Intelligent Agents

Chapter 2

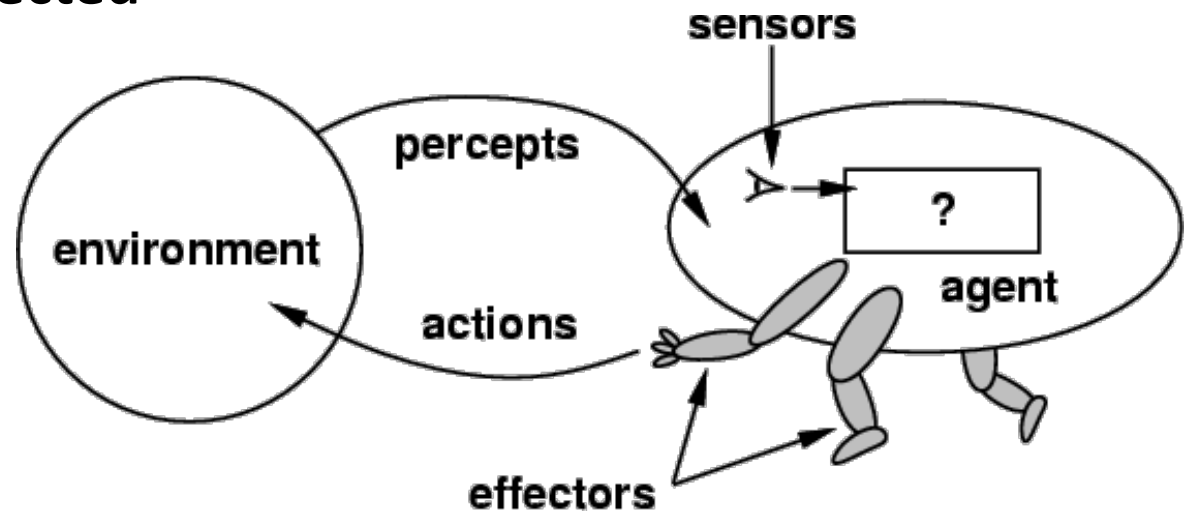
Agents and AI

- Much of our text is organized around the concept of **autonomous agents**
- It defines an agent as

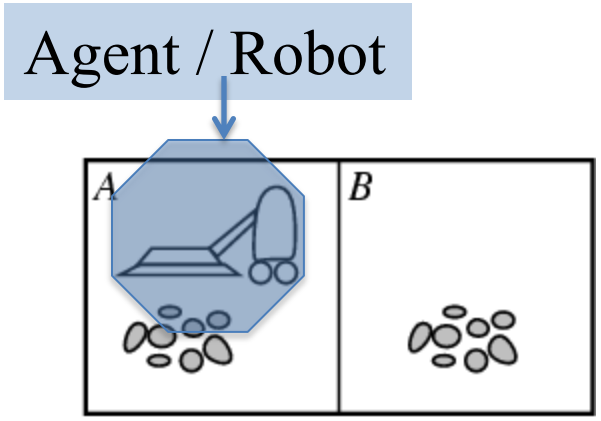
  anything that can be viewed as perceiving its **environment** through **sensors** and acting upon that environment through **actuators**
- Not all AI problems or tasks need or fit this concept, but it's quite general
- We'll explore the concept today

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 an intelligent agent should have
  - **Reactive** to the environment
  - Pro-active or **goal-directed**
  - **Interacts** with other agents through communication or via the environment
  - **Autonomous**



Example: simple vacuum agent



- **Percepts:** location and contents, e.g., [A, Dirty]
- **Actions:** *Left*, *Right*, *Suck*, *NoOp*

iRobot Roomba® 400
Vacuum Cleaning Robot

iRobot Corporation

Founder Rodney Brooks (MIT)

- Powerful suction and rotating brushes
- Automatically navigates for best cleaning coverage
- Cleans under and around furniture, into corners and along wall edges
- Self-adjusts from carpets to hard floors and back again
- Automatically avoids stairs, drop-offs and off-limit areas
- Simple to use— just press the Clean button and Roomba does the rest

Example: autonomous taxi

- **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 input, act to **maximize expected performance** measures based on
  (1) percept sequence, and
  (2) built-in and acquired knowledge
- Rationality → Need a ***performance measure*** to say how well a task has been achieved
- Types of performance measures: false alarm (false positive) & false dismissal (false negative) rates, speed, resources required, effect on environment, money earned, …

Omniscience and learning

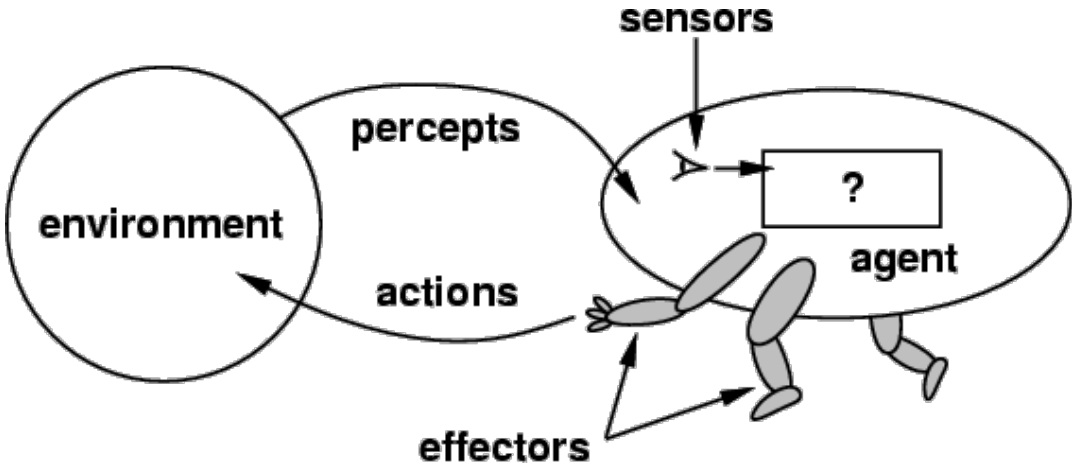
- Rational agents aren't expected to know everything
- But they are expected to use what they do know effectively
- Rationality includes ***information gathering*** -- If you don't know something that might be useful, find out!
- Rationality also can exploit ***learning*** – making generalization from past experience to fill in missing information

Autonomy

- A system is autonomous to extent that its behavior is determined by its experience
- A system isn't autonomous if guided by its designer according to *a priori* *decisions*
- An autonomous agent can always say "no"
- To be effective, agents must have:
  - Enough built-in knowledge to perform minimally
  - The ability to learn, to maintain, or improve its performance

PEAS model for agents

- A common framework for thinking about agents uses the PEAS model
- which stands for
  - Performance
  - Environment
  - Actuators
  - Sensors



PEAS model: Taxi Agent

Performance, **E**nvironment, **A**ctuators, **S**ensors

Example of an autonomous taxi

Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi driver	Safe, fast, legal, comfortable trip, maximize profits, minimize impact on other road users	Roads, other traffic, police, pedestrians, customers, weather	Steering, accelerator, brake, signal, horn, display, speech	Cameras, radar, speedometer, GPS, engine sensors, accelerometer, microphones, touchscreen

PEAS model: Other Agents

Agent Type	Performance Measure	Environment	Actuators	Sensors
Medical diagnosis system	Healthy patient, reduced costs	Patient, hospital, staff	Display of questions, tests, diagnoses, treatments	Touchscreen/voice entry of symptoms and findings
Satellite image analysis system	Correct categorization of objects, terrain	Orbiting satellite, downlink, weather	Display of scene categorization	High-resolution digital camera
Part-picking robot	Percentage of parts in correct bins	Conveyor belt with parts; bins	Jointed arm and hand	Camera, tactile and joint angle sensors
Refinery controller	Purity, yield, safety	Refinery, raw materials, operators	Valves, pumps, heaters, stirrers, displays	Temperature, pressure, flow, chemical sensors
Interactive English tutor	Student's score on test	Set of students, testing agency	Display of exercises, feedback, speech	Keyboard entry, voice

Note: agents need not be physical: 3 of the 5 are not

Some agent types

simple

complex

(0) Table-driven agents

Use percept sequence/action table to find next action. Implemented by a **lookup table**

(1) Simple reflex agents

Based on **condition-action rules**, stateless devices with no memory of past world states

(2) Agents with memory

have **represent states** and keep track of past world states

(3) Agents with goals

Have a state and **goal information** describing desirable situations; can take future events into consideration

(4) Utility-based agents

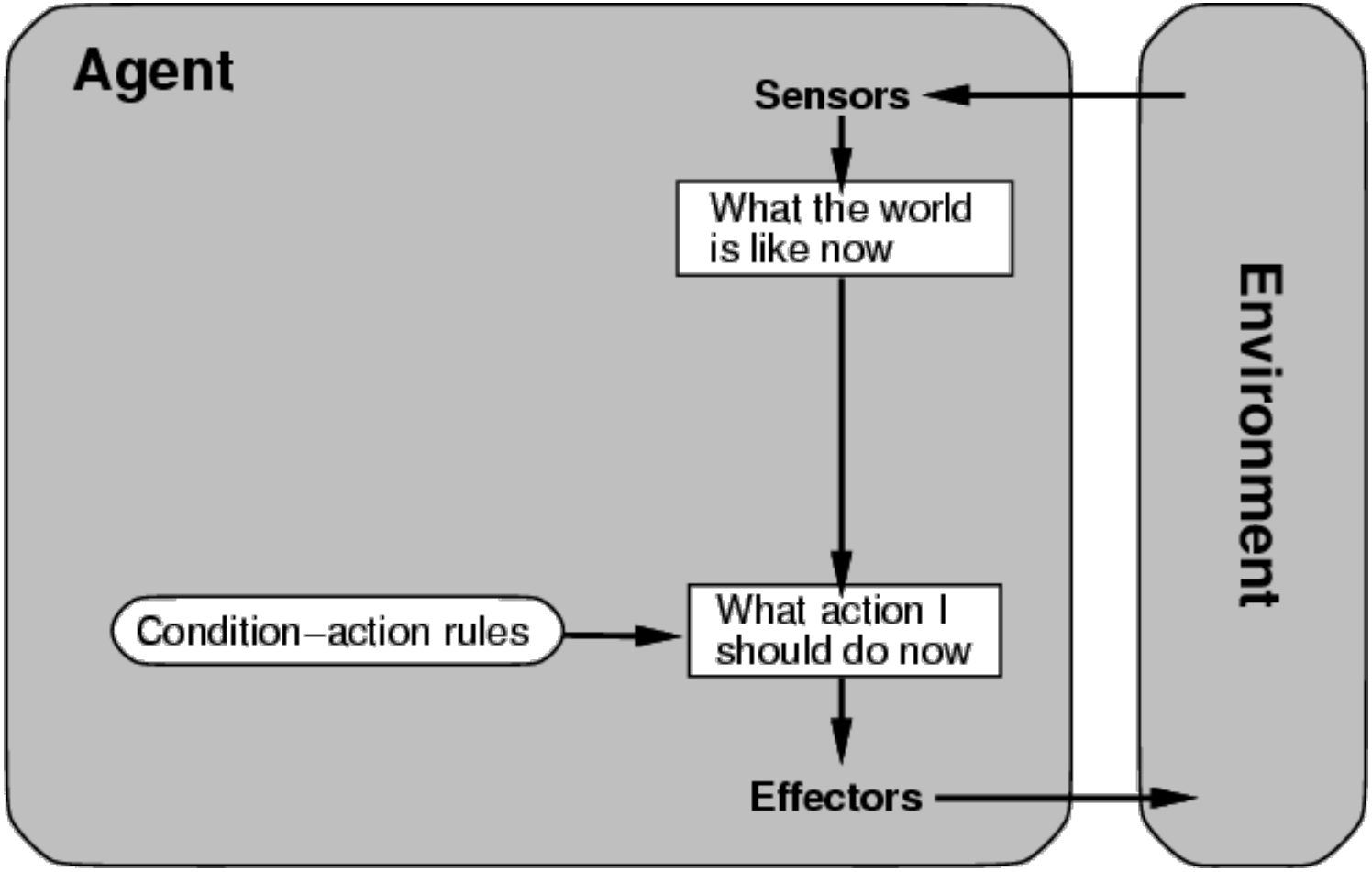
base decisions on **utility theory** in order to act rationally

(5) Learning agents

base decisions on **models learned** and updated through experience

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

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



(0) Table-driven agents

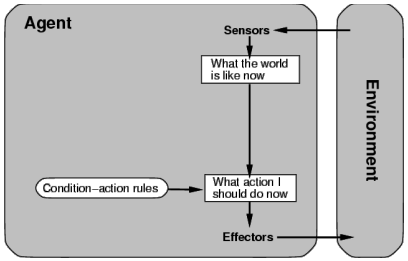


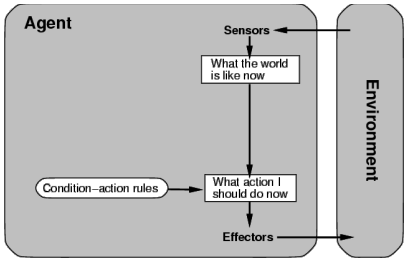
Table lookup of percept-action pairs mapping from every possible perceived state to optimal action for it

Problems:

- Table may be too big to generate and to store (e.g., chess has about 10^{120} states)
- No knowledge of non-perceptual parts of the current state (e.g., desirability of states)
- 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

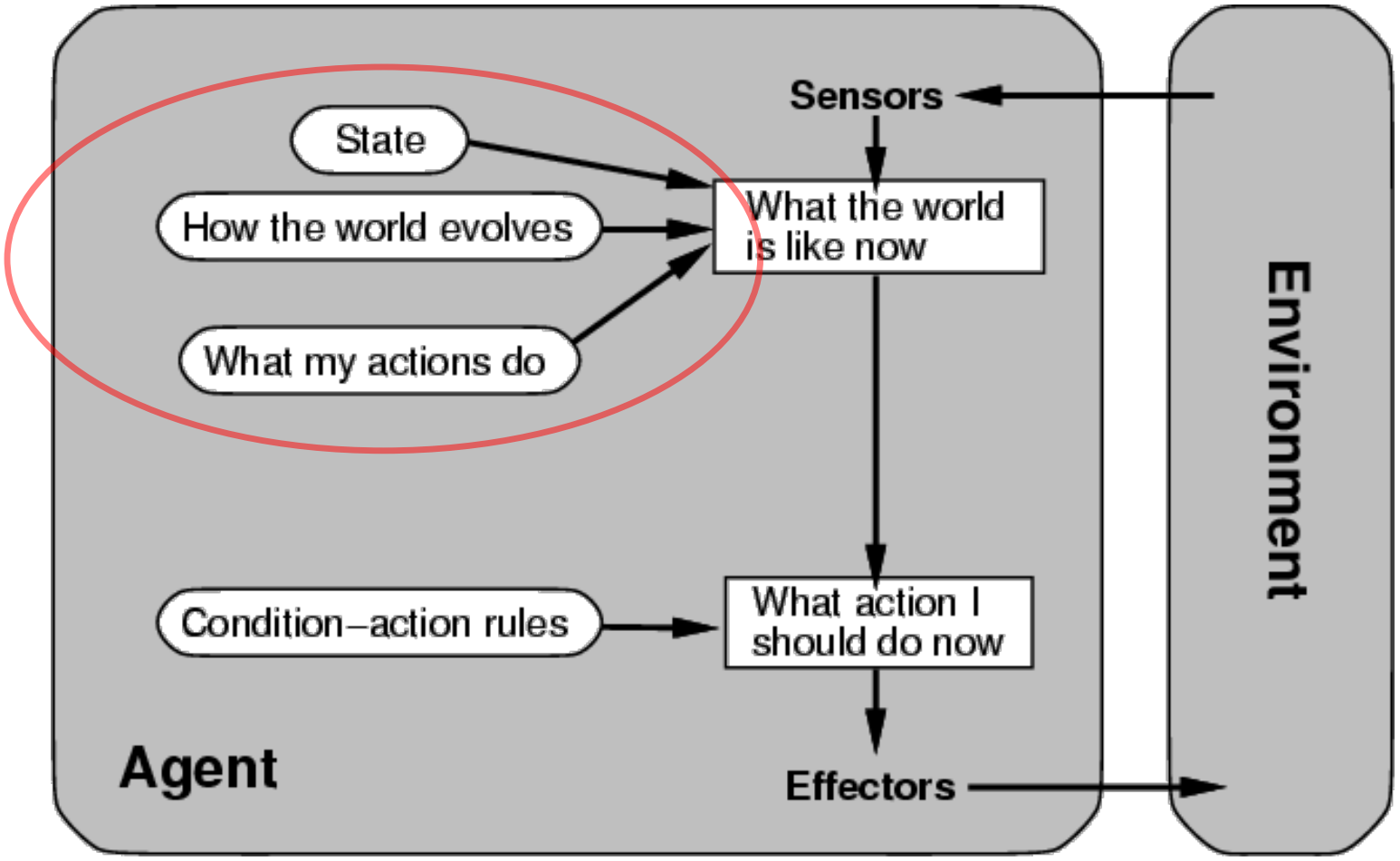
aka reactive agents



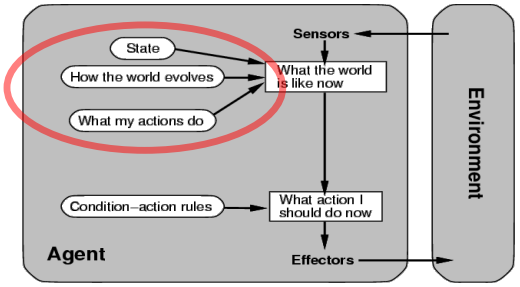
- **Rule-based reasoning** maps percepts to optimal action; each rule may handle a collection of perceived states
- **Problems remain, however**
  - Usually too big to generate and to store
  - No knowledge of non-perceptual parts of state
  - Not adaptive to changes in environment; collection of rules must be updated if changes occur
  - Can't condition actions on previous state
  - Difficult to engineer if number of rules large due to conflicts

(2) Architecture for an agent with memory

internal state keeps track of past states of the world



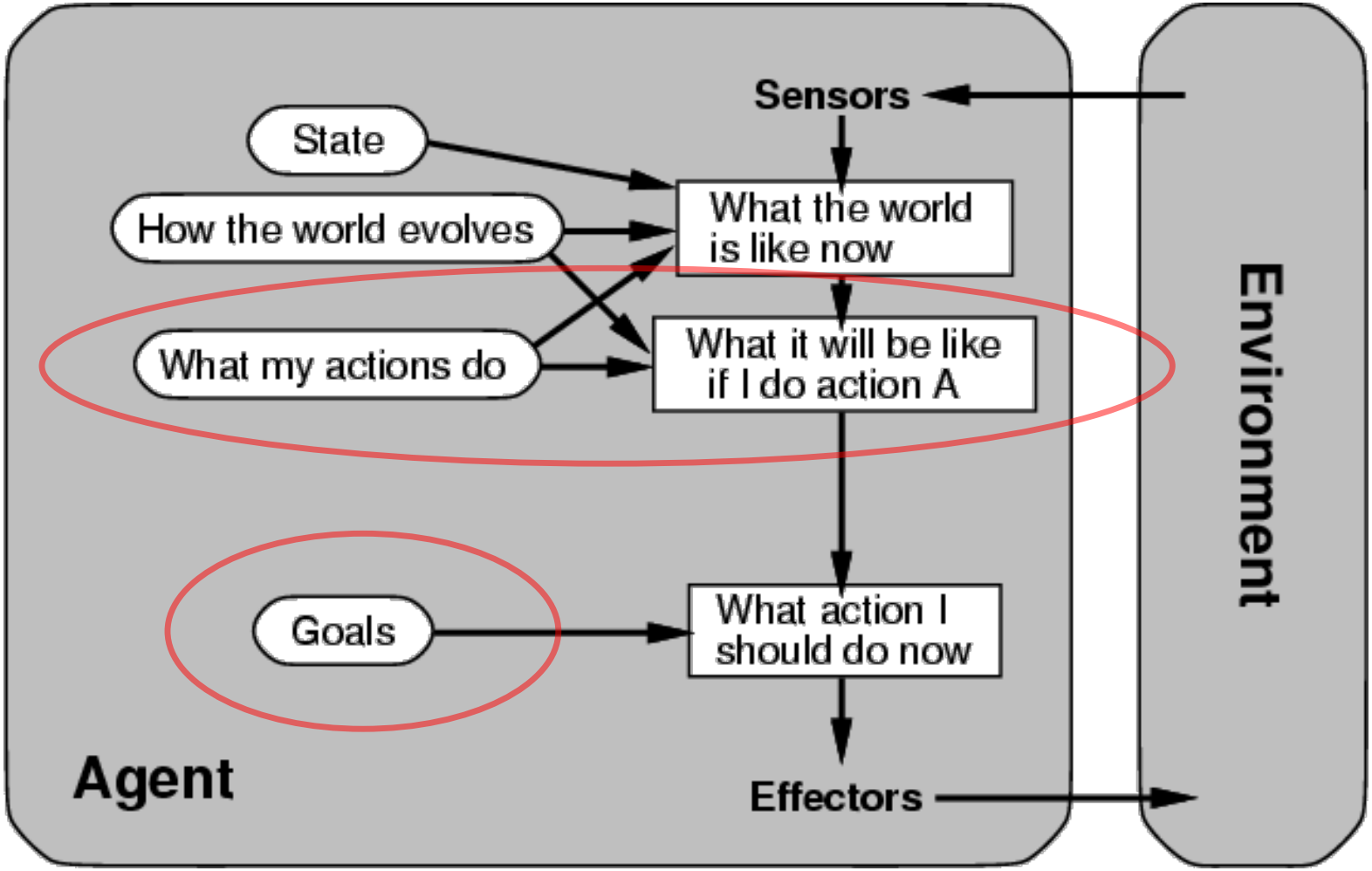
(2) Agents with memory



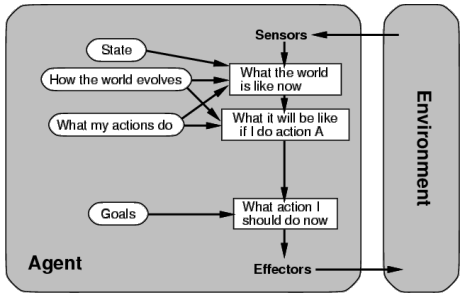
- Encode ***internal state*** of world to remember past as modeled by earlier percepts
  - Sensors often don't give entire world state at each input, so environment ***captured over time***
  - *State* can encode different "world states" that generate the same immediate percept
- Can prevent looping
- Requires ***representing change*** in the world
  - Might represent just latest state, but then can't reason about hypothetical courses of action

(3) Architecture for goal-based agent

state and **goal information** describe desirable situations allowing agent to take future events into consideration



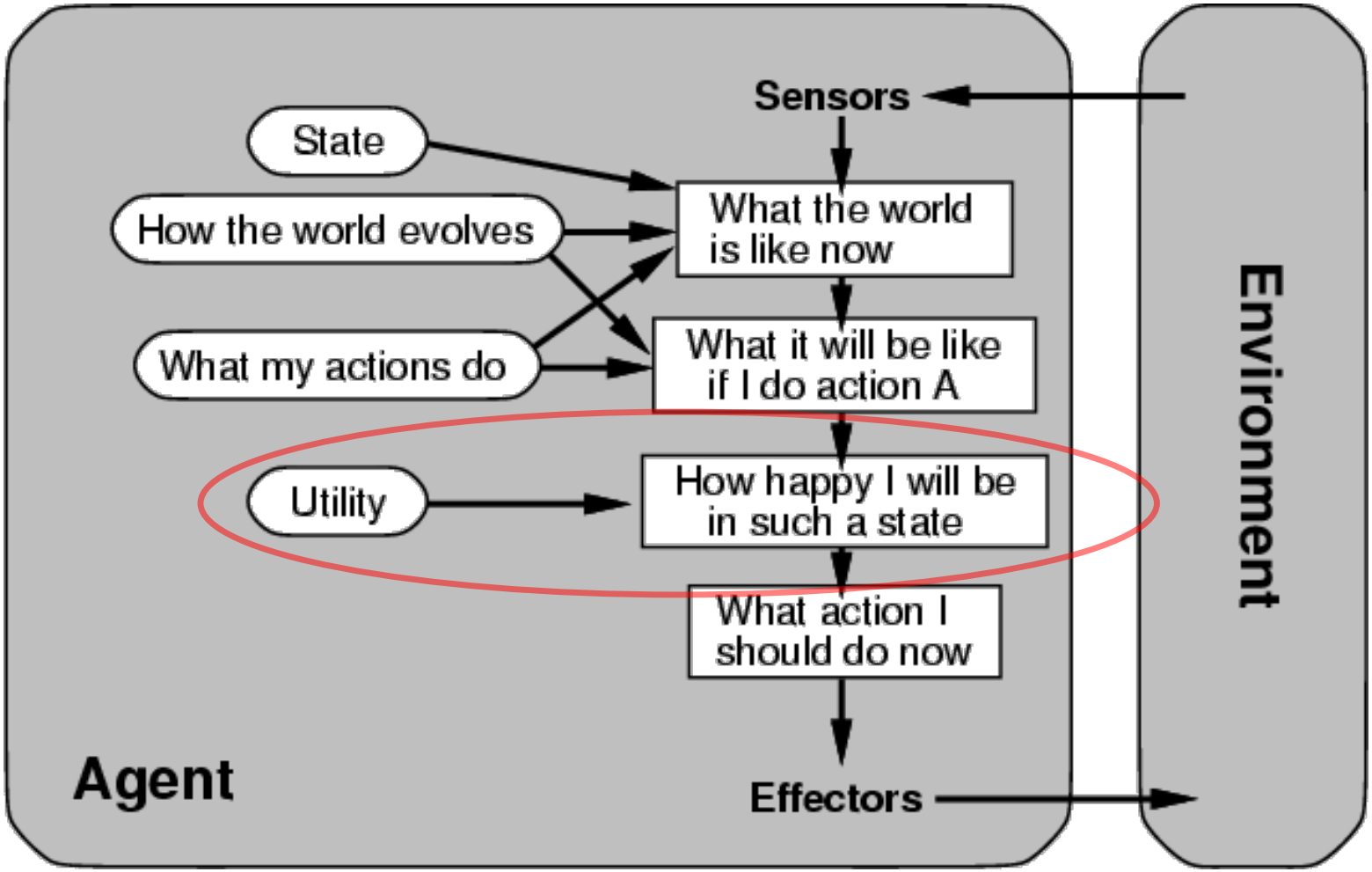
(3) Goal-based agents



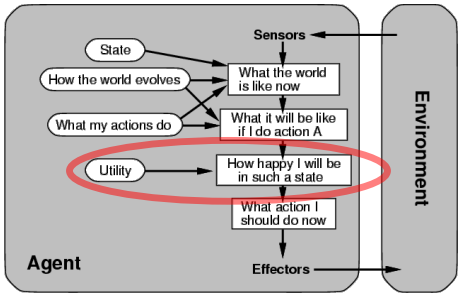
- **Deliberative** instead of **reactive**
- Choose actions to achieve a **goal**
- Goal: description of a desirable situation
- Keeping track of current state often not enough; add goals to decide which situations are good
- Achieving goal may require an **action sequence**
  - Model action **consequences**: *"what if I do...?"*
  - May use ***planning*** algorithms to produce action sequences

(4) Complete utility-based agent

base decisions on utility theory in order to act rationally

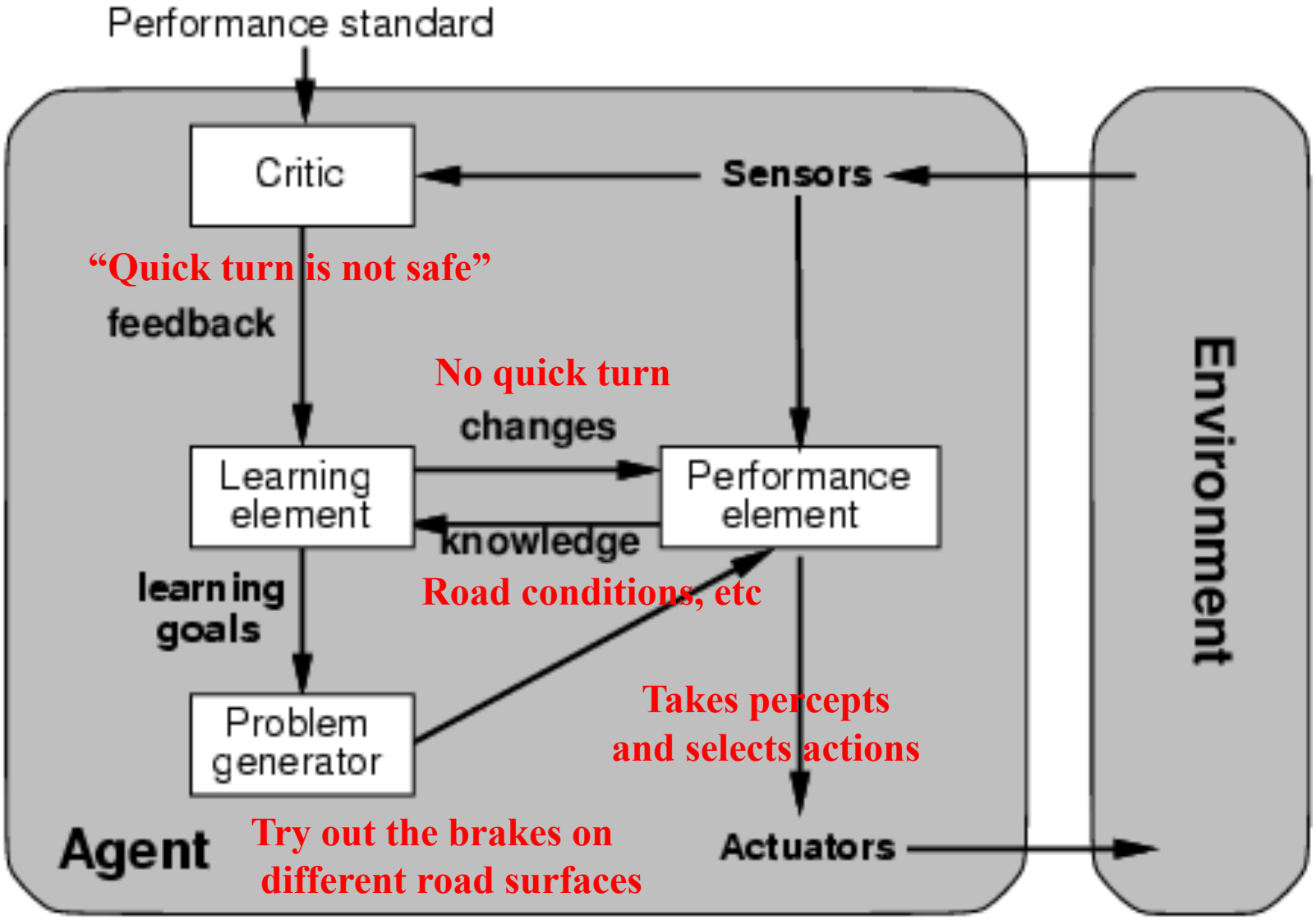


(4) Utility-based agents



- Given multiple possible alternatives, how to decide which is best?
- Goals give distinction between good/bad states, but often need a measure for ***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**)

(5) Learning agent



More complicated when agent must **learn utility information**, e.g., via **reinforcement learning** based on action payoff

Problem Environments

- Characteristics of the problem environment have a big impact on an agent's requirements
- Consider developing an agent to
  - Solve a crossword puzzle, vs.
  - Drive a taxi
- Identifying some general problem characteristics helps us understand it and possible approaches to solving it

Properties of Environments (1)

- **Fully/Partially observable**
  - Agent's sensors give complete state of environment needed to choose action: 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 current state and agent's action
  - **Stochastic** (i.e., non-deterministic) environments have multiple, unpredictable outcomes
- In fully observable, deterministic environments agents need not deal with uncertainty

Properties of Environments (2)

- **Episodic/Sequential**
  - In **episodic** environments subsequent episodes don't depend on actions in previous episodes
  - In **sequential** environments agent engages in a series of connected episodes
  - Episodic environments don't require agent to plan ahead
- **Static/Dynamic**
  - **Static** environments don't change as an agent is thinking
  - The passage of time as agent deliberates is irrelevant
  - The agent needn't observe world during deliberation

Properties of Environments (3)

- **Discrete/Continuous**
  - If number of distinct percepts and actions is limited, environment is **discrete**, otherwise it's **continuous**
- **Single agent/Multiagent**
  - In environments with multiple agents, each agent must consider strategic, game-theoretic aspects, for either **cooperative** *or* **competitive agents**
  - Most engineering environments don't have multiagent properties, whereas most social and economic systems get their complexity from 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**	**?**	**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	?	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	?	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	?	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

- **Agents** perceive and act in an environment, have an architecture and are implemented by an agent program
- **Ideal agents** chose actions to maximize their expected performance, given percept sequence observed so far
- **Autonomous agents** use own experience rather than built-in knowledge of environment by designer

Summary

- **Agent programs** map percepts to actions and update their internal state
  - **Reflex** agents respond immediately to percepts
  - **Goal-based** agents act to achieve their goal(s)
  - **Utility-based** agents maximize their utility function
- **Representing knowledge** is important for good agent design
- Challenging environments are partially observable, stochastic, sequential, dynamic, and continuous and contain multiple agents

Summary

- Not all AI problems a good fit for or require a full agent model, e.g., playing solitaire
- Nor are many AI tasks you might need to solve:
  - Classify movie reviews as negative, neutral or positive
  - Locate faces of people in an image
  - Develop an efficient theorem prover
  - Learn preferred thermostat settings in a home for each hour of each day of a week