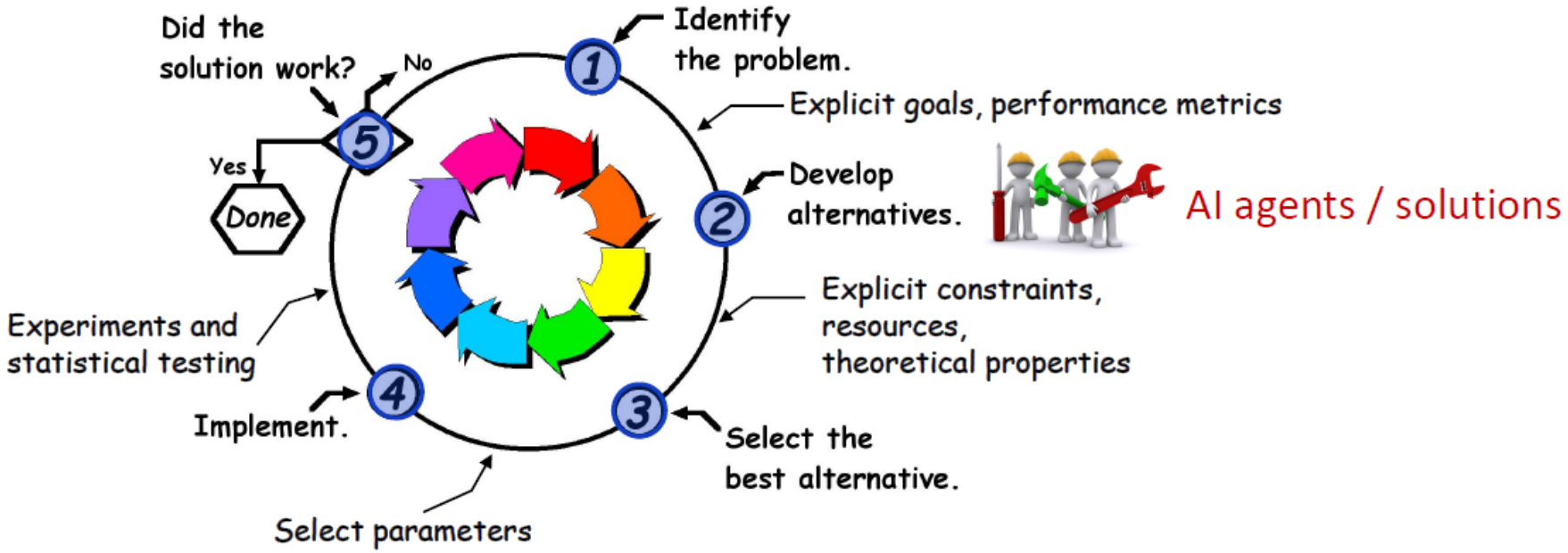
Intelligent/Rational Agents (Tác tử thông minh)

Dr. Nguyễn Văn Vinh

Problem-solving: (Iteratively) Selecting the right agent design

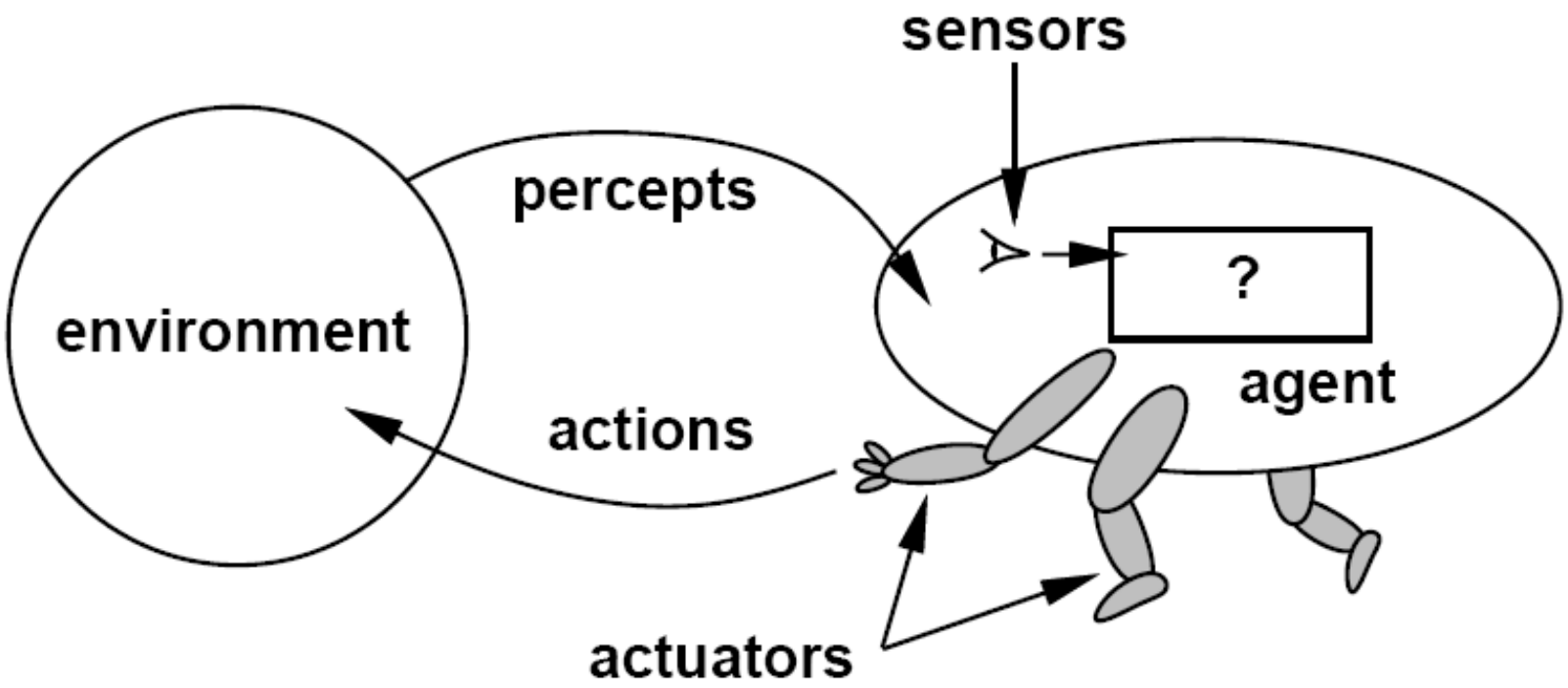
Steps to problem-solving



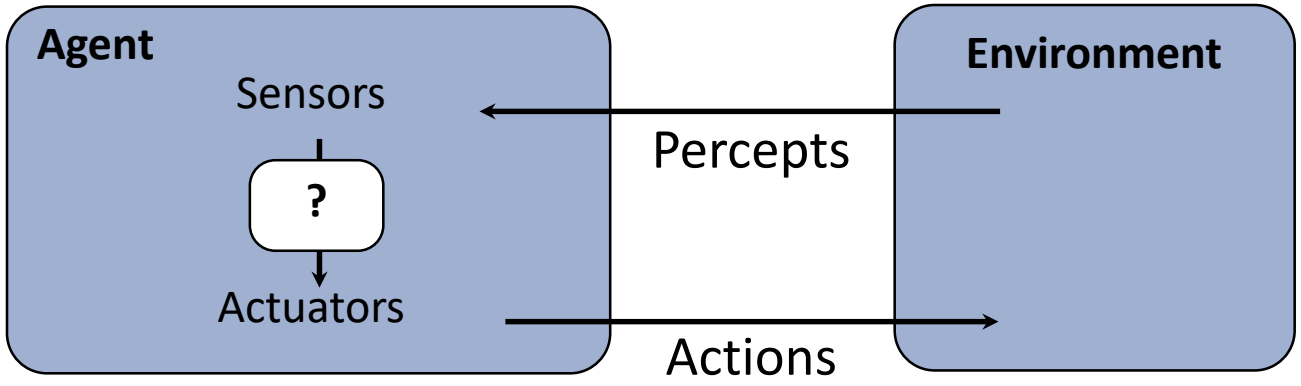
- The design of an **AI agent** for a complex task might require the integration of multiple components and techniques to deal with different aspects of the problem

Agents

- An agent is anything that can be viewed as perceiving (cảm nhận) its environment through sensors (các cảm biến) and acting upon (tác động) that environment through actuators (bộ chuyển động)

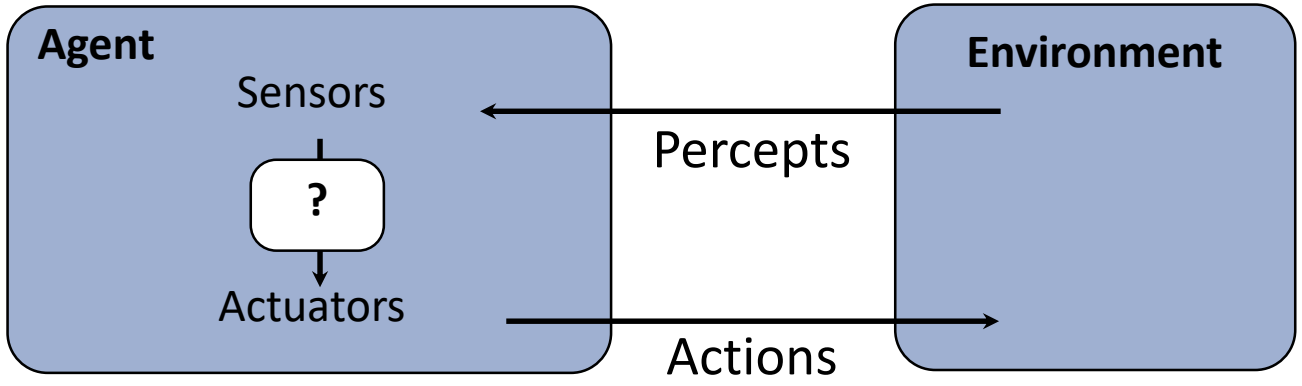


Agents and environments



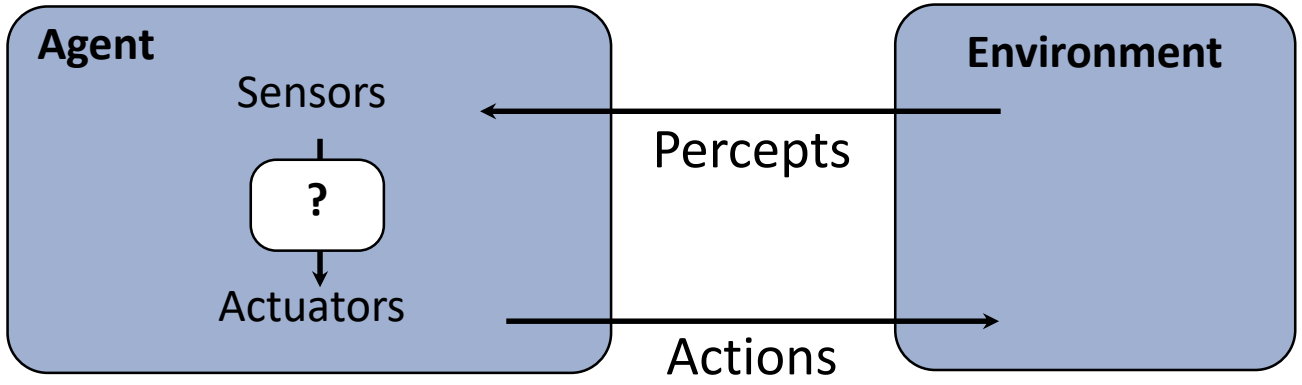
- Are humans agents?
- Yes!
  - Sensors = vision, audio, touch, smell, taste, proprioception
  - Actuators = muscles, secretions, changing brain state

Agents and environments



- Are pocket calculators agents?
- Yes!
  - Sensors = key state sensors
  - Actuators = digit display

Agents and environments



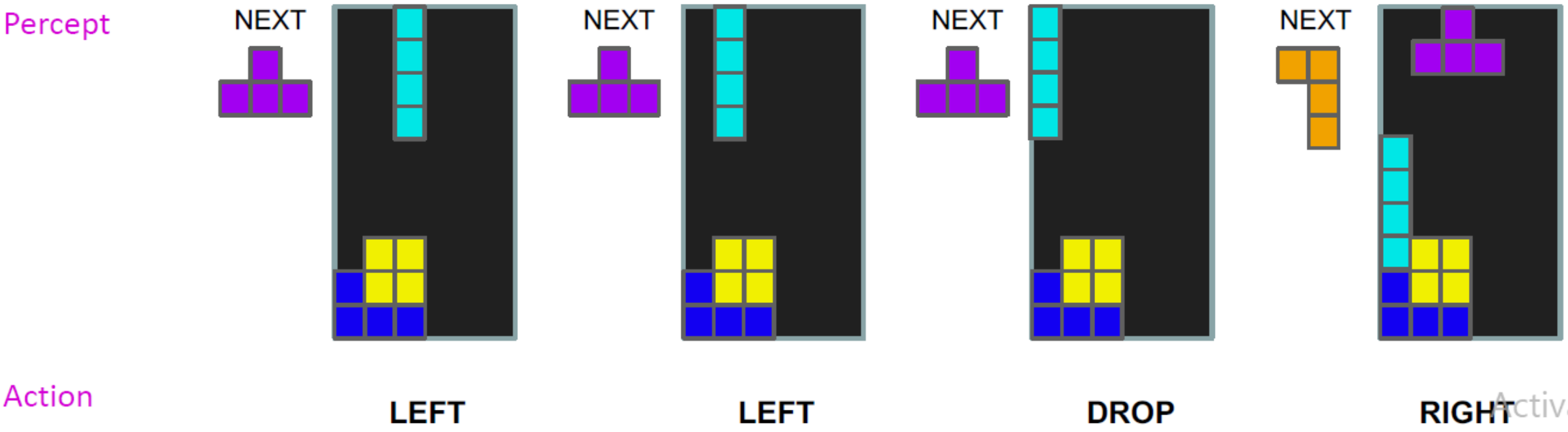
- **AI is more interested** in agents with large computational resources and environments that require nontrivial decision making

Agent functions

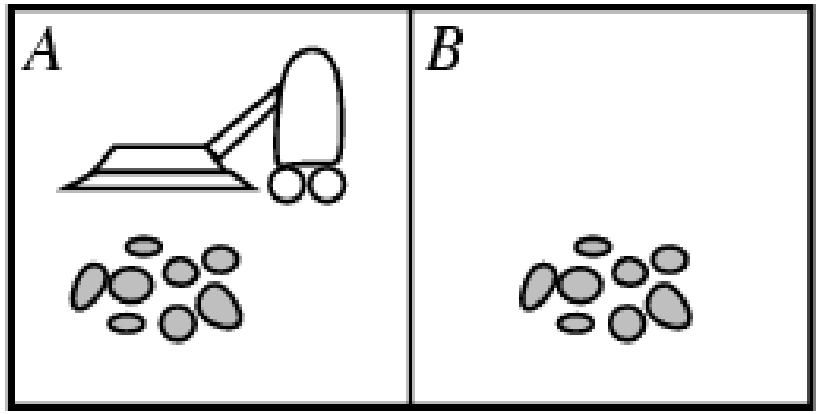
- The ***agent function*** maps from percept histories to actions:

$$f : \mathcal{P}^* \rightarrow \mathcal{A}$$

  - I.e., the agent's actual response to any sequence of percepts



Example: Vacuum-Agent (Agent máy hút bụi)



- **Percepts:**
  Location and status,
  e.g., [A,Dirty]
- **Actions:**
  Left, Right, Suck, NoOp

function Vacuum-Agent([location,status]) returns an action

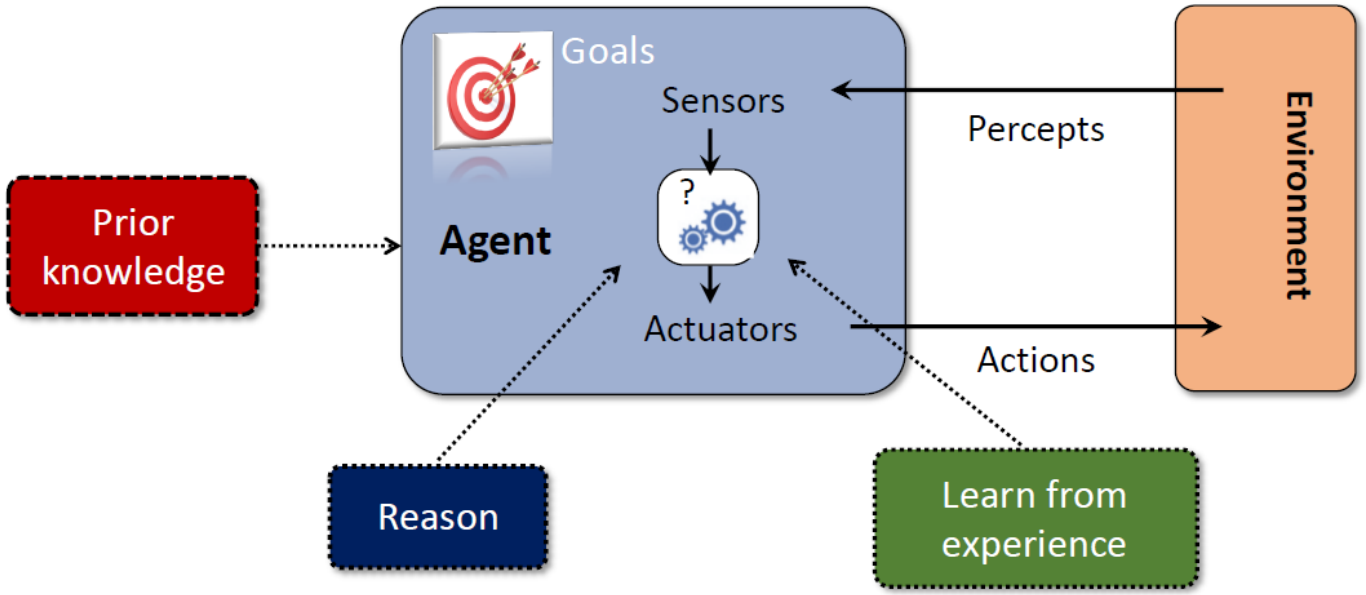
- *if* status = Dirty *then* return Suck
- *else if* location = A *then* return Right
- *else if* location = B *then* return Left

Agent function

Percept sequence	Action
[A,Clean]	Right
[A,Dirty]	Suck
[B,Clean]	Left
[B,Dirty]	Suck
[A,Clean],[B,Clean]	Left
[A,Clean],[B,Dirty]	Suck
etc	etc

Intelligent/Rational Agent

- An **intelligent/rational agent** is an entity *situated* in an environment (physical or virtual) that:
  - **perceives** its environment
  - and, based on the *known circumstances (sequence of percepts), capabilities, and knowledge*
  - takes **actions**
  - that (provably) **maximize** its expected chance of successfully achieving its **goals**



Intelligent/Rational agents

- **Performance measure (utility function) (đánh giá hiệu quả):**
  An *objective* criterion for success of an agent's behavior vs. goals
- **Expected utility:**

$$EU(action) = \sum_{outcomes} P(outcome \mid action) U(outcome)$$

- Can a rational agent make mistakes?

Intelligent/Rational Agent, a few caveats

Rationality only concerns *what decisions* are made, *not* the *thought process* behind them!

Don't care *how*

Rationality ≠ Omniscience

Percepts & knowledge can be and usually are limited

Rationality ≠ Clairvoyance

Environment's dynamics can be unknown

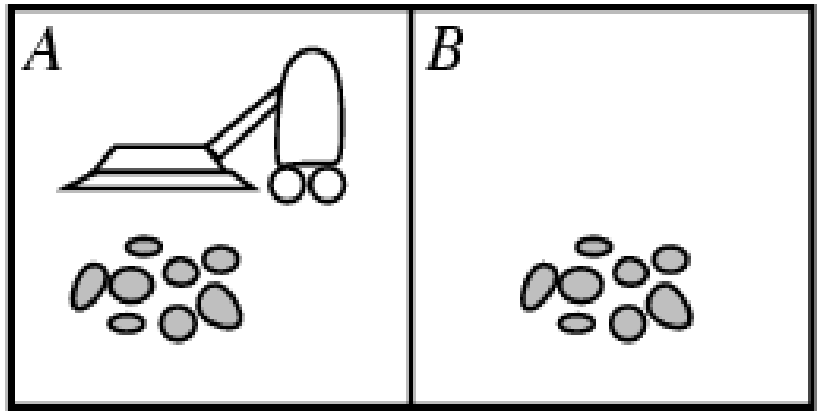
Rationality might require information gathering, exploration, and learning

Information Gathering is to ensure using the most informative percept sequence

Exploration and Learning can be necessary if the environment is (partially) unknown or dynamic

Back to Vacuum-Agent

- **Percepts:**
  Location and status,
  e.g., [A,Dirty]
- **Actions:**
  Left, Right, Suck, NoOp

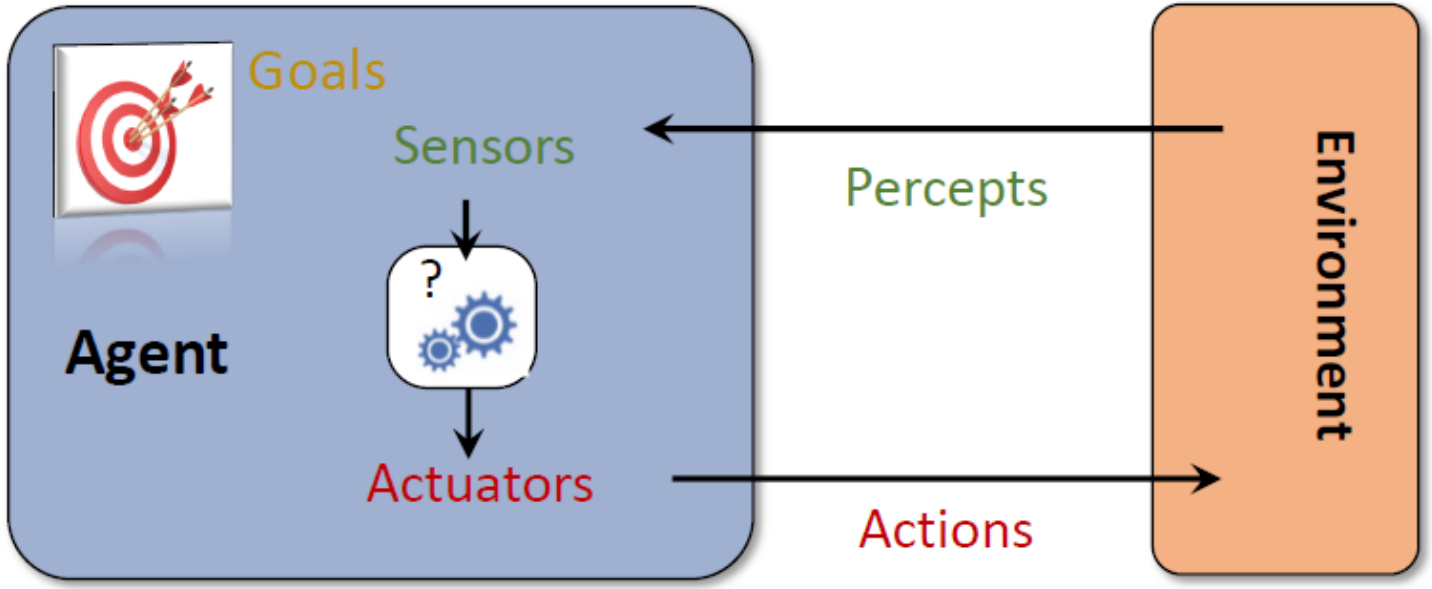


function Vacuum-Agent([location,status]) returns an action

- *if* status = Dirty *then* return Suck
- *else if* location = A *then* return Right
- *else if* location = B *then* return Left

- Is this agent rational?
  - Depends on performance measure, environment properties

Task environment (~Agent problem scenario): **PEAS**



- **Task environment** (~Problem): Performance measure + Environment + Actuators + Sensors (**PEAS**)

PEAS for Packman Game

Performance measure

- -1 per step; +10 food; +500 win; -500 die; +200 hit scared ghost

Environment

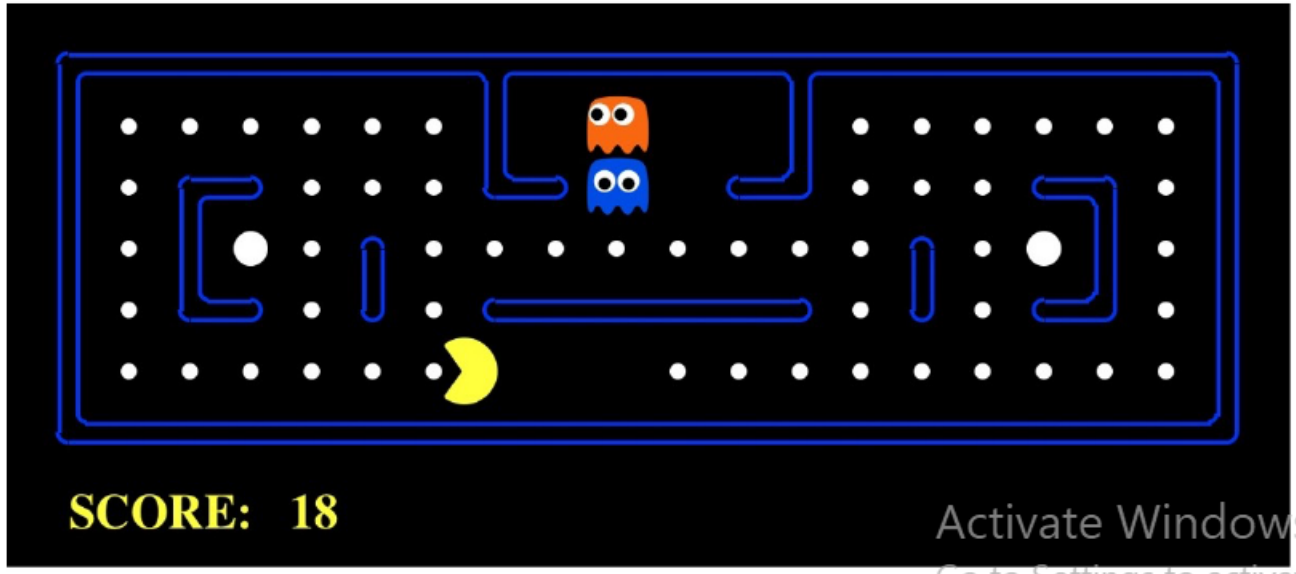
- Pacman dynamics (including ghost behavior), walls

Actuators

- Go: North, South, East, West; Stop

Sensors

- Entire state (game configuration) is visible



PEAS Example: Autonomous taxi

- Performance measure
  - Safe, fast, legal, comfortable trip, maximize profits
- Environment
  - Roads, other traffic, pedestrians, customers
- Actuators
  - Steering wheel, accelerator, brake, signal, horn
- Sensors
  - Cameras, LIDAR, speedometer, GPS, odometer, engine sensors, keyboard

Another PEAS example: Spam filter

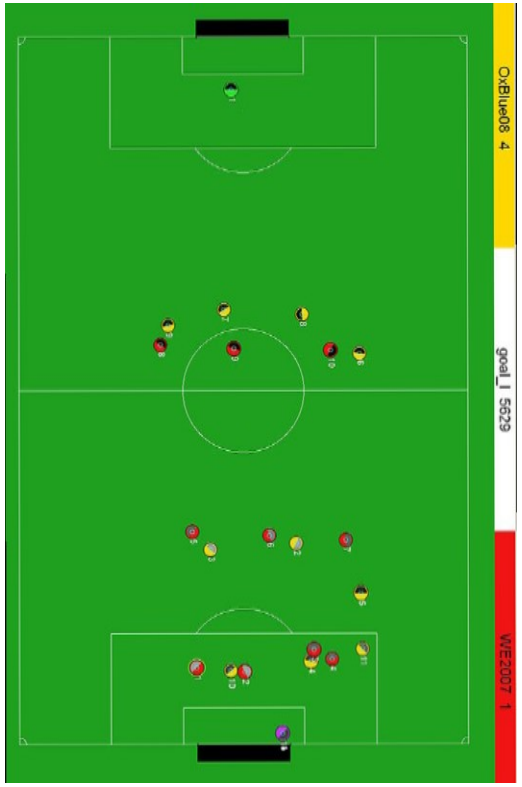
- Performance measure
  - Minimizing false positives, false negatives
- Environment
  - A user's email account, email server
- Actuators
  - Mark as spam, delete, etc.
- Sensors
  - Incoming messages, other information about user's account

A general taxonomy of task environments

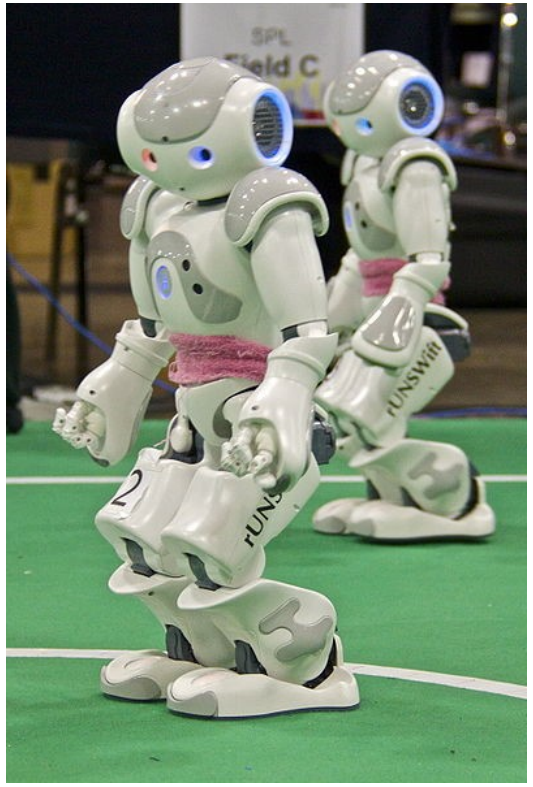
- Fully observable vs. partially observable
- Deterministic vs. stochastic
- Episodic vs. sequential
- Static vs. dynamic
- Discrete vs. continuous
- Single agent vs. multi-agent
- Known vs. unknown

Fully observable vs. partially observable

- Do the agent's sensors give it access to the complete state of the environment?
  - For any given world state, are the values of all the variables known to the agent?



vs.



Source: L. Zettlemoyer

Deterministic vs. stochastic

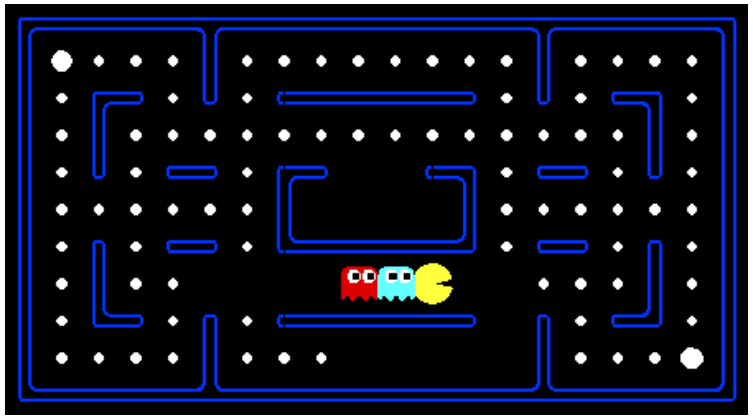
- Is the next state of the environment completely determined by the current state and the agent's action?
  - Is the transition model deterministic (unique successor state given current state and action) or stochastic (distribution over successor states given current state and action)?
  - **Strategic:** the environment is deterministic except for the actions of other agents

vs.

Episodic vs. sequential

- An episodic environment means that subsequent episodes do not depend on what actions occurred in previous episodes.

vs.



Static vs. dynamic

- Is the world changing while the agent is thinking?

vs.

Discrete vs. continuous

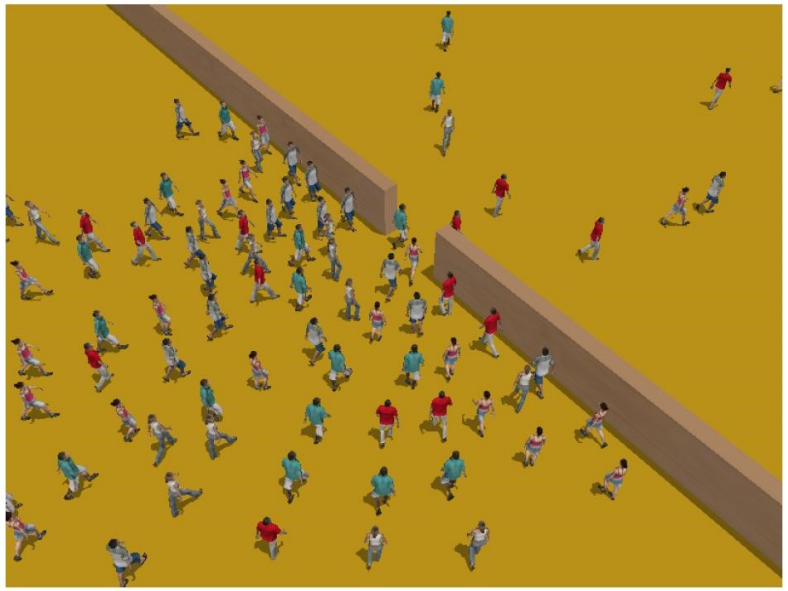
- An environment is said to be discrete if there are a finite number of actions that can be performed within it. Does the environment provide a fixed number of distinct percepts, actions, and environment states?
  - Are the values of the state variables discrete or continuous?

vs.

Single-agent vs. multiagent

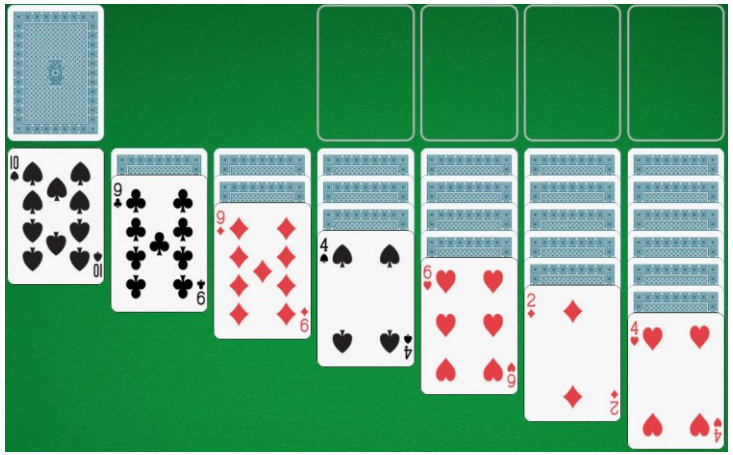
- Is an agent operating by itself in the environment?

vs.



Knowledge (known or unknown)

- In a known environment, all the *outcomes* (or *outcome probabilities*) of an action are given (the agent knows the “rules”). For example, in chess, the agent would know that when a piece is "taken" it is removed from the game?
- In an unknown environment the agent shall *learn* how it works



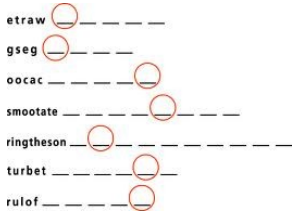
vs.

Quiz

	Fully	Deterministic	Episodic	Static	Discrete
Image Classification					
Solitaire					
Taxi driving					
Internet shopping					
Medical diagnosis					

A lots of real-world domains fall into the hardest case

Examples of different environments



Word jumble solver

Chess with a clock

Scrabble

Autonomous driving

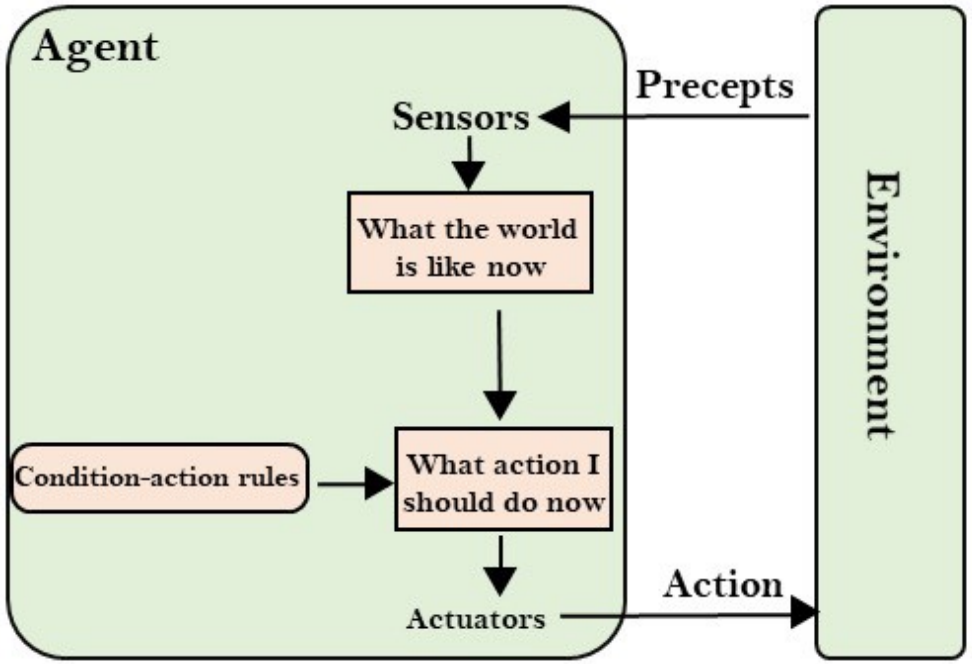
	Word jumble solver	Chess with a clock	Scrabble	Autonomous driving
Observable	Fully	Fully	Partially	Partially
Deterministic	Deterministic	Strategic	Stochastic	Stochastic
Episodic	Episodic	Sequential	Sequential	Sequential
Static	Static	Semidynamic	Static	Dynamic
Discrete	Discrete	Discrete	Discrete	Continuous
Single agent	Single	Multi	Multi	Multi

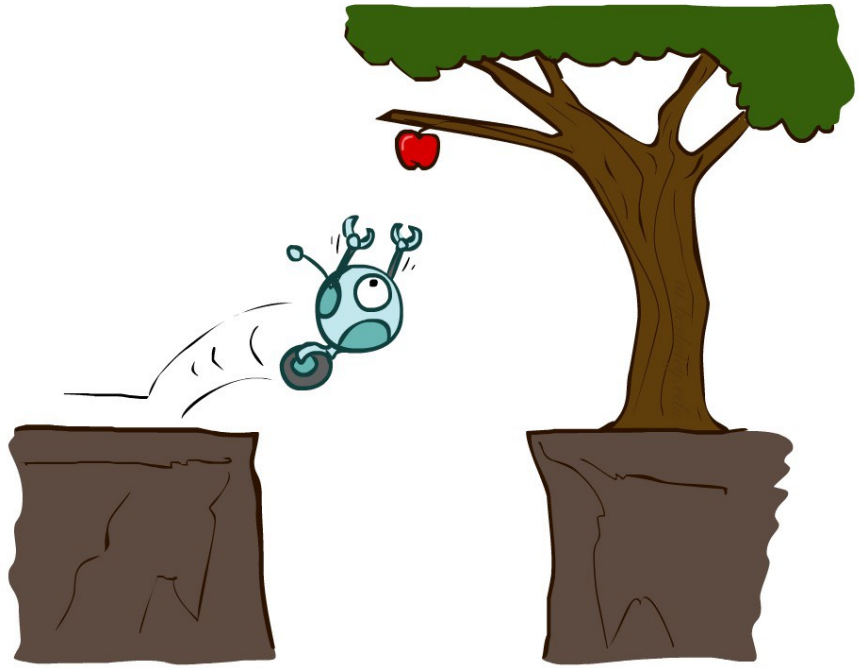
Hierarchy of agent types

- Simple reflex agents
- Model-based reflex agents
- Goal-based agents
- Utility-based agents
- Learning agents

Simple reflex agent

- Select action only based on ***current percept***, ignoring all past percepts
  - **If** x **then** y condition-action rules: It acts according to a rule whose condition matches the current state, as defined by the percept.
- Do not consider future consequences of actions & don't look in the past





Model-based reflex agent

- A model-based reflex agent keeps track of the current state of the world, using an internal model. It then chooses an action in the same way as the reflex agent, by finding the condition that matches the state as defined by percept and internal state (model)
- The model is a ***transition model*** representing how the world evolves in response to actions, which is however only used to build the (augmented) state representation, not to choose among the actions

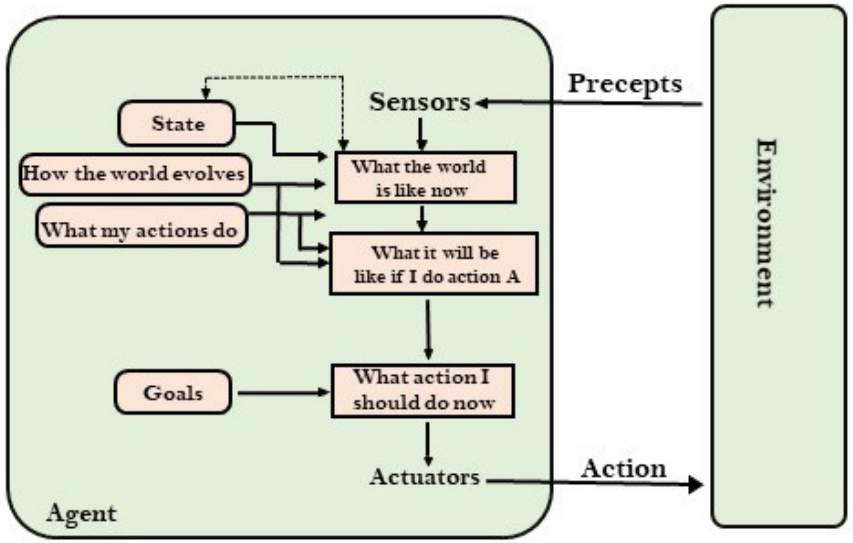
- The agent doesn't **deliberate**, it doesn't lookahead

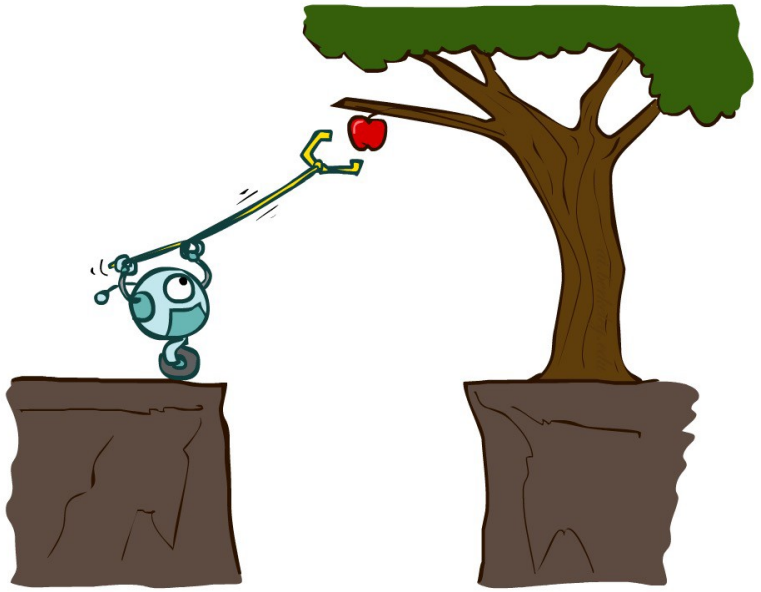
Agents that plan ahead

- **Planning agents (*lookahead*!):**
  - Must formulate **goals**
  - Must have a model of the world
  - Must have a ***transition model***: how the world evolves in response to actions
    - Used to lookahead and predict how the world WOULD BE
- Decisions based on ***predicted consequences*** of actions to achieve the goals in the best way

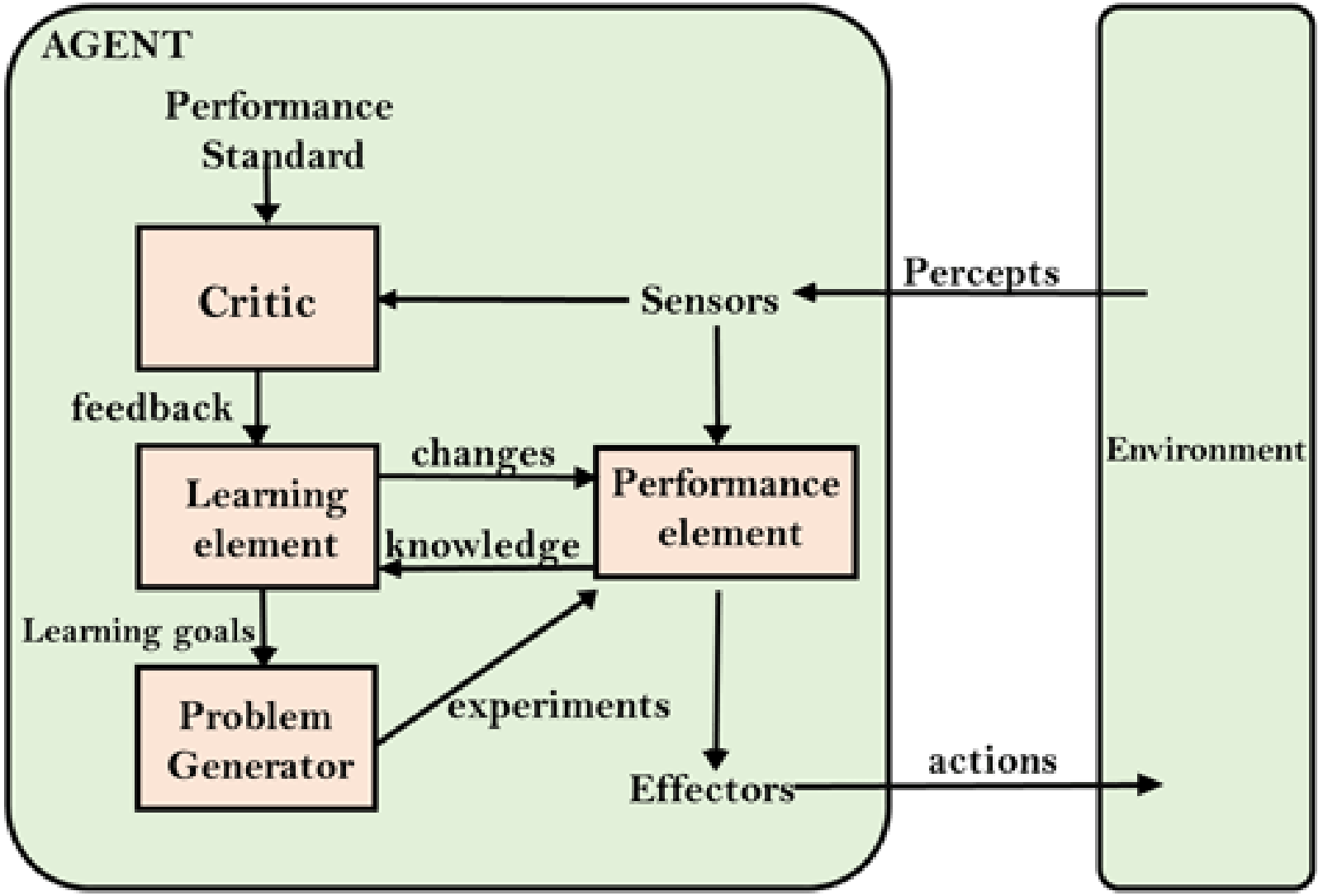
Planning agents: Goal-based agent

- Has explicit **goal(s)**, that describe situations that are desirable
- The agent uses goal information to select between possible actions in the current state by accounting for their effect in the *future* toward the achievement of goals.
- These agents may have to consider a long sequence of possible actions before deciding whether the goal is achieved or not. Such considerations of different scenario are called **searching** and **planning**, which makes an agent proactive.





Learning agent



(Deterministic) *Search* problems

Task environments:

- ✓ Fully observable
- ✓ Known
- ✓ Deterministic
- ✓ Discrete
- ✓ Sequential
- ✓ Single agent

Under these conditions, the solution of "any" problem is a sequence of actions, a ***plan*** (no contingency *policy* is needed)

Search: the process of looking for the best sequence of actions that *reaches the goal*

Agent type:

- ✓ Goal-based
- ➢ Rational

The solution found by a search algorithm is executed *ignoring percepts* (no surprises are expected!):
Open Loop

Topics in AI course

- Deterministic environments: search, constraint satisfaction, classical planning
  - Can be sequential or episodic
- Multi-agent, strategic environments: minimax search, games
  - Can also be stochastic, partially observable
- Stochastic environments
  - Episodic: Bayesian networks, pattern classifiers
  - Sequential, known: Markov decision processes
  - Sequential, unknown: reinforcement learning

Summary

- An **agent** perceives and acts in an environment, has an architecture, and is implemented by an agent program
- An **ideal agent** always chooses the action which maximizes its expected performance, given its percept sequence so far
- **Representing knowledge** is important for successful agent design.
- **Most challenging environments** are inaccessible, nondeterministic, nonepisodic, dynamic, and continuous, and multi-agent