# CSCI 3230

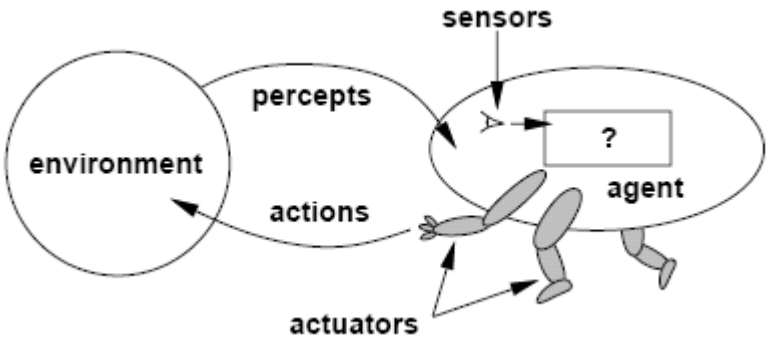
## Fundamentals of Artificial Intelligence

## Chapter 2 INTELLIGENT AGENTS

# Outline

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

# Agents and environments




- Agents include humans, robots, softbots, thermostats, etc.
- The agent function maps from percept histories (sequence) to actions:

$$f : P^* \rightarrow A$$

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

# Four Main Things

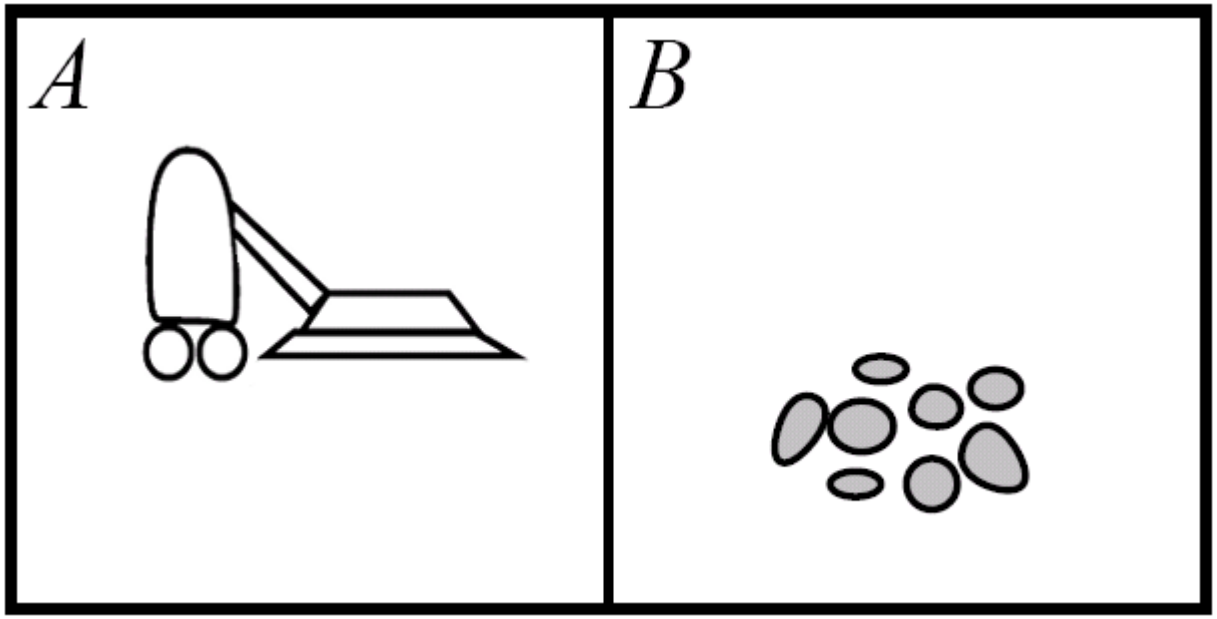
- *Percepts*: (sensor reading) (S)
  - Percept sequence is everything perceived so far
  - Sounds, images, text, facts, rules, knowledge
- *Actions*: (Actuators) (A)
  - Manipulation of effectors (actuators) that affect the environment. Output in general (e.g. text)
- *Goals*: performance measure (P)
  - More effective, more efficient (to achieve the goal)

and/or, usually trade-off

- *Environment*: (E)
  - To understand and react to the external world (environment) as appropriate

?student agents?

# Vacuum-cleaner world




- Percepts: location and contents, e.g. [A, Dirty]
- Actions: Right, Left, Suck, NoOp
- Goal?

# A vacuum-cleaner agent

| Percept sequence | Action |
|---|---|
| [A, Clean] | |
| [A, Dirty] | |
| [B, Clean] | |
| [B, Dirty] | |
| [A, Clean], [B, Clean] | |
| [A, Clean], [B, Dirty] | |

What is the right function?

Can it be implemented in a small agent program?

```
Function Reflex-Vacuum-Agent([location, agent])
return action
        if status = Dirty then return suck
        else if location = A then return Right
        else if location = B then return Left
```

Stopping criteria?

# Rationality

Rational Agent:

one that does the right things (right actions).

Performance measure:

- How successful:
  - Need an objective performance measure imposed by some authority.
  - E.g. performance measure for a floor cleaning agent:
    - Amount of dirt cleaned up within a time period?
    - Amount of electricity consumed?
    - How much free time the agent has to perform other work?

# Rationality 1

- When to evaluate the agent performance:
  - E.g. measure the amount of dirt in the 1st hour = evaluating the initial performance and not how clean overall.
  - Need to measure the performance in long run. Sampling pts & rate
  - Rationality ⇨ expected success given what has been achieved.
    - E.g. street crossing is rational because most of the time the crossing would be successfully.

- What is rational at any given time depends on four things:
  1. The performance measure that defines degree of success. Objective
  2. Everything that the agent has perceived so far. Facts
  3. What the agent knows about the environment. Knowledge
  4. The actions that the agent can perform. Capabilities e.g. surgeon??

# Rationality 2

- Definition of a rational agent: For each possible percept sequence, a rational agent should choose an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and whatever built-in knowledge (capabilities) it has.

Rational ≠ omniscient (know everything)

Rational ≠ clairvoyant (see future)

Rational ≠ successful

Rational ⇒ exploration, learning, autonomy(?)自主

# Rationality 3

- Autonomy
  - An agent's behavior is based on its experience and the built-in knowledge for the particular environment in which it operates.
  - A system is autonomous to the extent that its behavior is determined by its own experience.
  - E.g. evolution provides animals with enough built-in reflexes so that can survive long enough to learn for themselves.
  - It would be reasonable to provide an AI agent with some initial knowledge as well as an ability to LERAN (Meta-)

# Intelligent Agent

AIDA – Affective Intelligent Driving Agent [Youtube]

# PEAS

- Performance Environment Actuators Sensors

To design a rational agent, we must specify

the task environment – PEAS (problem definition)

Consider, e.g., the task of designing an automated taxi:

Performance measure??

Environment??

Actuators??

Sensors??

# PEAS 2

Answer:

To design a rational agent, we must specify the task environment.

Consider, e.g. the task of designing an automated taxi:

Performance measure??

Safety, destination, profits, legality, comfort, …

Environment??

Streets/freeways, traffic, pedestrians, weather, …

Actuators??

Steering, Accelerator, brake, horn, speaker/display,…

Sensors??

Video, accelerometers, gauges, engine sensors, keyboard (or voice recognition), GPS, …

# Internet shopping agent

Performance measure?? Different for buyers and sellers; ?

Environment??

Actuators??

Sensors??

# Environment types

## Properties of Environments

- Accessible vs. Inaccessible (fully vs. partially observable)
  - Sensors detect all aspects of environment relative to choosing an action
  - Sensors can access complete state = fully observable
- Deterministic vs. Nondeterministic
  - Is the next state entirely determined by the previous state and action?
- Episodic vs. Non-episodic (sequential)
  - Episode = (percept, action) pair
  - Episodic = episodes do not depend on actions in previous episodes, and need not think ahead
- Static vs. dynamic
- Discrete vs. continuous
- Single agent vs. multi-agent (competitive or cooperative)

Can be a mix

# Environment types 2

| | Crossword | Chess | Backgammon | Internet shop | Taxi |
|---|---|---|---|---|---|
| Observable?? | | | | | |
| Deterministic?? | | | | | |
| Episodic?? | | | | | |
| Static?? | | | | | |
| Discrete?? | | | | | |
| Single-agent?? | | | | | |

*Strat : Strategic- deterministic except for the actions of other agents

The environment type largely determines the agent design.

The real world is (of course) partially observable, stochastic, sequential, dynamic, continuous, multi-agent.

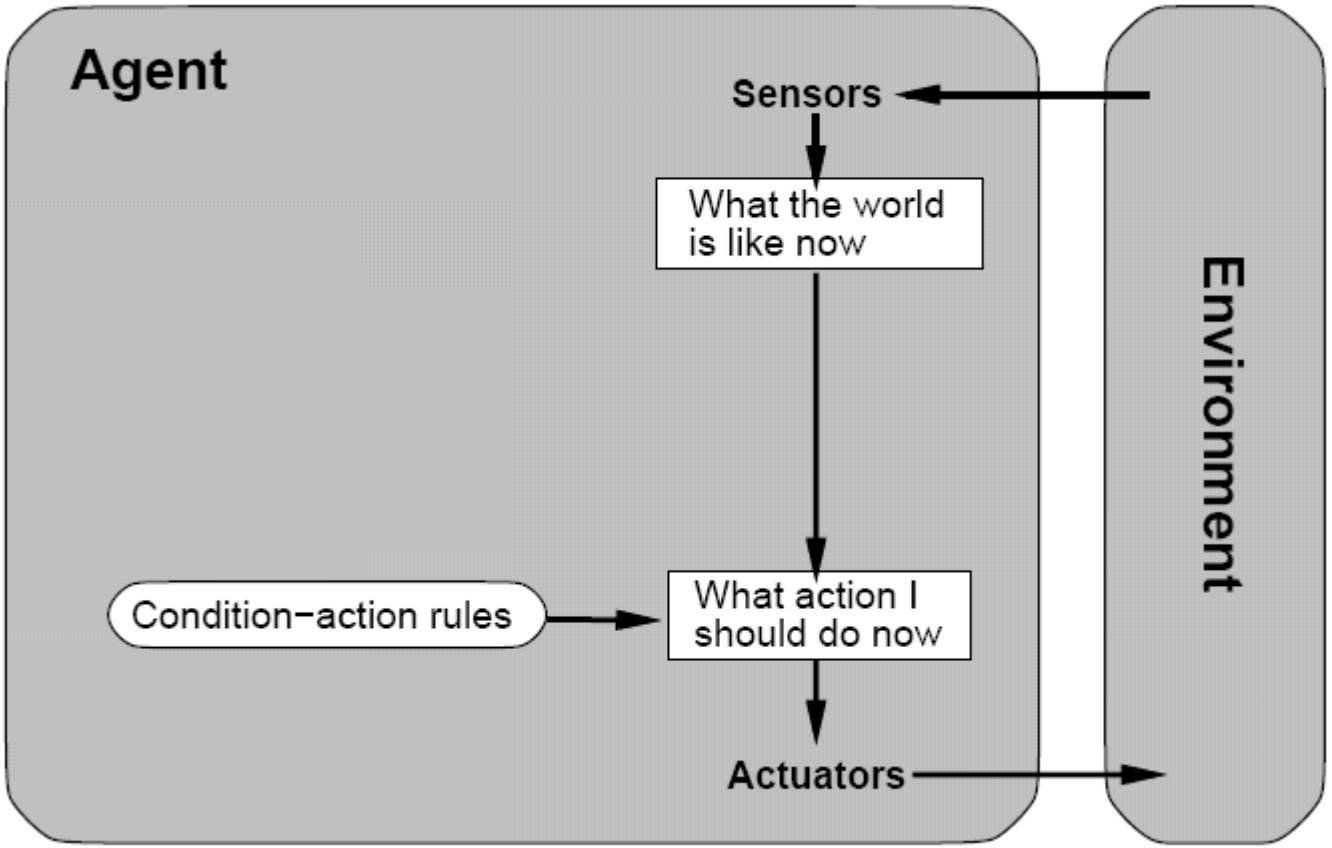
# Five main types of agents

Four basic types in order of increasing generality:

- Simple reflex agents
- Reflex agents with state (model-based)
- Goal-based agents  (model-based)
- Utility-based agents  (model-based)

All these can be turned into learning agents

# 1. Simple reflex agents



# (Five main types of agents)

## 1. Simple reflex Agents

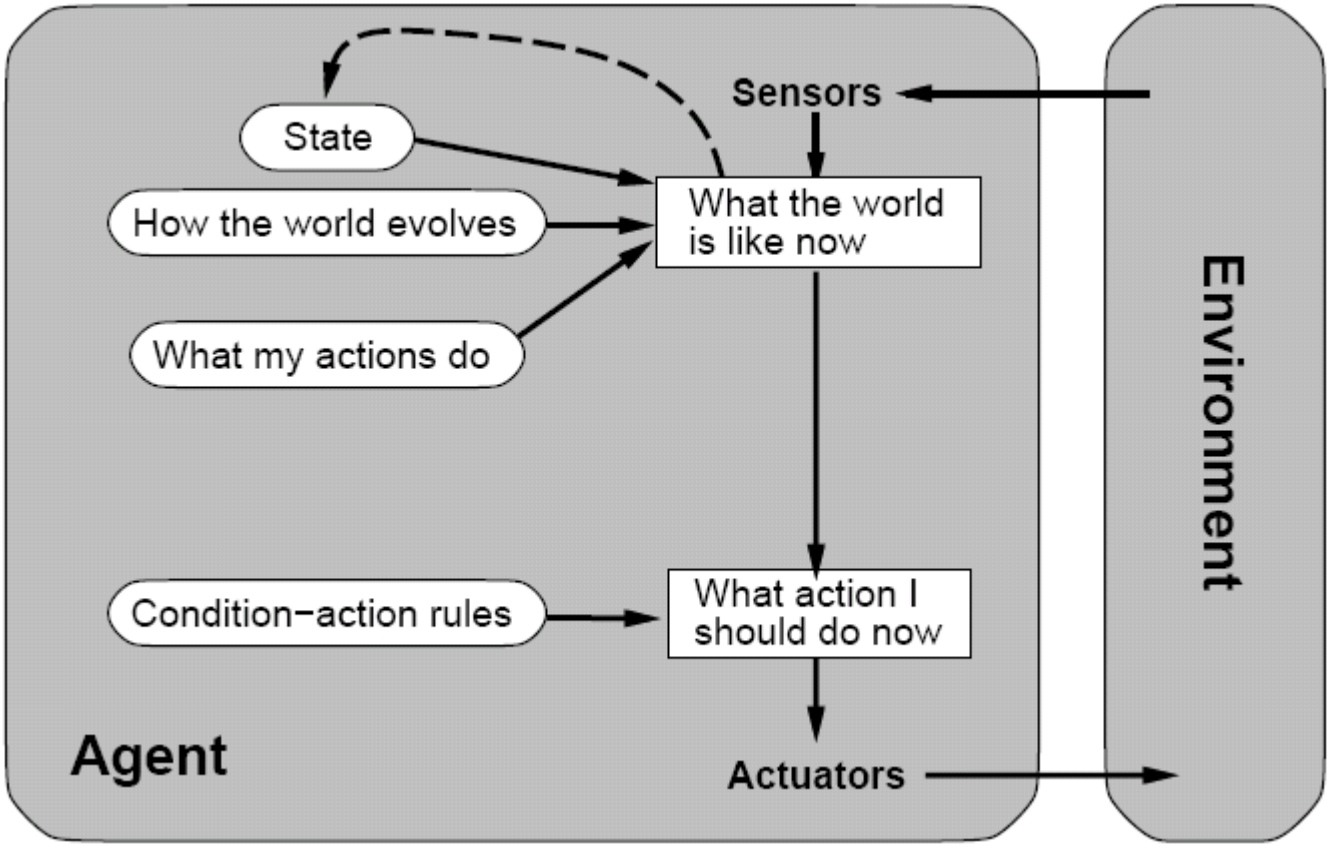
Agents with condition-Action Rules.

E.g. If car in front is breaking then initiate-braking.

**Function** Simple-reflex-agent (*percept*) return *action*
**static**: *rules*, a set of condition-action rules

*state* ← Interrupt-Input (*percept*)
*rule* ← Rule-match (*state*, *rules*) // identify matched *rule* id.
*action* ← Rule-action [*rule*]
**return** *action*

# 2. Model-based Reflex agents with state



# 2. Model-based Reflex Agents that keep track of the world with Internal State

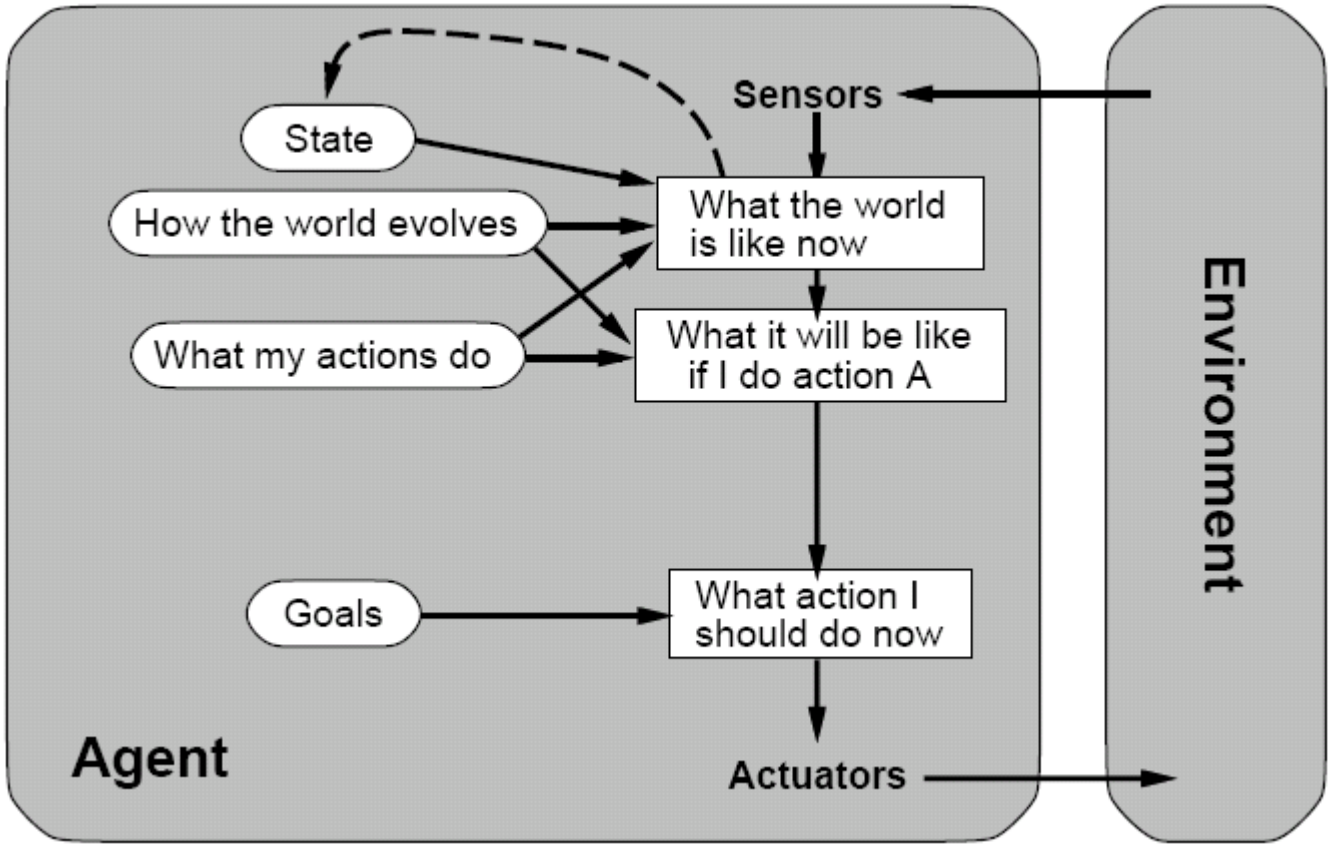
Simple agents that maintain some sort of internal state of the world (and a model) in order to choose an action.

E.g. a driving agent need internal states to decide whether it is possible to change lane. Turn left at junction.

**Function** Model-Based-Reflex-Agent (*percept*) **returns** action
  **static**: *state*, a description of the current world state
    *model*, a description of how the next state depends on current state and action
    *rules*, a set of condition-action rules
    *action*, the most recent action, initially none

  *state* ← Update-state (*state*, *action*, *percept*, *model*)
  *rule* ← Rule-match (*state*, *rules*)
  *action* ← Rule-action [*rule*]
  **return** *action*

# 3. Model-based, Goal-based agents



# (Five main types of agents)

## 3. Goal-based Agents

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

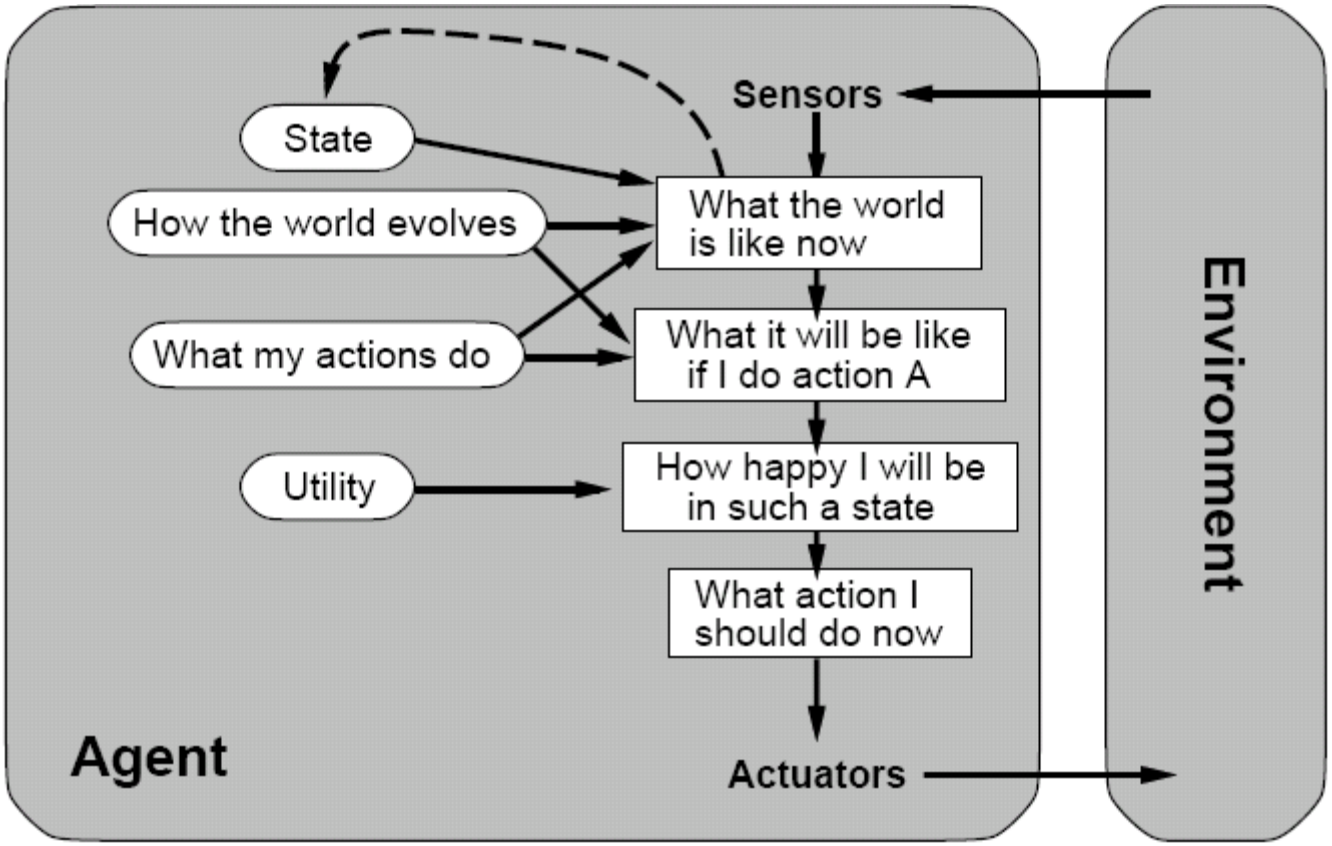
  e.g. At a road junction, the agent need to know whether to turn (left/right) or go straight.

- The agent need a GOAL describing the desirable situations.

  e.g. Passenger's destination

- Search (ch3-6) and Planning (ch10 – 11) are subfields of AI for finding action sequences that do achieve the agent's goals

# 4. Model-based, Utility-based agents



# (Five main types of agents)

## 4. Utility-based Agents

- Goals alone are not really enough to generate high-quality behavior.

  e.g. Many action sequences can get the car to its destination, i.e. achieving the goal.
  But some are quicker, safer, more reliable, cheaper, or more comfortable than others.

- If one world state is preferred to another, then it has higher utility for the agent.
- There may be more than one utility functions for selecting an action, and they may conflict each other.

  e.g. Speed and safety. Multi-objective optimization (utility functions). ?soln (Pareto front)

- We can only achieve some utilities, or have a compromise action.

# 5. Learning Agents

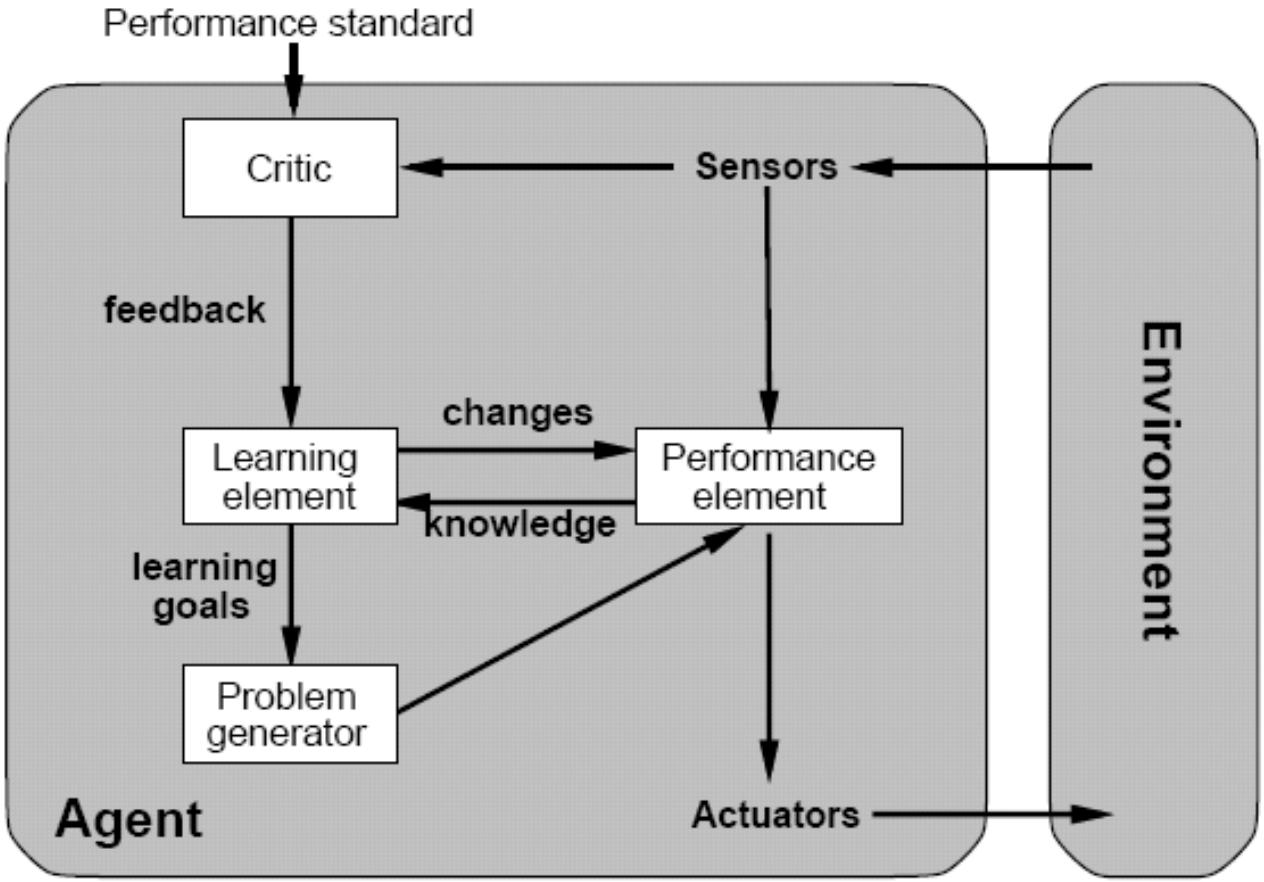



Fig. 2.15

# (Five main types of agents)

## 5. Learning Agents (1)

- LA has 4 conceptual components, Fig 2.15.
- The learning element – responsible for making improvements to the performance element.
- The performance element = previously entire agent: it takes in percepts and decides on actions.
- The learning element takes some knowledge about the performance element and some feedback on how the agent is doing, and determines how the performance element should be modified to (hopefully) do better in the future.

# (Five main types of agents)

## 5. Learning Agents (2)

- The critic tells the learning element how well the agent is doing w.r.t. a *fixed standard* of performance. Necessary because the percepts themselves provide no indication of agent's success.
  E.g., a chess program receiving a percept of checkmating its opponent needs a performance standard to know that is a good thing; the percept itself does not say so.
- The performance standard must be a fixed measure that is conceptually outside the agent; otherwise the agent could adjust its performance standards to meet its behavior. E.g…

# (Five main types of agents)

## 5. Learning Agents (3)

- The problem generator – responsible for suggesting problems & actions for new and informative experiences.
- The performance element tends to repeat doing the actions that are best, given what it knows. Exploitation.
- But if the agent is willing to explore a little, and do some perhaps suboptimal actions in the short run, it might discover much better action for a long run.
  The problem generator's job is to suggest these exploratory problems and actions.
- E.g. identify areas of behavior need improvements and suggest experiments –braking in different road surface conditions. Problem space coverage
- Data augmentation DNN: image random rotation, shifts, shear & flips