Coping with a Changing World:

Structure of Intelligent Agents and Environments

Alan Bundy

School of informatics

(slides courtesy of Bonnie Webber)

Structure of Intelligent Agents

An **agent** is anything that can be viewed as **perceiving** its environment through **sensors** and **acting** on that environment through **effectors**.

What we are concerned with is the **match** between **agent** properties (i.e., what an agent can perceive, how it can act, and what it supposed to achieve) and **environment** properties.

Agent: Mail sorting robot
Percepts: array of pixel intensities
Actions: route letter into bin
Goals: route letter into correct bin
Environment: conveyor belt of letters

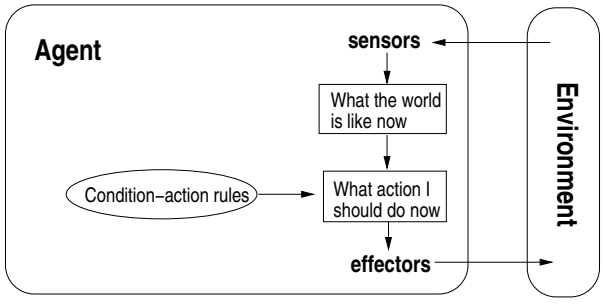
Agent: Intelligent house
Percepts: signals from temperature sensor, movement sensor, clock, sound sensor
Actions: room heaters on/off, lights on/off
Goals: occupants warm, rooms light when occupied, house energy efficient
Environment:
at various times, occupants enter and leave house, enter and leave rooms; daily variation in outside light and temperature

Agent: Car driver
Percepts: camera (array of pixels of various intensities), signals from GPS, speedometer, sonar
Actions: steer, accelerate, brake
Goals: safe, fast, legal trip
Environment:
streets, intersections, traffic signals, traffic lights, other moving vehicles, pedestrians

Simple Reflex Agents

For some agent-environment pairs, the choice of appropriate action depends only on the agent's immediate percepts.

This can be effected through **condition-action rules** in a simple reflex agent.



Agent: Mail sorting robot
Environment: Conveyor belt of letters

Agent: Breakfast robot??
Environment: Kitchen (But for what sorts of breakfasts: cereal&milk?? fried eggs?? tea?? What sorts of condition-action rules??)

function SIMPLE-REFLEX-AGENT(*percept*)
returns *action*
static: *rules*, a set of condition-action rules

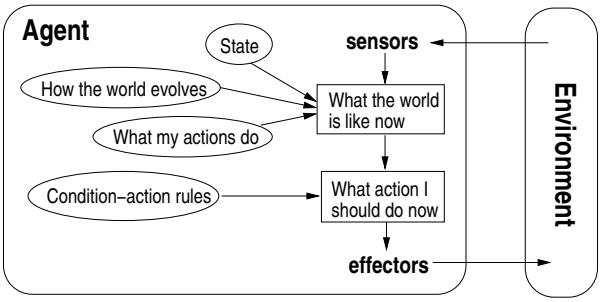
state ← INTERPRET-INPUT(*percept*)
rule ← RULE-MATCH(*state, rules*)
action ← RULE-ACTION[*rule*]
return *action*

Model-Based Reflex Agents

For some agent-environment pairs, the choice of appropriate action requires maintaining some knowledge of the past in the form of *a model*.

Why? Significantly different states of the environment – ones requiring different actions – may present the agent with the same perceptual input. Maintaining *internal state* may allow the agent to distinguish between them.

How much internal state is needed and what changes what is stored there, depends on the task/goal.



Compare

- taking two eggs out of a box and adding them to a pot of water;
- using a teaspoon to take two teaspoonfuls of salt out of a jar and add it to a pot of water.

Is a model needed? How complex?

Externalised state is sometimes a possible alternative to an internal model – changing the environment so that different states of the environment present different perceptual inputs.

That is what happening with the simple breakfast robot. If it is possible to externalise state, a simple reflex agent is all that is needed.

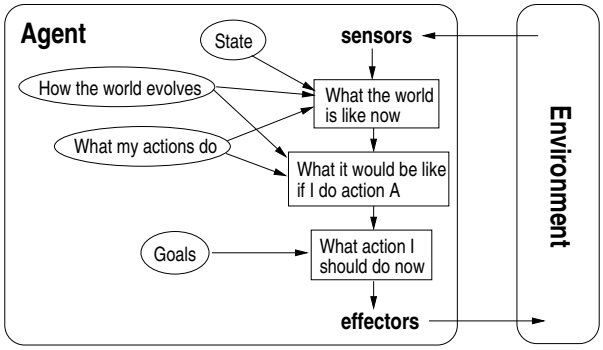
function REFLEX-AGT-WITH-MODEL(*prcpt*)
returns *action*
static: *model*, current world state (description)
rules, set of condition-action rules
K_{ev}, rules about how world evolves
K_{act}, description of *rules*

model ← UPDATE-MODEL(*model, prcpt,* K_{ev}, K_{act})
rule ← RULE-MATCH(*model, rules*)
action ← RULE-ACTION[*rule*]
model ← UPDATE-MODEL(*model, action*)
return *action*

Goal-Based Agents

Knowing the current state of the environment is not always enough to decide what to do.

Goal information specifies **desirable** situations. It is **not** used to gauge the **current state** of the environment ("Oh, isn't this delightful!"), but rather to assess whether any **currently possible action** can get the agent to a desirable state. So goal information is used as a **filter** on possible actions.



Agents can also **plan** to achieve goals through sequences of actions and monitor their progress using their **sensors**.

This is what AI **planning** is about. We will talk about this more when we discuss situation calculus and STRIPS.

Does either way of using goals require a model?

Does it make sense to consider agents with goals but without world models?

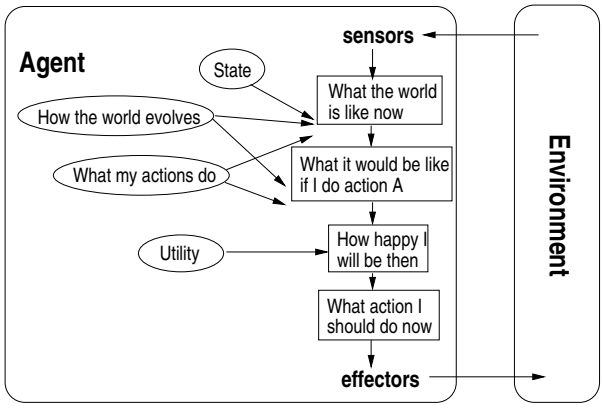
In simple reflex agents, goals are essentially compiled into their condition-action rules.

Utility-Based Agents

Goals provide a way of classifying states (i.e., into **discrete** classes). **Utility** allows quantifying how good a state is (i.e., a **continuous** metric). This allows an agent to

- better compare the states resulting from actions when deciding what to do;
- decide what to do when the consequences of actions are not certain.

How? Utility can be combined with **probability** yielding (**expected utility**) in a way that allows an agent to weigh the likelihood of success against the importance of goals.



We will not be covering utility-based agents in this module, but this topic is discussed in Russell & Norvig, Chapters 16 and 17 (1st and 2nd editions).

Learning Agents

How do agents improve their performance in the light of experience?

- Generate problems which will test performance.
- Perform activities according to rules, goals, model, utilities, *etc.*
- Monitor performance and identify non-optimal activity.
- Identify and implement improvements.

We will not be covering learning agents in this module, but this topic is discussed in Russell & Norvig, Chapters 18-21 (1st and 2nd editions) and AI2Bh Task 5.

Mid Lecture Exercise

Consider a robot vacuum cleaner. What sort of agent would it need to be?

Types of Environment 1

There are different sorts of environments, which affect what an agent has to be able to cope with.

In designing agents, one should always consider the pair of agent and environment together.

- **Fully Observable** vs. **Partially Observable**: If an agent's sensors give it full access to the complete state of the environment, the environment is fully observable, otherwise it is only partially observable or unobservable.

  Example?
- **Deterministic** vs. **Stochastic**: If the next state of the environment is completely determined by the current state and the agent's selected action, the environment is deterministic. An environment may *appear* stochastic if it is only partially observable.

  Example?

Types of Environment 2

- **Episodic** vs. **Sequential**: If future decisions do not depend on the actions an agent has taken, just the information from its sensors about the state it is in, then the environment is episodic.

  Example?
- **Static** vs. **Dynamic**: If the environment can change while the agent is deciding what to do, the environment is dynamic.

  Example?

Types of Environment 3

- **Discrete** vs. **Continuous**: If the sets of percepts and actions available to the agent are finite, and the individual elements are distinct and well-defined, then the environment is discrete.

  Example?

- **Single Agent** vs. **Multiagent**: Must other entities in the environment be modelled as agents? Are they cooperative or competitive?

  Example?

Types of Environment 4

The "changing world" that an agent has to cope with, might have any combination of these properties – from "benign" (i.e., fully observable, deterministic, episodic, static, discrete and single agent) to "chaotic" (i.e, partially observable, stochastic, sequential, dynamic, continuous and multiagent).

What are the properties of the environment that would be experienced by a mail-sorting robot? an intelligent house? a car-driving robot?

Summary

We have discussed

- simple reflex agents
- model-based reflex agents
- goal-based agents
- utility-based agents
- learning agents
- properties of environments