

Lezione

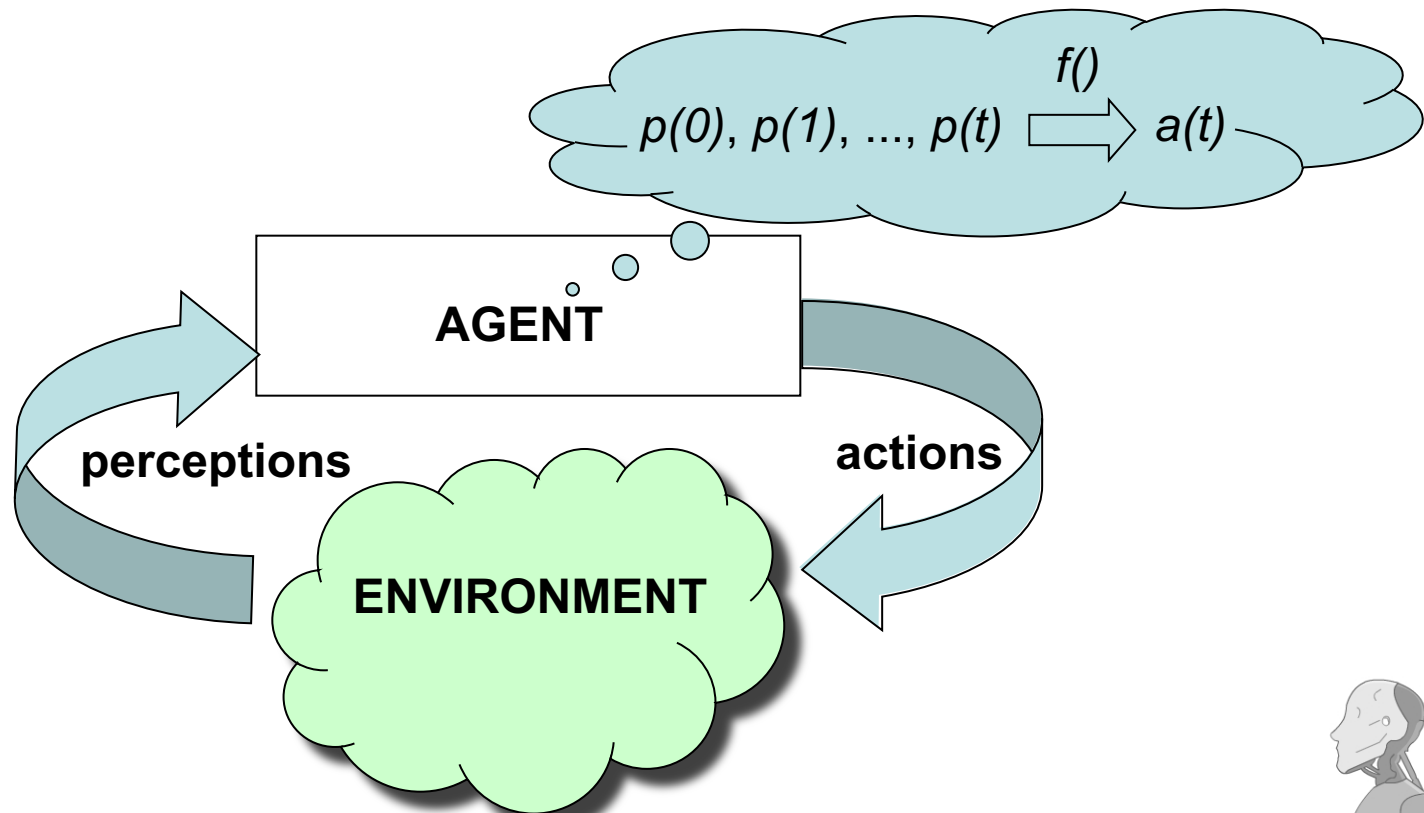
1



Agent architectures

Designing intelligent agents

- An agent is defined by its *agent function* $f()$ that maps a sequence of perceptions to an action



Rational agent

- It would be useful to design agent functions that make agents rational, namely that make them do the “right thing”
- How can rationality of an agent be defined?

Rational agent

For every possible sequence of perceptions, a rational agent chooses the action that maximizes the expected value of its performance measure, given its knowledge up to that moment



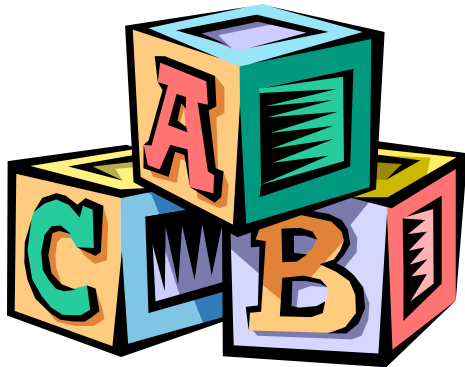
Perception sequence and available knowledge

- The perception sequence represents the available knowledge of an agent about its environment
- From the standpoint of an agent, its environment can be:
 - Completely/partially observable
 - Static/dynamic
 - Discrete/continuous
 - Single agent/multiagent



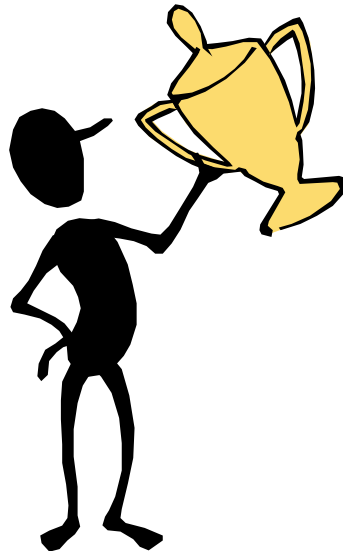
Actions

- Through actions an agent can change the state of its environment
- From the standpoint of an agent, its environment can be:
 - Deterministic/stochastic



Performance measure

- The performance measure is the criterion for evaluating the success of the behavior of an agent
- The performance measure is defined by the designer



What a rational agent does?

Rational agent

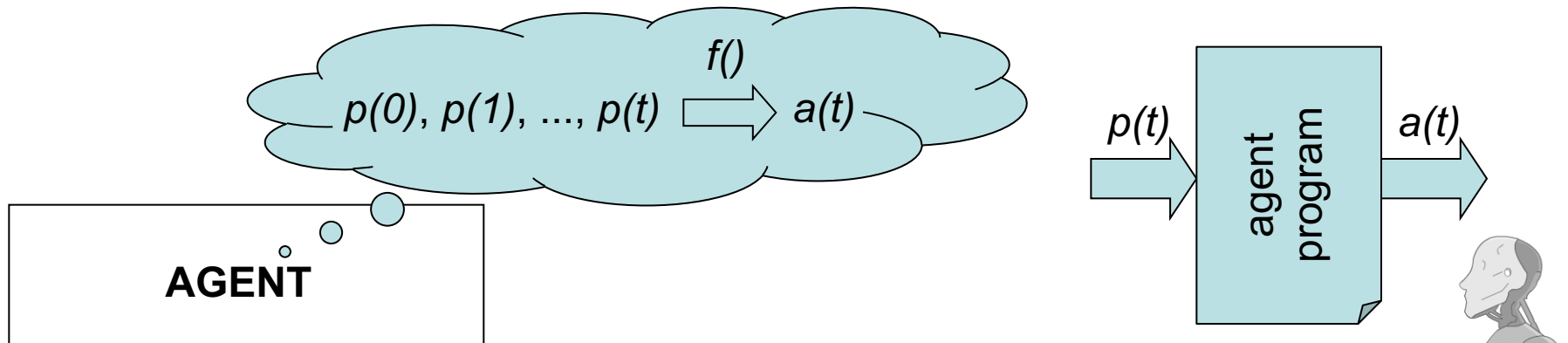
For every possible sequence of perceptions, a rational agent chooses the action that maximizes the **expected** value of its performance measure, given its knowledge **up to that moment**

- A rational agent is not omniscient
- A rational agent is not clairvoyant
- A rational agent can explore to acquire new information, can learn, ...



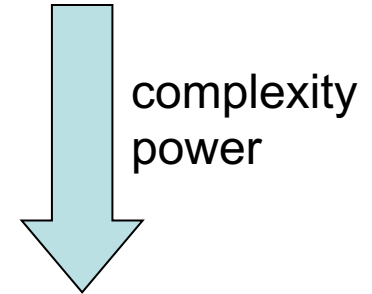
From agent functions to agent programs

- A designer develops an *agent program* that implements an agent function
- An agent program has the current perception $p(t)$ as input
 - An agent program can store the previous perceptions $p(0), p(1), \dots, p(t-1)$
- An agent program has an action $a(t)$ as output

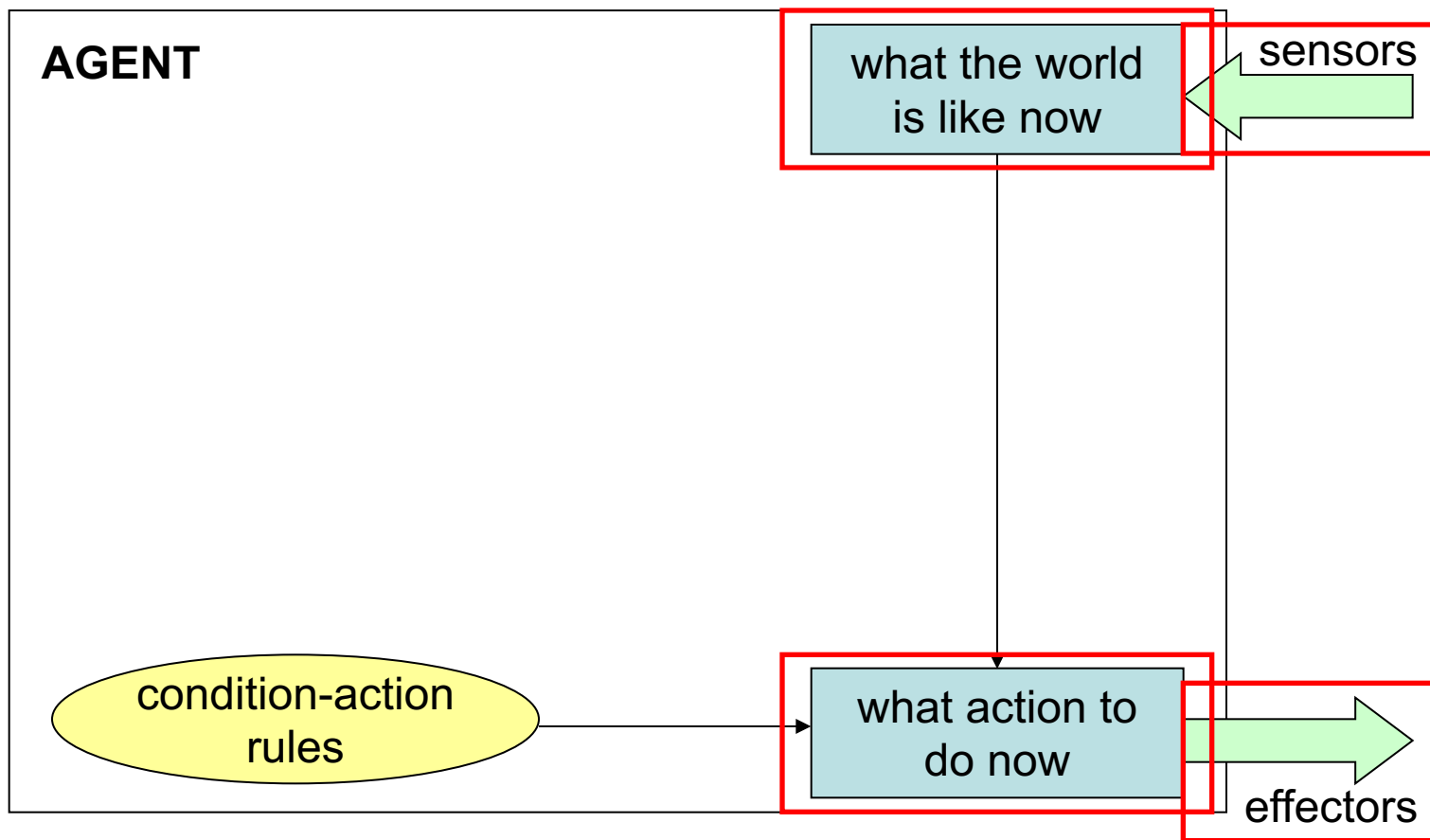


Structure of agent programs

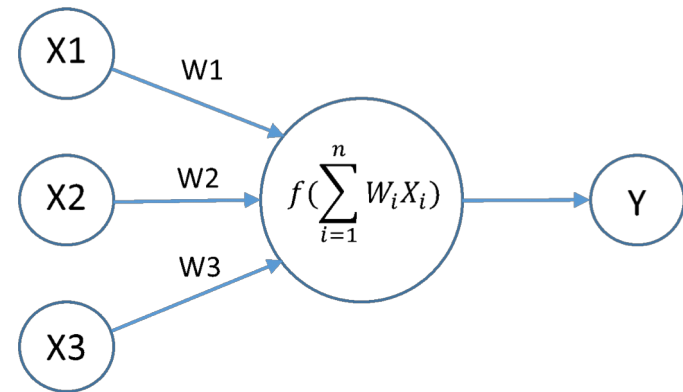
- Agent programs can be classified in four basic types:
 - Simple reflex agents
 - Reflex agents with state
 - Goal-based agents
 - Utility-based agents
- All these types of agent can also learn



Simple reflex agents



Examples of simple reflex agents

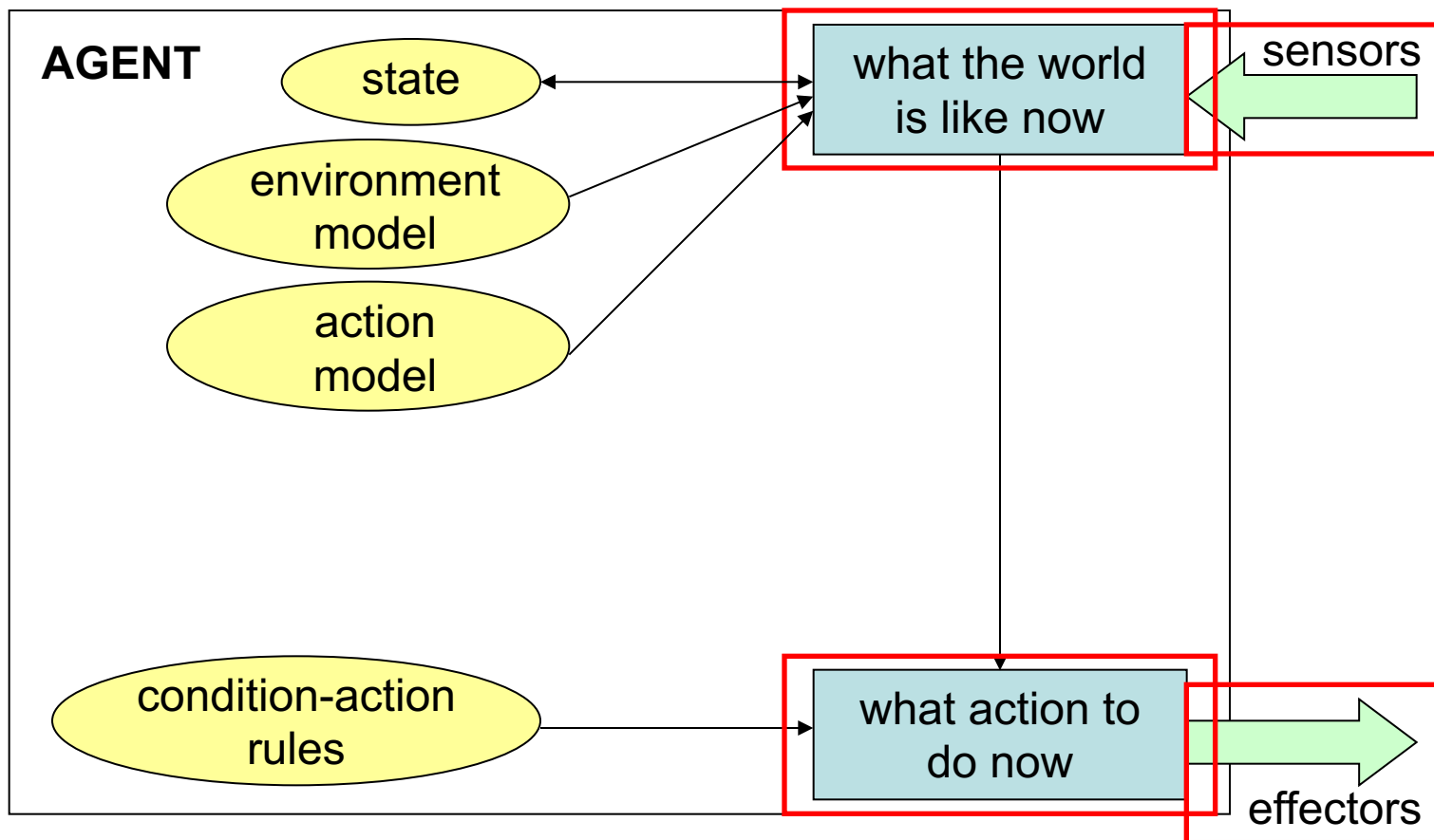


Neural networks

First versions of Roomba



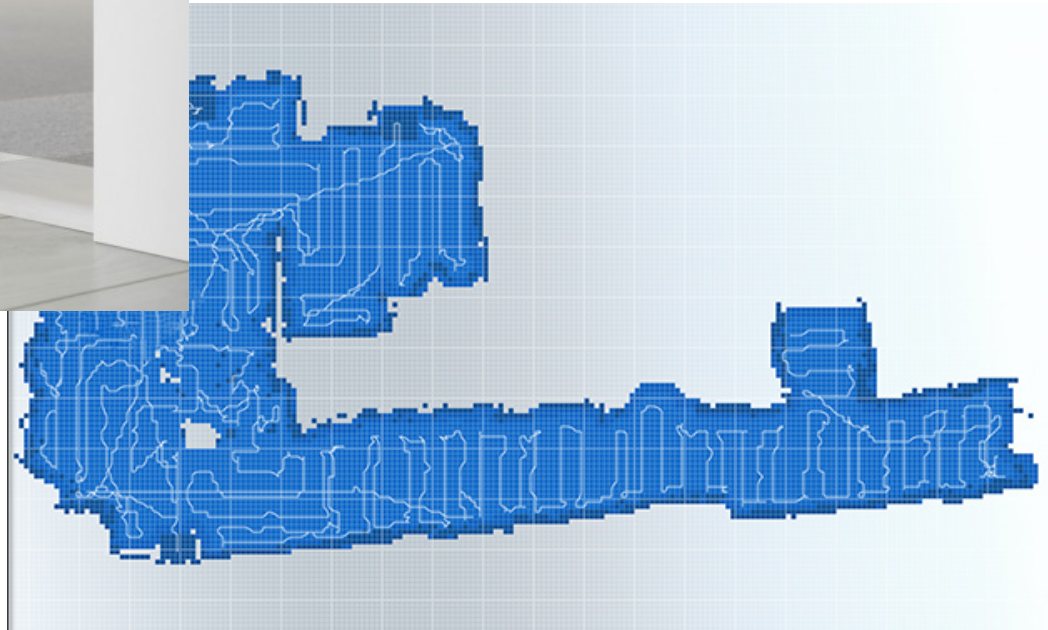
Reflex agents with state



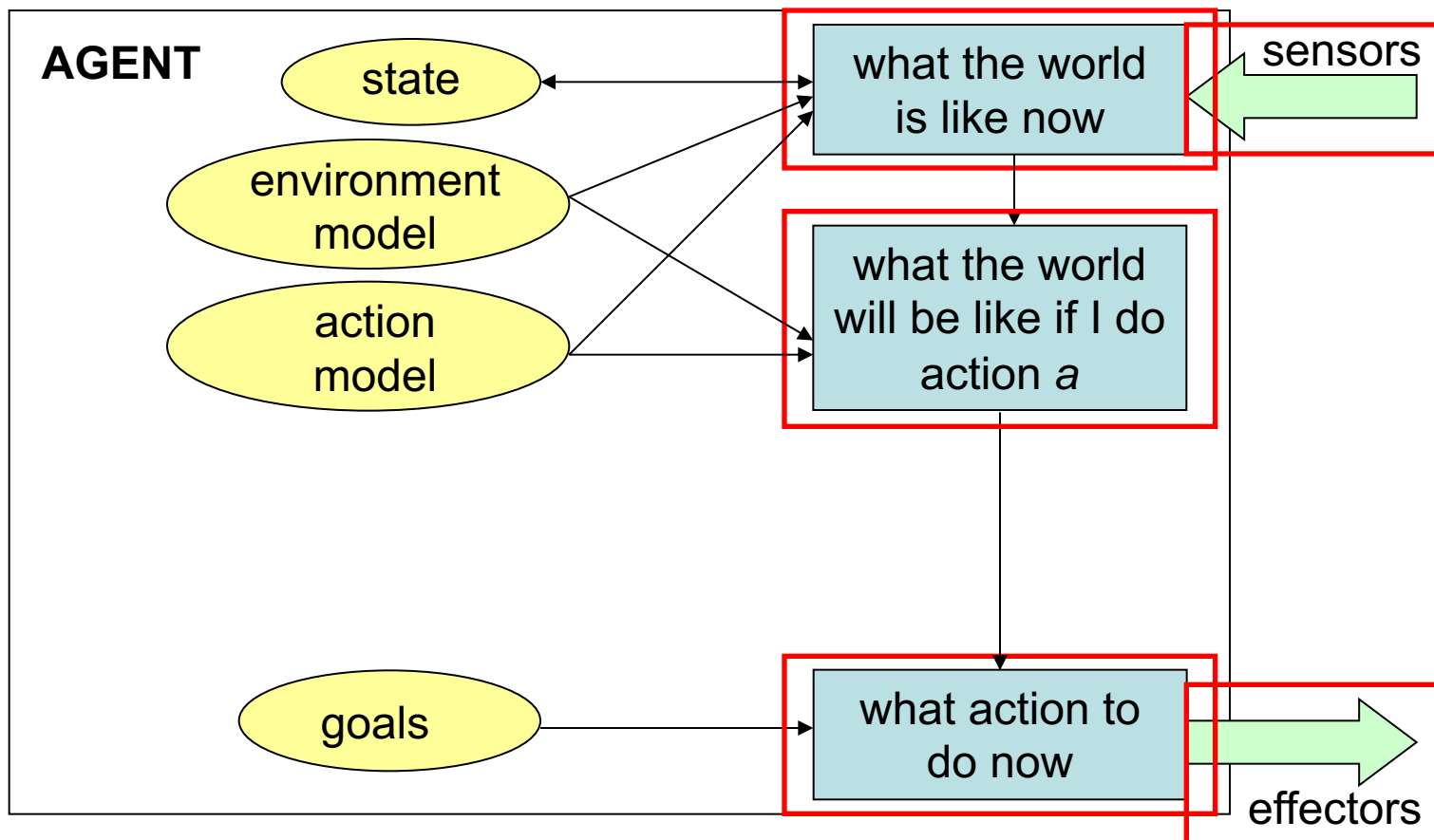
Examples of reflex agents with state



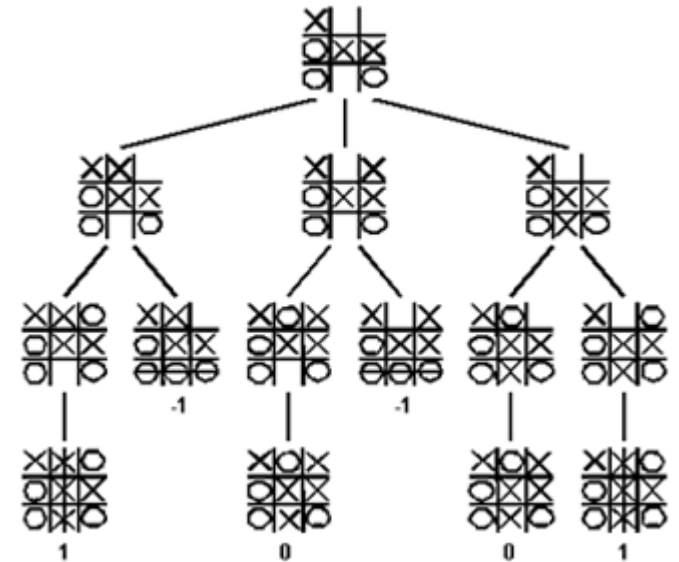
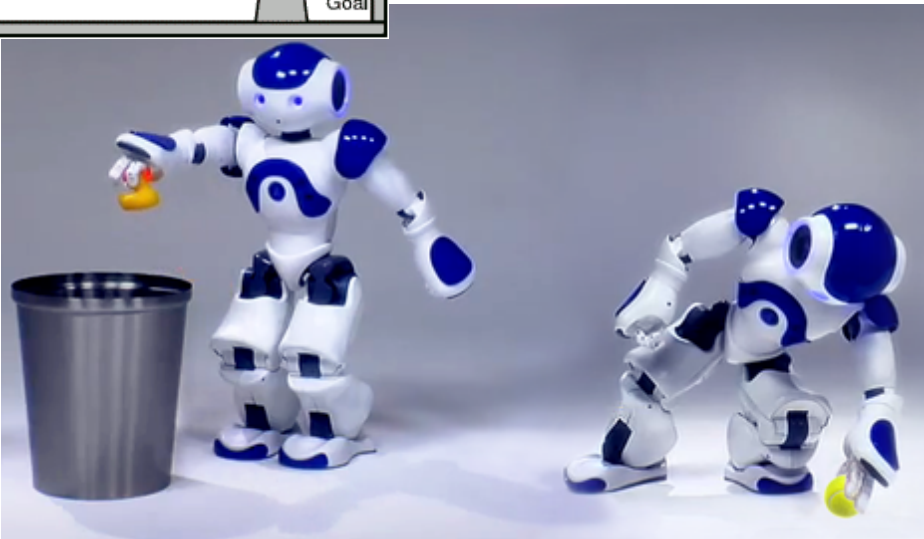
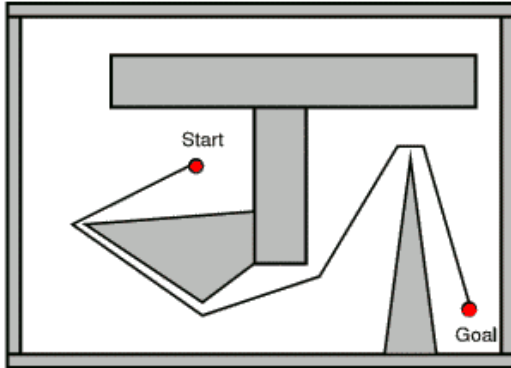
Dyson 360 Eye



Goal-based agents



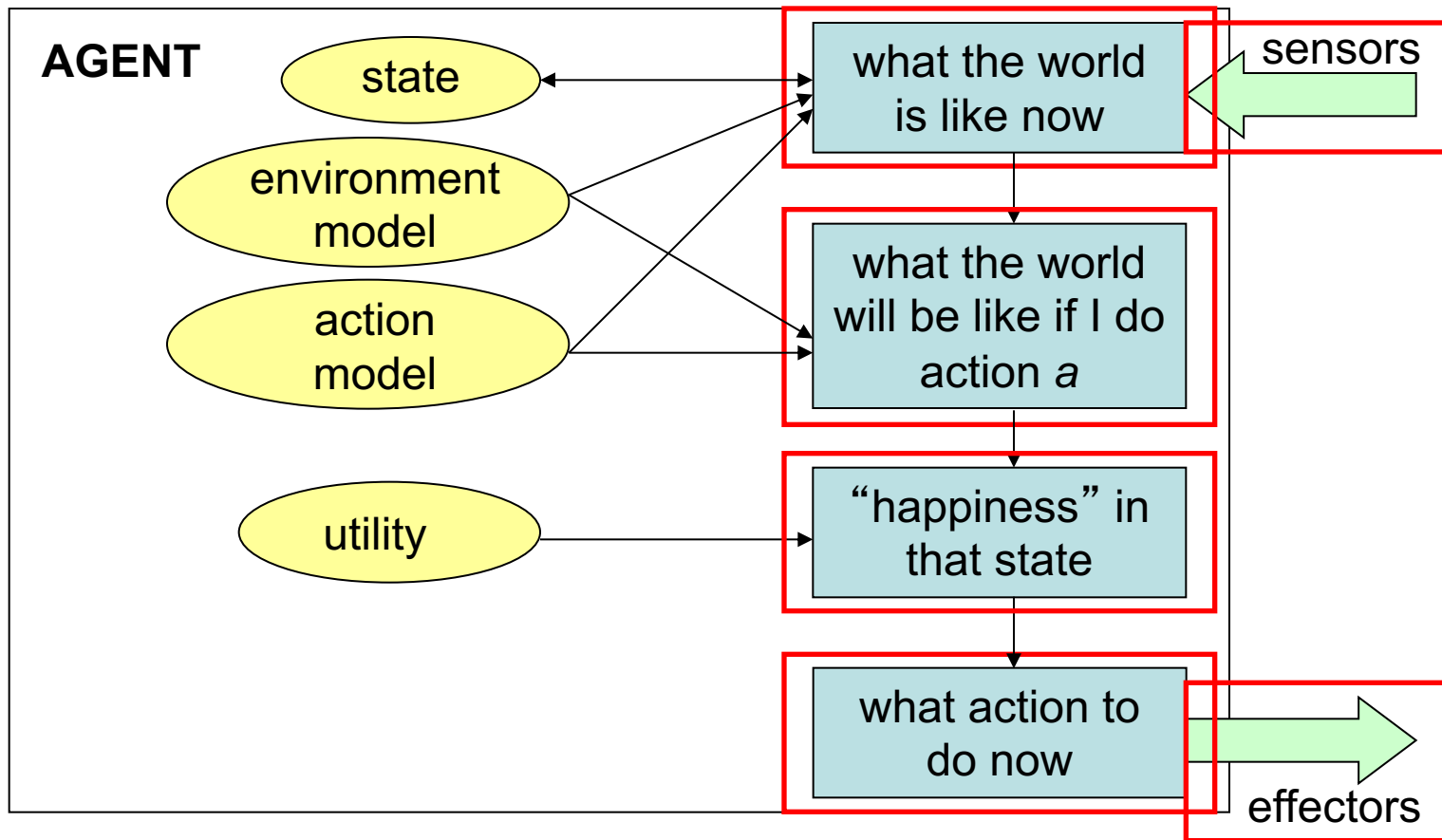
Examples of goal-based agents



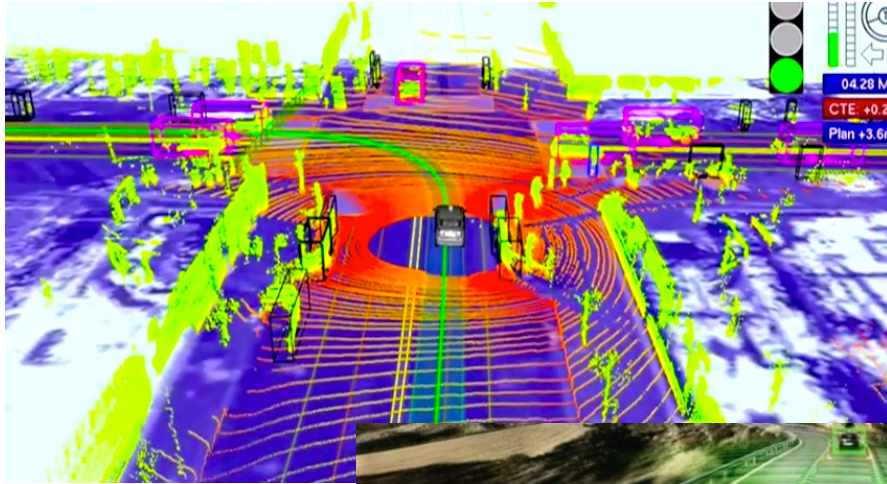
Action (movement) planning



Utility-based agents



Examples of utility-based agents



Decision on the best action (movement)



Learning agents

- All the agents presented before can improve their performance with learning
- Every component of the decisional process of an agent can be modified in order to perform better



Examples of learning agents

