

# CmpE 540

## Principles of Artificial Intelligence

Pınar Yolum  
[pinar.yolum@boun.edu.tr](mailto:pinar.yolum@boun.edu.tr)

Department of  
Computer Engineering  
Boğaziçi University

1

# Intelligent Agents

## Chapter 2

(Based mostly on the course  
slides from

<http://aima.cs.berkeley.edu/>

and

[http://www.cmpe.boun.edu.tr/~  
akin/](http://www.cmpe.boun.edu.tr/~akin/))

2

## Outline

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

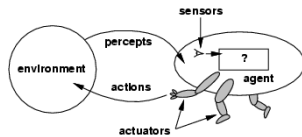
3

## Agents

- 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

4

## Agents and environments



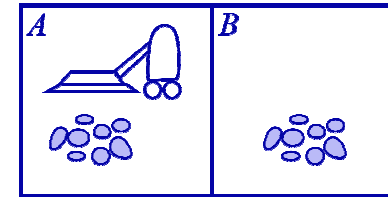
- The **agent function** maps from percept histories to actions:

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

- The **agent program** runs on the physical **architecture** to produce  $f$
- agent = architecture + program

5

## Vacuum-cleaner world



**Percepts:** location and contents, e.g., [A; Dirty]

**Actions:** *Left, Right, Suck, NoOp*

6

## 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

```
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
```

What is the **right** function?

Can it be implemented in a small agent program?

7

## Rationality

- A rational agent is one that does the **right thing**.
- More precisely, what is rational at any given time depends on four things:
  - The performance measure that defines the criterion of success.
  - The agent's prior knowledge of the environment.
  - The actions that the agent can perform.
  - The agent's percept sequence to date.

8

## Rational agents

- **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.

9

## 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: 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

10

## 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)

11

## 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

12

## 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

13

## Agent Characteristics

- **Situatedness:** The agent receives some form of sensory input from its environment, and it performs some action that changes its environment in some way. Examples of environments: the physical world and the Internet.
- **Autonomy:** The agent can act without direct intervention by humans or other agents and that it has control over its own actions and internal state.

14

## Agent Characteristics

- **Adaptivity :** The agent is capable of (1) reacting flexibly to changes in its environment; (2) taking goal-directed initiative (i.e., is pro-active), when appropriate; and (3) learning from its own experience, its environment, and interactions with others.
- **Sociability:** The agent is capable of interacting in a peer-to-peer manner with other agents or humans.

15

## Environment (1)

- Agents are situated in an environment
- Different environments (based on Russell and Norvig)
- Accessible environment:
  - If an agent can perceive everything happening around;
  - Agents have complete information.
  - Rarely happens in practice
  - Inaccessible environments: Internet, physical world

16

## Environment (2)

- Deterministic environment:
  - Each action has a guaranteed effect
  - An agent can determine the state of the world by knowing the state before an action happens and knowing the effect of the action
  - Physical world: Deterministic?

17

## Environment (3)

- History freedom:
  - Episode: Single cycle of an agent perceiving and taking an action
  - Episodic: If the choice depends on the current episode and not on previous episodes
    - Easier to operate
  - Sequential: History matters
    - Needs thinking ahead
    - Chess

18

## Environment (4)

- Static environment:
  - The environment only changes by the actions of the agent
  - Ex: Chess
- Dynamic environment:
  - Other agents' actions can change the environment
  - The environment itself can change over time
  - Ex: Internet

19

## Environment (5)

- Discrete environment:
  - Fixed, finite number of actions and percepts
  - Chess: At each time point, the number of actions is finite
- Continuous environment:
  - Not composed of discrete units
  - Either number of actions or percepts infinite
- Possible to convert continuous environments into discrete environments (with loss of precision)

20

## Environment (6)

- Single vs. Multiagent
- Open environment:
  - Autonomous entities
    - Can enter and leave frequently
    - Don't need to let anyone know
    - No control over what others will do
  - Cooperative entities
    - No use of sellers if they cannot cooperate with customers
    - Need for rules, regulations, contracts

21

## Environment types

	Solitaire	Backgammon	Internet Shopping	Taxi
Observable??				
Deterministic??				
Episodic??				
Static??				
Discrete??				
Single-agent??				

22

## Environment types

	Solitaire	Backgammon	Internet Shopping	Taxi
Observable??	Yes	Yes	No	No
Deterministic??	Yes	No	Partly	No
Episodic??	No	No	No	No
Static??	Yes	Semi	Semi	No
Discrete??	Yes	Yes	Yes	No
Single-agent??	Yes	No	Yes (except auctions)	No

23

## Agent functions and programs

- An agent is completely specified by the agent function mapping percept sequences to actions
- One agent function (or a small equivalence class) is rational
- Aim: find a way to implement the rational agent function concisely

24

## Table-lookup agent (1)

**function** TABLE-DRIVEN-AGENT( *percept*)  
**returns** an action

**static:**

*percepts*, a sequence, initially empty

*table*, a table of actions, indexed by percept sequences, initially fully specified

append *percept* to the end of *percepts*

*action* ← LOOKUP( *percepts*, *table*)

**return** *action*

25

## Table-lookup agent (2)

- Drawbacks:
  - Huge table
  - Take a long time to build the table
  - No autonomy
  - Even with learning, need a long time to learn the table entries

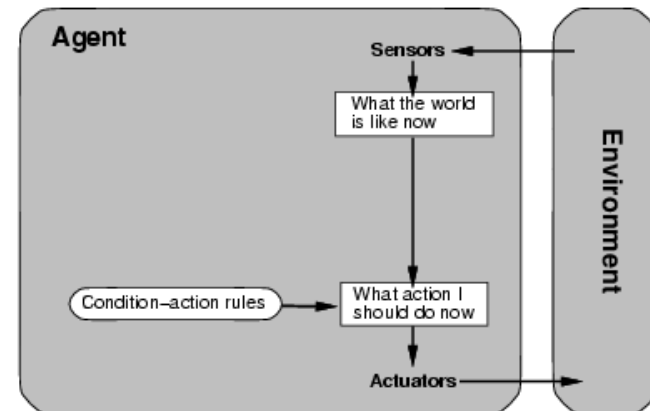
26

## Agent Types

- Four basic types in order of increasing generality:
  - simple reflex agents
  - reflex agents with state
  - goal-based agents
  - utility-based agents
- All these can be turned into learning agents

27

## Simple reflex agents

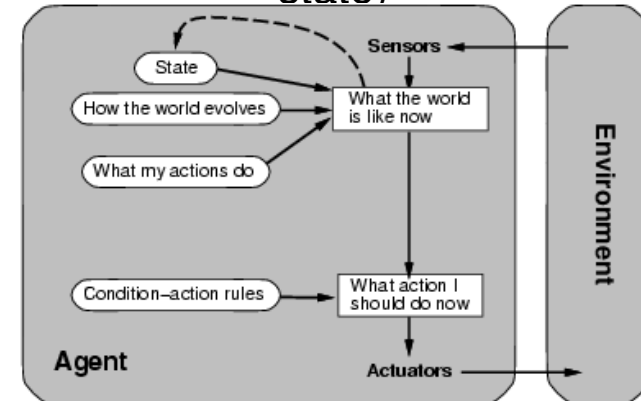


## Simple Reflex Agent

- Select actions based only on the *current* percept, ignoring the rest of the percept history.
- Table lookup of percept-action pairs defining all possible condition-action rules necessary to interact in an environment
- Problems
  - Too big to generate and to store (Chess has about  $10^{120}$  states, for example)
  - No knowledge of non-perceptual parts of the current state
  - Not adaptive to changes in the environment; requires entire table to be updated if changes occur
  - Looping: Can't make actions conditional

29

## Model-based reflex agents (with state)

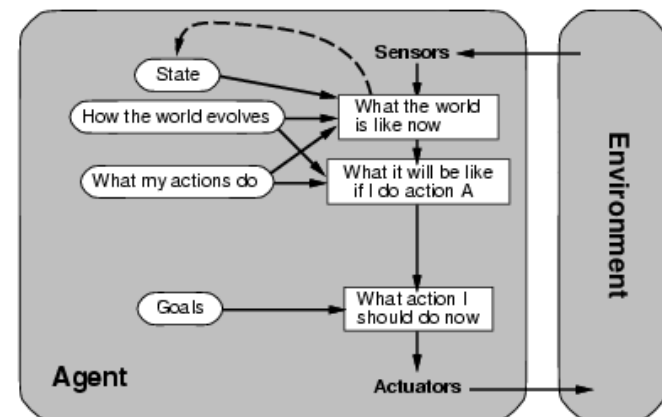


## Reflex Agent with State

- The knowledge about "how the world works" is called a **model** of the world.
- An agent that uses such a model is called a **model-based agent**.
- Encode "internal state" of the world to remember the past as contained in earlier percepts
- Needed because sensors do not usually give the entire state of the world at each input, so perception of the environment is captured over time. "State" used to encode different "world states" that generate the same immediate percept.
- Requires ability to represent change in the world; one possibility is to represent just the latest state, but then can't reason about hypothetical courses of action

31

## Goal-based agents



## Goal-Oriented Agents

- Add a goal
  - What is the agent trying to do?
  - Drive to Taksim; buy a book online
  - Perform actions that will realize the goal
- A set of condition-action rules is not enough
- Think about a goal state where the desired goal holds
- *Plan or search* a sequence of actions such that applying those actions will transform the current state into the goal state

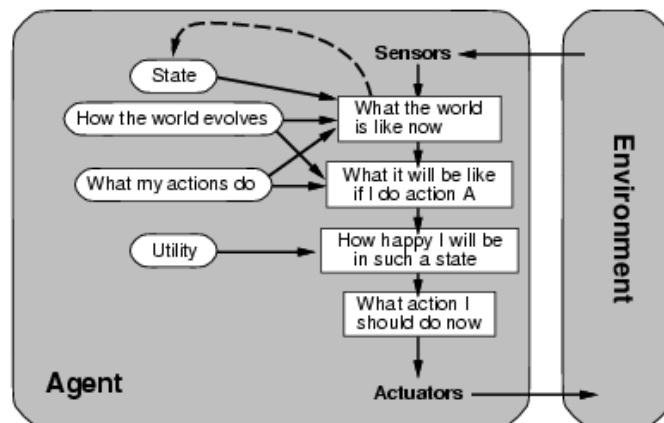
33

## Goal Based Agents

- Choose actions so as to achieve a (given or computed) goal= a description of a desirable situation
- Keeping track of the current state is often not enough--- need to add goals to decide which situations are good
- Deliberative instead of reactive
- May have to consider long sequences of possible actions before deciding if goal is achieved--- involves consideration of the future, "what will happen if I do...?"

34

## Utility-based agents



## Utility-Based Agents (1)

- A goal-oriented agent may achieve its goals in a number of ways.
- Nothing is said about whether one way is better than another
- However, one path can be preferred over another because it's shorter, cheaper, etc.
- **Utility**:
  - Quantitative measure of a chosen path
  - A function from states to real numbers

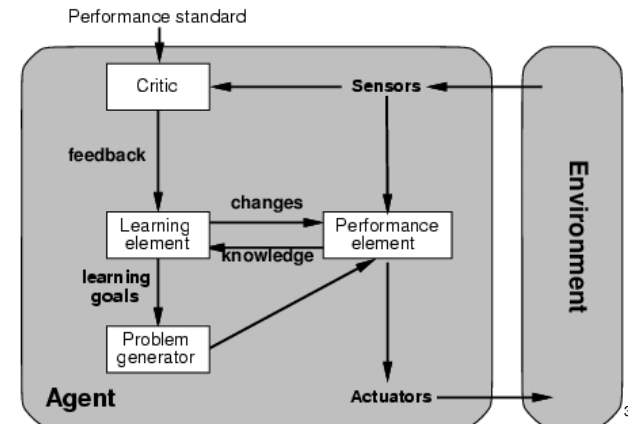
36

## Utility-Based Agents (2)

- Agents may have conflicting goals\*
  - Buy a book with a low cost and buy it fast
  - Which goal should have precedence over the other?
- Assign a utility to each goal
- Maximize expected utility

37

## Learning agents



38

## Components of a Learning Agent

- **learning element**, which is responsible for making improvements,
- **performance element**, which is responsible for selecting external actions. The performance element is the entire agent: it takes in percepts and decides on actions.
- **critic** gives feedback from the on how the agent is doing.
- **problem generator** is responsible for suggesting actions that will lead to new and informative experiences.

39

## Developing Agents (1)

- Depends on what is expected of the agent
- An agent that interacts with other agents over the Web
  - JAVA!
  - Better support for Web standards
  - Easy integration with other programming environments
- An agent that processes logical formulae (logic-based agents)
  - Prolog
  - Java+Prolog

40

## Developing Agents (2)

- Agent0, GOLOG, 3APL
  - Agent programming languages
  - Constructs for beliefs, goals, etc.
  - Specify rules to reason on these (update, delete)
  - Use an interpreter which will track which rules are fired and make modifications accordingly

41

## Developing Agents (3)

- Java Agent DEvelopment Framework (JADE)
  - FIPA-compliant
  - Available libraries for agent templates
- MadKit
  - Agents programmed in Java
  - Enforce an organization model so that each agent plays a role in the MAS
- OAA, JATLite, IBM Able

42