Introduction to Artificial Intelligence

Intelligent Agents

Bernhard Beckert

UNIVERSITÄT KOBLENZ-LANDAU

Wintersemester 2003/2004
Outline

- Agents and environments
- PEAS (Performance, Environment, Actuators, Sensors)
- Environment types
- Agent types
- Example: Vacuum world
Agents and environments

Agents include

- humans
- robots
- software robots (softbots)
- thermostats
- etc.
Agent functions and programs

Agent function

An agent is completely specified by the agent function

\[ f : \mathcal{P}^* \rightarrow \mathcal{A} \]

mapping percept sequences to actions
Agent functions and programs

Agent program

- runs on the physical architecture to produce $f$
- takes a single percept as input
- keeps internal state

function $\text{SKELETON-AGENT}(\text{percept})$ returns action

static: memory /* the agent’s memory of the world */

memory ← $\text{UPDATE-MEMORY}(\text{memory, percept})$
 action ← $\text{CHOOSE-BEST-ACTION}(\text{memory})$
memory ← $\text{UPDATE-MEMORY}(\text{memory, action})$

return action
AIMA code

Available at

http://aima.cs.berkeley.edu/code.html

in different languages (Java, Lisp, ...)

Code for each topic divided into four directories

agents: code defining agent types and programs
algorithms: code for the methods used by the agent programs
environments: code defining environment types, simulations
domains: problem types and instances for input to algorithms

For experiments

Often algorithms on domains rather than agents in environments
Rationality

Goal

Specified by performance measure, defining a numerical value for any environment history

Rational action

Whichever action maximizes the expected value of the performance measure given the percept sequence to date
Rationality

Goal

Specified by performance measure, defining a numerical value for any environment history

Rational action

Whichever action maximizes the expected value of the performance measure given the percept sequence to date

Note

\[
\text{rational} \neq \text{omniscient} \\
\text{rational} \neq \text{clairvoyant} \\
\text{rational} \neq \text{successful}
\]
Rationality

Goal

Specified by performance measure, defining a numerical value for any environment history

Rational action

Whichever action maximizes the expected value of the performance measure given the percept sequence to date

Note

rational $\neq$ omniscient
rational $\neq$ clairvoyant
rational $\neq$ successful

Agents need to: gather information, explore, learn, ...
Example: Designing an automated taxi

Performance

Environment

Actuators

Sensors
PEAS: The setting for intelligent agent design

Example: Designing an automated taxi

Performance: safety, reach destination, maximize profits, obey laws, passenger comfort, ...

Environment

Actuators

Sensors
Example: Designing an automated taxi

**Performance**
safety, reach destination, maximize profits,
obey laws, passenger comfort, ... 

**Environment**
streets, traffic, pedestrians, weather, customers, ... 

**Actuators**

**Sensors**
PEAS: The setting for intelligent agent design

Example: Designing an automated taxi

Performance
- safety, reach destination, maximize profits,
- obey laws, passenger comfort, ...

Environment
- streets, traffic, pedestrians, weather, customers, ...

Actuators
- steer, accelerate, brake, horn, speak/display, ...

Sensors
Example: Designing an automated taxi

**Performance**
safety, reach destination, maximize profits, obey laws, passenger comfort, ...

**Environment**
streets, traffic, pedestrians, weather, customers, ...

**Actuators**
steer, accelerate, brake, horn, speak/display, ...

**Sensors**
video, accelerometers, gauges, engine sensors, keyboard, GPS, ...
Example: Medical diagnosis system

Performance

Environment

Actuators

Sensors
PEAS: The setting for intelligent agent design

Example: Medical diagnosis system

Performance
Healthy patient, minimize costs, avoid lawsuits, ...
Example: Medical diagnosis system

Performance: Healthy patient, minimize costs, avoid lawsuits, ...

Environment: patient, hospital, staff, ...

Actuators

Sensors
PEAS: The setting for intelligent agent design

Example: Medical diagnosis system

Performance Healthy patient, minimize costs, avoid lawsuits, ...

Environment patient, hospital, staff, ...

Actuators questions, tests, diagnoses, treatments, referrals, ...

Sensors
PEAS: The setting for intelligent agent design

Example: Medical diagnosis system

Performance
Healthy patient, minimize costs, avoid lawsuits, ...

Environment
patient, hospital, staff, ...

Actuators
questions, tests, diagnoses, treatments, referrals, ...

Sensors
keyboard (symptoms, test results, answers), ...
Environment types

Fully observable  (otherwise: partially observable)

Agent’s sensors give it access to the complete state of the environment at each point in time
Environment types

**Fully observable**  (otherwise: **partially observable**)

Agent’s sensors give it access to the complete state of the environment at each point in time

**Deterministic**  (otherwise: **stochastic**)

The next state of the environment is completely determined by the current state and the action executed by the agent
(strategic: deterministic except for behavior of other agents)
Environment types

Fully observable  (otherwise: partially observable)

Agent’s sensors give it access to the complete state of the environment at each point in time

Deterministic  (otherwise: stochastic)

The next state of the environment is completely determined by the current state and the action executed by the agent (strategic: deterministic except for behavior of other agents)

Episodic  (otherwise: sequential)

The agent’s experience is divided atomic, independent episodes (in each episode the agent perceives and then performs a single action)
Environment types

**Static**  (otherwise: **dynamic**)

Environment can change while the agent is deliberating
(semidynamic: not the state but the performance measure can change)
Environment types

Static  (otherwise: dynamic)

Environment can change while the agent is deliberating
(semidynamic: not the state but the performance measure can change)

Discrete  (otherwise: continuous)

The environment’s state, time, and the agent’s percepts and actions
have discrete values
Environment types

Static (otherwise: dynamic)

Environment can change while the agent is deliberating
(semidynamic: not the state but the performance measure can change)

Discrete (otherwise: continuous)

The environment’s state, time, and the agent’s percepts and actions
have discrete values

Single agent (otherwise: multi-agent)

Only one agent acts in the environment
## Environment types

<table>
<thead>
<tr>
<th>Fully observable</th>
<th>Crossword puzzle</th>
<th>Chess</th>
<th>Backgammon</th>
<th>Internet shopping</th>
<th>Taxi</th>
<th>Part-picking robot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deterministic</td>
<td>yes</td>
<td>yes</td>
<td></td>
<td></td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>Episodic</td>
<td>no</td>
<td>no</td>
<td></td>
<td></td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>Static</td>
<td>yes</td>
<td>semi</td>
<td></td>
<td></td>
<td>semi</td>
<td></td>
</tr>
<tr>
<td>Discrete</td>
<td>yes</td>
<td>no</td>
<td></td>
<td></td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>Single agent</td>
<td>yes</td>
<td>no</td>
<td></td>
<td></td>
<td>no</td>
<td></td>
</tr>
</tbody>
</table>
## Environment types

<table>
<thead>
<tr>
<th></th>
<th>Crossword puzzle</th>
<th>Chess</th>
<th>Backgammon</th>
<th>Internet shopping</th>
<th>Taxi</th>
<th>Part-picking robot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully observable</td>
<td>yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deterministic</td>
<td>yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Episodic</td>
<td>no</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Static</td>
<td>yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discrete</td>
<td>yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Single agent</td>
<td>yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

B. Beckert: Einführung in die KI / KI für IM – p.12
### Environment types

<table>
<thead>
<tr>
<th>Feature</th>
<th>Crossword puzzle</th>
<th>Chess</th>
<th>Backgammon</th>
<th>Internet shopping</th>
<th>Taxi</th>
<th>Part-picking robot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully observable</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Deterministic</td>
<td>yes</td>
<td>strat.</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Episodic</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Static</td>
<td>yes</td>
<td>semi</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Discrete</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Single agent</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
</tbody>
</table>
# Environment types

<table>
<thead>
<tr>
<th></th>
<th>Crossword puzzle</th>
<th>Chess</th>
<th>Backgammon</th>
<th>Internet shopping</th>
<th>Taxi</th>
<th>Part-picking robot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully observable</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deterministic</td>
<td>yes</td>
<td>strat.</td>
<td>no</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Episodic</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Static</td>
<td>yes</td>
<td>semi</td>
<td>yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discrete</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Single agent</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Environment types

<table>
<thead>
<tr>
<th></th>
<th>Crossword puzzle</th>
<th>Chess</th>
<th>Backgammon</th>
<th>Internet shopping</th>
<th>Taxi</th>
<th>Part-picking robot</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fully observable</strong></td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Deterministic</strong></td>
<td>yes</td>
<td>strat.</td>
<td>no</td>
<td>(yes)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Episodic</strong></td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td><strong>Static</strong></td>
<td>yes</td>
<td>semi</td>
<td>yes</td>
<td>semi</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Discrete</strong></td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Single agent</strong></td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td></td>
</tr>
</tbody>
</table>
## Environment types

<table>
<thead>
<tr>
<th>Feature</th>
<th>Crossword puzzle</th>
<th>Chess</th>
<th>Backgammon</th>
<th>Internet shopping</th>
<th>Taxi</th>
<th>Part-picking robot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully observable</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Deterministic</td>
<td>yes</td>
<td>strat.</td>
<td>no</td>
<td>(yes)</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Episodic</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Static</td>
<td>yes</td>
<td>semi</td>
<td>yes</td>
<td>semi</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Discrete</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Single agent</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>

B. Beckert: Einführung in die KI / KI für IM – p.12
# Environment types

<table>
<thead>
<tr>
<th></th>
<th>Crossword puzzle</th>
<th>Chess</th>
<th>Backgammon</th>
<th>Internet shopping</th>
<th>Taxi</th>
<th>Part-picking robot</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fully observable</strong></td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td><strong>Deterministic</strong></td>
<td>yes</td>
<td>strat.</td>
<td>no</td>
<td>(yes)</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td><strong>Episodic</strong></td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td><strong>Static</strong></td>
<td>yes</td>
<td>semi</td>
<td>yes</td>
<td>semi</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td><strong>Discrete</strong></td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td><strong>Single agent</strong></td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
</tbody>
</table>
Environment types

The real world is

- partially observable
- stochastic
- sequential
- dynamic
- continuous
- multi-agent
Agent types

Four basic types

(in order of increasing generality)

- simple reflex agents
- reflex agents with state
- goal-based agents
- utility-based agents

All these can be turned into learning agents
Simple reflex agents

Agent

Environment

Sensors

What the world is like now

Condition/action rules

What action I should do now

Actuators
Model-based reflex agents
Goal-based agents

- **Agent**
  - **State**
  - **What the world is like now**
  - **What it will be like if I do action A**
  - **Goals**
  - **What action I should do now**

- **Environment**
  - **Sensors**
  - **How the world evolves**
  - **What my actions do**
  - **Actuators**
Utility-based agents

- State
- How the world evolves
- What my actions do
- Utility

Environment

Agent

Sensors

- What the world is like now

Actuators

- What it will be like if I do action A
- How happy I will be in such a state
- What action I should do now
The vacuum-cleaner world
The vacuum-cleaner world

Perceps

- location
- dirty / not dirty
The vacuum-cleaner world

Percepts
- location
- dirty / not dirty

Actions
- left
- right
- suck
- noOp
The vacuum-cleaner world

Performance measure

+100 for each piece of dirt cleaned up
−1 for each action
−1000 for shutting off away from home

Environment

– grid
– dirt distribution and creation
The vacuum-cleaner world