Multi Agenten Systeme

WS 00/01, Universität Koblenz-Landau

- **Donnerstag, 10ct (Doppelstunde)** sowie
- **vierzehntäglich, Freitag, 14ct Uhr (Doppelstunde)**.
- Vorlesung eher theoretisch, mit Schwerpunkt auf mathematisch-logischen Grundlagen.

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

Teil 1 (Kapitel 1–4) basiert auf

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

Es werden allgemeine Techniken und Methoden dargestellt.

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.

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**

Drei Fragen

(Q1) Was ist ein **Agent**?

(Q2) Wenn ein Software-Programm *P* kein Agent ist, wie kann man es **in einen Agenten transformieren**?

(Q3) Wenn (Q1) klar ist, welche **Software Infrastruktur** braucht man zur Interaktion zwischen Agenten? Welche Dienste sind notwendig?

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 bescheibt 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

10 Desiderata

1. **Agents are for everyone!** We need a method to agentize given programs.
2. Take into account that **Data is stored in a wide variety of data structures, and data is manipulated by an existing corpus of algorithms.**
3. A theory of agents must *not* depend upon the set of actions that the agent performs. Rather, **the set of actions that the agent performs must be a *parameter* that is taken into account in the semantics.**

4. **Every agent should execute actions based on some *clearly articulated* decision policy.** A **declarative** framework for articulating decision policies of agents is imperative.

5. Any agent construction framework must allow agents to perform the following types of reasoning:
   - **Reasoning about its beliefs** about other agents.
   - **Reasoning about uncertainty** in its beliefs about the world and about its beliefs about other agents.
   - **Reasoning about time**.

   These capabilities should be viewed as *extensions* to a core agent action language.

6. **Any infrastructure to support multiagent interactions *must* provide security.**
7. While the efficiency of the code underlying a software agent cannot be guaranteed (as it will vary from one application to another), **guarantees are needed that provide information on the performance of an agent relative to an oracle that supports calls to underlying software code.**

8. **We must identify efficiently computable *fragments* of the general hierarchy of languages alluded to above**, and our implementations must take advantage of the specific structure of such language fragments.

9. **A critical point is *reliability***—there is no point in a highly efficient implementation, if all agents deployed in the implementation come to a grinding halt when the agent "infrastructure" crashes.

10. The only way of testing the applicability of any theory is to **build a software system based on the theory**, to deploy a set of applications based on the theory, and to report on experiments based on those applications.

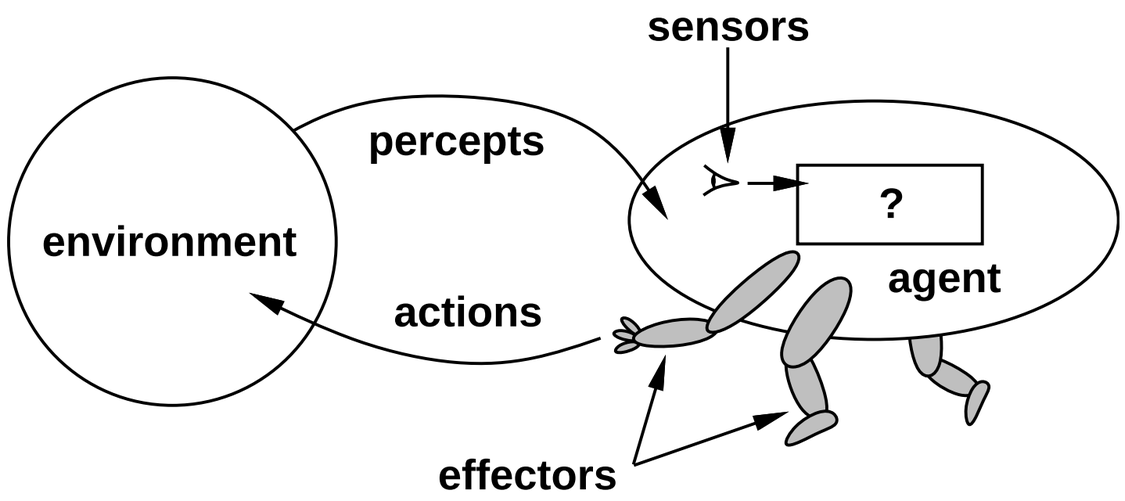
1.2 Intelligente Agenten

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.

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

$$\text{Menge der Percept-Sequenzen} \mapsto \text{Menge der Aktionen}$$

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!

Agents vs. Object Orientation

Objects have a

1. **state** (encapsulated): control over internal state,
2. message passing capabilities.

Java: private and public methods.

- Objects have control over their state, but **not over their behaviour**.
- An object can **not prevent others to use** its public methods.

Agents: They call other agents and request them to execute actions.

- Objects do it for free, agents do it for money.
- No analoga to reactiv, proactiv, social in OO.
- **MAS are multi-threaded**: each agent has a control thread.
  In OO only the sytem as a whole posesses one.

1.3 Mathematische Beschreibung

Definition 1.7 (Actions A, Percepts P, States S)
Sei $\mathbf{A} := \{\mathbf{a_1}, \mathbf{a_2}, \ldots, \mathbf{a_n}, \ldots\}$, *die Menge der* Aktionen, *und* $\mathbf{P} := \{\mathbf{p_1}, \mathbf{p_2}, \ldots, \mathbf{p_n}, \ldots\}$ *die Menge der* Wahrnehmungen *eines Agenten. Ferner sei* $\mathbf{S} := \{\mathbf{s_1}, \mathbf{s_2}, \ldots, \mathbf{s_n}, \ldots\}$ *die Menge der* Zustände, *mit denen man die Umgebung beschreibt.*

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

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

Wie entwickelt sich die Umgebung (der Zustand $\mathbf{s}$) bei Ausführung einer Aktion $\mathbf{a}$? Wir beschreiben dies durch eine Funktion

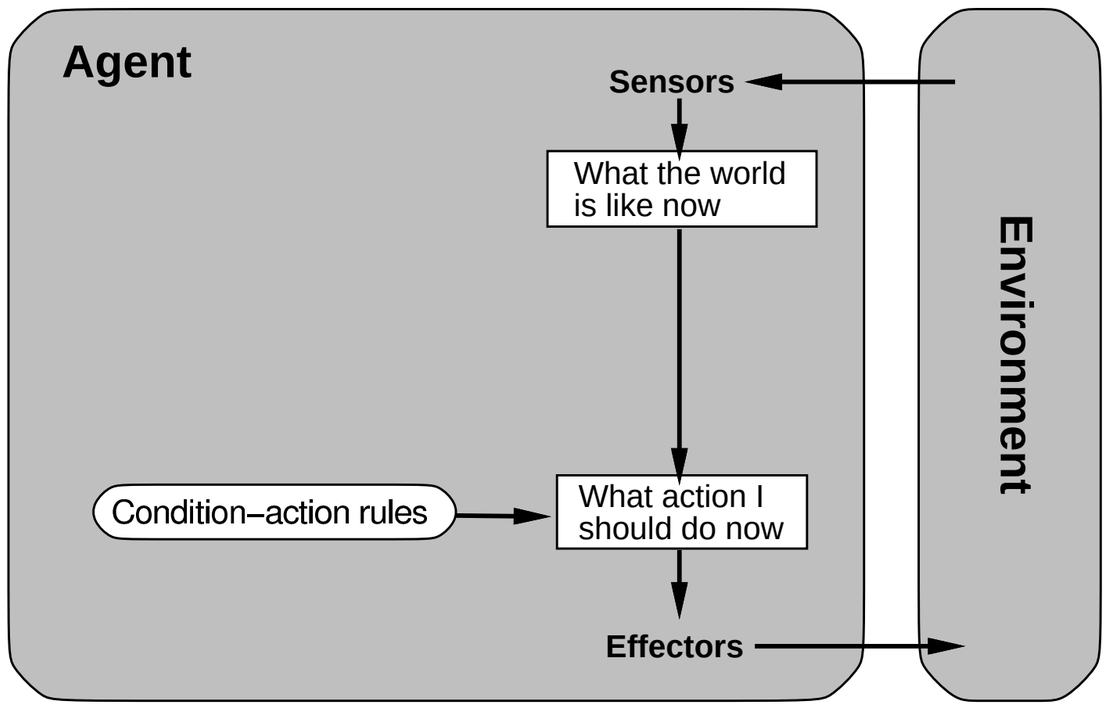
$$\mathbf{env} : \mathbf{S} \times \mathbf{A} \longrightarrow \mathbf{2^S},$$

dies schliesst nichtdeterministische Umgebungen ein.

Wie beschreiben wir nun Agenten? Wir könnten eine Funktion

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

benutzen.



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

$$\mathbf{h} : \mathbf{s_0} \rightarrow_{\mathbf{a_0}} \mathbf{s_1} \rightarrow_{\mathbf{a_1}} \ldots \mathbf{s_n} \rightarrow_{\mathbf{a_n}} \ldots$$

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* $\mathbf{h} : \mathbf{s_0} \rightarrow_{\mathbf{a_0}} \mathbf{s_1} \rightarrow_{\mathbf{a_1}} \ldots \mathbf{s_n} \rightarrow_{\mathbf{a_n}} \ldots$ *mit:*

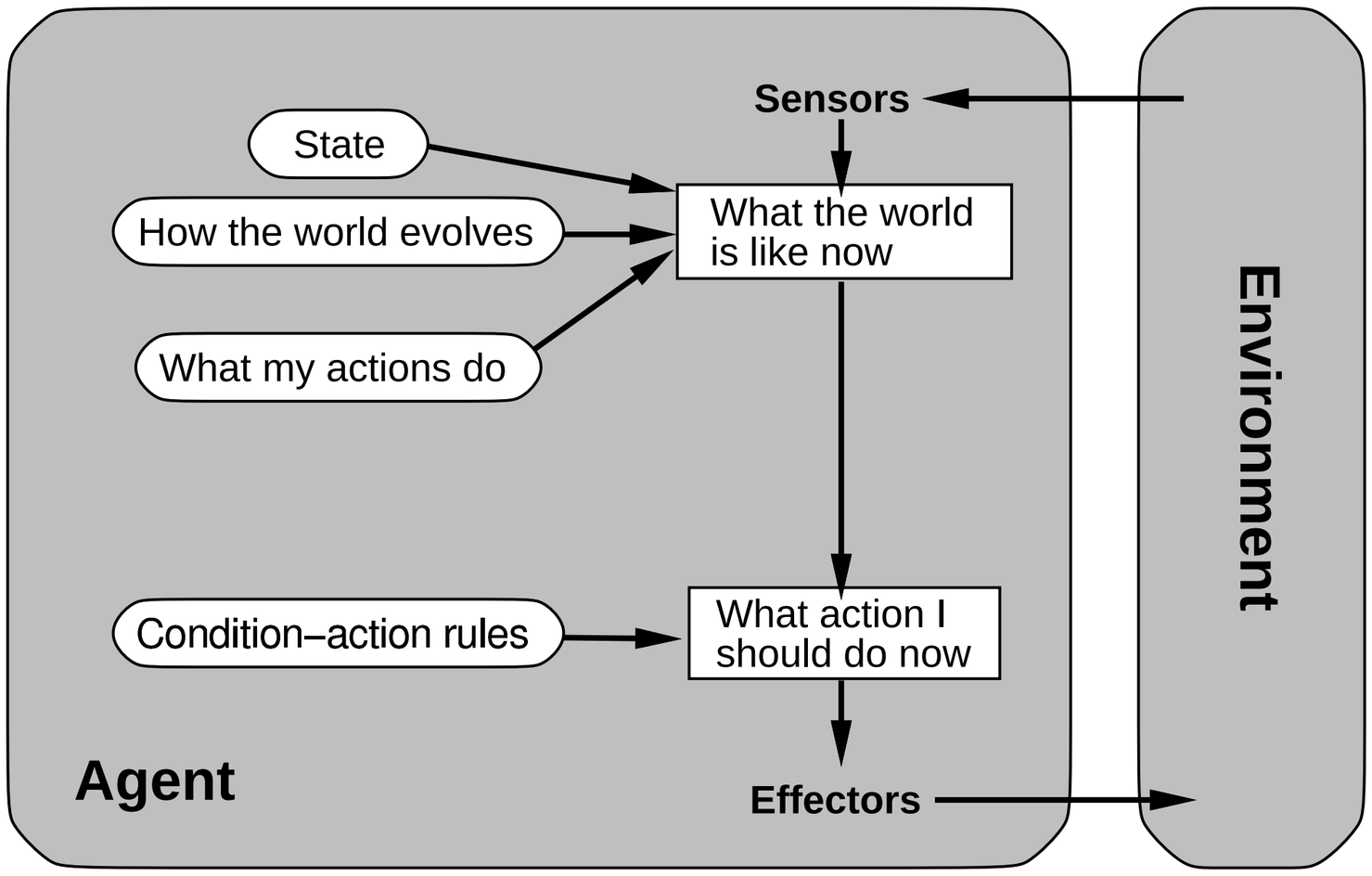
1. *für alle n:* $\mathbf{a_n} = \mathbf{action}(\langle \mathbf{s_1}, \ldots, \mathbf{s_n} \rangle)$,
2. *für alle n:* $\mathbf{s_n} = \mathbf{env}(\mathbf{s_{n-1}}, \mathbf{a_{n-1}})$.

Definition 1.9 (Standard Agent action)
Ein Standard Agent **action** *wird durch eine Funktion*

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

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

Anstatt die gesamte Historie, bzw. $\mathbf{P}^*$ zu verwenden, benutzt man oft auch **interne Zustände** $\mathbf{I} := \{\mathbf{i_1}, \mathbf{i_2}, \ldots \mathbf{i_n}, \ldots\}$.


Sensors
State
How the world evolves
What the world is like now
What my actions do
Condition–action rules
What action I should do now
Agent
Effectors
Environment

Definition 1.10 (Zustandsbasierter Agent action)

Ein zustandsbasierter Agent **action** *wird durch eine Funktion*

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

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

Definition 1.11 (Charakteristisches Verhalten)

Das charakteristische Verhalten eines zustandsbasierten Agenten **action** *in einer Umgebung* **env** *ist die Menge aller Sequenzen*

$$(\mathbf{i}_0, p_0) \rightarrow_{a_0} (\mathbf{i}_1, p_1) \rightarrow_{a_1} \ldots \rightarrow_{a_n} (\mathbf{i}_n, p_n), \ldots$$

mit

1. *für alle n:* $\mathbf{a_n} = \mathbf{action}(\mathbf{i_n})$,
2. *für alle n:* $\mathbf{next}(\mathbf{i_n}, p_n) = \mathbf{i_{n+1}}$,

Lemma 1.1 (Äquivalenz)
Standard und zustandsbasierte Agenten sind äquivalent bezüglich ihres charakteristischen Verhaltens.