressive: informieren über (mentalen) Zustand
- declarative: durchführen durch Äußerung:
  „I declare this basar open.“

Probleme mit natürlicher Sprache z.B. bei konditionalen Ausdrücken.

Sprechakttheorie – Anwendung

Ein *Performativ* ist ein Sprechakt mit einer eindeutigen Illokution (eig.: Sprechakt mit gleichzeitig vollzogener Handlung):

Sprechakttheorie – Anwendung

Ein *Performativ* ist ein Sprechakt mit einer eindeutigen Illokution (eig.: Sprechakt mit gleichzeitig vollzogener Handlung):

Nur solche Sprechakte werden als Typen für Nachrichten zwischen Agenten verwendet, weshalb die Nachrichten auch als Performative bezeichnet werden.

Sprechakttheorie – Anwendung

Ein *Performativ* ist ein Sprechakt mit einer eindeutigen Illokution (eig.: Sprechakt mit gleichzeitig vollzogener Handlung):

Nur solche Sprechakte werden als Typen für Nachrichten zwischen Agenten verwendet, weshalb die Nachrichten auch als Performative bezeichnet werden.

Performative erlauben die Festlegung einer Semantik für den Nachrichtenaustausch.

FIPA-ACL

- baut stark auf KQML auf
- eingebettet in umfangreiche Standardisierungsvorschläge zum Agentenmanagement und und und
- keine *Performative* sondern *communicative acts*
- Unterschiede bei der Semantikbeschreibung und den vorausgesetzten mentalen Zuständen
- keine Verwaltungsperformative, sondern explizite Regelung des Agent Management

Exkurs: FIPA

- Foundation for Intelligent Physical Agents
- gegr. 1996, Schweiz als Forum zur Erarbeitung von Standards im Bereich von MAS
- Internationale Firmen (bes. Telekommunikation), Universitäten

- Arbeitsbereich: Standards für Agenteninteraktion: Kommunikation/Koordination und Verwaltung
- Umfangreicher Korpus an Veröffentlichungen (Standardisierungsvorschläge), bis jetzt kein endgültiger Vorschlag (nach internem Lebenszyklus)

FIPA – Mission Statement

„The promotion of technologies and interoperability specifications that facilitate the end-to-end interworking of intelligent agent systems in modern commercial and industrial settings.“

FIPA – Mission Statement

„The promotion of technologies and interoperability specifications that facilitate the end-to-end interworking of intelligent agent systems in modern commercial and industrial settings.“

Foundation for **I**nteroperable **P**eer-to-peer **A**gents oder so...

FIPA – Übersicht

- Kommunikation: ACL, content language, communicative acts library, interaction protocols

FIPA – Übersicht

- Kommunikation: ACL, content language, communicative acts library, interaction protocols
- Kommunikationsgrundlagen: Namensdienst, Transportdienst(e), Verzeichnisdienst, ACL Message Structure (teilw. abstr. Arch.)

FIPA – Übersicht

- Kommunikation: ACL, content language, communicative acts library, interaction protocols
- Kommunikationsgrundlagen: Namensdienst, Transportdienst(e), Verzeichnisdienst, ACL Message Structure (teilw. abstr. Arch.)
- Angrenzend (Verwaltung): agent management, nomadic application, mobility, configuration management, ontology service... ← später mehr

FIPA – Übersicht

- Kommunikation: ACL, content language, communicative acts library, interaction protocols
- Kommunikationsgrundlagen: Namensdienst, Transportdienst(e), Verzeichnisdienst, ACL Message Structure (teilw. abstr. Arch.)
- Angrenzend (Verwaltung): agent management, nomadic application, mobility, configuration management, ontology service... ← später mehr
- Anwendungen: travel assistant, network management...

FIPA – Übersicht

- Kommunikation: ACL, content language, communicative acts library, interaction protocols
- Kommunikationsgrundlagen: Namensdienst, Transportdienst(e), Verzeichnisdienst, ACL Message Structure (teilw. abstr. Arch.)
- Angrenzend (Verwaltung): agent management, nomadic application, mobility, configuration management, ontology service... ← später mehr
- Anwendungen: travel assistant, network management...

`www.fipa.org`

FIPA: Communicative Acts

1. Request, Request When,... – Agree
2. Query If, Query Ref – Inform (If + Ref)
3. Propose – Accept, Reject
4. Cancel, Failure, Not Understood
5. ...

FIPA: Inhaltssprachen

Content Language Library stellt Bedingungen an CL auf

1. KIF
2. SL
3. RDF

Nachrichten -> Konversationen

Die Benutzung von KQML legt schon einfache Basiskonversationen fest:
`ask-if - tell` usw.

Nachrichten -> Konversationen

Die Benutzung von KQML legt schon einfache Basiskonversationen fest:
`ask-if - tell` usw.

Darauf bauen komplexere Kommunikationsabläufe auf.

Nachrichten -> Konversationen

Die Benutzung von KQML legt schon einfache Basiskonversationen fest:
`ask-if - tell` usw.

Darauf bauen komplexere Kommunikationsabläufe auf.

Dies erlaubt die Festlegung von festen Konversationen z.B. zum Zwecke der Koordinierung.
(Bsp.: Contract-Net)

Konversationen

- Nachrichtenversand (innerhalb einer Konversation) impliziert Erwartungen an Antwortnachricht(en)
- Übergreifende Struktur, erfordert Darstellungsweise getrennt von Kommunikationssprache
- Vorgegebene Konversationsstrukturen helfen dem Agenten bei der Zuordnung ankommender Nachrichten

Konversationen

- Spezifikation: Beschreibung in sowohl menschen- als auch maschinenlesbarer Form

Konversationen

- Spezifikation: Beschreibung in sowohl menschen- als auch maschinenlesbarer Form
- Benutzung: Festlegen auf Benutzung eines Konversationsprotokolls, Lernen neuer Konversationen...

Konversationen

- Spezifikation: Beschreibung in sowohl menschen- als auch maschinenlesbarer Form
- Benutzung: Festlegen auf Benutzung eines Konversationsprotokolls, Lernen neuer Konversationen...
- Konversationsprotokolle und Rollen: Beschreibung von Agenten durch Angabe ihrer Konversationsprotokolle

Konversationen

- Spezifikation: Beschreibung in sowohl menschen- als auch maschinenlesbarer Form
- Benutzung: Festlegen auf Benutzung eines Konversationsprotokolls, Lernen neuer Konversationen...
- Konversationsprotokolle und Rollen: Beschreibung von Agenten durch Angabe ihrer Konversationsprotokolle
- Mehr: Koordination bzw. AOSE

FIPA: Konversationsprotokolle

Beispiele:

1. Request, Query, Propose...
2. (iterated) Contract Net
3. Dutch/English Auction
4. Brokering, Recruiting

definiert in AUML und natürlicher Sprache...

Begriffsbestimmung

- Koordination: ggs. Abstimmen

(Duden)

Begriffsbestimmung

- Koordination: ggs. Abstimmen
- Kooperation: Zusammenarbeit

(Duden)

Begriffsbestimmung

- Koordination: ggs. Abstimmen
- Kooperation: Zusammenarbeit
- Konkurrenz: „zusammenlaufen, -treffen, aufeinander stoßen“, Wettbewerb/gleichzeitiges Bewerben

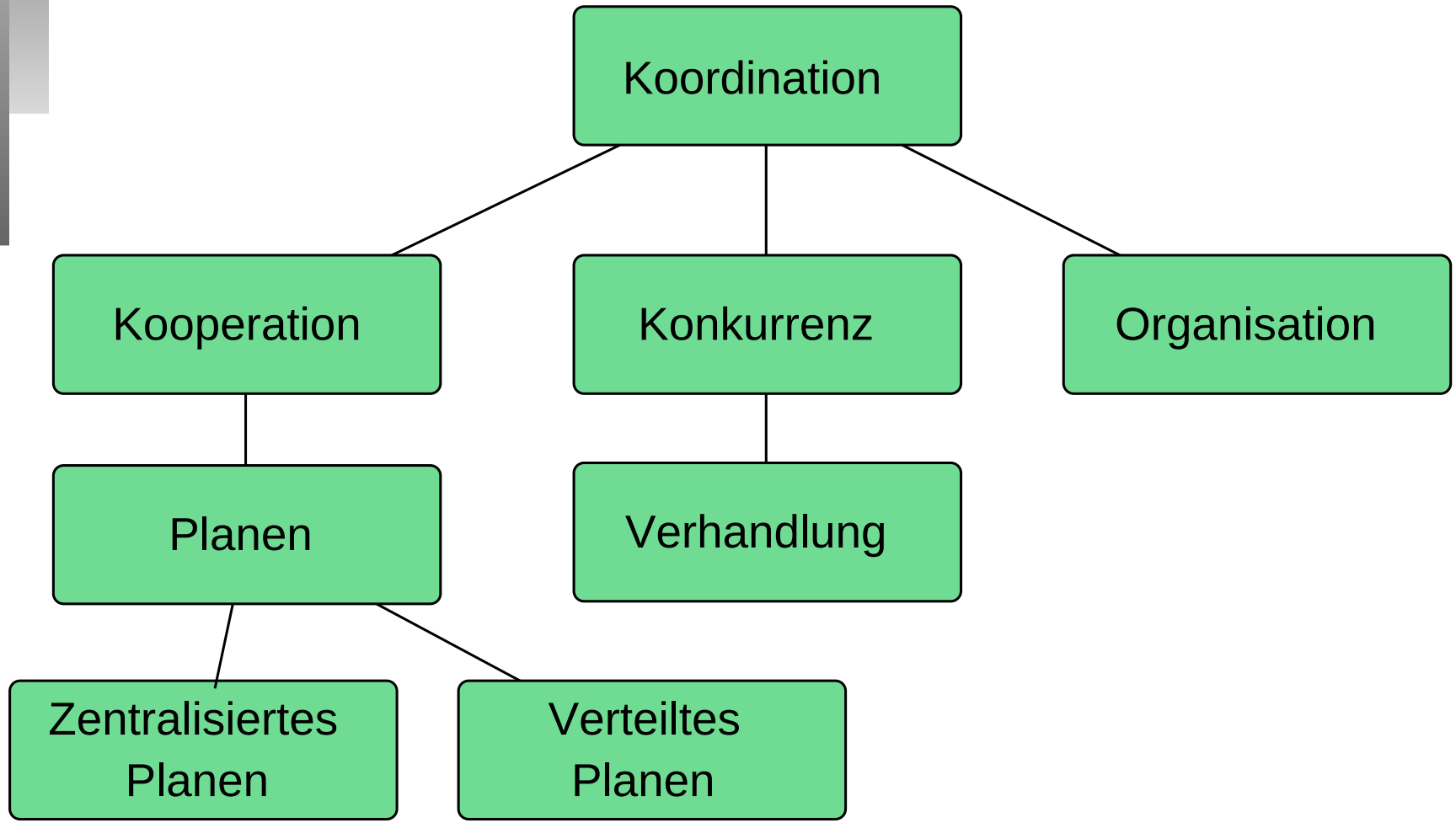
(Duden)

Begriffsbestimmung

- Koordination: ggs. Abstimmen
- Kooperation: Zusammenarbeit
- Konkurrenz: „zusammenlaufen, -treffen, aufeinander stoßen“, Wettbewerb/gleichzeitiges Bewerben
- Organisation: planmäßiger Aufbau, Gliederung

(Duden)

Begriffsbestimmung



Voraussetzungen

- Kooperation/verteiltes Planen: gemeinsames (System-) Ziel, kooperative Agenten
- Konkurrenz/Verhandlung: selbstinteressierte (egoistische) Agenten, widersprüchliche Ziele

Voraussetzungen

- Kooperation/verteiltes Planen: gemeinsames (System-) Ziel, kooperative Agenten
- Konkurrenz/Verhandlung: selbstinteressierte (egoistische) Agenten, widersprüchliche Ziele
- Weltmodell, Modell anderer Agenten
- Erwartungen an zukünftige Handlungen
  $\rightarrow$ Sozialität

Voraussetzungen

- Kooperation/verteiltes Planen: gemeinsames (System-) Ziel, kooperative Agenten
- Konkurrenz/Verhandlung: selbstinteressierte (egoistische) Agenten, widersprüchliche Ziele

Voraussetzungen

- Kooperation/verteiltes Planen: gemeinsames (System-) Ziel, kooperative Agenten
- Konkurrenz/Verhandlung: selbstinteressierte (egoistische) Agenten, widersprüchliche Ziele
- Weltmodell, Modell anderer Agenten
- Erwartungen an zukünftige Handlungen
  $\rightarrow$ Sozialität

Verteiltes Problemlösen und Planen

- gemeinsame Anstrengung mehrerer Problemlöser
- Bündelung der Kapazitäten (Wissen, Informationen, Fähigkeiten...)
- jeder Teilplan muss andere Pläne angemessen berücksichtigen

Verteiltes Problemlösen und Planen

- gemeinsame Anstrengung mehrerer Problemlöser
- Bündelung der Kapazitäten (Wissen, Informationen, Fähigkeiten...)
- jeder Teilplan muss andere Pläne angemessen berücksichtigen

„Thus, while working together leads to distributed problem solving, there is also the distributed problem of how to work together that must be solved.“ (Durfee)

Warum verteiltes Problemlösen?

zentrales Problemlösen unmöglich oder zu aufwendig:

- Aufteilung spart Ressourcen (z.B. Rechenzeit)
- benötigte Funktionalität liegt verteilt vor
- benötigtes Wissen/Informationen liegen verteilt vor
- höhere Ergebnisqualität:
  - Zeit
  - Vollständigkeit
  - Präzision
  - Sicherheit

Voraussetzungen

- Gruppenzusammenhalt (coherence):
  Die Gruppenmitglieder müssen gewillt und

Voraussetzungen

- Gruppenzusammenhalt (coherence):
  Die Gruppenmitglieder müssen gewillt und
- Gruppenkompetenz (competence):
  in der Lage sein, das Problem zu lösen.

Voraussetzungen

- Gruppenzusammenhalt (coherence):
  Die Gruppenmitglieder müssen gewillt und
- Gruppenkompetenz (competence):
  in der Lage sein, das Problem zu lösen.
- Kooperationsbereitschaft muss vorhanden sein, keine rein selbstinteressierten Agenten
- Forschungsinteresse liegt auf Kompetenz:
  Aufteilung in Teilprobleme, Verteilung der Teilprobleme, Zusammenfügen usw.: typische Planungsprobleme
  $\rightarrow$ verteiltes Planen

Aufgabenteilung

- **Aufgabe(n) dekomponieren:** Einzelaufgaben identifizieren, große Aufgaben dekomponieren

Aufgabenteilung

- **Aufgabe(n) dekomponieren:** Einzelaufgaben identifizieren, große Aufgaben dekomponieren
- **Aufgaben zuteilen:** Einzelaufgaben an (jeweils geeigneten) Agenten zuweisen.

Aufgabenteilung

- **Aufgabe(n) dekomponieren:** Einzelaufgaben identifizieren, große Aufgaben dekomponieren
- **Aufgaben zuteilen:** Einzelaufgaben an (jeweils geeigneten) Agenten zuweisen.
- **Aufgaben ausführen:** rekursive Aufteilung bzw. Erledigung atomarer Aufgaben

Aufgabenteilung

- **Aufgabe(n) dekomponieren:** Einzelaufgaben identifizieren, große Aufgaben dekomponieren
- **Aufgaben zuteilen:** Einzelaufgaben an (jeweils geeigneten) Agenten zuweisen.
- **Aufgaben ausführen:** rekursive Aufteilung bzw. Erledigung atomarer Aufgaben
- **Ergebnisse zusammenführen:** normalerweise durch den Auftraggeber

Aufgabenteilung: Beispiel

Türme von Hanoi

- **Aufgabe(n) dekomponieren:** jeweils grösste nicht richtig liegende Scheibe soll bewegt werden: Weg dahin und Weg von dort zum Ziel

Aufgabenteilung: Beispiel

Türme von Hanoi

- **Aufgabe(n) dekomponieren:** jeweils grösste nicht richtig liegende Scheibe soll bewegt werden: Weg dahin und Weg von dort zum Ziel
- **Aufgaben zuteilen:** idealisiert an unendlich viele gleichartige Agenten

Aufgabenteilung: Beispiel

Türme von Hanoi

- **Aufgabe(n) dekomponieren:** jeweils grösste nicht richtig liegende Scheibe soll bewegt werden: Weg dahin und Weg von dort zum Ziel
- **Aufgaben zuteilen:** idealisiert an unendlich viele gleichartige Agenten
- **Aufgaben ausführen:** weitere rekursive Zerlegung wie oben; atomare Bewegung

Aufgabenteilung: Beispiel

Türme von Hanoi

- **Aufgabe(n) dekomponieren:** jeweils grösste nicht richtig liegende Scheibe soll bewegt werden: Weg dahin und Weg von dort zum Ziel
- **Aufgaben zuteilen:** idealisiert an unendlich viele gleichartige Agenten
- **Aufgaben ausführen:** weitere rekursive Zerlegung wie oben; atomare Bewegung
- **Ergebnisse zusammenführen:** entsprechend rekursiver Aufstieg, Zusammenfügen der Teilergebnisse

Aufgabenteilung: Vor- und Nachteile

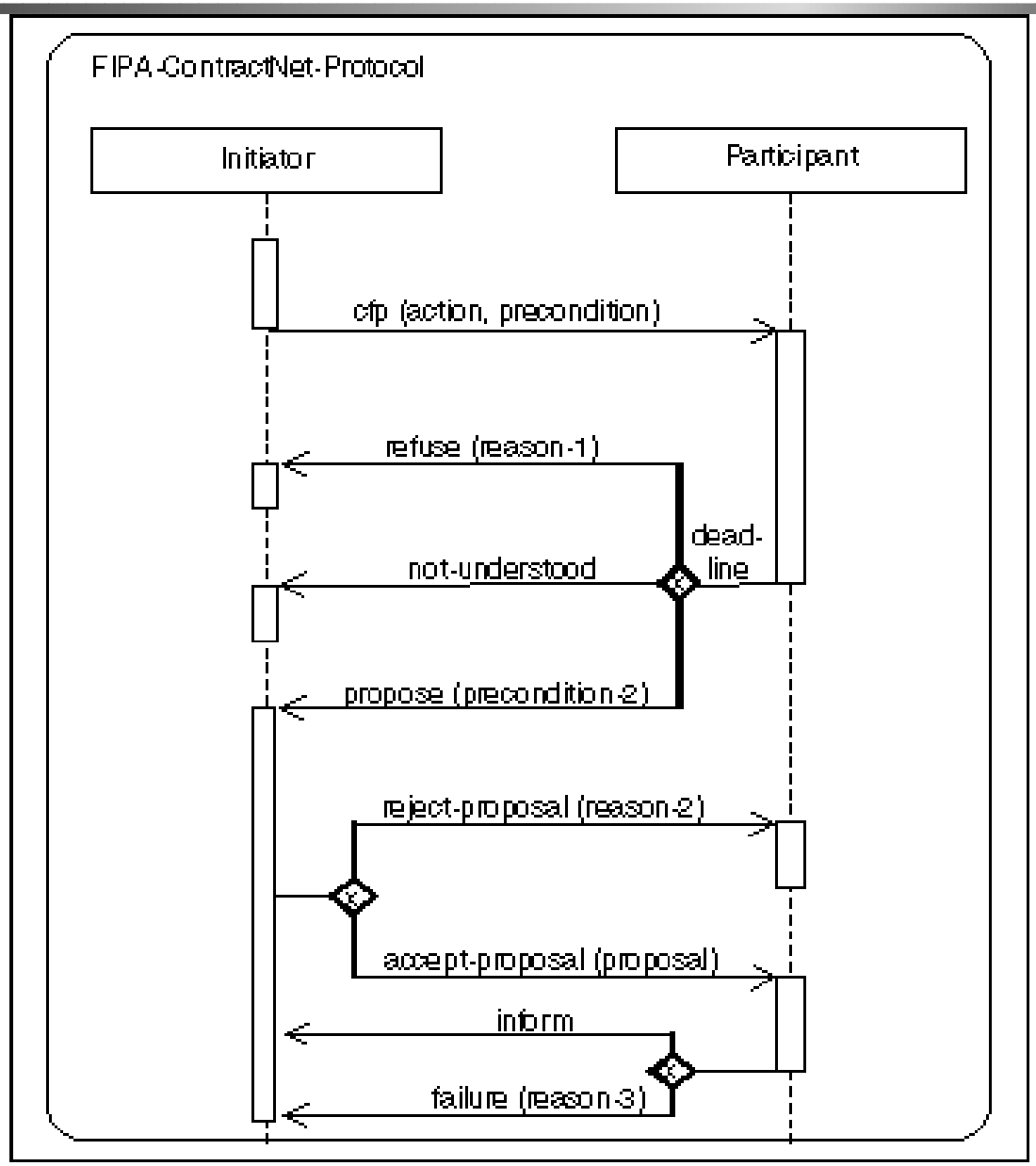
Vorteile:

- einfach
- Teilaufgaben fallen oft weg
- skalierbar (problembereichsabhängig)

Nachteile:

- Zuteilung kompliziert bei nicht homogenen Agenten:
  Zuteilungsmechanismus erforderlich
  z.B. einfache Variante des Contract Net
- Zusammenführung der Lösung nicht trivial

Aufgabenteilung: Contract Net



Contract net

Aufgabenteilung: Lösungen zusammenführen

Unterschiedliche Agenten bearbeiten Aufgaben desselben Typs:

1. Vertrauen
   - in fremde Ergebnisse
   - in eigene Ergebnisse
2. Vollständigkeit/Genauigkeit

Ergebnisse zusammenführen

(auch) als eigener Problemlösungsprozess: Ergebnisaustausch

1. beteiligte Agenten verfügen über ausschnitthafte, ungenaue Information
2. Agenten bilden eigene Teillösungen und Hypothesen (über Gesamtlösung)
3. Hypothesen werden kommuniziert
4. daraus bildet jeder Agent neue, verbesserte Hypothese usw. (2.)

Ergebnisaustausch: Probleme

Abwägung Kommunikation – Deliberation

Abwägung Vertrauen in fremde Ergebnisse

Ergebnisaustausch: Probleme

Abwägung Kommunikation – Deliberation

Abwägung Vertrauen in fremde Ergebnisse

Lösungsansatz: Blackboard zur Kommunikation

neuer Nachteil: Robustheit (ggü. Ausfall einzelner Planer) geht verloren

Ergebnisaustausch: Probleme

Abwägung Kommunikation – Deliberation

Abwägung Vertrauen in fremde Ergebnisse

Lösungsansatz: Blackboard zur Kommunikation

neuer Nachteil: Robustheit (ggü. Ausfall einzelner Planer) geht verloren

Beispiel: Distributed Constraint Satisfaction

Planen

Spezialfall: zu lösendes Problem ist Plan

1. Planen für verteilte Ausführung
2. verteiltes Planen

Planen für verteilte Ausführung

- Aus einer Zielbeschreibung, dem Satz an Operatoren und dem Startzustand kann (mit klassischen Planungsverfahren) ein partiell geordneter Plan erstellt werden.

Planen für verteilte Ausführung

- Aus einer Zielbeschreibung, dem Satz an Operatoren und dem Startzustand kann (mit klassischen Planungsverfahren) ein partiell geordneter Plan erstellt werden.
- Dieser Plan wird so in Teilpläne zerlegt, dass Abhängigkeiten möglichst nur zwischen Schritten eines Teilplanes und so wenig wie möglich zwischen den Teilplänen vorkommen.

Planen für verteilte Ausführung

- Aus einer Zielbeschreibung, dem Satz an Operatoren und dem Startzustand kann (mit klassischen Planungsverfahren) ein partiell geordneter Plan erstellt werden.
- Dieser Plan wird so in Teilpläne zerlegt, dass Abhängigkeiten möglichst nur zwischen Schritten eines Teilplanes und so wenig wie möglich zwischen den Teilplänen vorkommen.
- Die Teilpläne werden um Synchronisationspunkte ergänzt, die entsprechende Kommunikationshandlungen auslösen.

Planen für verteilte Ausführung

- Die Teilpläne werden ähnlich wie in der Aufgabenverteilung auf Agenten verteilt. Falls die Verteilung missglückt, Verwendung einer anderen Plandekomposition und so weiter. Falls kein Fehler auftritt, werden die fehlenden Bindungen an den Synchronisationspunkten um die Namen der die entsprechenden Teilpläne ausführenden Agenten ergänzt.

Planen für verteilte Ausführung

- Die Teilpläne werden ähnlich wie in der Aufgabenverteilung auf Agenten verteilt. Falls die Verteilung missglückt, Verwendung einer anderen Plandekomposition und so weiter. Falls kein Fehler auftritt, werden die fehlenden Bindungen an den Synchronisationspunkten um die Namen der die entsprechenden Teilpläne ausführenden Agenten ergänzt.
- Planausführung anstoßen und eventuell überwachen.

Verteiltes Planen

Ähnlich zur verteilten Ausführung/ zum Ergebnisaustausch: Austausch von Teilplänen

Besonderheit: Gesamtplan liegt evtl. niemals zentral vor – es wird verteilt geplant und verteilt ausgeführt

weitere Spezialisierungen:

- Überwachung des Ausführungsfortschrittes, Alternativen
- Überlappende Planung, Koordination und Ausführung

Hauptproblem: Es fehlt die unmittelbare Unterstützung der Agenten(system)konzepte durch Werkzeuge.

Wichtiger Aspekt des Projektseminars ist daher das Kennenlernen unterschiedlicher Werkzeuge, damit diese zielgerichtet für ihren jeweiligen Bereich ausgewählt und eingesetzt werden können.

Viel Spaß!