



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/).

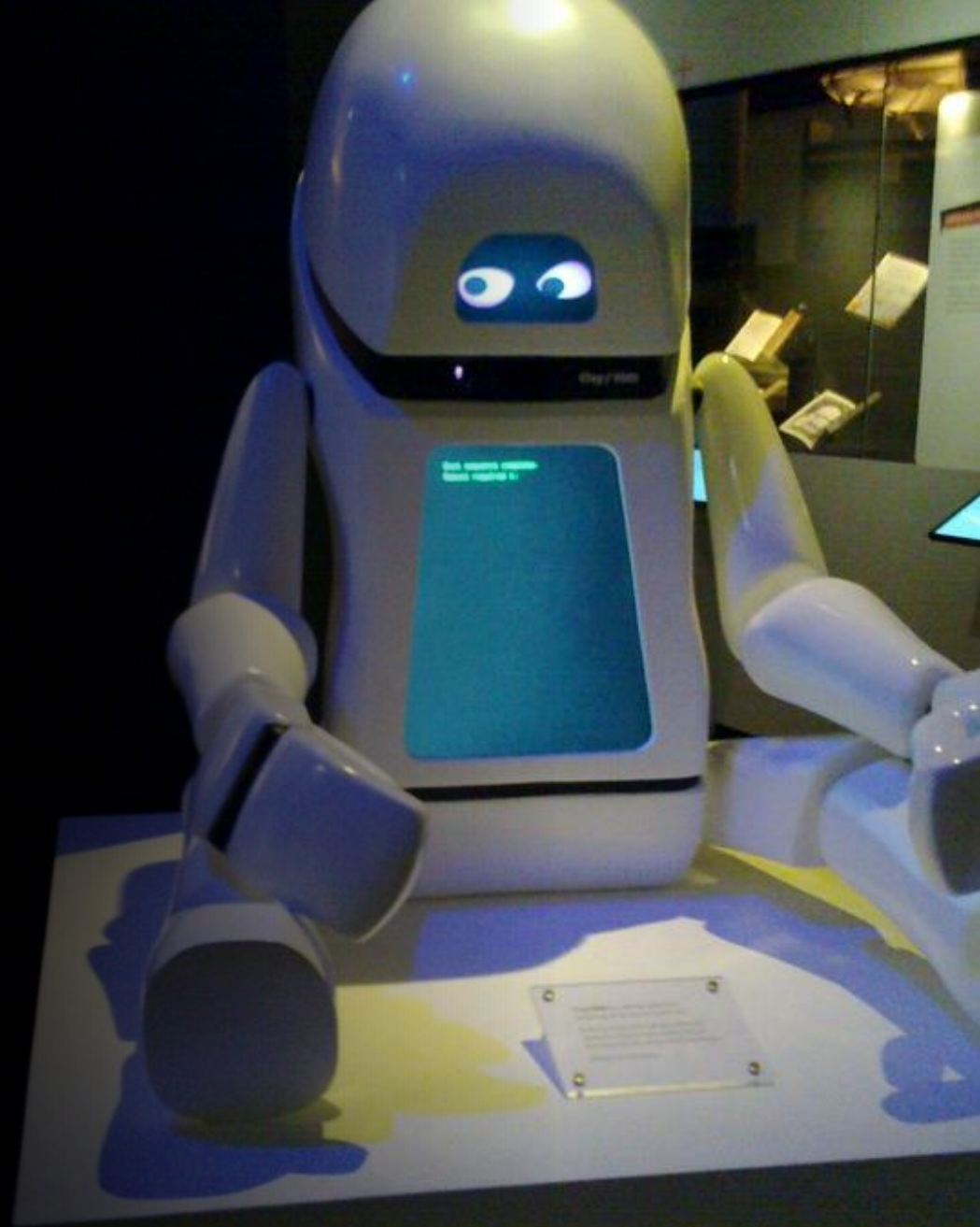
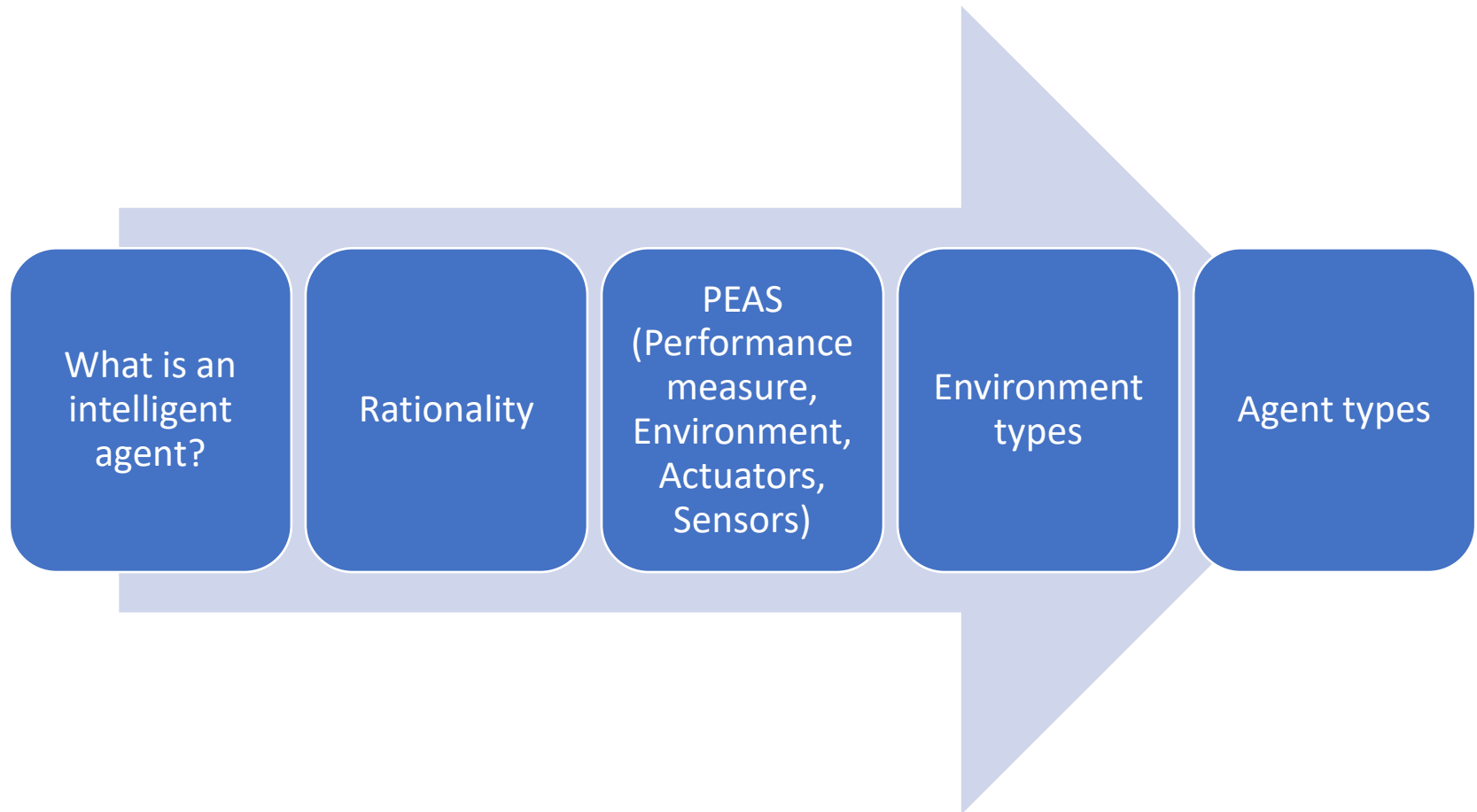
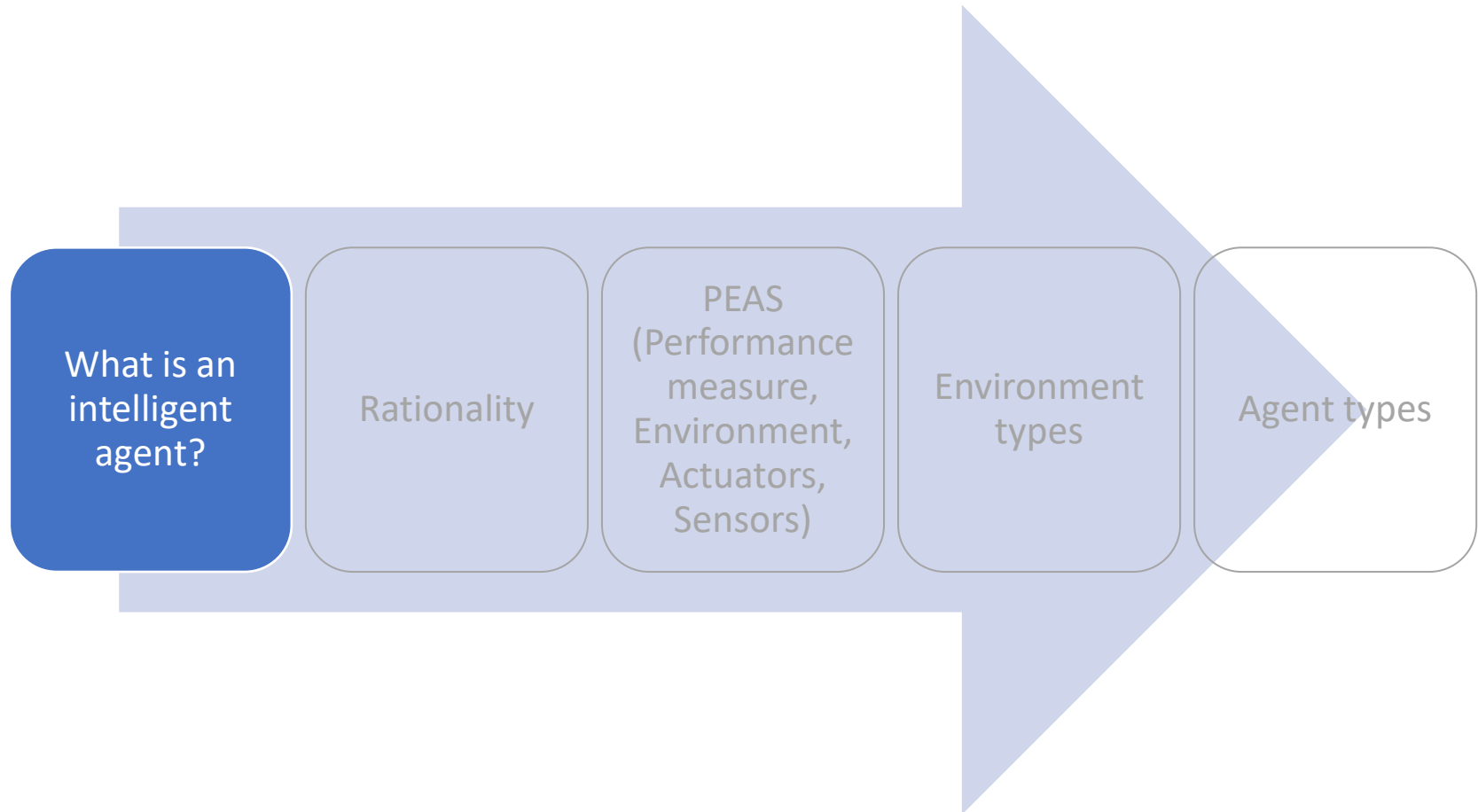


Image: "Robot at the British Library Science Fiction Exhibition"
by BadgerGravling

Outline

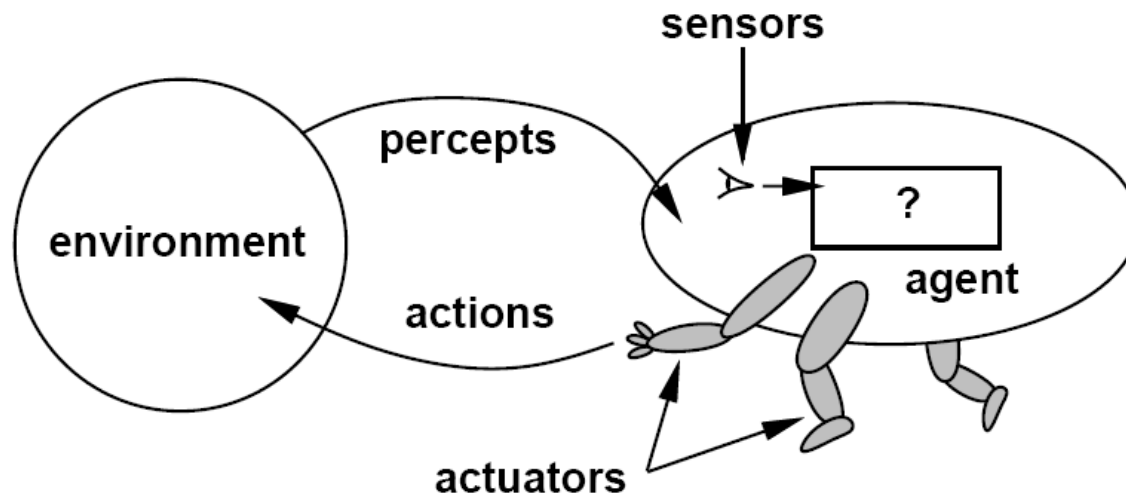


Outline



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 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 the set of *percept sequences* P^* to the *set of actions* A formulated as an abstract mathematical function $f : P^* \rightarrow A$.

The **agent program** is the concrete implementation for a given physical system.

Agent = architecture (hardware) + agent program



- Sensors
- Memory
- Computational power

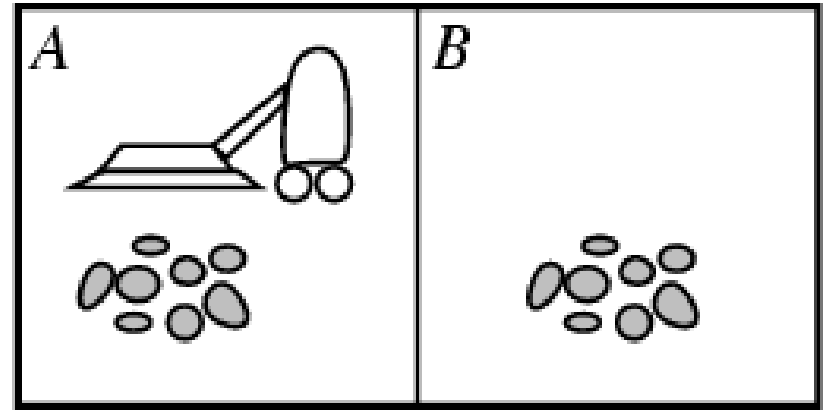
Example: Vacuum-cleaner World

- **Percepts:**

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

- **Actions:**

Left, Right, Suck, NoOp



Last Percept

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

<u>Percept Sequence</u>	<u>Action</u>
[A, Clean]	Right
[A, Dirty]	Suck
...	
[A, Clean], [B, Clean]	Left
...	
[A, Clean], [B, Clean], [A, Dirty]	Suck
...	

Implemented agent program:

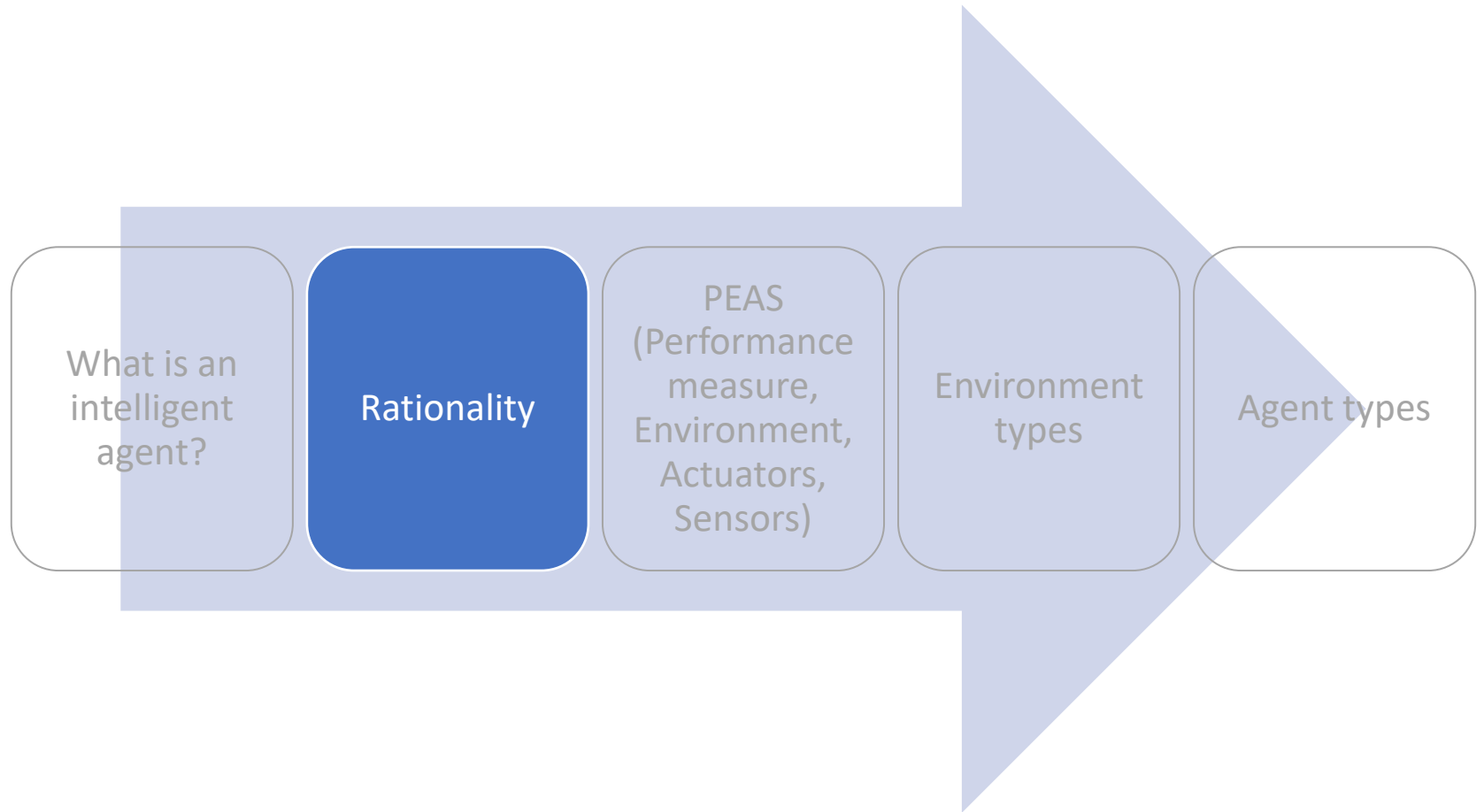
function Vacuum-Agent([location, status])

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!

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).
- 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)

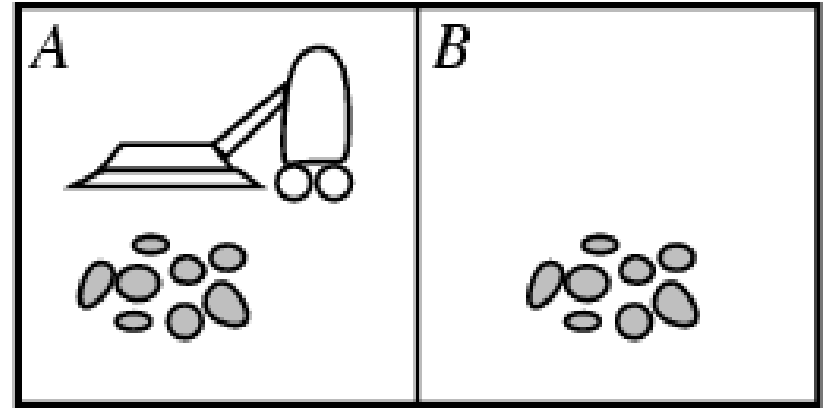
Example: Vacuum-cleaner World

- **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:

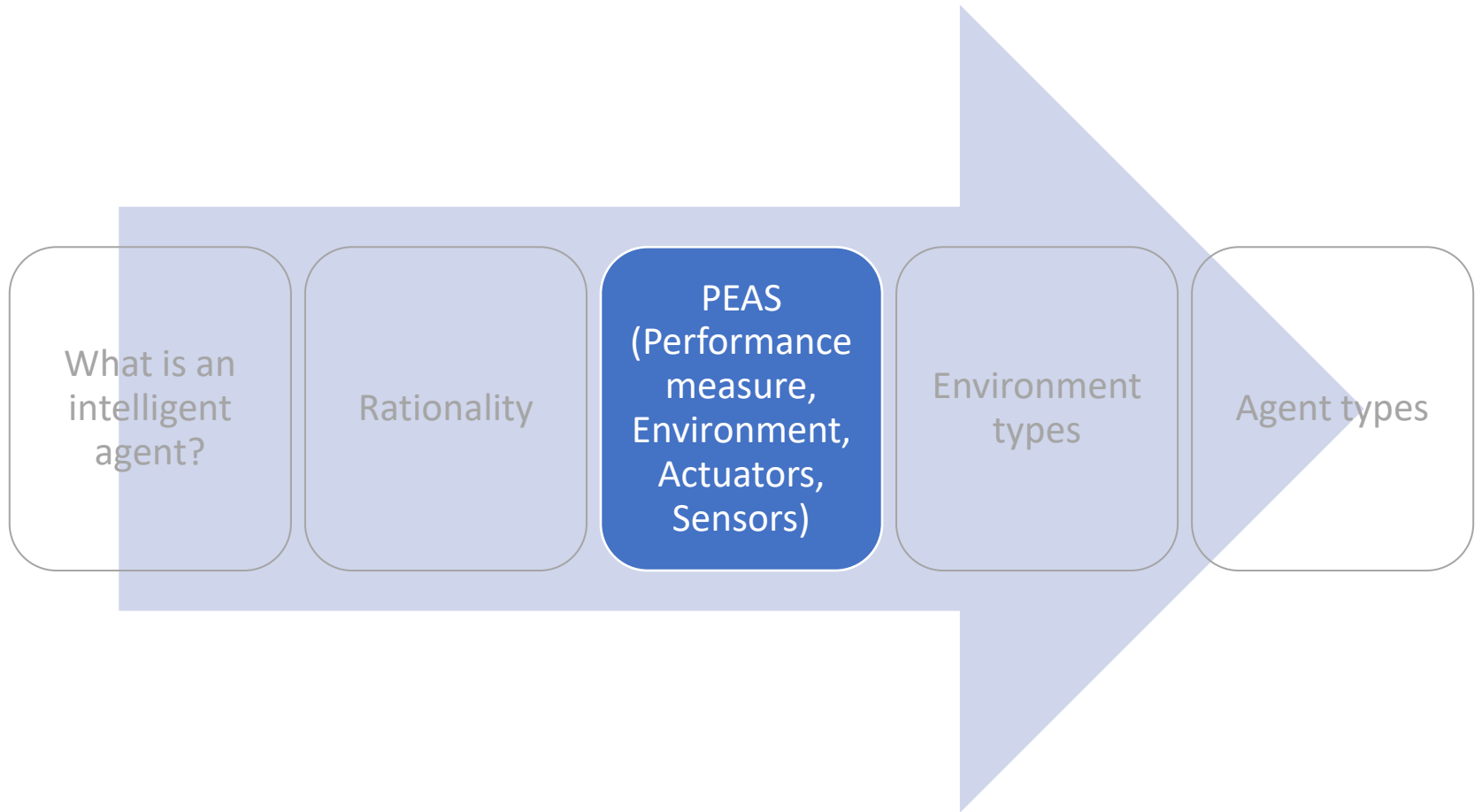
```
function Vacuum-Agent([location, status])
```

```
    returns an action
```

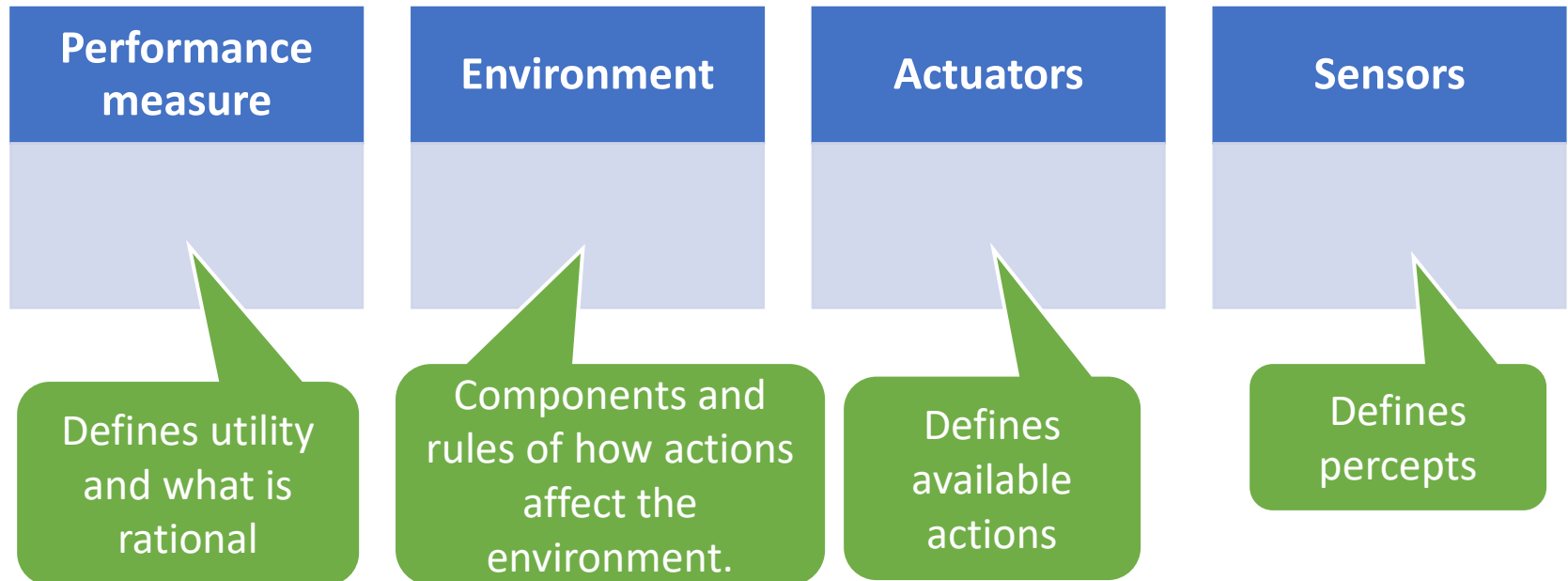
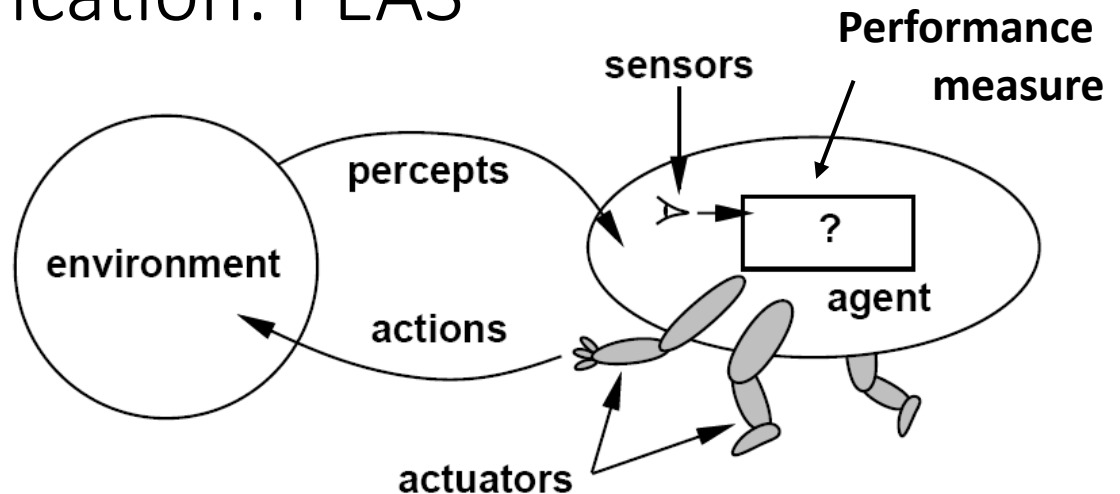
```
    if status = Dirty then return Suck
    else if location = A then return Right
    else if location = B then return Left
```

What could be a performance measure?
Is this agent rational?

Outline



Problem Specification: PEAS



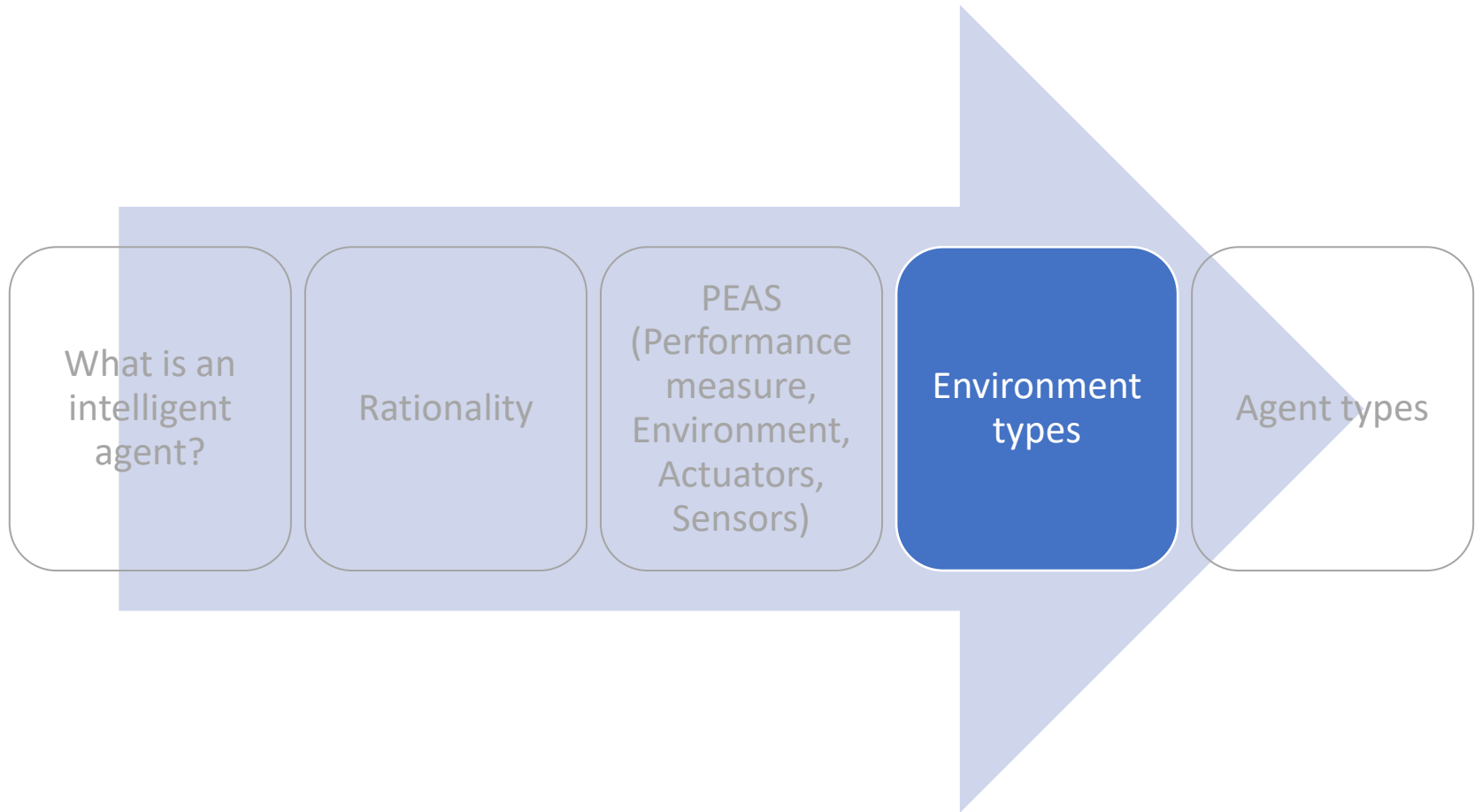
Example: Automated Taxi Driver

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

Example: Spam Filter

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

Outline



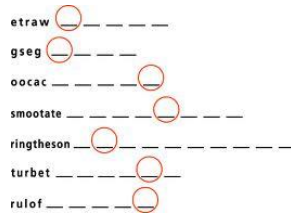
Environment Types

- **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

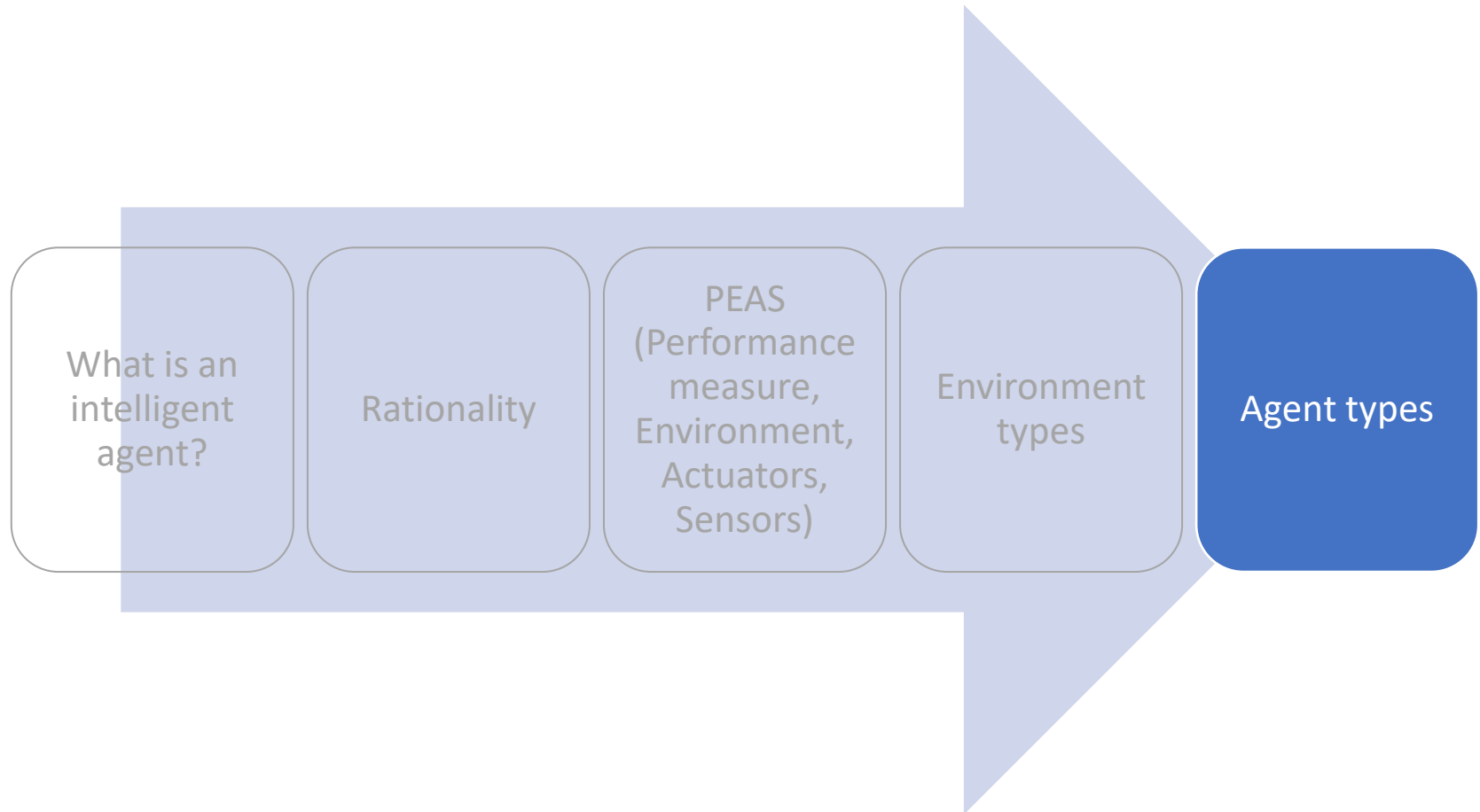
Multi

Multi

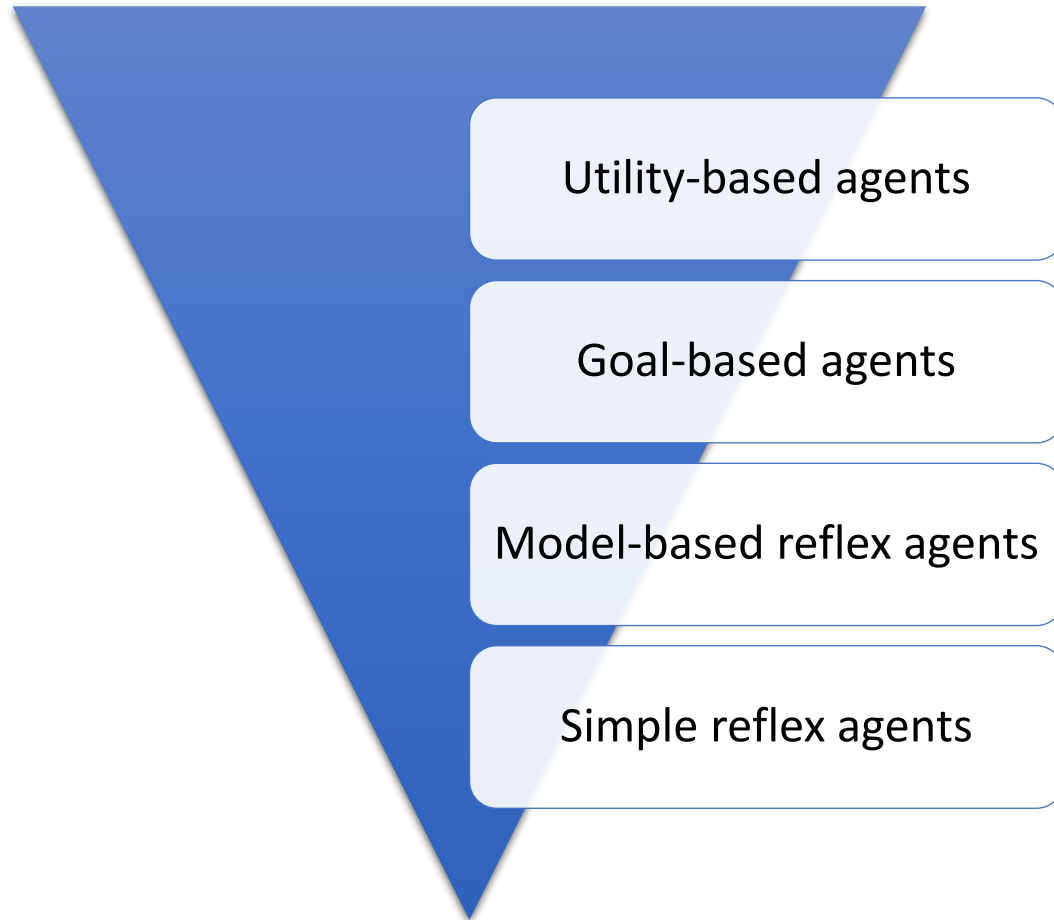
Multi

Actions have long-term effects.

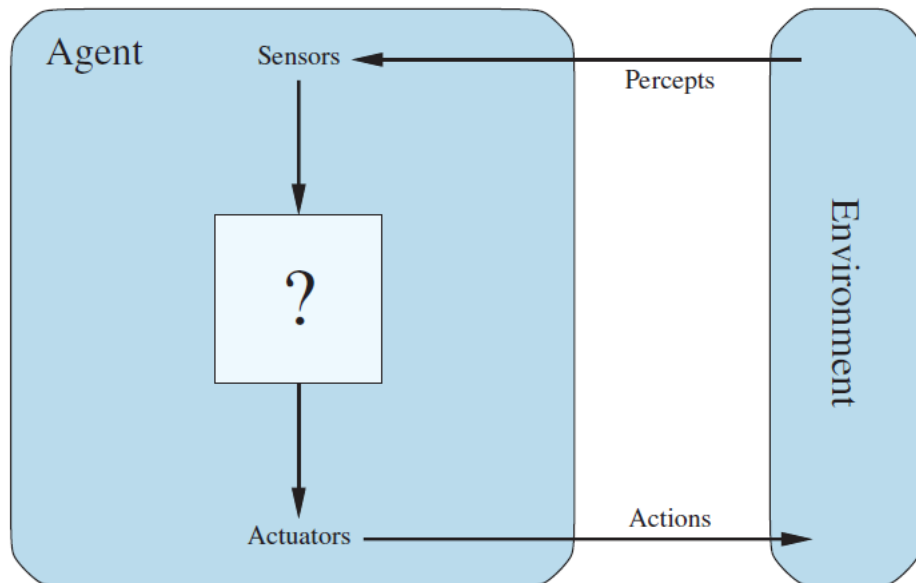
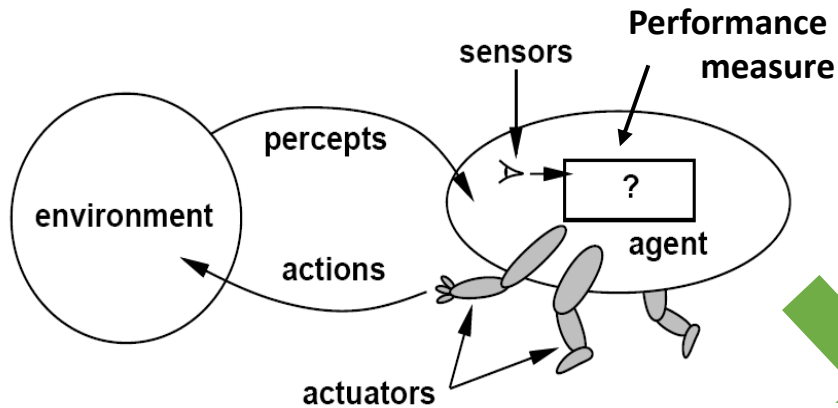
Outline



Hierarchy of Agent Types

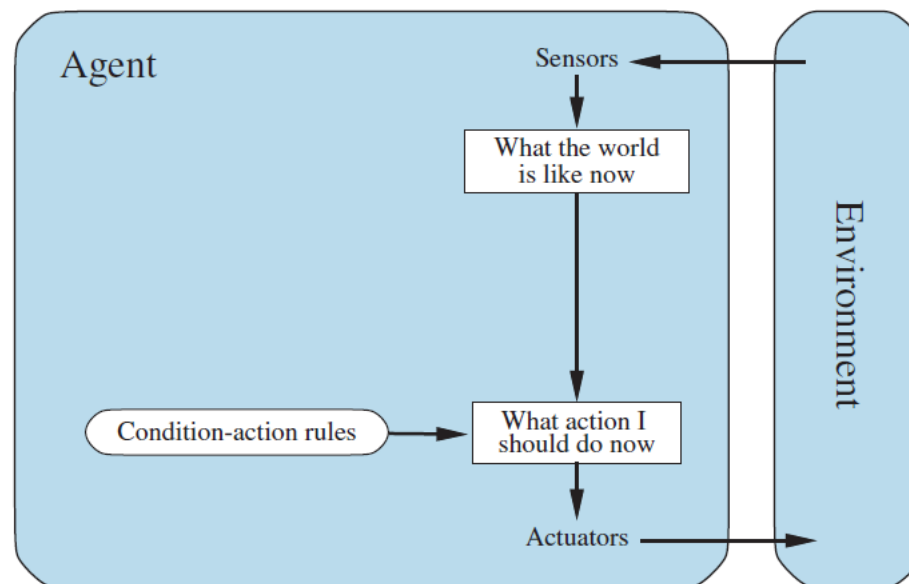


Structure of an Agent



Simple Reflex Agent

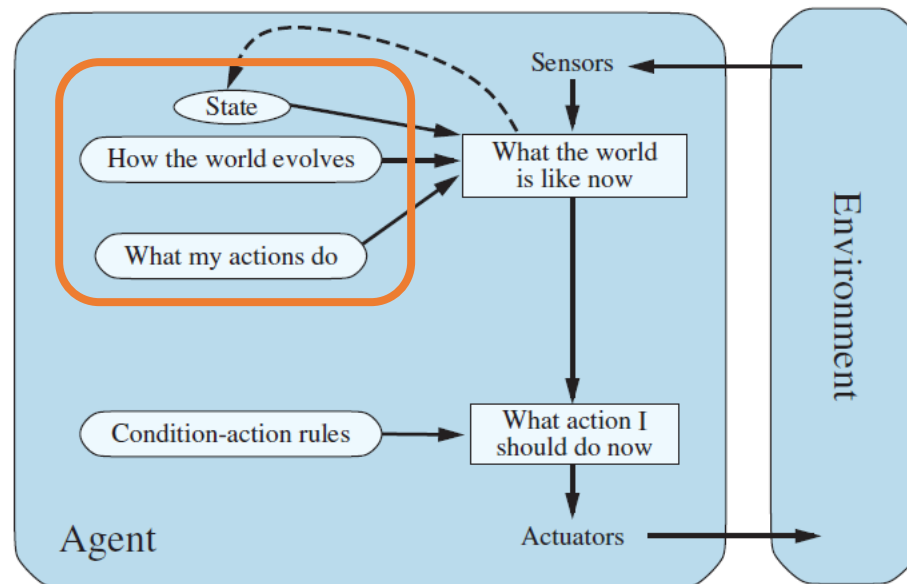
Rules select action only **based on current percept**, ignoring all past percepts (no memory). This is typically very fast!



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.

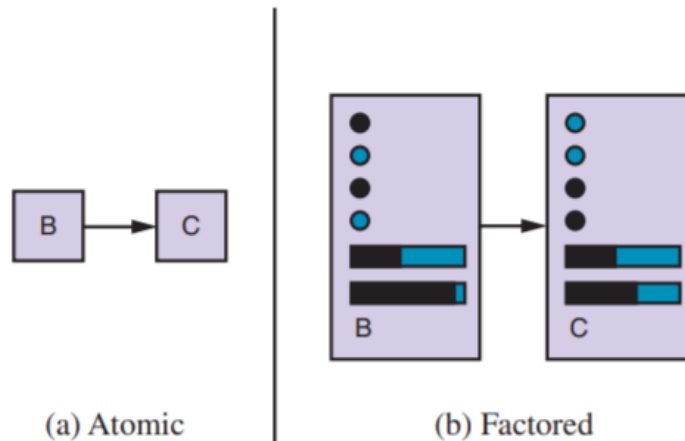


Example: simple, vacuum cleaner that remembers where it has already cleaned.

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



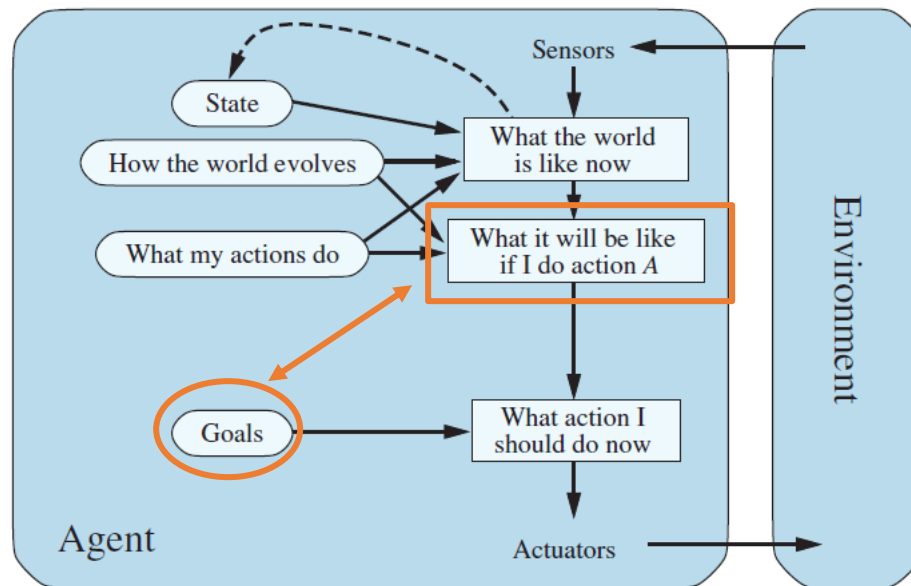
Smart thermostat

Percepts

States

Goal-based Agent

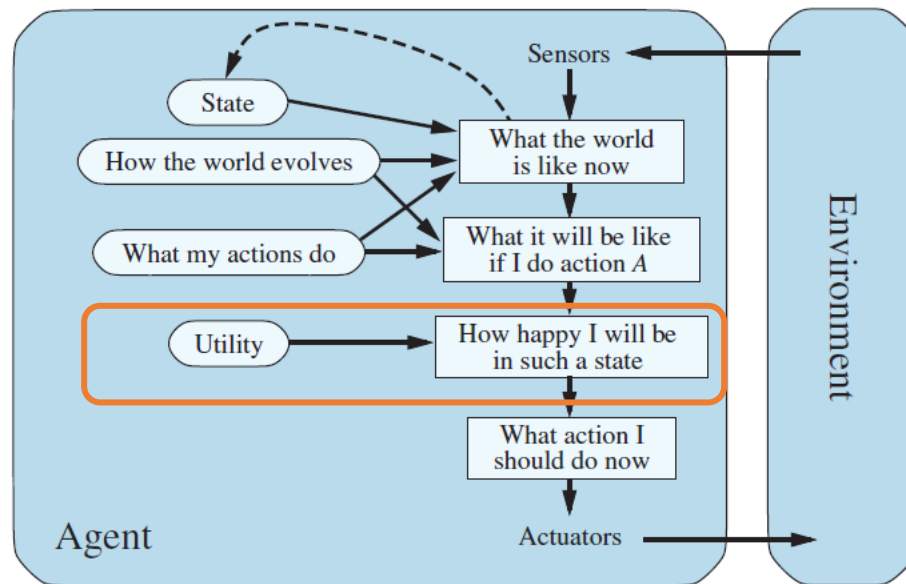
The agent chooses actions in the current state to reach a **goal state** as fast as possible. We need **search algorithms** to find action sequences that reach that goal.



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).

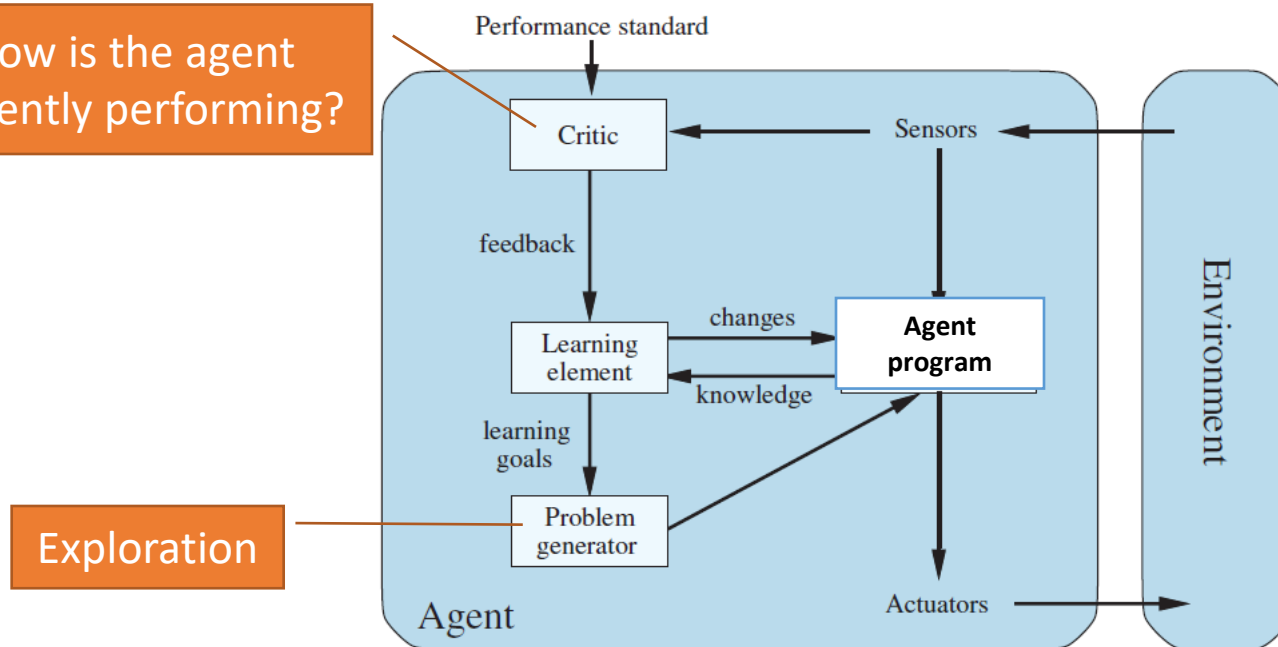


Example: An autonomous Mars rover prefers states where its battery is not critically low.

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



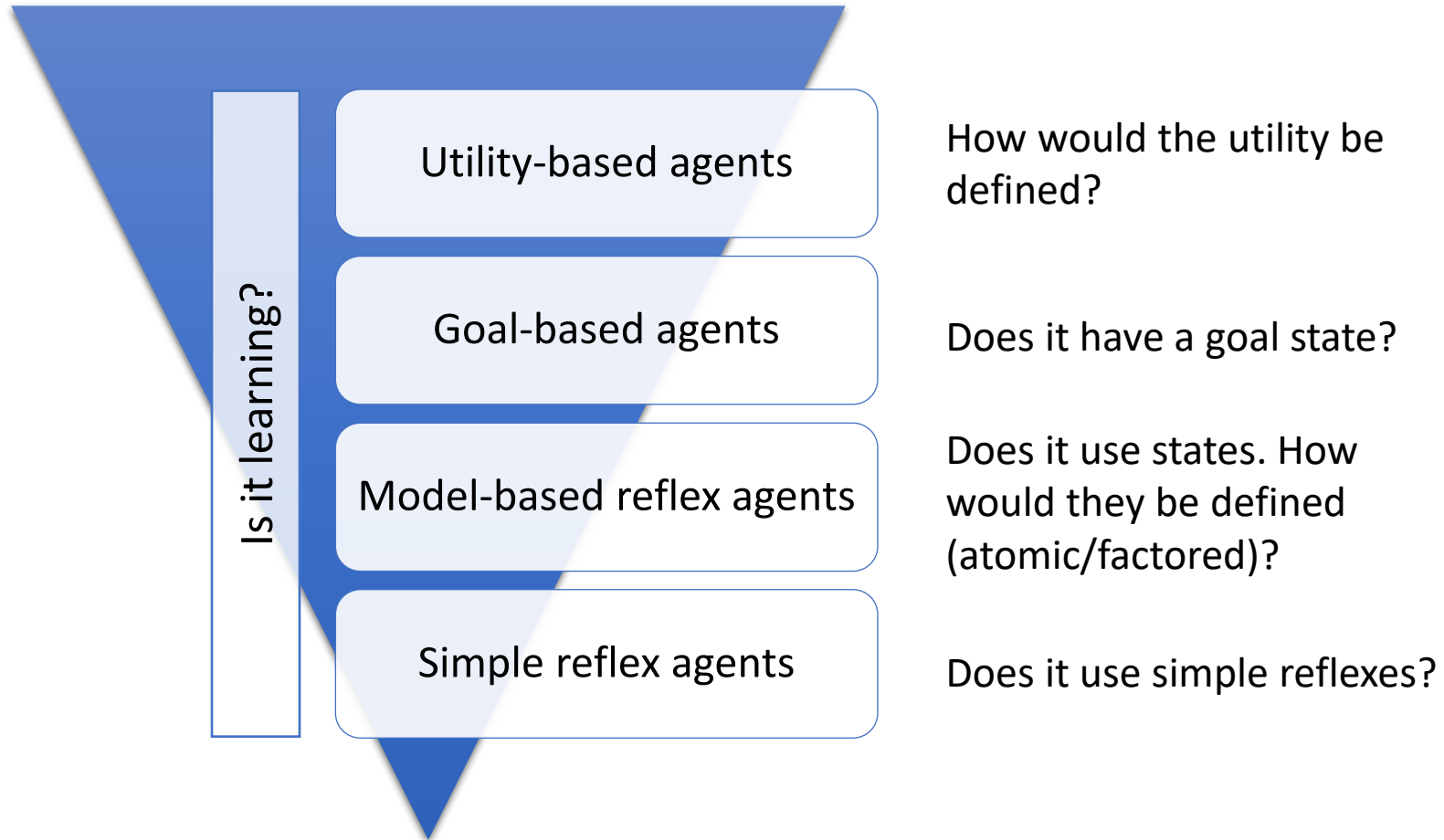
iRobot's Roomba brand has become as synonymous with robot vacuum as Q-tips is with cotton swabs. The Wi-Fi-enabled Roomba 960 is ample evidence why. It turns a tiresome chore into something you can almost look forward to. With three cleaning modes and dirt-detecting sensors, it kept all the floor surfaces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

Performance measure	Environment	Actuators	Sensors

What Type of Intelligent Agent is it?



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)