Polytech Montpellier – IG4

Artificial Intelligence & Multi-Agent Systems

Artificial Agents

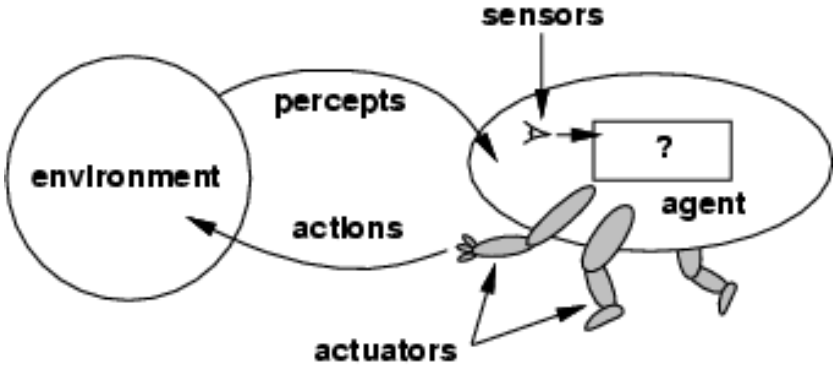
Agent – definition?

- Difficult to define, yet:
  - 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

Autonomy

- One important characteristic of agents is autonomy
- Rational agent is autonomous: takes its decisions (e.g., decide actions) according to the current situation, and changes their "plans" accordingly.

Agents and environments

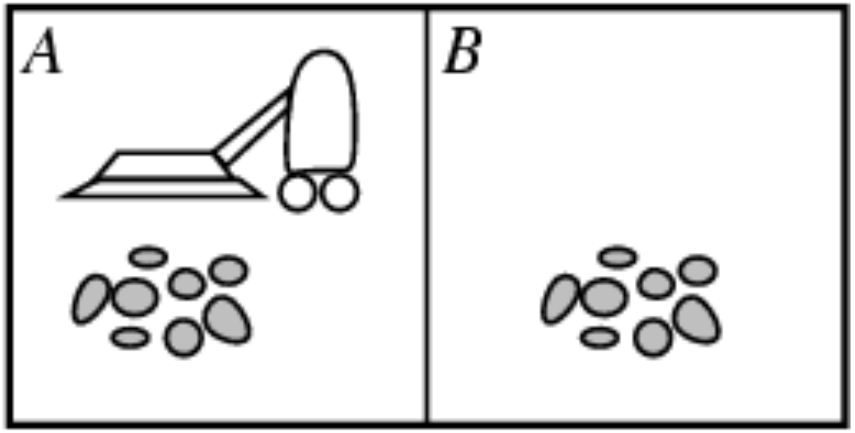


- The **agent function** maps from percepts to actions:

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

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

Vacuum-cleaner world



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

Rationality depends on PEAS

P: The performance measure (success or failure)

E: The agent’s prior knowledge about the environment

A: The actions that the agent can perform

S: The percept sequence

Rational Agent

- 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
- Performance measure:
  - An objective criterion for success of an agent’s behavior

Rational agent

- Rational ≠ omniscient
  – percepts may not supply all relevant information
- Rational ≠ clairvoyant
  – action outcomes may not be as expected
- Rational ⇒ exploration, learning, autonomy
- A rational agent chooses whichever action maximizes the expected value of the performance measure given the percept sequence to date
- Selected performance measure evaluates the environment sequence

Design objectives and rationality

- A rational agent performs actions in line with the design objectives.
- These objectives shape the performance measure.
- For example, in case of the vacuum cleaner, the designer could focus only on collecting as much dirt as possible in time T.

A vacuum-cleaner agent

- What is the right function?

Percept sequence	Action
[A, Clean]	Right
[A, Dirty]	Suck
[B, Clean]	Left
[B, Dirty]	Suck
[A, Clean], [A, Clean] [A, Clean],	Right
[A, Dirty]	Suck
…	…

- Can it be implemented in a small agent program?

```
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
```

PEAS

- To design a rational agent, we must specify the PEAS
- Consider, e.g., designing an automated taxi:
  - Performance measure?
  - Environment?
  - Actuators?
  - Sensors?

Automated taxi agent

- Performance measure:
  - safety, destination, profits, legality, comfort
- Environment:
  - streets/highways, traffic, pedestrians, weather
- Actuators:
  - steering, accelerator, brake, horn, speaker/display
- Sensors:
  - video, accelerometers, gauges, engine sensors, GPS

Internet shopping agent

- Performance measure:
  - price, quality, appropriateness, efficiency
- Environment:
  - current and future WWW sites, vendors, shippers
- Actuators:
  - display to user, follow URL, fill in form
- Sensors:
  - HTML pages (text, graphics, scripts)

Medical diagnosis system agent

- 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)

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

Interactive English tutor

- Performance measure:
  - Maximize student's score on test
- Environment:
  - Set of students
- Actuators:
  - Screen display (exercises, suggestions, corrections)
- Sensors:
  - Keyboard

Properties of environments

- Fully observable environments
  - The agent’s sensors give it access to the complete state of the environment at each point in time. The agent can obtain complete, accurate, up-to-date information about all aspects of the environment that are relevant to the choice of action.
  - The agent need not maintain any internal state to keep track of the world
  - The more accessible an environment is, the simpler it is to build agents to operate in it
- Partially observable environments
  - Because of noisy and inaccurate sensors, or because parts of the state are simply missing from the sensor data
  - Agent must make informed guesses about world

Properties of environments

- Deterministic environment
  - The next state depends only on current state and agent's action
  - Any action has a single guaranteed effect. There is no uncertainty about the state that will result from performing an action
  - If the environment is deterministic except for the actions of other agents, we say that the environment is strategic
- Stochastic environment
  - There is some uncertainty about the outcome of an action
  - Non-deterministic environments - possible outcomes only, not probabilities - present greater problems for agent design

Properties of environments

- Episodic environments
  - The agent’s experience is divided into atomic episodes. Each episode consists of the agent perceiving and then performing a single action
  - The episodes are independent. The choice of action in each episode depends only on the episode itself
- Sequential environments
  - The current decision could affect all future decisions
  - Episodic environments are much simpler than sequential because the agent does not need to think ahead

Properties of environments

- Discrete
  - Finite number of distinct states and percepts/actions, e.g. chess
- Continuous
  - Continuous time/state/actions: taxi driver

Properties of environments

- Static environment
  - Can be assumed to remain unchanged except by the performance of actions by the agent
  - the agent doesn’t need to keep looking at the world while it is deciding on an action, nor need it worry about the passage of time.
- Dynamic environment
  - Can change while an agent is deliberating
  - Has other processes operating on it, and changes in ways beyond the agent’s control
  - Demand quick decisions from the agent
  - Semidynamic when the world does not change but the agent’s performance score does - chess with clock

Properties of environments

- Known environment
  - The agent’s knowledge about how the environment works evolves
  - Note that a known environment (i.e., the agent knows all the rules that apply) may be only partially observable if the sensors are not properly working
- Unknown environment
  - The agent will have to learn how it works
  - A known environment can be partially observable.

Properties of environments

- Single agent vs multiagent
  - Which entities will be viewed as other agents?
  - Competitive and cooperative interactions
- The environment type largely determines the agent design

- The real world is (of course) partially observable, stochastic, sequential, dynamic, continuous, multi-agent

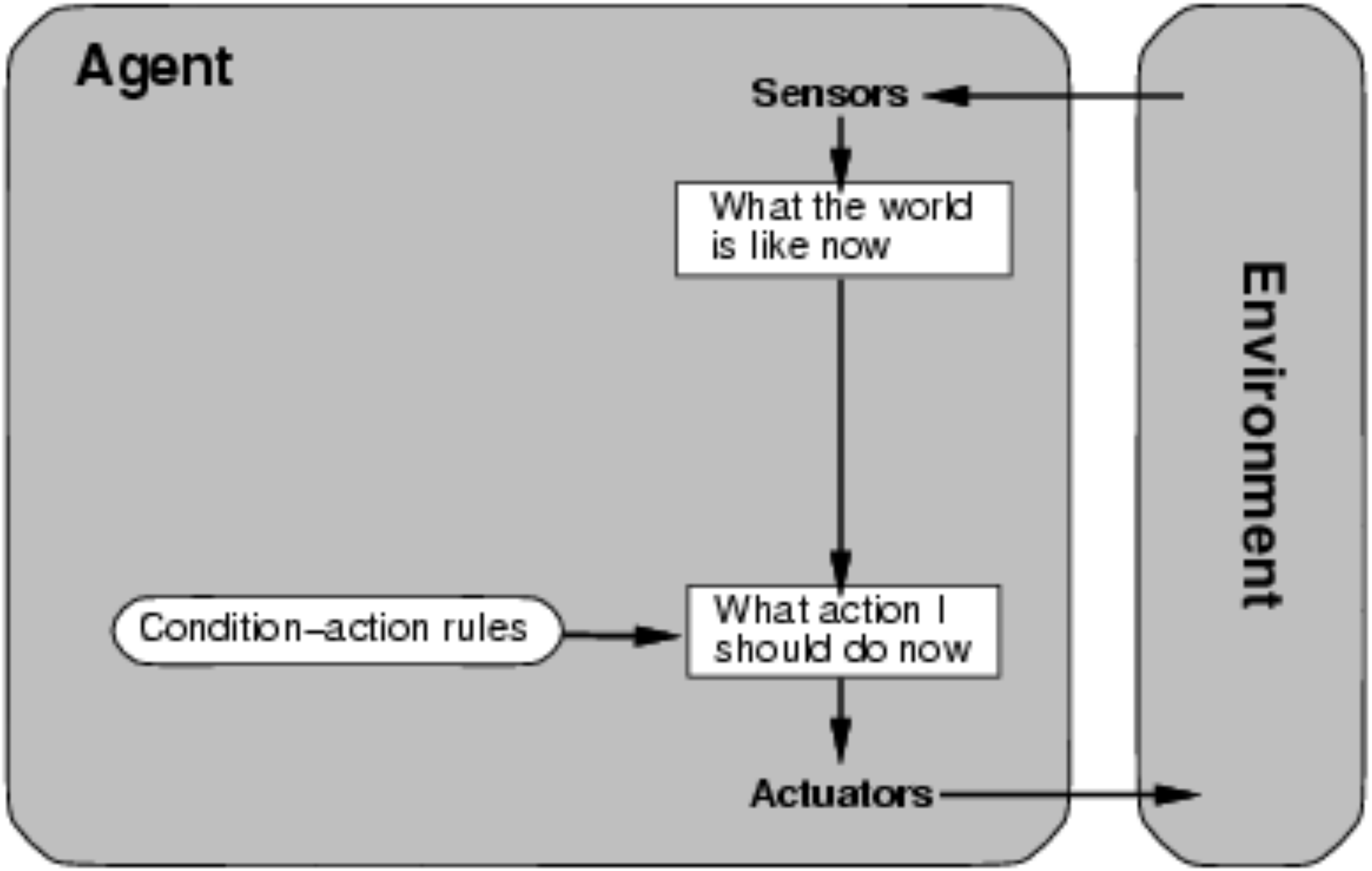
Examples on environment types

Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Crossword	fully	deterministic	sequential	static	discrete	single
Chess w/ clock	fully	deterministic	sequential	static	discrete	multi
Backgammon	fully	stochastic	sequential	static	discrete	multi
Taxi driving	partially	stochastic	sequential	dynamic	continuous	multi
Medical diagnosis	partially	stochastic	sequential	dynamic	continuous	single

Agent Types

- A typology … among others
- Four basic types in order of increasing generality:
  - Simple reflex agents
  - Model-based reflex agents
  - Goal-based agents
  - Utility-based agents

Simple reflex agent



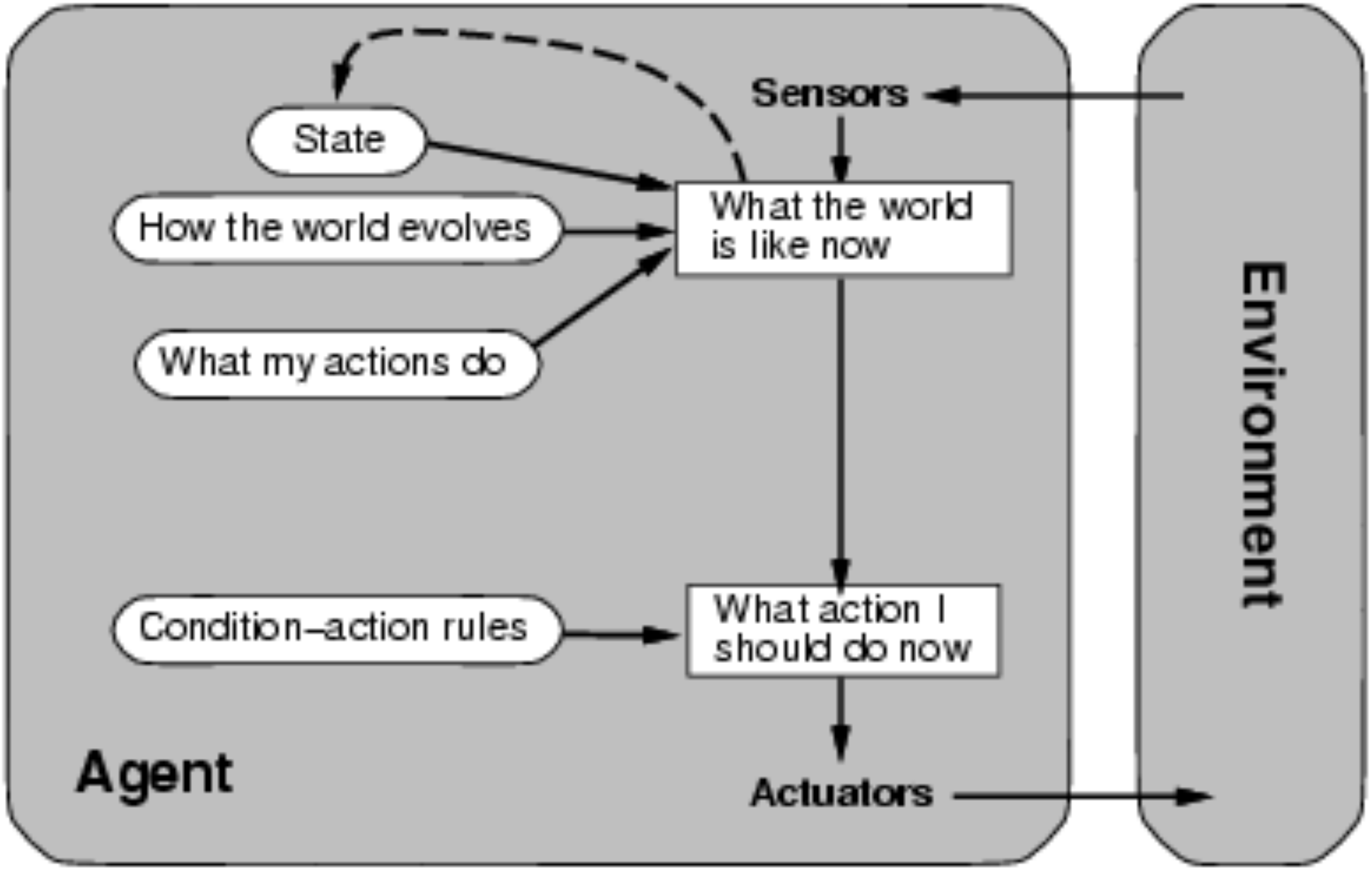
Simple Reflex Agent

```
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
```

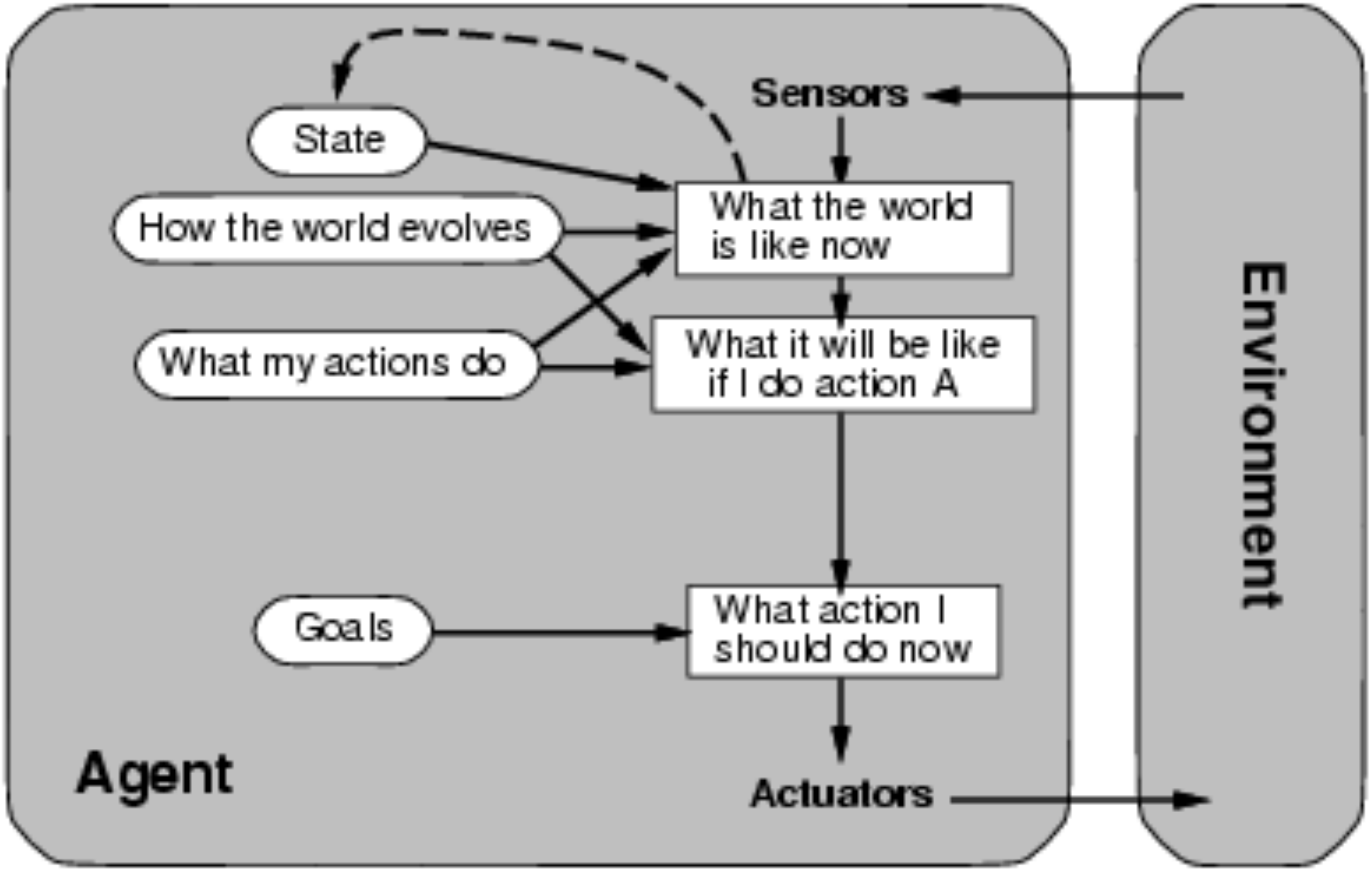
```
(setq joe (make-agent :name ’joe :body (make-agent-body)
                      :program (make-reflex-vacuum-agent-program))

(defun make-reflex-vacuum-agent-program ()
   #’(lambda (percept)
        (let ((location (first percept)) (status (second percept))
          (cond ((eq status ’dirty) ’Suck)
                ((eq location ’A) ’Right)
                ((eq location ’B) ’Left)))))
```

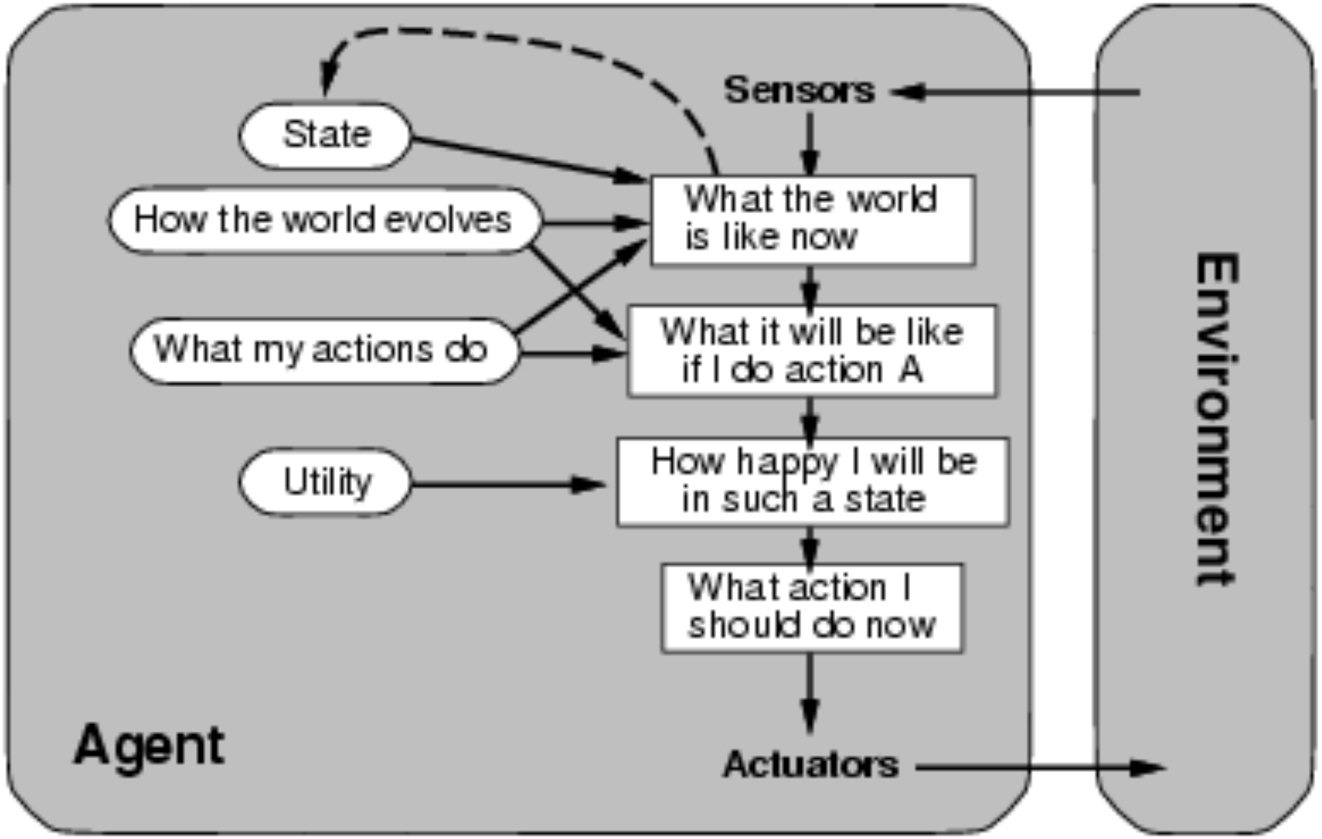
Model-based reflex agents



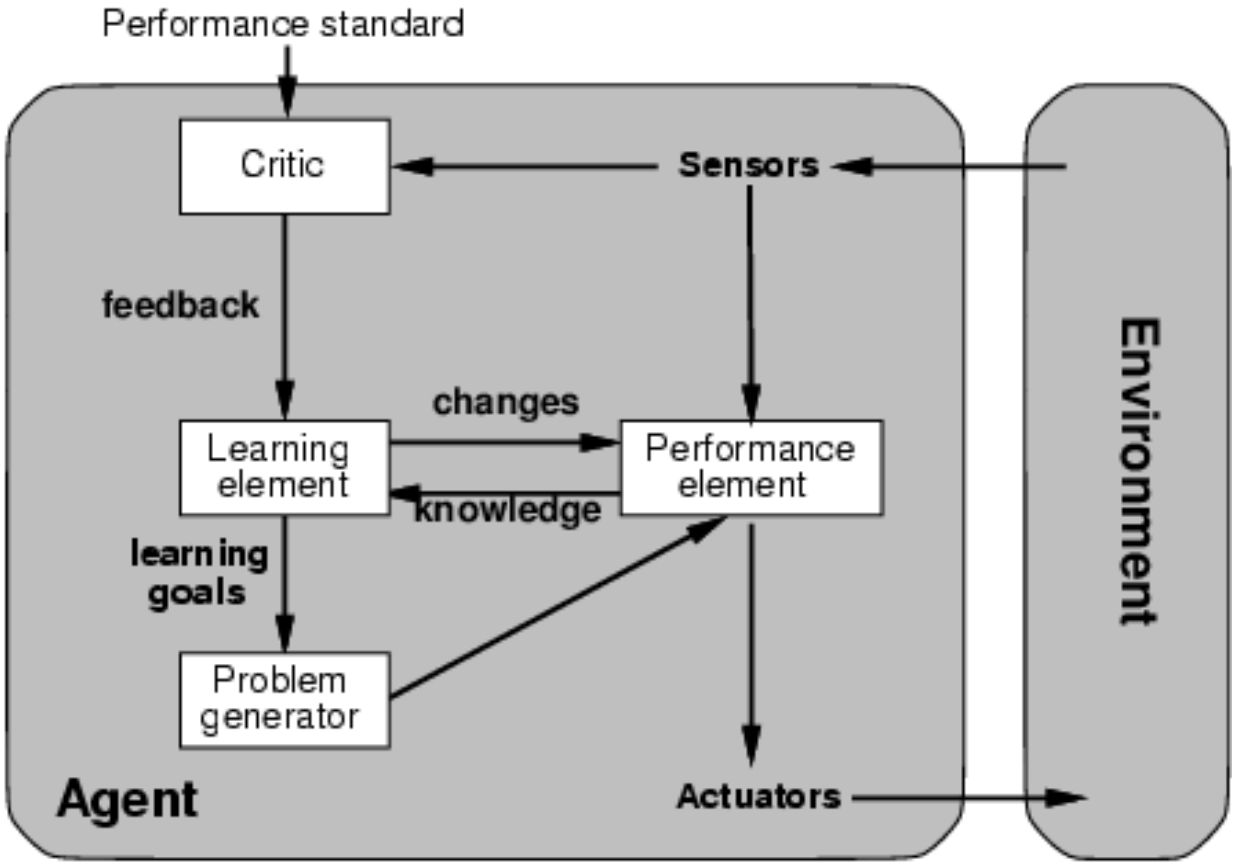
Goal-based agents


Sensors
State
How the world evolves
What my actions do
What the world is like now
What it will be like if I do action A
Goals
What action I should do now
Agent
Actuators
Environment

Utility-based agents


Sensors
State
How the world evolves
What my actions do
Utility
What the world is like now
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
Agent
Actuators
Environment

Learning agents


Performance standard
Critic
Sensors
feedback
changes
Learning element
Performance element
knowledge
learning goals
Problem generator
Agent
Actuators
Environment