Intelligent Agents

CSE-345: Artificial Intelligence

Outline

- Agents and Environments
- Rationality
- PEAS (Performance measure, Environment, Actuators, Sensors)
- Environment types
- Agent types

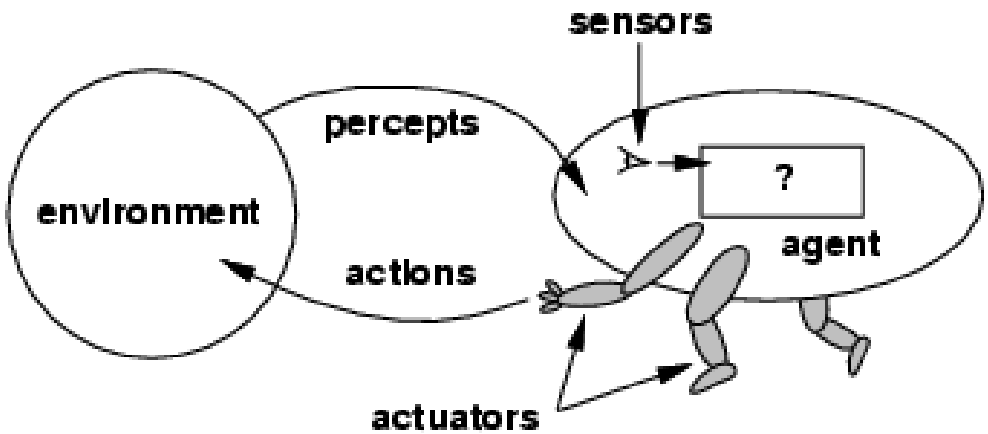
Agents

- An agent is anything that can be viewed as perceiving its environment through sensors & acting upon that environment through actuators

- **Human agent**:
  - eyes, ears, & other organs for sensors;
  - hands, legs, mouth, & other body parts for actuators

- **Robotic agent**:
  - cameras & infrared range finders for sensors;
  - various motors for actuators

Agents and Environments



- The agent function maps from percept histories to

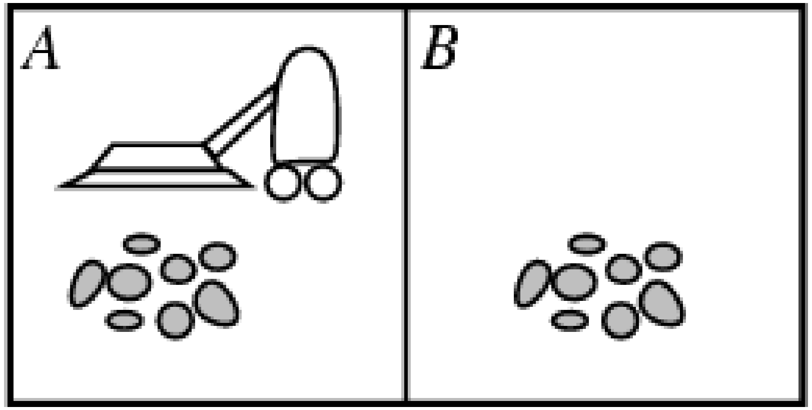
$$f: P^* \rightarrow A$$

- The agent program runs on the physical architecture to produce f

Agent = Architecture + Program

Vacuum-cleaner world

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



A vacuum-cleaner agent

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*
⋮	⋮

- What is the right function? Can it be implemented in a small agent program?

Agent Program vs. Agent Function

- If an agent has |P| possible perception, how many entries will the agent function have after T time steps?

$$\sum_{t-1}^{T} |P|^{t}$$

AI goal→ Design small agent programs to represent huge agent functions

A Possible Agent Program

```
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

- An agent should strive to "*do the right thing*", based on what it can perceive & the actions it can perform.
- The right action is the one that will cause the agent to be most successful
- Rational Agent: For each possible percept sequence, a rational agent should select an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has.

Rationality

- Fixed **performance measure** evaluates the environment sequence
  - one point per square cleaned up in time T
  - one point per clean square per time step, minus one per move
  - penalize for > k dirty squares
  - Penalize for > m units of electricity consumed per time step
  - Penalize for amount of noise generated
- A rational agent chooses whichever action maximizes the expected value of the performance measure given the percept sequence

  Rational $\neq$ omniscient

  - percepts may not supply all relevant information

  Rational $\neq$ clairvoyant

  - action outcomes may not be as expected

  Hence, Rational $\neq$ successful
- Rational $\Rightarrow$ exploration, learning, autonomy

PEAS

- **PEAS:** Performance measure, Environment, Actuators, Sensors
- To design a rational agent, we must specify the task environment

PEAS(Example)

- **Agent:** An automated taxi
- **Performance measure??** safety, destination, profits, legality, comfort, …….
- **Environment??** US streets/freeways, traffic, pedestrians, weather, …….
- **Actuators??** steering, accelerator, brake, horn, speaker/display, ….
- **Sensors??** video, accelerometers, gauges, engine sensors, keyboard, GPS, ……

PEAS(Example)

- 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(Example)

- 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(Example)

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

PEAS(Example)

- Agent: Internet shopping agent
- Performance measure?? price, quality, appropriateness, efficiency
- Environment?? current and future WWW sites, vendors, shippers
- Actuators?? display to user, follow URL, fill in form
- Sensors?? HTML pages (text, graphics, scripts)

Environment Types

- **Fully observable vs. Partially observable**
  - If an agent's sensors give it access to the complete state of the environment at each point in time then the environment is effectively and **fully observable**
    - if the sensors detect all aspects
    - That are relevant to the choice of action
  - An environment might be **partially observable** because of noisy and inaccurate sensors or because parts of the state are simply missing from the sensor data.
    - A local dirt sensor of the cleaner cannot tell
    - Whether other squares are clean or not
- **Deterministic vs. Stochastic**
  - Next state of the environment completely determined by the current state and the actions executed by the agent, then the environment is **deterministic**, otherwise, it is **Stochastic**.
    - Cleaner and taxi driver are stochastic because of some unobservable aspects → noise or unknown

Environment Types

- **Episodic vs. sequential**
  - Episodic
    - An episode = agent's single pair of perception & action
    - The quality of the agent's action does not depend on other episodes
      - Every episode is independent of each other
    - Episodic environment is simpler
      - The agent does not need to think ahead
  - Sequential
    - Current action may affect all future decisions

      -Ex. Taxi driving and chess.

Environment Types

- **Static vs. Dynamic**
  - A **dynamic environment** is always changing over time
    - E.g., the number of people in the street
  - While **static environment**
    - E.g., the destination
  - Semidynamic
    - environment is not changed over time
    - but the agent's performance score does
- Discrete vs. Continuous
  - If there are a limited number of distinct states, clearly defined percepts and actions, the environment is **discrete**
    - E.g., Chess game
  - **Continuous**: Taxi driving

Environment Types

- **Single agent VS. multiagent**
  - An agent operating by itself in an environment is a single agent
    - Playing a crossword puzzle – single agent
    - Competitive multiagent environment- Chess playing
    - Cooperative multiagent environment
      - Automated taxi driver
      - Avoiding collision

- The environment type largely determines the agent design
- The real world is (of course) partially observable, stochastic, sequential, dynamic, continuous, multi-agent

Examples of Task Environments and their Characteristics

Task Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Crossword puzzle	Fully	Deterministic	Sequential	Static	Discrete	Single
Chess with a clock	Fully	Strategic	Sequential	Semi	Discrete	Multi
Poker	Partially	Strategic	Sequential	Static	Discrete	Multi
Backgammon	Fully	Stochastic	Sequential	Static	Discrete	Multi
Taxi driving	Partially	Stochastic	Sequential	Dynamic	Continuous	Multi
Medical diagnosis	Partially	Stochastic	Sequential	Dynamic	Continuous	Single
Image-analysis	Fully	Deterministic	Episodic	Semi	Continuous	Single
Part-picking robot	Partially	Stochastic	Episodic	Dynamic	Continuous	Single
Refinery controller	Partially	Stochastic	Sequential	Dynamic	Continuous	Single
Interactive English tutor	Partially	Stochastic	Sequential	Dynamic	Discrete	Multi

Agent Types

- Will consider four types of agent:
  - Simple reflex agents
  - Reflex agents with state (Agents that keep track of the world)/Model-based Reflex Agent
  - Goal-based agents
  - Utility-based age
- All these can be turned into learning agents

Simple Reflex Agents

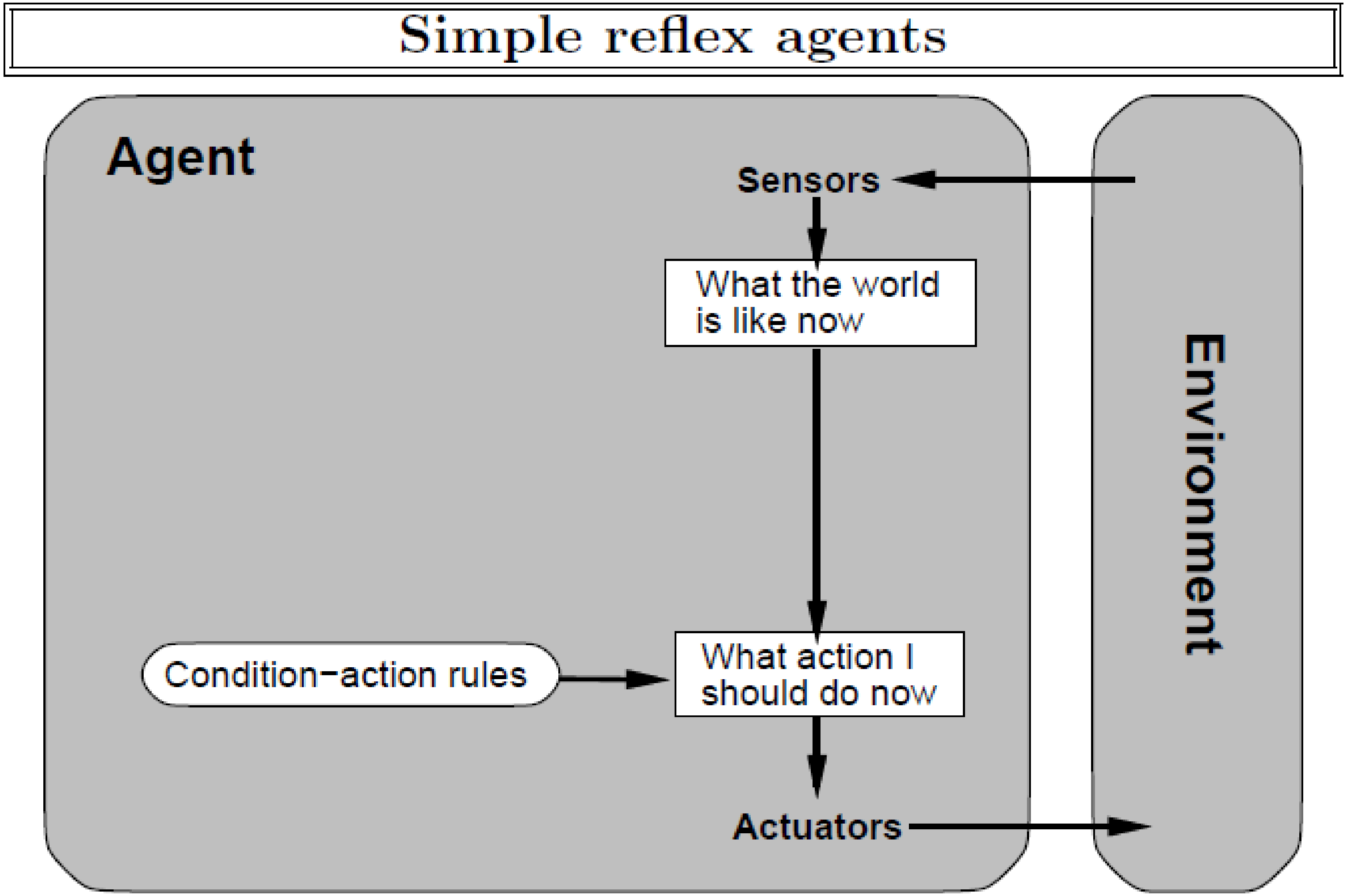
- Action depends only on immediate percepts.
- Uses just ***condition-action rules***
  - rules are like the form "if … then …"
- Works by
  - finding a rule whose condition matches the current situation (as defined by the percept)
  - doing the action associated with that rule.
- Efficient but have narrow range of applicability because
  - knowledge sometimes cannot be stated explicitly
- Work only
  - if the environment is fully observable

Simple Reflex Agent (Example)

- Agent: Automated Taxi
- Environment: streets, traffic, pedestrians, weather
- Rule:

if *car-in-front-is-braking* **then** *initiate-braking*

Simple Reflex Agents (Architecture)



Simple Reflex Agent (Program)

function SIMPLE-REFLEX-AGENT(*percept*)
returns *action*
  persistent: *rules* (set of condition-action rules)
    state ← INTERPRET-INPUT(*percept*)
    rule ← RULE-MATCH(*state*, *rules*)
    action ← rule.ACTION
return *action*

Reflex Agents with State

- For the world that is partially observable
  - the agent has to keep track of an internal state
    - That depends on the percept history
    - Reflecting some of the unobserved aspects
    - E.g., driving a car and changing lane
- Requiring two types of knowledge
  - How the world evolves independently of the agent?
  - How the agent's actions affect the world?

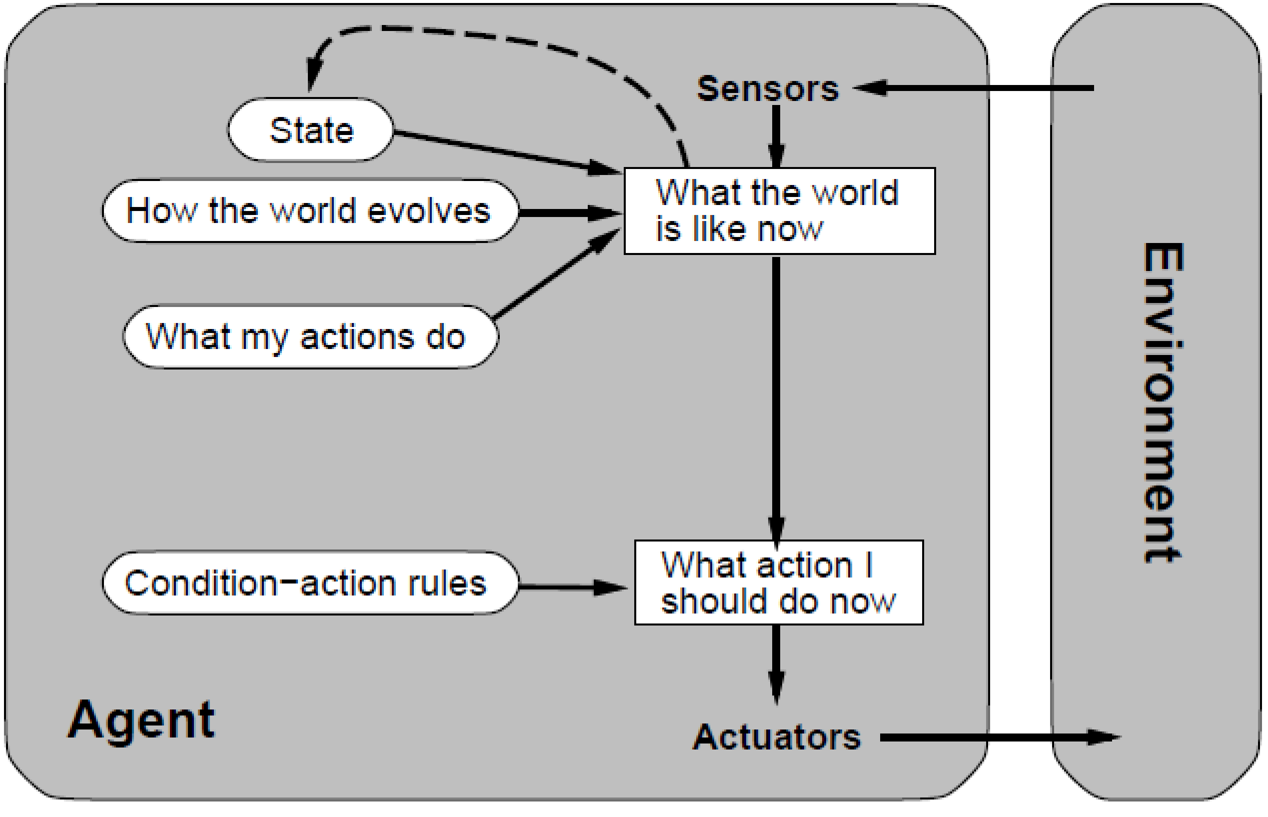
Reflex Agents with State (Example)

Automated Taxi

- Driving a car and changing lane
- Consider the taxi driver following a car without a central mounted brake light and also, consider where the brake light is used as indicator
  - not always possible to tell if car is braking
  - driver will have to maintain internal state in order to choose action
- In the brake light scenario
  - internal state is previous frame from camera
  - used to detect when two red lights at the edge of the vehicle go on or off simultaneously
- Changing Lane Scenario
  - when driver is not looking in rear view mirror vehicles in next lanes are invisible
  - state in which they are present and absent are indistinguishable
  - but, to decide on a lane change, the driver needs to know whether they are there or not
- Updating the internal state requires two kinds of encoded knowledge
  - first, information of how the world evolves independent of the agent - an overtaking car generally will be closer behind than it was a moment ago.
  - second, information of how the agents actions affect the world - changing lanes causes a gap in the lane of origin, driving 5 miles means you are 5 miles away .

Reflex Agents with State (Architecture)

Reflex agents with state



Reflex Agents with State (Program)

```
function REFLEX-AGENT-WITH-STATE(percept)
  returns action
  persistent: state, description of current world state
              model, description of how the next state depends on
                        current state and action
              rules,  a set of condition-action rules
              action, the most recent action, initially none
  state ← UPDATE-STATE(state, action, percept, model)
  rule ← RULE-MATCH(state, rules)
  action ← rule.ACTION
  return action
```

Goal-based Agents

- Current state of the environment is always not enough to decide what to do.
- Needs some sort of **goal** information, which describes situations that are desirable
- Actions chosen to achieve goals, based on
  - the current state and the current percept
  - Sometimes this will be simple, when goal satisfaction results immediately from a single action;
  - Sometimes it will be more tricky, when the agent has to consider long sequences of twists and turns to find a way to achieve the goal.
- Search and planning two other sub-fields in AI to find out the action sequences to achieve its goal
- Goal-based agents are less efficient but more flexible

Agent ← Different goals ← Different tasks

Goal-based Agents (Example)

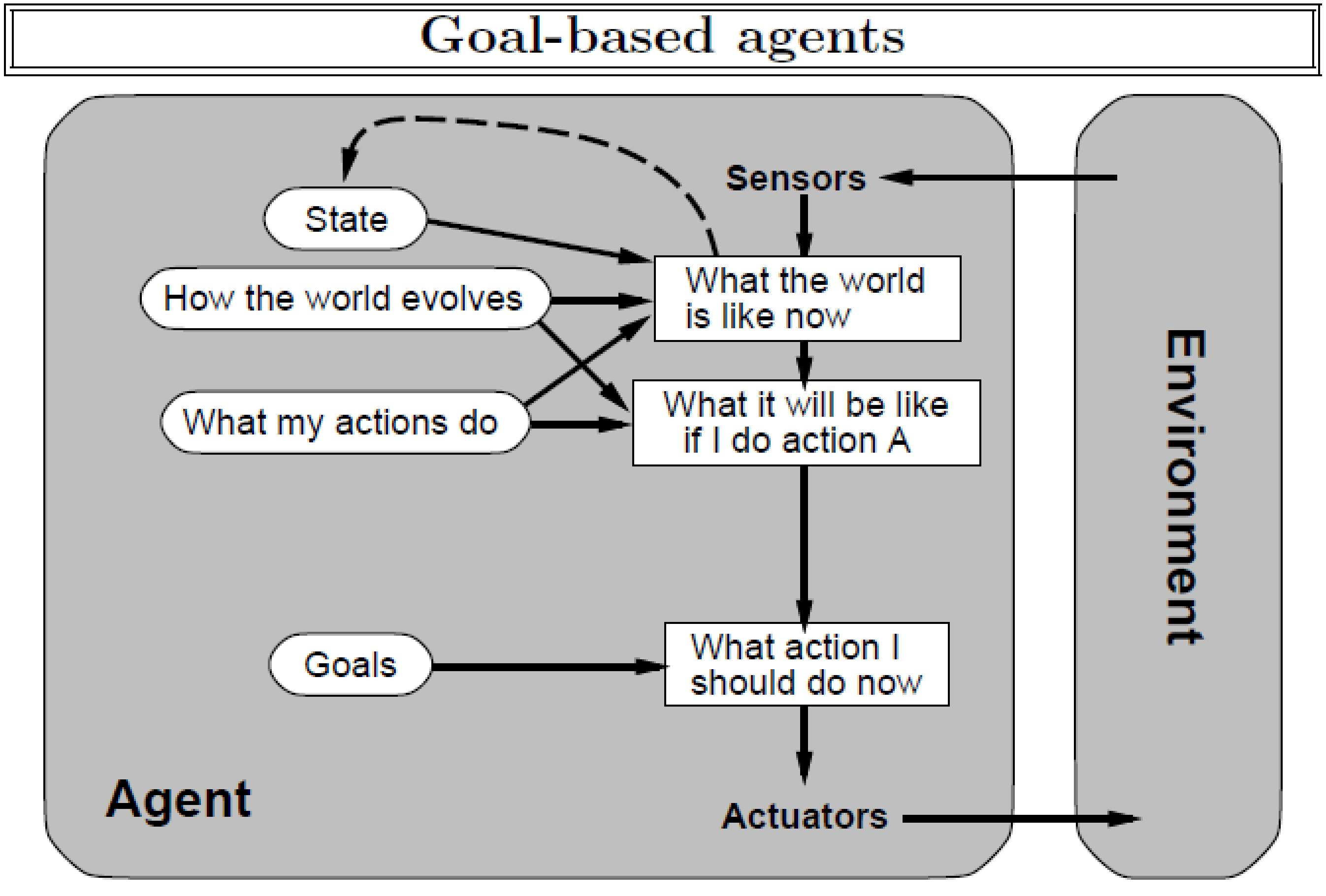
Automated Taxi

- Action
  - Can turn left, right, or go straight on.
  - Right decision depends on where the taxi is trying to get to.
- Goal- passenger's destination
- Uses knowledge about a goal to guide its actions

- Reflex agent breaks when it sees brake lights.
- Goal based agent reasons –

  Brake light -> car in front is stopping -> I should stop -> I should use brake

Goal-based Agents (Architecture)



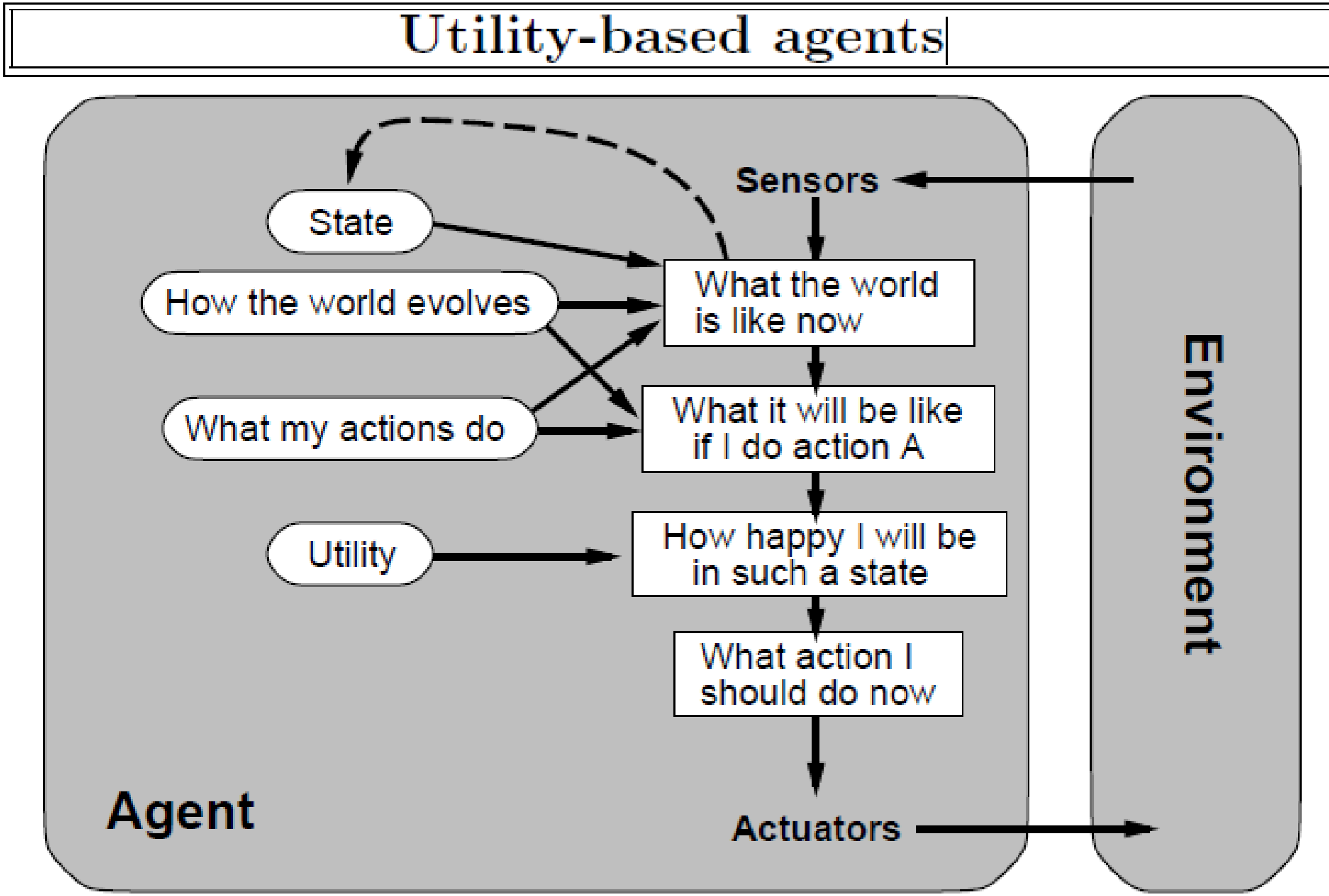
Utility-based Agents

- Goals alone are not really enough to generate high-quality behavior
- Many action sequences may lead to the goals
  - some are better and some worse
- If goal means success, then **utility** means the degree of success (how successful it is)
- It is said state A has higher utility, if state A is more preferred than others
- Utility is therefore a function
  - that maps a state onto a real number
  - the degree of success
- Utility has several advantages:
  - When there are conflicting goals,
    - Only some of the goals but not all can be achieved
    - Utility describes the appropriate trade-off
  - When there are several goals
    - None of them are achieved **certainly**
    - Utility provides a way for the decision-making

Utility-based Agents (Example)

- Agent: Automated Taxi.
- Environment: roads, vehicles, signs, etc.
- Goals: stay safe, reach destination, be quick, obey law, save fuel, etc.

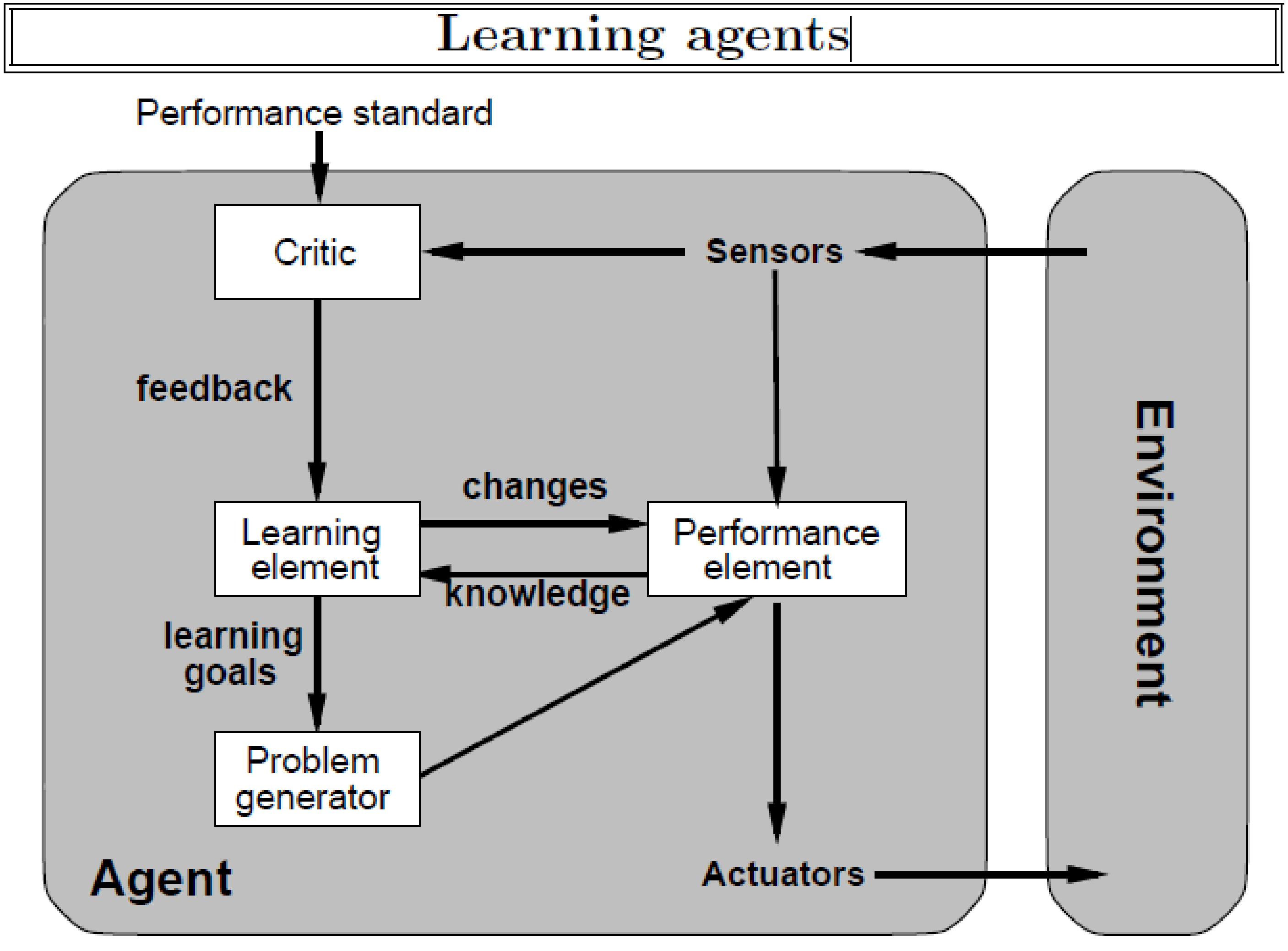
Utility-based Agents (Architecture)



Learning Agents

- After an agent is programmed, can it work immediately?
  - No, it still need teaching
- In AI,
  - Once an agent is done
  - Teach it by giving it a set of examples
  - Test it by using another set of examples
- We then say the agent learns
  - A learning agent
- How do agents improve their performance in the light of experience?
  - Generate problems which will test performance.
  - Perform activities according to rules, goals, model, utilities, etc.
  - Monitor performance and identify non-optimal activity.
  - Identify and implement improvements.

Learning Agents (Example)



Learning Agents

- Four conceptual components
  - Performance Element – which takes in percepts and decides on appropriate actions in the same way as a non-learning agent.
  - Critic – which uses a fixed standard of performance to tell the learning element how well the agent is doing.
  - Learning Element – that receives information from the critic and makes appropriate improvements to the performance element.
  - Problem Generator – that suggests actions that will lead to new and informative experiences (e.g. as in carrying out experiments). **(Teacher)**

Summary

- **Agents** interact with environments through actuators and sensors
- The **agent function** describes what the agent does in all circumstances
- **Agent programs** implement (some) agent functions
- The **performance measure** evaluates the environment sequence
- A perfectly **rational agent** maximizes expected performance
- **PEAS** descriptions define task environments
- **Environments** are categorized along several dimensions:

  observable? deterministic? episodic? static? discrete?

  single-agent?
- Several basic **agent architectures** exist:

  reflex, reflex with state, goal-based, utility-based

Assignment-1

1. Write down the PEAS description for:
   – Satellite Image Analysis System
   – Refinery Controller
2. What are the characteristic of the environment that would be experienced by
   – a mail-sorting robot?
   – an intelligent house?
3. You are designing a vacuum-cleaning robot to sweep your house while you are away. You decide to make it a reflex agent with internal state. What sort of rules or reflex should it have and what sort of sensors do they presuppose? Suppose you want it to be able to go dump its dust load when full, or go to a recharging station when its batteries are low. Is the reflex agent an adequate architecture and if not what must be added?

The END