

Multi Agenten Systeme

VU SS 00, TU Wien

Teil 1 (Kapitel 1–4) basiert auf

Multi-Agent Systems (Gerhard Weiss), MIT Press, June 1999.

Es werden allgemeine Techniken und Methoden dargestellt (BDI-, Layered-, Logic based Architekturen, Decision Making, Kommunikation/Interaktion, Kontrakt Netze, Coalition Formation).

Teil 2 (Kapitel 5–9) basiert auf

Heterogenous Active Agents (Subrahmanian, Bonatti, Dix, Eiter, Kraus, Özcan and Ross), MIT Press, May 2000.

Hier wird ein spezifischer Ansatz vorgestellt, der formale Methoden aus dem logischen Programmieren benutzt, aber nicht auf PROLOG aufsetzt (Code Call Mechanismus, Aktionen, Agenten Zyklus, Status Menge, Semantiken, Erweiterungen um Beliefs, Implementierbarkeit).

Übersicht

- 1. Einführung, Terminologie**
- 2. 4 Grundlegende Architekturen**
- 3. Distributed Decision Making**
- 4. Contract Nets, Coalition Formation**
- 5. *IMPACT* Architecture**
- 6. Legacy Data and Code Calls**
- 7. Actions and Agent Programs**
- 8. Regular Agents**
- 9. Meta Agent Programs**

Kapitel 1. Einführung, Terminologie

Überblick

1.1 Allgemeines

1.2 Intelligente Agenten

1.3 Mathematische Beschreibung

1 Einführung, Terminologie

1.1 Allgemeines

Definition 1.1 (Distributed Artificial Intelligence (DAI))

Das Gebiet, das Systeme untersucht in denen mehrere autonom agierende Entitäten zusammenarbeiten um ein gegebenes Ziel zu erreichen. Diese Entitäten heissen Agenten, das Gebiet Multiagenten Systeme.

Beispiel: Robocup (Simulationsliga, Middle League)

Warum?

Grosse Informationssysteme sind **verteilt, offen, heterogen**.

Dies verlangt nach **intelligenten, interagierenden Agenten**, die autonom agieren.

Agent: Programme, die auf einer Plattform implementiert sind und Sensoren haben um auf die Umgebung zu reagieren.

Intelligent: Performanz-Maße, um Ziele zu erreichen. Rational vs. allwissend, *decision making*

Interaktiv: mit anderen Agenten (oder Menschen) durch Beobachtung der Umgebung. Alles im Hinblick auf ein Ziel.

Coordination: Cooperation vs. Competition

MAS versus Klassische DAI

(MAS) Mehrere Agenten koordinieren ihr Wissen und ihre Aktionen (Semantik beschreibt dies).

(DAI) Spezielles Problem wird in kleinere Probleme zerlegt (Knoten).
Diese Knoten haben gemeinsames Wissen. Die Lösungsmethode wird zentral vorgegeben.

Heute ist DAI oft synonym mit MAS: sowohl (1) als auch (2).

AI	DAI
Agent	System von Agenten
Intelligenz: Eigenschaft eines Agenten	Intelligenz: Eigenschaft mehrerer Agenten
Kognitive Prozesse eines Agenten	Soziale Prozesse mehrerer Agenten

Drei Fragen

- (Q1) Was ist ein Agent?
- (Q2) Wenn ein Programm P kein Agent ist, wie kann man es in einen transformieren?
- (Q3) Wenn (Q1) klar ist, welche Software Infrastruktur braucht man zur Interaktion zwischen Agenten? Welche Dienste sind notwendig?

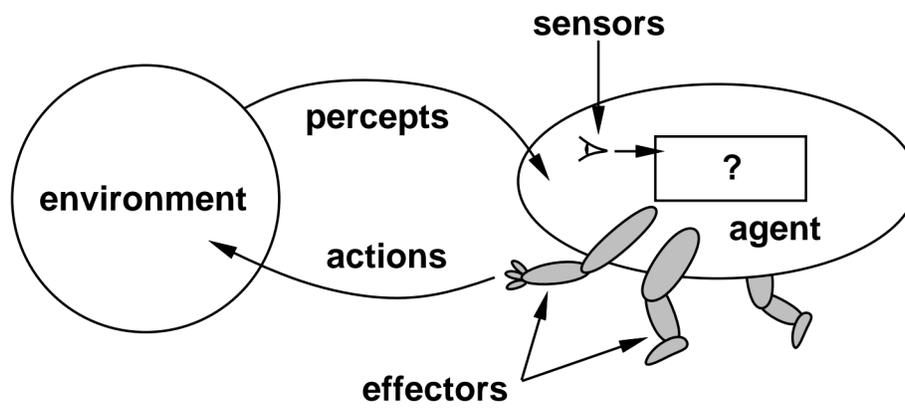
1.2 Intelligente Agenten

Definition 1.2 (Agent)

Ein Agent ist ein Computer System, das in seiner Umgebung agiert und autonome Aktionen ausführt um gewisse Ziele zu erreichen.

Lernen, Intelligenz

Umgebung ist nicht deterministisch.



Definition 1.3 (Rationaler, allwissender Agent)

*Rationale Agenten sind solche, die das **Richtige** tun.*

(Ein Maß für die Performanz wird benötigt.)

Allwissende Agenten sind solche, die die Resultate ihrer Aktionen im voraus wissen.

Rationale Agenten sind i.a. nicht allwissend!

Aphorismus von Karl Kraus: Im Zweifelsfall entscheide man sich für das richtige.

10-1

Wie ist das **Richtige** definiert und wovon hängt es ab?

1. Performanz-Maß (möglichst objektiv),
2. Percept-Sequenz: was bisher wahrgenommen wurde,
3. Wissen des Agenten über die Umgebung,
4. wie der Agent agieren kann.

Definition 1.4 (Idealer rationaler Agent)

Ein idealer rationaler Agent wählt für jede Percept-Sequenz genau die Aktion aus, die sein Performanz-Maß maximiert (bei gegebenem Wissen über die Umgebung).

Agenten können durch Abbildungen

Menge der Percept-Sequenzen \mapsto Menge der Aktionen

mathematisch beschrieben werden.

Die interne Struktur eines Agenten ist

Agent = Architektur + Programm

Agenten und ihre PAGE-Beschreibung:

Agent Type	Percepts	Actions	Goals	Environment
Medical diagnosis system	Symptoms, findings, patient's answers	Questions, tests, treatments	Healthy patient, minimize costs	Patient, hospital
Satellite image analysis system	Pixels of varying intensity, color	Print a categorization of scene	Correct categorization	Images from orbiting satellite
Part-picking robot	Pixels of varying intensity	Pick up parts and sort into bins	Place parts in correct bins	Conveyor belt with parts
Refinery controller	Temperature, pressure readings	Open, close valves; adjust temperature	Maximize purity, yield, safety	Refinery
Interactive English tutor	Typed words	Print exercises, suggestions, corrections	Maximize student's score on test	Set of students

Frage: Wie beeinflussen Eigenschaften der Umgebung das Design eines dazu passenden Agenten?

Definition 1.5 (Eigenschaften der Umgebung)

Zugänglich/Unzugänglich: Falls nicht vollst. zugängl. braucht man interne Zustände.

Determinist./Indeterm.: Falls die Umgebung unzugänglich ist, kann sie indeterministisch scheinen, obwohl sie es nicht ist.

Episodisch/Nichtepisodisch: Percept-Aktion-Sequenzen sind voneinander unabhängig. Abgeschlossene Episoden.

Statisch/Dynamisch: Dynamisch heißt: während der Agent noch überlegt, ändert sich schon die Welt. Semi-dynamisch heißt: Zwar ändert sich nicht die Welt, aber das Performanz-Maß des Agenten.

Diskret/Stetig: bezieht sich auf die Menge der Wahrnehmungen und Aktionen.

Beispiel für semi-dynamisch: Schach spielen mit Uhr. Wenn das Blättchen fällt, ist es aus.

Environment	Accessible	Deterministic	Episodic	Static	Discrete
Chess with a clock	Yes	Yes	No	Semi	Yes
Chess without a clock	Yes	Yes	No	Yes	Yes
Poker	No	No	No	Yes	Yes
Backgammon	Yes	No	No	Yes	Yes
Taxi driving	No	No	No	No	No
Medical diagnosis system	No	No	No	No	No
Image-analysis system	Yes	Yes	Yes	Semi	No
Part-picking robot	No	No	Yes	No	No
Refinery controller	No	No	No	No	No
Interactive English tutor	No	No	No	No	Yes

xbiff und **software demons** sind Agenten. Aber sicher nicht intelligent.

Definition 1.6 (Intelligente Agenten)

Ein intelligenter Agent ist ein Agent mit folgenden Eigenschaften:

1. **Reaktiv**: Reaktion auf Änderungen im Environment in gewissen Zeitabständen um ihre Ziele zu erreichen.
2. **Pro-aktiv**: Initiative übernehmen, zielgerichtetes Verhalten
3. **Sozial**: Interaktion mit anderen um ihre Ziele zu erreichen.

Pro aktiv alleine reicht nicht (C-Programme). Denn die Umgebung kann sich ändern während der Ausführung des Programms.

Schwierigkeit: Richtige Balance zwischen Pro-aktiv and Reaktiv!

1.3 Mathematische Beschreibung

Definition 1.7 (Actions \mathbf{A} , Percepts \mathbf{P} , States \mathbf{S})

Sei $\mathbf{A} := \{a_1, a_2, \dots, a_n, \dots\}$, die Menge der Aktionen, und $\mathbf{P} := \{p_1, p_2, \dots, p_n, \dots\}$ die Menge der Wahrnehmungen eines Agenten. Ferner sei $\mathbf{S} := \{s_1, s_2, \dots, s_n, \dots\}$ die Menge der Zustände, mit denen man die Umgebung beschreibt.

Was nimmt ein Agent wahr, in einem Zustand s ? Dies ist eine Funktion

$$\text{see} : \mathbf{S} \longrightarrow \mathbf{P}.$$

Wie entwickelt sich die Umgebung (der Zustand s) bei Ausführung einer Aktion a ?

Wir beschreiben dies durch eine Funktion

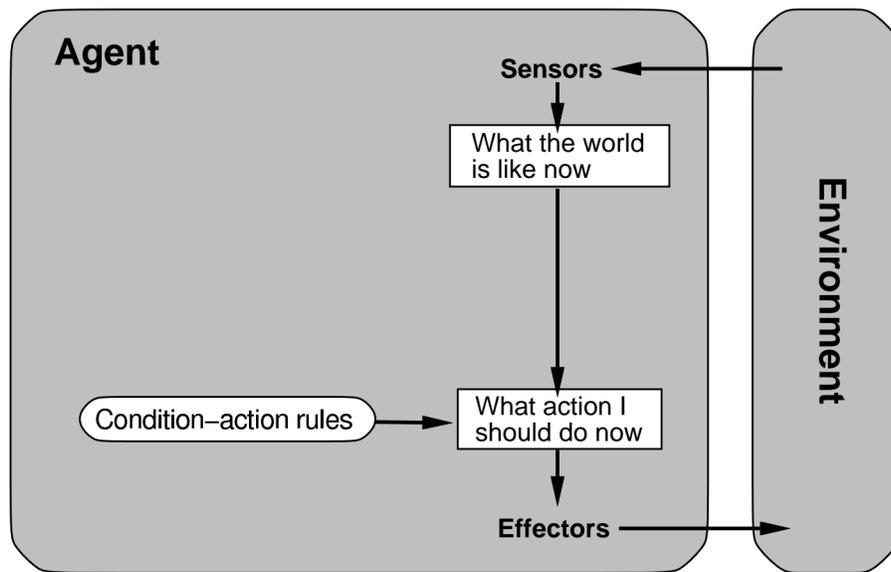
$$\text{env} : \mathbf{S} \times \mathbf{A} \longrightarrow 2^{\mathbf{S}},$$

dies schliesst nichtdeterministische Umgebungen ein.

Wie beschreiben wir nun Agenten? Wir könnten eine Funktion

$$\text{action} : P \rightarrow A$$

benutzen.



Dies ist aber sehr schwach! Besser ist es, die Historie

$$\mathbf{h} : s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} \dots s_n \xrightarrow{a_n} \dots$$

bzw. die Sequenz der Wahrnehmungen zu benutzen.

Definition 1.8 (Charakteristisches Verhalten)

Das charakteristische Verhalten eines Agenten **action** in einer Umgebung **env** ist die Menge **Hist** aller Historien **h** mit:

1. für alle n : $a_n = \mathbf{action}(\langle s_1, \dots, s_n \rangle)$,
2. für alle n : $s_n = \mathbf{env}(s_{n-1}, a_{n-1})$.

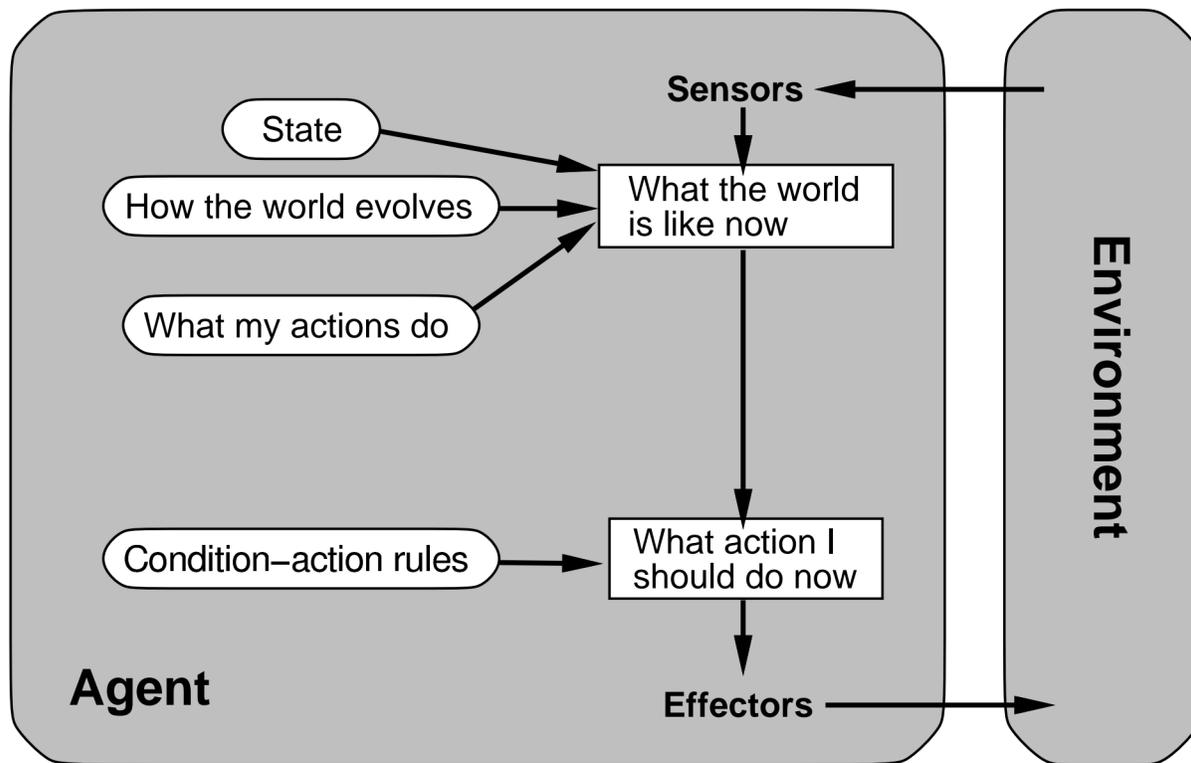
Definition 1.9 (Standard Agent *action*)

Ein Standard Agent *action* wird durch eine Funktion

$$\mathbf{action} : \mathbf{P}^* \longrightarrow \mathbf{A}$$

zusammen mit $\mathbf{see} : \mathbf{S} \longrightarrow \mathbf{P}$ und $\mathbf{env} : \mathbf{S} \times \mathbf{A} \longrightarrow 2^{\mathbf{S}}$ beschrieben.

Anstatt die gesamte Historie, bzw. \mathbf{P}^* zu verwenden, benutzt man oft auch **interne Zustände** $\mathbf{I} := \{\mathbf{i}_1, \mathbf{i}_2, \dots, \mathbf{i}_n, \dots\}$.



Definition 1.10 (Zustandsbasierter Agent *action*)

Ein zustandsbasierter Agent *action* wird durch eine Funktion

$$\text{action} : \mathbf{I} \longrightarrow \mathbf{A}$$

zusammen mit $\text{see} : \mathbf{S} \longrightarrow \mathbf{P}$ und $\text{next} : \mathbf{I} \times \mathbf{P} \longrightarrow \mathbf{I}$ beschrieben. Dabei gibt $\text{next}(i, p)$ den Nachfolgezustand von i an bei Wahrnehmung von p .

Lemma 1.1 (Äquivalenz)

Standard und zustandsbasierte Agenten sind äquivalent bezüglich ihres charakteristischen Verhaltens.

References

- Apt, K., H. Blair, and A. Walker (1988). Towards a Theory of Declarative Knowledge. In J. Minker (Ed.), *Foundations of Deductive Databases and Logic Programming*, pp. 89–148. Washington DC: Morgan Kaufmann.
- Arens, Y., C. Y. Chee, C.-N. Hsu, and C. Knoblock (1993). Retrieving and Integrating Data From Multiple Information Sources. *International Journal of Intelligent Cooperative Information Systems* 2(2), 127–158.
- Bayardo, R., et al. (1997). Infosleuth: Agent-based Semantic Integration of Information in Open and Dynamic Environments. In J. Peckham (Ed.), *Proceedings of ACM SIGMOD Conference on Management of Data*, Tucson, Arizona, pp. 195–206.
- Brink, A., S. Marcus, and V. Subrahmanian (1995). Heterogeneous Multimedia Reasoning. *IEEE Computer* 28(9), 33–39.
- Chawathe, S., et al. (1994, October). The TSIMMIS Project: Integration of Heterogeneous Information Sources. In *Proceedings of the 10th Meeting of the Information Processing Society of Japan*, Tokyo, Japan. Also available via

anonymous FTP from host db.stanford.edu, file /pub/chawathe/1994/tsimmi-overview.ps.

Genesereth, M. R. and S. P. Ketchpel (1994). Software Agents. *Communications of the ACM* 37(7), 49–53.

Rogers Jr., H. (1967). *Theory of Recursive Functions and Effective Computability*. New York: McGraw-Hill.

Wiederhold, G. (1993). Intelligent Integration of Information. In *Proceedings of ACM SIGMOD Conference on Management of Data*, Washington, DC, pp. 434–437.

Wilder, F. (1993). *A Guide to the TCP/IP Protocol Suite*. Artech House.