EBERHARD KARLS UNIVERSITÄT TÜBINGEN

MATHEMATISCH-NATURWISSENSCHAFTLICHE FAKULTÄT

Artificial Intelligence
Chapter 2: Intelligent Agents

Andreas Zell

After the Textbook: Artificial Intelligence,
A Modern Approach
by Stuart Russel and Peter Norvig (3rd Edition)

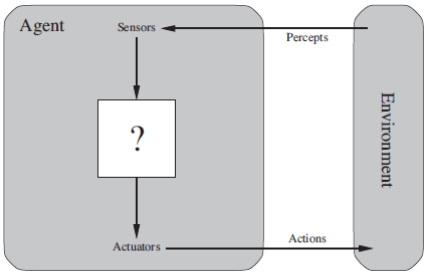
2. Intelligent Agents

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

- Rational agent: one that behaves as well as possible
- This behaviour depends on the environment:
- Some environments are more difficult than others

2.1 Agents and Environments

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

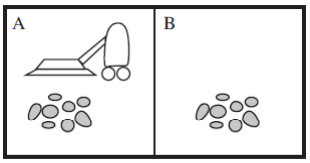


- Human agent:
  - Sensors: eyes, ears, and other organs,
  - Actuators: hands, legs, mouth, other body parts
- Robotic agent:
  - Sensors: cameras, sonars, laser range finders;
  - Actuators: various motors, gripper

2.1 Agents and Environments

- Percept: an agent's perceptual inputs at any time.
- Percept sequence: history of percepts
- An agent's actions may depend on its whole history
- Agent function: mapping of percept sequence to action: $f : P^* \rightarrow A$
- Agent program: the internal implementation of the abstract mathematical agent function

Example Vacuum Cleaner Miniworld



- Only two locations: A, B
- Floor clean or dirty
- Percept: [A, clean]
- Actions: right, left, suck

Simple agent function:

Percept sequence	Action
[A, clean]	right
[A, dirty]	suck
[B, clean]	left
[B, dirty]	suck
[A, clean], [A, clean]	right
[A, clean], [A, dirty]	suck
[A, clean], [B, clean]	left
[A, clean], [B, dirty]	suck
[A, dirty], [A, clean]	right
…	

2.2 The Concept of Rationality

- A rational agent is “one that does the right thing”.
- …but what is the right thing?
- Need performance measures.
- Performance measure: objective criterion for success of an agent’s behaviour
- E.g., performance measure of the vacuum-cleaner agent could be amount of dirt cleaned up, amount of time taken, amount of electricity consumed, cleanliness of floor afterwards, …

2.2 The Concept of Rationality

- Rationality here depends on four things:
  1. The performance measure that defines the criterion of success
  2. The agent's prior knowledge of the environment
  3. The actions that the agent can perform
  4. The agent's percept sequence to date.
- Rational agent: 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 whatever built-in knowledge the agent has.

Omniscience, Learning, Autonomy

- Rationality is distinct from omniscience (all-knowing with infinite knowledge)
- Agents can perform actions in order to modify future percepts so as to obtain useful information (information gathering, exploration)
- A rational agent should not only gather information, but also learn as much as possible from what it perceives
- An agent is autonomous if its behavior is determined by its own experience (with ability to learn and adapt). Rational agents should be autonomous.

2.3 The Nature of Environments

- Specifying the task environment:
- PEAS: Performance measure, Environment, Actuators, Sensors

- Example agent: Automated taxi driver:
  - Performance measure: Safe, fast, legal, comfortable trip, maximize profits
  - Environment: Roads, other traffic, pedestrians, customers
  - Actuators: Steering wheel, accelerator, brake, signal, horn
  - Sensors: Cameras, sonar, speedometer, GPS, odometer, engine sensors, keyboard

Specifying the Task Environment

- Example Agent: Medical diagnosis system
  - Performance measure: Healthy patient, minimize costs, lawsuits
  - Environment: Patient, hospital, staff
  - Actuators: Screen display (questions, tests, diagnoses, treatments, referrals)
  - Sensors: Keyboard (entry of symptoms, findings, patient's answers)

- Example Agent: Part-picking robot
- Example Agent: Interactive English tutor

Properties of task environments

- Fully observable (vs. partially observable): An agent's sensors give it access to the complete state of the environment at each point in time.
- Deterministic (vs. stochastic): The next state of the environment is completely determined by the current state and the action executed by the agent. (If the environment is deterministic except for the actions of other agents, then the environment is strategic)
- Episodic (vs. sequential): The agent's experience is divided into atomic "episodes" (each episode consists of the agent perceiving and then performing a single action), and the choice of action in each episode depends only on the episode itself.

Properties of task environments

- Static (vs. dynamic): The environment is unchanged while an agent is deliberating. (The environment is semidynamic if the environment itself does not change with the passage of time but the agent's performance score does)
- Discrete (vs. continuous): A limited number of distinct, clearly defined percepts and actions.
- Single agent (vs. multiagent): An agent operating by itself in an environment.

Examples of Task Environments

Task Environment	Observ-able	Deter-ministic	Episodic	Static	Discrete	Agents
Crossword puzzle	Fully	Det.	Sequ.	Static	Discrete	Single
Chess with a clock	Fully	Strateg.	Sequ.	Semi	Discrete	Multi
Poker	Partially	Stoch.	Sequ.	Static	Discrete	Multi
Backgammon	Fully	Stoch.	Sequ.	Static	Discrete	Multi
Taxi driving	Partially	Stoch.	Sequ.	Dynam.	Contin.	Multi
Medical Diagnosis	Partially	Stoch.	Sequ.	Dynam.	Contin.	Single
Image analysis	Fully	Det.	Sequ.	Semi	Contin.	Single
Part-picking robot	Partially	Stoch.	Sequ.	Dynam.	Contin.	Single
Refinery controller	Partially	Stoch.	Sequ.	Dynam.	Contin.	Single
English tutor	Partially	Stoch.	Sequ.	Dynam.	Discrete	Multi

2.4 The Structure of Agents

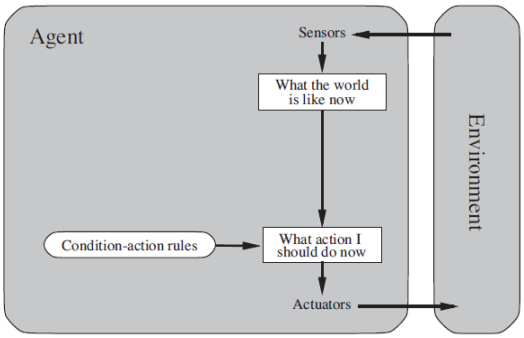
- Agent architecture: computing device with physical sensors and actuators
- Agent = architecture + program

- Four basic types in order of increasing generality:
  - Simple reflex agents
  - Model-based reflex agents
  - Goal-based agents
  - Utility-based agents

Simple Reflex Agents

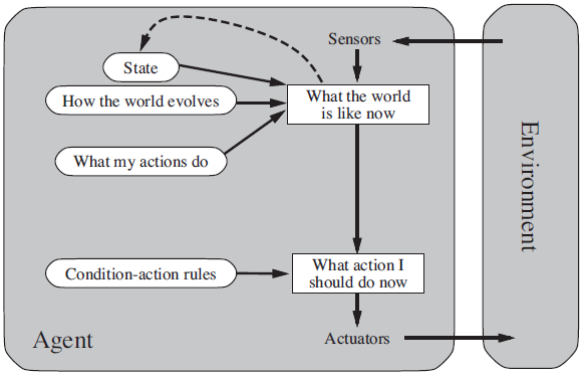
function SIMPLE-REFLEX-AGENT(*percept*) **returns** an action
  persistent: *rules*, a set of condition–action rules

  state ← INTERPRET-INPUT(*percept*)
  rule ← RULE-MATCH(*state*, *rules*)
  action ← *rule*.ACTION
  return *action*



Model-based reflex agents

- It keeps track of the current state of the world, using an internal model. It then chooses an action in the same way as the reflex agent.



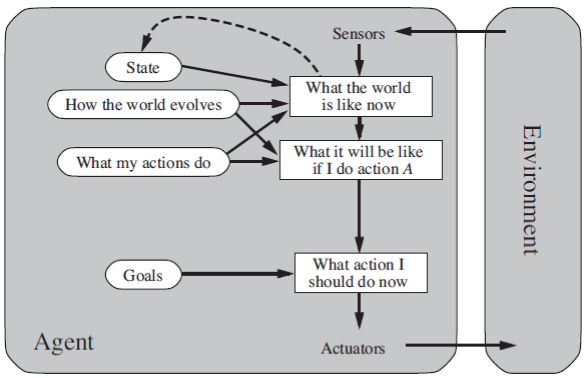
Model-based reflex agents

function MODEL-BASED-REFLEX-AGENT(*percept*) **returns** an action
  persistent: *state*, the agent's current conception of the world state
    model, a description of how the next state depends on current state and action
    rules, a set of condition–action rules
    action, the most recent action, initially none

state ← UPDATE-STATE(*state*, *action*, *percept*, *model*)
rule ← RULE-MATCH(*state*, *rules*)
action ← *rule*.ACTION
return *action*

Model-based, goal-based agent

- A model-based, goal-based agent. It keeps track of the world state and of a set of goals, and chooses an action that will (eventually) lead to achievement of its goals.

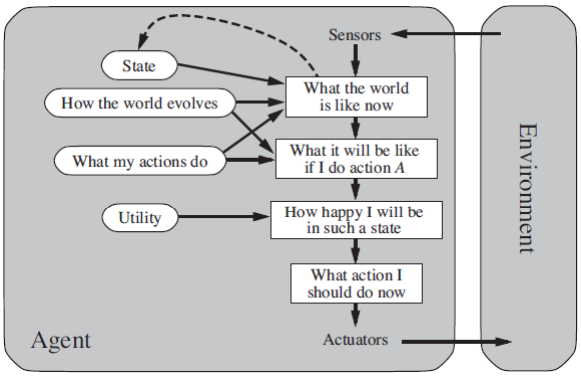


Utility-based Agents

- Goals alone are not enough to achieve high-quality behaviour in most environments
- Some action sequences are better (cheaper, quicker, more reliable, …) than others.
- A utility function maps a state (or a sequence of states) onto a real number, which describes the associated degree of happiness of the agent

Utility-based Agents

- It uses a model of the world, along with a utility function. Then it chooses the action that leads to the best expected utility,
- Expected utility is computed by averaging over all possible outcome states, weighted by the probability of the outcome.

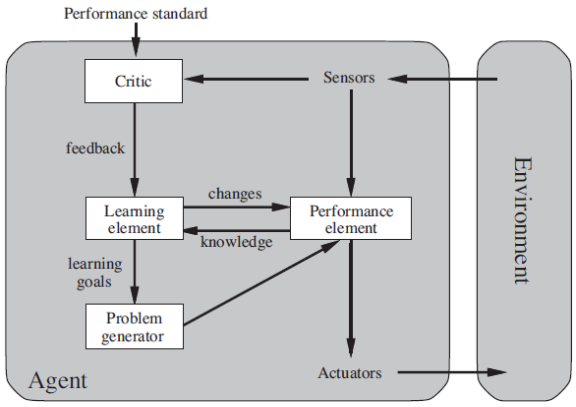


Learning Agents

- Programming intelligent machines by hand is extremely time consuming
- Better method: program a learning algorithm and then teach machines what they should learn
  - Example: Supervised learning of artificial neural networks (ANNs) or support vector machines (SVMs)
- Or better still, let these machines learn by themselves (under some constraints)
  - Reinforcement learning of artificial neural networks
- Learning is nowadays **the** big issue in AI!

Learning Agents

- A general learning agent, consisting of 4 components:
  - Learning element
  - Performance element
  - Critic
  - Problem generator



Learning Agents

- Learning element: is responsible for making improvements, uses feedback from the critic on how the agent is doing and determines how the performance element should be modified
- Performance element: is responsible for selecting external actions
- Critic: tells the learning element how well the agent is doing with respect to a performance standard (the percept itself does not say so).
- Problem generator: is responsible for suggesting actions that lead to new information