

# CS 5/7320 Artificial Intelligence

## Intelligent Agents AIMA Chapter 2

---

Slides by Michael Hahsler  
based on slides by Svetlana Lazepnik  
with figures from the AIMA textbook.



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

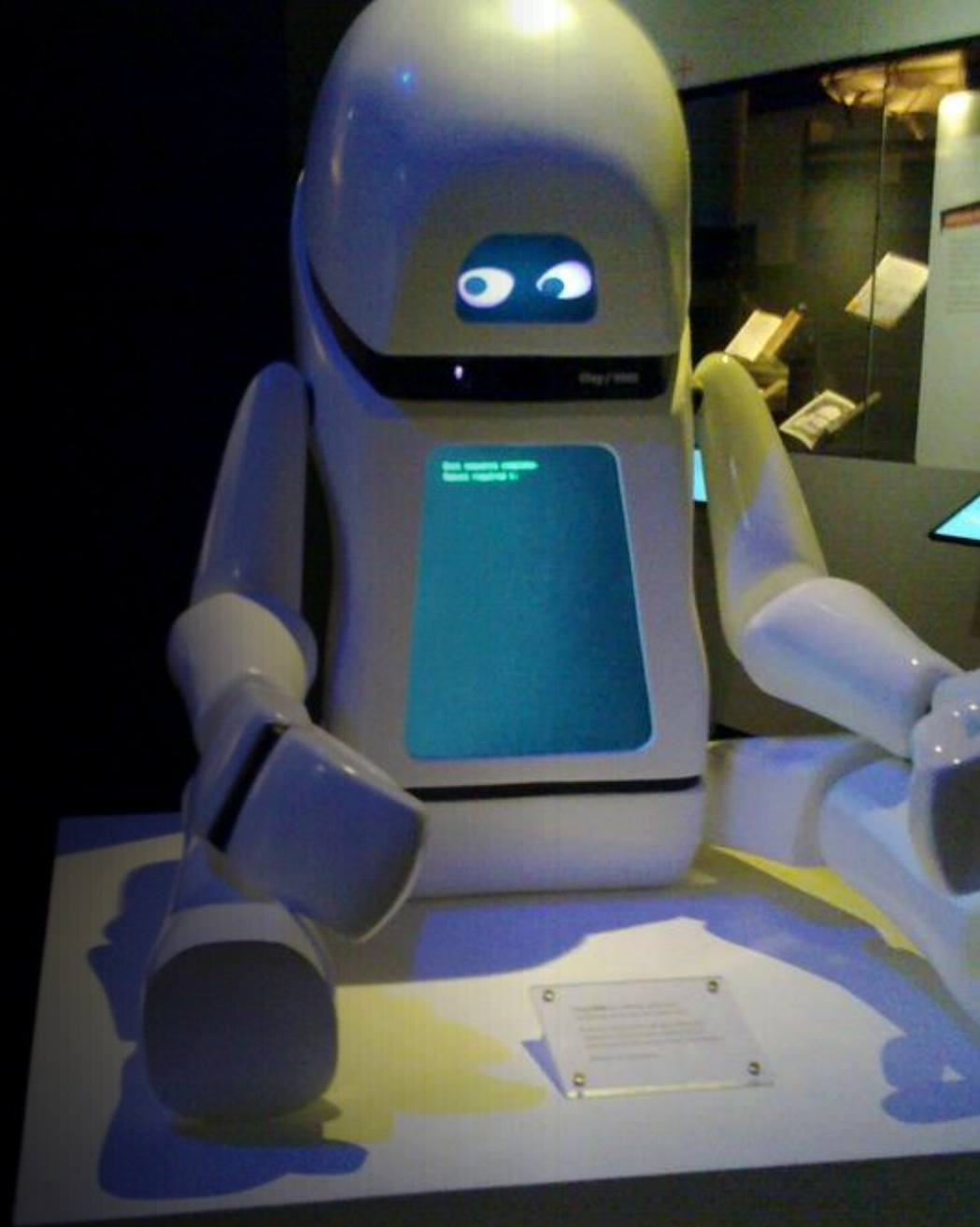
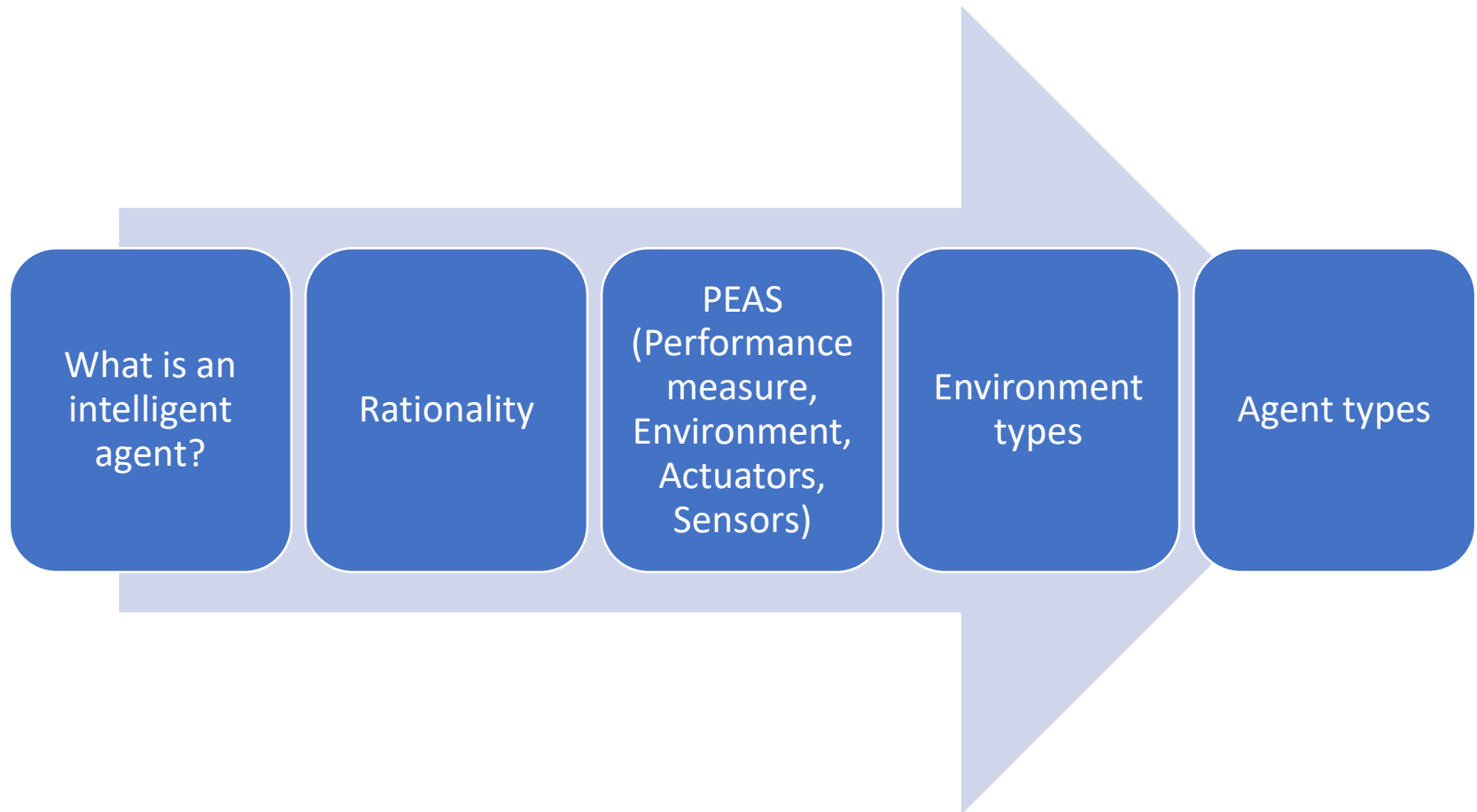
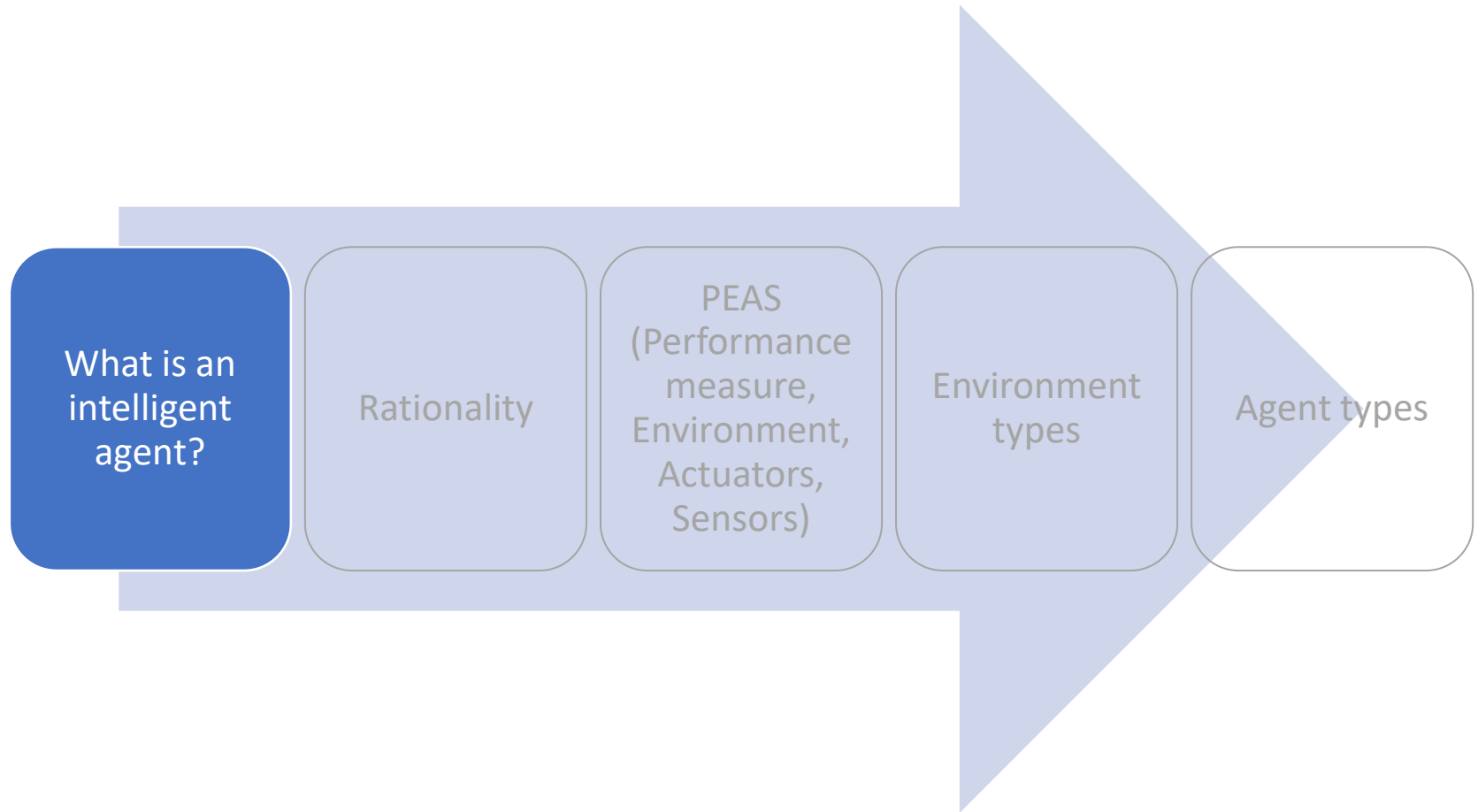


Image: "Robot at the British Library Science Fiction Exhibition"  
by BadgerGravling

# Outline

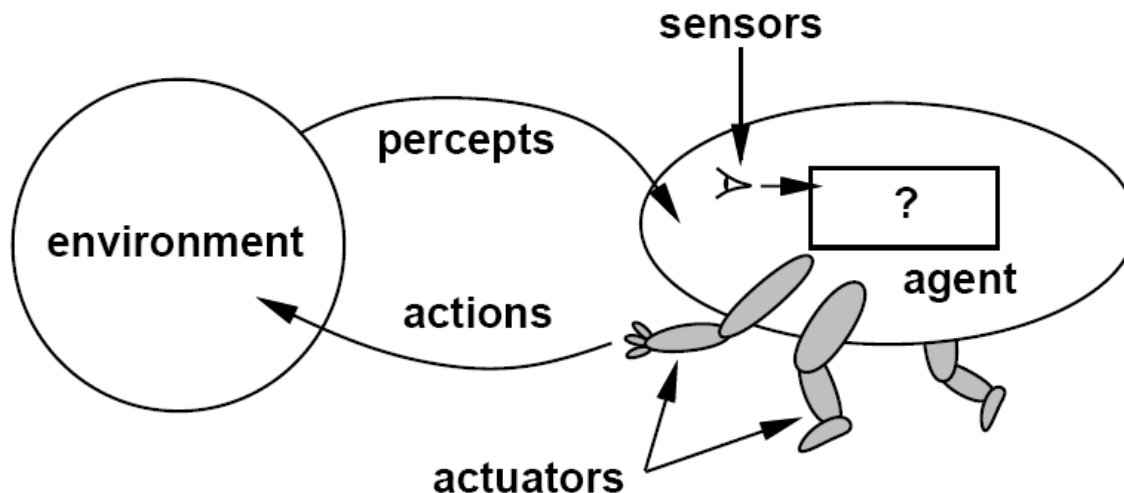


# Outline



# What is an Agents?

- An **agent** is anything that can be viewed as **perceiving** its **environment** through **sensors** and **acting** upon that environment through **actuators**.

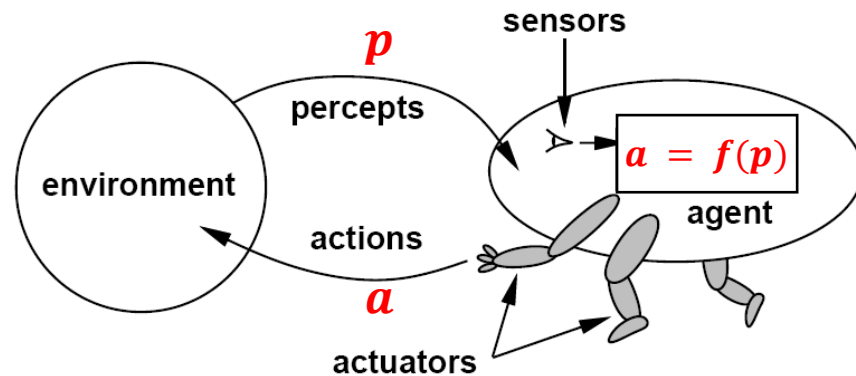


- **Control theory:** A **closed-loop control system** (= feedback control system) is a set of mechanical or electronic devices that automatically regulate a process variable to a desired state or set point without human interaction. The agent is called a controller.
- **Softbot:** Agent is a software program that runs on a host device.

# Agent Function and Agent Program

The **agent function** maps from the set of all possible *percept sequences*  $P^*$  to the *set of actions*  $A$  formulated as an abstract mathematical function.

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



The **agent program** is a concrete implementation of this function for a given physical system.

Agent = architecture (hardware) + agent program (implementation of  $f$ )

- Sensors
- Memory
- Computational power

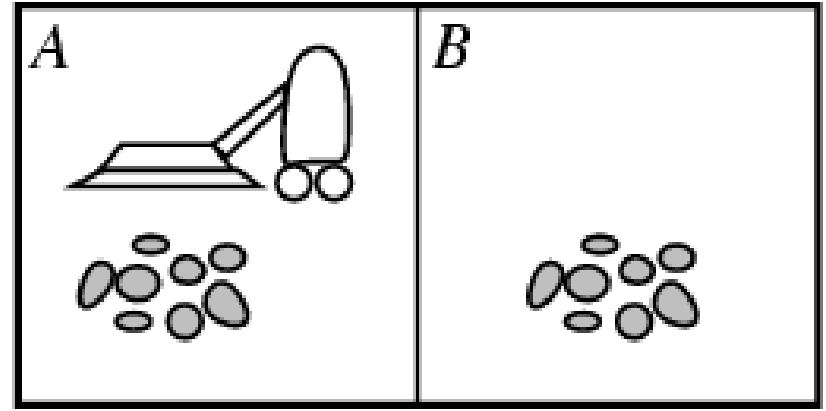
# Example: Vacuum-cleaner World

- **Percepts:**

Location and status,  
e.g., [A, Dirty]

- **Actions:**

Left, Right, Suck, NoOp



Most recent  
Percept  $p$

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

Percept Sequence	Action
[A, Clean]	Right
[A, Dirty]	Suck
...	
[A, Clean], [B, Clean]	Left
...	
[A, Clean], [B, Clean], [A, Dirty]	Suck
...	

Implemented agent program:

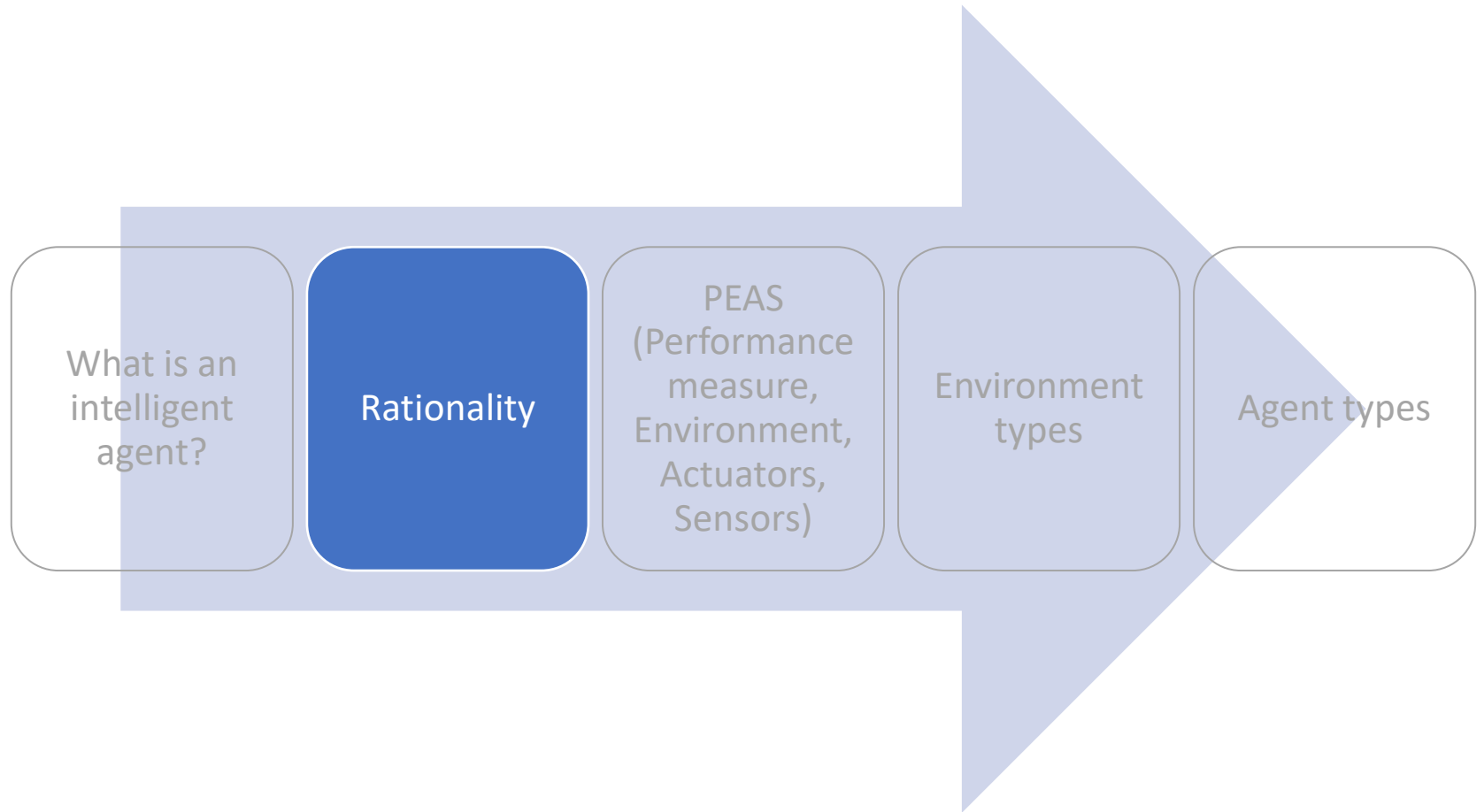
**function Vacuum-Agent**([location, status])

returns an **action**  $a$

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

**Problem:** This table can become infinitively large!

# Outline



# Rational Agents: What is Good Behavior?

## Foundation

- **Consequentialism**: Evaluate behavior by its consequences.
- **Utilitarianism**: Maximize happiness and well-being.

## Definition of a rational agent:

*“For each possible percept sequence, a rational agent should select an **action** that is **expected to maximize its performance measure**, given the evidence provided by the **percept sequence** and the **agent’s built-in knowledge**.”*

- **Performance measure**: An *objective* criterion for success of an agent's behavior (often called utility function or reward function).
- **Expectation**: Outcome averaged over all possible situations that may arise.

## This means:

- **Rationality  $\neq$  Omniscience** (rational agents can make mistakes if percepts and knowledge do not suffice to make a good decision)
- **Rationality  $\neq$  Perfection** (rational agents maximize **expected** outcomes not actual outcomes)
- **It is rational to explore and learn** (i.e., **use percepts** to supplement prior knowledge and become autonomous)



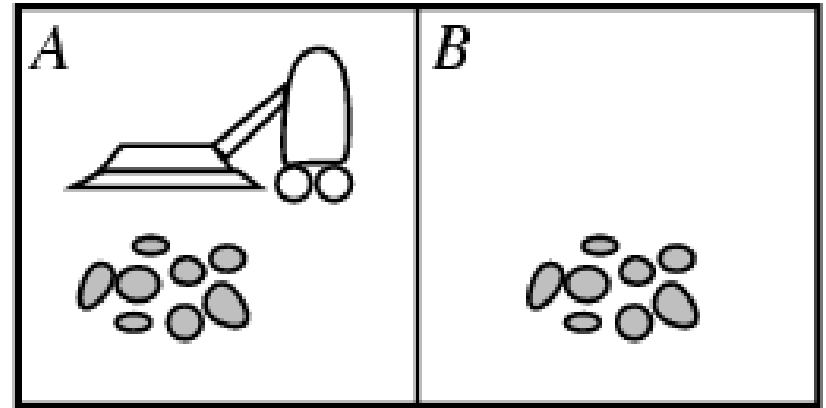
# Example: Vacuum-cleaner World

- **Percepts:**

Location and status,  
e.g., [A, Dirty]

- **Actions:**

Left, Right, Suck, NoOp



Agent function:

Percept Sequence	Action
[A, Clean]	Right
[A, Dirty]	Suck
...	
[A, Clean], [B, Clean]	Left
...	

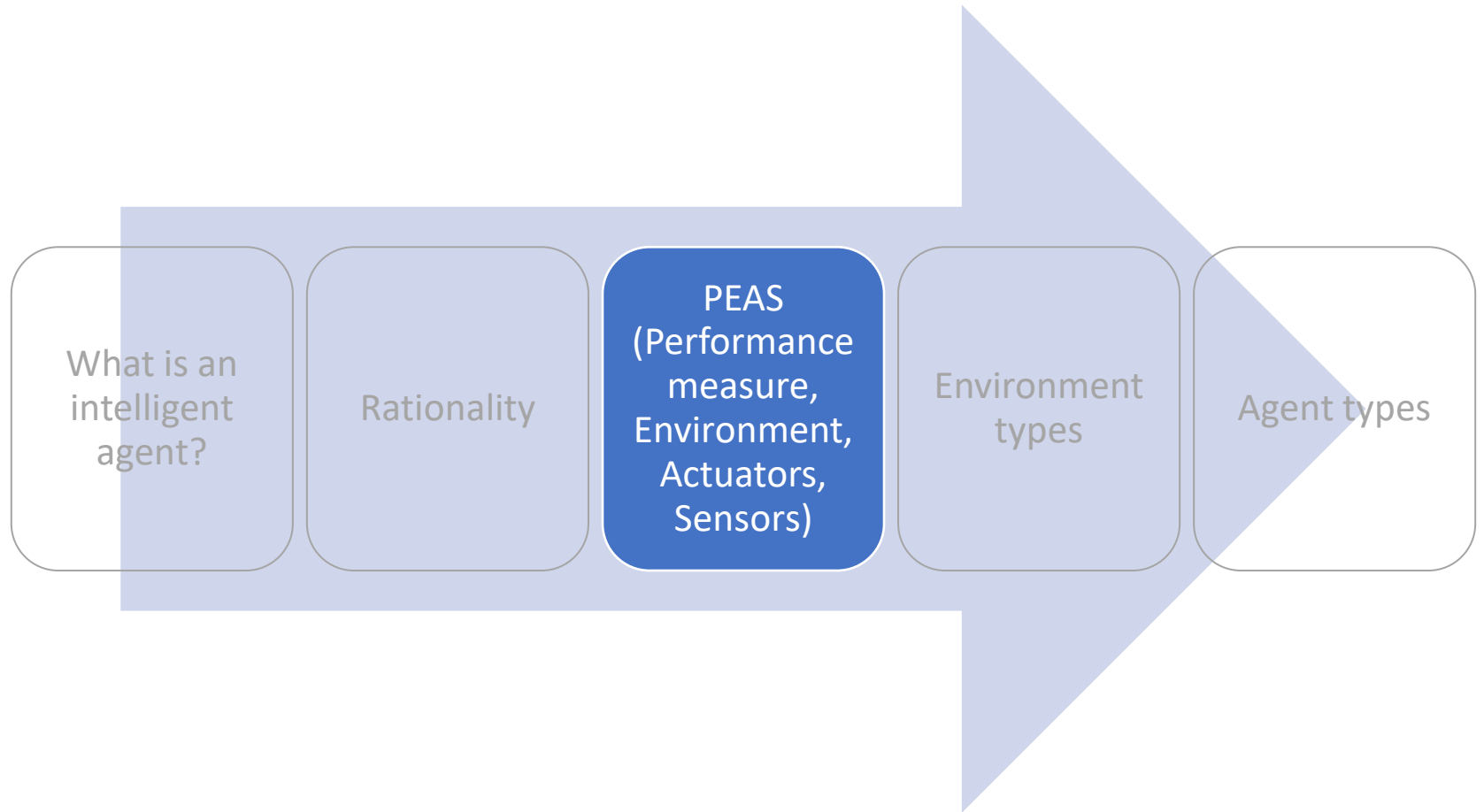
Implemented agent program:

```
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
```

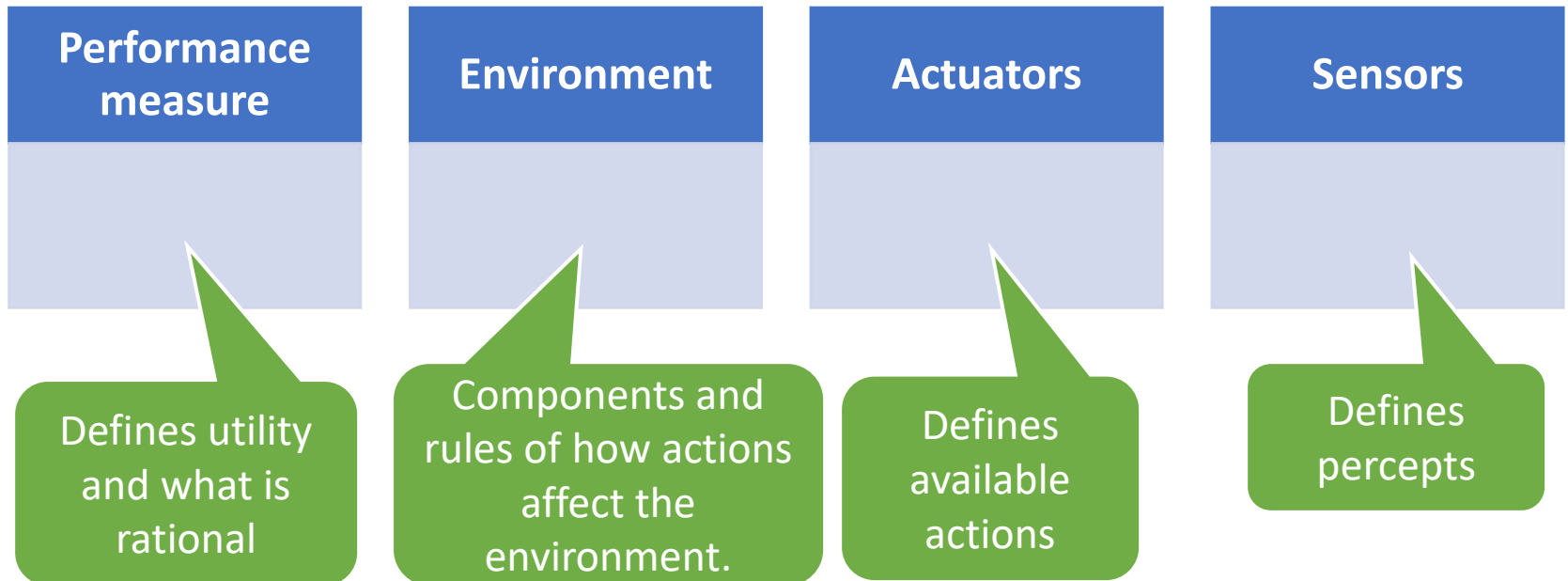
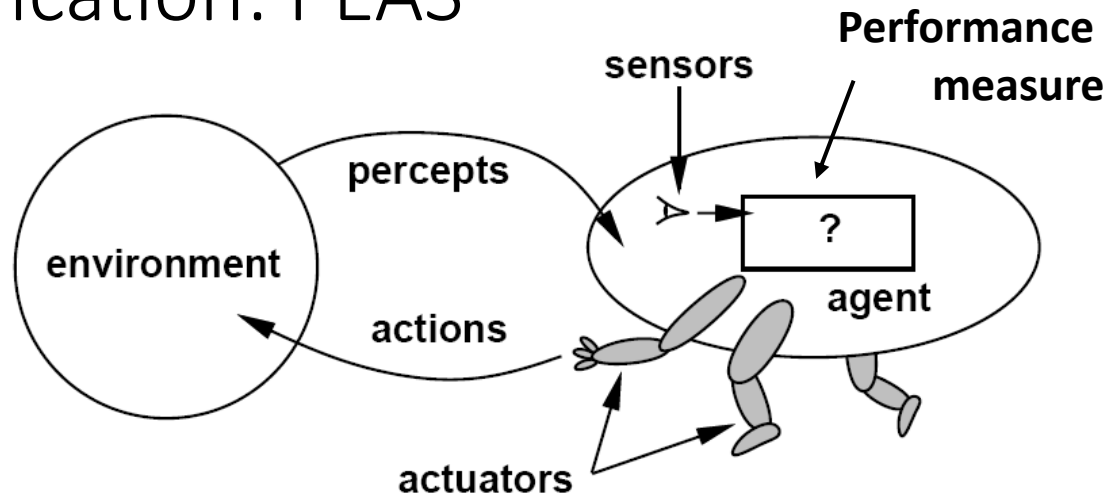
**What could be a performance measure?**

**Is this agent program rational?**

# Outline



# Problem Specification: PEAS



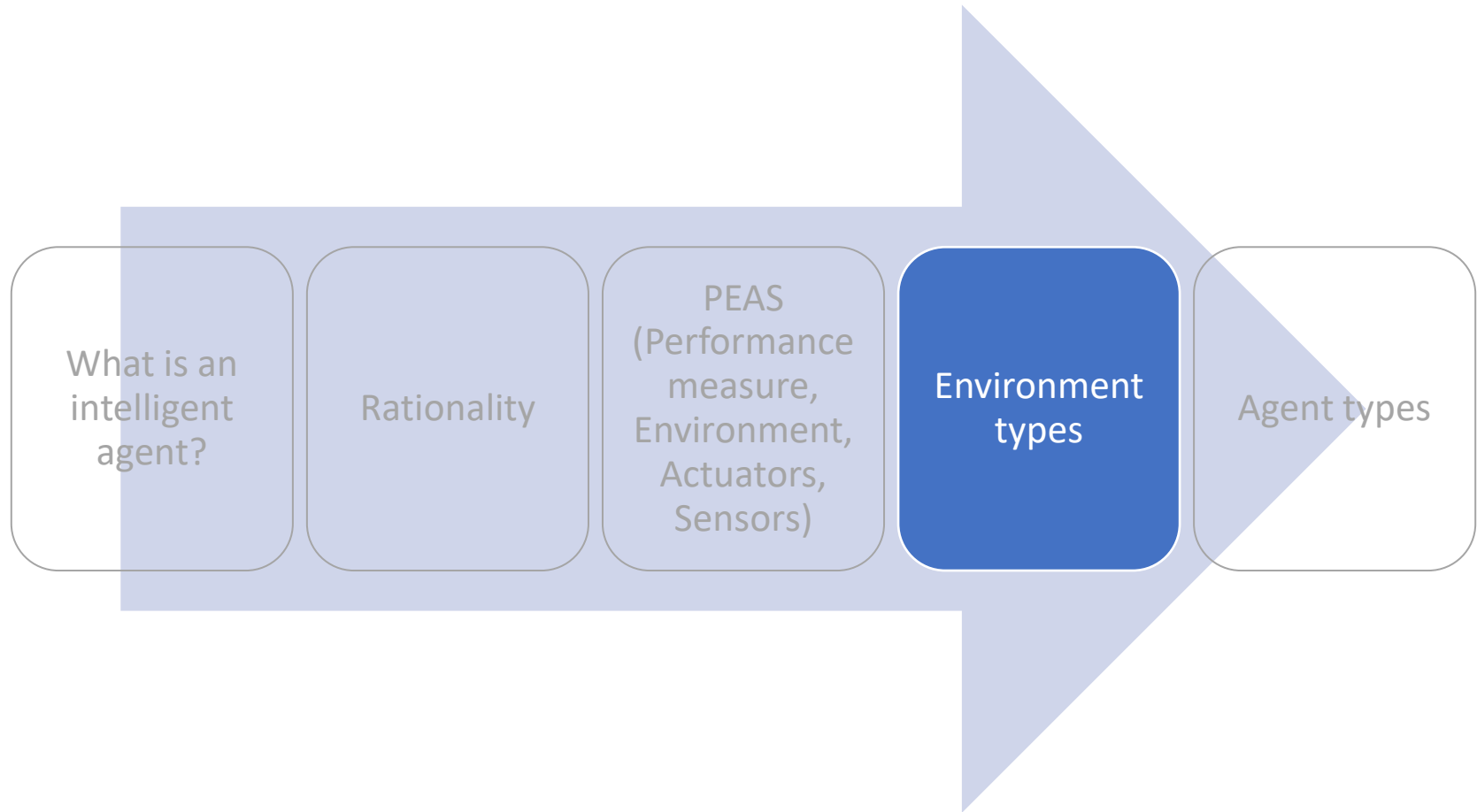
# Example: Automated Taxi Driver

Performance measure	Environment	Actuators	Sensors
<ul style="list-style-type: none"><li>• Safe</li><li>• fast</li><li>• legal</li><li>• comfortable trip</li><li>• maximize profits</li></ul>	<ul style="list-style-type: none"><li>• Roads</li><li>• other traffic</li><li>• pedestrians</li><li>• customers</li></ul>	<ul style="list-style-type: none"><li>• Steering wheel</li><li>• accelerator</li><li>• brake</li><li>• signal</li><li>• horn</li></ul>	<ul style="list-style-type: none"><li>• Cameras</li><li>• sonar</li><li>• speedometer</li><li>• GPS</li><li>• Odometer</li><li>• engine sensors</li><li>• keyboard</li></ul>

# Example: Spam Filter

Performance measure	Environment	Actuators	Sensors
<ul style="list-style-type: none"><li>• Accuracy: Minimizing false positives, false negatives</li></ul>	<ul style="list-style-type: none"><li>• A user's email account</li><li>• email server</li></ul>	<ul style="list-style-type: none"><li>• Mark as spam</li><li>• delete</li><li>• etc.</li></ul>	<ul style="list-style-type: none"><li>• Incoming messages</li><li>• other information about user's account</li></ul>

# Outline



# Environment Types

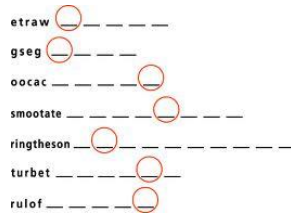
- **Fully observable (vs. partially observable):** The agent's sensors give it access to the complete state of the environment at each point in time.
- **Deterministic (vs. stochastic):** The next state of the environment is completely determined by the current state and the agent's action.
  - **Strategic:** the environment mechanics are deterministic, but the next state is also determined by the actions of other agents who follow their own strategy. This makes the environment look stochastic to our agent.
- **Episodic (vs. sequential):** Episode = get precept + do action. The agent's choice of action in one episode does not affect the next episodes.

# Environment Types

- **Static (vs. dynamic):** The environment is not changing while an agent is deliberating.
  - **Semidynamic:** the environment does not change while deliberating, but the agent's performance score depends on how fast it acts.
- **Discrete (vs. continuous):** The environment provides a fixed number of distinct percepts, actions, and environment states.
  - Time can also evolve in a discrete or continuous fashion.
- **Single agent (vs. multi-agent):** An agent operating by itself in an environment.
- **Known (vs. unknown):** The agent knows the rules of the environment.



# Examples of Different Environments



Word jumble solver



Chess with a clock



Scrabble



Taxi driving

Observable

Fully

Fully

Partially

Partially

Deterministic

Deterministic

Strategic

Stochastic  
+Strategic

Stochastic

Episodic

Episodic

Sequential

Sequential

Sequential

Static

Static

Semidynamic

Static

Dynamic

Discrete

Discrete

Discrete

Discrete

Continuous

Single agent

Single

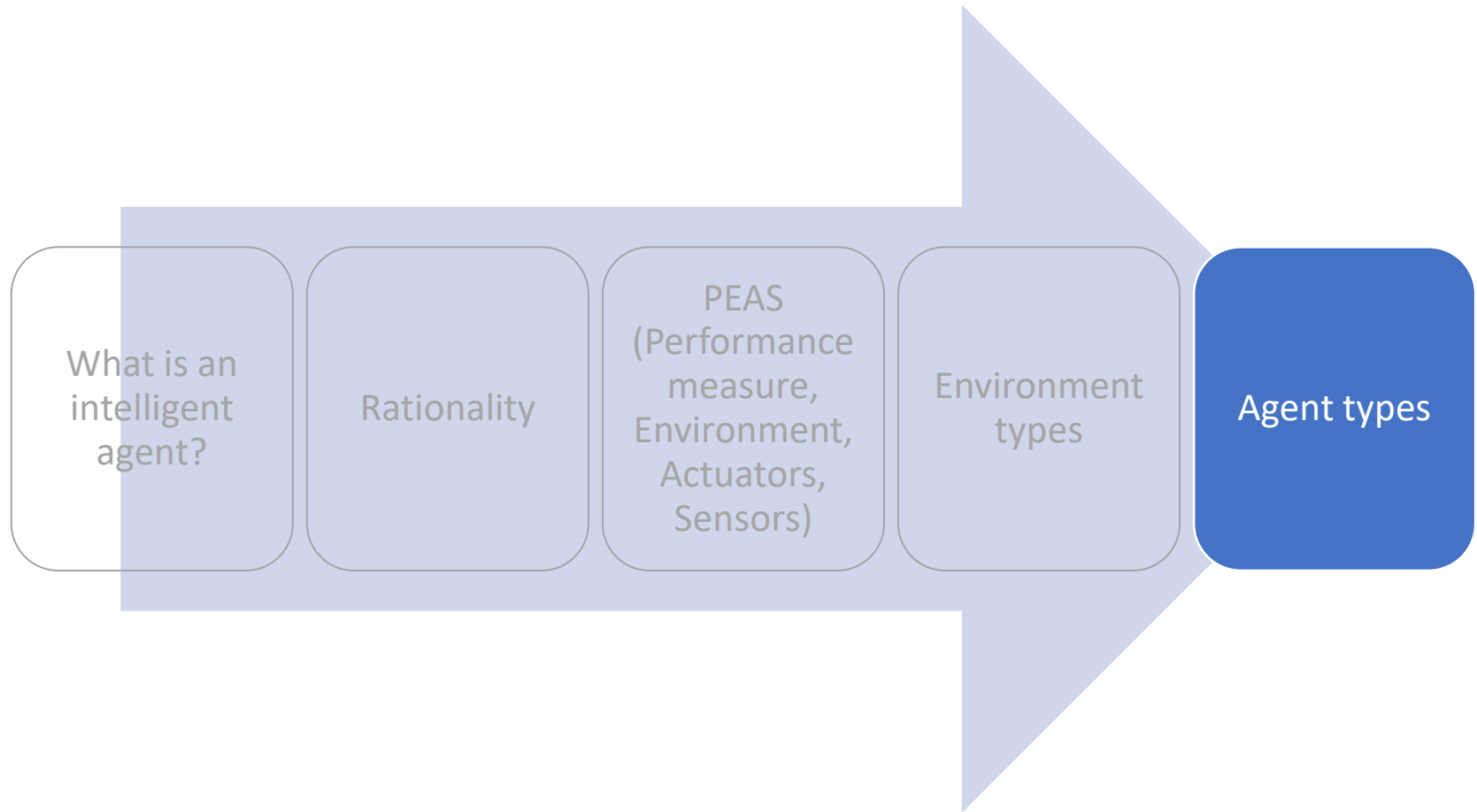
Multi

Multi

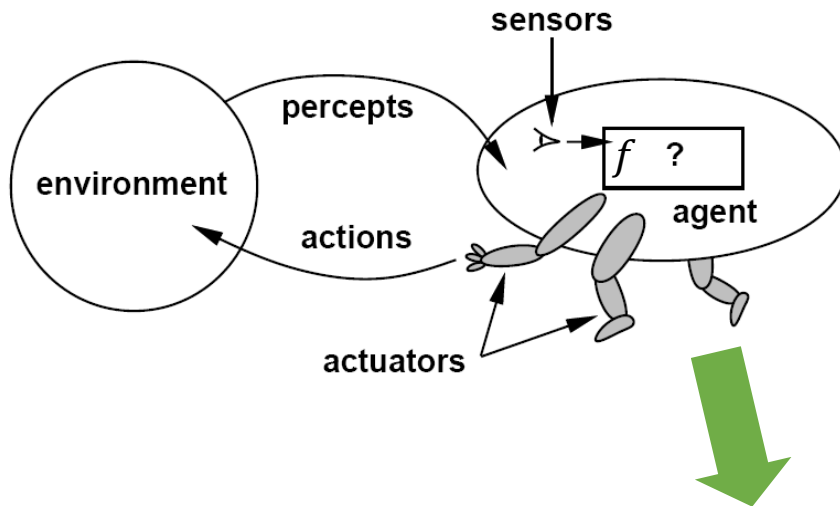
Multi

Actions have long-term effects.

# Outline



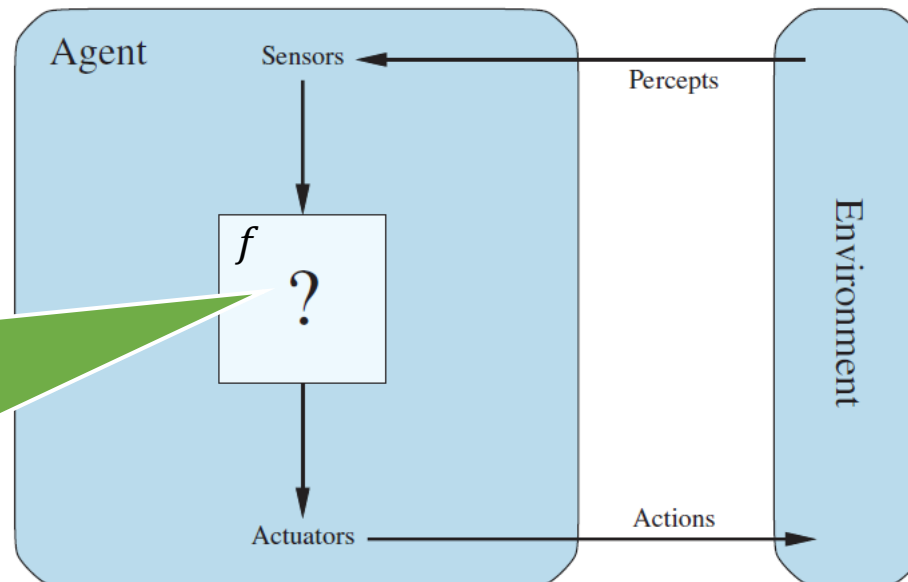
# Designing a Rational Agent



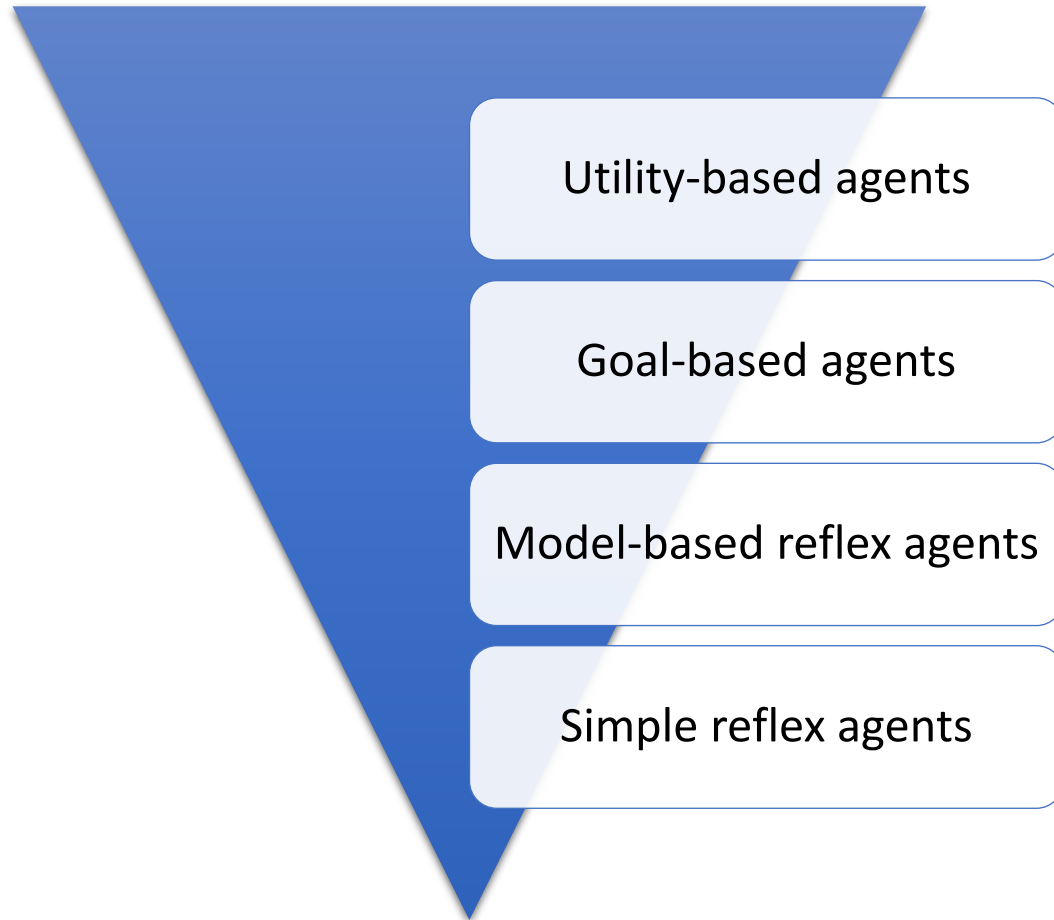
Remember the definition of a rational agent:

*“For each possible percept sequence, a rational agent should select an **action** that is **expected to maximize its performance measure**, given the evidence provided by the **percept sequence** and the **agent’s built-in knowledge**.”*

- Assess performance measure
- Remember percept sequence
- Built-in knowledge

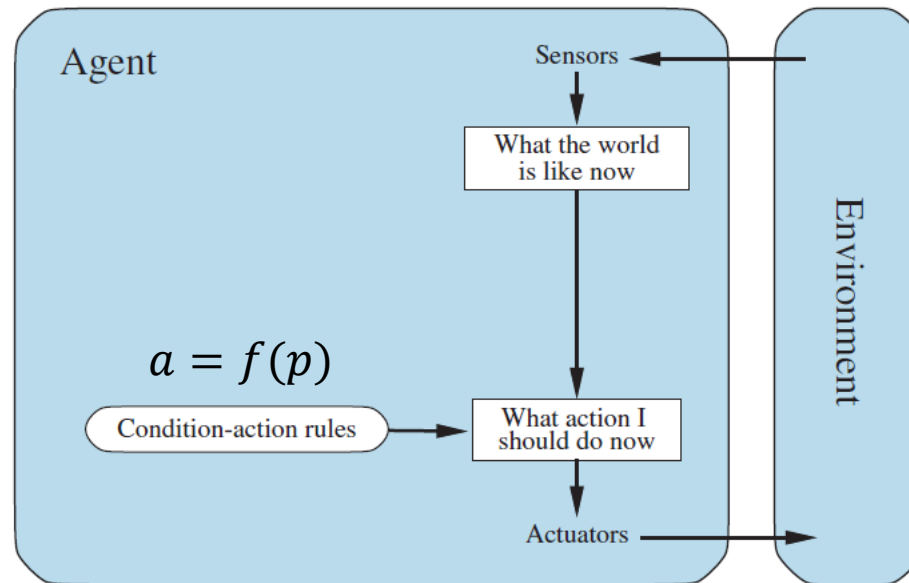


# Hierarchy of Agent Types



# Simple Reflex Agent

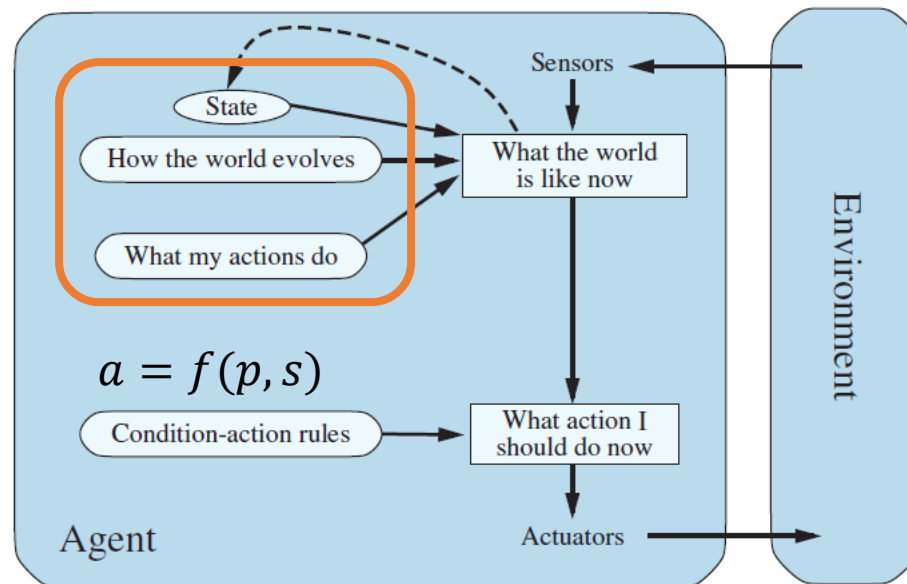
- Uses only built-in knowledge in the form of **rules** that select action only **based on the current percept**. This is typically very fast!
- The agent does not know about the performance measure, but well-designed rules can lead to good performance.
- The agent needs no memory and ignores all past percepts.



**Example:** A simple vacuum cleaner that uses rules based on its current sensor input.

# Model-based Reflex Agent

- Maintains a **state variable** to keep track of aspects of the environment that cannot be currently observed. I.e., it has memory and knows how the environment reacts to actions.
- There is now more information for the **rules** to make better decisions.

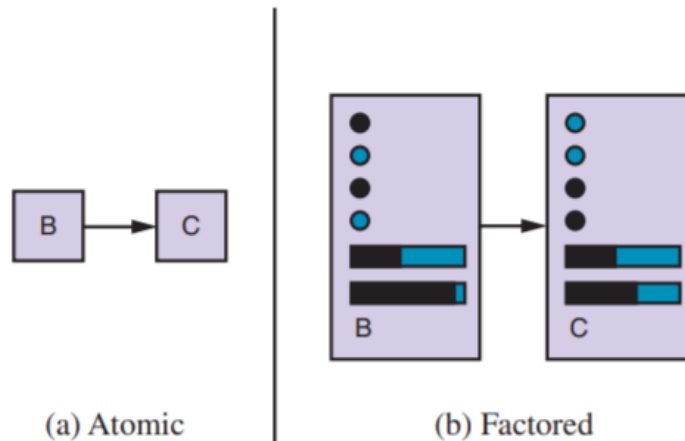


**Example:** A vacuum cleaner that remembers where it has already cleaned.

# State Representation

States help to keep track of the environment. The representation can be

- **Atomic:** Just a label for a black box. E.g., A, B
- **Factored:** A vector of attribute values. E.g., [location = left, status = clean, temperature = 75 deg. F]



**State Space:** The set of all possible states  $S$ . This set is typically very large!

# Old-school vs. Smart Thermostat



Old-school thermostat

**Percepts**

**States**



Smart thermostat

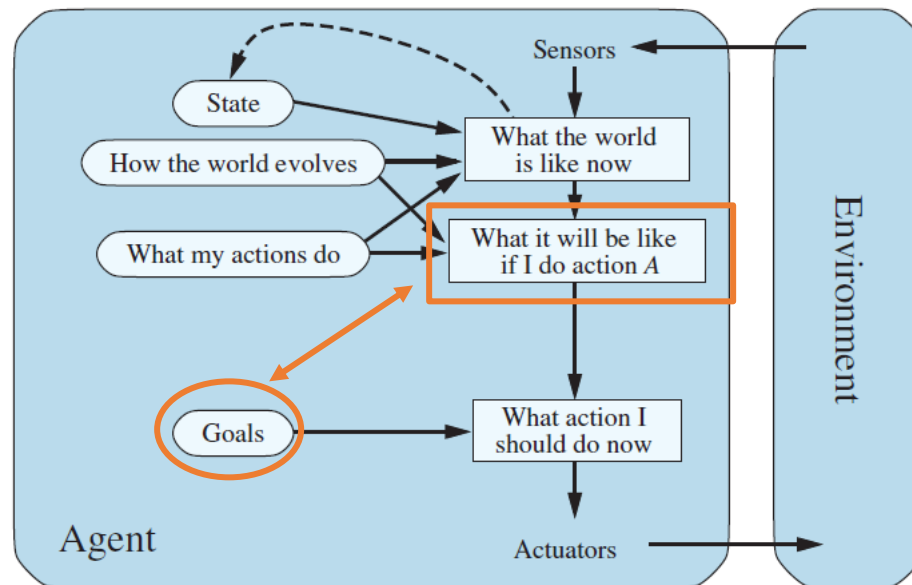
**Percepts**

**States**



# Goal-based Agent

- The agent has the task to reach a defined **goal state**.
- The agent needs to move towards the goal. It can use **search algorithms** to plan actions that lead to the goal.
- The performance measure is typically the cost to reach the goal.

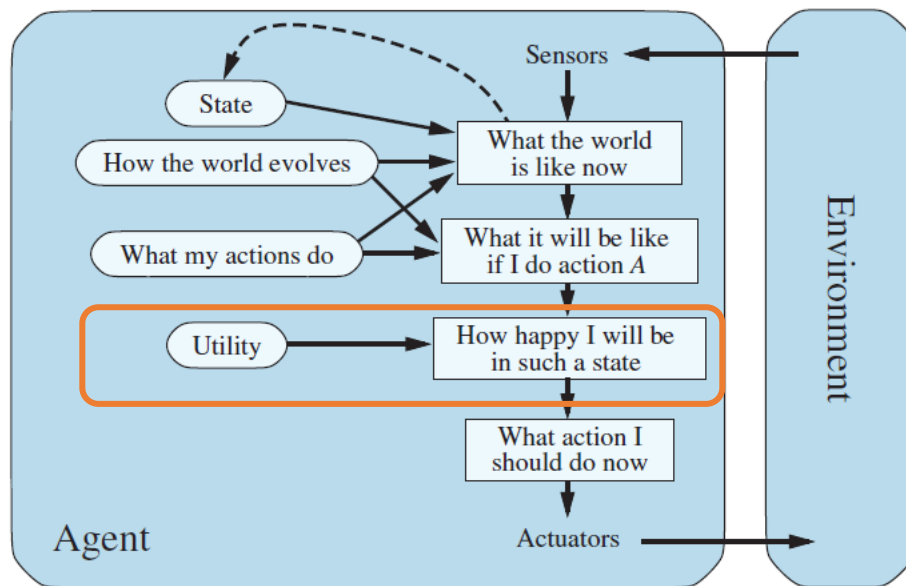


$$a_s = \operatorname{argmin}_{a \in A} [\operatorname{cost}(s, s_1, s_2, \dots, s_n | s_n = \text{goal})]$$

**Example:** Solving a puzzle. What action gets me closer to the solution?

# Utility-based Agent

- The agent uses a utility function to evaluate the **desirability of each possible states**. This is typically expressed as the reward of being in a state  $R(s)$ .
- Performance measure: Choose actions to maximize expected utility over time (i.e., stay in desirable states).



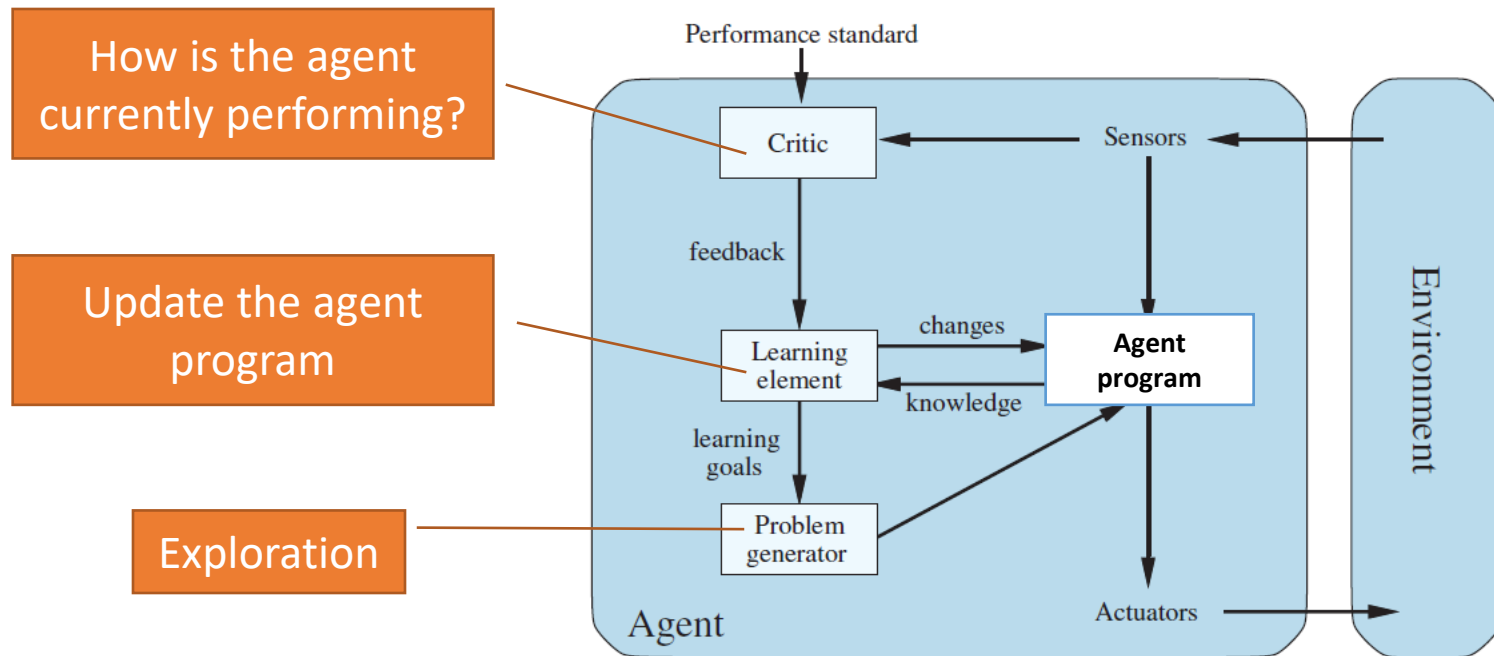
$$a = \operatorname{argmax}_{a \in A} \mathbb{E} \left[ \underbrace{\sum_{t=0}^T \gamma^t r_t}_{\text{Expected future reward}} \right]$$

**Techniques:** Markov decision processes, reinforcement learning

**Example:** An autonomous Mars rover prefers states where its battery is not critically low.

# Agents that Learn

The **learning element** modifies the agent program (reflex-based, goal-based, or utility-based) to improve its performance.



# Smart Thermostat



Change temperature when you are too cold/warm.

Reflex Agent?

Goal-based?

Utility-based?

## Smart thermostat

### Percepts

- Temp: deg. F
- Outside temp.
- Weather report
- Energy curtailment
- Someone walking by
- Someone changes temp.
- Day & time
- ...

### States

#### Factored states

- Estimated time to cool the house
- Someone home?
- How long till someone is coming home?
- A/C: on, off

# What Type of Intelligent Agent is this?

Features are:

- Control via App
- Cleaning Modes
- Navigation
- Mapping
- Boundary blockers



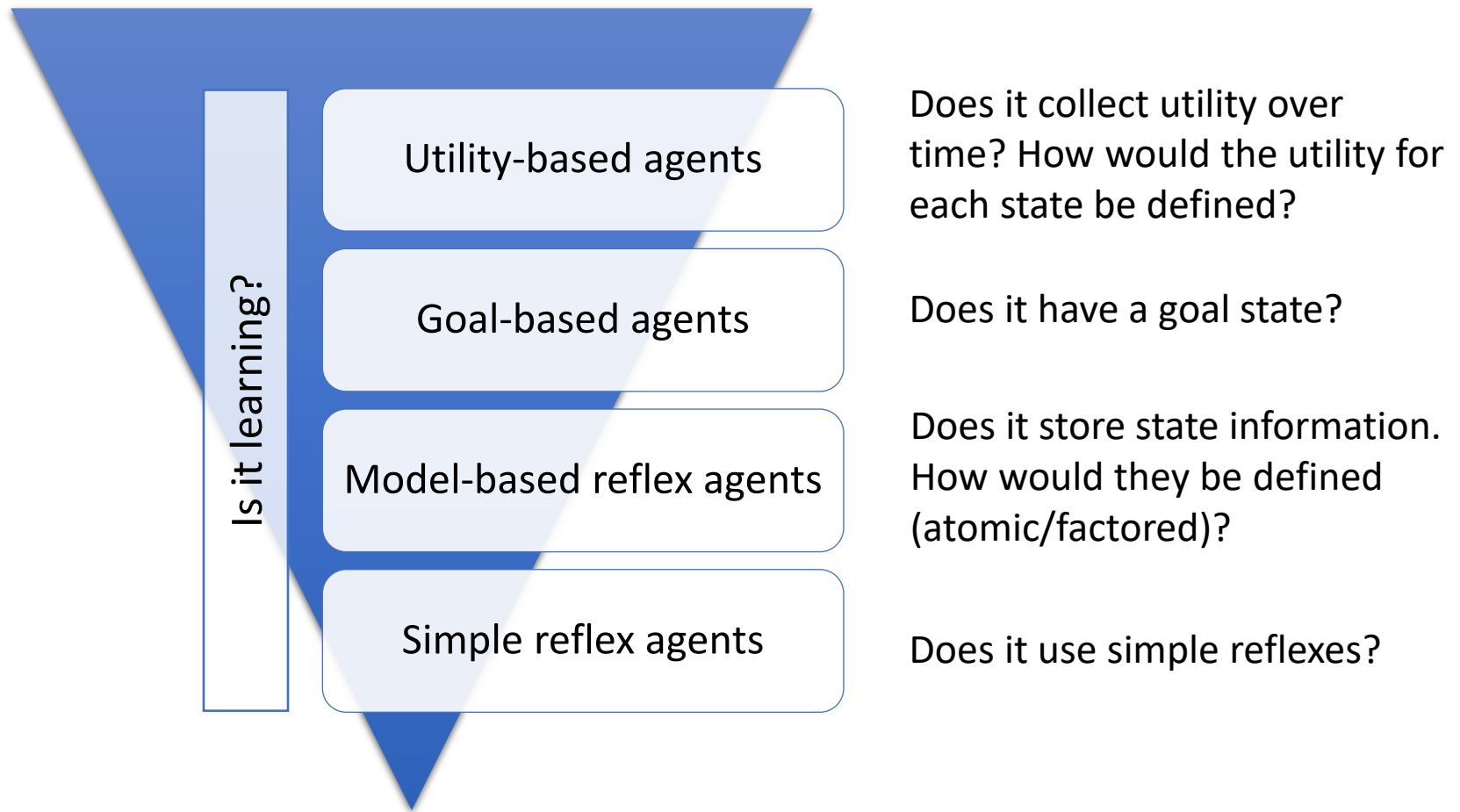
iRobot's Roomba brand has become as synonymous with robot vacuum as Q-tips is with cotton swabs. The Wi-Fi-enabled Roomba 960 is ample evidence why. It turns a tiresome chore into something you can almost look forward to. With three cleaning modes and dirt-detecting sensors, it kept all the floor surfaces in our testing immaculate, and its camera-driven navigation and mapping were superb. Its easy-to-use app provides alerts and detailed cleaning reports. The ability to control it with Amazon Alexa and Google Home voice commands are just the cherry on top.

Source: <https://www.techhive.com/article/3269782/best-robot-vacuum-cleaners.html>

# PEAS Description of a Modern Robot Vacuum

Performance measure	Environment	Actuators	Sensors

# What Type of Intelligent Agent is a Modern Robot Vacuum?



# What Type of Intelligent Agent is this?



Default (GPT-3.5)



the sun is shining. It is

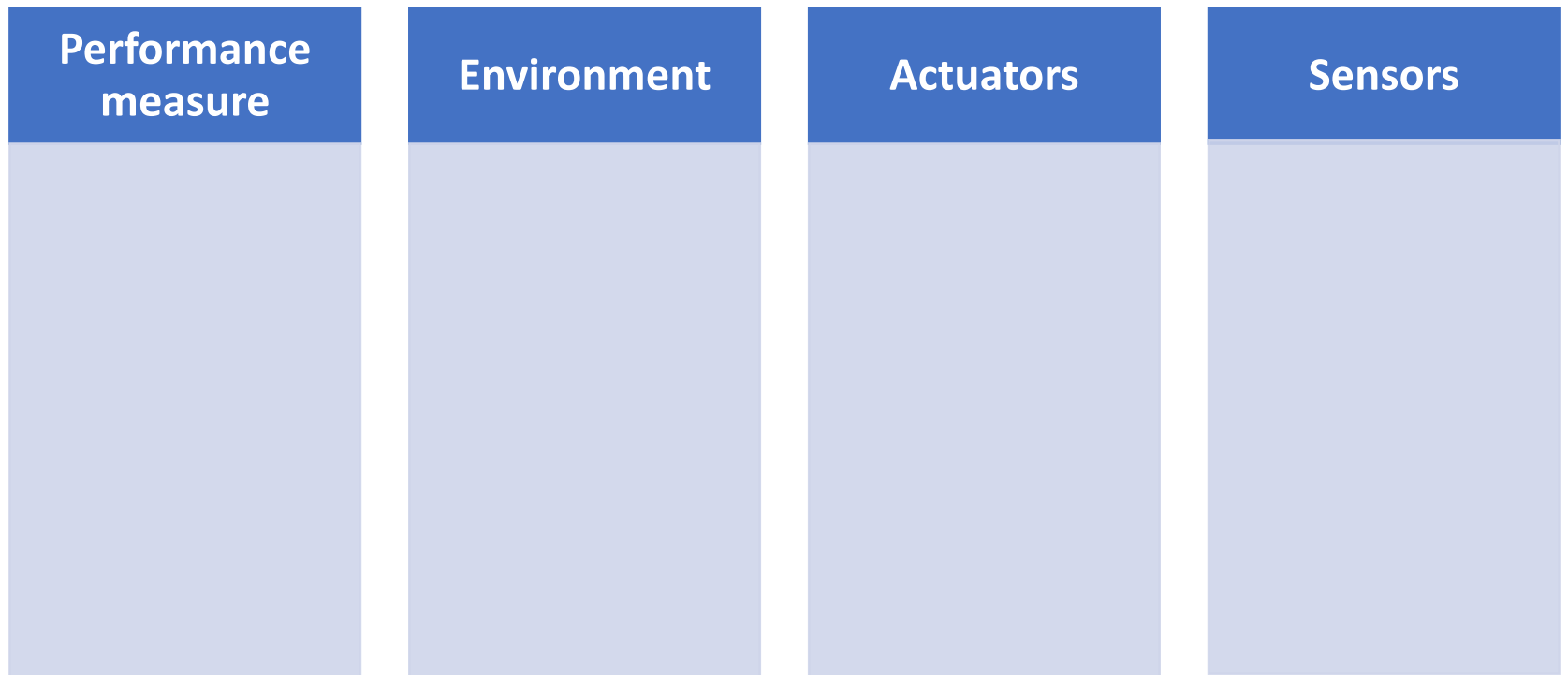


a beautiful day. The warm rays of sunlight illuminate the surroundings, creating a cheerful and inviting atmosphere. People are out and about, enjoying outdoor activities, and the clear blue sky adds to the overall sense of positivity. Whether you're taking a leisurely stroll, having a picnic, or simply basking in the sun's glow, it's a perfect time to appreciate the natural beauty and soak in the uplifting energy that the sunshine brings.

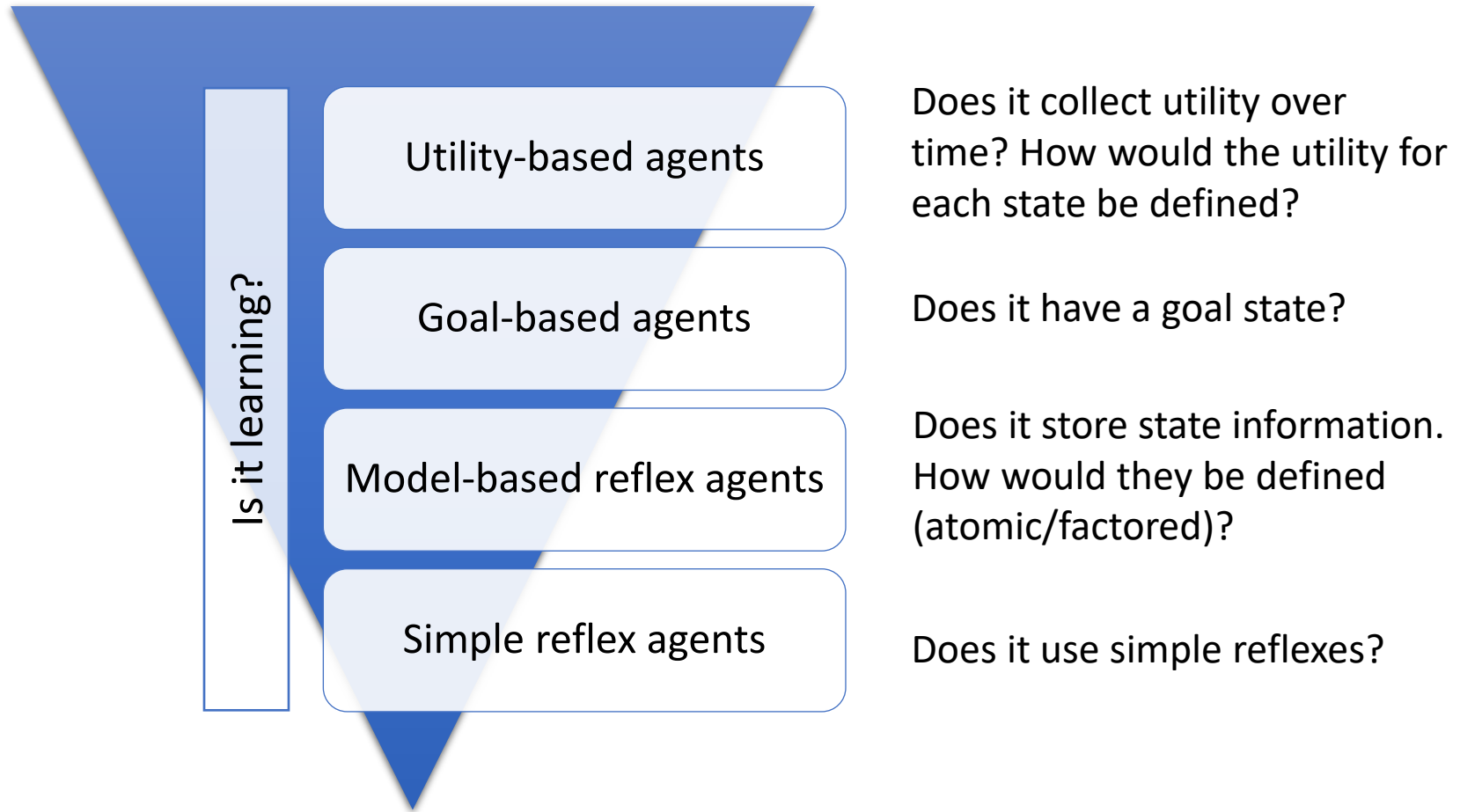




# PEAS Description of ChatGPT



# What Type of Intelligent Agent is ChatGPT?



- Does it pass the Touring test?
- Is it a rational agent?

# Intelligent Systems as Sets of Agents: Self-driving Car



It should learn!

Utility-based agents

Goal-based agents

Model-based reflex agents

Simple reflex agents

Make sure the passenger has a pleasant drive (not too much sudden breaking = utility)

Plan the route to the destination.

Remember where every other car is and calculate where they will be in the next few seconds.

React to unforeseen issues like a child running in front of the car quickly.

# Conclusion

Intelligent agents inspire the research areas of modern AI

**Search** for a goal  
(e.g., navigation).

**Optimize** functions  
(e.g., utility).

Stay within given  
**constraints**

(constraint satisfaction problem;  
e.g., reach the goal without  
running out of power)

Deal with **uncertainty**  
(e.g., current traffic on the  
road).

**Learn** a good agent  
program from data  
and improve over time  
(machine learning).

**Sensing**  
(e.g., natural language  
processing, vision)