Agent Issues

Issues about Agent

Concept of rationality

Performance Measure

Software Agent

Task Environments

Concept of Rationality

Rational agent

- Do correct thing
- Every action (output) is correct /rational

Correctness for an action

- Lead the agent to be most successful under given objective
- Need ways to measure success / correctness
  - *Performance measure*

Performance measure

An objective function

- Determines how successful the agent is

An agent

- Based on its percepts
- Returns action sequence
- If the sequence is desirable (actions → high score)
  - Agent is performing well
- No universal performance measure
- Different objectives have different measures

Performance measure

Design performance measure

- What is needed in the environment – Results – (rationally)
- Not how the agent should behave – Steps – (humanly)

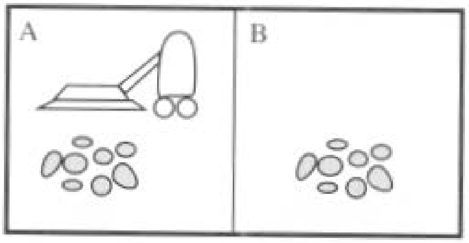
E.g. in vacuum-cleaner world

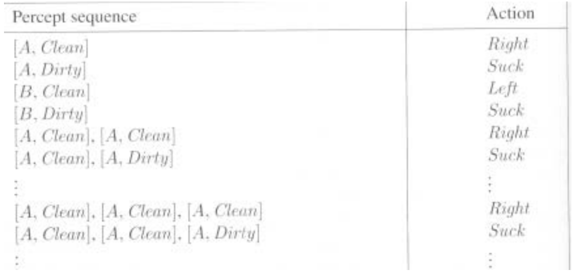
- Want the floor clean
  - no matter how the agent behave
  - just care the results
- No restriction on agent’s behavior
- Just define the results we desire

Rationality

Rationality depends on

- Performance measure that defines criterion of success
- Prior knowledge of the environment (percept)
- Actions can perform (action)
- Percept sequence up to now (memory)





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*
⋮	⋮
[*A*, *Clean*], [*A*, *Clean*], [*A*, *Clean*]	*Right*
[*A*, *Clean*], [*A*, *Clean*], [*A*, *Dirty*]	*Suck*
⋮	⋮

Rational agent

Given percept sequence

Select an action

- expected to maximize its performance measure

Based on

- Current percept and percept sequence
- Built-in knowledge

Example

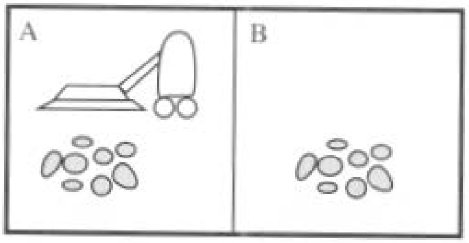
Vacuum Cleaner World as a relational agent

Performance measure

- Run 10,000 time steps
  - Each time step awards 1 point if square is clean
  - 2 squares, highest = 10,000 * 2 = 20,000 points

Prior knowledge about the environment

- Geography of environment
- 2 squares



Example

Actions that can perform

- Left, Right, Suck and NoOp
- Its effects

Current percept, percept sequences

- Location
- Cleanliness of current location

With these 4 items

- Agent is rational
- But not Omniscience
  - cannot ensure actions are *actually* correct

Software Agent

Environment may not be the real world

- Flight simulator, video games, Internet
- Artificial and very complex environments
- Hardware agents cannot work inside
- Software agent (soft robots)
  - All parts are software
  - Enemies in game, search engine

Task environment

Rational agent

- Designed to work on the environment

Environment is studied and specified

- P.E.A.S. description
  - **P**erformance measure (Objective)
  - **E**nvironment (Domain knowledge)
  - **A**ctuators (Actions)
  - **S**ensors (Percepts)

Example

Automated taxi driver

Performance Measure

- Judge automated driver
- Factors considered
  - Correct destination
  - Fuel consumption
  - Trip time and/or cost
  - Safety and comfort
  - Violations of traffic laws
  - etc

Example

Environment

- Interaction with customer
- Variety of roads
- Traffic lights, other vehicles, pedestrians, road works, police cars, etc.

"Where to Mister?"

Example

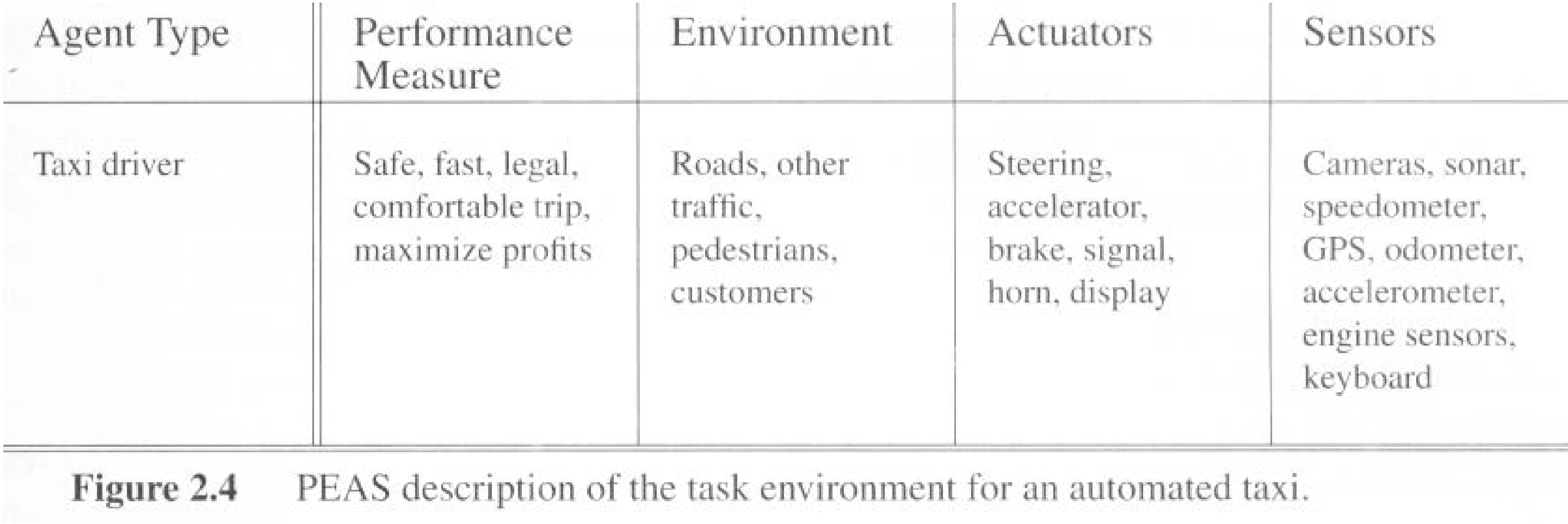
Actuators (outputs)

- Display (communicate with customers)
- Steering, accelerator, gear shifting and braking

Sensors (inputs)

- GPS
- Detect other vehicles, road situations
- Many more devices

Sketch of example



Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi driver	Safe, fast, legal, comfortable trip, maximize profits	Roads, other traffic, pedestrians, customers	Steering, accelerator, brake, signal, horn, display	Cameras, sonar, speedometer, GPS, odometer, accelerometer, engine sensors, keyboard

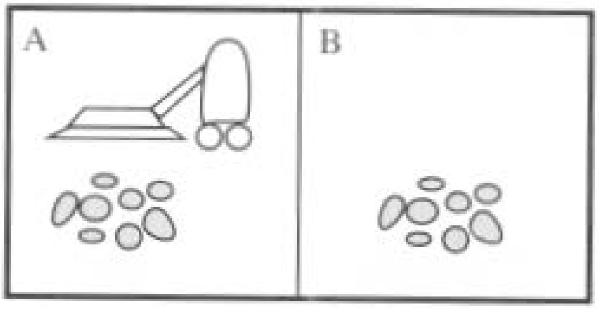
Figure 2.4 PEAS description of the task environment for an automated taxi.

Properties of Environment

Observable

Fully observable

- Sensors → Complete state of the world (all the 2 squares)
- Effectively and fully observable
  - Detect all aspects relevant to action choice



Partially observable

- Local dirt sensor
- Do not know cleanliness of other squares

Deterministic VS Stochastic

Deterministic

- Next state of environment determined by
    - Current state
    - Agent’s actions

Strategic

- Deterministic
- Actions from other agents

Stochastic

- Some unobservable aspects
- Cleaner and taxi driver

Episodic VS Sequential

Episodic

- An episode
  - Single pair of perception & action
- Quality of action
  - Not depend on other episodes
- Every episode is independent of each other
- Simpler, no need to think ahead

Sequential

- Current action may affect all future decisions
- Chess / games

Static VS Dynamic

Static

- Do not change over time

Dynamic

- Always changing over time

Semidynamic

- Environment is not changed over time
- But performance score does
  - E.g. earlier achieve the goal, higher awards

Discrete VS Continuous

Discrete

- Percepts and actions
  - Limited number
  - Distinct, clearly defined
- Vacuum cleaner: 2 squares, dirt or not, etc.

Continuous

- 0~1, in between there may exist real numbers

Single Agent VS Multiagent

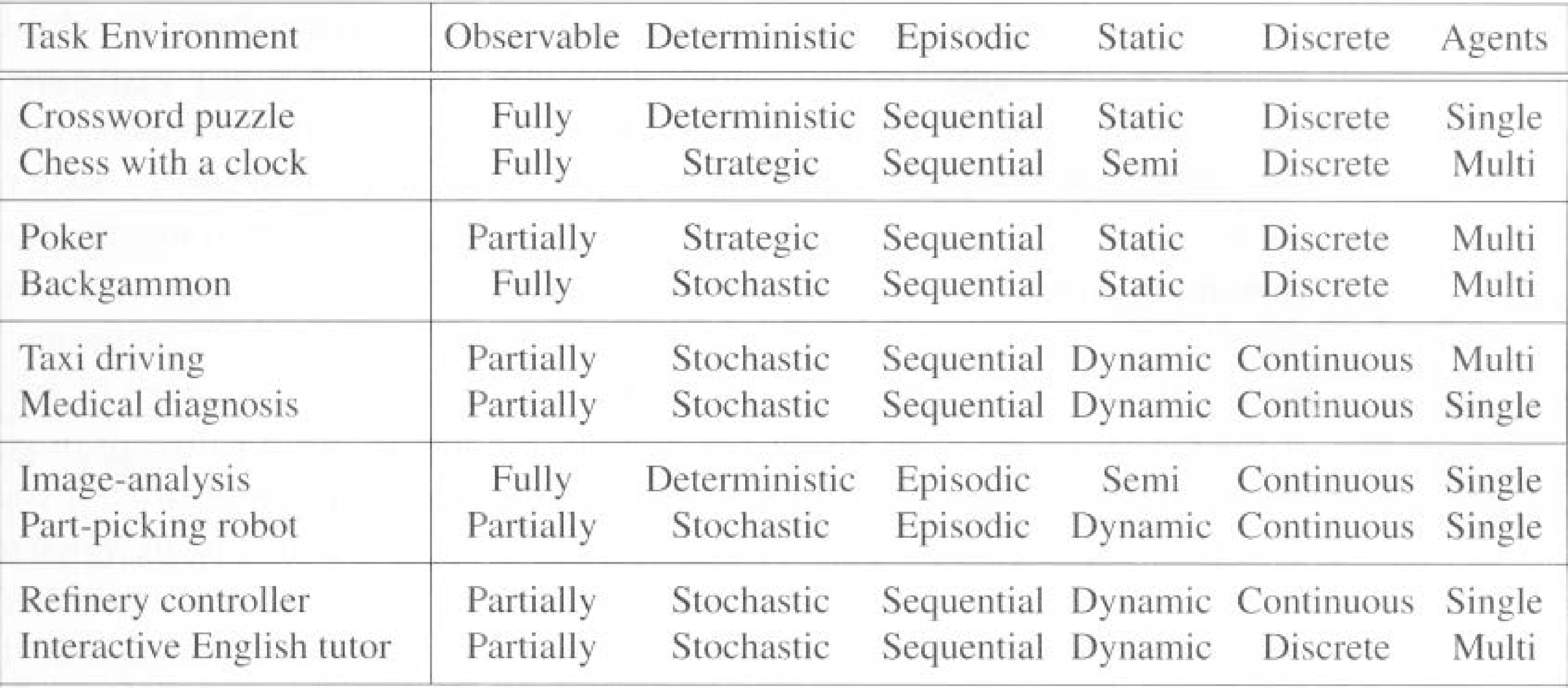
Single Agent

- Only one agent
- Playing crossword puzzle

Multiagent

- More than one agent
- Chess playing
- *Competitive multiagent environment*
  - Chess playing
- *Cooperative multiagent environment*
  - Automated taxi driver

Examples of environments



Task Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Crossword puzzle	Fully	Deterministic	Sequential	Static	Discrete	Single
Chess with a clock	Fully	Strategic	Sequential	Semi	Discrete	Multi
Poker	Partially	Strategic	Sequential	Static	Discrete	Multi
Backgammon	Fully	Stochastic	Sequential	Static	Discrete	Multi
Taxi driving	Partially	Stochastic	Sequential	Dynamic	Continuous	Multi
Medical diagnosis	Partially	Stochastic	Sequential	Dynamic	Continuous	Single
Image-analysis	Fully	Deterministic	Episodic	Semi	Continuous	Single
Part-picking robot	Partially	Stochastic	Episodic	Dynamic	Continuous	Single
Refinery controller	Partially	Stochastic	Sequential	Dynamic	Continuous	Single
Interactive English tutor	Partially	Stochastic	Sequential	Dynamic	Discrete	Multi

Figure 2.6 Examples of task environments and their characteristics.

Types of Agent Programs

Types of Agent Programs

Simple reflex agents

Model-based reflex agents

Goal-based agents

Utility-based agents

Learning agents

Simple Reflex Agents

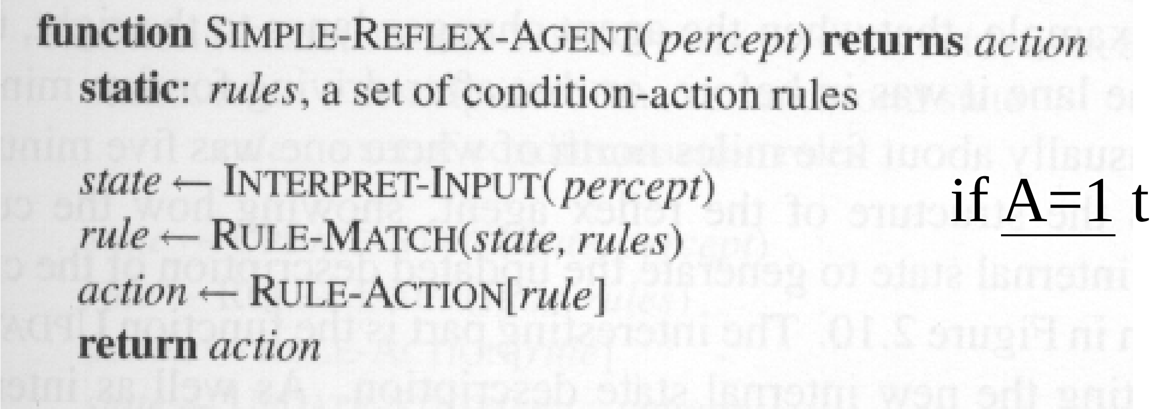
Just condition-action rules

- In the form “if … then …”

Efficient

Narrow range of applicability

- Some knowledge (rules) cannot be stated explicitly
- Work only if the environment is fully observable

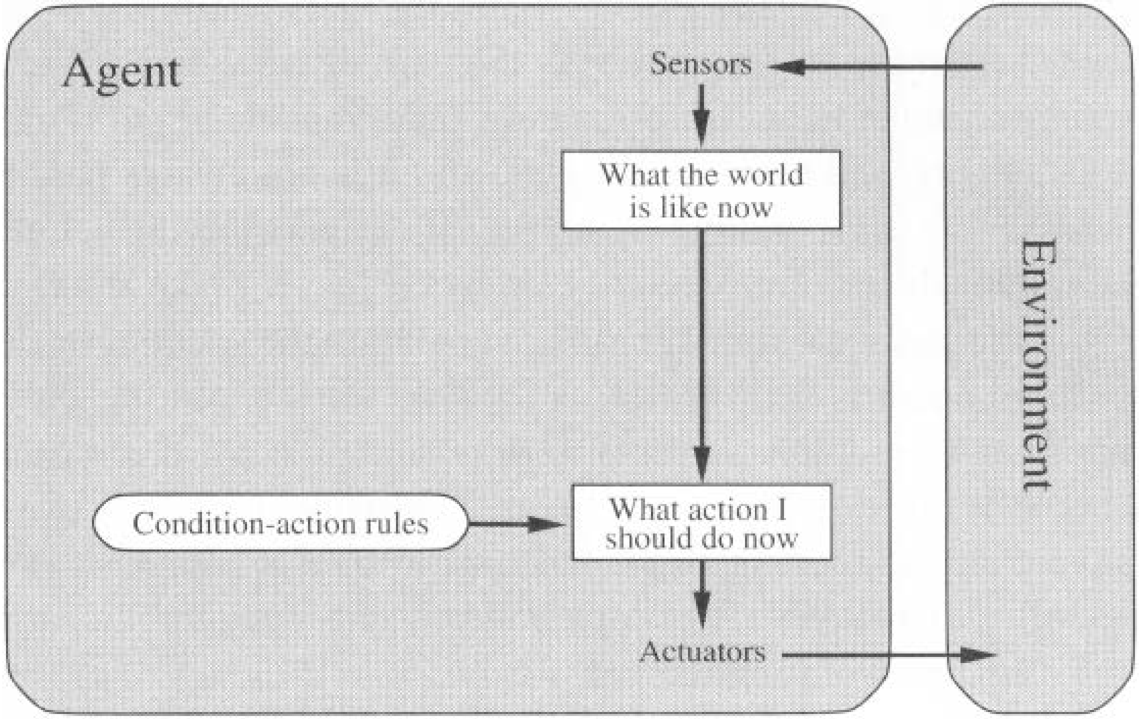


```
function SIMPLE-REFLEX-AGENT(percept) returns action
    static: rules, a set of condition-action rules

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

if A=1 then do;

Simple Reflex Agents



Model-based Reflex Agents

For partially observable environment

Internal state

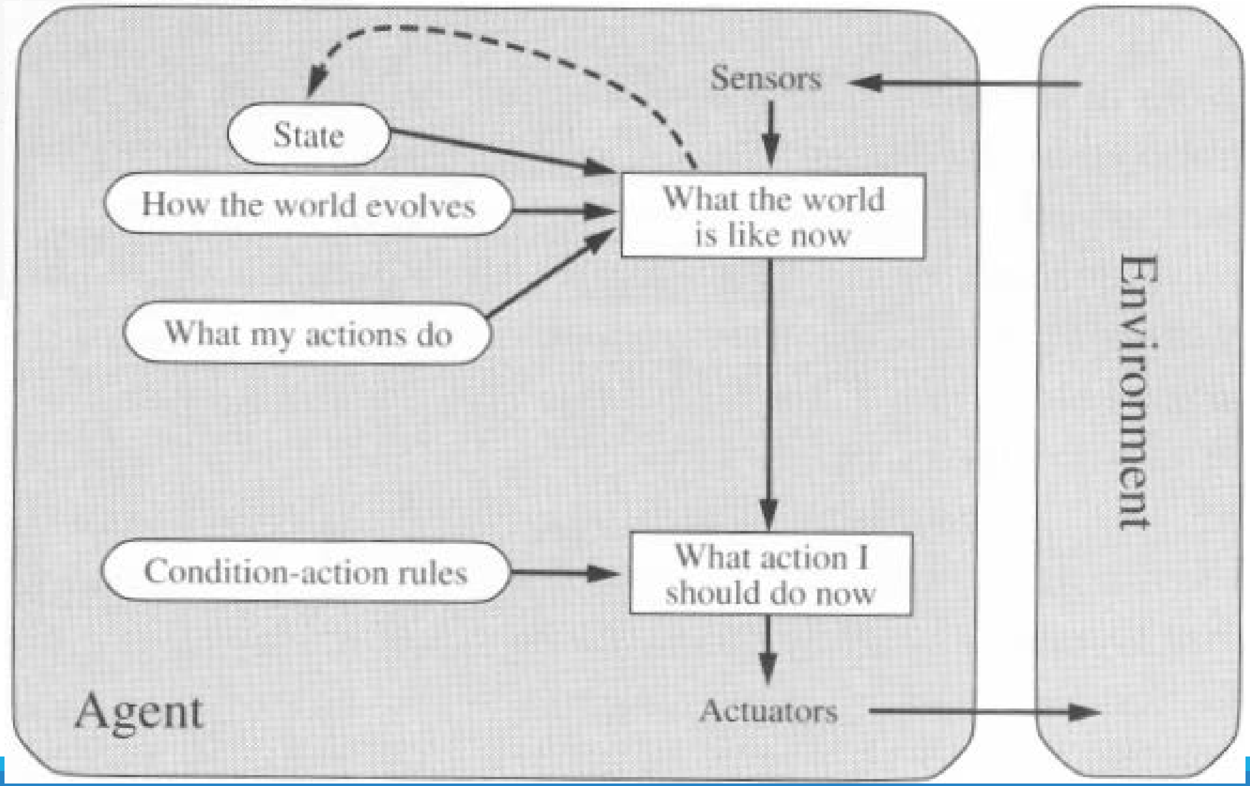
- A state combining past percepts (state ← percept history)
- Keep track of the percept history
- Reflect some of the unobserved aspects
  - for the current percept
- Two types of knowledge
  - How the world evolves independent of the agent
  - How the agent’s actions affect the world

Model-based Reflex Agents

```
function REFLEX-AGENT-WITH-STATE(percept) returns action
   static: state, a description of the current world state
           rules, a set of condition-action rules

   state ← UPDATE-STATE(state, percept)
   rule ← RULE-MATCH(state, rules)
   action ← RULE-ACTION[rule]
   state ← UPDATE-STATE(state, action)
   return action
```

Agent is with knowledge / memory (the internal state)



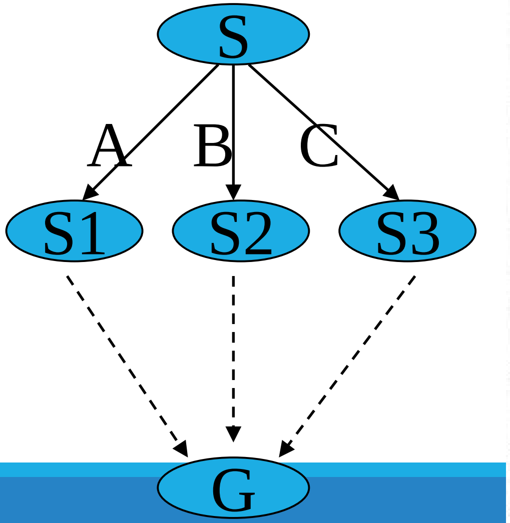
Goal-based Agents

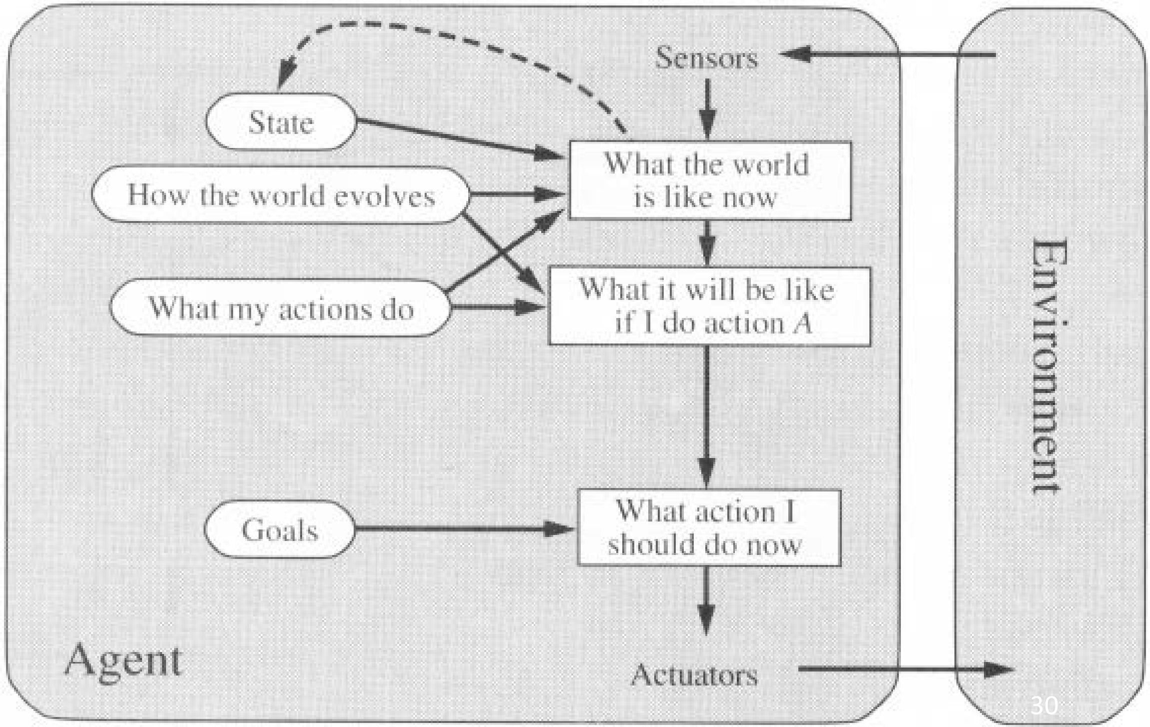
A goal/objective to achieve

- Judgment of rationality

Actions chosen are towards the goal

- Internal state
- Current percept

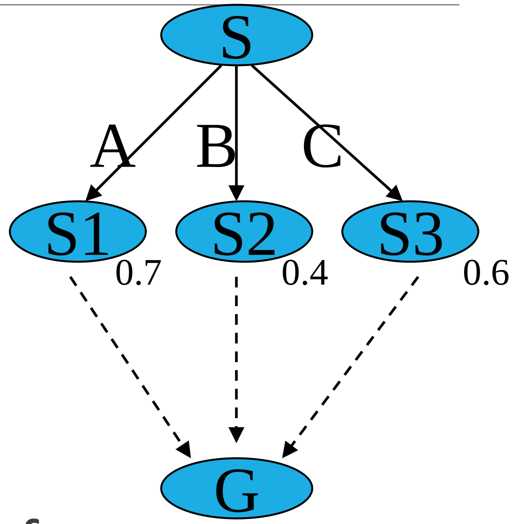



Sensors
State
What the world is like now
How the world evolves
What my actions do
What it will be like if I do action A
Goals
What action I should do now
Agent
Actuators
Environment

Goal-based agents

- Less efficient
  - Many checks on the states
- More flexible
  - Agent ← different goals ← different tasks
- Search and planning
  - Find out action sequences to achieve its goal

Utility-based Agents



High quality actions

- Goals alone are not enough
- E.g. Goal: Fill in our stomachs
  - Meals in Canteen, good or not ?

Many action sequences → Goals

- Some are better and some worse

Goal means success, utility means degree of success

- How successful it is

Utility

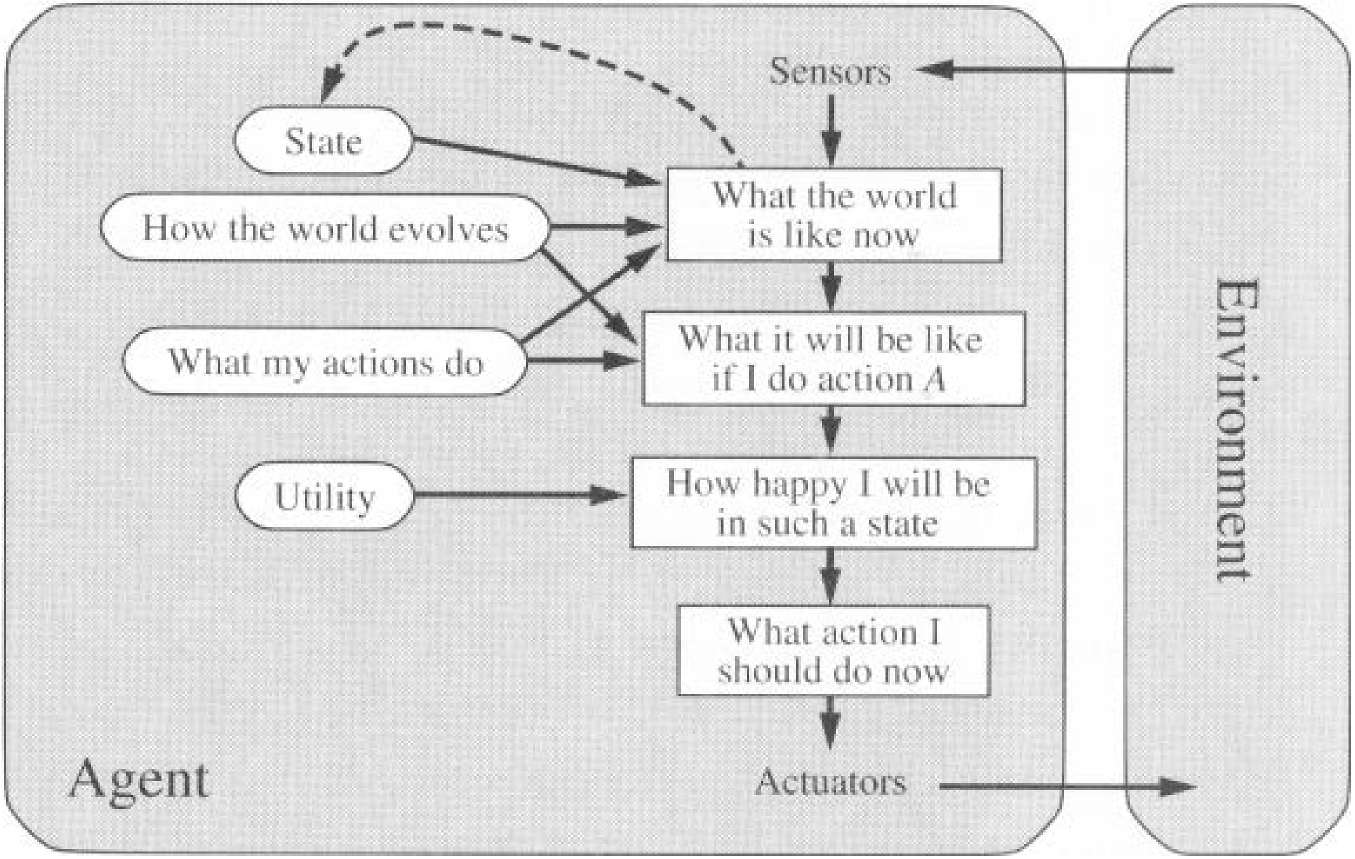
- Degree of success
- A function that maps a state to a real number

Utility-based Agents

Several advantages

- Conflicting goals
  - Delicious and cheap meal
    - meal in Canteen, cheap but yucky
    - meal in Hotel, expensive but delicious
  - Only some of the goals can be achieved
  - Utility describes the trade-off
- Several goals
  - None of them are achieved certainly
  - Utility provides a way for decision-making

Utility-based Agents


Sensors
State
What the world is like now
How the world evolves
What it will be like if I do action *A*
What my actions do
Utility
How happy I will be in such a state
What action I should do now
Agent
Actuators
Environment

Learning Agent

A programmed agent

- May work perfectly immediately
- May make wrong decision / action

Learning agent

- Learn and adjust its decision / action
  - From feedback of the environment
- Better in solving the same problem in future
- More difficult
  - Programmed to have this ability
  - Machine learning techniques

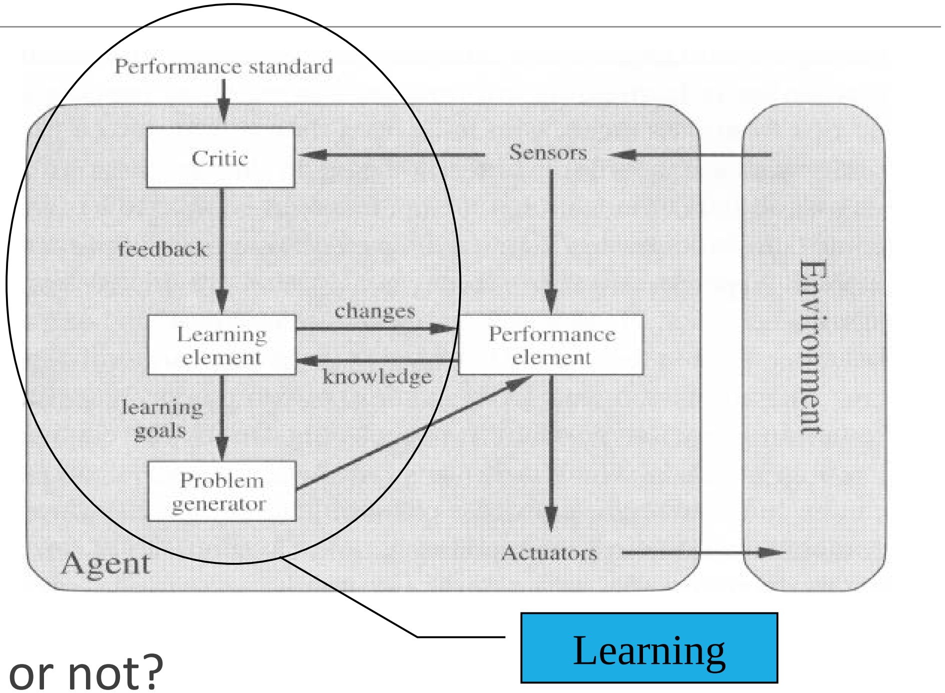
Learning Agents

Performance element

- Selecting external actions

Learning element

- Making improvement



Critic

- Feedback from user, good or not?

Problem generator

- New derived situation, try new solution