



# 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/).

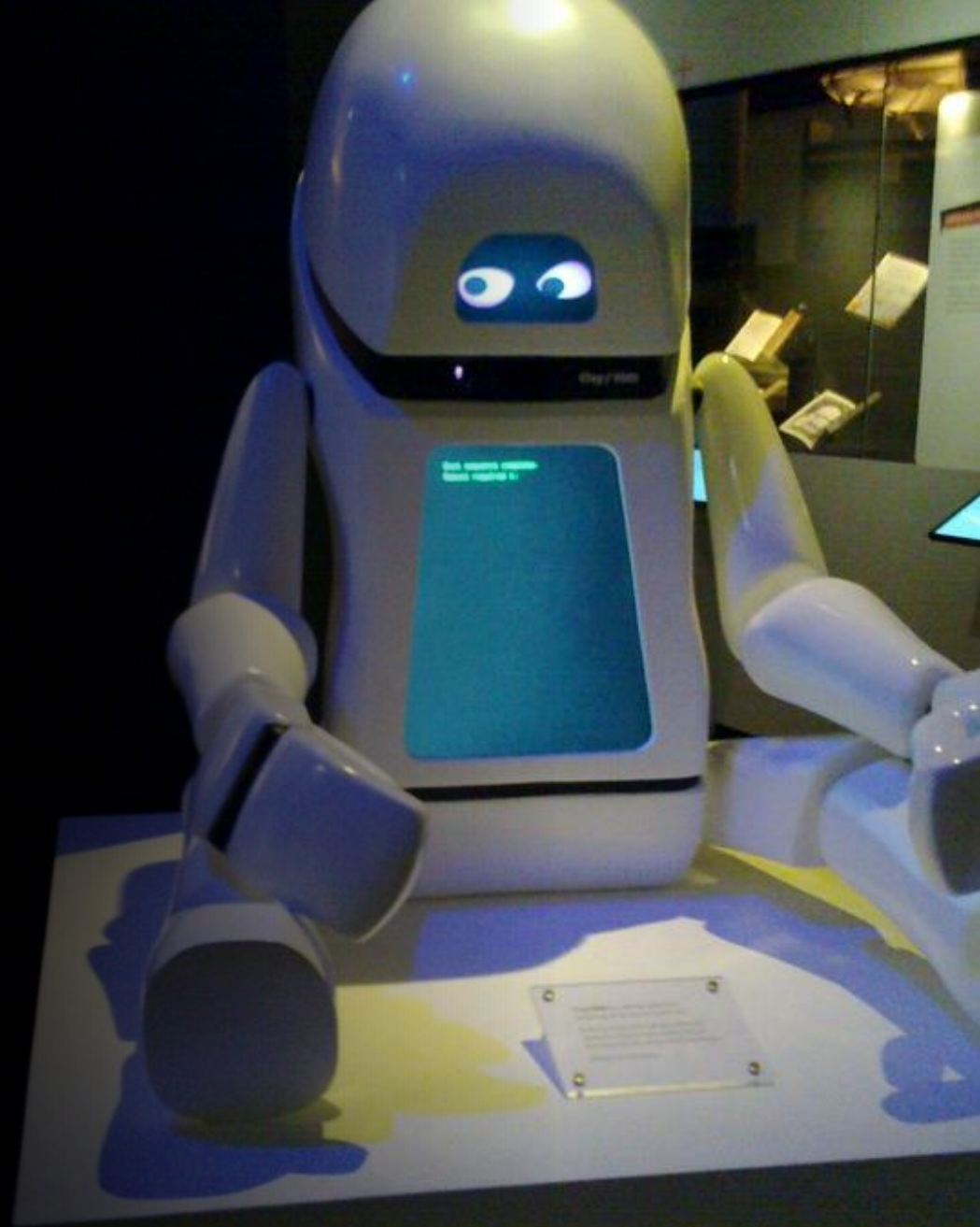
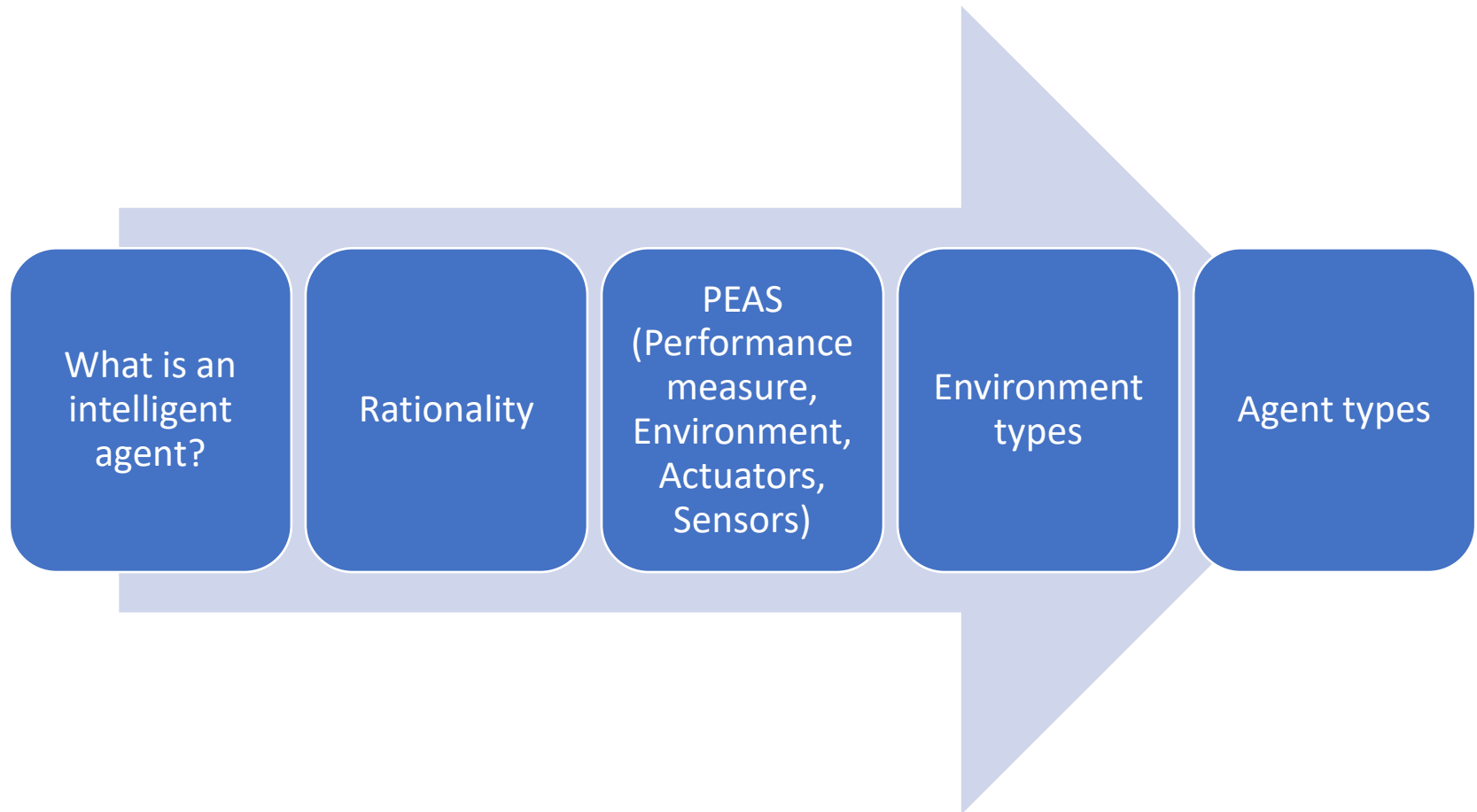


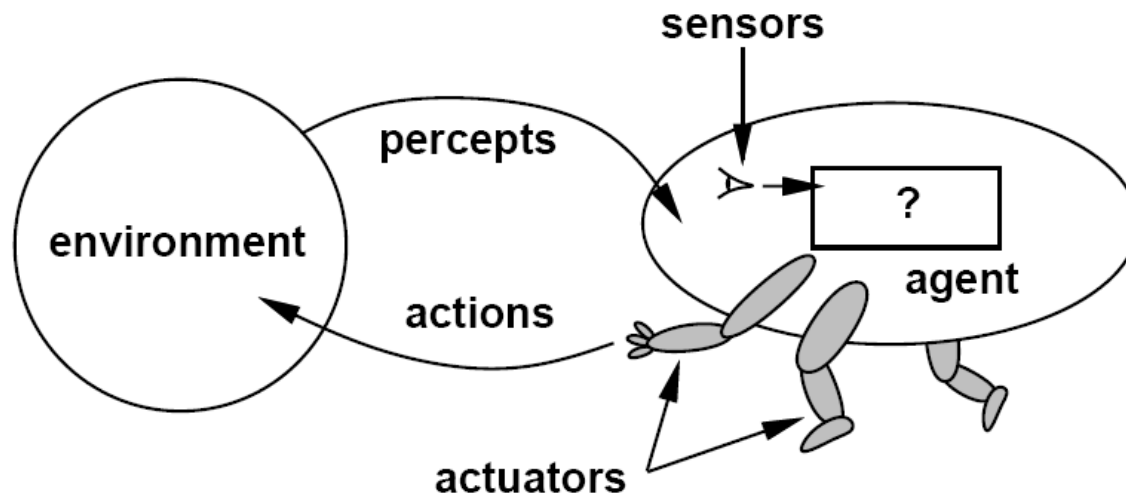
Image: "Robot at the British Library Science Fiction Exhibition"  
by BadgerGravling

# Outline



# 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 regulates a process variable to a desired state or set point without human interaction. The agent is called controller.
- **Softbot**: Agent is a software program that runs on a host device.

# Agent function and agent program

The **agent function** maps from *percept sequences*  $P^*$  to *actions*  $A$  formulated as an abstract mathematical function  $f : P^* \rightarrow A$  (e.g., a table).

The **agent program** is the concrete implementation running in a physical system.

Agent = architecture (hardware) + agent program



- Sensors
- Memory
- Computational power

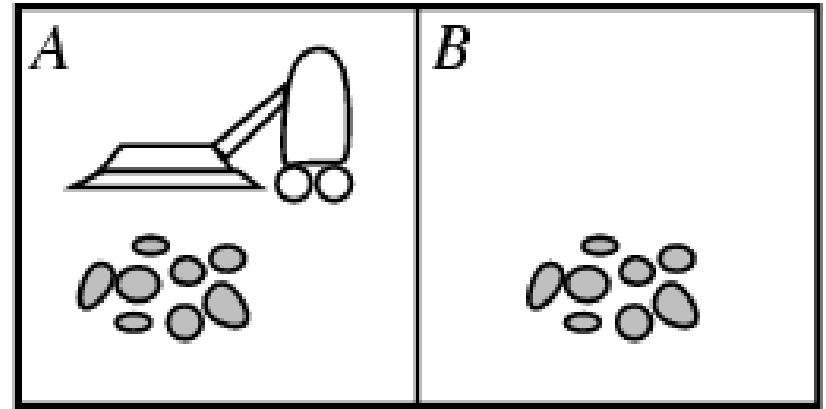
# Example: Vacuum-cleaner world

- **Percepts:**

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

- **Actions:**

Left, Right, Suck, NoOp



Last Percept

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**

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

This table can become infinitively large!

# 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).
- 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)

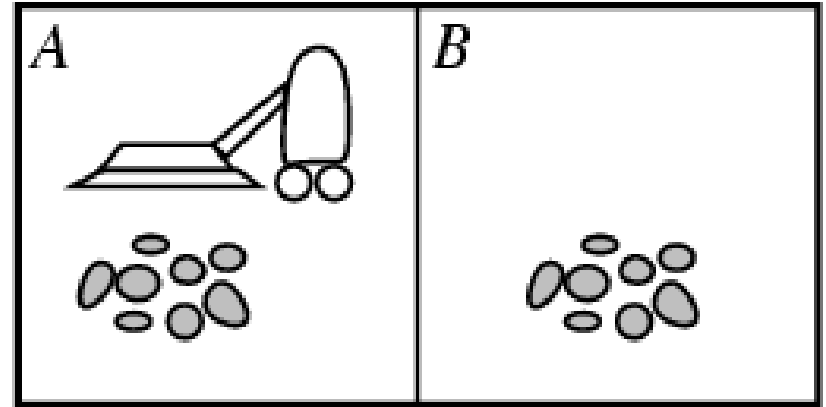
# Example: Vacuum-cleaner world

- **Percepts:**

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

- **Actions:**

Left, Right, Suck, NoOp



Agent function:

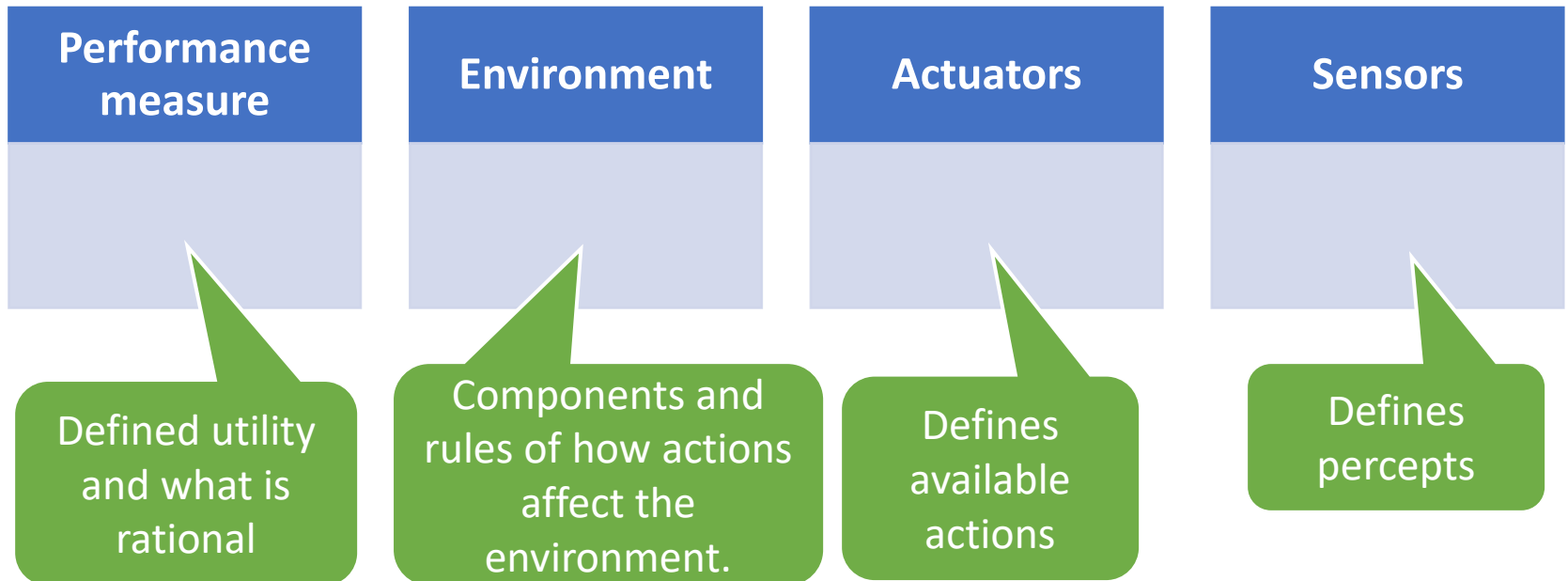
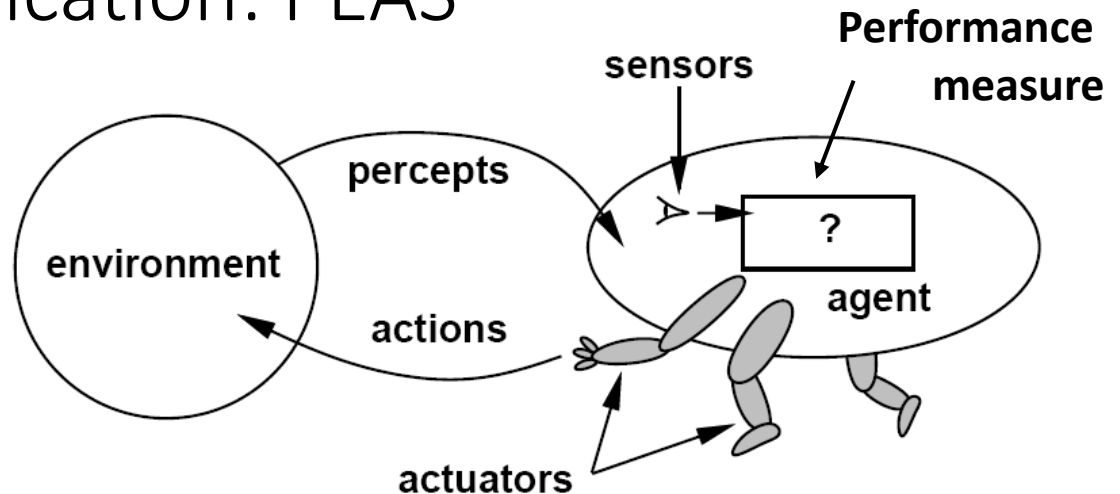
Percept Sequence	Action
[A, Clean]	Right
[A, Dirty]	Suck
...	
[A, Clean], [B, Clean]	Left
...	

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 rational?**

# 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>• 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>

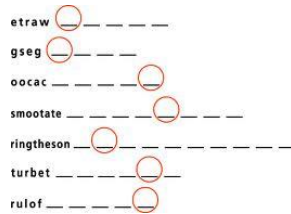
# Environment types

- **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 is deterministic except for the actions of other agents are stochastic but follow a strategy.
- **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 unchanged while an agent is deliberating.
  - **Semidynamic:** the environment does not change with the passage of time, but the agent's performance score does.
- **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

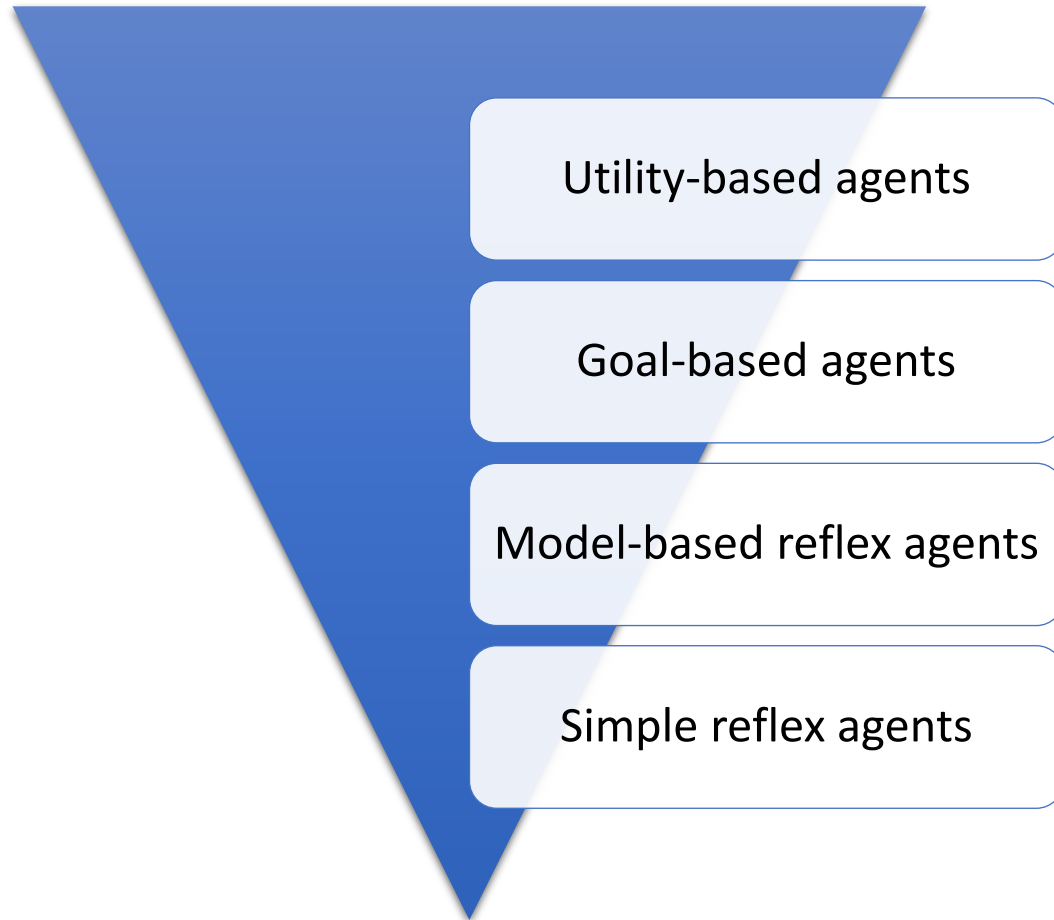


Taxi driving

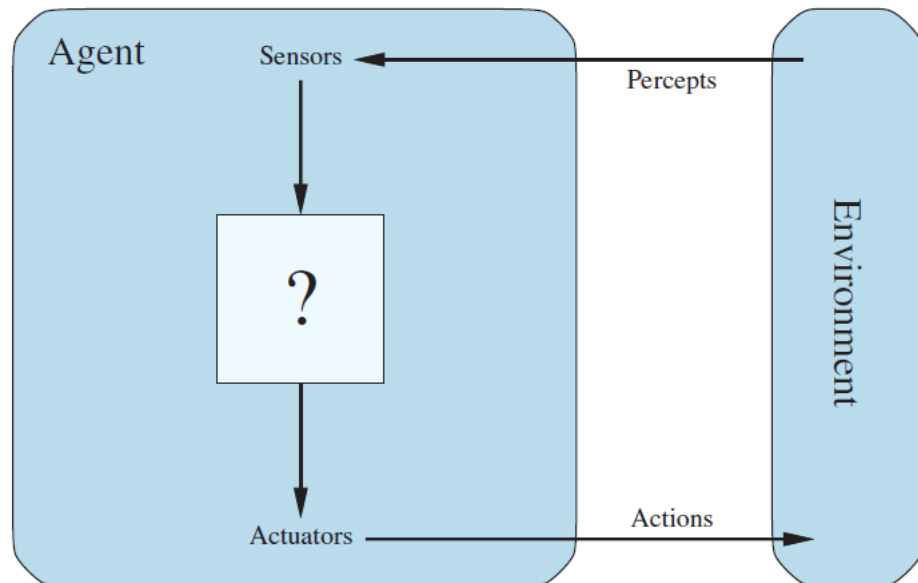
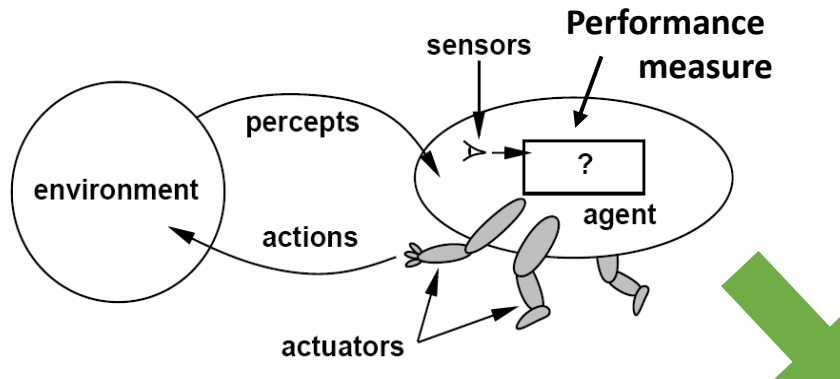
Observable	Fully	Fully	Partially	Partially
Deterministic	Deterministic	Strategic	Stochastic +Strategic	Stochastic
Episodic	Episodic	Sequential	Sequential	Sequential
Static	Static	Semidynamic	Static	Dynamic
Discrete	Discrete	Discrete	Discrete	Continuous
Single agent	Single	Multi	Multi	Multi

Actions have long-term effects.

# Hierarchy of agent types

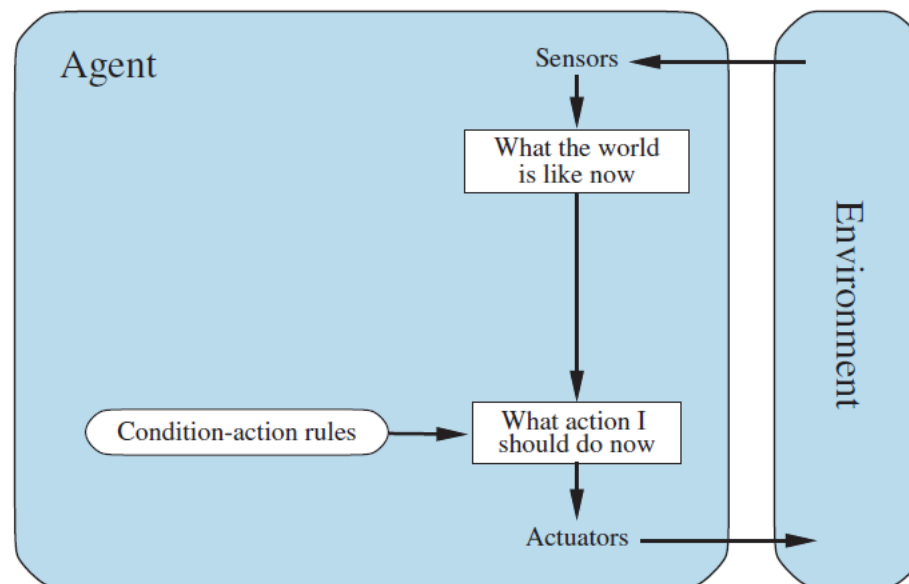


# Structure of an Agent



# Simple reflex agent

Rules select action only **based on current percept**, ignoring all past percepts (no memory). This is typically very fast!

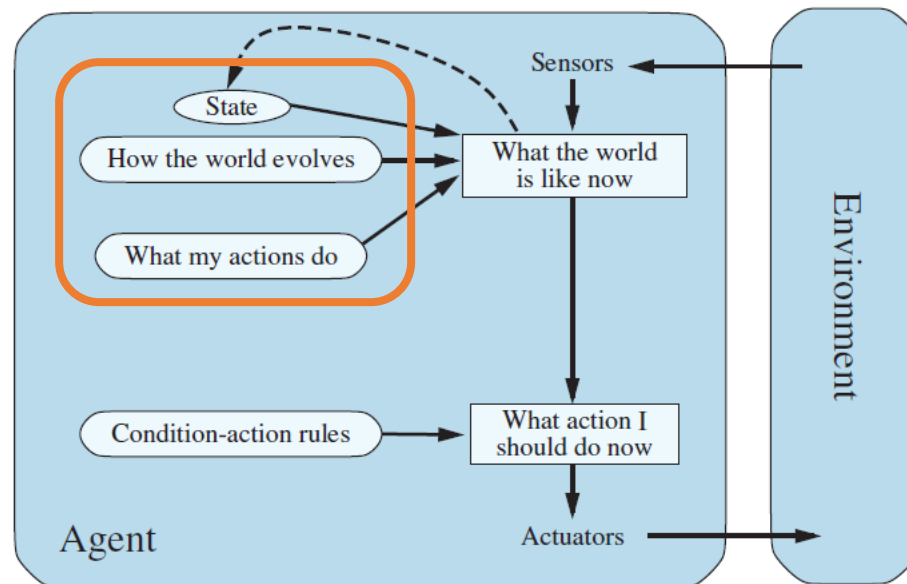


**Example:** simple, rule-based vacuum cleaner from before.



# Model-based reflex agent

Maintains **internal state** (memory) to keep track of aspects of the environment that cannot be currently observed. There is now more information for the rules to make decisions.

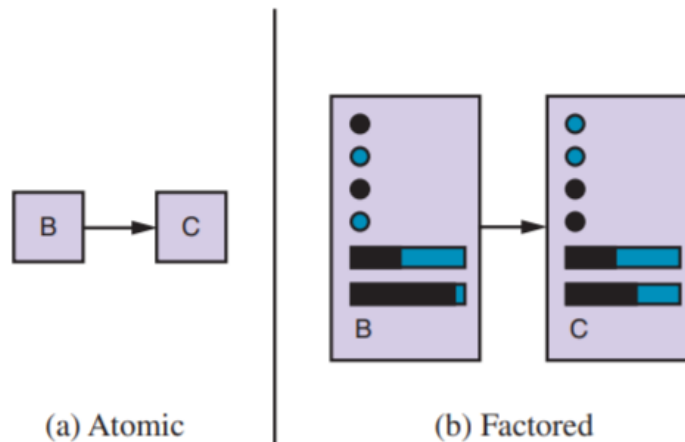


**Example:** simple, vacuum cleaner that remembers where it has already cleaned.

# 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.

# Old-school vs. Smart thermostat



Old-school thermostat

**Percepts**

**States**



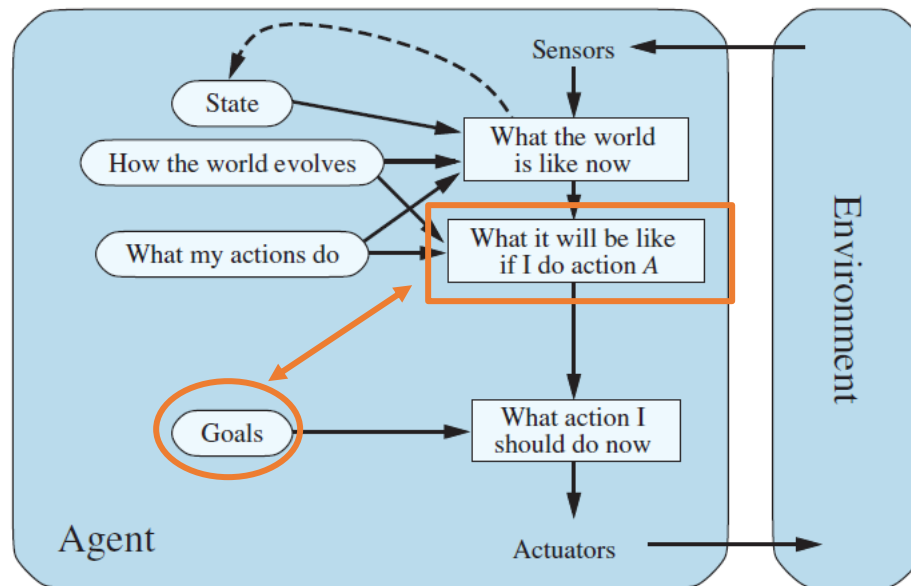
Smart thermostat

**Percepts**

**States**

# Goal-based agent

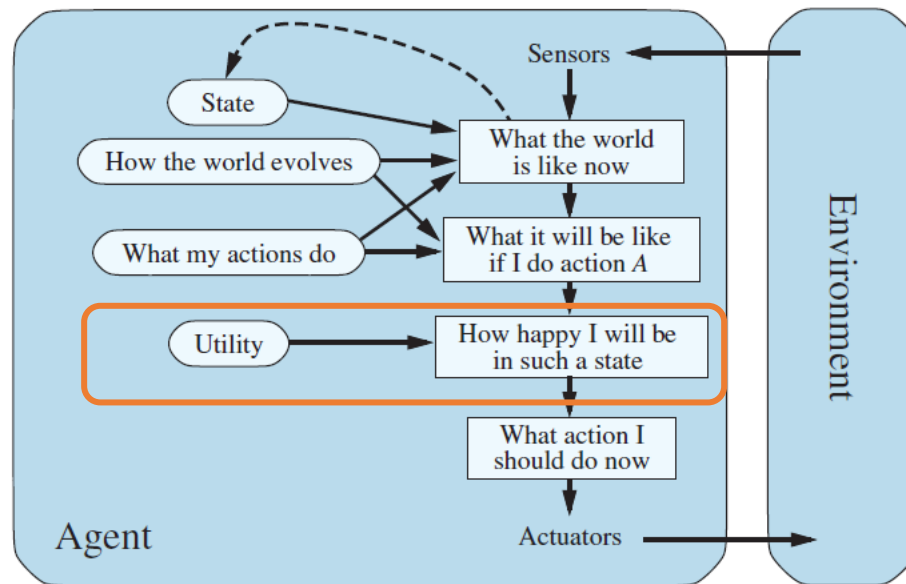
The agent chooses actions in the current state to reach a **goal state** as fast as possible. We need **search algorithms** to find action sequences that reach that goal.



**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 different states**. Choose the action to maximize expected utility over time (i.e., stay in desirable states).

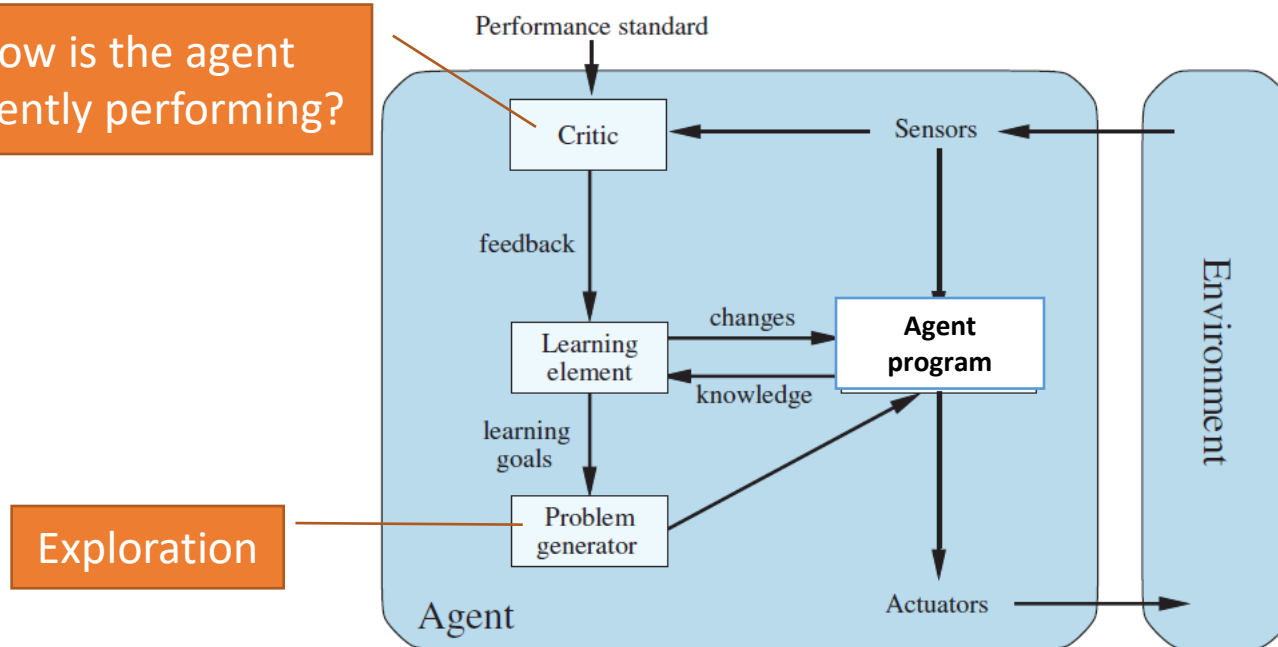


**Example:** An autonomous Mars rover prefers states where its battery is not critically low.

# Agents that Learn

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

How is the agent currently performing?



# 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

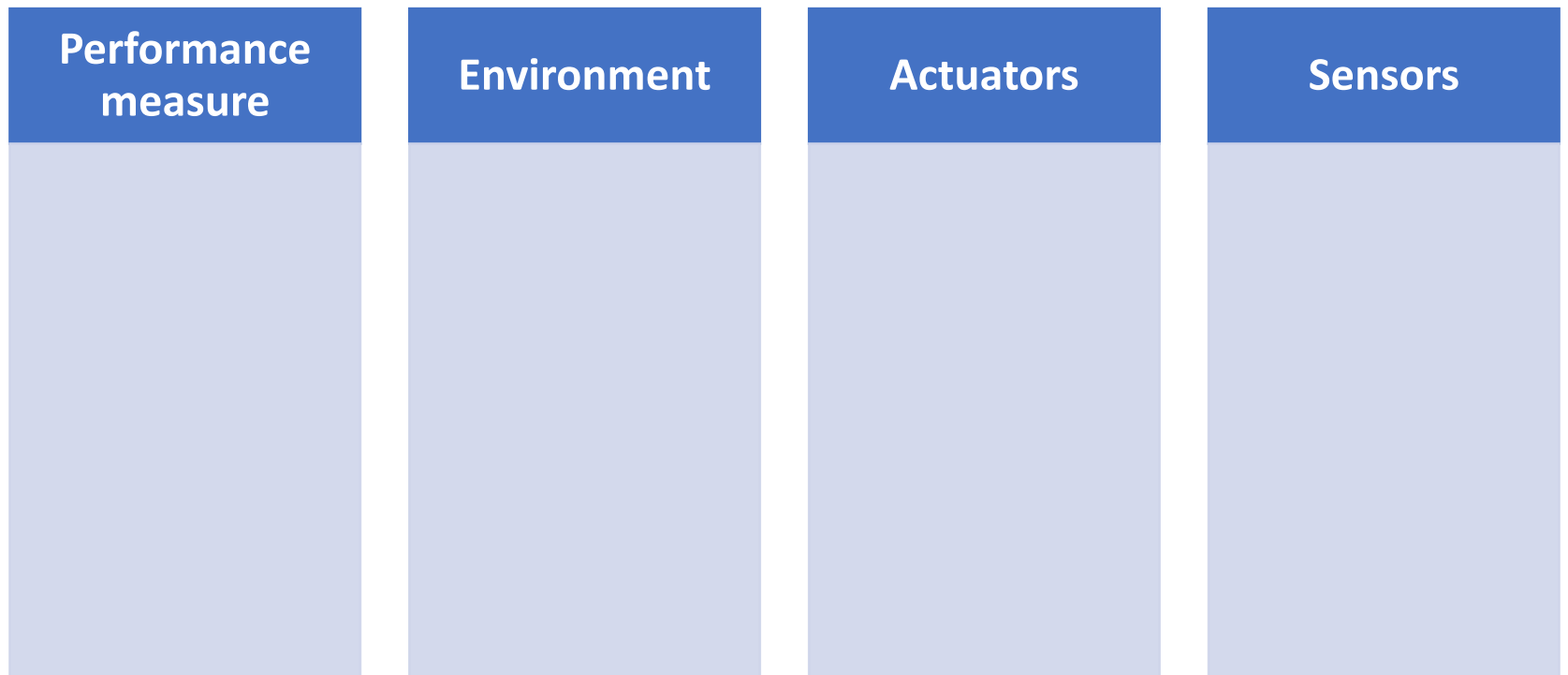


iRobot's Roomba brand has become as synonymous with robot vacuum as Q-tips is with cotton swabs. The Wi-Fi-enabled Roomba 960 is ample evidence why. It turns a tiresome chore into something you can almost look forward to. With three cleaning modes and dirt-detecting sensors, it kept all the floor surfaces 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.

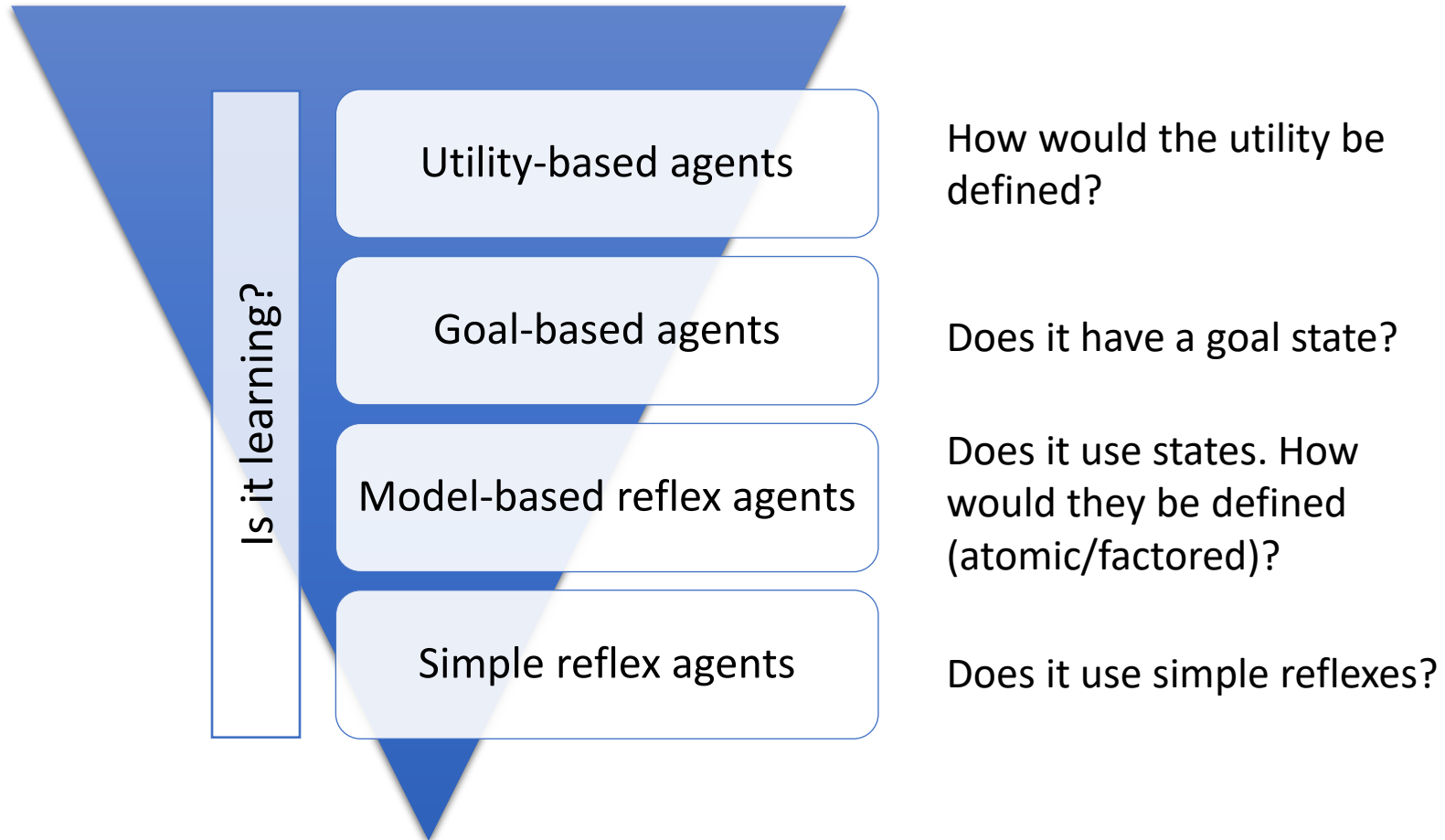
Source: <https://www.techhive.com/article/3269782/best-robot-vacuum-cleaners.html>



# PEAS Description of a modern robot vacuum



# What type of intelligent agent is a modern robot vacuum?



# 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)