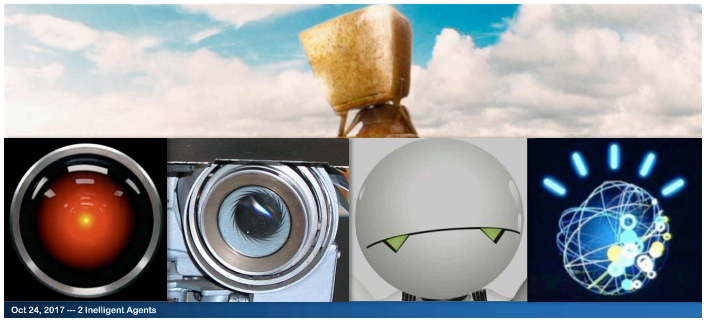
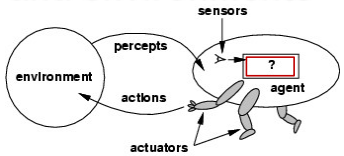
Artificial Intelligence

Prof. Björn Ommer
HCI & IWR



Agents and environments



- The agent function maps from percept histories to actions:

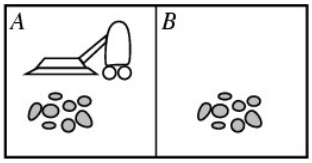
$$f : \mathcal{P}^* \rightarrow \mathcal{A}$$

- The agent program runs on the physical architecture to produce f
- agent = architecture + program

Outline – Intelligent Agents

- Agents and environments
- Rationality
- PEAS (Performance measure, Environment, Actuators, Sensors)
- Environment types
- Agent types

Vacuum-cleaner world



- Percepts: [location, contents], e.g., [A,Dirty]
- Actions: *Left*, *Right*, *Suck*, *NoOp*

Agents

- An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators
- Human agent
  - eyes, ears, and other organs for sensors; hands, legs, mouth, and other body parts for actuators
- Robotic agent
  - cameras and infrared range finders for sensors; various motors for actuators

A vacuum-cleaner agent

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
...	...

```
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
```

- What is the right function?
- Can it be implemented in a small agent program?

Rational agents

- An agent should strive to *"do the right thing"*, based on what it can perceive and the actions it can perform. The right action is the one that will cause the agent to be most successful
- Performance measure:
  - An objective criterion for success of an agent's behavior
  - E.g., performance measure of a vacuum-cleaner agent
    - amount of dirt cleaned up,
    - amount of time taken,
    - amount of electricity consumed,
    - amount of noise generated, etc.

PEAS

- Designing an intelligent agent: must first specify the task environment
- PEAS:
  - Performance measure,
  - Environment,
  - Actuators,
  - Sensors

Rational agents

- Rational Agent: For each possible percept sequence, select 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.
- Rational ≠ omniscient
  - percepts may not supply all relevant information
- Rational ≠ clairvoyant
  - action outcomes may not be as expected
- Hence, rational ≠ successful
- Rational ⇨ exploration, learning, autonomy

PEAS - Example

- Consider, e.g., the task of designing an 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

Rational agents

- Agents can perform actions in order to modify future percepts so as to obtain useful information (information gathering, exploration)
- An agent is autonomous if its behavior is determined by its own experience (with ability to learn and adapt) and not by direct commands from a user etc.

vs.

PEAS - 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)

PEAS - Example

- Agent: Part-picking robot
  - Performance measure:
    - Percentage of parts in correct bins
  - Environment:
    - Conveyor belt with parts, bins
  - Actuators:
    - Jointed arm and hand
  - Sensors:
    - Camera, joint angle sensors

Environment types

	Chess with a clock	Chess w/o a clock	Taxi driving
Fully observable	Yes	Yes	No
Deterministic	Strategic	Strategic	No
Episodic	No	No	No
Static	Semi	Yes	No
Discrete	Yes	Yes	No
Single agent	No	No	No

- The environment type largely determines the agent design
- The real world is (of course) partially observable, stochastic, sequential, dynamic, continuous, multi-agent

Environment types

- 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.

Agent functions and programs

- An agent is completely specified by the agent function mapping percept sequences to actions
- One agent function (or a small equivalence class) is rational
- Aim: find a way to implement the rational agent function concisely

Environment types

- 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.

Table-lookup agent

- Predefined set of action rules for each given percept sequence
- Drawbacks:
  - Huge table
  - Takes a long time to build the table
  - No autonomy
  - Even with learning, need a long time to learn the table entries and too little training data to cover all eventualities

Outlook: Agent types

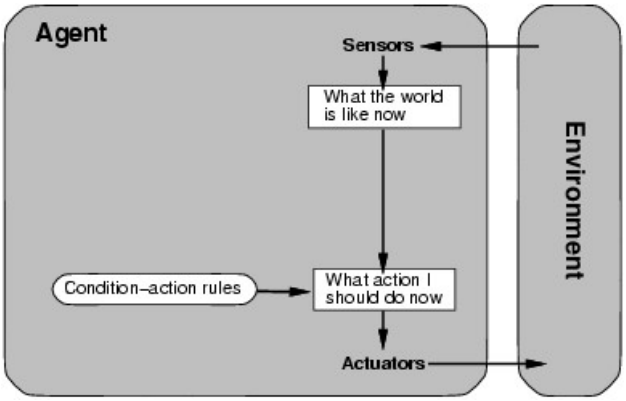
Four basic types in order of increasing generality:

- **Simple reflex agents**
- **Model-based reflex agents (reflex agent with state)**
- **Goal-based agents**
- **Utility-based agents**

Simple reflex agents

```
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
```


Simple reflex agents



Reflex Agent w/o Model of the State of the World

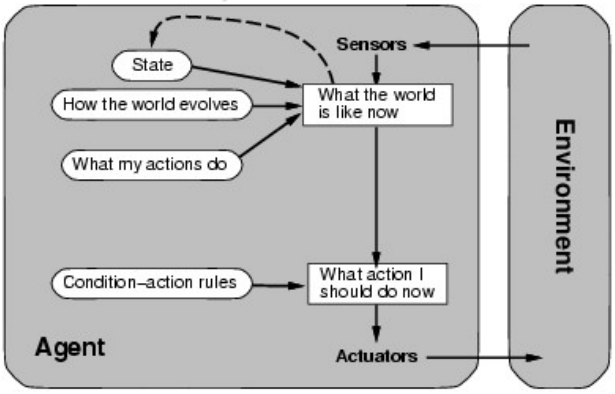
Simple reflex agents

- **find a rule whose condition matches the current situation (as defined by the percept) and then doing the action associated with that rule.**

```
function SIMPLE-REFLEX-AGENT(percept) returns action
  static: rules, a set of condition-action rules

  state ← INTERPRET-INPUT(percept)
  rule ← RULE-MATCH(state, rules)
  action ← RULE-ACTION[rule]
  return action
```


Model-based reflex agents (with internal state)

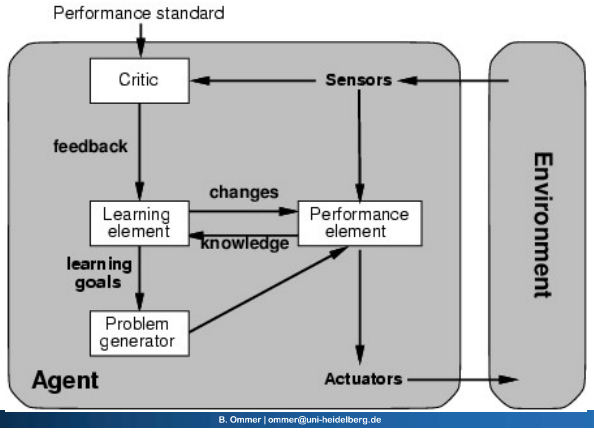


- **Find a rule whose condition matches the current situation (as defined by the percept and the stored internal state) and then doing the action associated with that rule.**

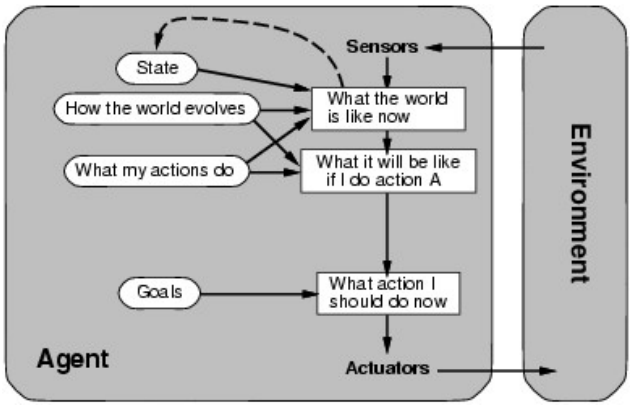
```
function REFLEX-AGENT-WITH-STATE(percept) returns action
  static: state, a description of the current world state
          rules, a set of condition-action rules

  state ← UPDATE-STATE(state, percept)
  rule ← RULE-MATCH(state, rules)
  action ← RULE-ACTION[rule]
  state ← UPDATE-STATE(state, action)
  return action
```


Learning agents

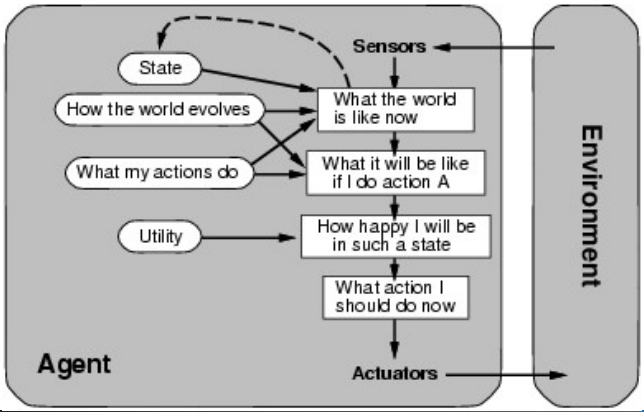


Goal-based agents



- **More details & examples of these agents in later lectures**

Utility-based agents



Summary

- Agents interact with environments through actuators and sensors
- The agent function describes what the agent does in all circumstances
- The performance measure evaluates the environment sequence
- A perfectly rational agent maximizes expected performance
- Agent programs implement (some) agent functions

Summary contd.

- PEAS descriptions define task environments
- Environments are categorized along several dimensions:
  - observable? deterministic? episodic? static? discrete? single-agent?
- Several basic agent architectures exist:
  - reflex, reflex with state, goal-based, utility-based