MONASH University
Information Technology

MONASH University
Information Technology

FIT3080 – Intelligent Systems

Intelligent Agents
Chapter 2

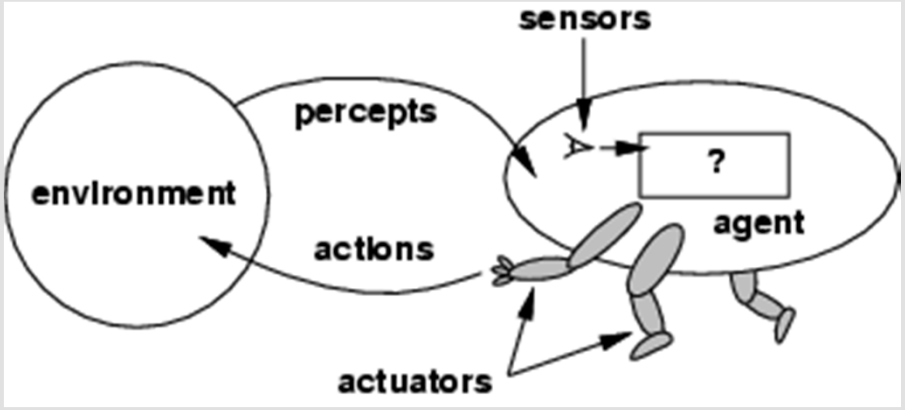
Outline

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

Agents

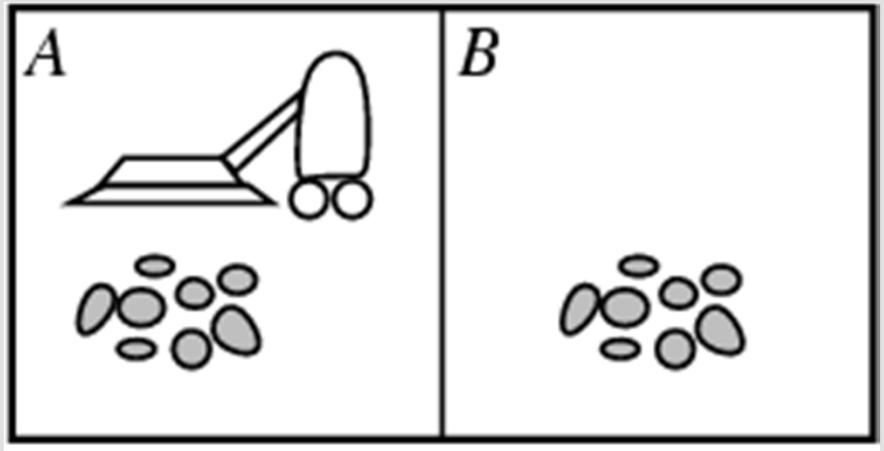
- **An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators**
- **Human agent:**
  - eyes, ears and other organs for sensors
  - hands, legs, mouth and other body parts for actuators
- **Robotic agent:**
  - cameras and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

Example: Vacuum-cleaner World and Agent



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

  If *status=Dirty* **return** Suck

  ElseIf *Location=A* **return** Right

  ElseIf *Location=B* **return** Left

Rationality and Rational Agents

- **Rationality depends on**
  - Performance measure
  - The agent's prior knowledge of the environment
  - The actions that the agent can perform
  - The percept sequence to date
- **Definition:**
  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 the agent's built-in knowledge

Rational Agents

- **Rationality is NOT omniscience**
- **Agents can perform actions to modify future percepts in order to obtain useful information → exploration, learning**
- **An agent is autonomous if its behavior is determined by its own experience**

Task Environment – PEAS

To design a rational agent, we must specify the *Task Environment*

- **PEAS**
  - Performance measure
  - Environment
  - Actuators
  - Sensors

PEAS – Example (I)

Automated taxi driver:

- **Performance measure**
  - Safe, fast, legal, comfortable trip, minimize fuel consumption, maximize profits
- **Environment**
  - Road types, road contents, customers, operating conditions
- **Actuators**
  - Control over the car, communication with other vehicles and passengers
- **Sensors**
  - Cameras, sonar, speedometer, GPS, odometer, engine sensors, speech recognizer

PEAS – Example (II)

Internet shopping agent:

- **Performance measure**
  - cheap, good quality, appropriate product
- **Environment**
  - current WWW sites, vendors
- **Actuators**
  - display to user, follow URL, fill in form
- **Sensors**
  - HTML pages (text, graphics, scripts)

Environment Types (I)

The environment type largely determines the agent design

- **Fully (partially) observable – An agent's sensors give it access to the complete state of the environment at all times**
- **Known (unknown) – An agent knows the "laws" of the environment**
- **Single (multi) agent – An agent operating by itself in an environment**
- **Deterministic (stochastic) – The next state is completely determined by the current state and the action executed by the agent**

Environment Types (II)

- **Episodic (sequential) – The agent's experience is divided into atomic *episodes*. The next episode does NOT depend on previous actions**
  - In each episode an agent perceives a percept and performs a single action
- **Static (dynamic) – The environment is unchanged while an agent is deliberating**
- **Discrete (continuous) – Pertains to number of states, the way time is handled, and number of percepts and actions**
  - E.g., state may be continuous, but actions may be discrete

Environment Types – Examples

	Sorting laundry	8-puzzle	Back-gammon	Medical diagnosis	Taxi
Observable?					
Known?					
Single agent?					
Deterministic?					
Episodic?					
Static?					
Discrete?					

The real world is partially observable, unknown, multi-agent, stochastic, sequential, dynamic, continuous

Environments and Methodologies

	Search	Logical inference	Bayesian networks	Markov decision processes	Reinfor-cement learning
Observable?	✓	✓		✓	✓
Known?	✓	✓	✓	✓	×
Single agent?					
Deterministic?	✓	✓	×	×	×
Episodic?					
Static?	✓	✓	✓	✓	✓
Discrete?	✓	✓	✓	✓	✓

Agent Functions and Programs

- **An agent is completely specified by the *agent function* that maps percept sequences to actions**
- **Aim: design a program that implements the rational agent function concisely**

Agent Types

Based on the function = how actions are selected

Agent Type	Action selected based on
Simple reflex	current percept
Model based	+ internal state
Goal based	+ goal
Utility based	+ utility function
Learning	

How Components of Agent Programs Work?

Depends on the representations of states:

- **Atomic – each state is indivisible (Search, Game playing, Markov Decision Processes)**
- **Factored – splits each state into attributes, each of which has a value (Propositional logic, Planning, Bayesian Networks, Machine learning)**
- **Structured – represents how things are related to each other (First order logic, Bayesian networks, Semantic networks)**

Summary (I)

- **Agents interact with environments through actuators and sensors**
- **Agent function – describes what the agent does in all circumstances**
- **Agent program – implements the agent function in an architecture**
- **Performance measure – evaluates the behaviour of an agent in an environment**
- **A perfectly rational agent maximizes expected value of the performance measure**

Summary (II)

- **PEAS define task environments:**
  - performance, environment, actuators, sensors
- **Environments are categorized along several dimensions:**
  - observable? known? single-agent? deterministic? episodic? static? discrete?
- **Basic agent architectures:**
  - reflex, reflex with state, goal-based, utility-based
  - learning
- **Types of states:**
  - atomic, factored, structured

Reading

- **Russell, S. and Norvig, P. (2010), *Artificial Intelligence* – Chapter 2**

Next Lecture Topic

- **Lecture Topic 2**
  - Problem Solving as Search