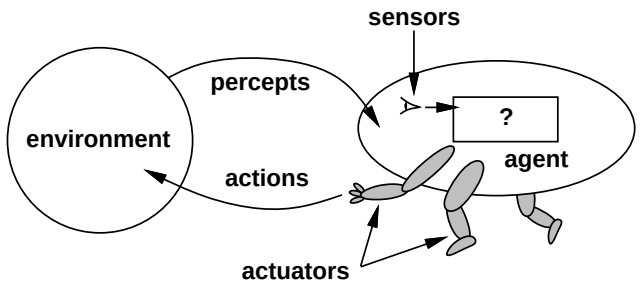
COMP9414/9814/3411: Artificial Intelligence

3. Agent Types

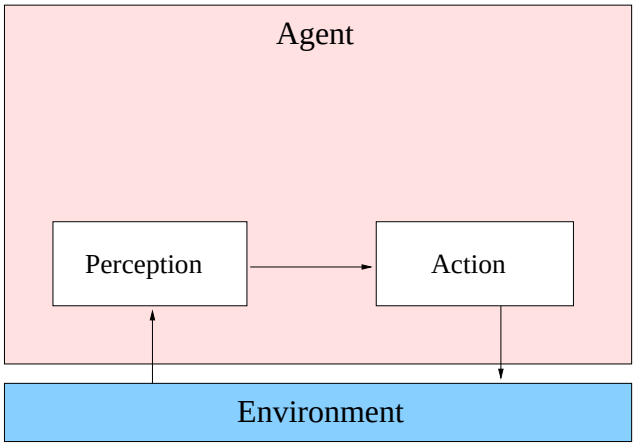
Outline

- Reactive Agents
- Model-Based Agents
- Planning Agents
- Learning Agents

Agent Model



Reactive Agent

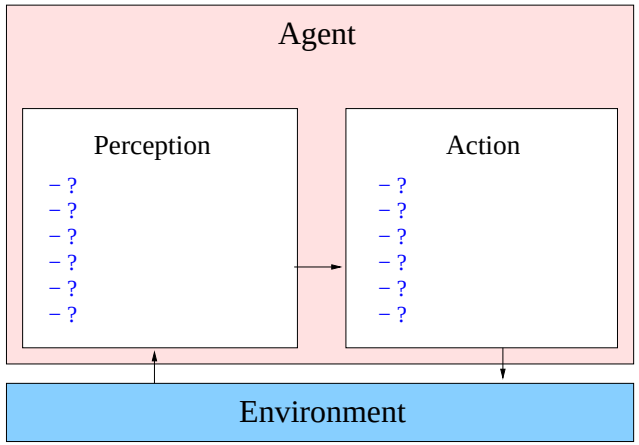


Reactive Agents

- Choose the next action based only on what they currently perceive, using a "policy" or set of rules which are simple to apply
- Sometimes pajoratively called "simple reflex agents" – but they can do surprisingly sophisticated things!
  - Swiss robots
  - simulated hockey

Robots

Exercise - Perception and Action



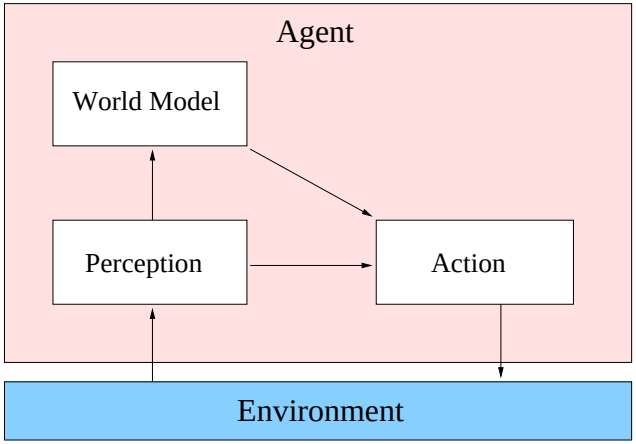
Limitations of Reactive Agents

- Try playing the Wumpus World as a Reactive Agent, i.e. only knowing the current percept and nothing else.
- Sometimes you need to remember previous percepts in order to make sensible choices.

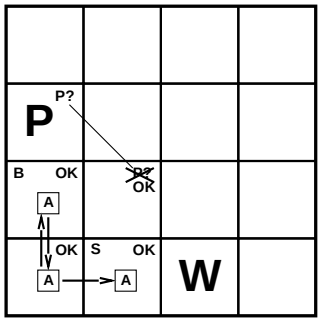
Limitations of Reactive Agents

- 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.

Limitations of Model-Based Agent

?	?	?	B ^
?	?	?	B
B S	?	B	
	B		

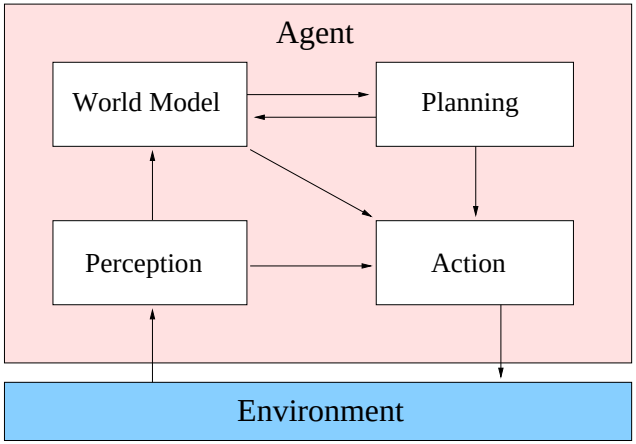
- Sometimes we may need to plan several steps into the future.

Limitations of Model-Based Agent

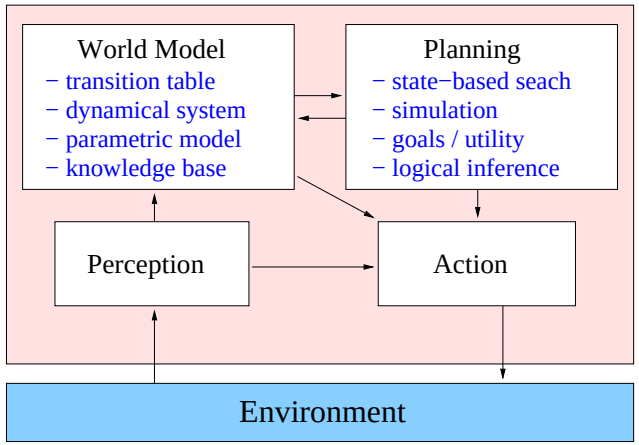
An agent with a world model but no planning can look into the past, but not into the future; it will perform poorly when the task requires any of the following:

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

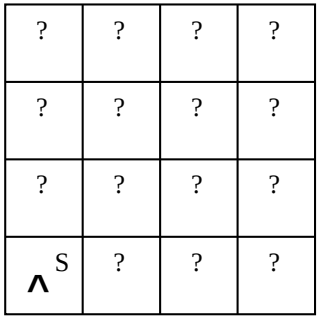
Planning Agent



Models and Planning



Reasoning about Future States



- What is the best action in this situation?

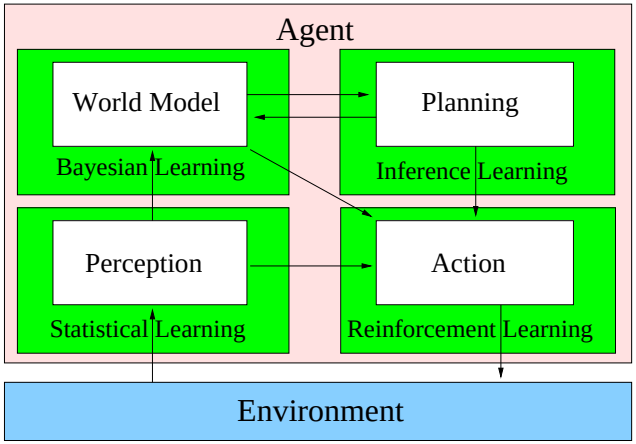
Faking it

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

Learning Agent



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

Learning

We must distinguish complexity of learning from complexity of application.

The policy for the simulated hockey player took several days of computation to derive (in this case, by evolutionary computation) but, once derived, it can be applied in real time.