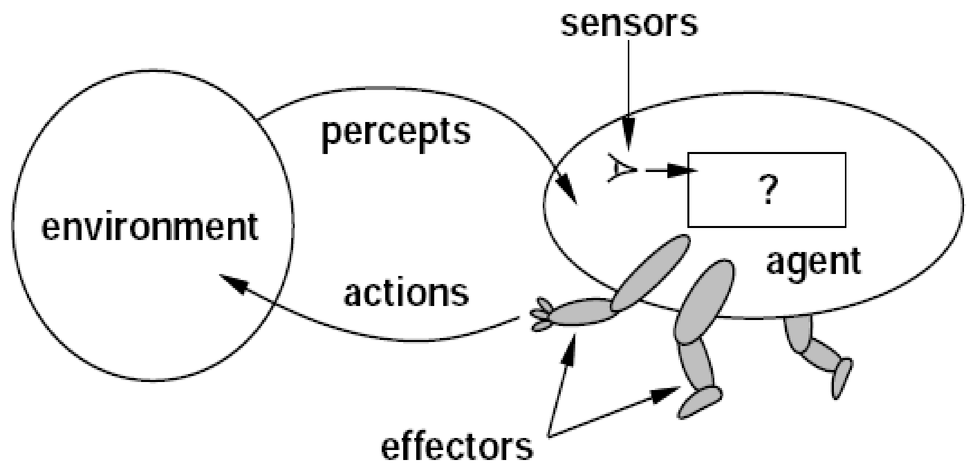
TRƯỜNG ĐẠI HỌC NÔNG LÂM
NLU
1955
THÀNH PHỐ HỒ CHÍ MINH

FACULTY OF INFORMATION TECHNOLOGY

Artificial Intelligence Fundamentals
(NM TTNT)

Semester 1, 2022/2023

Chapter 2. Intelligent Agents



Content

- Agents and Environments
- Rationality
- Properties of Agents
- PEAS
  - Performance measure,
  - Environment,
  - Actuators,
  - Sensors
- Environment Types
- Agent Types

Agents

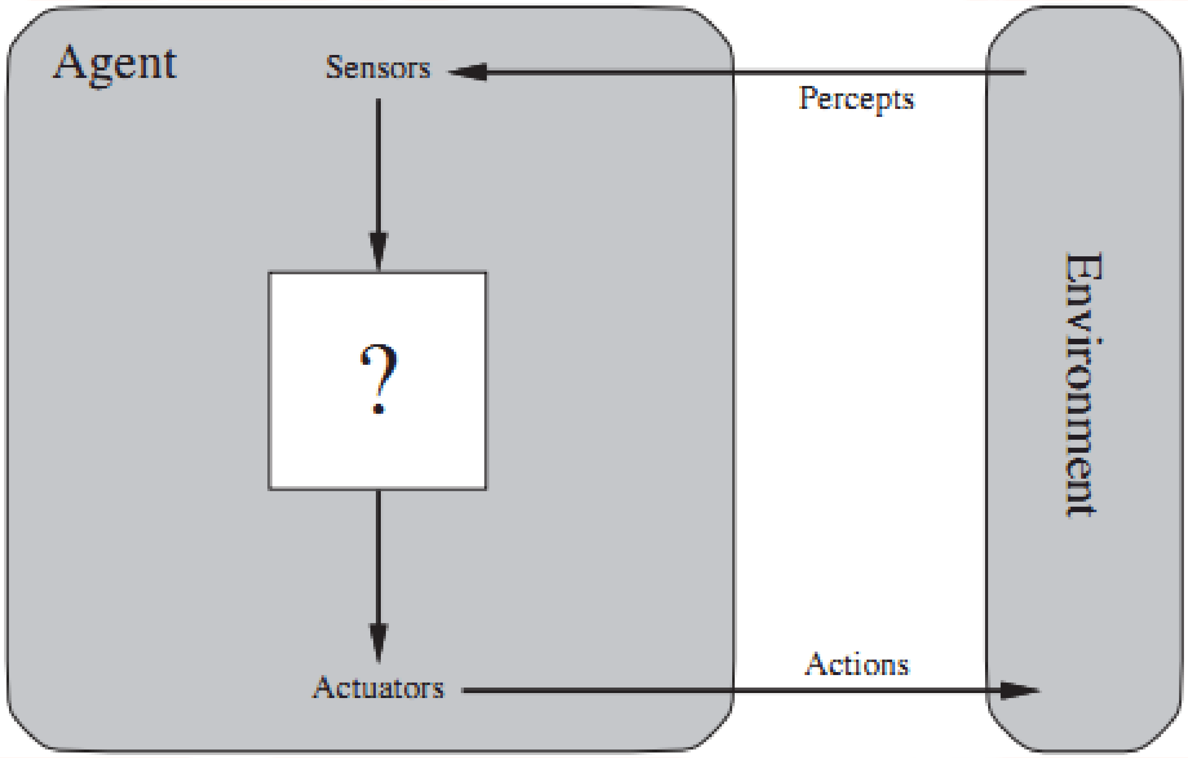
- Agent is anything (human, software, robot, …) that:
  - perceives its environment through sensors
  - acts upon that environment through actuators

Example:

- Human agent:
  - eyes, ears, and other organs for sensors;
  - hands, legs, mouth, and other body parts for actuators
- Robotic agent:
  - cameras and infrared range finders for sensors;
  - various motors for actuators

Agents (cont.)

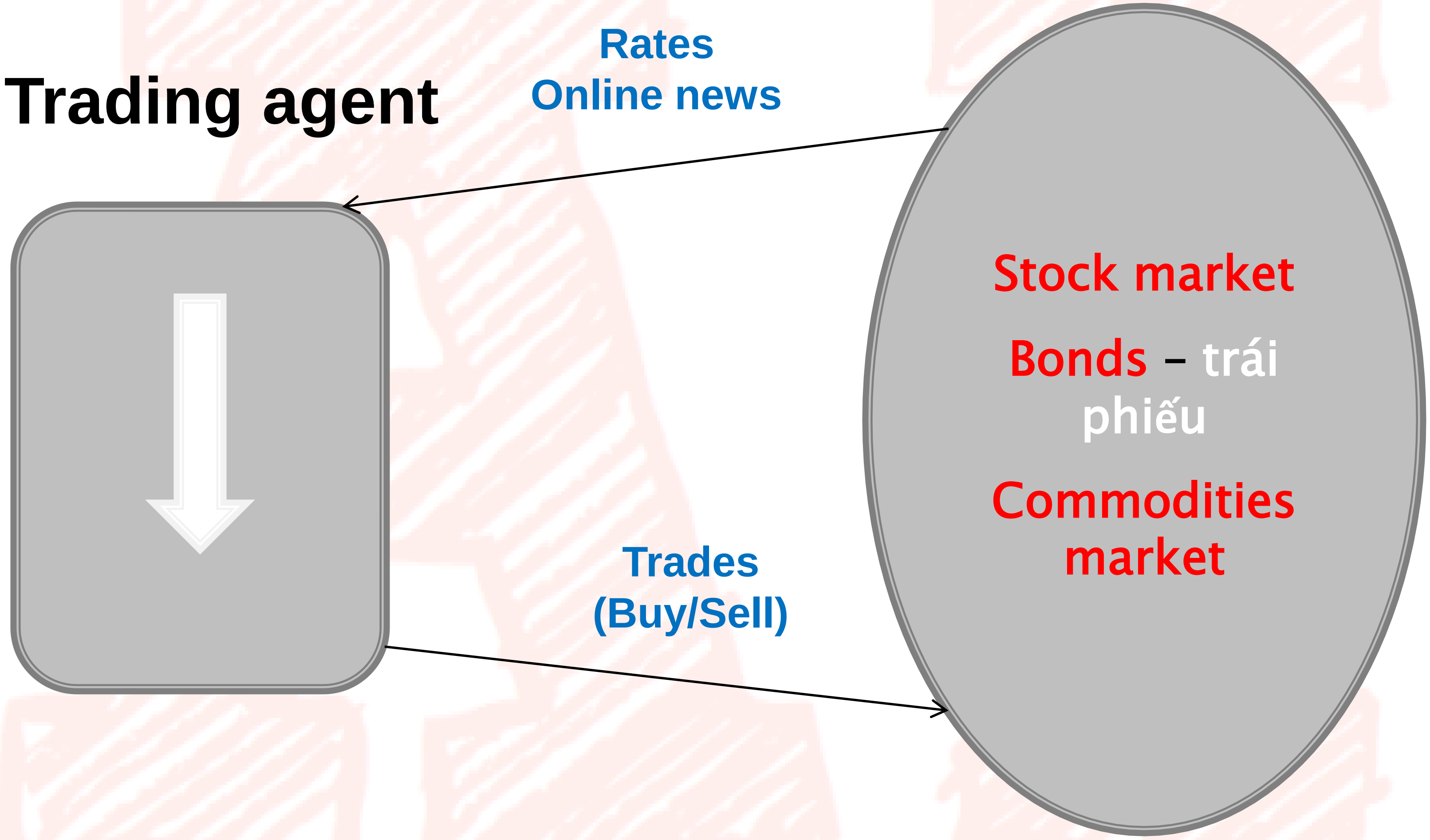
- Agents interact with environments through sensors and actuators.



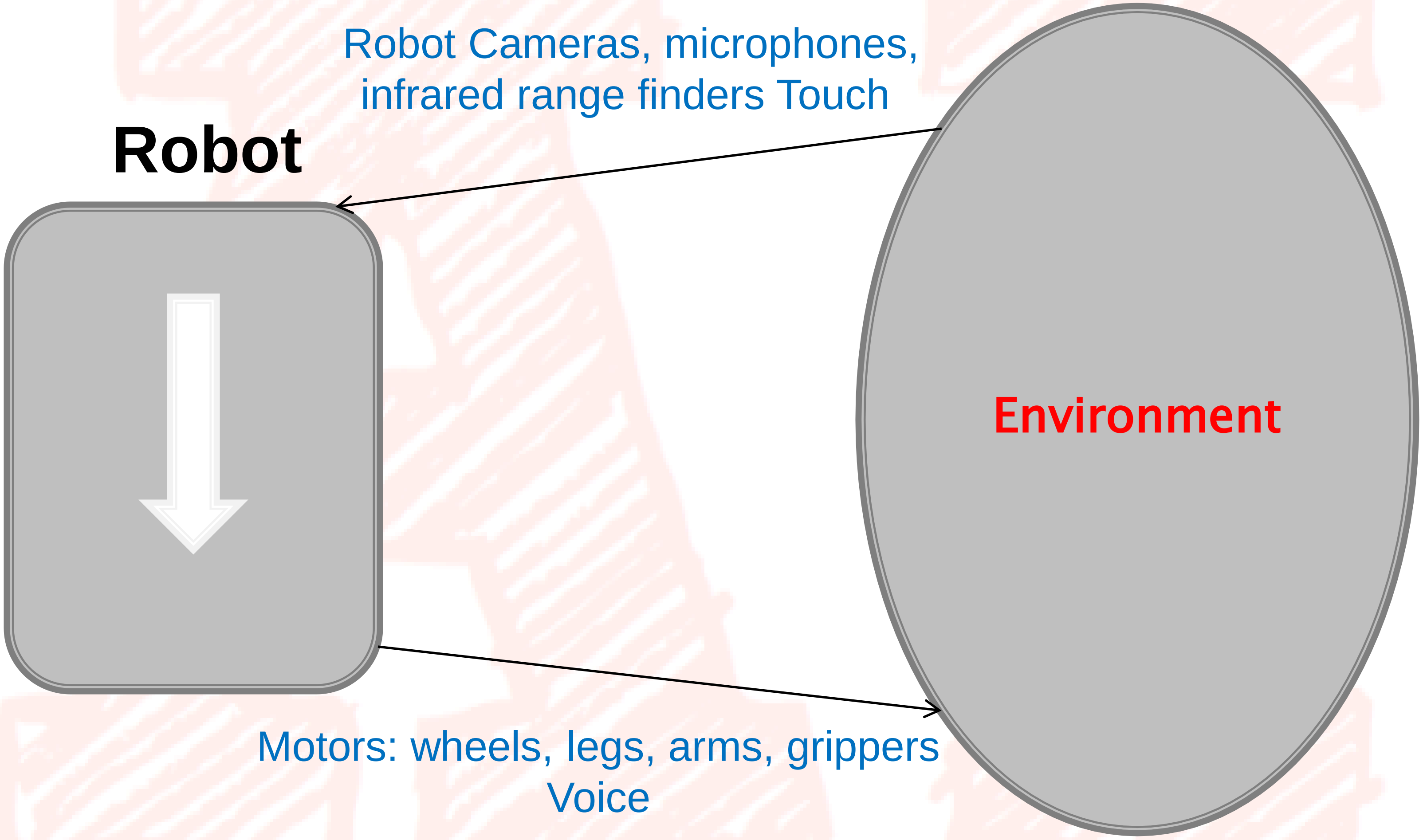
Application of Intelligent Agents

- AI has successful been used in
  - Finance
  - Robotics
  - Games
  - Medicine
  - …

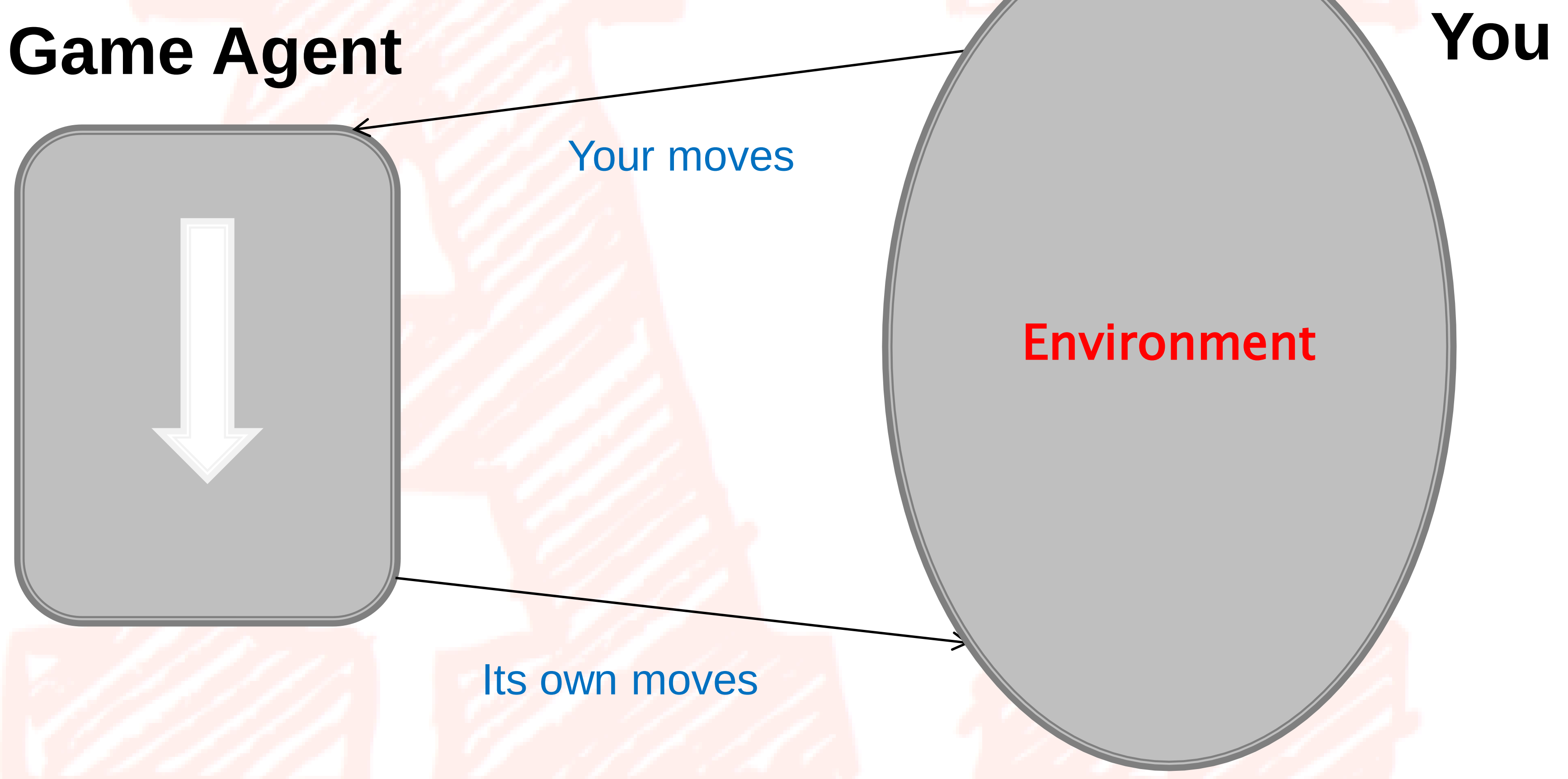
AI in Finance



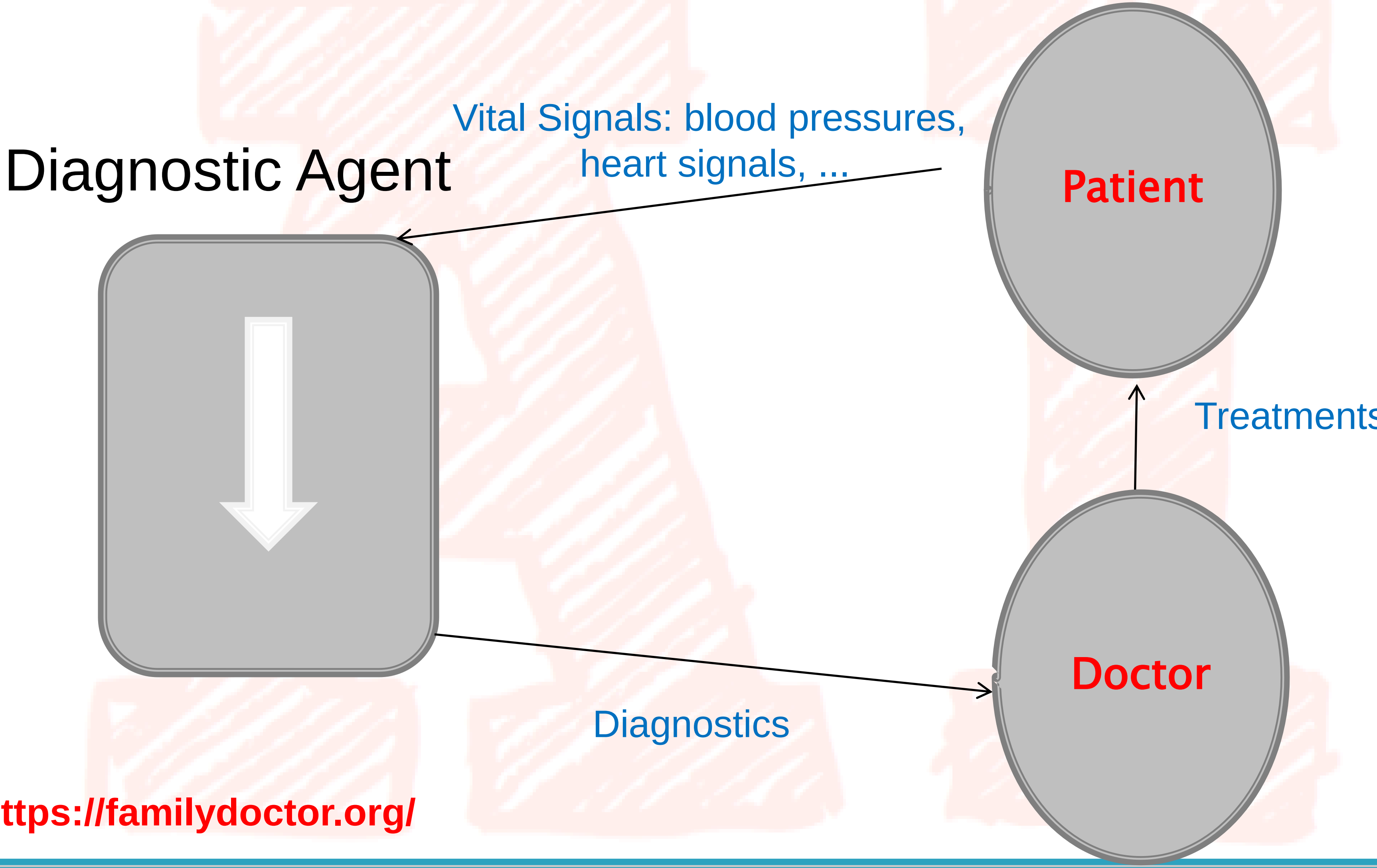
AI in Robotics



AI in Game



AI in Medicine



https://familydoctor.org/

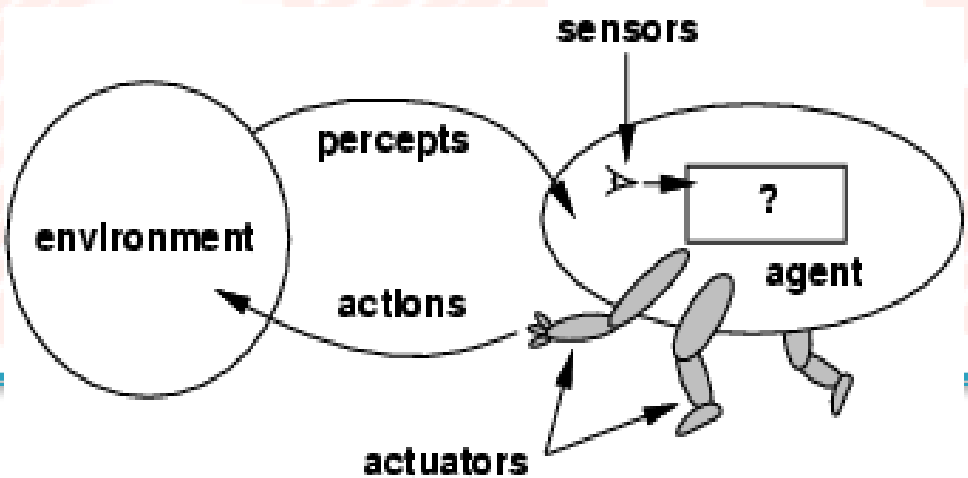
Agents and Environments

- An agent's choice of action can depend on the entire percept sequence observed to date
- An agent's behavior is described by the agent function maps from percept histories to actions:

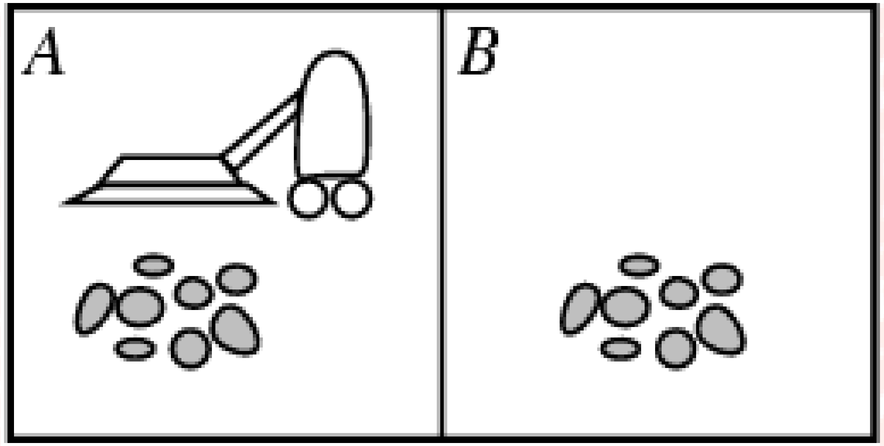
$$[f: \mathcal{P}^* \rightarrow \mathcal{A}]$$

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

agent = architecture + program

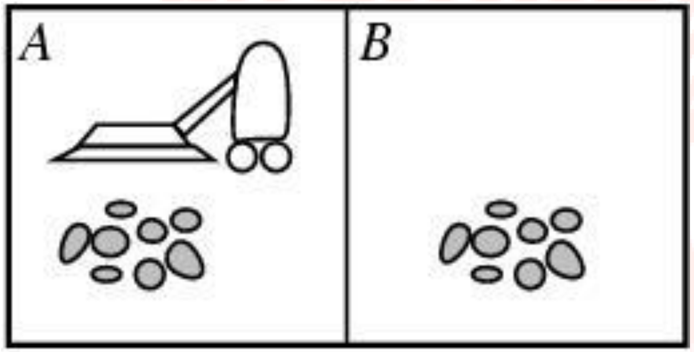


Vacuum-cleaner world



- Environment: square A and B
- Percepts: location and contents, e.g., [A,Dirty]
- Actions: *Left, Right, Suck, NoOp*

Vacuum-cleaner World



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
…	…

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*

- What is the right function?
- Can it be implemented in a small agent program?

Rational Agents

- A rational agent is one that does the right thing based on what it can perceive and the actions it can perform.
- What is the right thing?
  - Approximation: The right action is the one that will cause the agent to be most successful.
  - Measure of success?
- Performance measure: An objective criterion for success of an agent's behavior
  - E.g., performance measure of a vacuum-cleaner agent could be amount of dirt cleaned up, amount of time taken, amount of electricity consumed, amount of noise generated, etc.

Rational agents (cont.)

- What is rational at a given time depends on four things:
  - Performance measure,
  - Percept sequence to date (sensors),
  - Prior environment knowledge,
  - Actions
- An Ideal Rational Agent, for each possible percept sequence, should select an action that is expected to maximize its performance measure, given
  - the evidence provided by the percept sequence
  - its knowledge, both built-in and acquired

Properties of Agents

- Autonomy
  - can act without direct intervention by humans or other agents
  - Agent's behavior is determined by its own experience (with ability to learn and adapt)
  - Some aspects of the current situation must trigger a response.

➔Rational agent should be autonomous.

PEAS

- To design a rational agent we must specify its task environment.

- PEAS:
  - **Performance measure**: Goals/desires the agent should try to achieve
  - **Environment**: in which the agent exists
  - **Actuators**: Actions which may react the environment
  - **Sensors**: Percepts/observations of the environment

PEAS Example: Automated Taxi Driver

- Performance measure:
  - Safe, fast, legal, comfortable trip, maximize profits, ...
- Environment:
  - Roads, other traffic, pedestrians, customers

- Actuators:
  - **Steering wheel** (tay lái), **accelerator** (chân ga), **brake** (phanh), **signal**, **horn** (còi)
- Sensors:
  - **Cameras**, **sonar** (to detect distances to other cars and obstacles), **speedometer (km/h)**, **GPS**, **odometer** (for measuring the distance traveled), **engine sensors**, **keyboard**

PEAS Example: Internet shopping

- Performance measure:
  - price, quality, appropriateness, ..
- Environment:
  - current and future WWW sites, vendors, shippers
- Actuators:
  - display to user, follow URL, fill in form
- Sensors:
  - HTML pages (text, graphics, scripts)

PEAS Example: Medical diagnosis system

- Performance measure:
  - Healthy patient, minimize costs, lawsuits
- Environment:
  - Patient, hospital, staff
- Actuators:
  - Screen display (questions, tests, diagnoses, treatments, referrals)
- Sensors:
  - Keyboard (entry of symptoms, patient's answers)

PEAS Example: Part-picking robot

- Performance measure:
  - Percentage of parts in correct bins
- Environment:
  - conveyor belt with parts, bins.
- Actuators:
  - Jointed arm and hand.
- Sensors:
  - Camera, joint angle sensors.

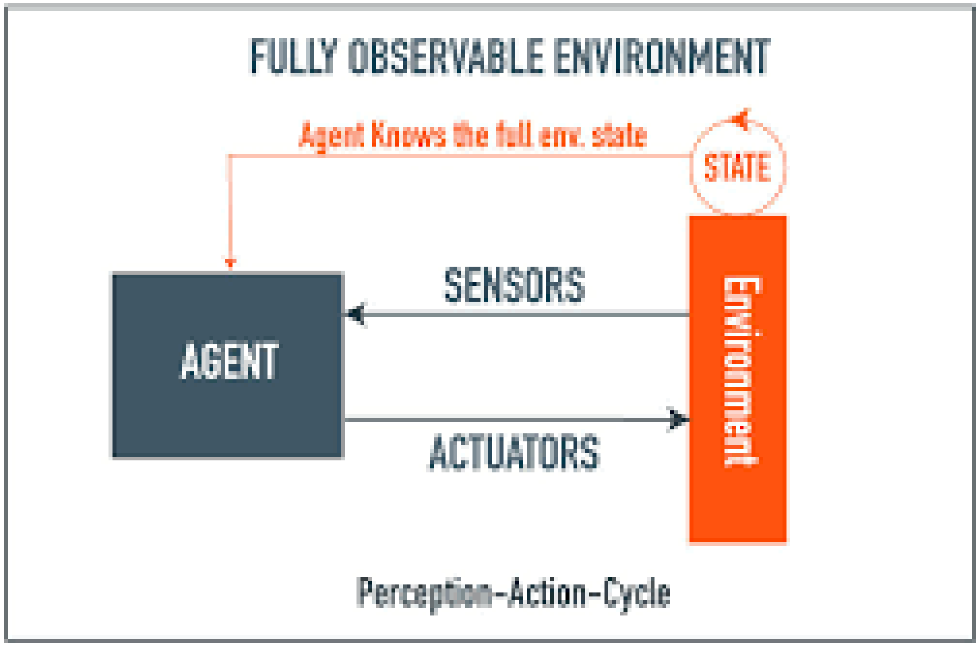
PEAS Example: Interactive English tutor

- Performance measure:
  - Maximize student's score on test
- Environment:
  - Set of students
- Actuators:
  - Screen display (exercises, suggestions, corrections)
- Sensors:
  - Keyboard (typed words)

Environment Types

Environment Types

- **Fully observable** (vs. **partially observable**)
  - An agent's sensors give it access to the complete state of the environment at any point in time.
  - Noisy and inaccurate sensors can result in partially observable environments.



Environment Types

- **Deterministic** (vs. **stochastic**)
  - The next state of the environment is completely determined by the current state and the action executed by the agent (**deterministic**).
  - If the environment is deterministic except for the actions of other agents, then the environment is strategic
  - Randomness and chance are common causes non-deterministic environments (**stochastic**).

Environment Types

- **Episodic** (vs. **sequential**):
  - The agent's experience is divided into atomic "episodes"
    - each episode consists of the agent perceiving and then performing a single action
    - the choice of action in each episode does not depend on the actions in prior episodes.
  - Games are often sequential requiring one to think ahead.
  - Expert advice systems – an **episode** is a single question and answer.
  - Classification task: an agent that has to spot defective parts on an assembly line bases each decision on the current part, regardless of previous decisions (**episode**)

➔ Most environments (and agents) are sequential

Environment Types

- **Static** (vs. **dynamic**):
  - The environment is unchanged while an agent is deliberating (between the time of perceiving and acting).
  - The environment is semi-dynamic if the environment itself does not change with the passage of time but the agent's performance score does
  - Time is an important factor in dynamic environments, since perceptions can become "stale".

Environment Types

- **Discrete** (vs. **continuous**): – state of the environment
  - An environment is discrete if there are a limited number of distinct, clearly defined percepts and actions.

- **Single agent** (vs. **multiagent**):
  - An environment is multiagent if more than one agents effect the each other's performance.
  - Multiagent environments can be competitive and/or cooperative.

Environment Types Example

Environment Types (Ex.)

- Solitaire:
- Internet Shopping Agents (ISAs) allow consumers to costlessly search many online retailers and buy at the lowest price
- Taxi driver:

Environment Types (Ex.)

Environment types of these agents?

	Solitaire	Internet shopping	Taxi Driver
Observable??			
Deterministic??			
Episodic??			
Static??			
Discrete??			
Single-agent??			

Environment types (Ex.)

Fully vs. partially observable: an environment is full observable when the sensors can detect all aspects that are relevant to the choice of action.

	Solitaire	Internet shopping	Taxi Driver
Observable??	FULL	PARTIAL	PARTIAL
Deterministic??			
Episodic??			
Static??			
Discrete??			
Single-agent??			

Environment Types (Ex.)

Deterministic vs. stochastic: if the next environment state is completely determined by the current state and the executed action then the environment is deterministic

	Solitaire	Internet shopping	Taxi Driver
Observable??	FULL	PARTIAL	PARTIAL
Deterministic??	YES	YES	NO
Episodic??			
Static??			
Discrete??			
Single-agent??			

Environment Types (Ex.)

Episodic vs. sequential: In an episodic environment the agent's experience can be divided into atomic steps where the agent perceives and then performs a single action. The choice of action depends only on the episode itself

	Solitaire	Internet shopping	Taxi Driver
Observable??	FULL	PARTIAL	PARTIAL
Deterministic??	YES	YES	NO
Episodic??	NO	NO	NO
Static??			
Discrete??			
Single-agent??			

In sequential environments, the current decision could affect all future decisions

Environment Types (Ex.)

Static vs. dynamic: If the environment can change while the agent is choosing an action, the environment is dynamic. Semi-dynamic if the agent's performance changes even when the environment remains the same.

	Solitaire	Internet shopping	Taxi Driver
Observable??	FULL	PARTIAL	PARTIAL
Deterministic??	YES	YES	NO
Episodic??	NO	NO	NO
Static??	YES	SEMI	NO
Discrete??			
Single-agent??			

Environment Types (Ex.)

Discrete vs. continuous: This distinction can be applied to the state of the environment, the way time is handled and to the percepts/actions of the agent.

	Solitaire	Internet shopping	Taxi Driver
Observable??	FULL	PARTIAL	PARTIAL
Deterministic??	YES	YES	NO
Episodic??	NO	NO	NO
Static??	YES	SEMI	NO
Discrete??	YES	YES	NO
Single-agent??			

Environment Types (Ex.)

Single vs. multi-agent: Does the environment contain other agents who are also maximizing some performance measure that depends on the current agent's actions?

	Solitaire	Internet shopping	Taxi Driver
Observable??	FULL	PARTIAL	PARTIAL
Deterministic??	YES	YES	NO
Episodic??	NO	NO	NO
Static??	YES	SEMI	NO
Discrete??	YES	YES	NO
Single-agent??	YES	NO	NO

Environment Types

- The environment type largely determines the agent design

- The simplest environment is
  - Fully observable, deterministic, episodic, static, discrete and single-agent.

- Most real situations are:
  - Partially observable, stochastic, sequential, dynamic, continuous and multi-agent.
    - The most hardest environment

Agent Types

Agent Types

- Four basic types in order of increasing generality:
  - Simple reflex agents
    - lookup table
    - if-then rules
  - Model-based reflex agents
  - Goal-based agents
  - Utility-based agents
- All these can be turned into learning agents.

Agent Types: Simple Table-Based Reflex

- Use a table to lookup where each percept is matched to an action

- Problems/Limitations?
  - table may be too big to generate and store
  - not adaptive to changes in the environment; instead table must be updated
  - can't make actions conditional
  - reacts only to current percept; no history kept

Table-lookup agent

function TABLE-DRIVEN-AGENT(*percept*) **returns** an action
  persistent: *percepts*, a sequence, initially empty
    table, a table of actions, indexed by percept sequences, initially fully specified

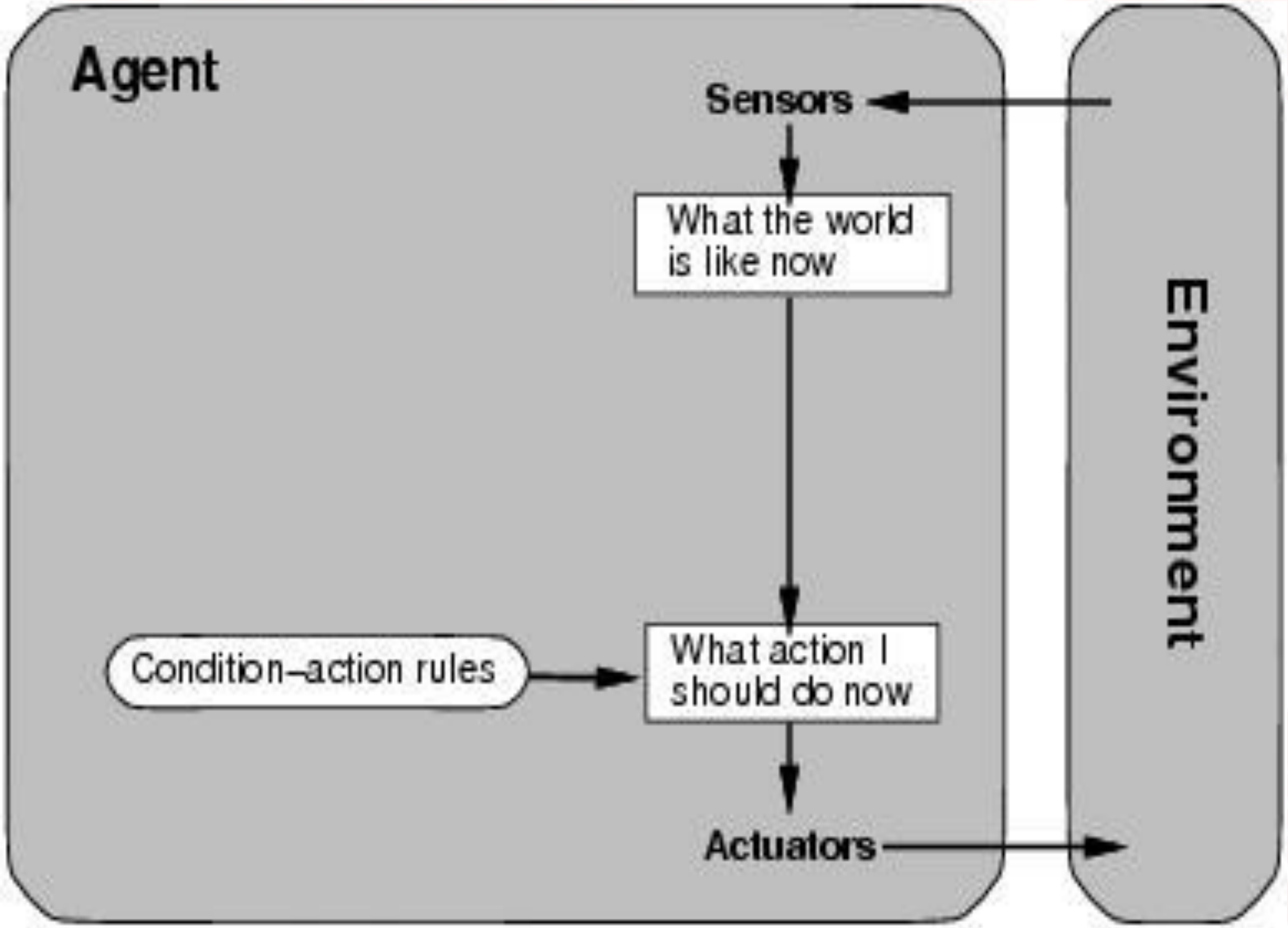
  append *percept* to the end of *percepts*
  action ← LOOKUP(*percepts*, *table*)
  return *action*

- Drawbacks:
  - Huge table
  - Take a long time to build the table
  - No autonomy
  - Even with learning, need a long time to learn the table entries

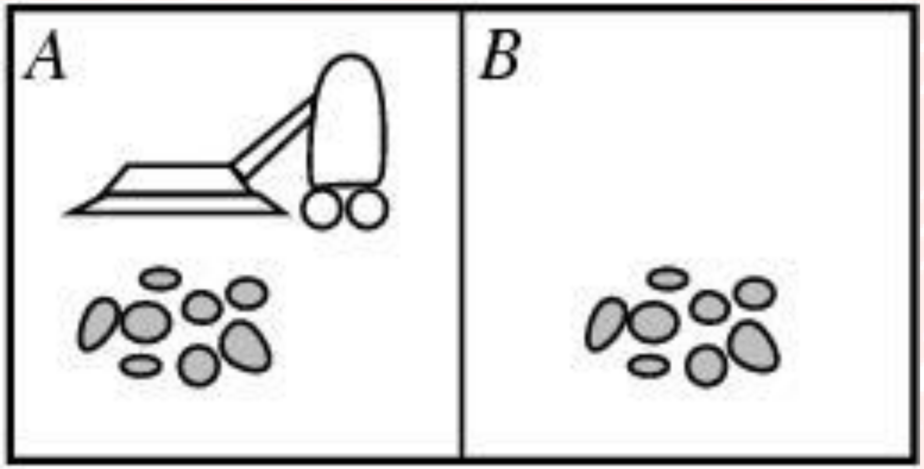
Agent Types: Simple Rule-Based Reflex

- Select action on the basis of *only the current* percept.
- No need to consider all percepts
- Implemented through *condition-action rules*
  - If dirty then suck
- Can adapt to changes in the environment by adding rules
- Problems/Limitations?
  - reacts only to current percept; no knowledge of non-perceptual parts of the current state

Agent Types: Simple Rule-based Reflex



The vacuum-cleaner world



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*

Agent Types: Simple reflex

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

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

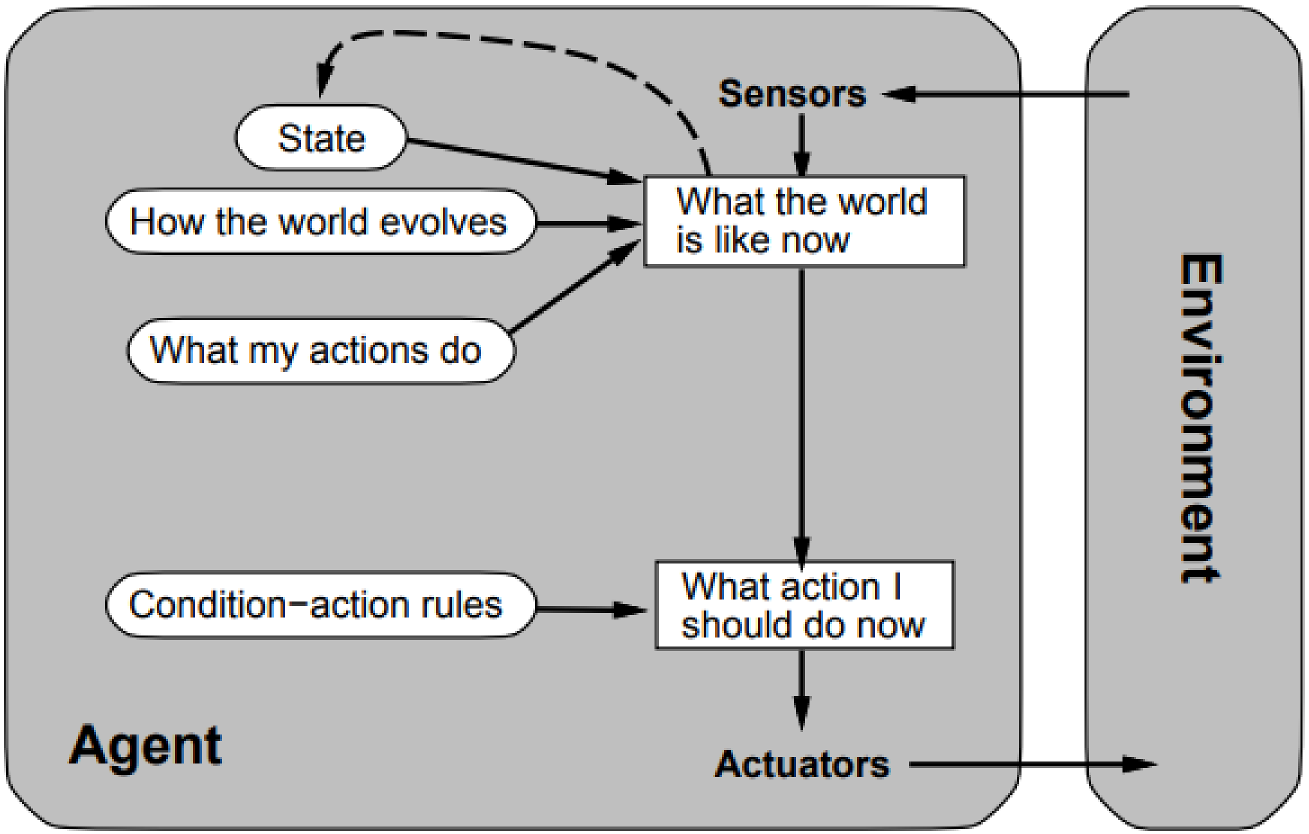
Will only work if the environment is fully observable otherwise infinite loops may occur.

Agent Types: Model-based reflex agents

- To tackle partially observable environments.
  - Maintain internal state
- Over time update state using world knowledge
  - How does the world change.
  - How do actions affect world.

  $\Rightarrow$Model of World
- Problems/Limitations?
  - not deliberative, agent types so far are reactive

Agent Types: Model-based reflex agents



Agent Types: Model-based Reflex Agents

function MODEL-BASED-REFLEX-AGENT(*percept*) **returns** an action
  persistent: *state*, the agent's current conception of the 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
  return *action*

Agent Types: Model-based Reflex Agents

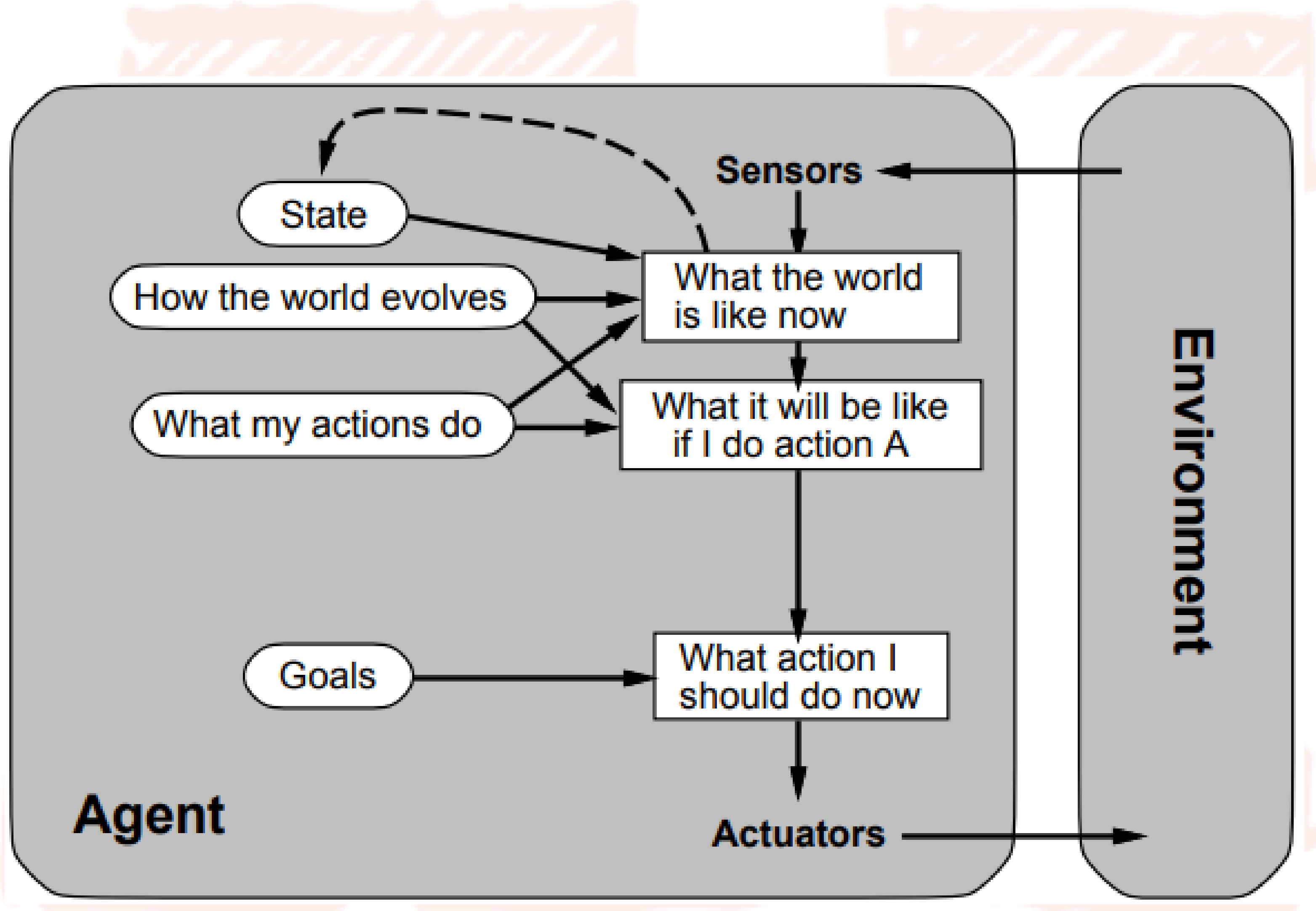
```
ModelBasedVacuumAgent(Agent):
Init:
  model = {“Loc_A” : None, “Loc_B” : None}
Action execute(location, state) {
  model[location] = state
if (model[“Loc_A”] == model[“Loc_B”] == “clean”)
  return “NoOp”
else if status == “dirty”          return “suck”
else if location == Loc_A          return “right”
else                               return “left”
}
```

Agent Types: Goal-based Agents

- Chose actions to achieve a desired goal
  - Search/planning often used

- Problems/Limitations?
  - May have to consider long sequences of possible actions before goal is achieved
  - Involves consideration of the future, "*What will happen if I do...?*"
  - How are competing goals treated?
  - What about degrees of success?

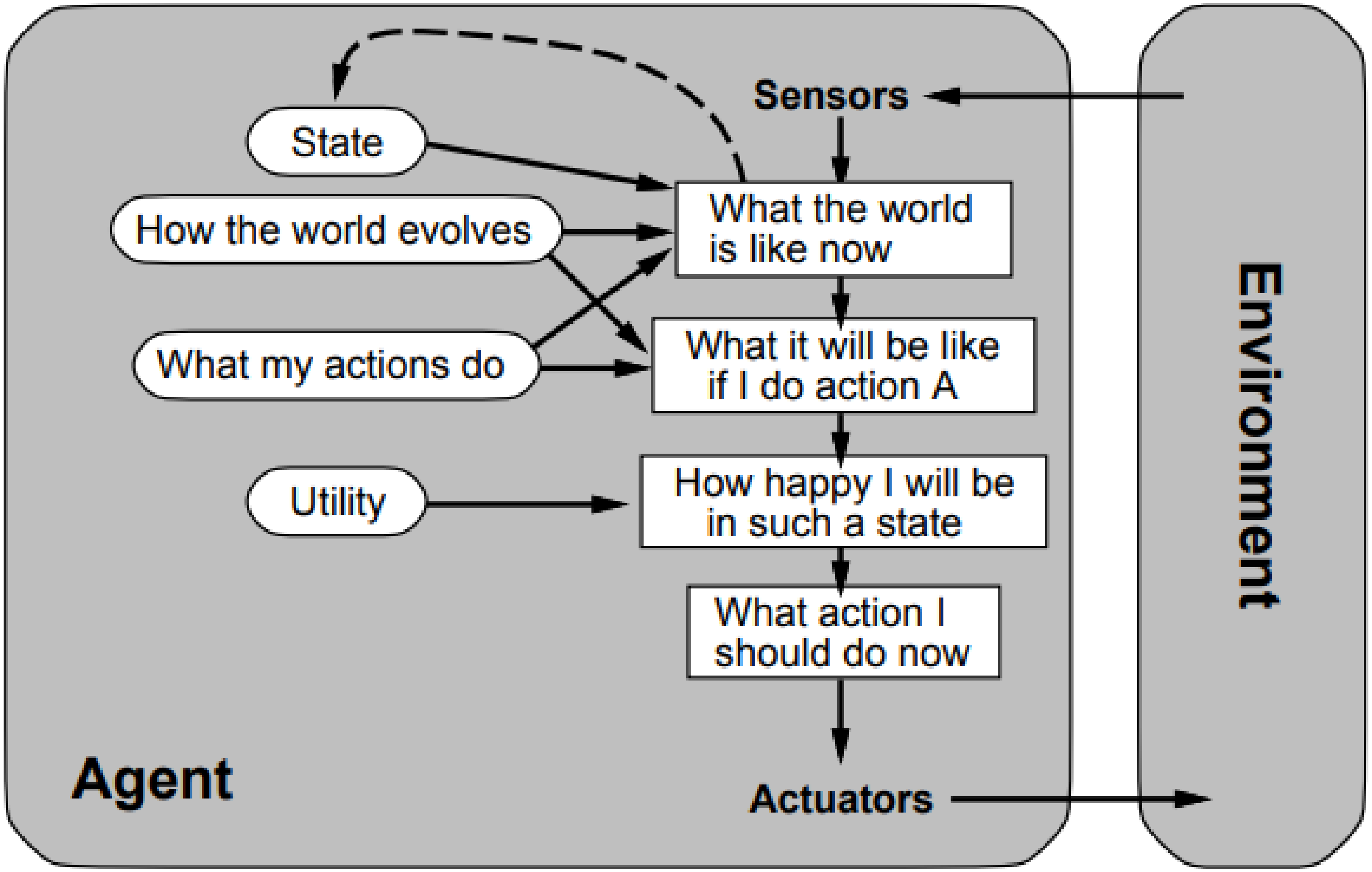
Agent Types: Goal-based



Agent Types: Utility-based Agents

- Goals are not always enough. Achieve goals while trying to maximize some utility value
- A utility function maps a state onto a real number which describes the associated degree of “happiness”, “goodness”, “success”.
- Allows decisions comparing choice between
  - Conflicting goals
  - Likelihood of success and importance of goal
- Where does the utility measure come from?
  - Economics: money
  - Biology: number of offspring
  - Your life?

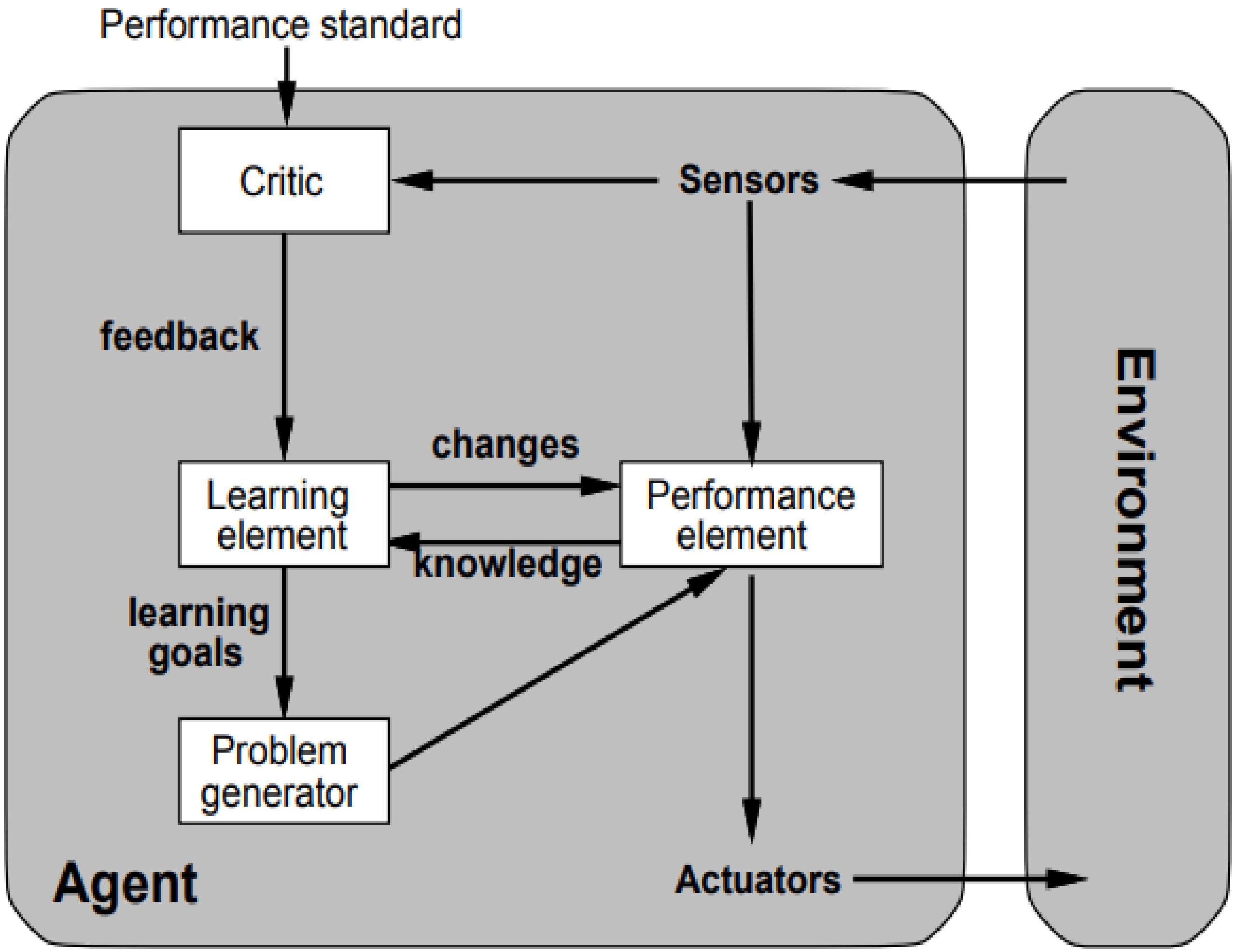
Agent types: Utility-based



Utility for Self-Driving Cars

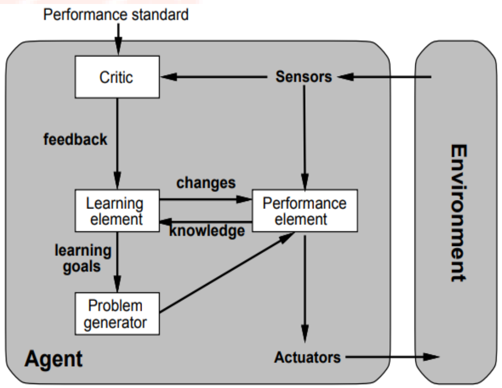
- What is the performance metric?
  - Safety – no accidents
  - Time to destination
  - What if accident is unavoidable? E.g.
    - Is it better to crash into an old person than in to a child?

Agent Types: Learning Agents



Agent Types: Learning (cont.)

- Learning element:
  - Introduce improvements in performance element.
  - Critic provides feedback on agents performance based on fixed performance standard.
- Performance element:
  - Selecting actions based on percepts.
- Problem generator:
  - Suggests actions that will lead to new and informative experiences.
  - Exploration vs. exploitation



Sumary

- 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
- 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

FACULTY OF INFORMATION TECHNOLOGY