

Agents and Environments

Lecture 2

How do we characterize environments?

What is an agent?

What characterizes rational behavior?

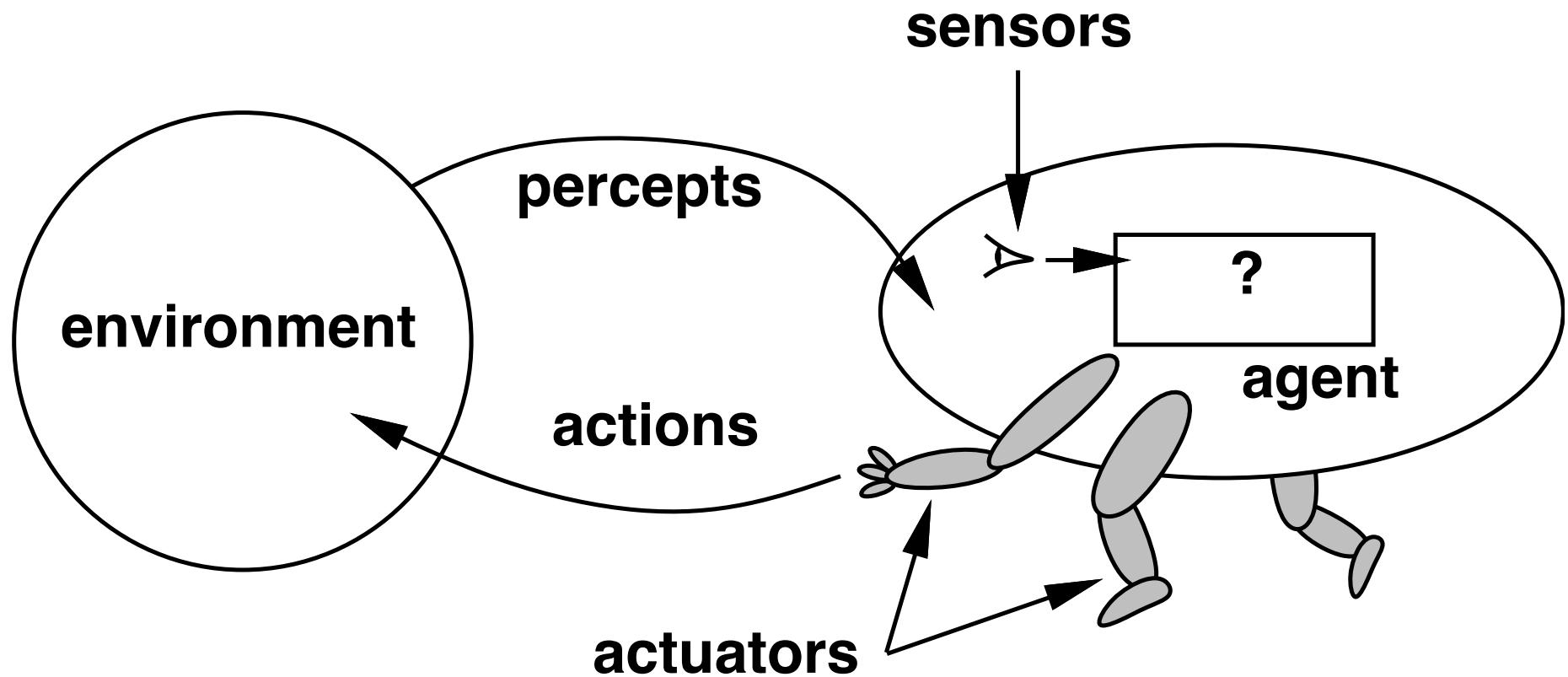


Agenda

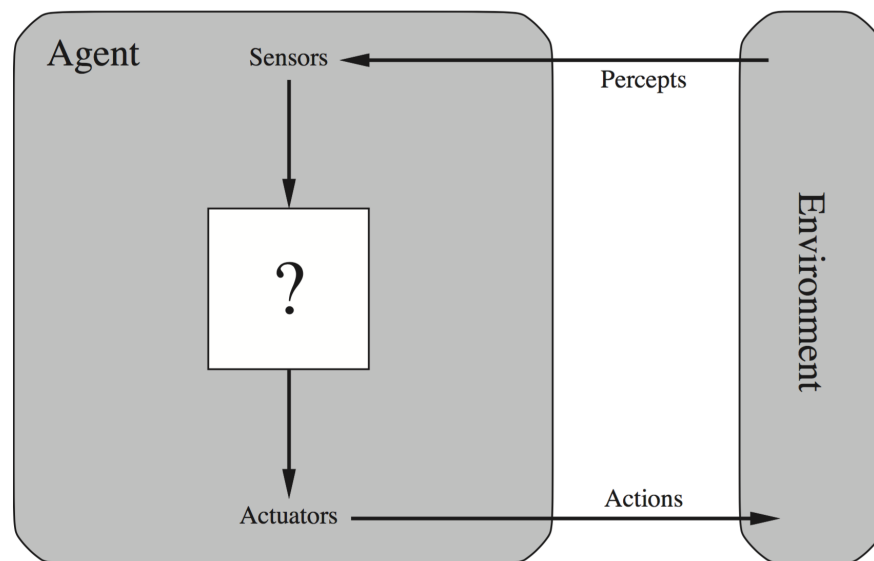
- Interaction model
- Rationality
- Task environments
- Types of agents



Agent-Environment Interaction



Agent-Environment Interaction



- An **agent** is anything that perceives its environment through **sensors** and acts via **actuators**
 - In AI: non-trivial decision-making + significant computation
- **Percept** refers to sensor values at an instant; **percept sequence** is a complete history



Agent Behavior

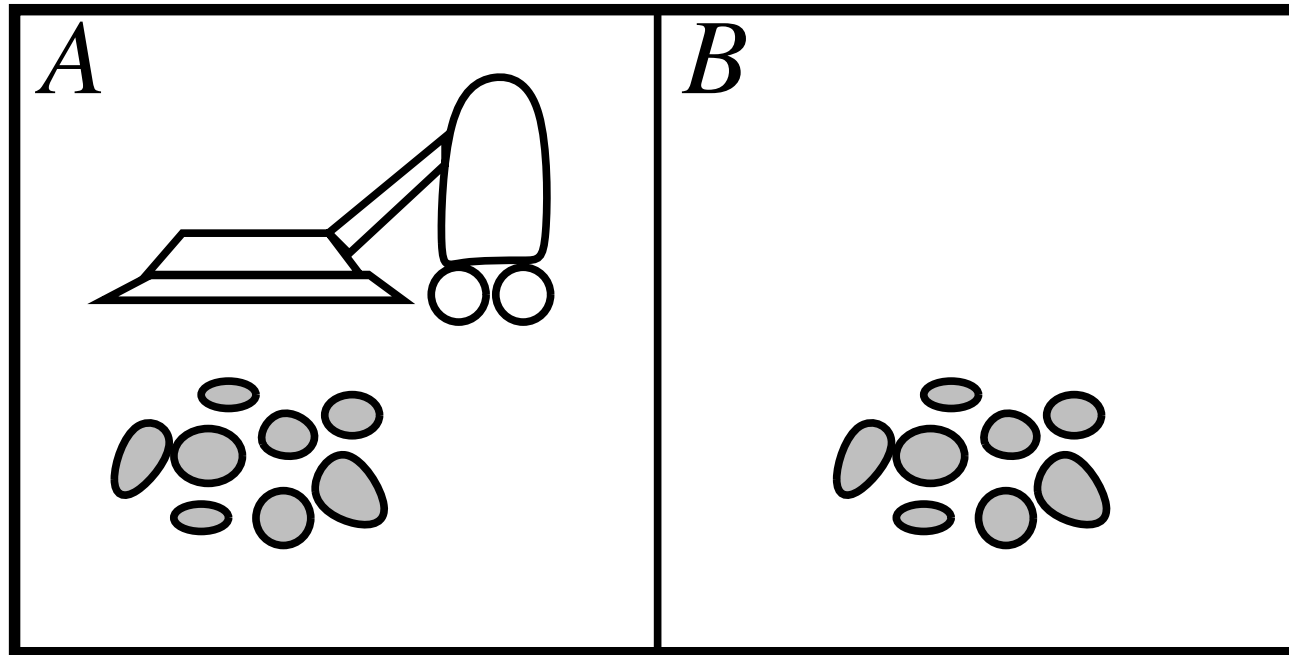
- Mathematically/externally, we consider the **agent function** as a mapping between an arbitrary percept sequence and an action

$$f : P^* \rightarrow A$$

- As AI practitioners, we implement the function via an **agent program**



Example: vacuum-cleaner World



Percepts: [location, status] (e.g. [A, Dirty])

Actions: Left, Right, Suck, NoOp



Example vacuum-cleaner Agent

Percept sequence	Action
$[A, \textit{Clean}]$	\textit{Right}
$[A, \textit{Dirty}]$	\textit{Suck}
$[B, \textit{Clean}]$	\textit{Left}
$[B, \textit{Dirty}]$	\textit{Suck}
$[A, \textit{Clean}], [A, \textit{Clean}]$	\textit{Right}
$[A, \textit{Clean}], [A, \textit{Dirty}]$	\textit{Suck}
\vdots	\vdots

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

if $status = \textit{Dirty}$ **then return** \textit{Suck}
else if $location = A$ **then return** \textit{Right}
else if $location = B$ **then return** \textit{Left}

What is the right function?



Evaluating Behavior

To evaluate agent behavior, we consider a **performance measure**

$$f : S_E^* \rightarrow V$$

Notes:

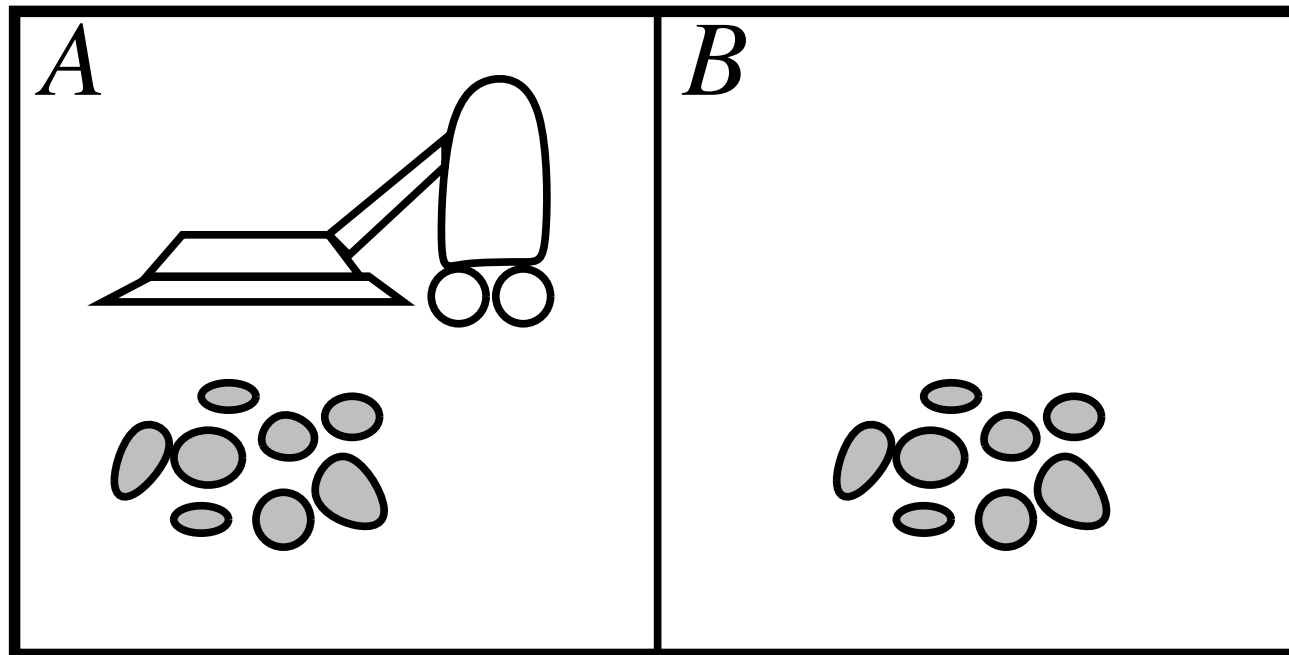
- Evaluates environment states, not agent percepts (more on observability later) or states (i.e. no fooling ourselves)
- One of many, not always easy to specify
 - Should be based upon desired outcomes, not expected agent design/operation



Example Performance Measures

- One point per square cleaned
 - Penalize per move
 - Penalize for $> k$ dirty squares

...



Defining Rationality

For each *possible* percept sequence, a rational agent should...

select an action that is expected to maximize its performance measure, given...

1. the percept sequence, and
2. *a priori* (i.e. prior) knowledge.



Exercise

Provide a reasoned argument as to whether an agent executing the program below is rational given the following assumptions:

- One point for each clean square at each time step over 1000 time steps
- Geography is known, but initial environmental state is not; clean stays clean, cleaning always works
- Perception is always accurate

```
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
```



Exercise

Provide a reasoned argument as to whether an agent executing the program below is rational given the following assumptions:

- One point for each clean square at each time step over 1000 time steps; **minus one point per Watt**
- Geography is known, but initial environmental state is not; clean stays clean, cleaning always works
- Perception is always accurate

```
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
```



Rationality \neq Omniscience

Knowing the actual outcome of one's actions.



Rationality \neq Perfection

Rationality implies information gathering, exploration, and learning

- Agents that rely upon prior knowledge vs. percepts lacks **autonomy**



PEAS Model

- Before designing an agent, we should fully specify the task environment (i.e. problem) it is to solve
- **P**erformance Measure
- **E**nvironment
- **A**ctuators
- **S**ensors



PEAS: Example

- **P**erformance
 - **E**nvironment
 - **A**ctuators
 - **S**ensors
- Safe, fast, legal, comfortable, profit!
 - Roads, traffic, pedestrians, customers
 - Steering, acceleration, brake, signal, horn, payment
 - Camera, sonar, speedometer, GPS, odometer, accelerometer, engine



Properties of Task Environments (1)

- Observability
 - Partially vs. Fully
- Agents
 - Single vs. Multi (competitive/cooperative)
- Certainty
 - Stochastic vs. Deterministic



Properties of Task Environments (2)

- Temporal independence
 - Episodic vs. Sequential
- Environmental change [during deliberation]
 - Static vs. Dynamic
- Representation [of states, time, percepts/actions]
 - Discrete vs. Continuous
- A priori environmental model
 - Known vs. unknown



Environment: Example (1)

- Fully observable
- Single agent
- Deterministic
- Sequential
- Static
- Discrete
- Known



Environment: Example (2)

- Partially observable
- Multi-agent, semi-cooperative
- Stochastic
- Sequential
- Dynamic
- Continuous
- Known

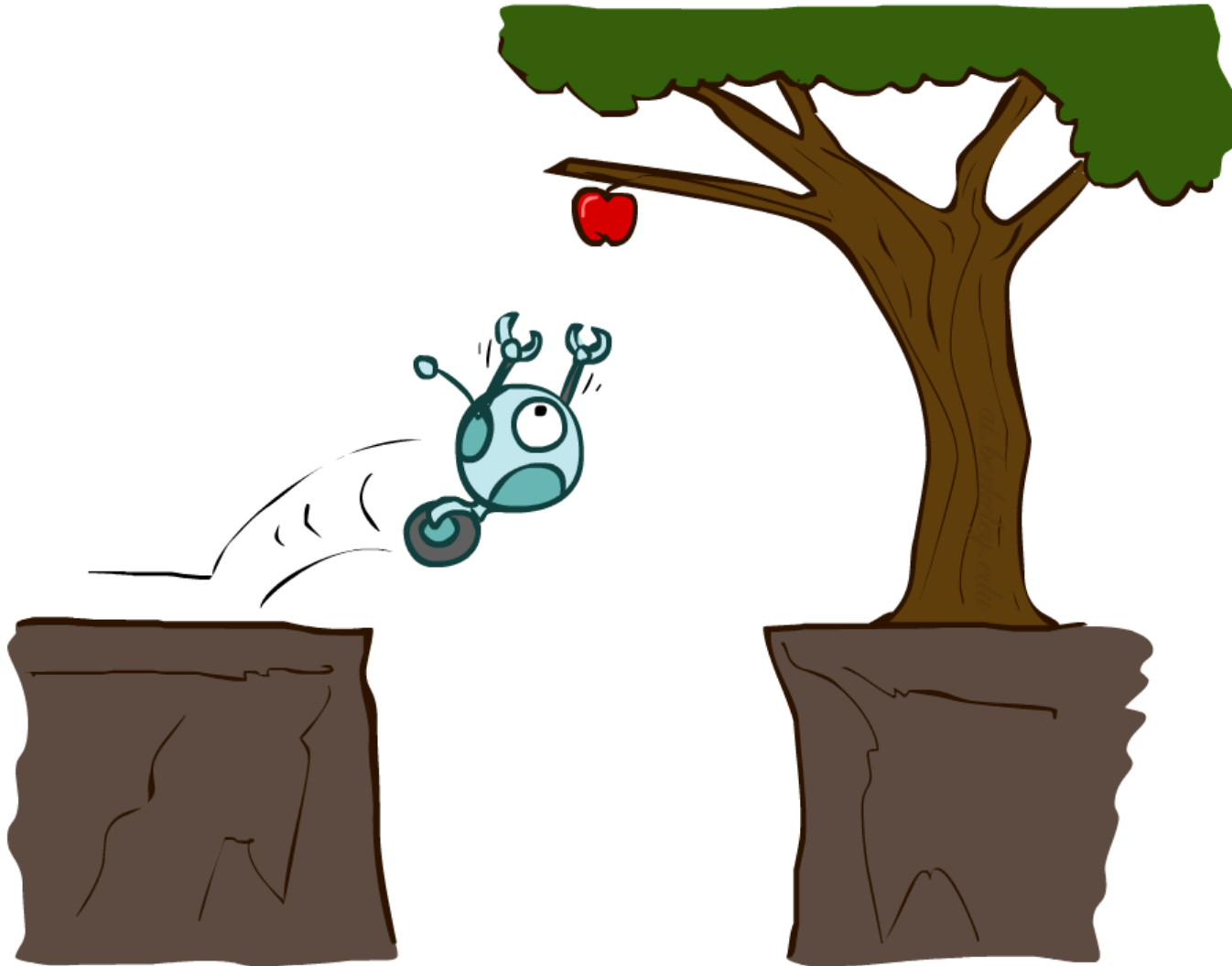


Agent Structure

- agent = architecture + program
- The key challenge for AI is to write [smallish] programs that produce rational behavior given complex environments
- We now examine 4 representative agent architectures

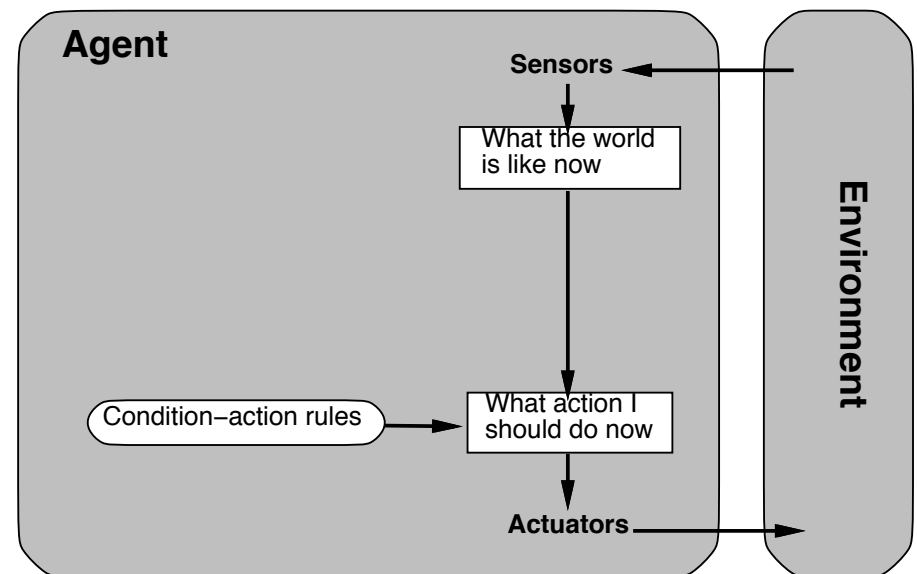


Reflexive Action



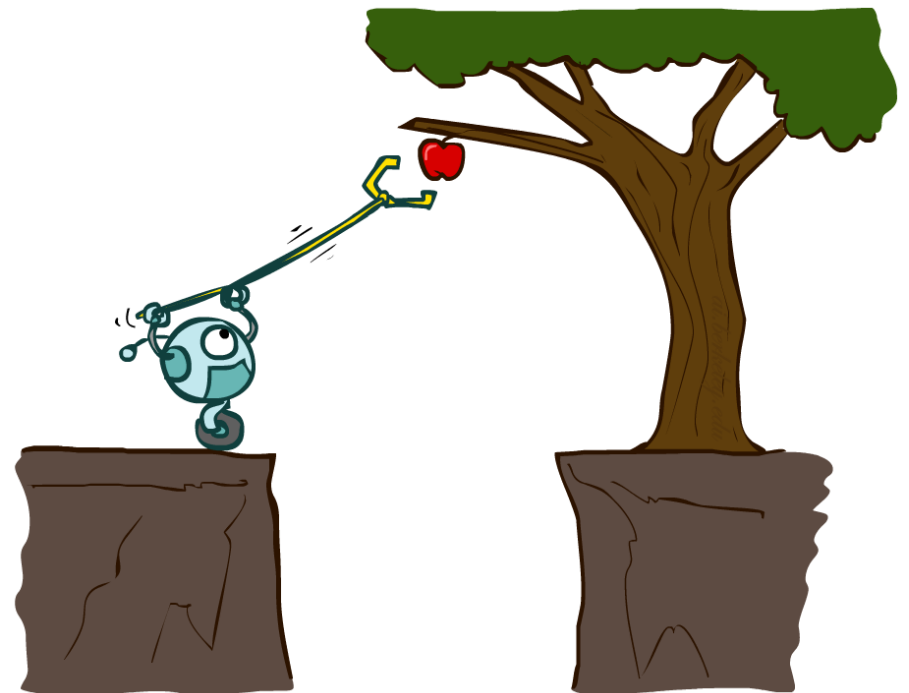
Simple Reflex Agents

- Select actions based upon the current percept, ignoring history
- Sees the world as it is, does not consider future consequences



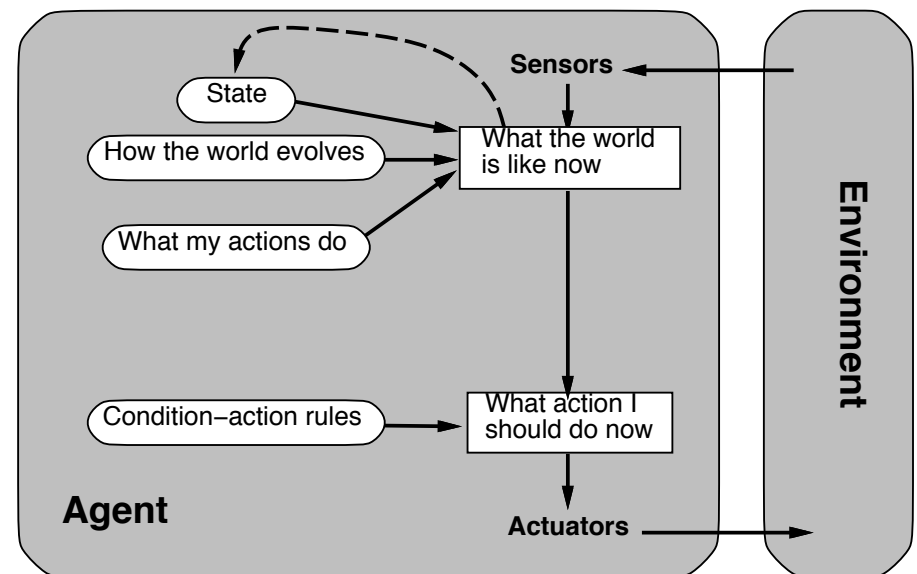
Adding Planning

- To handle partial observability, the agent needs to maintain **internal state**
 - Information it can't presently sense
- Updating requires **models** of the world
 - How the world evolves
 - Results of actions



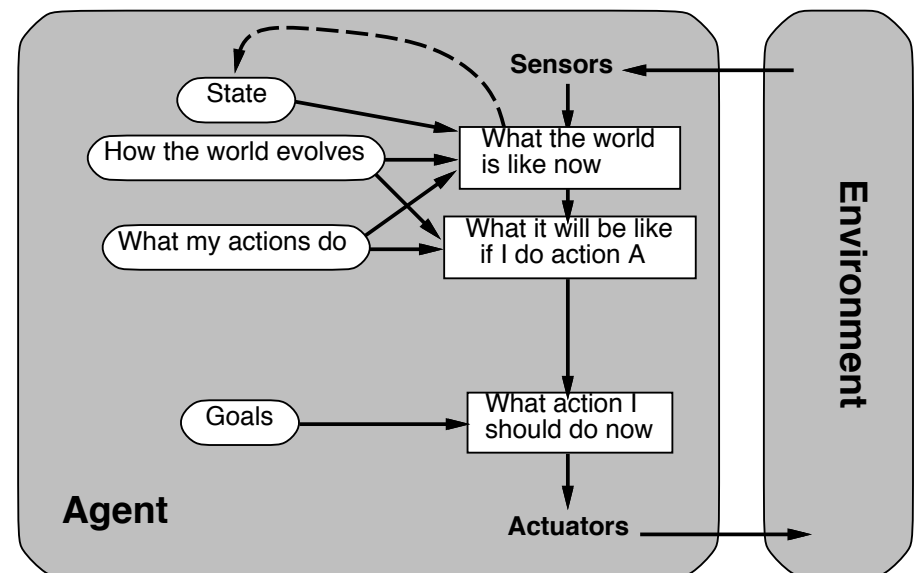
Model-based Reflex Agents

- Agent uses model + state to expand inputs to rules



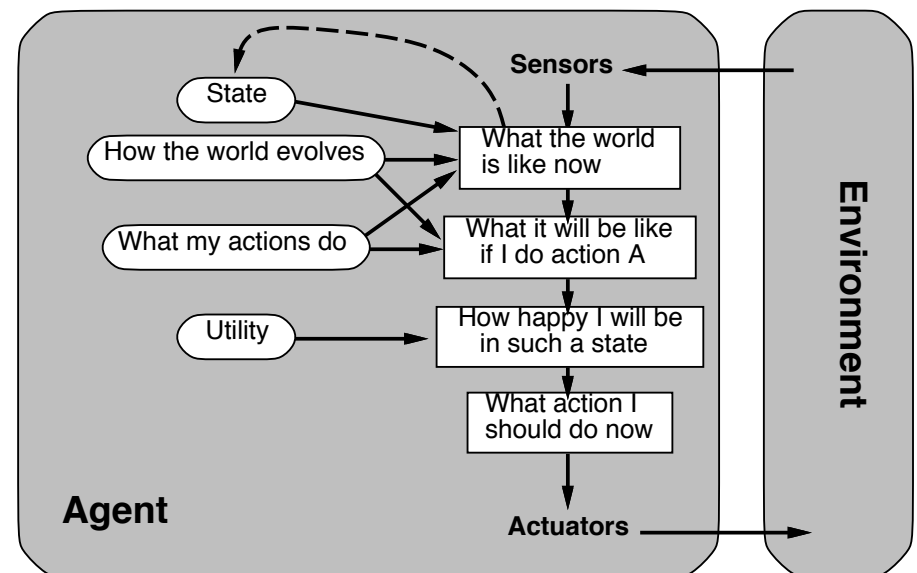
Goal-based Agents

- Incorporates both what the world is like, and goals are to be achieved
- More flexibility than rules: as long as new information relates to goals, can adapt



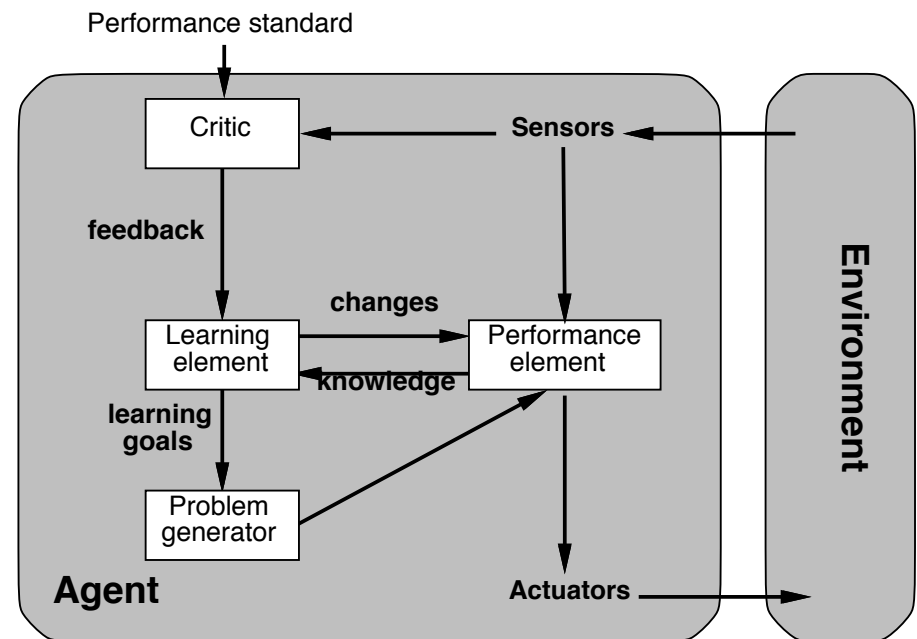
Utility-based Agents

- Utility: internalized performance measure
- Expands binary nature of goals
- A rational agent **must** behave *as if* it possesses a utility function whose expected value it tries to maximize



Learning Agents

- Performance element converts percepts into actions
- Learning element improves over time
- Critic converts percepts into good/bad (**reward/penalty**)
- Problem generator suggests actions to lead to “informative” experiences



Summary (1)

- **Agents** interact with **environments** through **sensors** and **actuators**
- The **agent function** describes what the agent *does* in all circumstances; the **agent program** is an actual *implementation*
- The **performance measure** evaluates the environment sequence; a **rational agent** maximizes *expected performance*



Summary (2)

- **PEAS** descriptions define task environments
 - Environments are described along numerous dimensions (observability, agents, certainty, temporal independence, environmental change, representation)
- Agent = architecture + program
 - Architectures: **reflexive** (with model]), **goal-based**, **utility-based**, **learning**

