Vorlesung
Grundlagen der
Künstlichen Intelligenz

Reinhard Lafrenz / Prof. A. Knoll

Robotics and Embedded Systems
Department of Informatics – I6
Technische Universität München

www6.in.tum.de
lafrenz@in.tum.de
089-289-18136
Room 03.07.055

Wintersemester 2012/13 22.10.2012
Chapter 2

Intelligent Agents
What is an Agent?

- An *agent* is anything that can be viewed as *perceiving* its *environment* through *sensors* and *acting* upon that environment through *actuators*

- Sensing modalities

<table>
<thead>
<tr>
<th>Human</th>
<th>Robot</th>
</tr>
</thead>
<tbody>
<tr>
<td>vision</td>
<td>(computer) vision</td>
</tr>
<tr>
<td>hearing</td>
<td>Audio (or speech ) recognition</td>
</tr>
<tr>
<td>smelling</td>
<td>odometry</td>
</tr>
<tr>
<td>touch</td>
<td>force/torque</td>
</tr>
<tr>
<td>proprioceptive sensors</td>
<td>encoders and force/torque</td>
</tr>
</tbody>
</table>
What is an Agent?

- An *agent* is anything that can be viewed as *perceiving* its *environment* through *sensors* and *acting* upon that environment through *actuators*.

- Acting

<table>
<thead>
<tr>
<th>Human</th>
<th>Robot</th>
</tr>
</thead>
<tbody>
<tr>
<td>arms and legs</td>
<td>arms, legs, and/or wheels</td>
</tr>
<tr>
<td>hands</td>
<td>end-effectors (grippers, tools for e.g. drilling, welding)</td>
</tr>
<tr>
<td>other body motion</td>
<td>depending on system</td>
</tr>
<tr>
<td>facial expression</td>
<td>artificial emotion expression</td>
</tr>
<tr>
<td>talking</td>
<td>speech output</td>
</tr>
</tbody>
</table>
What is an Agent?

- Goal: Design an agent that acts “successfully“ in its environment
Examples of (robotic) agents
The Vacuum World

- Percepts: location and contents: < A|B, Clean|Dirty >
  e.g. <A, Dirty>
- Actions: Left, Right, Suck, NoOp

- How to describe the agent’s behavior?
The Vacuum Cleaning Agent

- The **agent function** maps from percept histories to actions:
  \[
  [f: \mathcal{P}^* \rightarrow \mathcal{A}]
  \]

- The **agent program** is an implementation of \( f \), given an agent and its architecture

- \( \text{agent} = \text{architecture} + \text{program} \)

- How to describe and implement \( f \)?
The Vacuum Cleaning Agent

- Defining the agent function by a table

<table>
<thead>
<tr>
<th>Sequence of percepts</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; A, Clean &gt;</td>
<td>Right</td>
</tr>
<tr>
<td>&lt; A, Dirty &gt;</td>
<td>Suck</td>
</tr>
<tr>
<td>&lt; B, Clean &gt;</td>
<td>Left</td>
</tr>
<tr>
<td>&lt; B, Dirty &gt;</td>
<td>Suck</td>
</tr>
<tr>
<td>&lt; A, Clean &gt;, &lt; A, Clean &gt;</td>
<td>Right</td>
</tr>
<tr>
<td>&lt; A, Clean &gt;, &lt; A, Dirty &gt;</td>
<td>Suck</td>
</tr>
<tr>
<td>&lt; A, Clean &gt;, &lt; B, Clean &gt;</td>
<td>Left</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>&lt; A, Clean &gt;, &lt; B, Clean &gt;, &lt; A, Dirty &gt;</td>
<td>Suck</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

- Is this an appropriate representation?
The Vacuum Cleaning Agent

- What is the (or a?) correct function
- Can it be implemented in a compact way?

```
function REFLEX-VACUUM-AGENT( <location, status> )
returns an action

if status = Dirty then return Suck
else if location = A then return Right
else if location = B then return Left
```

- This describes a finite-state automaton or finite-state machine
- Does this lead to a rational behavior?
Rational agents

- Doing the “right“ thing based on the available information

- Rationality depends on
  – Performance measure: An objective criterion for success of an agent's behavior (degree of goal achievement)
  – A priori knowledge about the environment
  – Possible actions
  – Sequence of percepts

- For each possible percept sequence, a rational agent should select an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and the built-in knowledge
Rational Agents

- Rational vs. Omniscient
- Maximizing the expected performance, not the actual one

Examples:
- Distance to goal given the current state estimation (map)
- Number of work pieces processed by an industrial robot
- Value of collected information (software agent)
Rational Agents

- Autonomy: An agent is *autonomous* if its behavior is determined by its own experience, i.e. the sequence of percepts. Autonomy implies the ability to learn and adapt.

- Sufficient time and sufficient number of percepts are needed for the ability to adapt to new situations and to learn new actions which are adequate in the given situation
PEAS description of a rational agent

- **Performance Measure**: Assess the performance of an agent
- **Environment**: The environment in which the agent is able to act and to achieve its goals
- **Actions**: The actions the agent is able to perform
- **Sensors**: The information an agent can perceive
PEAS description of a thermostat

- **Performance measure**: constant room temperature
- **Environment**: room, house, car
- **Actions**: open or close valves, NoOp
- **Sensors**: “data“ from temperature sensor

- Percepts: temperature $T$ at a given time
- Actions: open or close valve by $x\%$
The agent program of a thermostat

\[ \Delta T = T_{\text{current}} - T_{\text{desired}} \]

- Agent function uses \( \Delta T \) to select an appropriate action

\[ f: \Delta T \rightarrow A \]

```plaintext
function THERMOSTAT-AGENT(temperature)
returns an action

if temperature <= desired then return OpenValve(k* \( \Delta T \))
else if temperature > desired then return CloseValve(k*\( \Delta T \))
else return NoOp
```

- Is this sufficient? (consider that the general agent function takes the sequence of percepts)
The agent program of a thermostat

- How to measure the performance?
  - At each point in time?
    - Oscillation possible
  - Average over a certain period?
    - Slow reaction possible
  - Taking more than one sensor reading into account
    - How to determine the next action?
  - After goal achievement?
    - How to determine the next action?

- This also affects the implementation and the behavior of the agent function
## PEAS descriptions of other agents

<table>
<thead>
<tr>
<th>Agent</th>
<th>Performance measure</th>
<th>Environment</th>
<th>Actions</th>
<th>Sensors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Medical diagnosis system</td>
<td>Healthy patient, minimize costs, lawsuits</td>
<td>Patient, hospital, staff</td>
<td>Screen display (questions, tests, diagnoses, treatments, etc.)</td>
<td>Keyboard (entry of symptoms, findings, patient's answers)</td>
</tr>
<tr>
<td>Autonomous taxi driver</td>
<td>Safe, fast, legal, comfortable trip, maximize profits</td>
<td>Roads, traffic, pedestrians, customers</td>
<td>Steering wheel, accelerator, brake, horn, …</td>
<td>Cameras, sonar, GPS, odometer, voice recognition</td>
</tr>
<tr>
<td>Controller in chemical industry</td>
<td>Puirity, quantity, safety,</td>
<td>Industrial environment</td>
<td>Pumps, valves, heating, etc.</td>
<td>Sensors for temp., pressure, spectrometers, …</td>
</tr>
<tr>
<td>Part-picking robot</td>
<td>Percentage of parts in correct bins</td>
<td>Conveyor belt with parts, bins</td>
<td>Robot arm and hand</td>
<td>Camera, joint angle sensors</td>
</tr>
<tr>
<td>Interactive English tutor</td>
<td>Maximize student’s score on test</td>
<td>Set of students, examiners</td>
<td>Display of exercises, suggestions, corrections</td>
<td>Keyboard input</td>
</tr>
</tbody>
</table>
# PEAS descriptions of other agents

<table>
<thead>
<tr>
<th>Agent</th>
<th>Performance measure</th>
<th>Environment</th>
<th>Actions</th>
<th>Sensors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soccer playing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Searching for titanium on the sea floor</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bidding for an item at an auction</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Classification of environments

<table>
<thead>
<tr>
<th>Fully observable</th>
<th>Partially observable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single agent</td>
<td>Multiple agents</td>
</tr>
<tr>
<td>Deterministic</td>
<td>Stochastic</td>
</tr>
<tr>
<td>Episodic</td>
<td>Sequential</td>
</tr>
<tr>
<td>Static</td>
<td>Dynamic</td>
</tr>
<tr>
<td>Discrete</td>
<td>Continuous</td>
</tr>
</tbody>
</table>

- How would you characterize
  - Medical diagnosis systems
  - Playing chess (with clock)
  - Playing soccer
  - Taxi driving
Taxonomy of agents

- Table-driven agent

```
function TABLE-DRIVEN-AGENT(percept)
returns an action

percepts: a sequence, initially empty
table: the table with percept sequences and actions, fully specified

append percept to the end of percepts
action ← lookup (table, percepts)
return action
```

- Might get very large
- Generation takes time (by programmer or learning algorithm)
Taxonomy of agents

- Table-driven agent
  - What is the size of the table of the agent has $P$ possible percepts and works for $T$ time steps?
  - What would be the table size for a 1h drive with an autonomous taxi using a 30frames/sec 640x40 / 24bit camera as the percepts?

27 MB/sec $\Rightarrow$ table size $10^{250,000,000,000}$

State space for Chess: $10^{150}$
Estimated number of atoms in the universe: $10^{80}$

... practically infeasible
Taxonomy of agents

- Simple reflex agent

- Direct use of sensory data often intractable
Taxonomy of agents

- Simple reflex agent

```plaintext
function SIMPLE-REFLEX-AGENT(percept)  
returns an action  

rules: a set of condition-action rules  

state ← interpret-input (percepts)  
rule ← rule-match(state, rules)  
action ← rule-action (rule)  
return action
```

- Interpretation of percepts
Taxonomy of agents

- Agents with internal world model

  - If history of percepts is needed to select an action, it must be represented in appropriate way
Taxonomy of agents

- Reflex agent with internal state

```plaintext
function REFLEX-AGENT-WITH-STATE(percept)
returns an action

state: a description of the current world state
rules: a set of condition-action rules

state ← interpret-input (state, percepts)
rule ← rule-match(state, rules)
action ← rule-action (rule)
state ← update-state(state, action)

return action
```
Taxonomy of agents

- Goal-based agents

- Sole percepts are often not sufficient to choose an action
- Explicit representation of goals and using it for action selection
Taxonomy of agents

- Utility-based agents
  - In most cases, several actions can be performed in a given state
  - Choose the action based on the utility of the expected next state
Taxonomy of agents

- Learning agents
Representation of states and transitions

- (a) Atomic: “blackbox“, no internal structure, e.g. a name
  - Used e.g. for search algorithms, games, HMM, MDP
- (b) Factored: vector of variables or attributes
  - Used e.g. for reasoning with propositional logic, constraint satisfaction
- (c) Structured: objects and relations
  - Bases for first-order logic, relational (or object-oriented) databases, probability models, knowledge-based learning, etc.
Discussion

Are the following statements true or false? Why?

a) An agent that senses only partial information about the state cannot be perfectly rational.

b) There exist task environments in which no pure reflex agent can behave rationally.

c) There exists a task environment in which every agent is rational.

d) The input to an agent program is the same as the input to the agent function.

e) Every agent function is implementable by some program/machine combination.
Discussion

f) Suppose an agent selects its action uniformly at random from the set of possible actions. There exists a deterministic task environment in which this agent is rational.

g) It is possible for a given agent to be perfectly rational in two distinct task environments.

h) Every agent is rational in an unobservable environment.

i) A perfectly rational poker-playing agent never loses.
Summary

- An agent is an entity that perceives and acts
- It consists of an architecture and a program
- A ideal rational agent is expected to maximize its performance measure, given the evidence provided by the percept sequence and the built-in knowledge
- There are different types of environments, some are more challenging than others
  - E.g. partially observable, stochastic, sequential, dynamic, continuous
- There are different types of agents
  - Reflex agents, only reacting to the percepts
  - Goal-based agents, trying to achieve given goal(s)
  - Utility-based agents, maximizing their performance