Lecture 1

Intelligent Agents

Lusi Li
Department of Computer Science
Old Dominion University

Reading for This Class:
Chapter 2, Russell and Norvig

Special thanks to Dr. Yaohang Li for sharing the slides

Outline

- Agents and Environments
- Rationality
- PEAS Description
- Environment Types
- Agent Types
  - Simple reflex agent
  - Model-based reflex agent
  - Goal-based agent
  - Utility-based agent
  - Learning agent

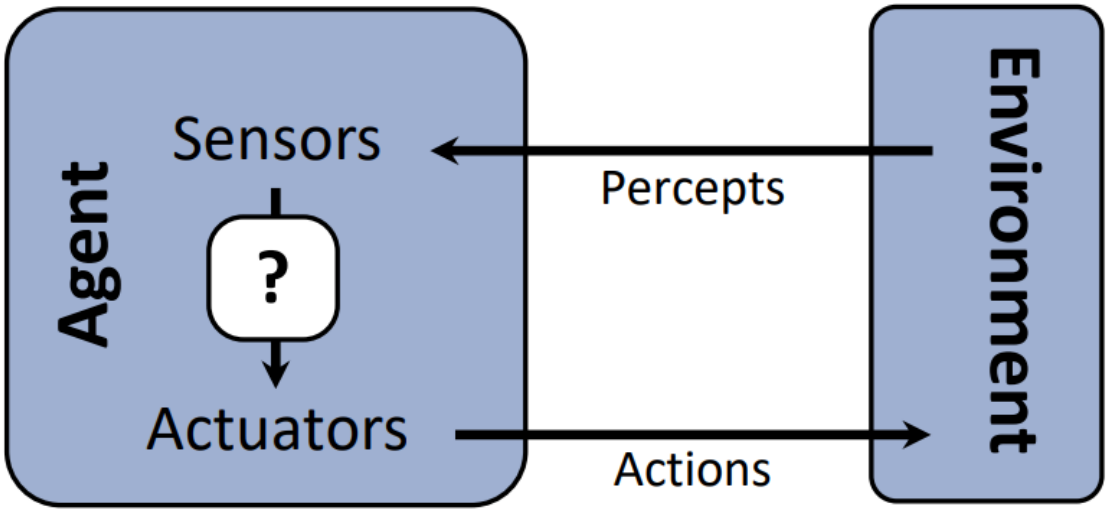
- Next Class
  - Problem-Solving Agent

Introduction to Intelligent Agents

- **Why study agents?**
  - Problem Solving
  - Automation and Efficiency
  - Adaptability and Learning
  - Personalization and User Interaction
  - …
- **What is an agent?**
  - in general, an entity that interacts with its environment
    - perception through sensors
    - actions through effectors or actuators

How can we design an "intelligent" agent to solve a specific task in a particular environment?

Agents and Environments



An agent perceives its environment through sensors and acts on the environment through actuators.

Different kinds of agents: humans, robots, softbot, etc.

Example of Agents

- **Human agent**
  - **eyes, ears, skin, taste buds, etc. for sensors**
  - **hands, fingers, legs, mouth, etc. for actuators**
    - powered by muscles
- **Robot**
  - **camera, infrared, bumper, etc. for sensors**
  - **grippers, wheels, lights, speakers, etc. for actuators**
    - often powered by motors
- **Software agent**
  - **functions as sensors**
    - information provided as input to functions in the form of encoded bit strings or symbols
  - **functions as actuators**
    - results deliver the output

Terminologies

- **Percept: the agent's perceptual inputs**
- **Percept sequence: the complete history of everything the agent has perceived**
- **Agent function: mapping any given percept sequence to an action**

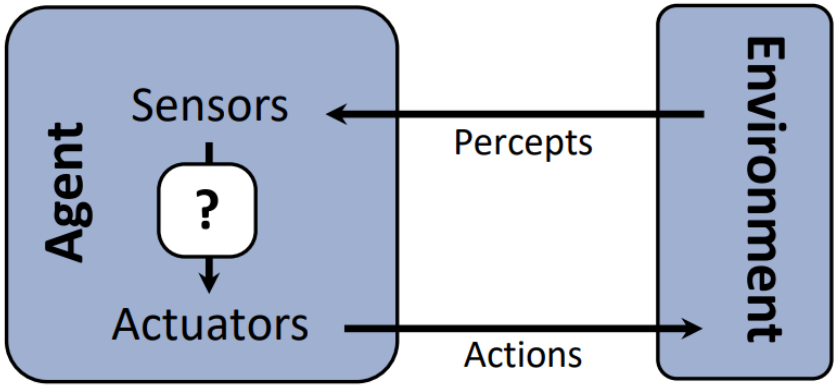
$$\text{AgentFunction}(\langle p^{(1)} p^{(2)} \ldots p^{(t)} \rangle) = a^{(t)}$$

Abstract mathematical description

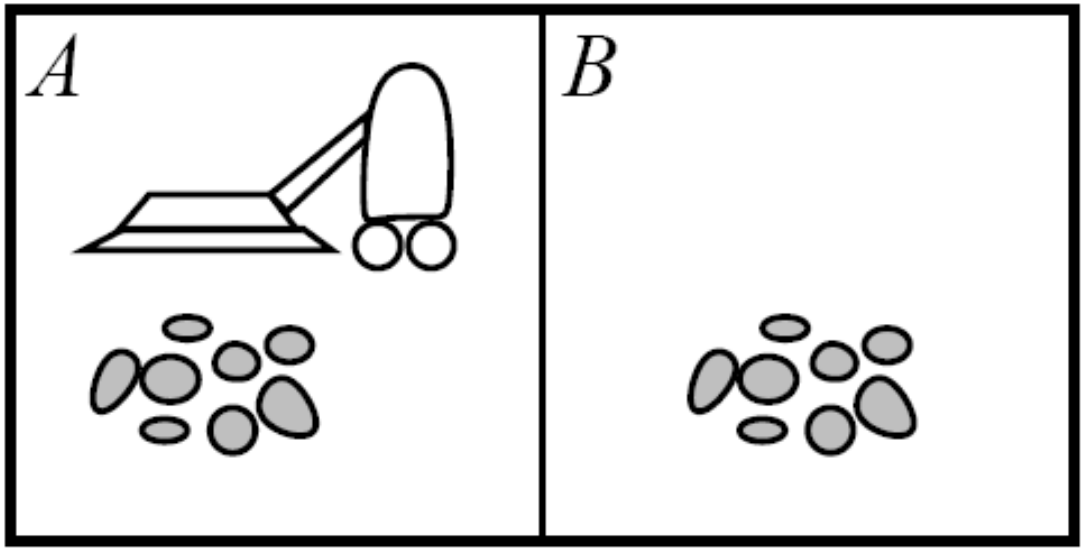
- **Agent program: running on the physical architecture to produce the agent function**

Concrete implementation

- **Agent = architecture + program**



Vacuum-Cleaner World



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

Vacuum-Cleaner World

Tabulation of the agent function

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

Function

Input: location, status
Output: action

```
1: if status = Dirty then
2:     return Suck
3: end if
4: if location = A then
5:     return Right
6: end if
7: if location = B then
8:     return Left
9: end if
```

- **What is the right agent function?**
- **How well does the agent perform?**
- **Can it be implemented in a small agent program?**

Rationality

- A *rational agent* is the one that does "the right thing", based on what it can perceive and what actions it can perform

- Concept of Rationality
  - Rational ≠ omniscient
    - Percepts may not supply all relevant information
  - Rational ≠ clairvoyant
    - Action outcomes may not be as expected
  - Rationality vs Perfection
    - Rationality maximizes expected performance
    - Perfection maximizes actual performance
  - Rational → exploration, learning, autonomy

Rational Agents

- A *rational agent* is the one that does "the right thing", based on what it can perceive and what actions it can perform
- "The right thing" is to select the action that will cause the agent to be most successful
- Performance measure: an objective criterion for success of an agent's behavior by evaluating any given sequence of environment states
  - A sequence of actions -> A sequence of environmental states
  - No fixed performance measure for all tasks and agents
  - Not easy to design a good one

Performance Measure Examples

- **With a rational agent, what you ask for is what you get.**
- **Performance measure examples**
  - **Vacuum agent**

    - **the amount of tiles cleaned during a certain period**
      - based on the agent's report, or validated by an objective authority
      - doesn't consider expenses of the agent, side effects
        - energy, noise, loss of useful objects, damaged furniture, scratched floor
      - might lead to unwanted activities
        - agent re-cleans clean tiles, covers only part of the room, drops dirt on tiles to have more tiles to clean, etc.

    - **alternative measures:**
      - 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?

Rational Agents

- How rational an agent can depend on
  - performance measure
  - what actions it can perform
  - its current percept sequence
  - prior knowledge of the environment

- Definition of a 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.

Environment

- **Environment**
  - **determines to a large degree the interaction between the "outside world" and the agent**
    - **the "outside world" is not necessarily the "real world" as we perceive it**
  - **in many cases, environments are implemented within computers**
    - **they may or may not have a close correspondence to the "real world"**

The Nature of Environment

- **Environment Properties**
  - **fully observable vs. partially observable**
    - **sensors capture all relevant information from the environment**
  - **deterministic vs. stochastic (non-deterministic)**
    - **changes in the environment are predictable**
  - **episodic vs. sequential (non-episodic)**
    - **independent perceiving-acting episodes**
  - **static vs. dynamic**
    - **no changes while the agent is "thinking"**
  - **discrete vs. continuous**
    - **limited number of distinct percepts/actions**
  - **single vs. multiple agents**
    - **interaction and collaboration among agents**
    - **competitive, cooperative**

Specifying the Task Environment

- **To design a rational agent, we must specify its task environment.**

Four elements of a task environment: PEAS Description

Performance Measures

used to evaluate how well an agent solves the task at hand

Environment

surroundings beyond the control of the agent

Actuators

determine the actions the agent can perform

Sensors

provide information about the current state of the environment

PEAS Description

Example: agent = vacuum cleaner

Performance Measures	cleanliness of the floor time needed energy consumed
Environment	grid of tiles dirt on tiles possibly obstacles, varying amounts of dirt
Actuators	movement (wheels, tracks, legs, ...) dirt removal (nozzle, gripper, ...)
Sensors	position (tile ID reader, camera, GPS, ...) dirtiness (camera, sniffer, touch, ...) possibly movement (camera, wheel movement)

PEAS Description

Example: agent = taxi driver

Performance Measures	Safe, fast, legal, comfortable trip
Environment	Roads, other traffic, pedestrians, custmers, weather
Actuators	Steering wheel, accelerator, brake, signal, horn
Sensors	Cameras, LIDAR (light/radar), speedometer, GPS, odometer, engine sensors, keyboard

Chess Player PEAS Description

Example: agent = chess player

Performance Measures	Winning the game Time spent in the game
Environment	Chessboard Positions of every piece
Actuators	Move a piece
Sensors	Input from the keyboard

PEAS Description

Example: agent = medical diagnosis system

Performance Measures	Healthy patient, minimize costs, lawsuits
Environment	Patient, hospital, staff
Actuators	Screen display of questions, tests, diagnoses, treatments, referrals
Sensors	Keyboard (entry of symptoms, findings, patient's answers)

Agent Program

- Agent = architecture + program
- Job of AI: design an agent program that implements the agent function mapping percept sequences to actions

- Program must suit architecture

- Agents receive one percept at a time
  - they may or may not keep track of the percept sequence

- The differences:
  - Agent program's input: the current percept
  - Agent function's input: the entire percept history

Agent Program

- **Basic Framework: input (current percept)**

```
function SKELETON-AGENT(percept) returns action
  static: memory
  memory := UPDATE-MEMORY(memory, percept)
  action := CHOOSE-BEST-ACTION(memory)
  memory := UPDATE-MEMORY(memory, action)
return action
```

Agent Program Types

- **Classification of Agent Programs**
  - different ways of achieving the mapping from percepts to actions
  - different levels of complexity
- **Agent Program Types**
  - simple reflex agents
  - model-based reflex agents
  - goal-based agents
  - utility-based agents
  - learning agents

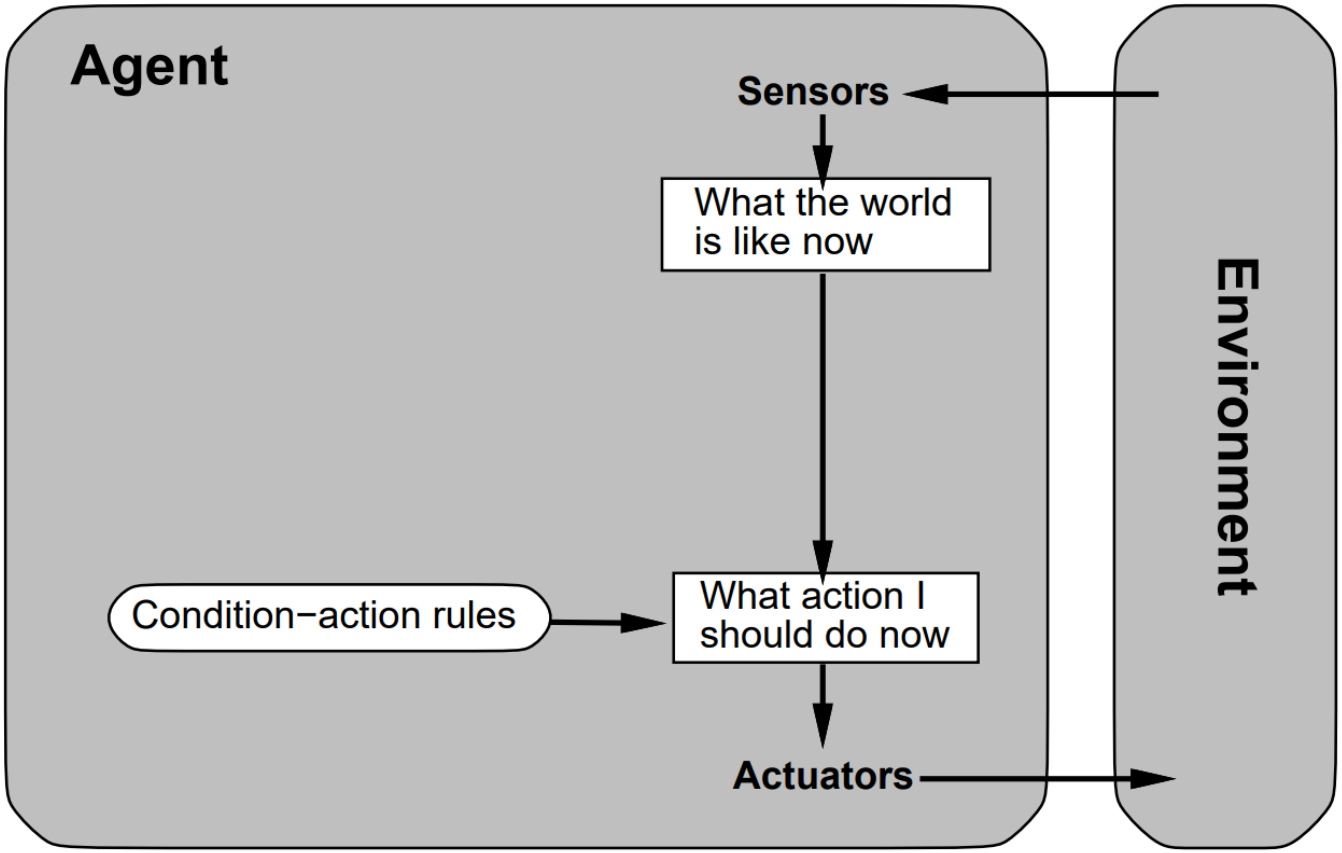
Simple Reflex Agents

- **Simple Reflex Agent**
  - **action depends only on the current percept**
  - **has no memory of percept history**
  - **agent function is based on condition-action rules**
    - **a rule maps a *percept* , i.e., a condition to an *action***
    - **E.g., if *car-in-front-is-braking* then *brake***

- **Actions are selected based on the current percepts.**
- **Rule-based reasoning**

```
function SIMPLE-REFLEX-AGENT(percept) returns action
  static: rules   // set of condition-action rules
  condition := INTERPRET-INPUT(percept)
  rule      := RULE-MATCH(condition, rules)
  action    := RULE-ACTION(rule)
return action
```

Simple Reflex Agent



Pros: easy to implement, efficient when environment is fully observable

Cons: limited power in complex environment since no memory

Simple Reflex Agent

Example: vacuum agent

```
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
```

Model-based Reflex Agents

- **Maintain an internal state that:**
  - **helps with some partially observable environments**
  - **keeps track of the current state of the world**
  - **updated by the knowledge below:**
    - **how the agent's own actions affect the world?**
    - **how the world evolves independently from the agent?**
- **The knowledge of "how the world works" is called a model of the world**

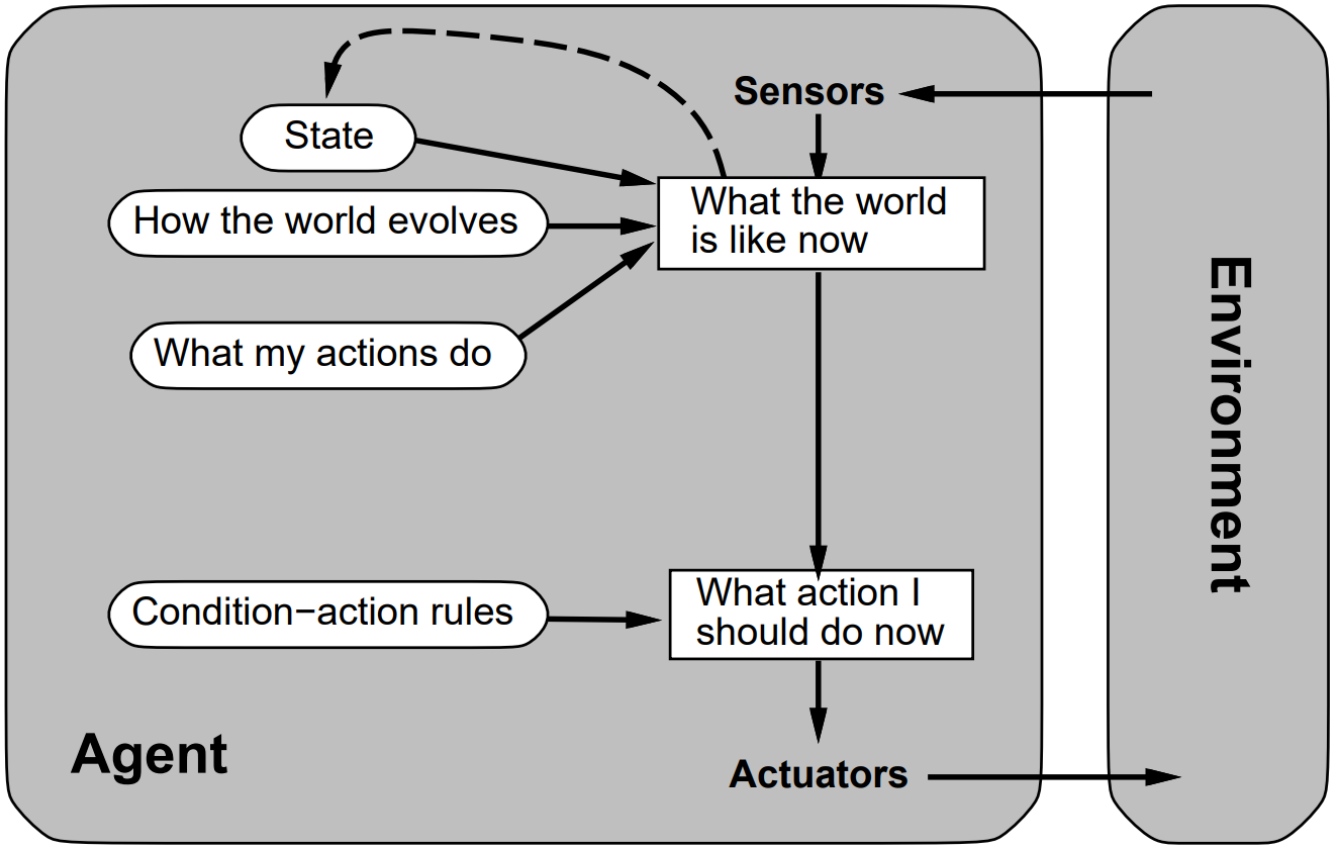
- **Actions are selected based on the model and current percepts.**

```
function REFLEX-AGENT-WITH-STATE (percept) returns action
   static: rules          //set of condition-action rules
           state          //internal state; description of the current world state
           action         //most recent action, initially none

   state          := UPDATE-STATE(state, action, percept) //new internal state
   rule           := RULE-MATCH(state, rules)
   action         := RULE-ACTION[rule]

   return action
```

Model-based Reflex Agents



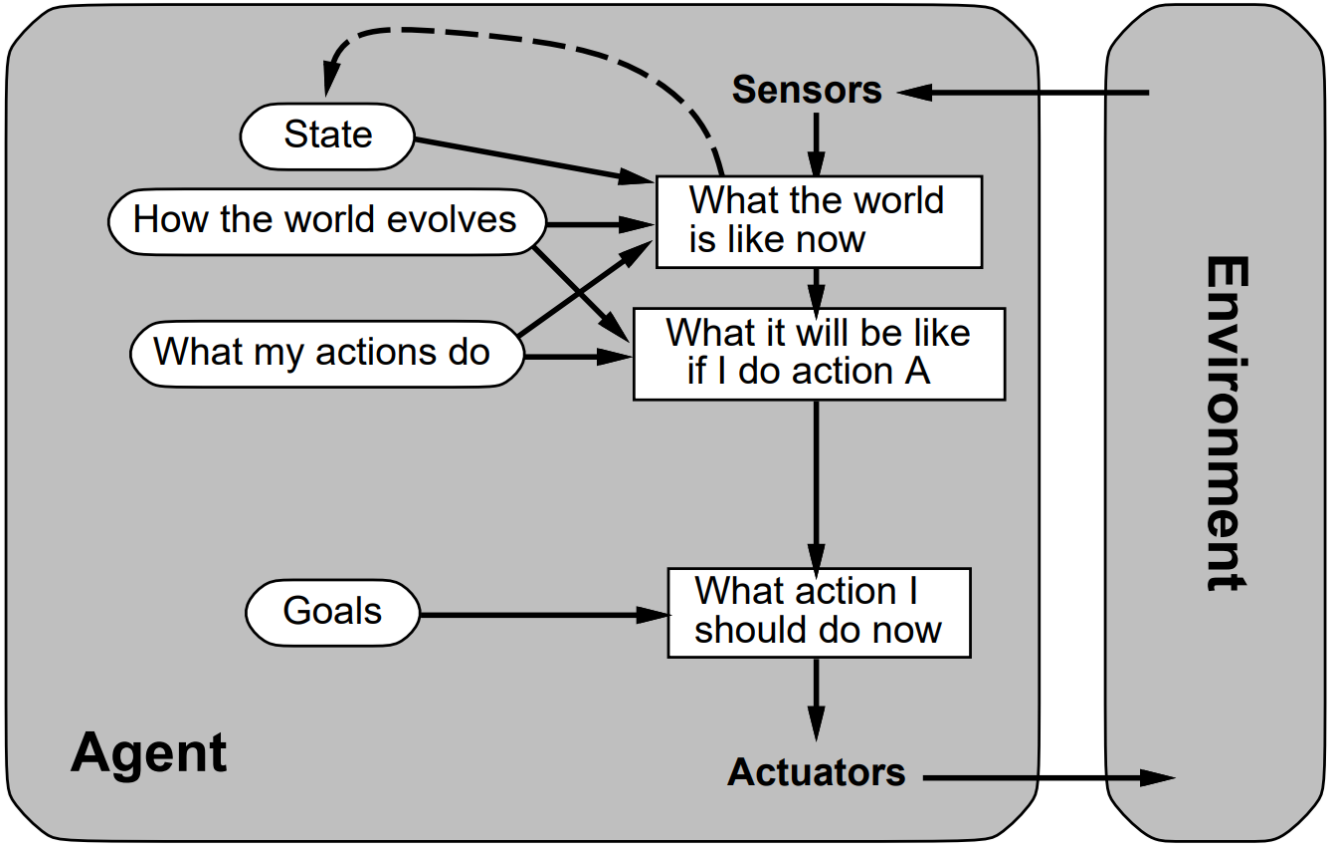
Pros: can handle partially observable environments by keeping an internal state

Cons: no way to express goals and preferences relative to goals

Goal-based Agents

- Often, to make the correct decision, the agent needs some sort of goal information that describes situations that are desirable.
- Selection actions that will achieve goals
- Goal-based action selection
  - easy when the results can be related to the goal after each action
  - in general, it can be difficult to attribute goal satisfaction results to individual actions
  - may require consideration of the future
    - What will happen if I do this? (search, reasoning or planning)
    - Will this make me happy? (happy or unhappy)

Goal-based Agents



Pros: more flexible in dynamic environments and accepting different goals

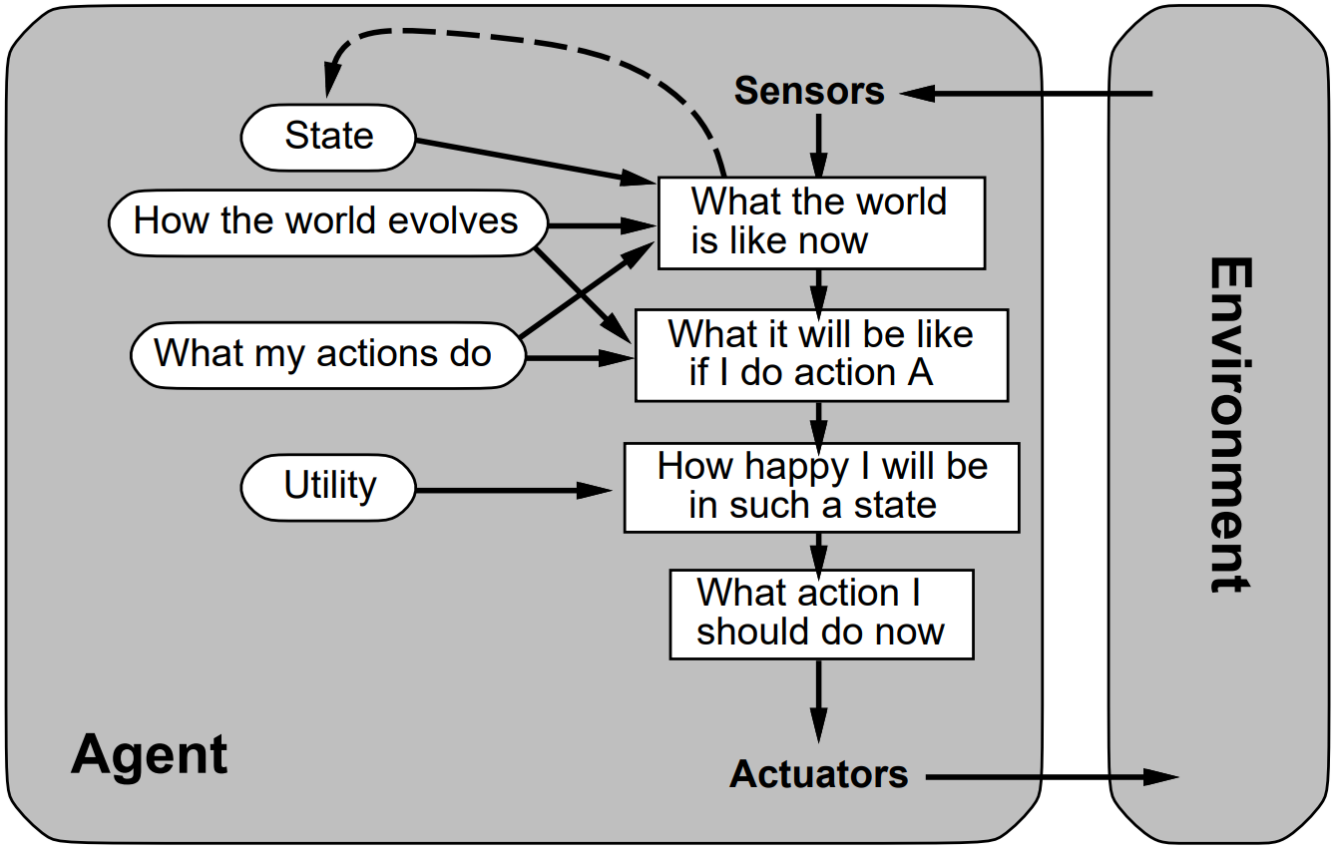
Cons: no way to express preferences relative to goals

Utility-based Agents

- There may be many ways to get to a destination but some may be better than others.
  - What about a relative measure of achievement?

- A utility function: an internalization of the performance measure
  - maps a state onto a real number
    - describes the "degree of happiness"
  - agent chooses the action that maximizes the expected utility
  - permits rational actions for more complex tasks
    - resolution of conflicts between goals (tradeoff)
    - multiple goals (likelihood of success, importance)
    - a utility function is necessary for rational behavior, but sometimes it is not made explicit

Utility-based Agents



Pros: effective for conflicting goals or trade-offs between goals

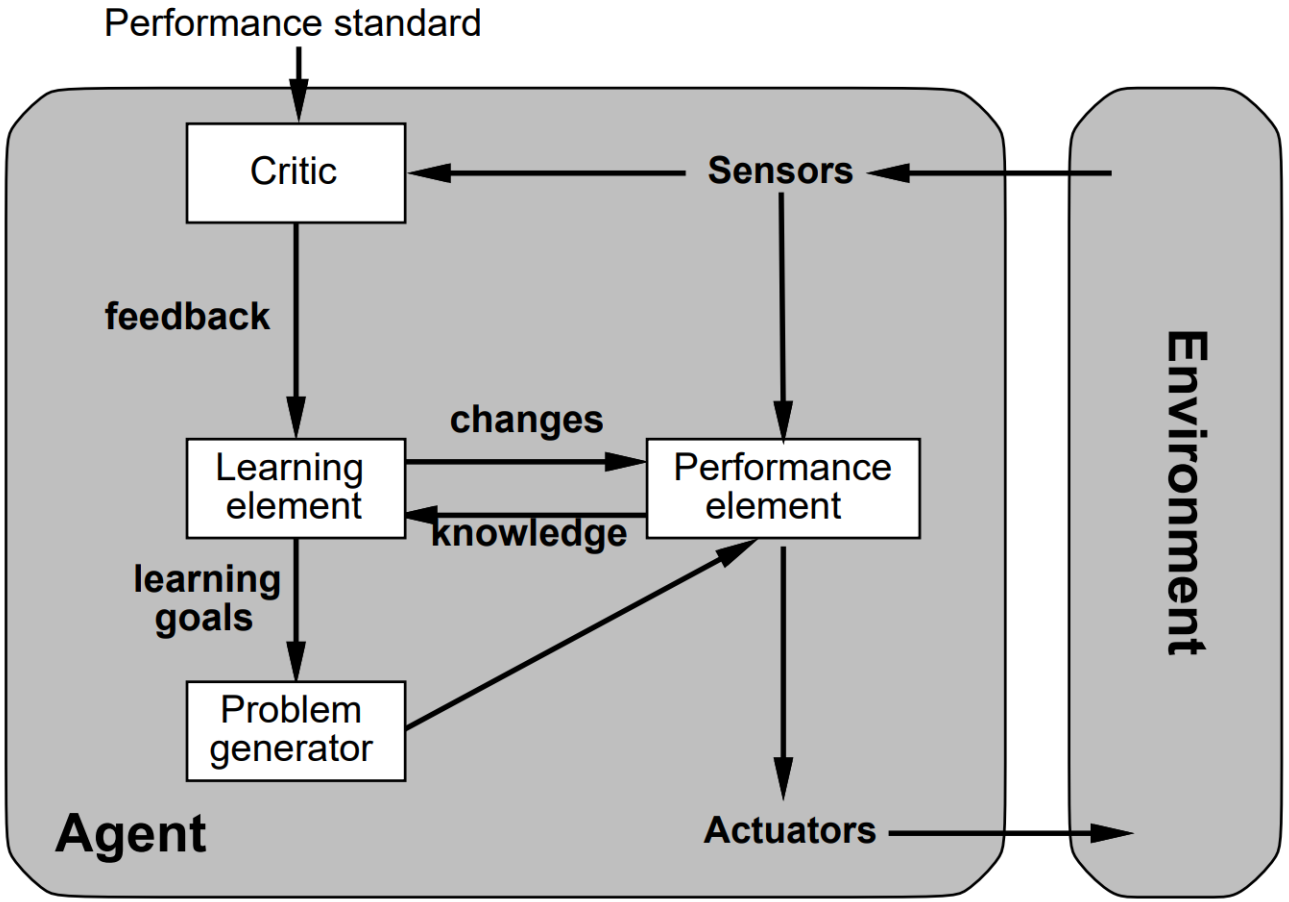
Cons: difficult to have a perfect utility function because of computational complexity

Learning Agents

A learning agent can learn from its experiences with the four components

- **performance element**
  - takes in percepts and decides on actions
  - can be one of the previously described agents
- **learning element**
  - evaluates the correctness of the performance element with the feedback from critic
- **critic**
  - provides feedback about the performance of the agent
  - can be external; sometimes part of the environment
- **problem generator**
  - suggests actions
  - required for novel solutions (creativity)

Learning Agents



Pros: improve performance over time in dynamic environments

Cons: need sufficient learning, complex to design and implement

When to Use Which Type of Agent?

- Depends on the problem (environment)
  - Fully observable/deterministic/stateful/adversarial …
- Depends the amount of data available
  - Often we need to learn how the world behaves
- Depends on the dimensionality of your observations

Summary

- agents perceive and act in an environment
  - Definition of Percept
  - Description of the environment (PEAS)
- ideal agents maximize their performance measure
  - Rationality
  - Performance Measure
  - Rational Agent
- basic agent types
  - simple reflex
  - Model-based reflex
  - goal-based
  - utility-based
  - learning
- some environments may make life harder for agents
  - inaccessible, non-deterministic, non-episodic, dynamic, continuous

What I want you to do

- Review Chapter 2
- Review Class Notes
- Read Chapter 3