Lecture 2

Agent and Environment

*

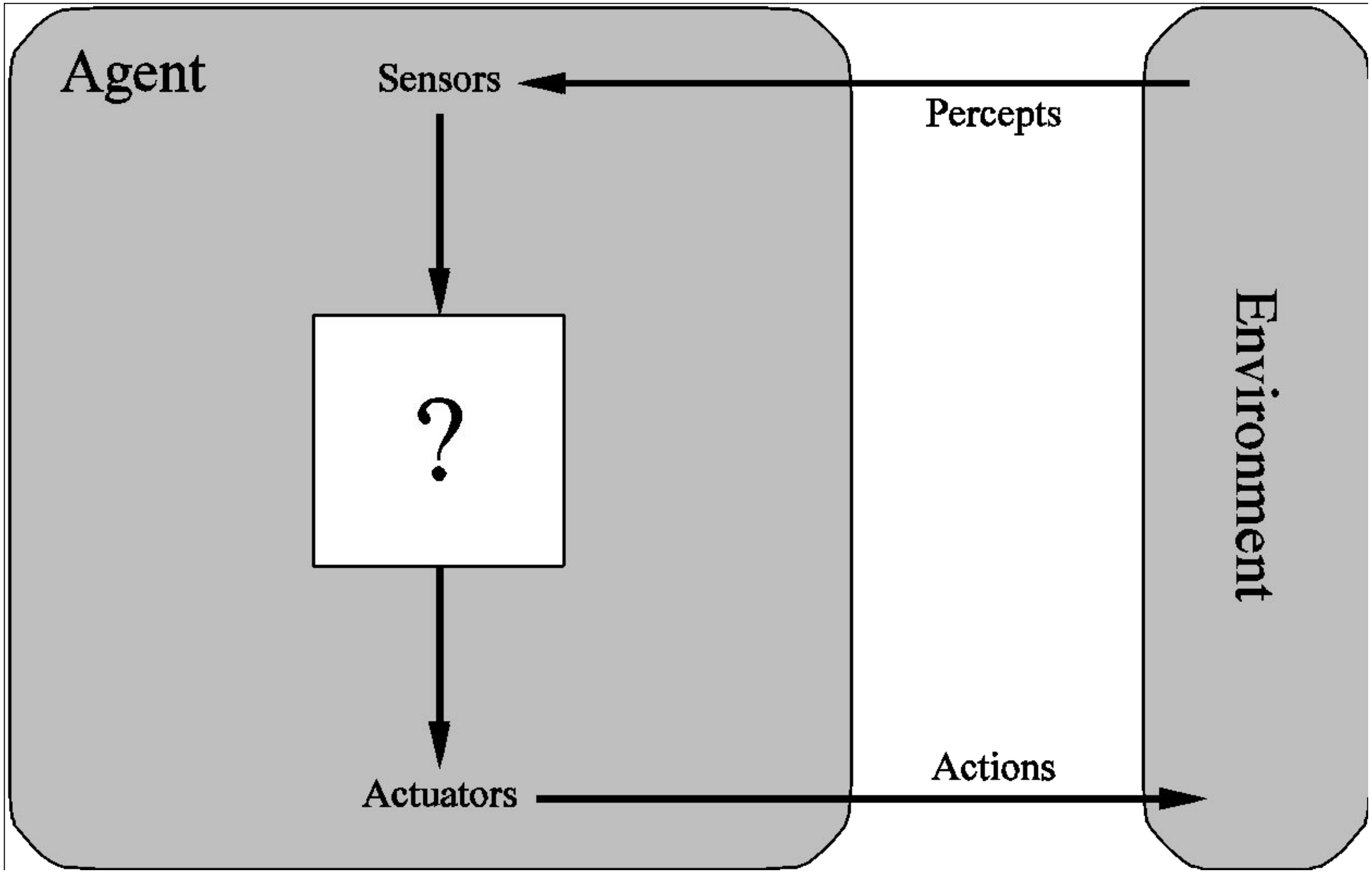
Today's class

- What's an agent?
  - Definition of an agent
  - Rationality and autonomy
  - Types of agents
  - Properties of environments

How do you design an intelligent agent?

- Definition: An **intelligent agent** perceives its environment via **sensors** and acts rationally upon that environment with its **actuators**.
- A discrete agent receives **percepts** one at a time, and maps this percept sequence to a sequence of discrete **actions**.
- Properties
  - Autonomous
  - Reactive to the environment
  - Pro-active (goal-directed)
  - Interacts with other agents via the environment

Agents



What do you mean, sensors/percepts and actuators/actions?

- Humans
  - Sensors: Eyes (vision), ears (hearing), skin (touch), tongue (gustation), nose (olfaction), neuromuscular system (proprioception)
  - Percepts:
    - At the lowest level – electrical signals from these sensors
    - After preprocessing – objects in the visual field (location, textures, colors, …), auditory streams (pitch, loudness, direction), …
  - Actuators: limbs, digits, eyes, tongue, …
  - Actions: lift a finger, turn left, walk, run, carry an object, …
- The Point: percepts and actions need to be carefully defined, possibly at different levels of abstraction

A more specific example: Automated taxi driving system

- **Percepts**: Video, sonar, speedometer, odometer, engine sensors, keyboard input, microphone, GPS, …
- **Actions**: Steer, accelerate, brake, horn, speak/display, …
- **Goals**: Maintain safety, reach destination, maximize profits (fuel, tire wear), obey laws, provide passenger comfort, …
- **Environment**: Urban streets, freeways, traffic, pedestrians, weather, customers, …
- **Different aspects of driving may require different types of agent programs!**

Rationality

- An ideal **rational agent** should, for each possible percept sequence, do whatever actions will maximize its expected performance measure based on

  (1) the percept sequence, and

  (2) its built-in and acquired knowledge.
- Rationality includes information gathering, not “rational ignorance.” (If you don’t know something, find out!)
- Rationality → Need a performance measure to say how well a task has been achieved.
- Types of performance measures: false alarm (false positive) and false dismissal (false negative) rates, speed, resources required, effect on environment, etc.

Autonomy

- A system is autonomous to the extent that its own behavior is determined by its own experience.
- Therefore, a system is not autonomous if it is guided by its designer according to a priori decisions.
- To survive, agents must have:
  - Enough built-in knowledge to survive.
  - The ability to learn.

Some agent types

- **Table-driven agents**
  - use a percept sequence/action table in memory to find the next action. They are implemented by a (large) **lookup table**.
- **Simple reflex agents**
  - are based on **condition-action rules**, implemented with an appropriate production system. They are stateless devices which do not have memory of past world states.
- **Agents with memory**
  - have **internal state**, which is used to keep track of past states of the world.
- **Agents with goals**
  - are agents that, in addition to state information, have **goal information** that describes desirable situations. Agents of this kind take future events into consideration.
- **Utility-based agents**
  - base their decisions on **classic axiomatic utility theory** in order to act rationally.

Table-driven/reflex agent architecture

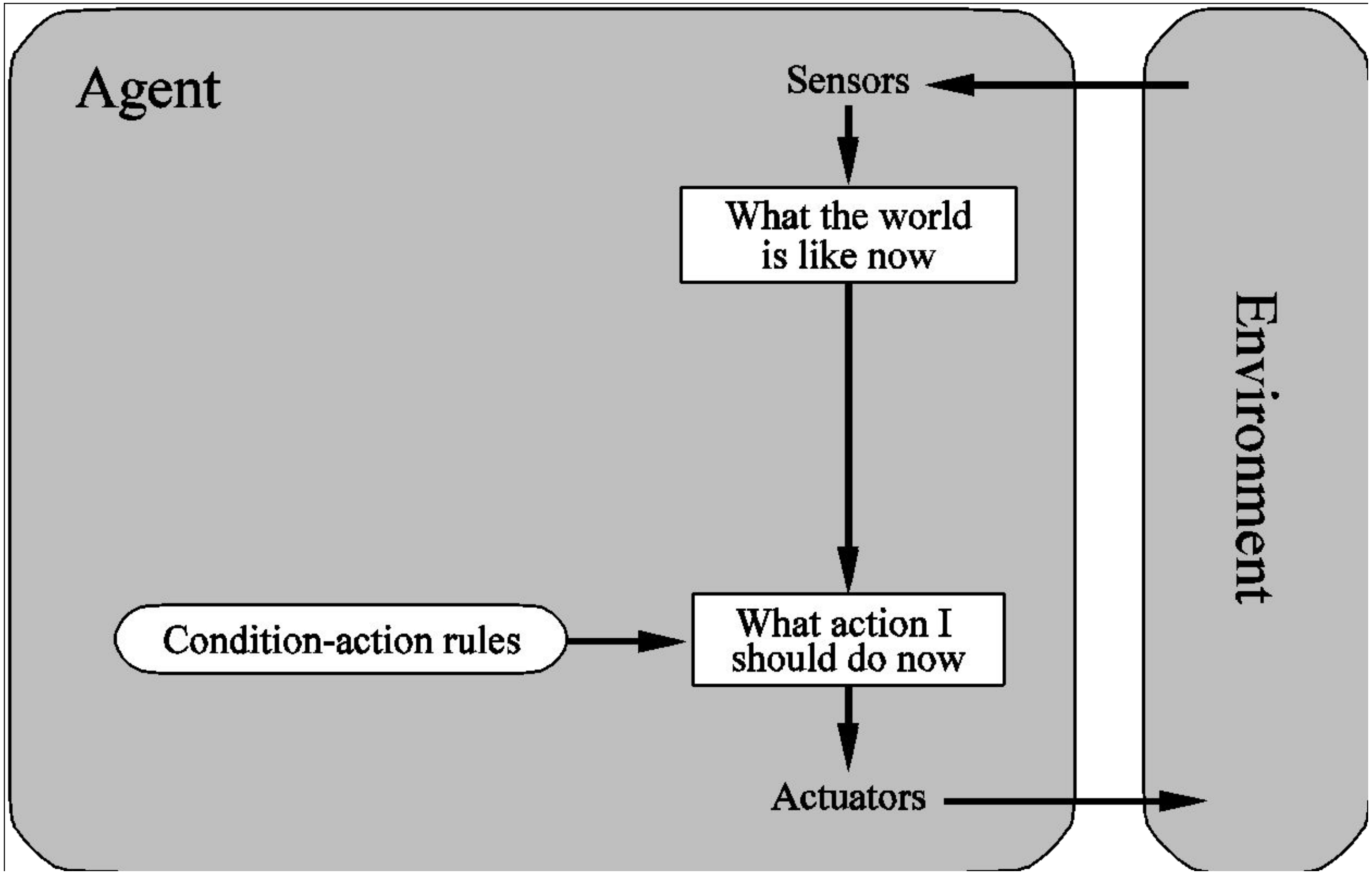
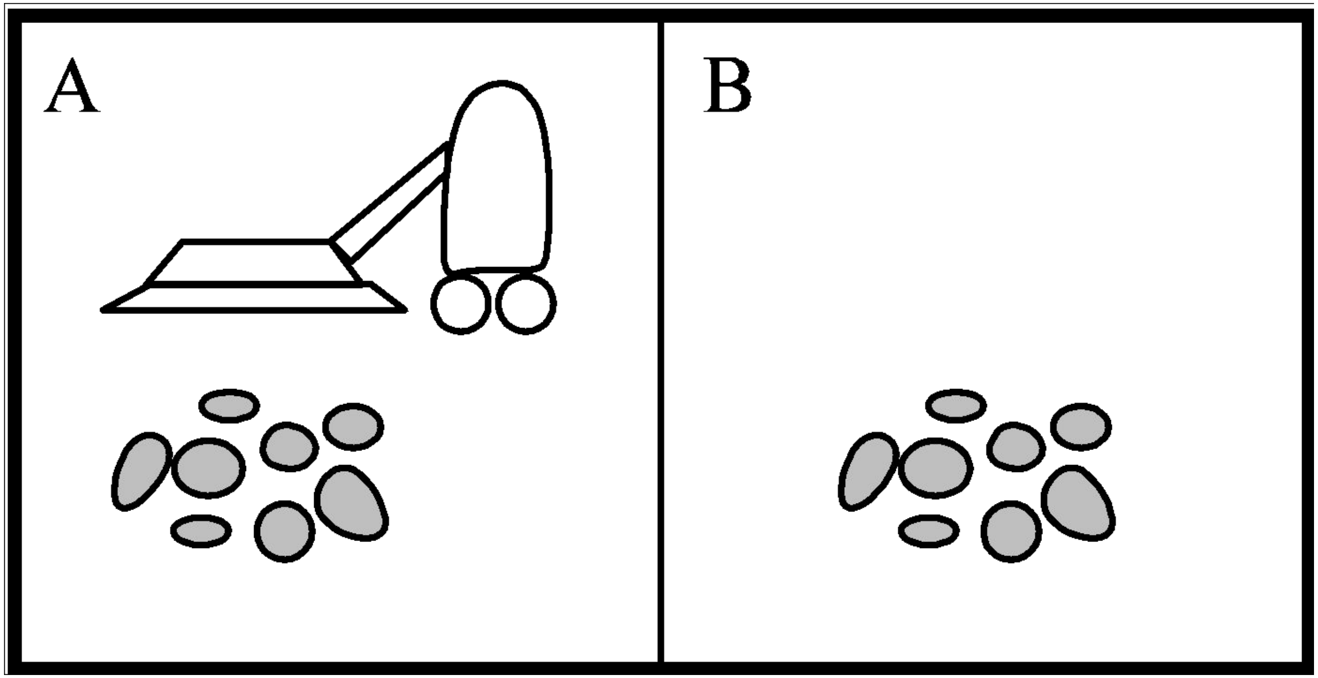


Table-driven agents

- **Table lookup** of percept-action pairs mapping from every possible perceived state to the optimal action for that state
- **Problems**
  - Too big to generate and to store (Chess has about 10^{120} states, for example)
  - No knowledge of non-perceptual parts of the current state
  - Not adaptive to changes in the environment; requires entire table to be updated if changes occur
  - Looping: Can't make actions conditional on previous actions/states

Example: Vacuum Cleaner



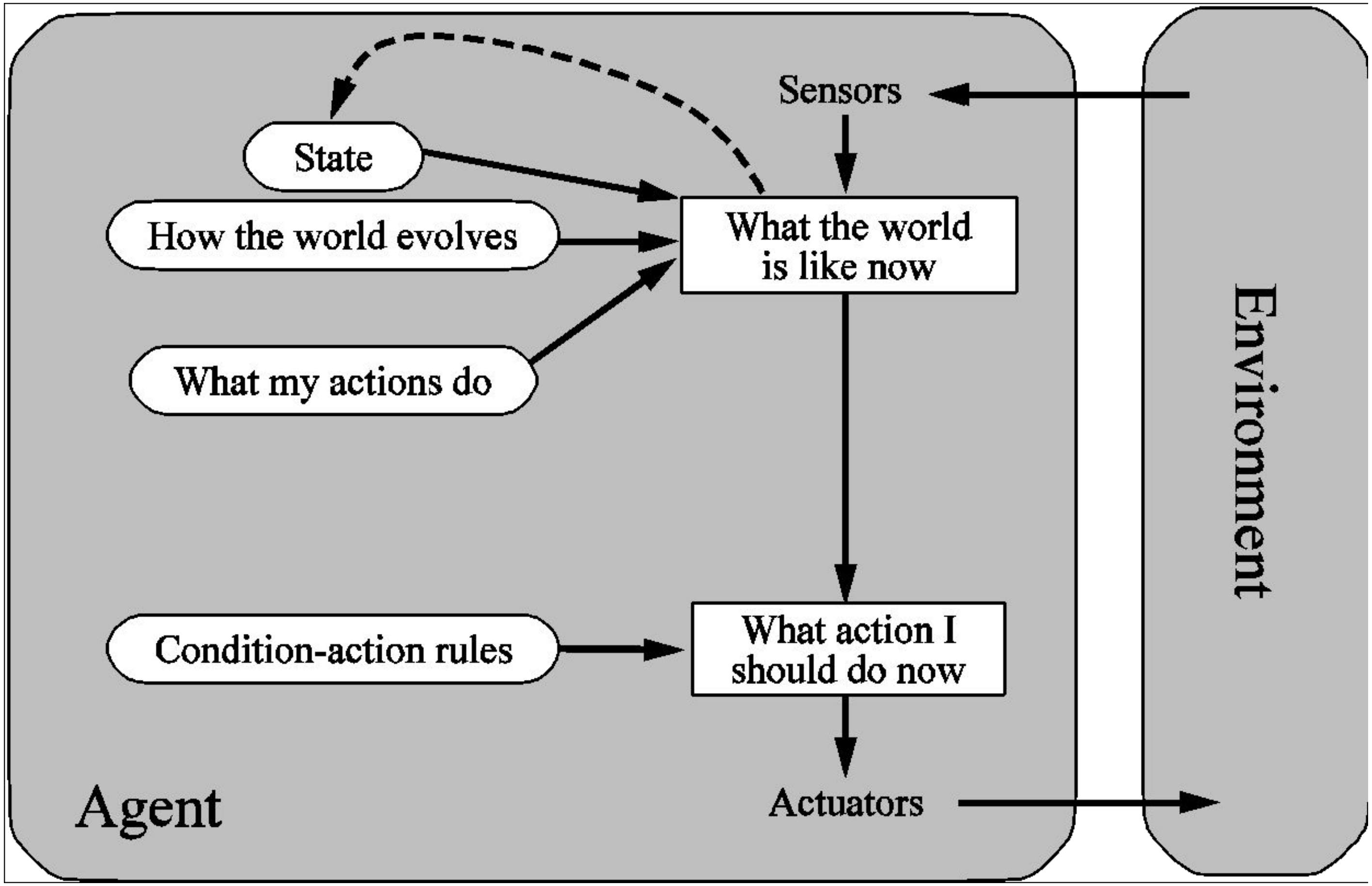
Table/Simple Function

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
…	
…	
[A, Clean], [A, Clean], [A, Dirty]	Suck

Simple reflex agents

- **Rule-based reasoning** to map from percepts to optimal action; each rule handles a collection of perceived states
- **Problems**
  - Still usually too big to generate and to store
  - Still no knowledge of non-perceptual parts of state
  - Still not adaptive to changes in the environment; requires collection of rules to be updated if changes occur
  - Still can’t make actions conditional on previous state

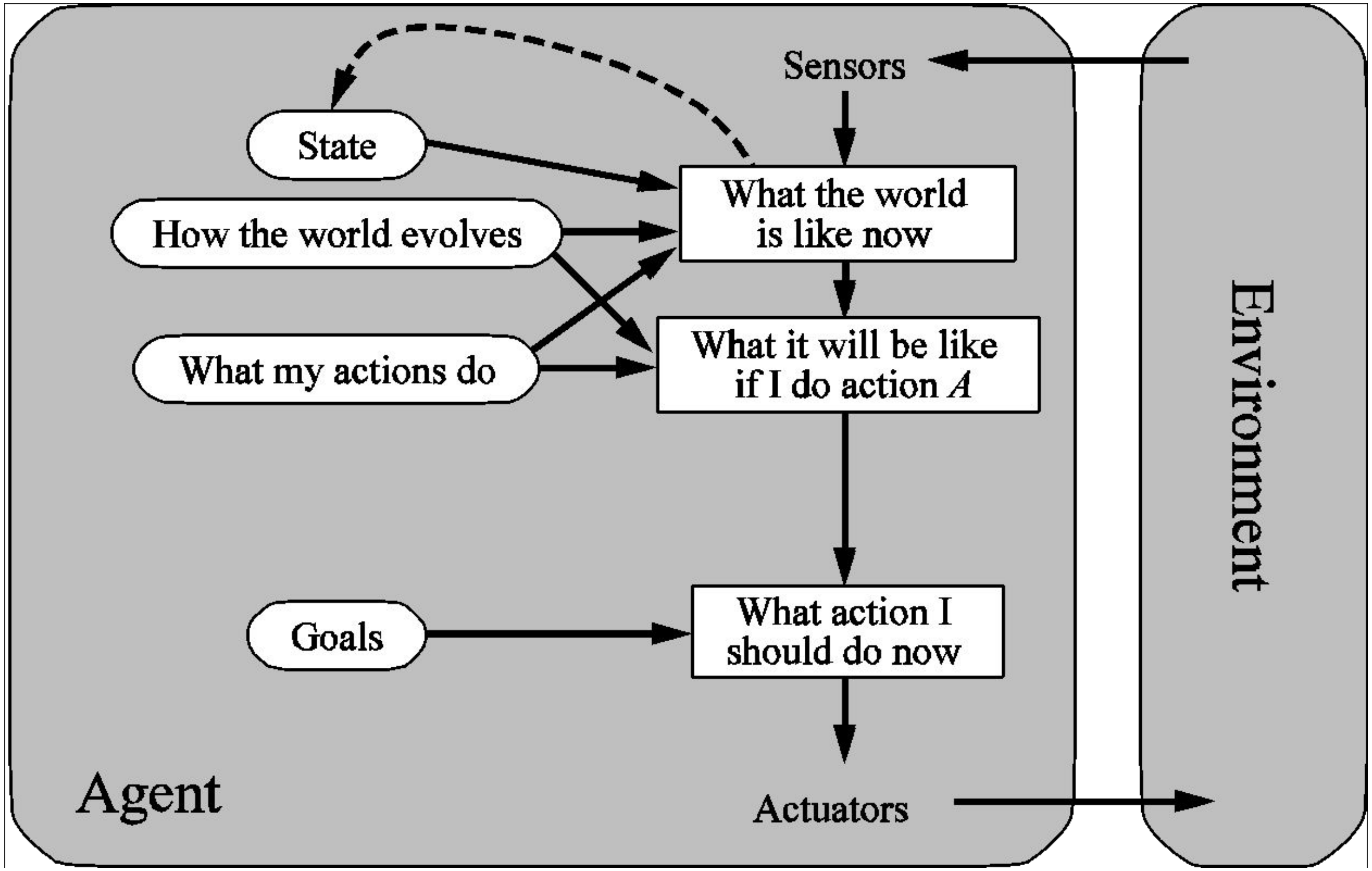
Architecture for a model-based agent



Model-based Agents

- Encode “internal state” of the world to remember the past as contained in earlier percepts.
- Needed because sensors do not usually give the entire state of the world at each input, so perception of the environment is captured over time. “State” is used to encode different "world states" that generate the same immediate percept.
- Requires ability to represent change in the world; one possibility is to represent just the latest state, but then can’t reason about hypothetical courses of action.

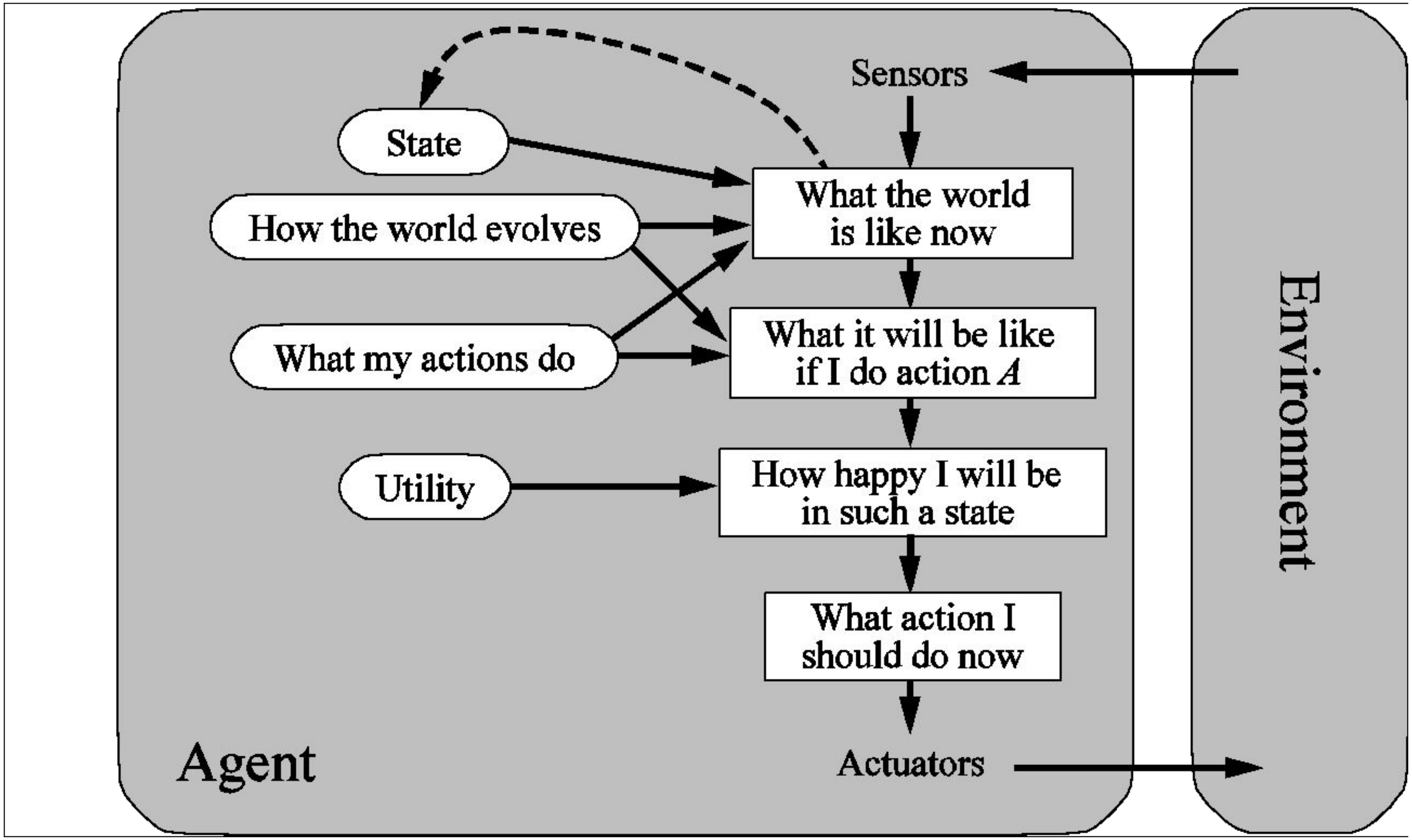
Architecture for goal-based agent



Goal-based agents

- Choose actions so as to achieve a (given or computed) goal.
- A goal is a description of a desirable situation.
- Keeping track of the current state is often not enough − need to add goals to decide which situations are good
- **Deliberative** instead of **reactive**.
- May have to consider long sequences of possible actions before deciding if goal is achieved – involves consideration of the future, *“what will happen if I do...?”*

Architecture for a complete utility-based agent



Utility-based agents

- When there are multiple possible alternatives, how to decide which one is best?
- A goal specifies a crude distinction between a happy and unhappy state, but often need a more general performance measure that describes "degree of happiness."
- Utility function **U: State** $\rightarrow$ **Reals** indicating a measure of success or happiness when at a given state.
- Allows decisions comparing choice between conflicting goals, and choice between likelihood of success and importance of goal (if achievement is uncertain).

Properties of Environments

- **Fully observable/Partially observable.**
  - If an agent's sensors give it access to the complete state of the environment needed to choose an action, the environment is **fully observable**.
  - Such environments are convenient, since the agent is freed from the task of keeping track of the changes in the environment.
- **Deterministic/Stochastic**.
  - An environment is **deterministic** if the next state of the environment is completely determined by the current state of the environment and the action of the agent; in a **stochastic** environment, there are multiple, unpredictable outcomes
  - In a fully observable, deterministic environment, the agent need not deal with uncertainty.

Properties of Environments II

- **Episodic/Sequential**.
  - An **episodic** environment means that subsequent episodes do not depend on what actions occurred in previous episodes.
  - In a **sequential** environment, the agent engages in a series of connected episodes.
  - Such environments do not require the agent to plan ahead.
- **Static/Dynamic.**
  - A **static** environment does not change while the agent is thinking.
  - The passage of time as an agent deliberates is irrelevant.
  - The agent doesn't need to observe the world during deliberation.

Properties of Environments III

- **Discrete/Continuous.**
  - If the number of distinct percepts and actions is limited, the environment is **discrete**, otherwise it is **continuous**.
- **Single agent/Multi-agent.**
  - If the environment contains other intelligent agents, the agent needs to be concerned about strategic, game-theoretic aspects of the environment (for either cooperative *or* competitive agents)
  - Most engineering environments don't have multi-agent properties, whereas most social and economic systems get their complexity from the interactions of (more or less) rational agents.

Characteristics of environments

	Fully observable?	Deterministic?	Episodic?	Static?	Discrete?	Single agent?
Crossword						
Poker						
Taxi driving						
Image analysis						
Medical diagnosis						

Characteristics of environments

	Fully observable?	Deterministic?	Episodic?	Static?	Discrete?	Single agent?
Crossword	yes	Yes	No	Yes	Yes	Yes
Poker						
Taxi driving						
Image Analysis						
Medical diagnosis						

Characteristics of environments

	Fully observable?	Deterministic?	Episodic?	Static?	Discrete?	Single agent?
Crossword	yes	Yes	No	Yes	Yes	Yes
Poker	**No**	**No**	**No**	**Yes**	**Yes**	**No**
Taxi driving						
Image Analysis						
Medical diagnosis						

Characteristics of environments

	Fully observable?	Deterministic?	Episodic?	Static?	Discrete?	Single agent?
Crossword	yes	Yes	No	Yes	Yes	Yes
Poker	No	No	No	Yes	Yes	No
Taxi driving	**No**	**No**	**No**	**No**	**No**	**No**
Image Analysis						
Medical diagnosis						

Characteristics of environments

	Fully observable?	Deterministic?	Episodic?	Static?	Discrete?	Single agent?
Crossword	yes	Yes	No	Yes	Yes	Yes
Poker	No	No	No	Yes	Yes	No
Taxi driving	No	No	No	No	No	No
Image Analysis	**Yes**	**Yes**	**Yes**	**Semi**	**No**	**Yes**
Medical diagnosis						

Characteristics of environments

	Fully observable?	Deterministic?	Episodic?	Static?	Discrete?	Single agent?
Crossword	yes	Yes	No	Yes	Yes	Yes
Poker	No	No	No	Yes	Yes	No
Taxi driving	No	No	No	No	No	No
Image Analysis	Yes	Yes	Yes	Semi	No	Yes
Medical diagnosis	**No**	**No**	**No**	**No**	**No**	**Yes**

→ Lots of real-world domains fall into the hardest case!

Summary

- An **agent** perceives and acts in an environment, has an architecture, and is implemented by an agent program.
- An **ideal agent** always chooses the action which maximizes its expected performance, given its percept sequence so far.
- An **autonomous agent** uses its own experience rather than built-in knowledge of the environment by the designer.
- An **agent program** maps from percept to action and updates its internal state.
  - **Reflex agents** respond immediately to percepts.
  - **Goal-based agents** act in order to achieve their goal(s).
  - **Utility-based agents** maximize their own utility function.
- **Representing knowledge** is important for successful agent design.
- The most challenging environments are partially observable, stochastic, sequential, dynamic, and continuous, and contain multiple intelligent agents.