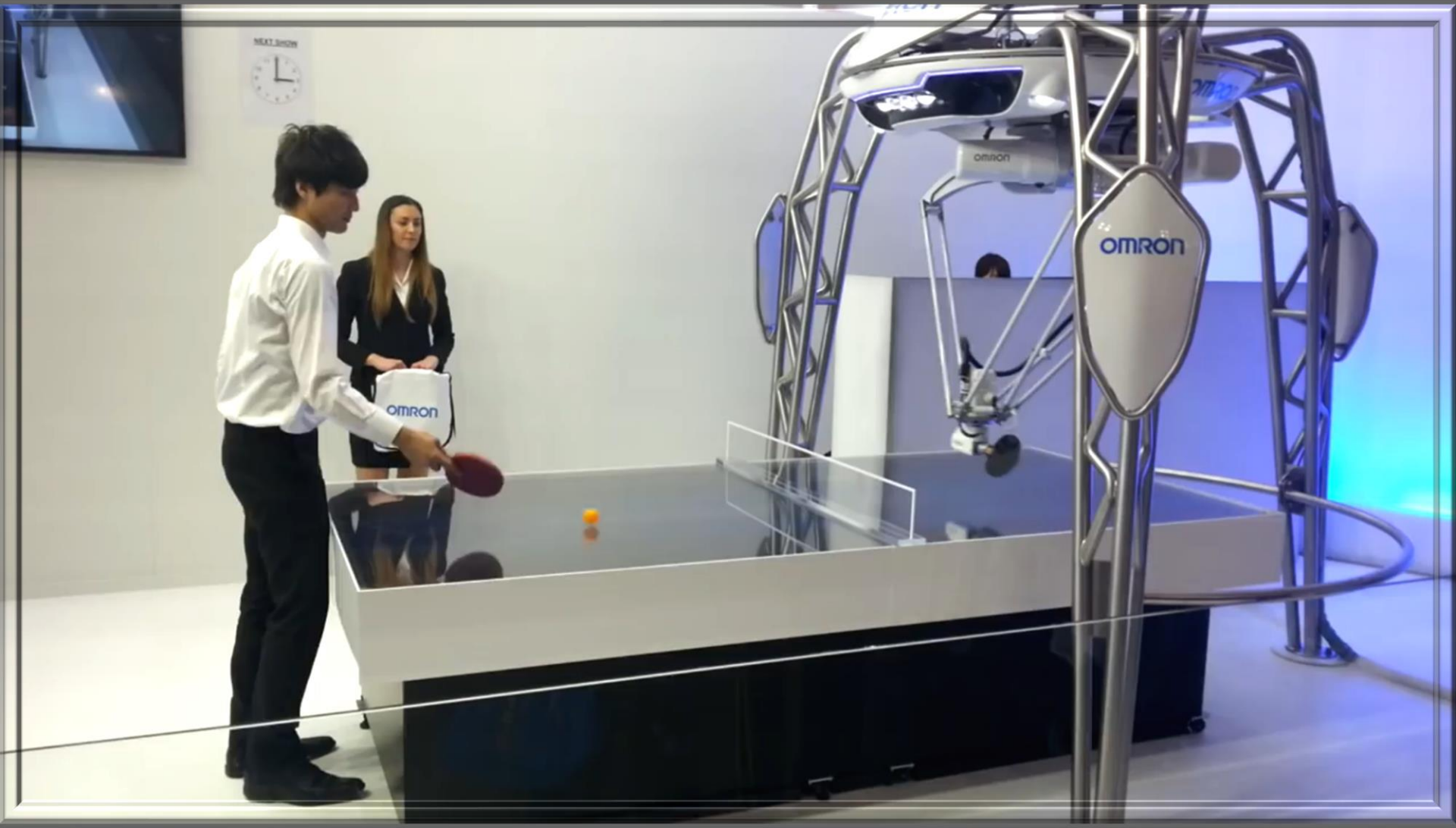
Lecture #3

Outline

- Agents and environments
- Rationality
- PEAS (Performance Measure, Environment, Actuators, Sensors)
- Environment types
- Agent types

Question?

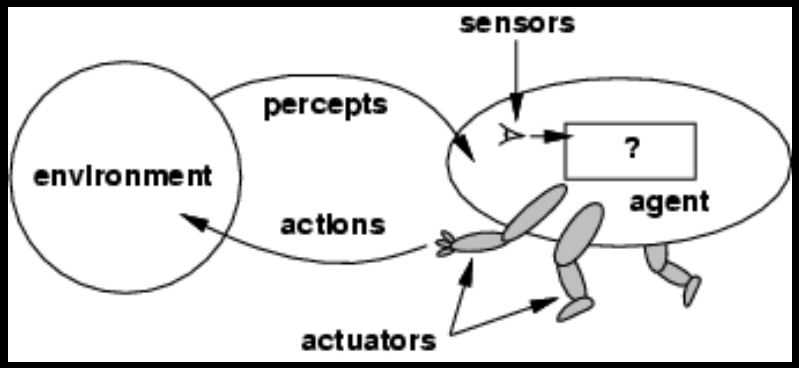


Driving in downtown Cairo is too unpredictable

- An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators
- Human agent: eyes, ears, and other organs for sensors; hands,
- legs, mouth, and other body parts for actuators
- Robotic agent: cameras and infrared range finders for sensors;

various motors for actuators

Agents and environments



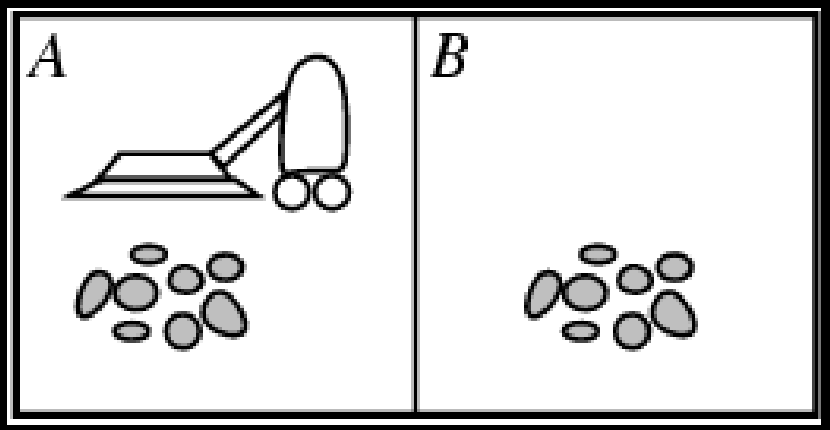
The agent function maps from percept histories to actions:

$$[f: \mathcal{P}^* \rightarrow \mathcal{A}]$$

The agent program runs on the physical architecture to produce f

agent = architecture + program

Vacuum-cleaner world



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

function 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

E.g., **performance measure** of a vacuum-cleaner agent could be amount of dirt cleaned up, amount of time taken, amount of electricity consumed, amount of noise generated, etc.

Rational agents

- Rational Agent: For each possible percept sequence, a rational agent should select an action that is expected to maximize its performance measure.
-

An agent acts intelligently when

- what it does is appropriate for its functions and its goals,
- it is flexible to changing environments and changing goals,
- it learns from experience, and
- it makes appropriate choices given its perceptual and computational limitations.
- An agent typically cannot observe the state of the world directly; it has only a finite memory and it does not have unlimited time to act.

- A **computational** agent is an agent whose decisions about its actions can be explained in terms of computation.
- the decision can be broken down into primitive operation that can be implemented in a physical device.

Knowledge is the information about a domain that can be used to solve problems in that domain.

Rational agents

- Rationality is distinct from omniscience (all-knowing with infinite knowledge)
-
- Agents can perform actions in order to modify future percepts so as to obtain useful information (information gathering, exploration)
-
- An agent is autonomous if its behavior is determined by its own experience (with ability to learn and adapt)
-

PEAS

- PEAS: Performance measure, Environment, Actuators, Sensors
- Must first specify the setting for intelligent agent design
-
- Consider, e.g., the task of designing an automated taxi driver:
-
  - Performance measure
  - Environment
  - Actuators
  - Sensors

PEAS

- Must first specify the setting for intelligent agent design
-
- Consider, e.g., the task of designing an automated taxi driver:
-
  - Performance measure: Safe, fast, legal, comfortable trip, maximize profits
  - Environment: Roads, other traffic, pedestrians, customers
  - Actuators: Steering wheel, accelerator, brake, signal, horn
  - Sensors: Cameras, sonar, speedometer, GPS, odometer, engine sensors, keyboard

PEAS

- Agent: Medical diagnosis system
- Performance measure: Healthy patient, minimize costs, lawsuits
- Environment: Patient, hospital, staff
- Actuators: Screen display (questions, tests, diagnoses, treatments, referrals)
- Sensors: Keyboard (entry of symptoms, findings, patient's answers)

PEAS

- Agent: Part-picking robot
- Performance measure: Percentage of parts in correct bins
- **Environment**: Conveyor belt with parts, bins
- **Actuators**: Jointed arm and hand
- **Sensors**: Camera, joint angle sensors

PEAS

- Agent: Interactive English tutor
- Performance measure: Maximize student's score on test
- Environment: Set of students
- Actuators: Screen display (exercises, suggestions, corrections)
- Sensors: Keyboard

An Autonomous Delivery Robot

Environment types

Fully observable (vs. partially observable): An agent's sensors give it access to the complete state of the environment at each point in time.

Deterministic (vs. stochastic): The next state of the environment is completely determined by the current state and the action executed by the agent. (If the environment is deterministic except for the actions of other agents, then the environment is strategic)

Episodic (vs. sequential): The agent's experience is divided into atomic "episodes" (each episode consists of the agent perceiving and then performing a single action), and the choice of action in each episode depends only on the episode itself.

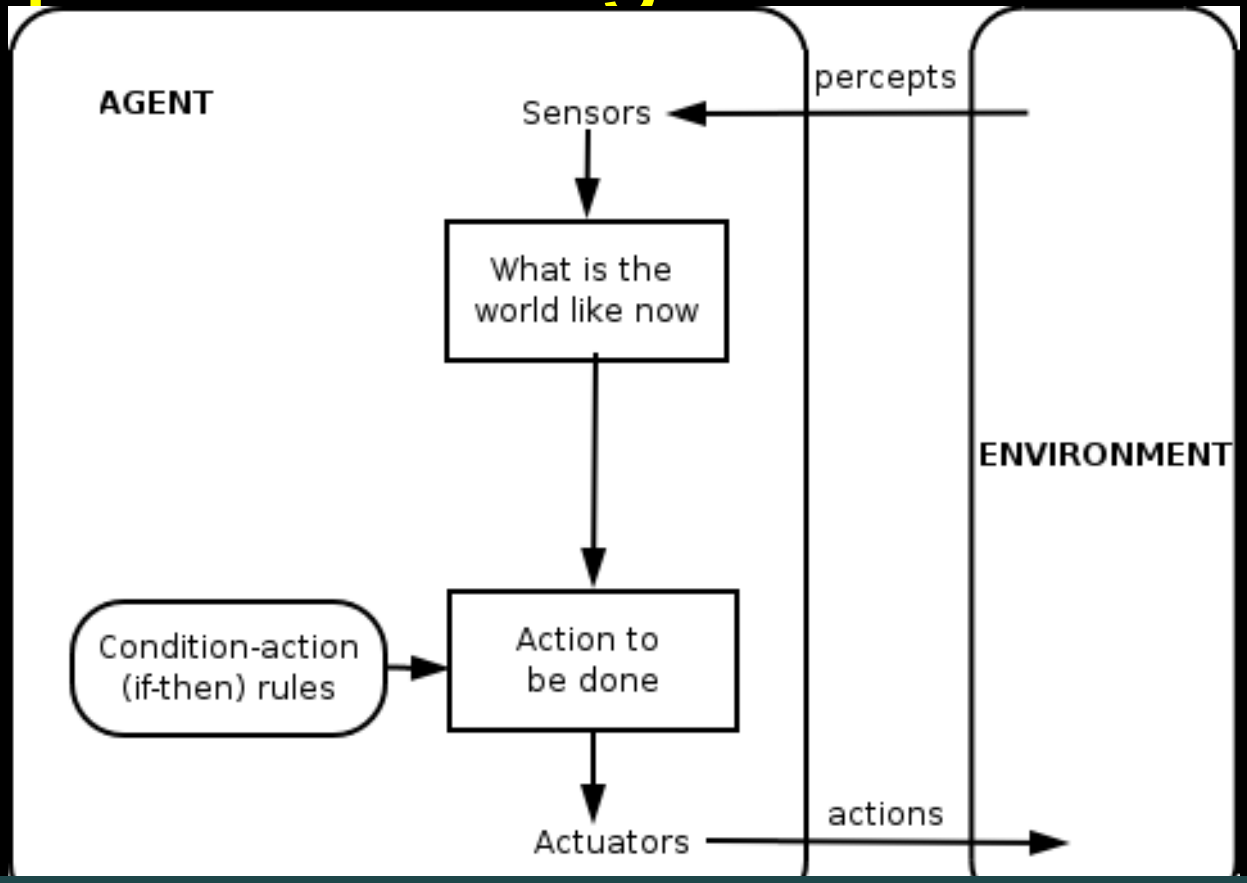
Environment types

- ✓ Static (vs. dynamic): The environment is unchanged while an agent is deliberating. (The environment is semidynamic if the environment itself does not change with the passage of time but the agent's performance score does)
- ✓
- ✓ Discrete (vs. continuous): A limited number of distinct, clearly defined percepts and actions.
- ✓
- ✓ Single agent (vs. multiagent): An agent operating by itself in an environment.
- ✓

Agent types

- Simple reflex agents
- Model-based reflex agents
- Goal-based agents
- Utility-based agents

Simple reflex agents



A simp hes the curren

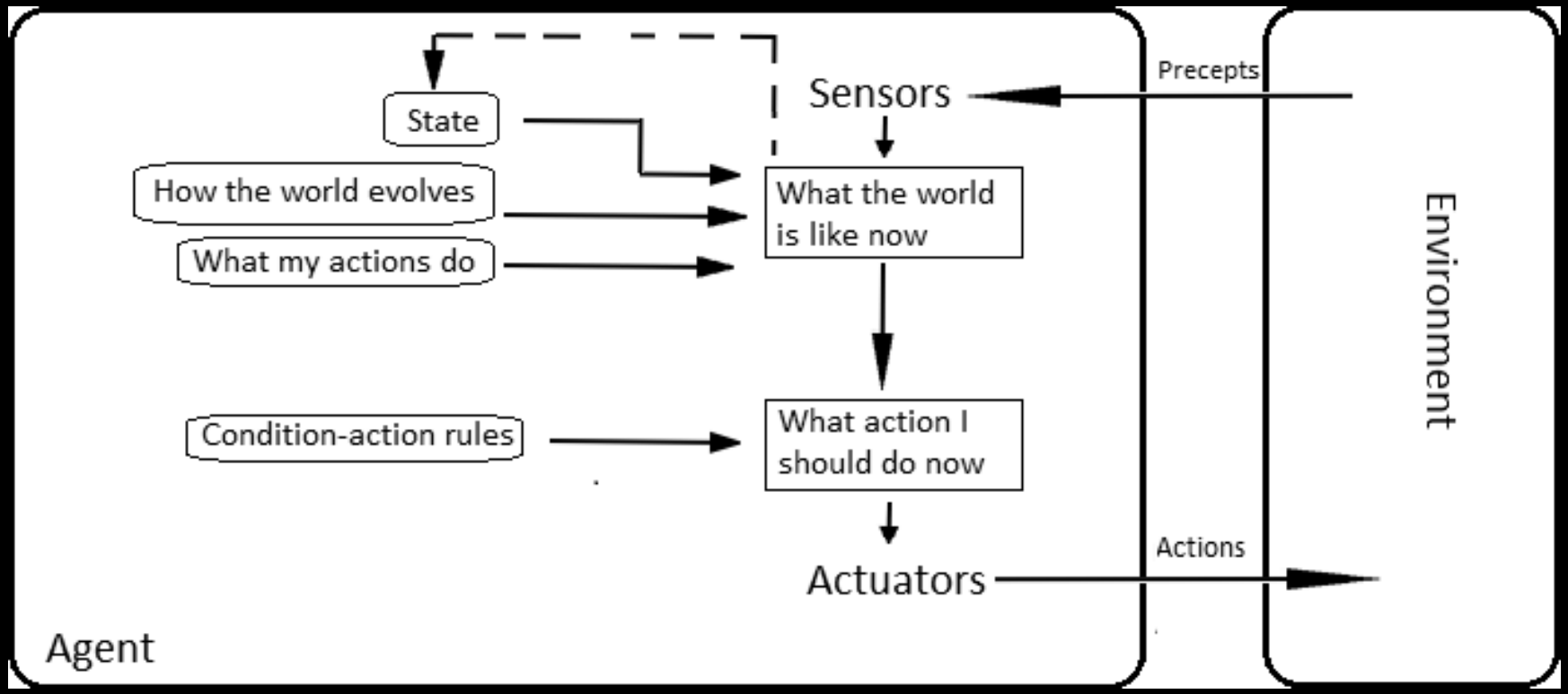
function SIMPLE-REFLEX-AGENT(*percept*) **returns** an action

persistent: *rules*, a set of condition–action rules

state ← INTERPRET-INPUT(*percept*) *rule* ← RULE-MATCH(*state*, *rules*) *action* ← *rule*.ACTION

return *action*

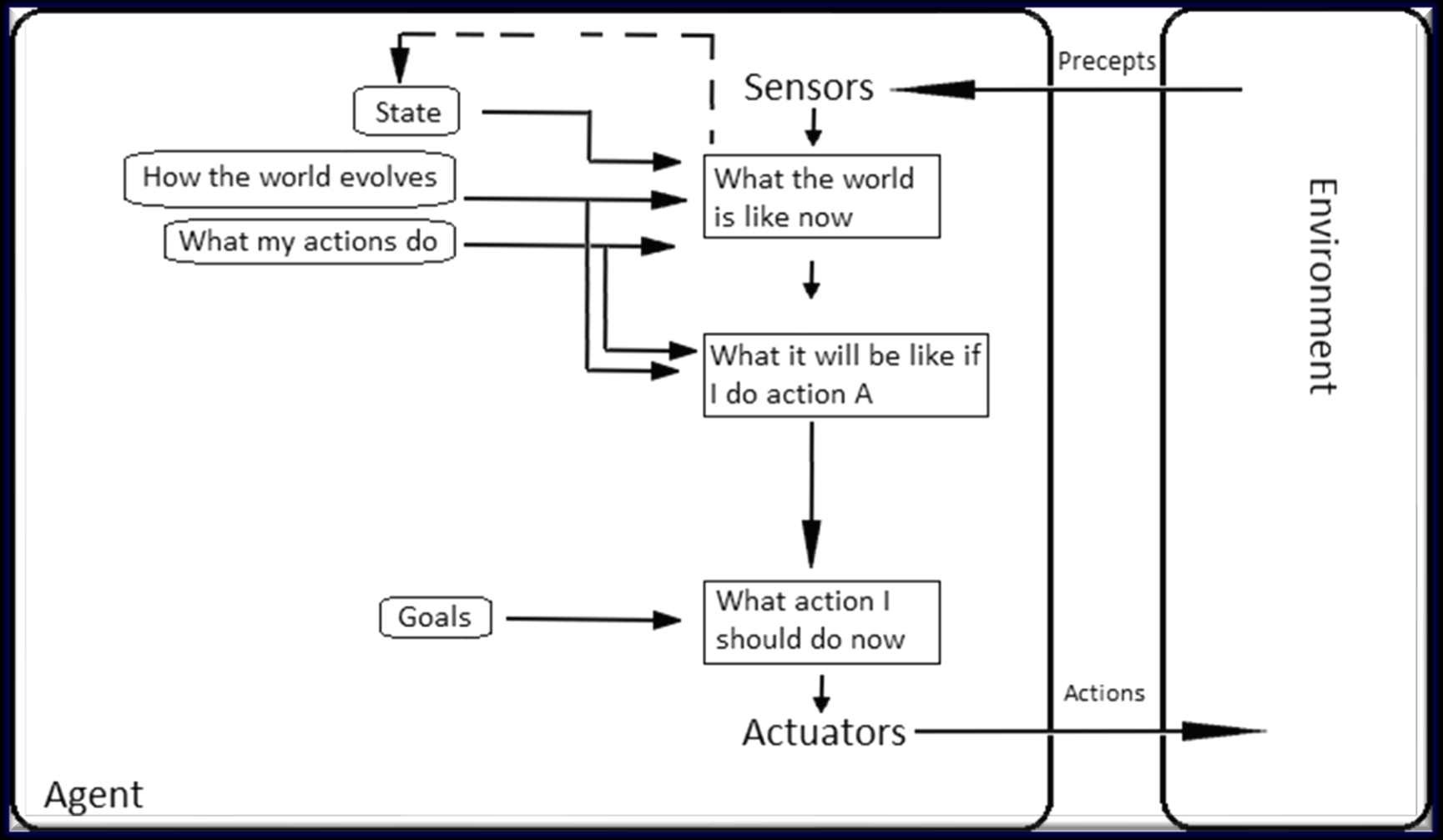
Model-based reflex agents



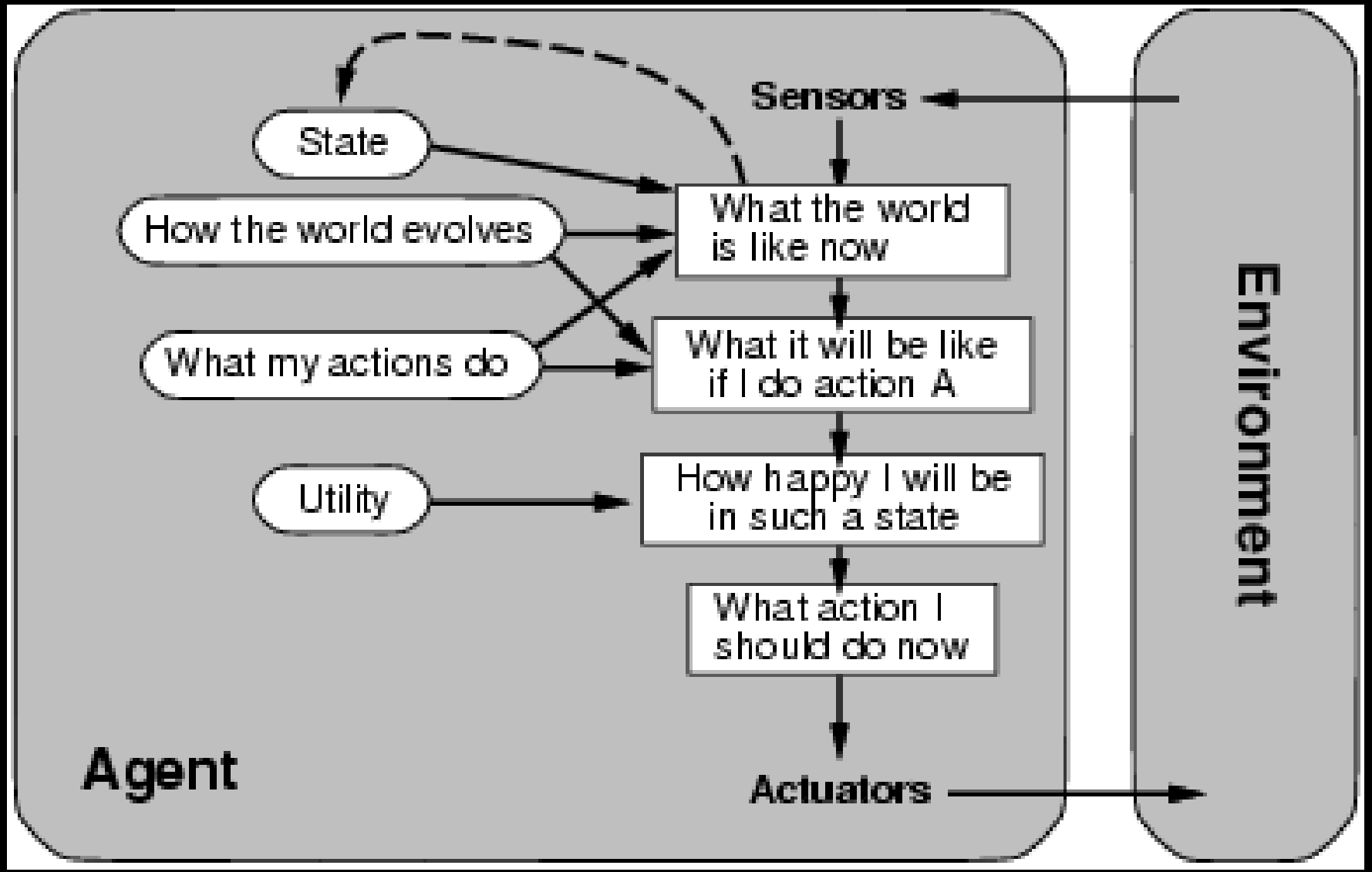
the agent should maintain some sort of **internal state** that depends on the percept history and thereby reflects at least some of the unobserved aspects of the current state.

For the **braking** problem, the internal state is not too extensive— just the previous frame from the camera, allowing the agent to detect when two red lights at the edge of the vehicle go on or off simultaneously. For other driving tasks such as changing lanes, the agent needs to keep track of where the other cars are if it can't see them all at once. And for any driving to be possible at all, the agent needs to keep track of where its keys are.

Goal-based agents



Utility-based agents



Learning agents