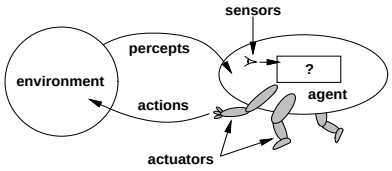
Intelligent Agents

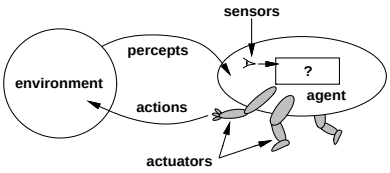
Russell and Norvig: 2

Agents



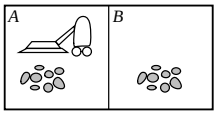
- Agent – perceives the environment through sensors and acts on it through actuators
- Percept – agent's perceptual input (the basis for its actions)
- Percept Sequence – complete history of what has been perceived.

Agent Function



- Agent Function – maps a give percept sequence into an action; describes what the agent does.
- Externally – Table of actions
- Internally – Agent Program

Vacuum Cleaner World

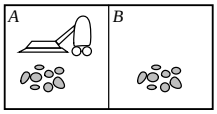


- Percepts: which square (A or B); dirt?
- Actions: move right, move left, suck, do nothing
- Agent function: maps percept sequence into actions
- Agent program: function's implementation
- How should the program act?

Agent Characterization

- Meant to be a tool for analyzing systems – not characterizing them as agent versus non-agent
- Lots of things can be characterized as agents (artifacts) that act on the world
- AI operates where
  - Artifacts have significant computational resources
  - Task Environments require nontrivial decision making

Vacuum Cleaner World



- Percepts: which square (A or B); dirt?
- Actions: move right, move left, suck, do nothing
- Agent function: maps percept sequence into actions
- Agent program: function's implementation
- How should the program act?

Rational Agent – does the right thing

What does that mean? One that behaves as well as possible given the Environment in which it acts. How should success be measured? On consequences.

- Performance measure
  - Embodies criterion for success
    - Amount of dirt cleaned?
    - Cleaned floors?
  - Generally defined in terms of desired effect on environment (not on actions of agent)
  - Defining measure not always easy!

Rationality Depends on:

1. Performance measure that defines criterion for success.
2. Agent's prior knowledge of the environment.
3. Actions the agent can perform.
4. Agent's percept sequence to date.

*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

*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.*

- Notice the rationality is dependent on EXPECTED maximization.
- Agent might need to learn how the environment changes, what action sequences to put together, etc…

Rationality

*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.*

- Notice that an agent may be rational because the designer thought of everything, or it may have learned it itself (more autonomous)

Task Environment

- The "problems" for which rational agents are the "solutions"

PEAS Description of Task Environment

- Performance Measure
- Environment
- Actuators (actions)
- Sensors (what can be perceived)

Properties of Task Environments (affect appropriate agent design)

- **Fully observable vs partially observable**
  - Fully observable gives access to complete state of the environment
  - Complete state means aspects relevant to action choice
  - global vs local dirt sensor

Properties of Task Environments (affect appropriate agent design)

- **Single Agent vs Multi-agent**
  - Single Agent – crossword puzzle
  - Multi-agent – chess, taxi driving? (are other drivers best described as maximizing a performance element?)
  - Multi-agent means other agents may be competitive or cooperative and may require communication
  - Multi-agent may need communication

Properties of Task Environments (affect appropriate agent design)

- **Deterministic vs Stochastic**
  - Deterministic – next state completely determined by current state and action
  - Uncertainty may arise because of defective actions or partially observable state (i.e., agent might not see everything that affects the outcome of an action).

Properties of Task Environments (affect appropriate agent design)

- **Episodic vs Sequential**
  - Episodic the agent's experience divided into atomic episodes
  - Next episode not dependent on actions taken in previous episode. E.g., assembly line
  - Sequential – current action may affect future actions. E.g., playing chess, taxi
  - short-term actions have long-term effects
  - must think ahead in choosing an action

Properties of Task Environments (affect appropriate agent design)

- **Static vs Dynamic**
  - does environment change while agent is deliberating?
  - Static – crossword puzzle
  - Dynamic – taxi driver

Properties of Task Environments (affect appropriate agent design)

- **Discrete vs Continuous**.

Can refer to

  - the state of the environment (chess has finite number of discrete states)
  - the way time is handled (taxi driving continuous – speed and location of taxi sweep through range of continuous values)
  - percepts and actions (taxi driving continuous – steering angles)

Properties of Task Environments (affect appropriate agent design)

- **Known vs Unknown**
  - This does not refer to the environment itself, but rather the agent's knowledge of it and how it changes.
  - If unknown, the agent may need to learn

Properties of Task Environments (affect appropriate agent design)

- Easy: Fully observable, Deterministic, Episodic, Static, Discrete, Single agent.
- Hard: Partially observable, Sochastic, Sequential, Dynamic, Continuous, Multi-Agent

Environment types

	Chess with a clock	Chess without a clock	Taxi driving
Fully observable	Yes	Yes	No
Deterministic	Strategic	Strategic	No
Episodic	No	No	No
Static	Semi	Yes	No
Discrete	Yes	Yes	No
Single agent	No	No	No

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

Agent Programs

- Need to develop agents – programs that take the current percept as input from the sensors and return an action to the actuators.

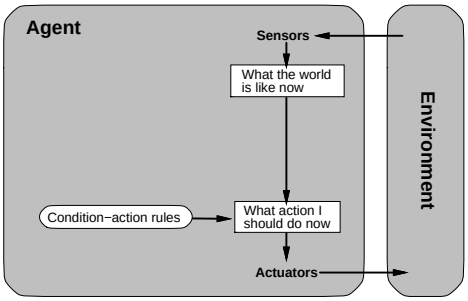
Possible Agent Program

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*
⋮	⋮

Agent Programs

- Need to develop agents – programs that take the current percept as input from the sensors and return an action to the actuators.
- The key challenge for AI is to find out how to write programs that, to the extent possible, produce rational behavior from a small amount of code.

Simple Reflective Agent



Simple Reflexive Agent

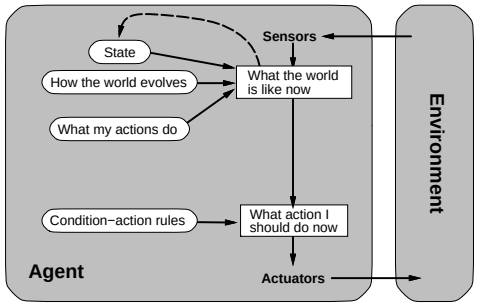
- Handles simplest kind of world
- Agent embodies a set of condition-action rules
- If percept then action
- Agent simply takes in a percept, determines which action could be applied, and does that action.
- NOTE:
- Action dependent on current percept only
- Only works in fully observable environment

Simply Reflexive Vacuum Agent

- Implements the agent function (described in earlier table)

```
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
```

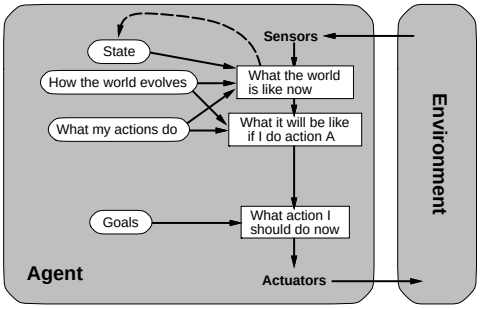

Model-Based Reflex Agent



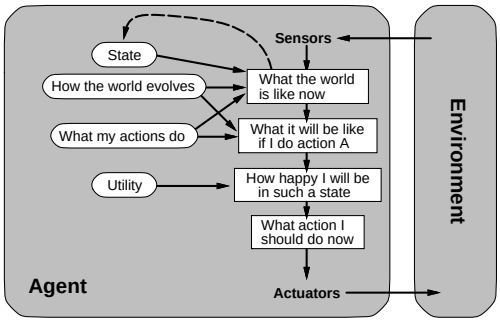
Model-Based Reflex Agent

- Upon getting a percept
  - Update the state (given the current state, the action you just did, and the observations)
  - Choose a rule to apply (whose conditions match the state)
  - Schedule the action associated with the chosen rule

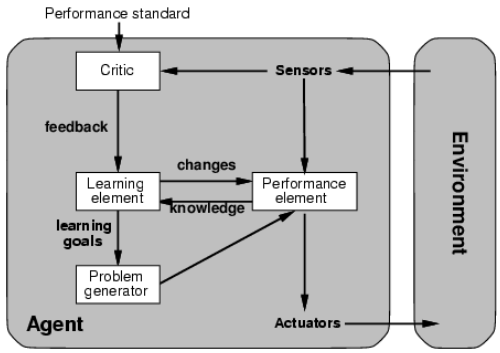
Goal Based Agent



Utility-Based Agent



Learning agents



Learning Agent Components

1. Learning Element – responsible for making improvements (on what ever aspect is being learned…)
2. Performance Element – responsible for selecting external actions. In previous parts, this was the entire agent!
3. Critic – gives feedback on how agent is going and determines how performance element should be modified to do better in the future
4. Problem Generator – suggests actions for new and informative experiences.

Summary Chapter 2

- Agents interact with environments through actuators and sensors
- The agent function describes what the agent does in all circumstances.
- The performance measure evaluates the environment sequence.
- A perfectly rational agent maximizes expected performance.
- Agent programs implement (some) agent functions.

Summary (cont)

- 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