Intelligent Agents

Lecture 2

Lecture 1 Recap

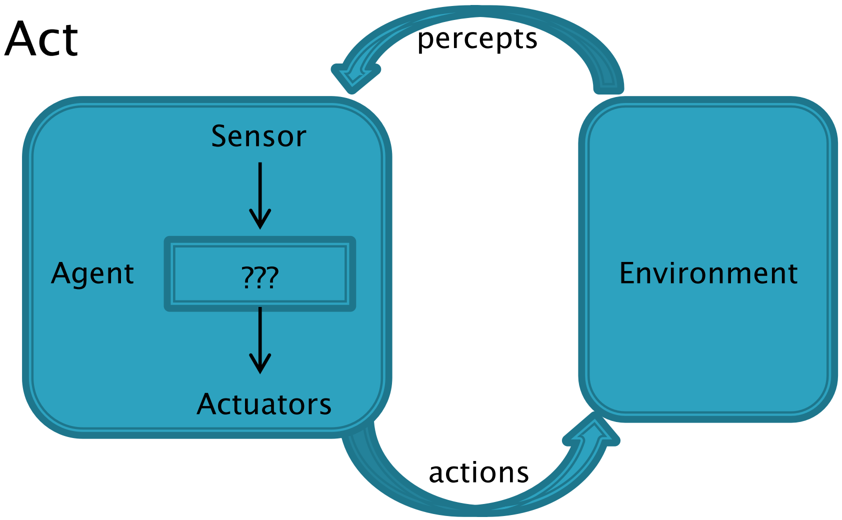
- What is AI?
  - Thinking Humanly, Thinking Rationally, Acting Humanly, Acting Rationally
- Goal of AI
  - Cognitive Science
  - Turing test
- Current State of AI
  - Acting Rationally

Lecture 1 Recap

- Subfields of Artificial Intelligence
  - Problem solving
  - Knowledge representation and reasoning studies
  - Planning
  - Learning
  - Vision
  - Natural language

Introduction

- Recall: Current State of AI
- Acting Rationally
  - Take the best possible action given their goals, knowledge and Constraints
- Modelling as an agent
  1. Systems that interact with an **environment**
  2. using **sensors** to receive perceptual inputs (called **percepts**) from it,
  3. and **actuators** to act upon it.
  4. A process of Perceive, Think and Act
- Agent = architecture + program
  - physical sensors and actuators



Introduction

- A human agent
  - Sensors: eyes, ears
  - Actuators: hands, legs, vocal tract
- A robotic vacuum agent
  - Sensors: cameras and infrared range finders
  - Actuators: wheels, vacuum, mop
- A software agent
  - Sensors: receives keystrokes, file contents, and network packets
  - Actuators: displaying on the screen, writing files, and sending network packets.

Percept

- Percept
  - state of an agent's sensors at a given moment in time
- Perceptual sequence
  - Sequence of all Percepts received
  - the complete history of everything the agent has ever perceived
- Example:
  - Using your phone's camera as an example, explain Percept and Perceptal sequence when taking a picture

Agent function

- Agent function
  - maps any given percept sequence to an action
- Agent function table
  - lists an action for every possible combination of perceptual history
  - Feasibility?
    - Can we define all possible sequence?
    - If not, can we place a bound on the length of percept sequences we want to consider?
- Can implemented by an **agent program**

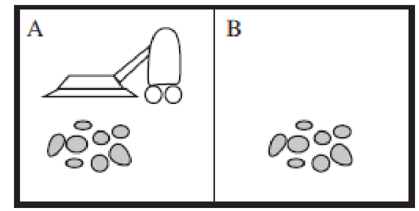


Figure 2.2 A vacuum-cleaner world with just two locations.

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*, *Clean*]	*Right*
[*A*, *Clean*], [*A*, *Clean*], [*A*, *Dirty*]	*Suck*
⋮	⋮

Figure 2.3 Partial tabulation of a simple agent function for the vacuum-cleaner world shown in Figure 2.2.

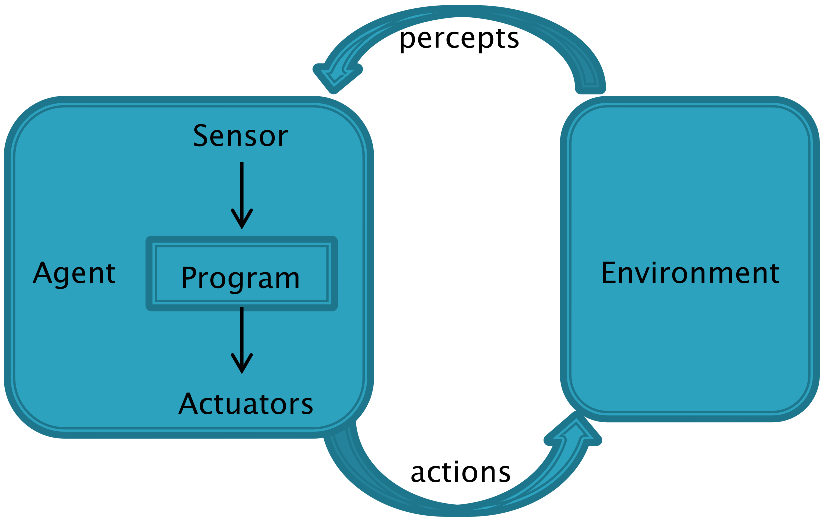
```
function REFLEX-VACUUM-AGENT([location,status]) returns an action
  if status = Dirty then return Suck
  else if location = A then return Right
  else if location = B then return Left
```

Figure 2.8 The agent program for a simple reflex agent in the two-state vacuum environment. This program implements the agent function tabulated in Figure 2.3.

(Ref: Russell & Norvig)

The (Intelligent) Agent Program

- Exhibit rational behavior
  - Define **performance measure**, a numerical metric that expresses the goals of an agent
- Perform decision making
- Perform action selection
  - Allow agent program to determine the best way to achieve that goal
- Intelligent agent
  - Setting for intelligent agent design
    - **PAGE** (Percepts, Actions, Goals, Environment)
  - Specify the task environment
    - **PEAS** (Performance Measure, Environment, Actuators and Sensors)



PAGE VS PEAS

- Automated taxi example

PAGE

Percepts	Actions	Goals	Environment
Video, engine sensors	Accelerate, brake, horn	Safety, reach destination	Urban street, pedestrians

PEAS

Performance measure	Environment	Actuators	Sensors
Safety, reach destination	Urban street, pedestrians	Accelerator, brake, horn	Video, engine sensors

PAGE VS PEAS

- PAGE

Agent Type	Percepts	Actions	Goals	Environment
Medical diagnosis system	Symptoms, findings, patient's answers	Questions, tests, treatments	Healthy patient, minimize costs	Patient, hospital
Satellite image analysis system	Pixels of varying intensity, color	Print a categorization of scene	Correct categorization	Images from orbiting satellite
Part-picking robot	Pixels of varying intensity	Pick up parts and sort into bins	Place parts in correct bins	Conveyor belt with parts
Refinery controller	Temperature, pressure readings	Open, close valves; adjust temperature	Maximize purity, yield, safety	Refinery
Interactive English tutor	Typed words	Print exercises, suggestions, corrections	Maximize student's score on test	Set of students

PAGE VS PEAS

- PEAS

Agent Type	Performance Measure	Environment	Actuators	Sensors
Medical diagnosis system	Healthy patient, reduced costs	Patient, hospital, staff	Display of questions, tests, diagnoses, treatments, referrals	Keyboard entry of symptoms, findings, patient's answers
Satellite image analysis system	Correct image categorization	Downlink from orbiting satellite	Display of scene categorization	Color pixel arrays
Part-picking robot	Percentage of parts in correct bins	Conveyor belt with parts; bins	Jointed arm and hand	Camera, joint angle sensors
Refinery controller	Purity, yield, safety	Refinery, operators	Valves, pumps, heaters, displays	Temperature, pressure, chemical sensors
Interactive English tutor	Student's score on test	Set of students, testing agency	Display of exercises, suggestions, corrections	Keyboard entry

Figure 2.5 Examples of agent types and their PEAS descriptions.

(Ref: Russell & Norvig)

Rational agent

- Given an agent, we define a rational action as the action
  - expected to maximize its performance metric,
  - given its percept sequence
  - and prior knowledge
- A performance measure is a numerical metric that expresses the goals of an agent.

The limitations of design

- If an agent can use it's own experience rather than prior knowledge (instilled by designer), then it is more desirable and "smarter"
  - More autonomy means using more own experience
  - Require less expert knowledge
  - More adaptable than agents that require a great deal of problem-specific knowledge
  - So we aim to build autonomous agents
- But autonomous agents are harder to design
  - Easier to build a problem-specific agent with all the knowledge that it might need
  - Than to build a general one that can acquire that knowledge

Degree of Autonomy

- A self driving vehicle
  - Type 1: Using an in-built 3D model to plan and execute the journey
  - Type 2: Will follow a predetermined route, encoded in its computer
  - Type 3: Uses location services as well as online mapping applications and traffic updates

Properties of task environment [ref:C3310 Study guide]

1. Fully observable vs partially observable
2. Deterministic vs stochastic
3. Episodic vs sequential
4. Static vs dynamic
5. Discrete vs continous
6. Known vs unknown
7. Single versus multi-agent

Why do we need this?

Properties of task environment [ref:C3310 Study guide]

- Fully observable vs partially observable
  - Fully observable task environment: Sensors giving agent access relevant to choice of action
  - Convenient because agent does not need to maintain any internal state to keep track of the "world"
  - Can be made partially observable because of noisy and inaccurate sensors or because parts of the state are simply missing from the sensor data
  - Example:
    - Chess
    - Driving
    - Missing/Failed Vacuum sensors

Properties of task environment [ref:C3310 Study guide]

- Deterministic vs stochastic
  - Deterministic: current state of the environment and current action are sufficient to exactly predict the next state
  - Stochastic task: some uncertainty, so the next state is often expressed as a **probability distribution**
  - Most real situations are complex
    - impossible to keep track of all the unobserved aspects
    - treated as stochastic for practical purposes
  - Vacuum example:
    - randomly appearing dirt

Properties of task environment [ref:C3310 Study guide]

- Episodic vs sequential
  - Episodic: agent faces a sequence of independent tasks (or episodes)
    - next episode is independent on the actions taken in previous episodes
  - Sequential: agent’s next state always depends on the environment and its current state
    - current decisions affect future decisions, or rely on previous ones
  - Classify the following
    - Spotting defective parts on assembly line
    - Chess
    - Driving

Properties of task environment [ref:C3310 Study guide]

- Static vs dynamic
  - Static: if nothing in the environment changes, apart from agent and its impact
  - Dynamic: environment may change over time
    - if an agent hasn't decided yet, that counts as deciding to do nothing
  - Classify the following
    - Crossword puzzles
    - Chess with timer
    - Driving

Properties of task environment [ref:C3310 Study guide]

- Discrete vs continuous
  - applies to the **state of the environment**, to the way **time is handled**, and to the **percepts** and **actions** of the agent
  - Example:
    - Chess
      - Environment has a finite number of distinct states
      - Discrete set of percepts and actions.
    - Driving
      - Speed and location of car and other vehicles is continuous and ever changing
      - Driving actions are also continuous (steering angles, etc.).

Properties of task environment [ref:C3310 Study guide]

- Known vs unknown
  - the agent's state of knowledge about how the environment functions
  - Known: the outcomes / outcome probabilities for all actions are given
    - Action A will result in Outcome B
  - Unknown: agent will have to learn how it works in order to make good decisions
    - Action A will result in what Outcome??
  - Example
    - Solitaire card games
    - First time using an app
  - Distinction between (Fully/partially observable) vs (Known/unknown)

Properties of task environment [ref:C3310 Study guide]

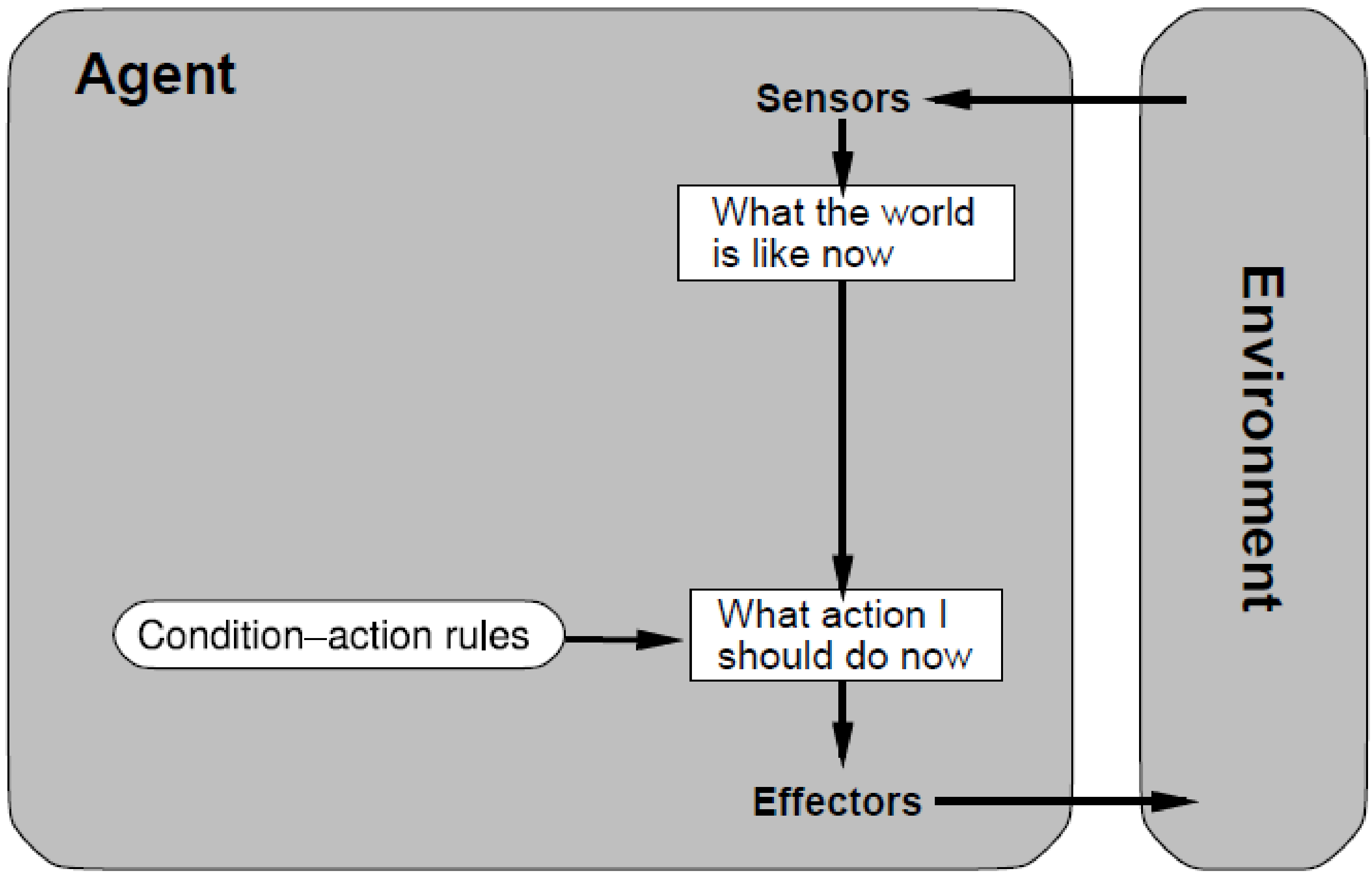
- Single versus multi-agent
  - multi-agent: other agents exists in the environment with their own performance measure
  - Competitive or co-operative Multi-agent task environments
    - Key question: Are their performance measure aligned?
    - Two agents playing chess against each other
    - Self driving cars

Types of agents

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

*Not a type of agent by itself, but rather a characteristic that can be incorporated into the above 4 types of agent

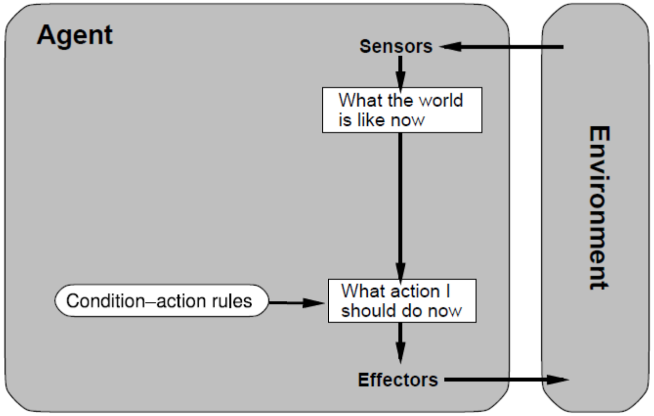
Simple Reflex Agent (Ref: Russell & Norvig)



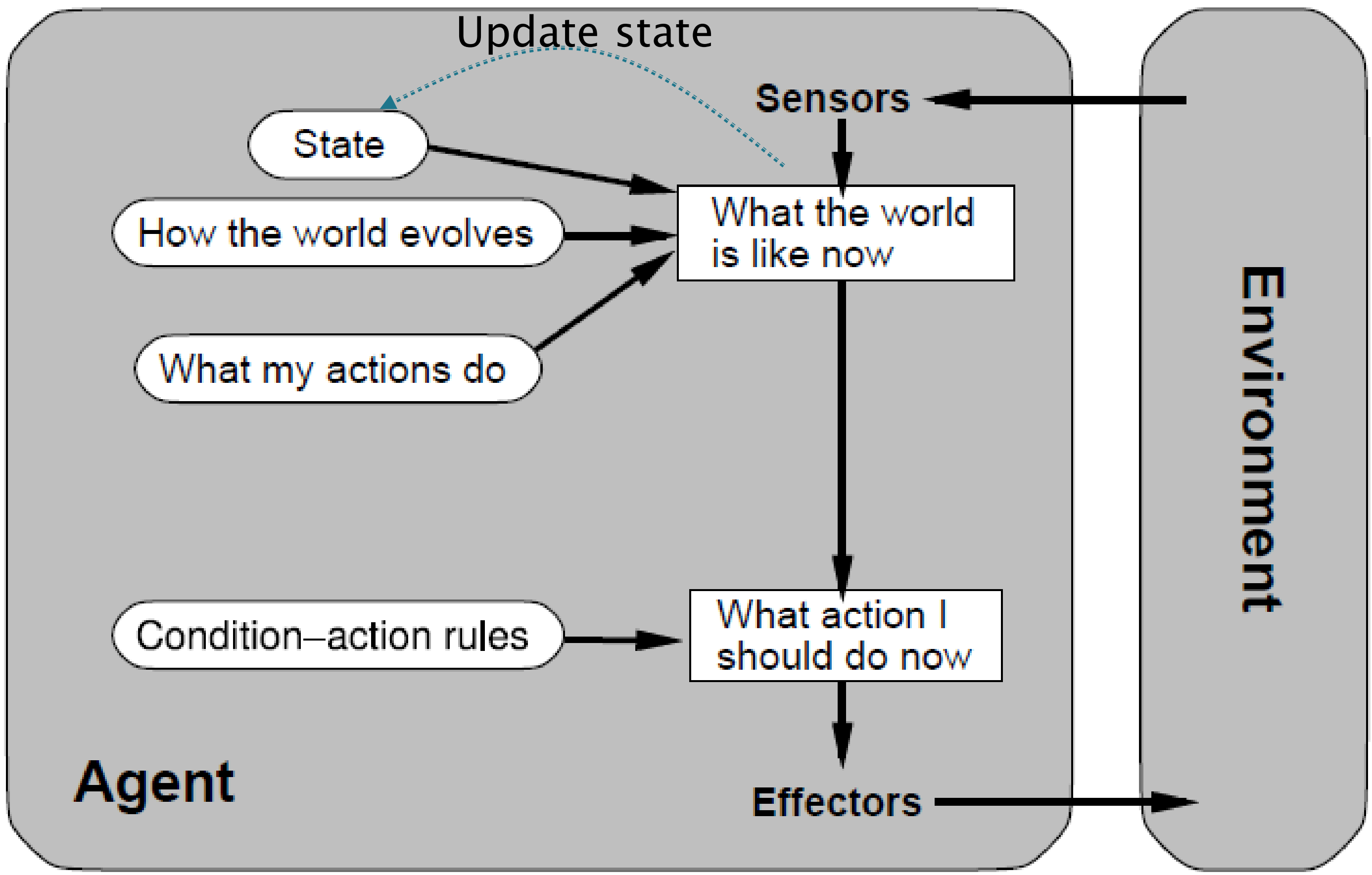
Simple Reflex Agent (Ref: Russell & Norvig)

- Select actions on the basis of the current percept, ignoring the rest of the **percept history**
- condition-action rule
  - if *car-in-front-is-braking* then *initiate-braking*
  - if *light-turns-green* then *move-forward*
- Can only work if the environment is fully observable
  - the correct action is based on the current percept only
  - Vacuum example: what happens if the location sensor is taken away?

Percept[	Action
[A, Clean]	[Right]
[A, Dirty]	[Suck]
[B, Clean]	[Left]
[B, Dirty]	[Suck]

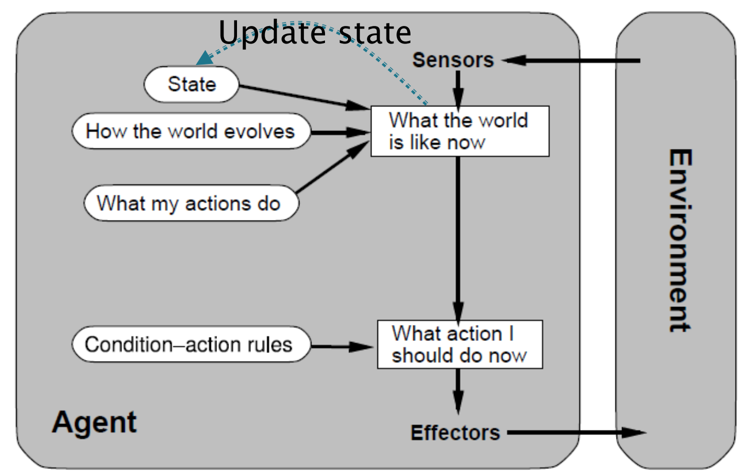


Model-based Reflex Agent (Ref: Russell & Norvig)

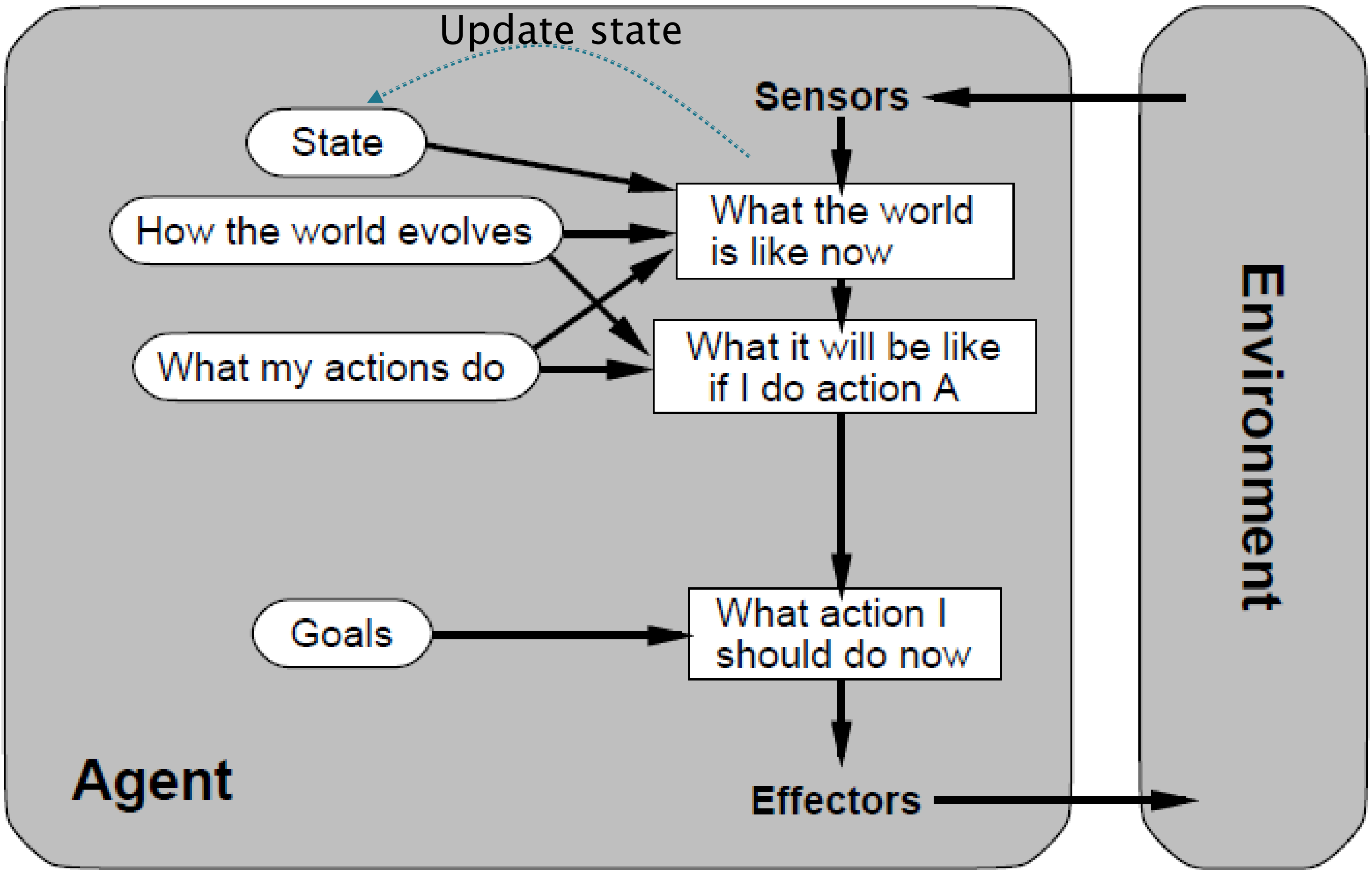


Model-based Reflex Agent (Ref: Russell & Norvig)

- Contains a model of the world / “how the world works”
  - Allows the agent to keep track of the world, despite partial observability
- function UPDATE-STATE
  - augment the information in the current percept
  - responsible for creating the new internal state description
  - best guess (best guesses), with an element of uncertainty
- Model is based on
  - how the world evolves independently from the agent
  - how the agent actions affects the world

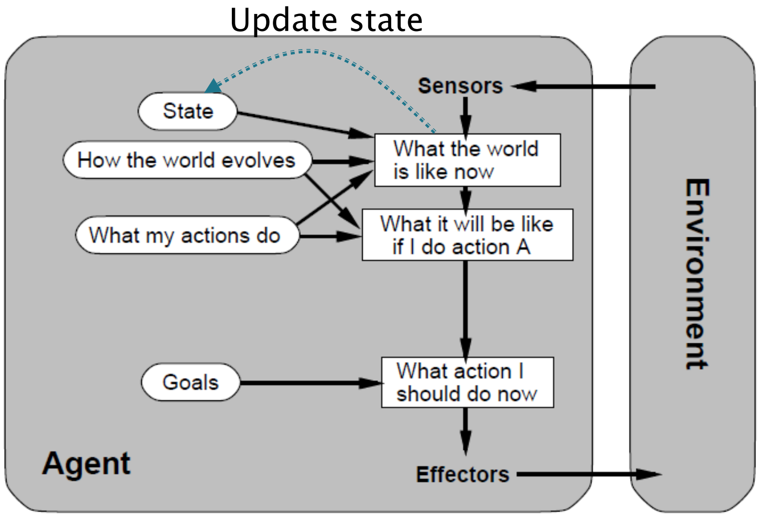


Goal based Agent (Ref: Russell & Norvig)

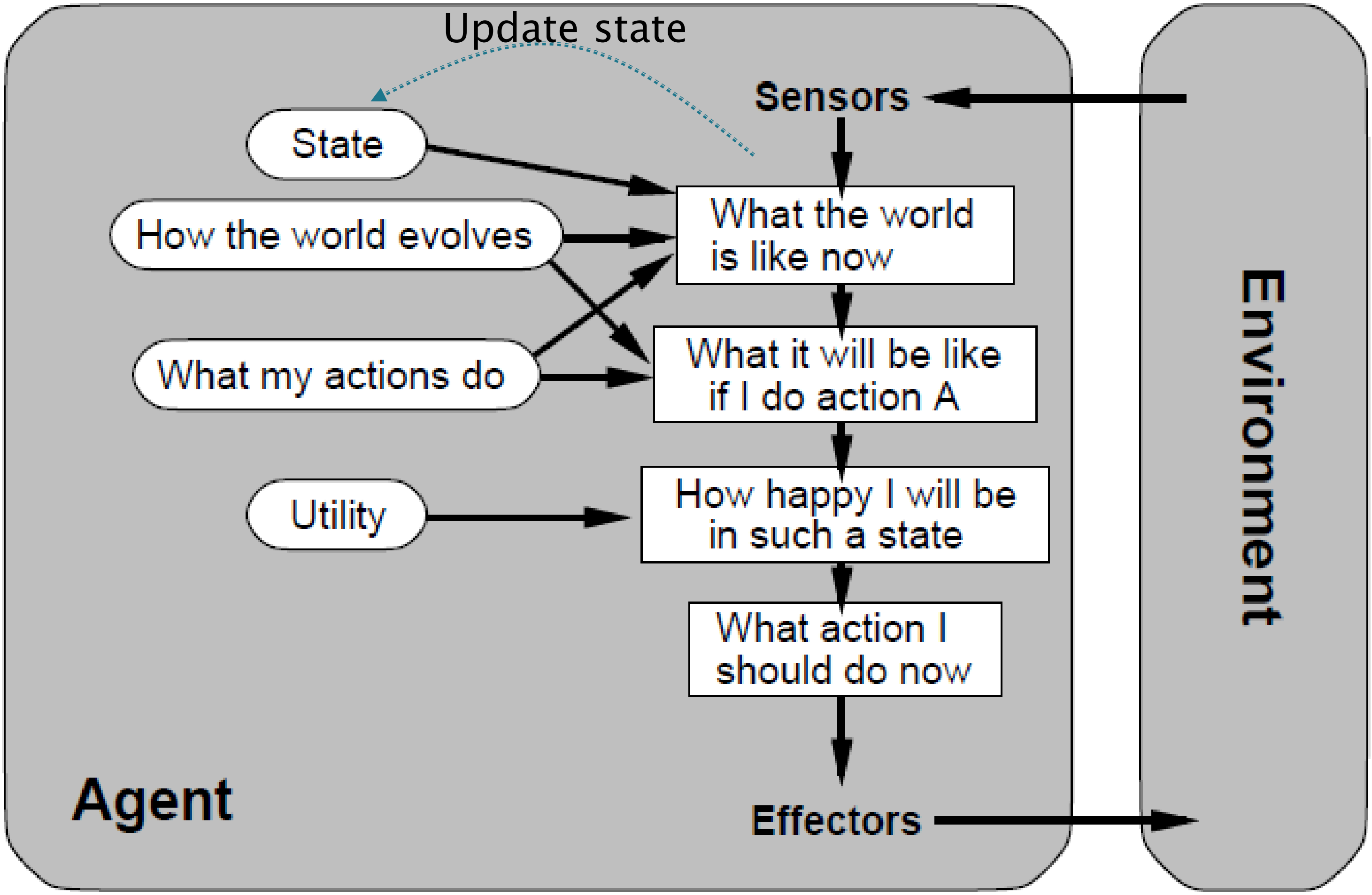


Goal based Agent (Ref: Russell & Norvig)

- Current state of environment might be insufficient to make a decision
  - employ knowledge of the goals encoded in their performance metric
  - employ search and planning to select actions
- Increased flexibility in
  - Responding to changing environment
  - Accepting different goals
- Example: Driving
  - Where are we going?
  - What is the route?

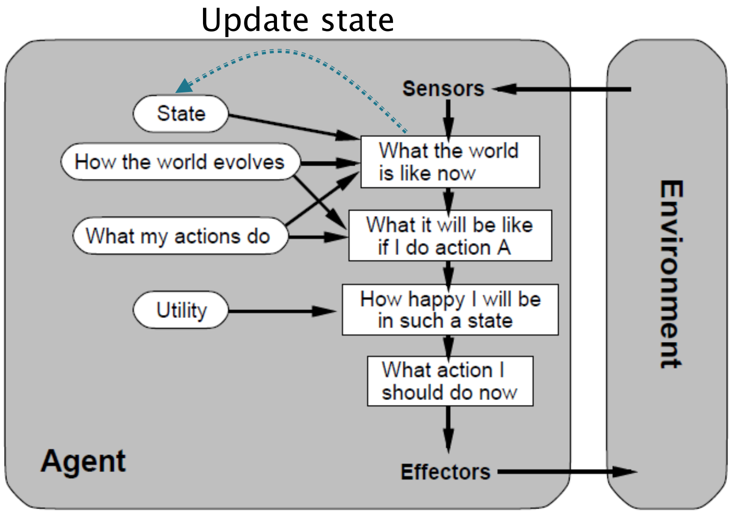


Utility based Agent (Ref: Russell & Norvig)

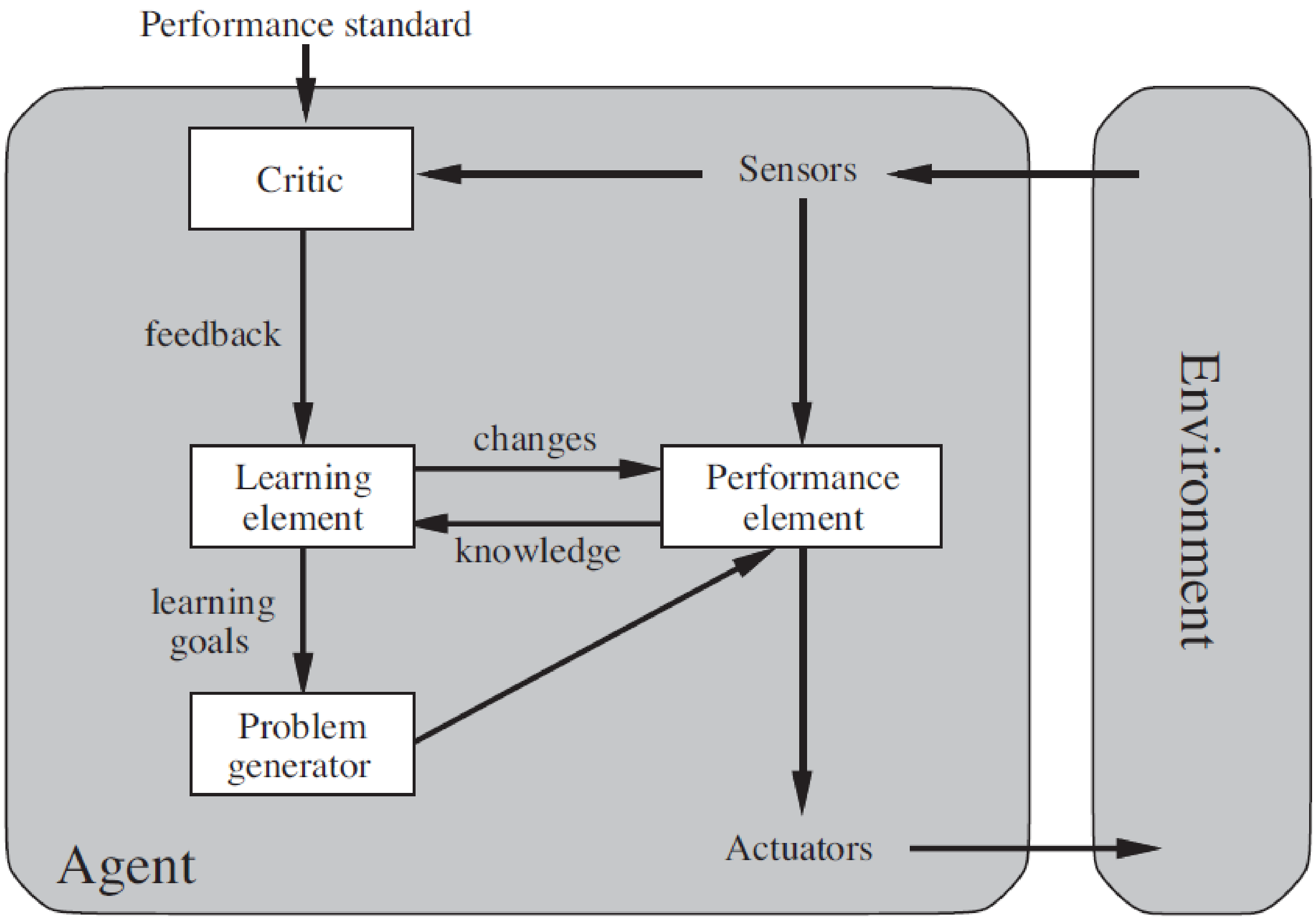


Utility based Agent (Ref: Russell & Norvig)

- To have the maximized expected utility (total satisfaction received)
  - Many ways of achieving the goals but what is the “best”?
- An agent’s **utility function** is essentially an internalization of the performance measure
  - Helps decision-making
    - conflicting goals (by helping find a trade-off)
    - several possible goals, none of which is achievable with certainty

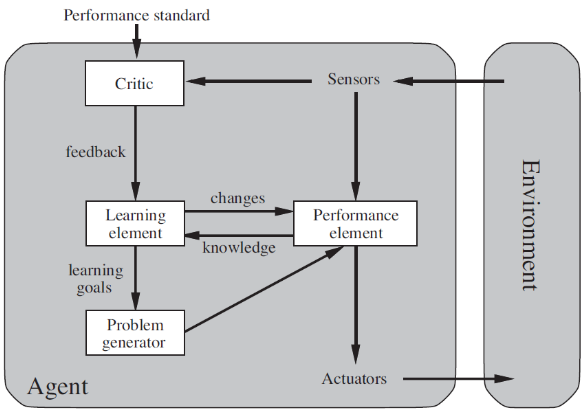


Learning Agent (Ref: Russell & Norvig)



Learning Agent (Ref: Russell & Norvig)

- To build learning machines and then to teach them
  - allows the agent to operate in initially unknown environments
  - become more competent than its initial knowledge alone might allow
  - four conceptual components
    - learning element, performance element, critic, problem generator



Exercises

- To be discussed in class