COMP3411/9414: Artificial Intelligence

Module 1

Intelligent Agents – Agent Types

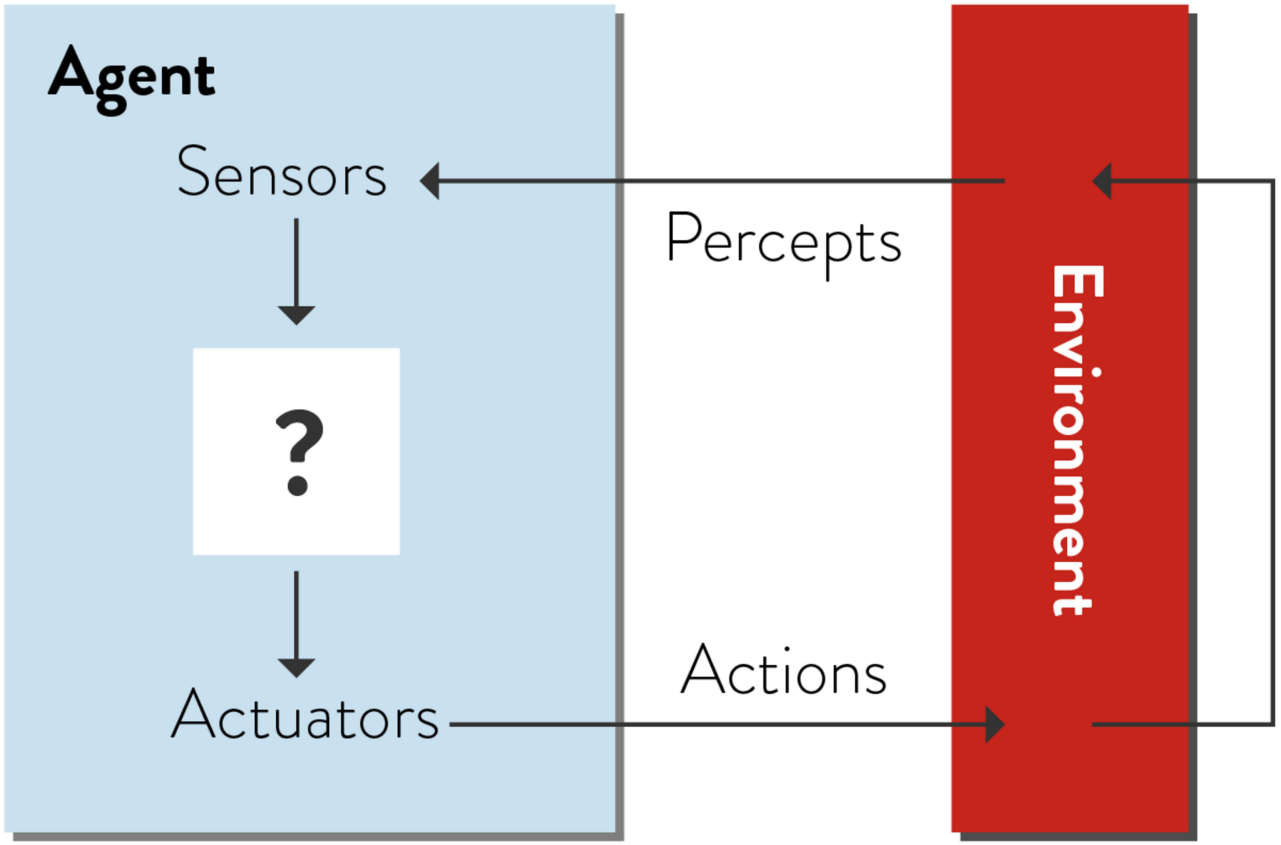
Russell & Norvig, Chapter 2.

Agent Types

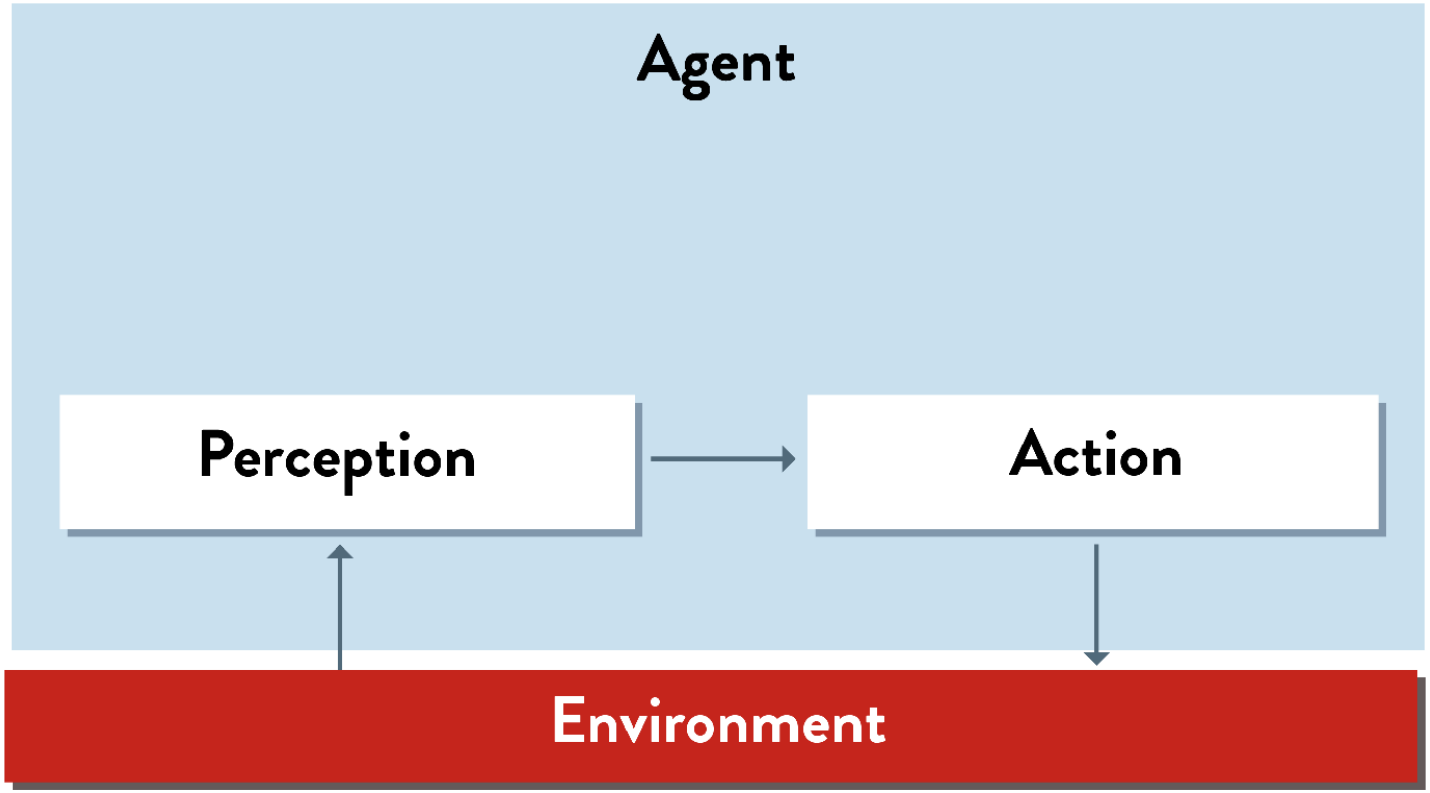
In this course we will consider different types of agent:

- Reactive Agent
- Model-Based Agent
- Planning Agent
- Utility-based agent
- Game Playing Agent
- Learning Agent

Agent Model



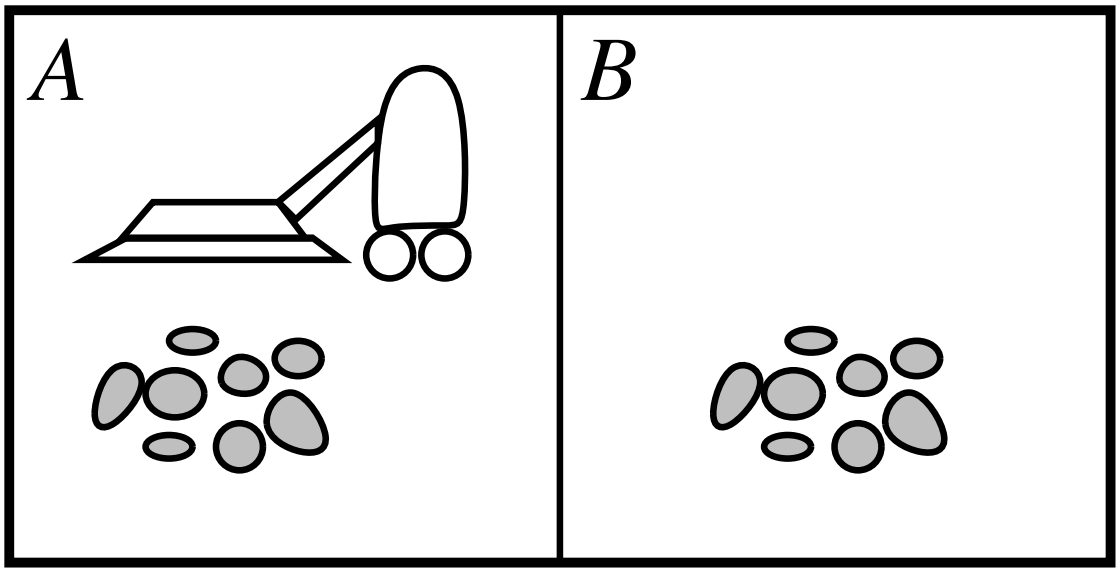
Reactive Agent



Reactive Agent

- Choose the next action based only on what they currently perceive, using a "policy" or set of rules which are simple to apply
- Sometimes called "simple reflex agents" – but they can do surprisingly sophisticated things!
  - Swiss robots (https://www.youtube.com/watch?v=B0wM-eKSxhk)
  - Pool balancing
  - Simulated hockey

Vacuum-cleaner world



Percepts: location and contents, e.g., [*A, Dirty*]

Actions: *Left, Right, Suck, NoOp*

A vacuum-cleaner -simple reflex agent

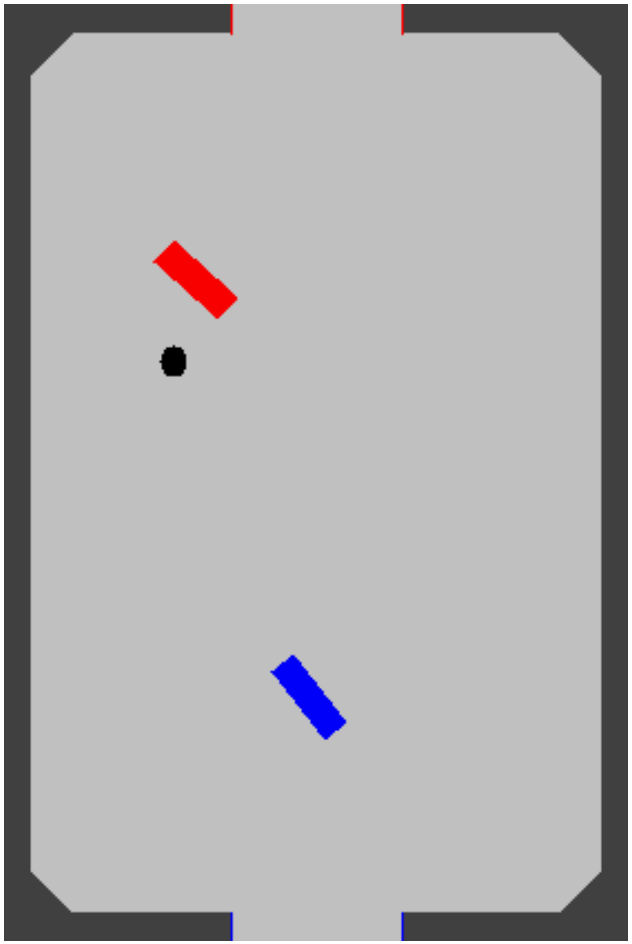
Percept sequence	Action
$[A, Clean]$	$Right$
$[A, Dirty]$	$Suck$
$[B, Clean]$	$Left$
$[B, Dirty]$	$Suck$
$[A, Clean]$, $[A, Clean]$	$Right$
$[A, Clean]$, $[A, Dirty]$	$Suck$
⋮	⋮

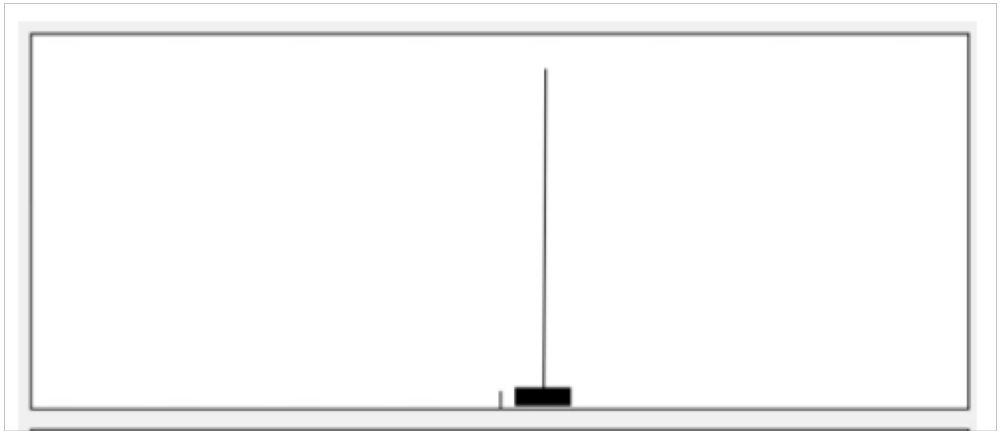
function REFLEX-VACUUM-AGENT([*location*,*status*]) **returns** an action

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

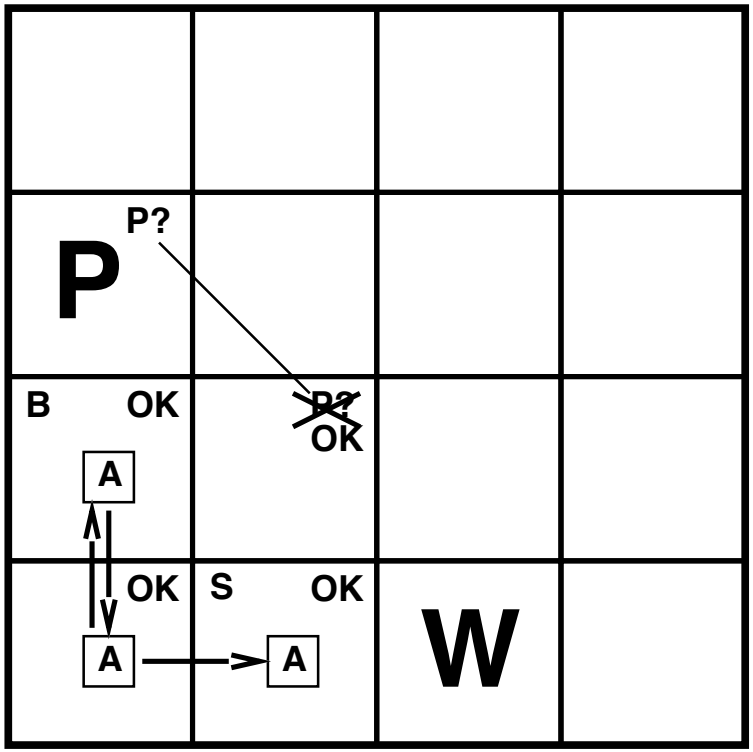
condition-action rules

Reactive Agent



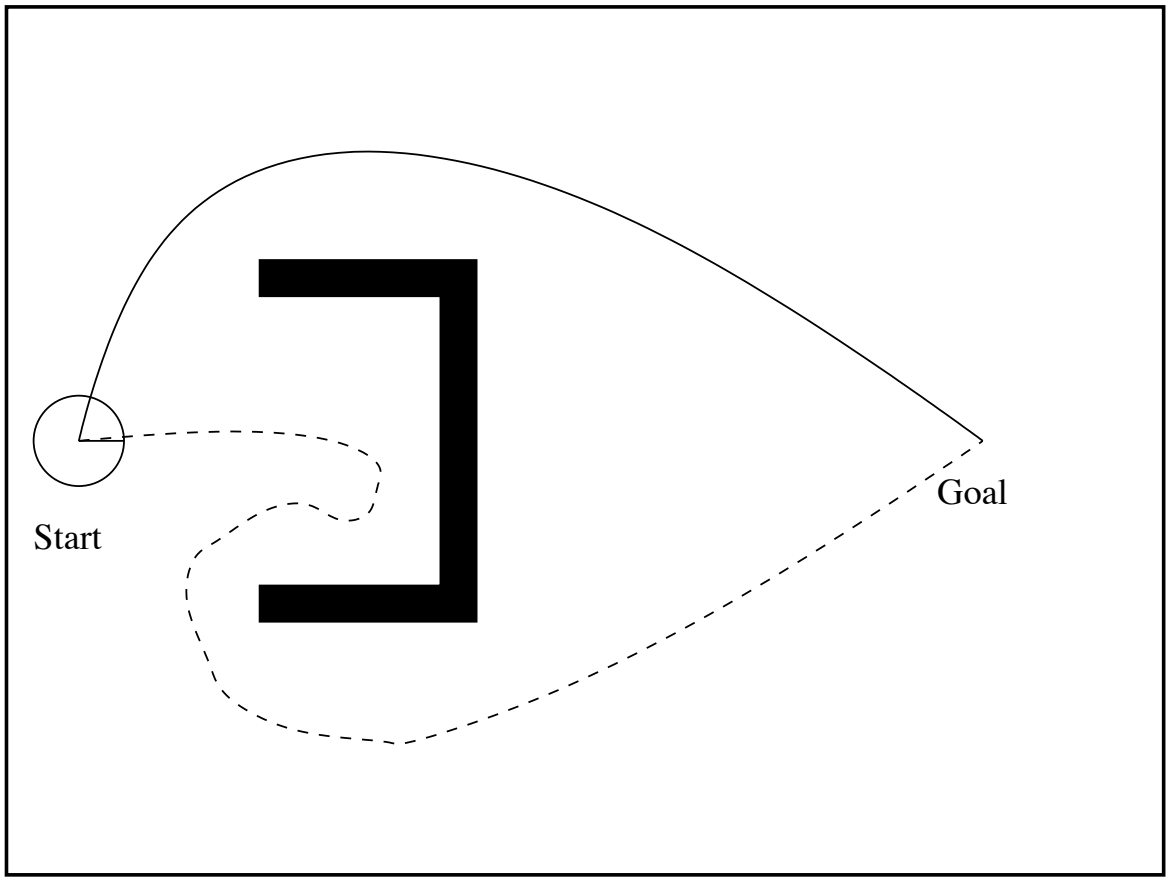


Limitations of Reactive Agents



- Sometimes we need to remember previous percepts in order to make sensible choices.

Limitations of Reactive Agents

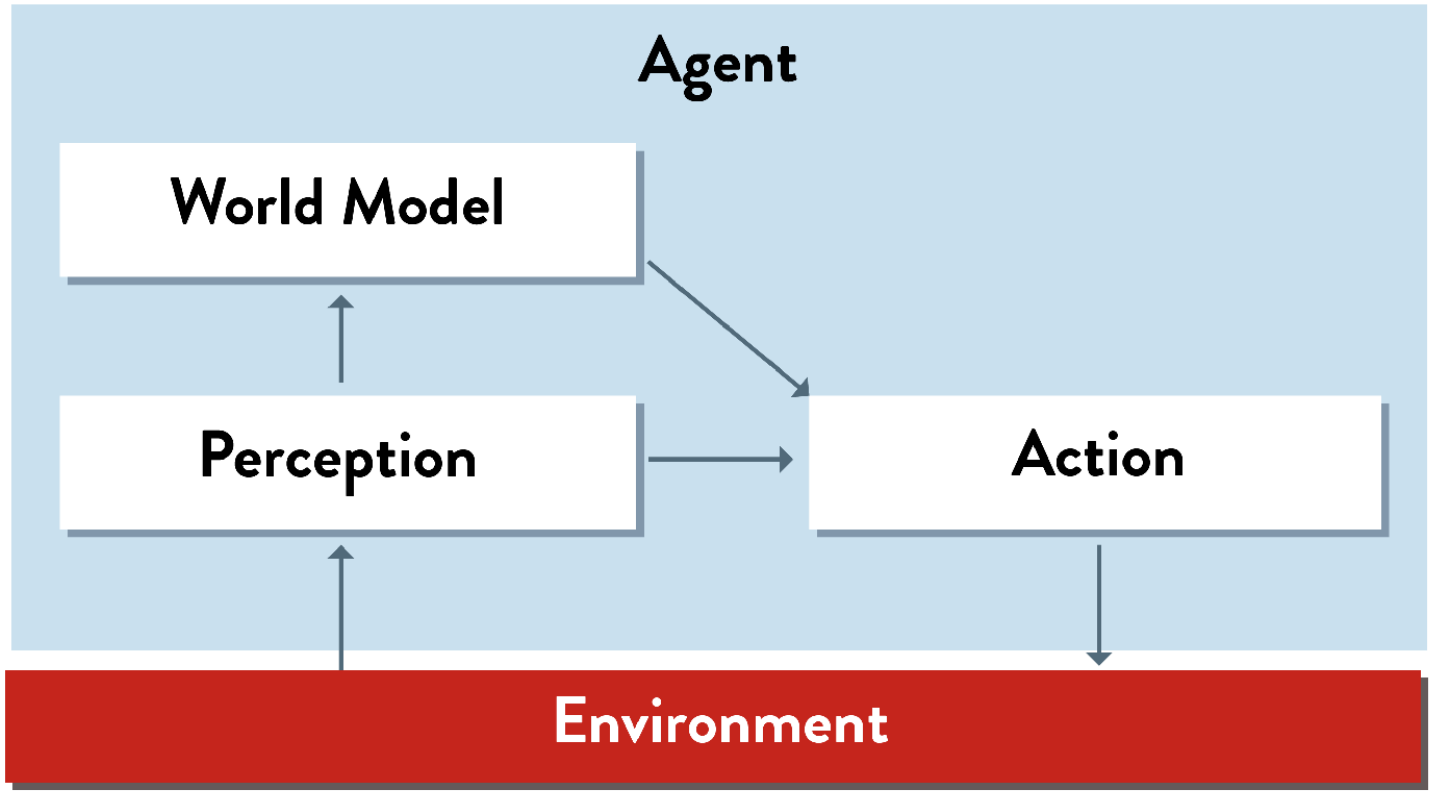


Limitations of Reactive Agents

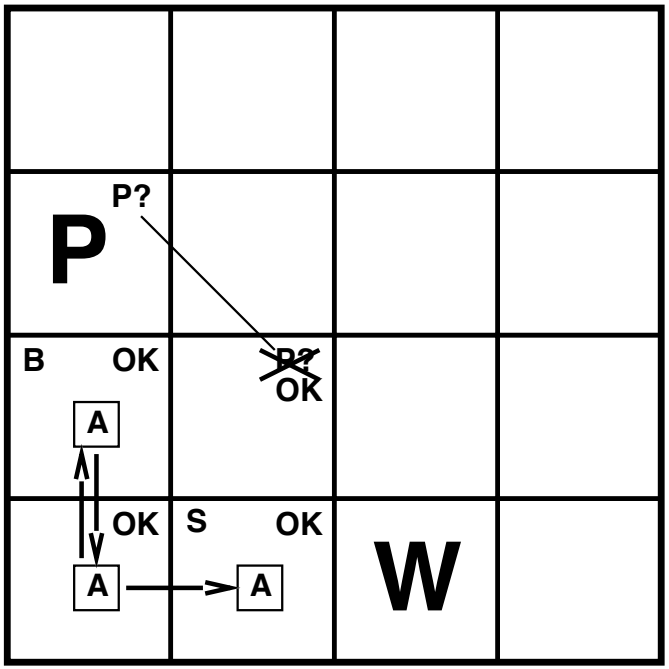
- Reactive Agents have no memory or "state"
  - unable to base decision on previous observations
  - may repeat the same sequence of actions over and over

- This phenomenon can also be observed in nature
  - wasp dragging stung grasshopper into its nest

Model-Based Agent



Advantages of Model-Based Agent



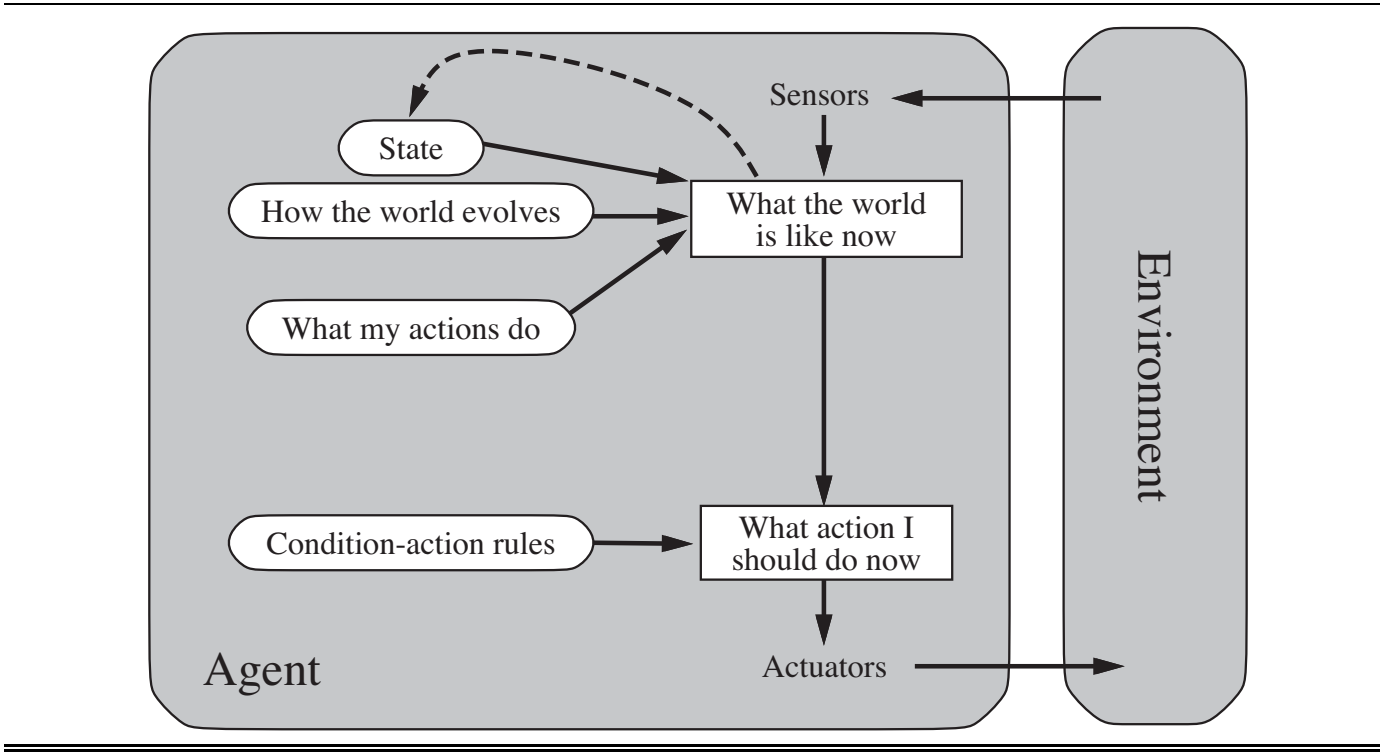
- Model-Based Agent can keep a "map" of the places it has visited, and remember what it perceived there. Sometimes we may need to plan several steps into the future.

Model-based agents

Model-based reflex agents

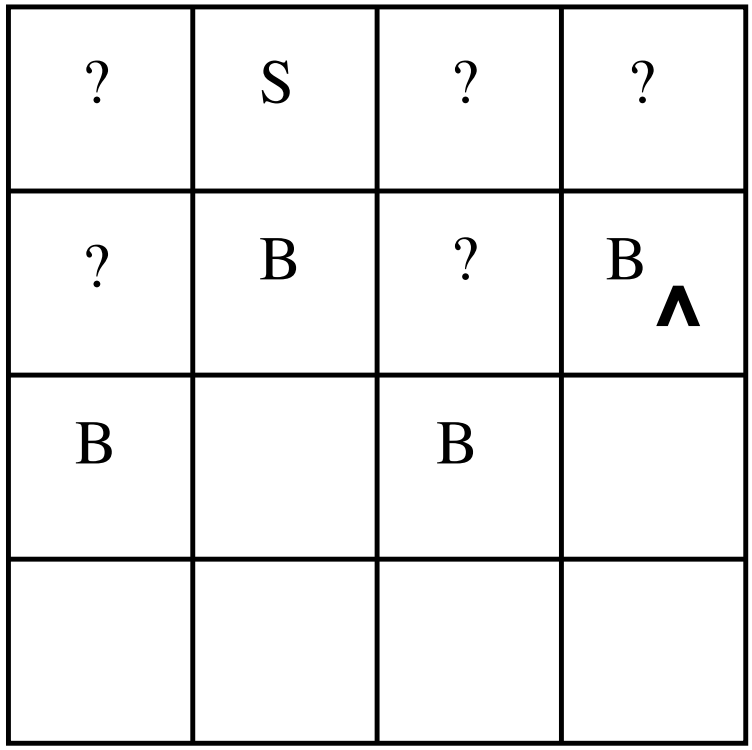
- The most effective way to handle partial observability is for the agent to *keep track of the part of the world it can't see now*.
- The agent should maintain some sort of **internal state** that depends on the percept history and thereby reflects at least some of the unobserved aspects of the current state.
- Knowledge about "how the world works"—is called a **model** of the world. An agent that uses such a model is called a **model-based agent**.

Model-based agent



A model-based reflex agent. 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.

Limitations of Model-Based Agents



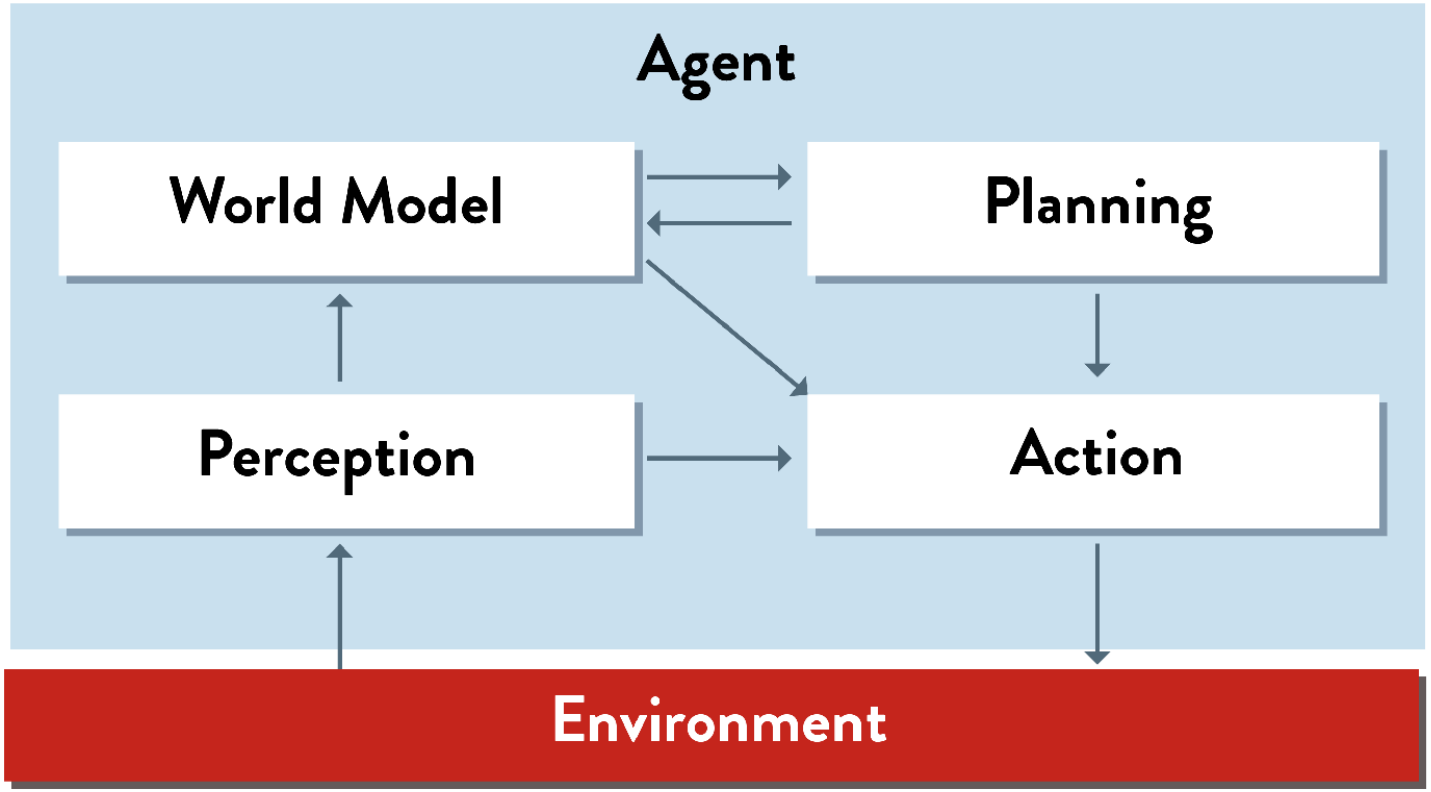
- Sometimes we may need to plan several steps into the future.

Limitations of Model-Based Agents

An agent with a world model but no planning can look into the past, but not into the future; it will perform poorly when the task requires any of the following:

- searching several moves ahead
  - Chess, Rubik's cube
- complex tasks requiring many individual step
  - cooking a meal, assembling a watch
- logical reasoning to achieve goals
  - travel to New York

Planning Agent



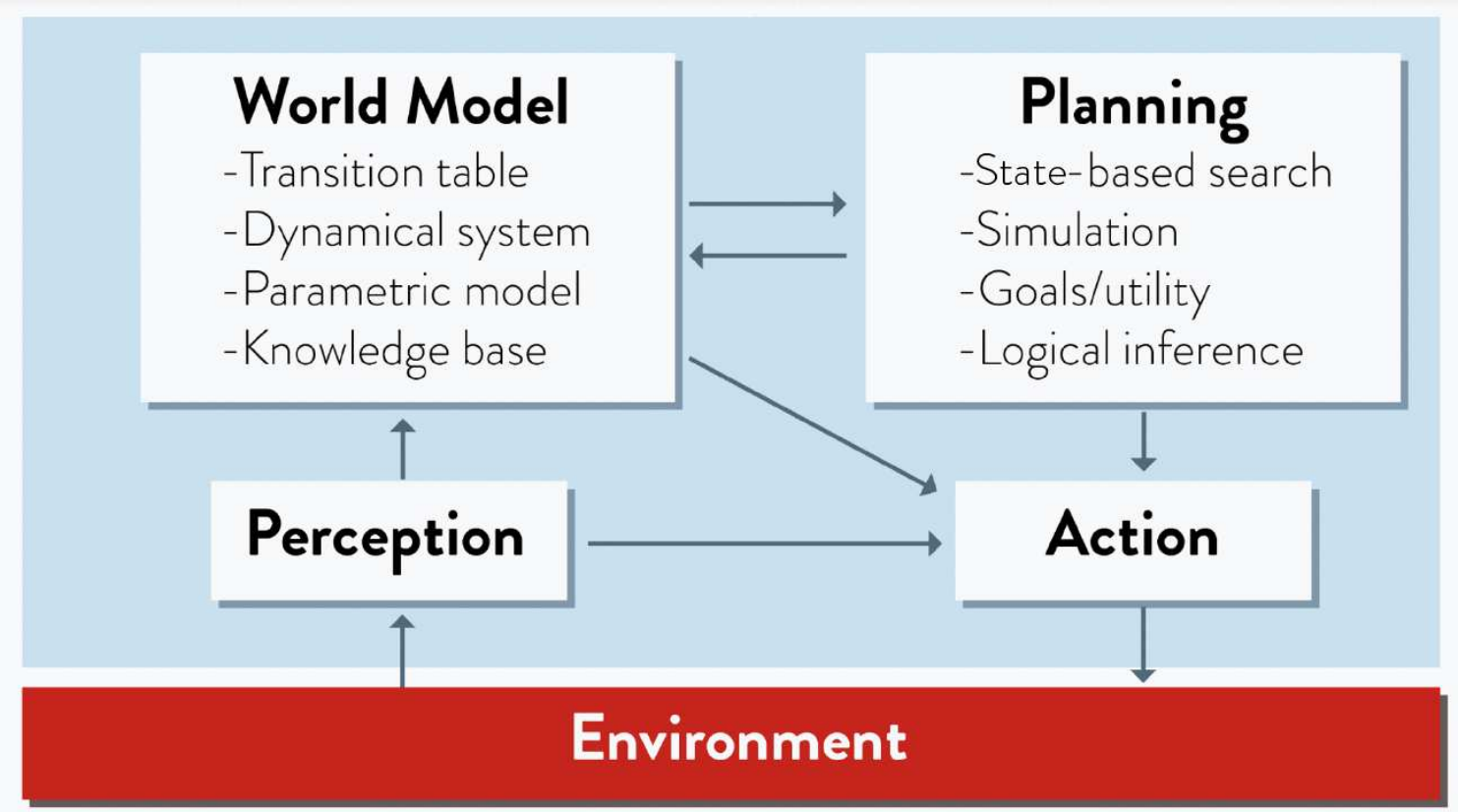
Goal-Based Agent

Planning Agent

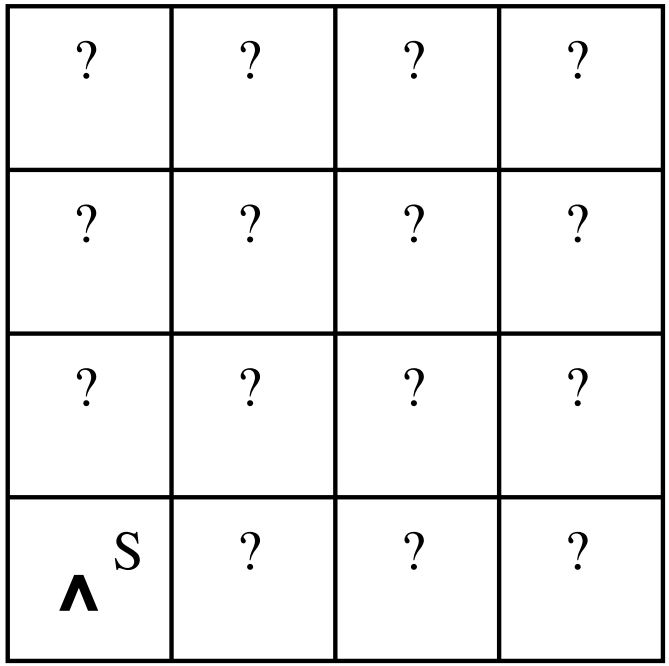
- Decision making of this kind is fundamentally different from the condition– action rules
- It involves consideration of the future
  - "What will happen if I do such-and-such?" and
  - "Will that make me happy?"

    - In the reflex agent designs, this information is not explicitly represented, because the built-in rules map directly from

Models and Planning



Reasoning about Future States



- What is the best action in this situation?

Faking it

- Sometimes an agent may appear to be planning ahead but is actually just applying reactive rules.

```
if( Glitter ) then
    Grab
else if( Stench ) then
    Shoot
else
 randomly Left, Right or Forward
```

- These rules can be hand-coded, or learned from experience.
- Agent may appear intelligent, but is not flexible in adapting to new situations.

Planning Agent

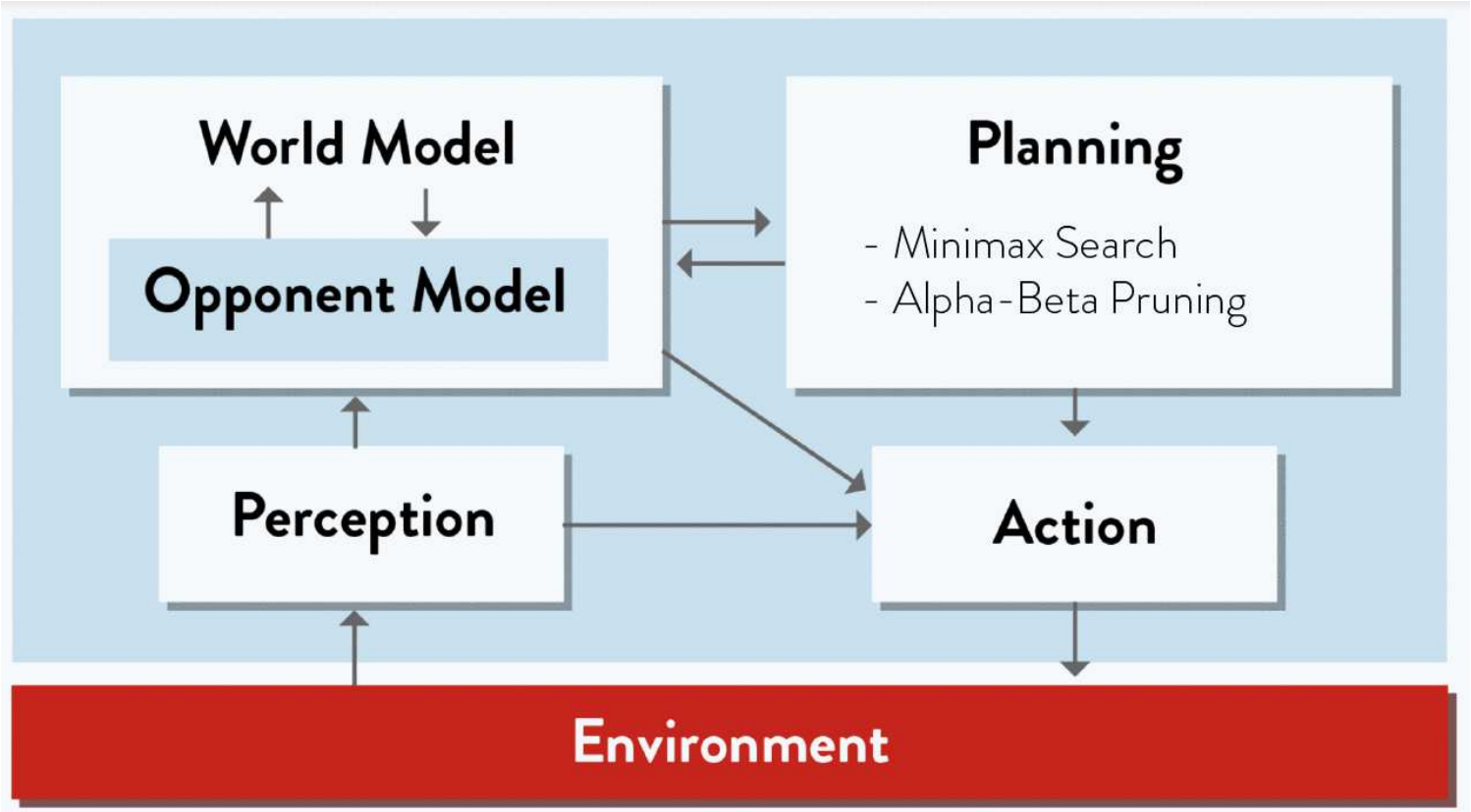
- The planning agent or goal-based agent is more flexible because the knowledge that supports its decisions is represented explicitly and can be modified.

- The agent's behavior can easily be changed.

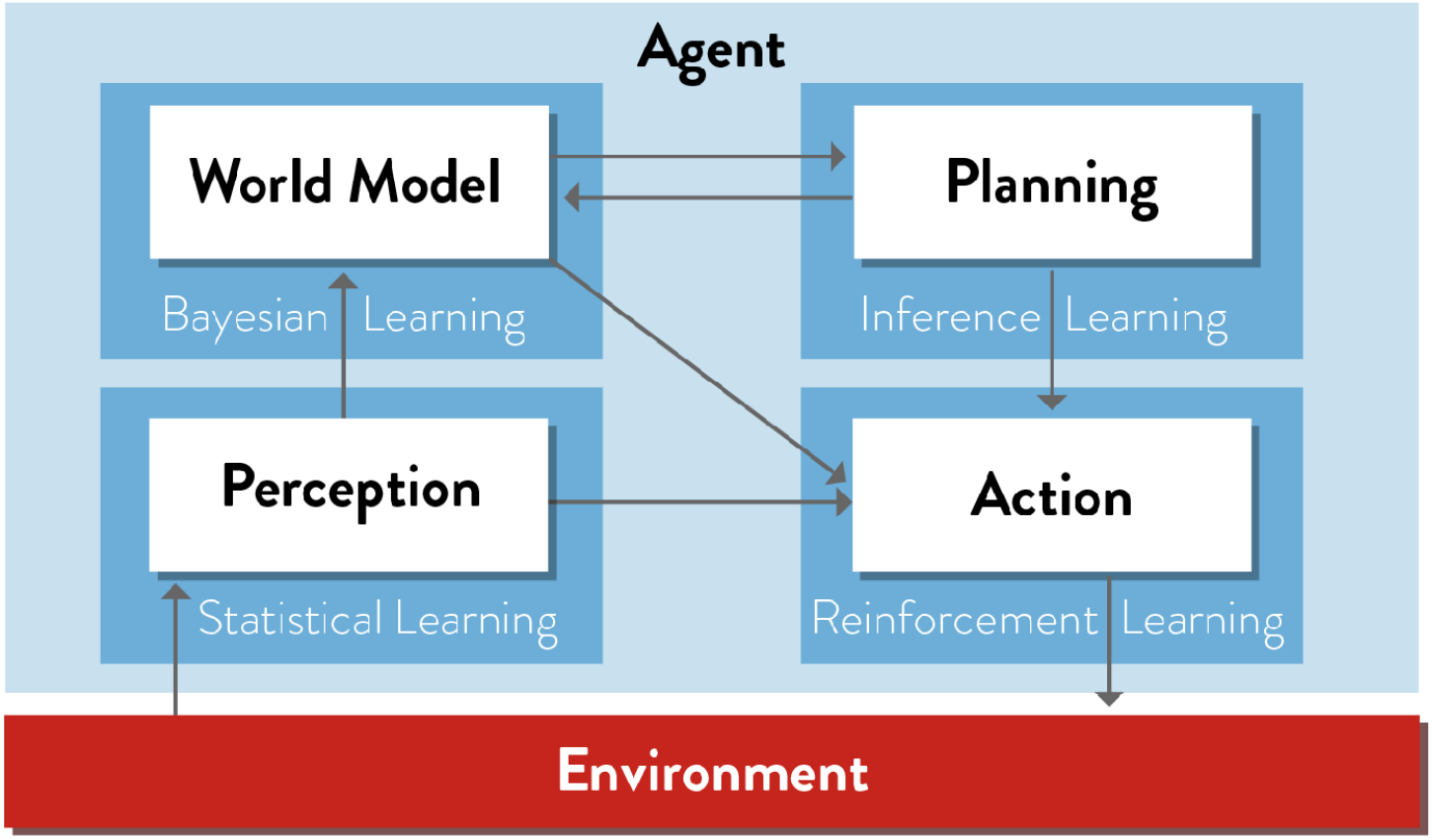
Utility-based agent

- A rational utility-based agent chooses the action that maximizes the **expected utility** of the action outcomes
  - that is, the utility the agent expects to derive, on average, given the probabilities and utilities of each outcome.
- The utility-based agent is not easy to implement
  - It has to model and keep track of its environment
  - Tasks involved a great deal of research on perception, representation, reasoning, and learning.
  - It can be implemented as a Decision-making agent that must handle the uncertainty inherent in stochastic or partially observable environments.

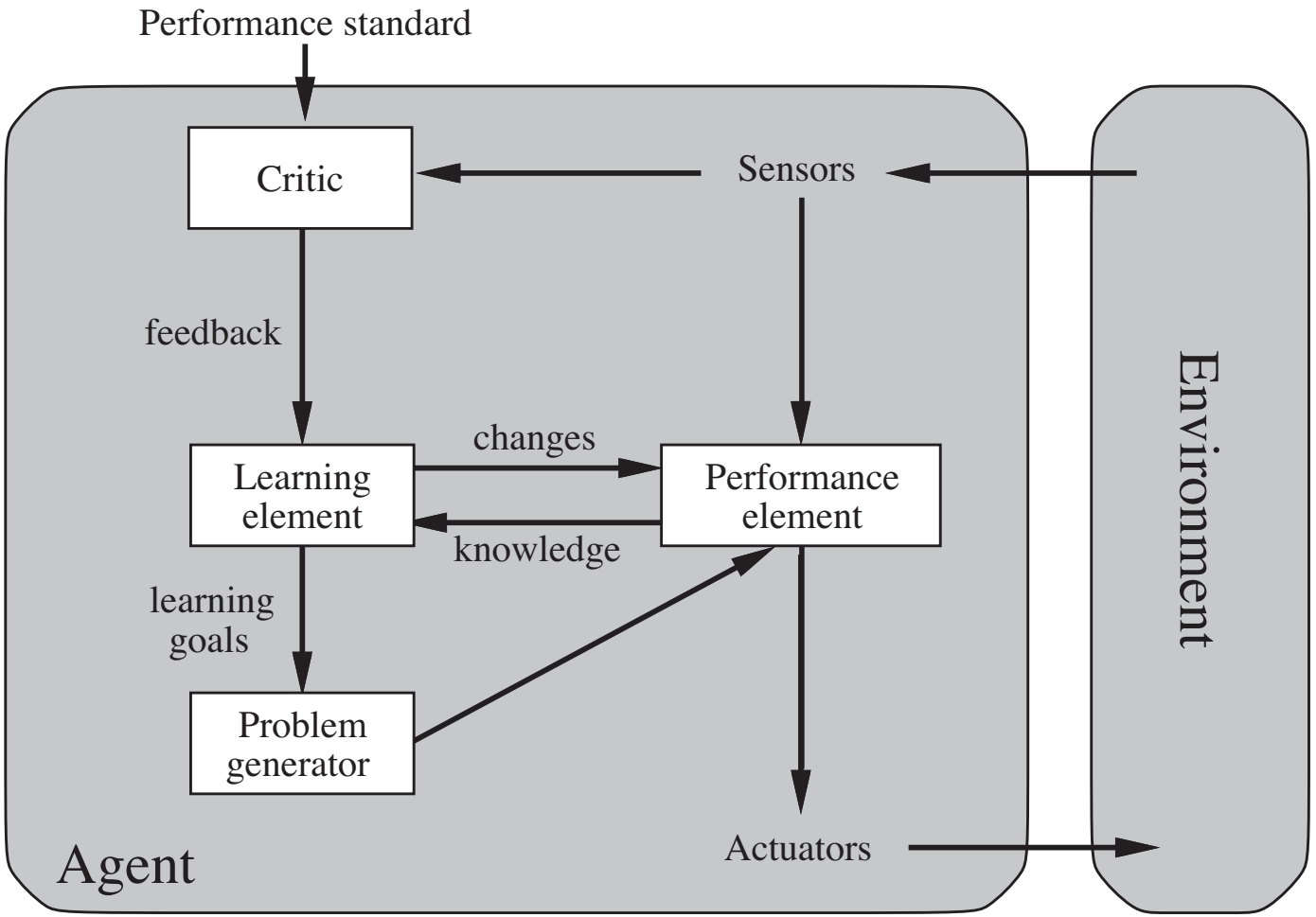
Game Playing Agent



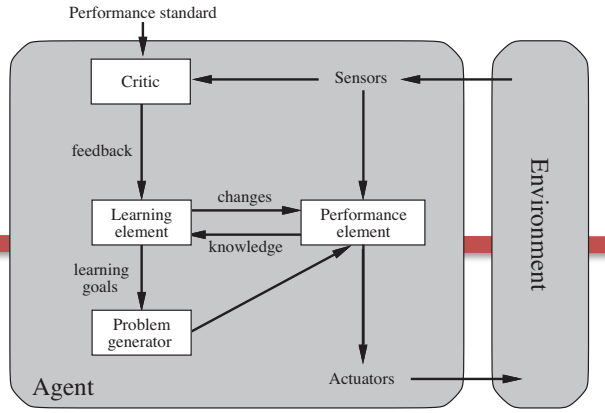
Learning Agent



Learning Agent



Learning Agent



- A learning agent can be divided into four conceptual components:
  - the **learning element**, which is responsible for making improvements,
  - the **performance element**, which is responsible for selecting external actions.
  - The **critic element -** tells the learning element how well the agent is doing with respect to a fixed performance standard.
  - The **problem generator** -for suggesting actions that will lead to new and informative experiences.

The performance element takes in percepts and decides on actions. The learning element uses feedback from the **critic** on how the agent is doing and determines how the performance element should be modified to do better in the future.

Learning

- Learning is not a separate module, but rather a set of techniques for improving the existing modules
- Learning is necessary because:
  - may be difficult or even impossible for a human to design all aspects of the system by hand
  - the agent may need to adapt to new situations without being re-programmed by a human

Summary

- The **agent program** implements the agent function.
- A variety of basic agent-program designs reflecting the kind of information made explicit and used in the decision process.
  - The designs vary in efficiency, compactness, and flexibility.
  - The appropriate design of the agent program depends on the nature of the environment.

- **Simple reflex agents** respond directly to percepts,
- **model-based reflex agents** maintain internal state to track aspects of the world that are not evident in the current percept.
- **Planning (Goal-base) agents** act to achieve their goals, and
- **utility-based agents** try to maximize their own expected "happiness."
- All agents can improve their performance through **learning**.