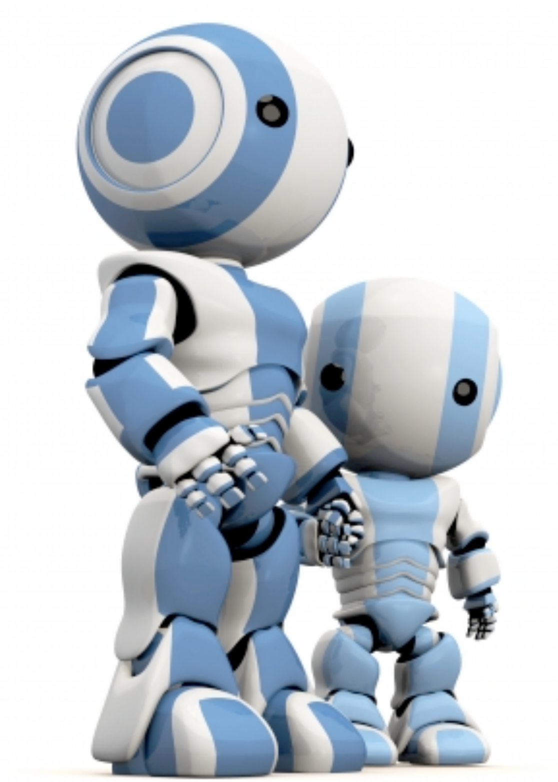
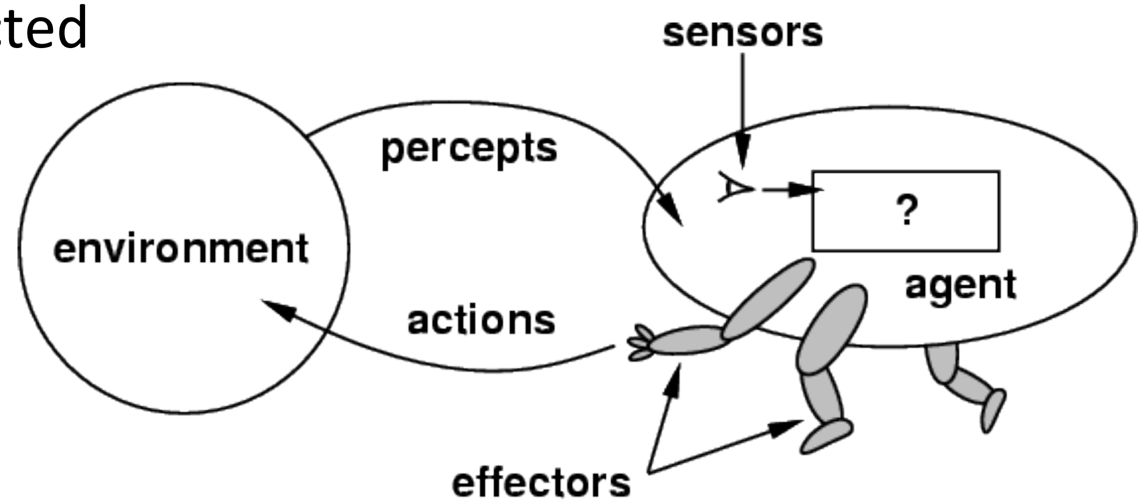


Intelligent Agents

Chapter 2

How do you design an intelligent agent?

- **Intelligent agents** perceive environment via **sensors** and act rationally on them with their **effectors**
- Discrete agents receive **percepts** one at a time, and map them to a sequence of discrete **actions**
- General properties
  - Reactive to the environment
  - Pro-active or goal-directed
  - Interacts with other agents through communication or via the environment
  - Autonomous



sensors/percepts and effectors/actions ?

Humans have

- **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), …
- **Effectors:** 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 specific example: Automated taxi driving system

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

Rationality

- Ideal **rational agents** should, for each possible percept sequence, act to maximize expected performance measure based on
  (1) the percept sequence, and
  (2) its built-in and acquired knowledge
- Rationality includes information gathering -- If you don’t know something, find out!
- Rationality $\rightarrow$ 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
- A system is not autonomous if it is guided by its designer according to a priori decisions
- An autonomous agent can always say “no”
- To survive, agents must have:
  - Enough built-in knowledge to survive
  - The ability to learn

Some agent types

simple

(0) Table-driven agents

– Use percept sequence/action table to find the next action. Implemented by a (large) **lookup table**

(1) Simple reflex agents

– Based on **condition-action rules**, implemented with an appropriate production system; stateless devices with no memory of past world states

(2) Agents with memory

– have **internal state** used to keep track of past states of the world

(3) Agents with goals

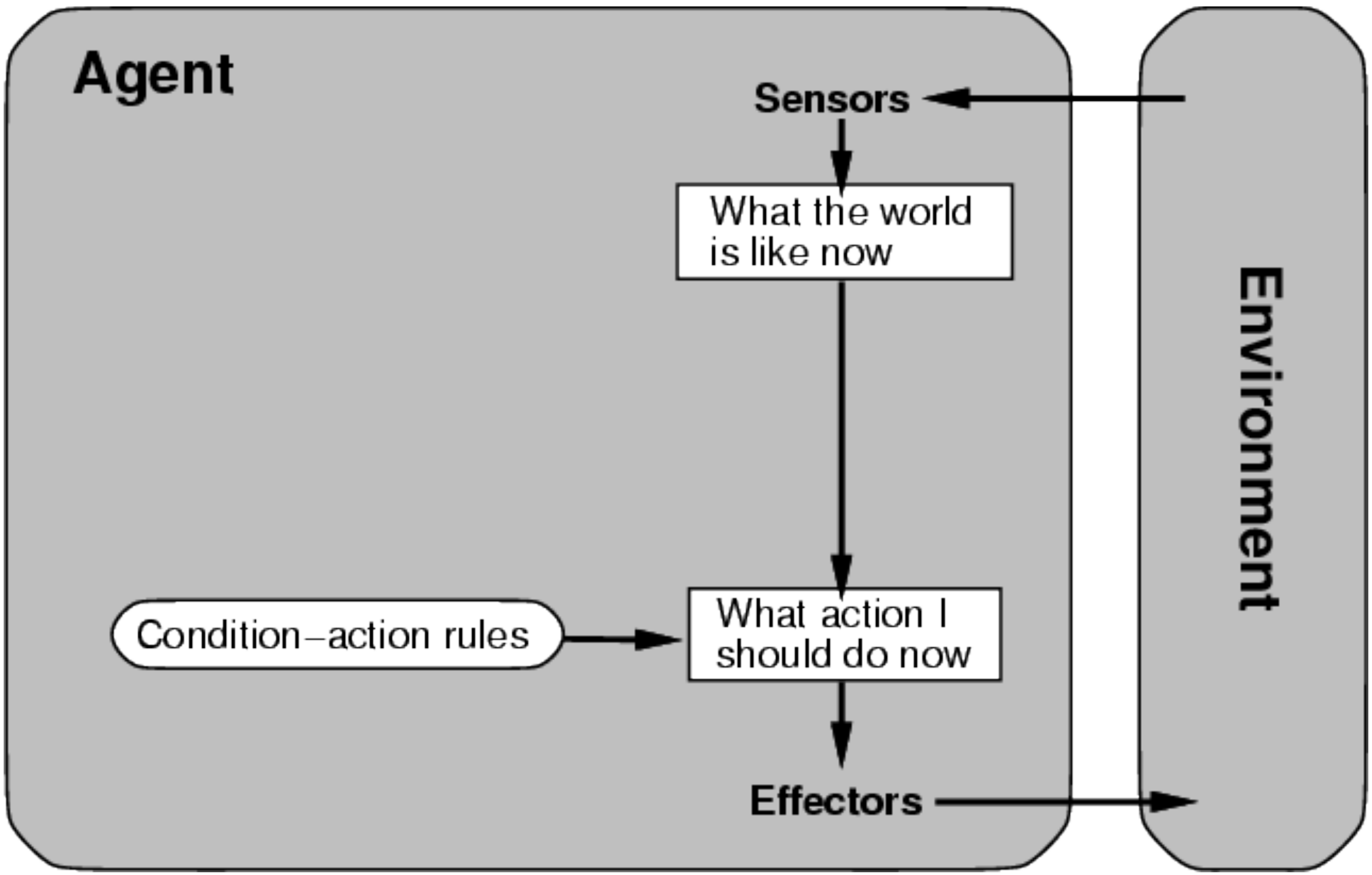
– Agents with a state and **goal information** describing desirable situations. Agents of this kind take future events into consideration.

(4) Utility-based agents

– base decisions on **classic axiomatic utility theory** in order to act rationally

complex

(0/1) Table-driven/reflex agent architecture



(0) Table-driven agents

Table lookup of percept-action pairs mapping from every possible perceived state to optimal action for it

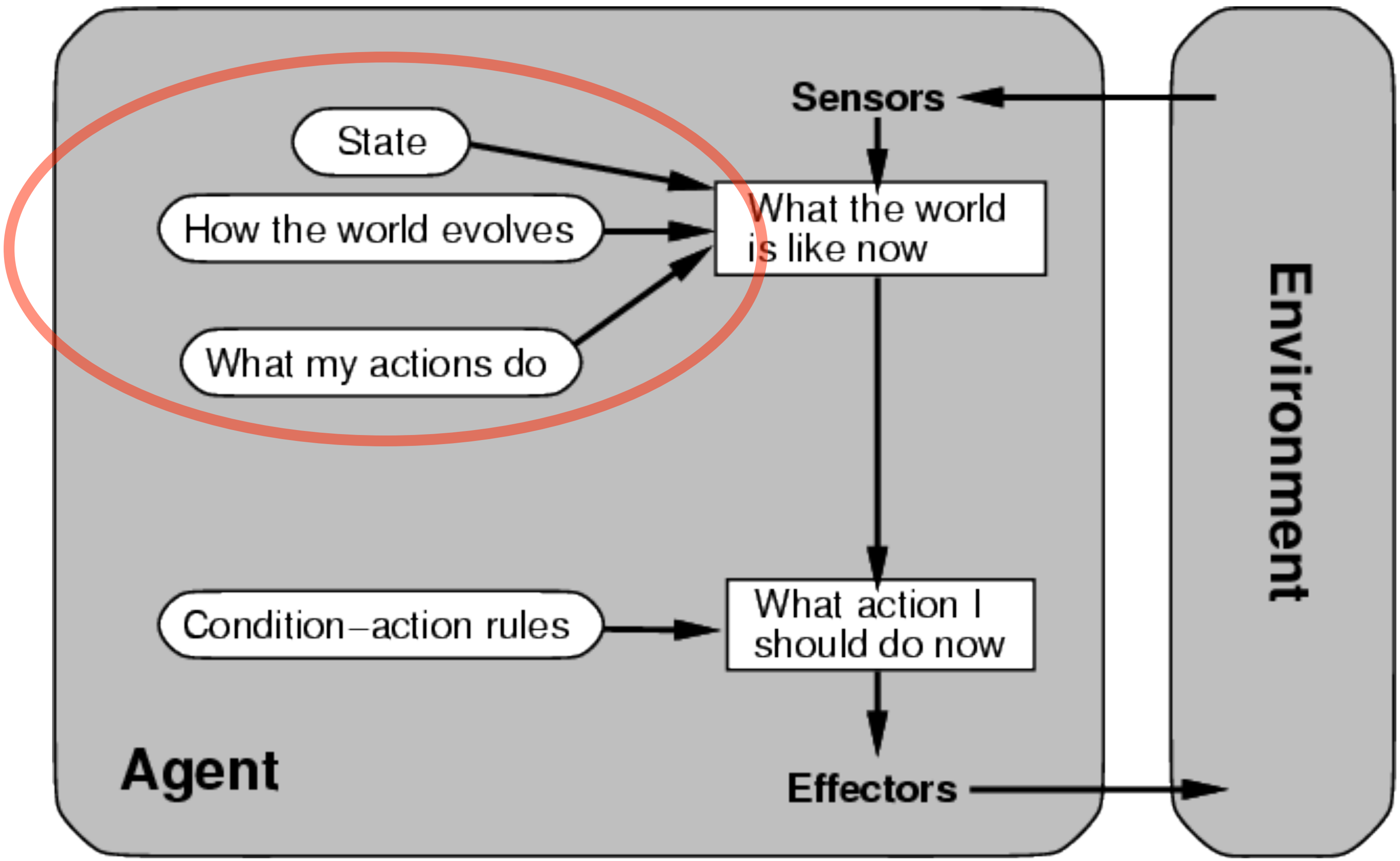
Problems:

- 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; entire table must be updated if changes occur
- Looping: Can't make actions conditional on previous actions/states

(1) Simple reflex agents

- **Rule-based reasoning** maps percepts to optimal action; each rule handles collection of perceived states (aka reactive agents)
- **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 environment; collection of rules must be updated if changes occur
  - Still can't condition actions on previous state
  - Difficult to engineer if the number of rules is large due to conflicts

(2) Architecture for an agent with memory

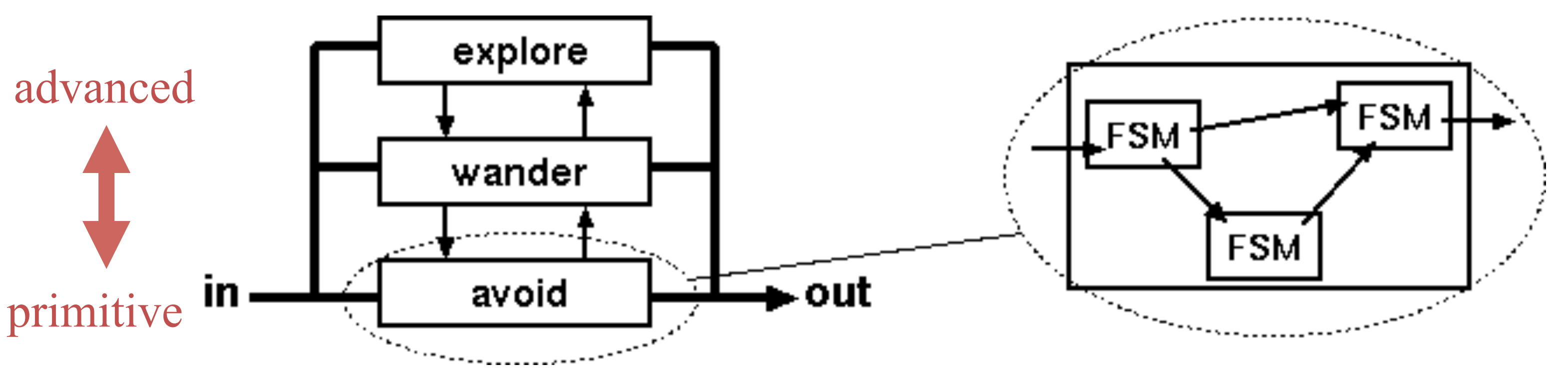


(2) Agents with memory

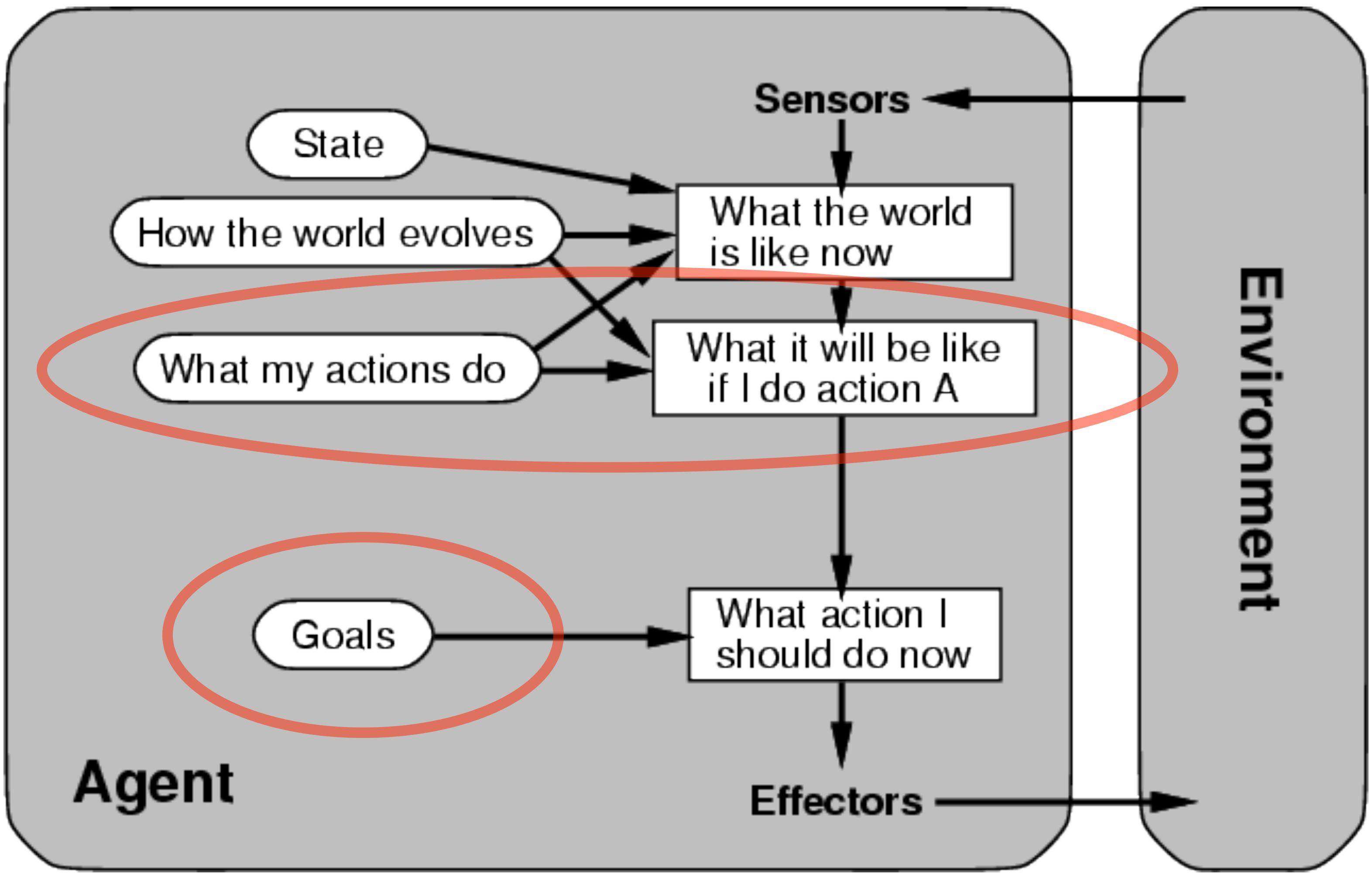
- Encode *internal state* of world to remember past as contained in earlier percepts
  – Note: sensors don't usually give the entire world state at each input, so environment perception is *captured over time*.
  – *State* is used to encode different "world states" that generate the same immediate percept
- Requires *representing change* in the world
  – one possibility is representing just the latest state, but then can't reason about hypothetical courses of action

(2) Brooks' s Subsumption Architecture

- Rod Brooks, director of MIT AI Lab
- Main idea: build complex, intelligent robots by decomposing behaviors into hierarchy of skills, each completely defining a complete percept-action cycle for one very specific task.
- Examples: avoiding contact, wandering, exploring, recognizing doorways, etc.
- Each behavior modeled by a finite-state machine with a few states (tho each may correspond to complex function/module).
- Behaviors are loosely coupled, asynchronous interactions.



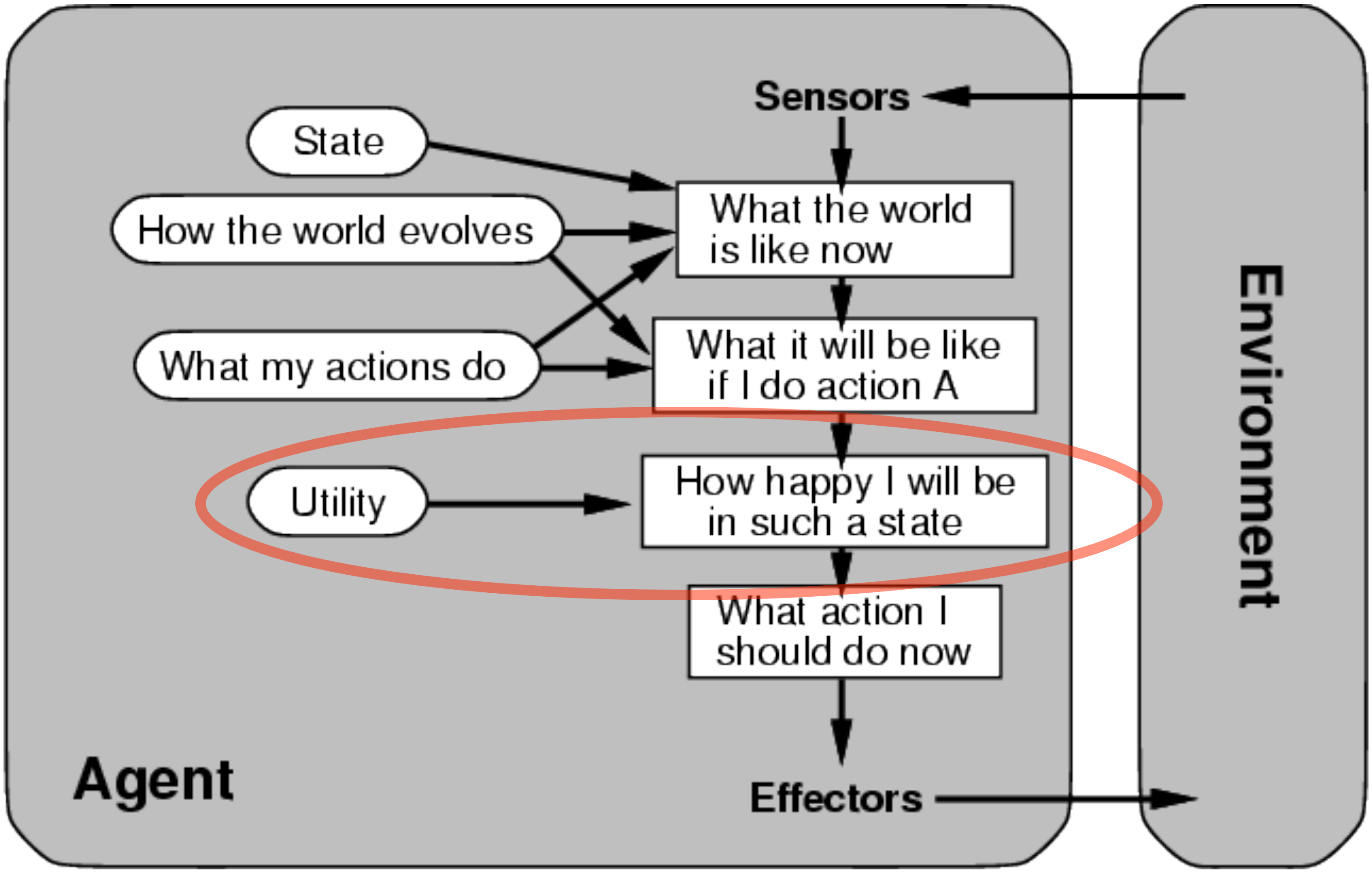
(3) Architecture for goal-based agent



(3) Goal-based agents

- **Deliberative** instead of **reactive**
- Choose actions so as to achieve a goal (given or computed)
- Goal is a description of a desirable situation
- Keeping track of current state often not enough - must add goals to decide which situations are good
- Achieving goal may require long action sequence
  - Must model action consequences: "*what will happen if I do...?*"
  - Planning

(4) a complete utility-based agent



(4) Utility-based agents

- For multiple possible alternatives, how to decide which one is best?
- Goals give a crude distinction between happy and unhappy states, but often need a performance measure that describes *degree*
- Utility function **U: State→Reals** gives measure of success/happiness for given state
- Allows decisions comparing choices between conflicting goals and likelihood of success and importance of goal (if achievement **uncertain**)

Properties of Environments

- **Fully/Partially observable**
  - If agent's sensors give it complete state of environment needed to choose action, the environment is **fully observable**
  - Such environments are convenient, freeing agents from keeping track of the environment's changes
- **Deterministic/Stochastic**
  - Environment is **deterministic** if next state is completely determined by the current state and the agent's action
  - in a **stochastic** (i.e., non-deterministic) environment, there are multiple, unpredictable outcomes
- In fully observable, deterministic environments agents need not deal with uncertainty

Properties of Environments

- **Episodic/Sequential**
  - In **episodic** environments subsequent episodes don't depend on actions in previous episodes
  - In **sequential** environments the agent engages in a series of connected episodes
  - Episodic environments don't require the agent to plan ahead
- **Static/Dynamic**
  - **Static** environments doesn't change as the agent is thinking
  - The passage of time as an agent deliberates is irrelevant
  - The agent needn't 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/Multiagent**
  - In an environment with other agents, the agent must consider strategic, game-theoretic aspects of the environment (for either cooperative *or* competitive agents)
  - Most engineering environments don't have multiagent 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?
Solitaire						
Backgammon						
Taxi driving						
Internet shopping						
Medical diagnosis						

A **Yes** in a cell means that aspect is simpler; a **No** more complex

Characteristics of environments

	Fully observable	Deterministic	Episodic	Static	Discrete?	Single agent?
Solitaire	**No**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**
Backgammon						
Taxi driving						
Internet shopping						
Medical diagnosis						

A **Yes** in a cell means that aspect is simpler; a **No** more complex

Characteristics of environments

	Fully observable?	Deterministic?	Episodic?	Static?	Discrete?	Single agent?
Solitaire	No	Yes	Yes	Yes	Yes	Yes
Backgammon	**Yes**	**No**	**No**	**Yes**	**Yes**	**No**
Taxi driving						
Internet shopping						
Medical diagnosis						

A **Yes** in a cell means that aspect is simpler; a **No** more complex

Characteristics of environments

	Fully observable?	Deterministic?	Episodic?	Static?	Discrete?	Single agent?
Solitaire	No	Yes	Yes	Yes	Yes	Yes
Backgammon	Yes	No	No	Yes	Yes	No
Taxi driving	**No**	**No**	**No**	**No**	**No**	**No**
Internet shopping						
Medical diagnosis						

A **Yes** in a cell means that aspect is simpler; a **No** more complex

Characteristics of environments

	Fully observable?	Deterministic?	Episodic?	Static?	Discrete?	Single agent?
Solitaire	No	Yes	Yes	Yes	Yes	Yes
Backgammon	Yes	No	No	Yes	Yes	No
Taxi driving	No	No	No	No	No	No
Internet shopping	**No**	**No**	**No**	**No**	**Yes**	**No**
Medical diagnosis						

A **Yes** in a cell means that aspect is simpler; a **No** more complex

Characteristics of environments

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

	Fully observable?	Deterministic?	Episodic?	Static?	Discrete?	Single agent?
Solitaire	No	Yes	Yes	Yes	Yes	Yes
Backgammon	Yes	No	No	Yes	Yes	No
Taxi driving	No	No	No	No	No	No
Internet shopping	No	No	No	No	Yes	No
Medical diagnosis	**No**	**No**	**No**	**No**	**No**	**Yes**

A **Yes** in a cell means that aspect is simpler; a **No** more complex

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

Summary

- An **agent program** maps percepts to actions and updates its internal state
  - **Reflex agents** respond immediately to percepts
  - **Goal-based agents** act to achieve their goal(s)
  - **Utility-based agents** maximize their own utility function
- **Representing knowledge** is important for good agent design
- Most challenging environments are partially observable, stochastic, sequential, dynamic, and continuous, and contain multiple intelligent agents

Summary

- Not every AI problem is a good fit for an agent model
- Playing Solitaire, for example, probably is not
- Nor are many AI tasks you might be asked to solve:
  - Classify movie reviews as negative, neutral or positive
  - Locate faces of people in an image
  - Learn preferred thermostat settings for each hour of each day of a week