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.

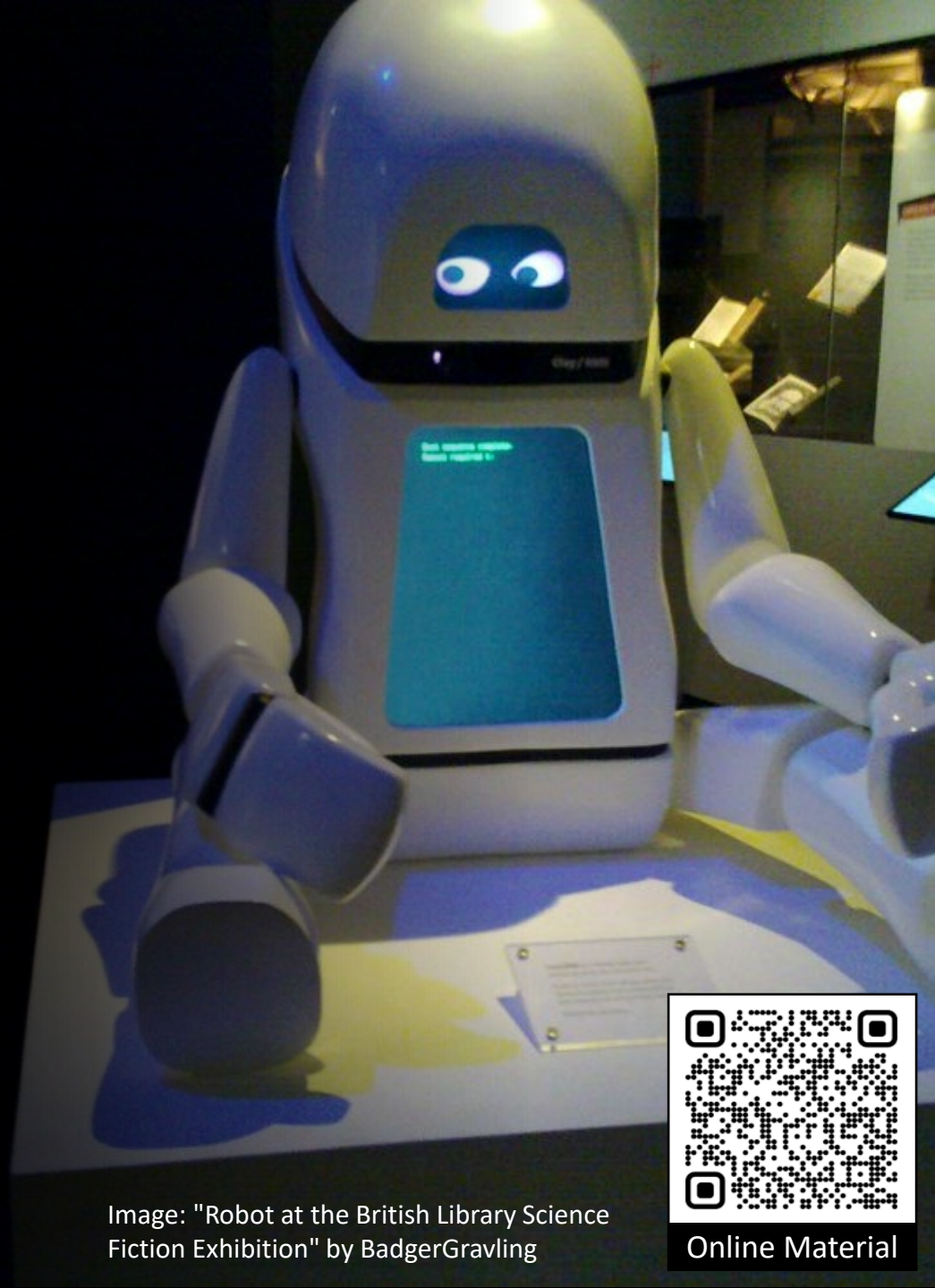
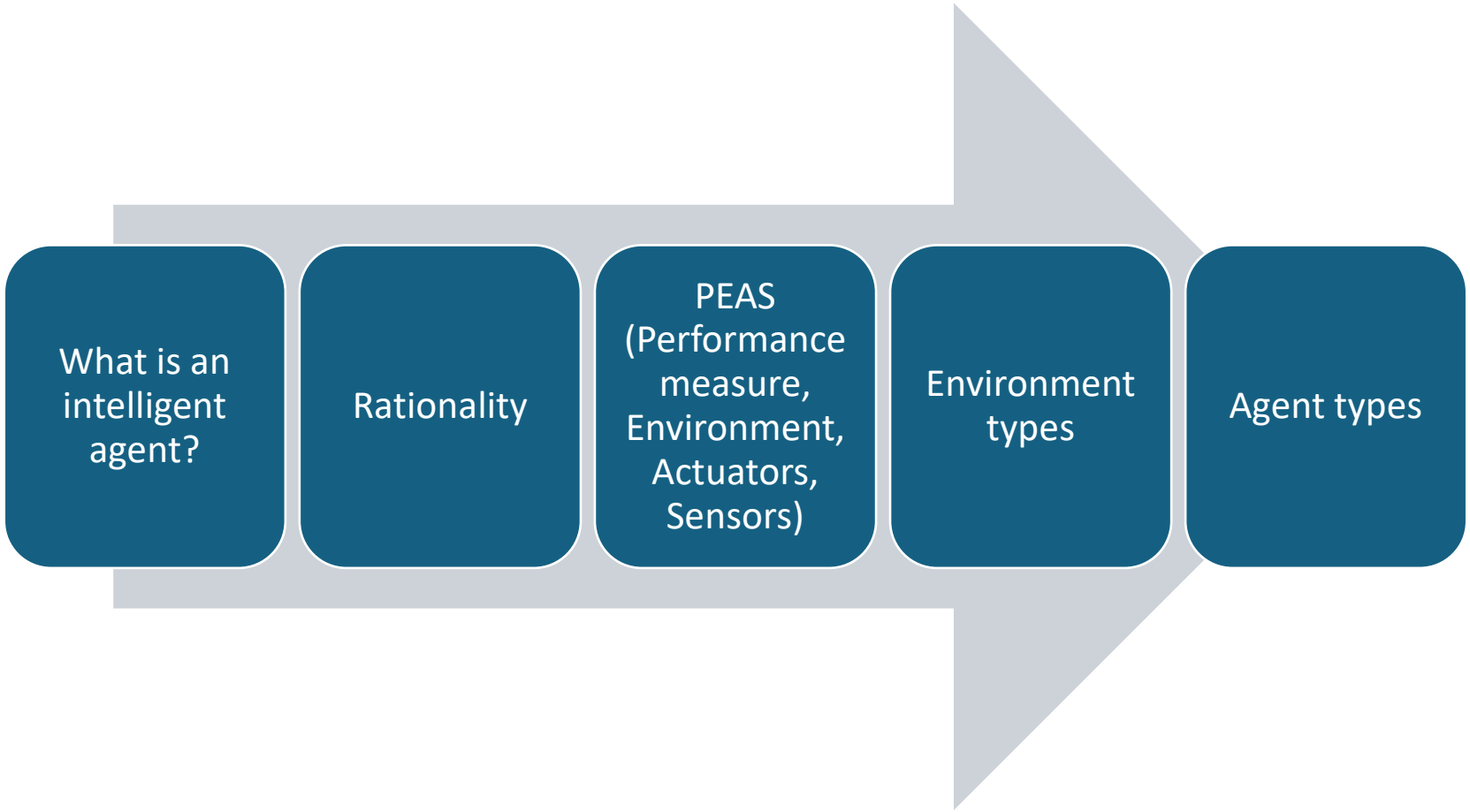


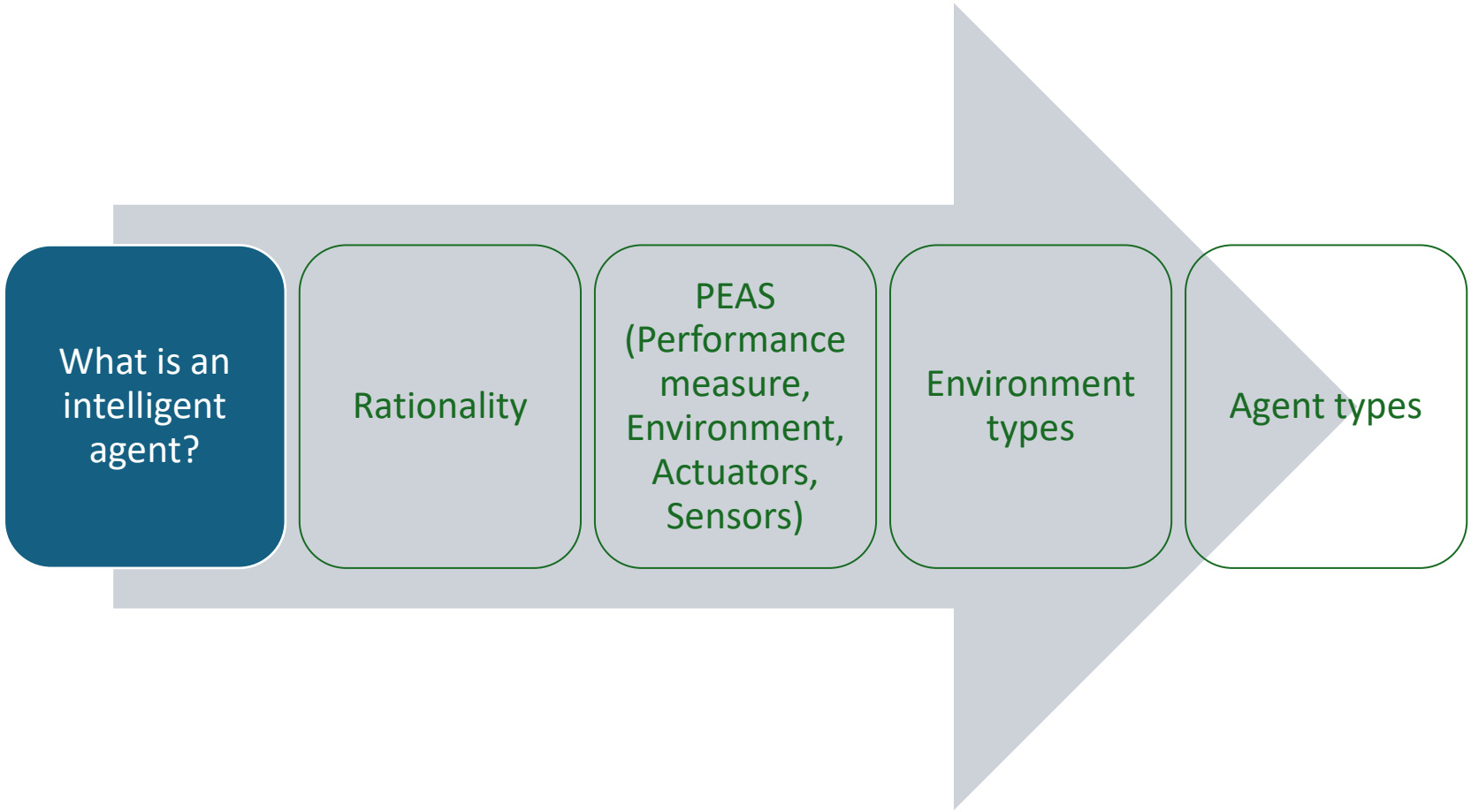
Image: "Robot at the British Library Science Fiction Exhibition" by BadgerGravling

Online Material

Outline

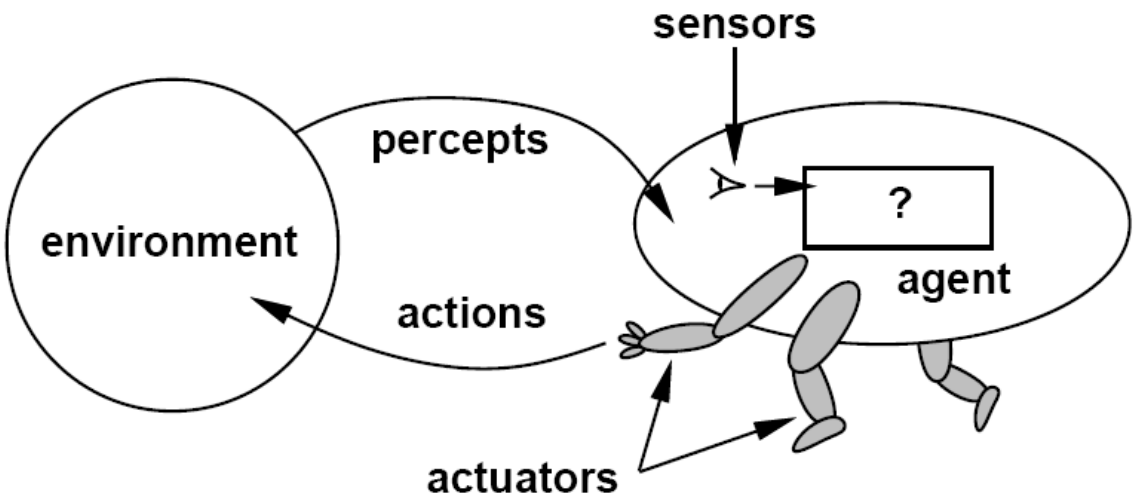

What is an intelligent agent?
Rationality
PEAS (Performance measure, Environment, Actuators, Sensors)
Environment types
Agent types

Outline: What is an Intelligent Agent



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.

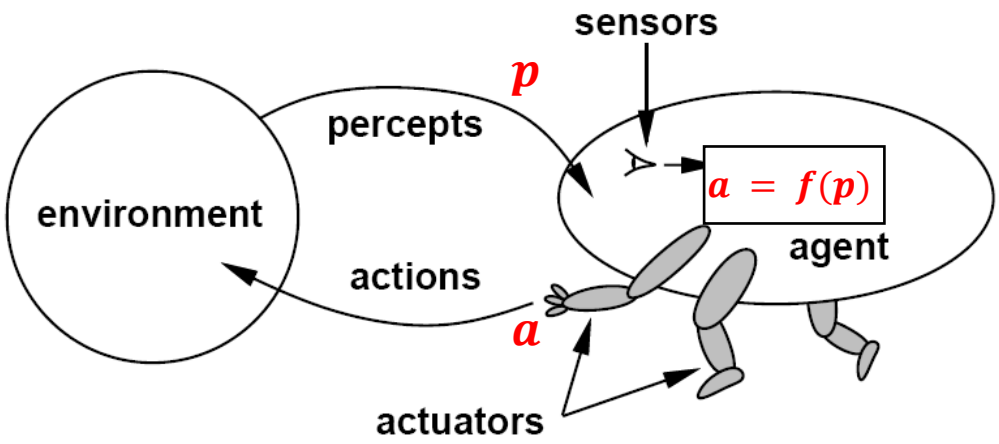


- **Control theory**: A **closed-loop control system** (= feedback control system) is a set of mechanical or electronic devices that automatically regulate a process variable to a desired state or set point without human interaction. The agent is called a controller.
- **Softbot**: Agent is a software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

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

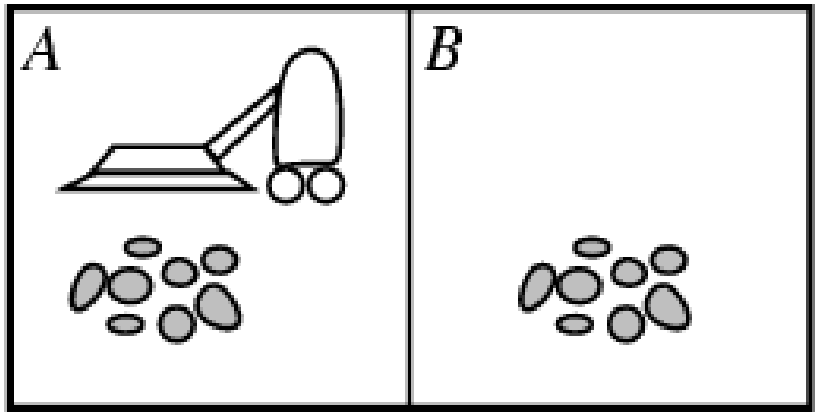


The agent program is a concrete implementation of this function for a given physical system.

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

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
…	

Problem: This table can become infinitively large!

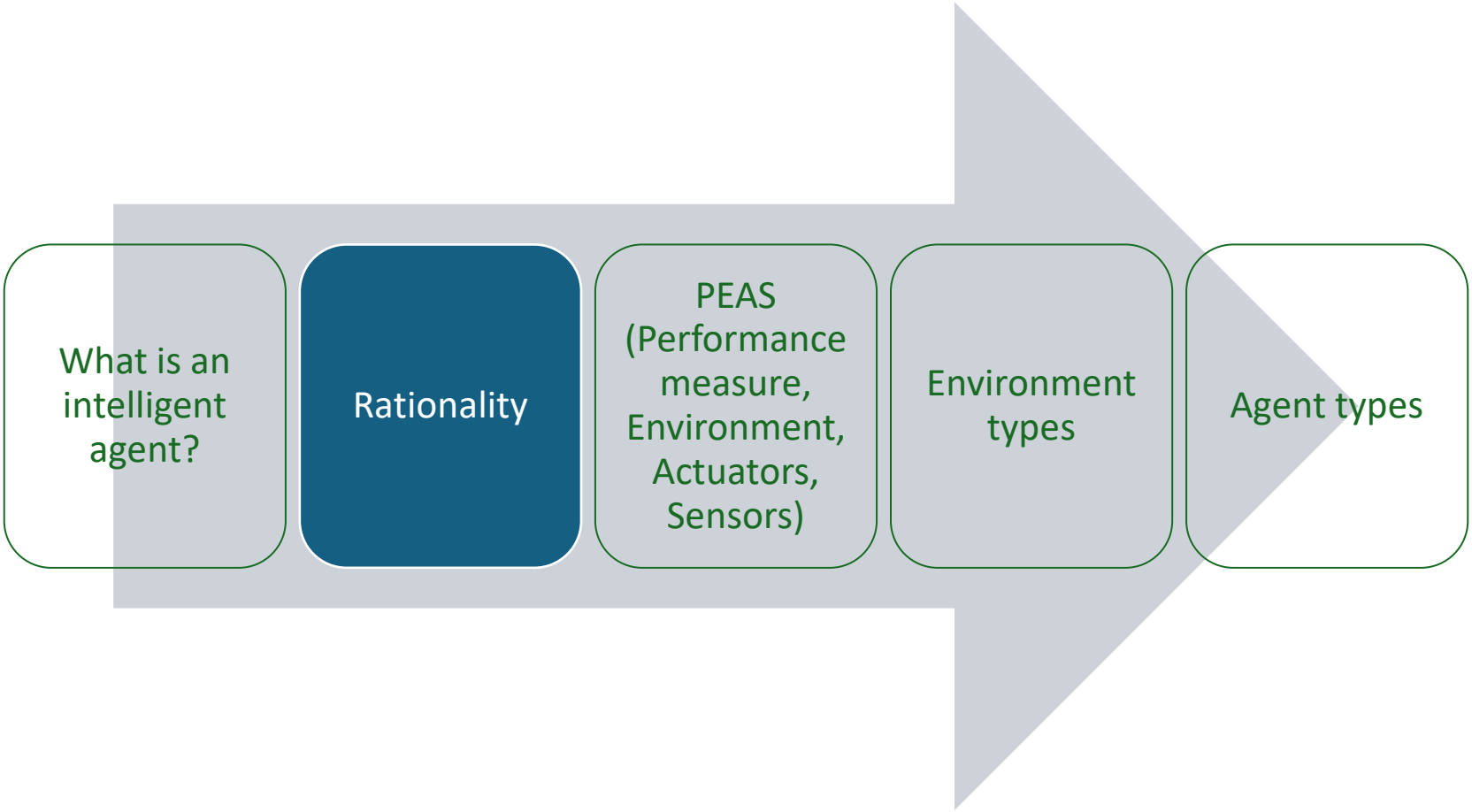
Most recent Percept p

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
```

Outline: Rationality



Rational Agents: What is Good Behavior?

Foundation from normative moral theory and economics:

- **Consequentialism**: Evaluate actions by their 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 **maximizes its expected 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.

Rule: Pick the action that maximize the expected utility

$$a = \text{argmax}_{a \in \text{A}} \, E(U \mid a)$$

Rational Agents

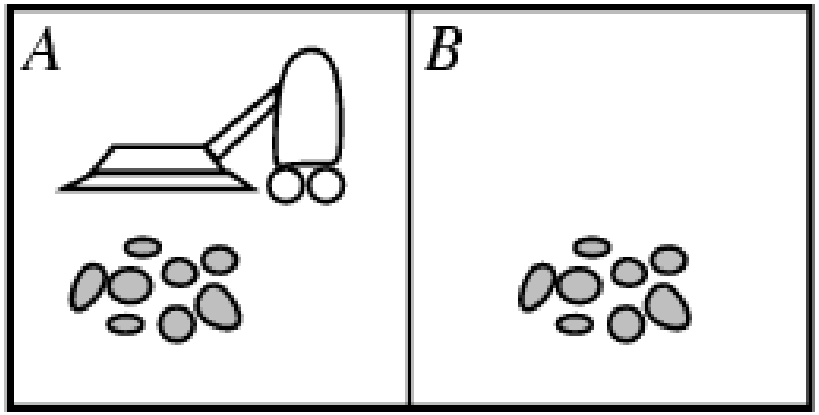
Rule: Pick the action that maximize the expected utility

$$a = \text{argmax}_{a \in \text{A}} \, E(U \mid a)$$

This means:

- **Rationality is an ideal** – it implies that no one can build a better agent
- **Rationality ≠ Omniscience** – rational agents can make mistakes if percepts and knowledge do not suffice to make a good decision
- **Rationality ≠ 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
- **Rationality is often bounded** by available memory, computational power, available sensors, etc.

Example: Performance Measure for the 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
…	

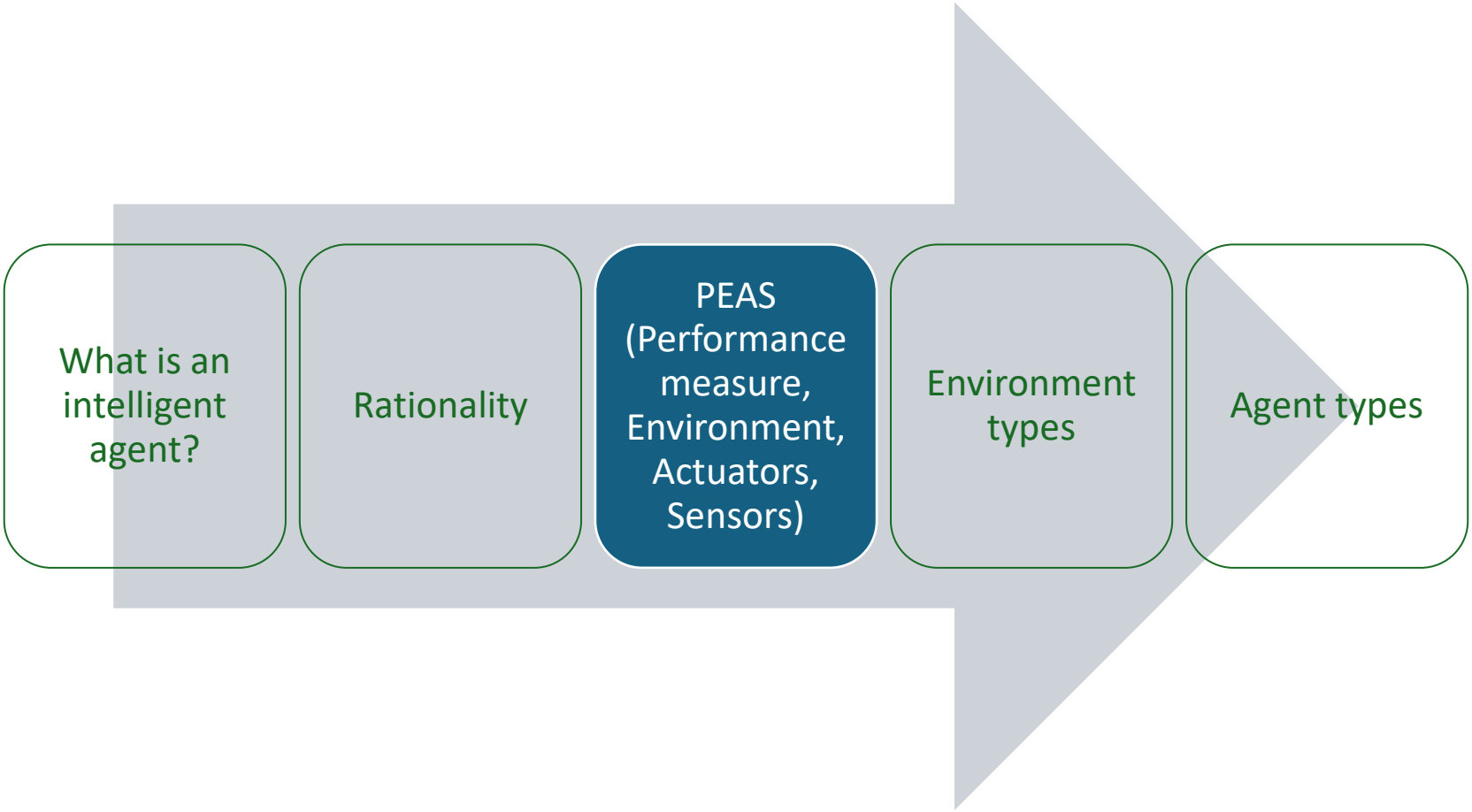
Implemented agent program:

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

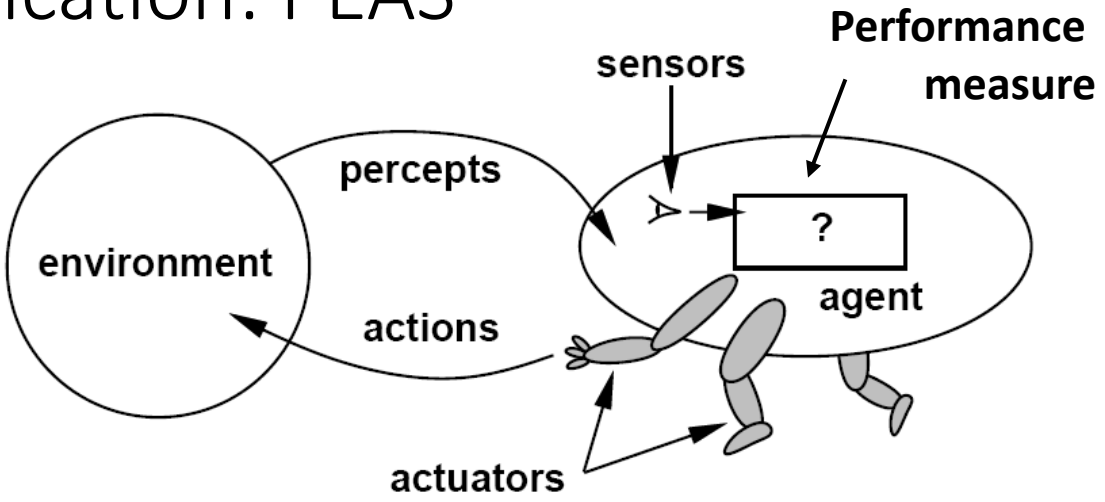
```
if status = Dirty then return Suck
else if location = A then return Right
else if location = B then return Left
```

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

Outline: PEAS



Problem Specification: PEAS



Performance measure	Environment	Actuators	Sensors
Defines utility and what is rational	Components and rules of how actions affect the environment.	Defines available actions	Defines percepts

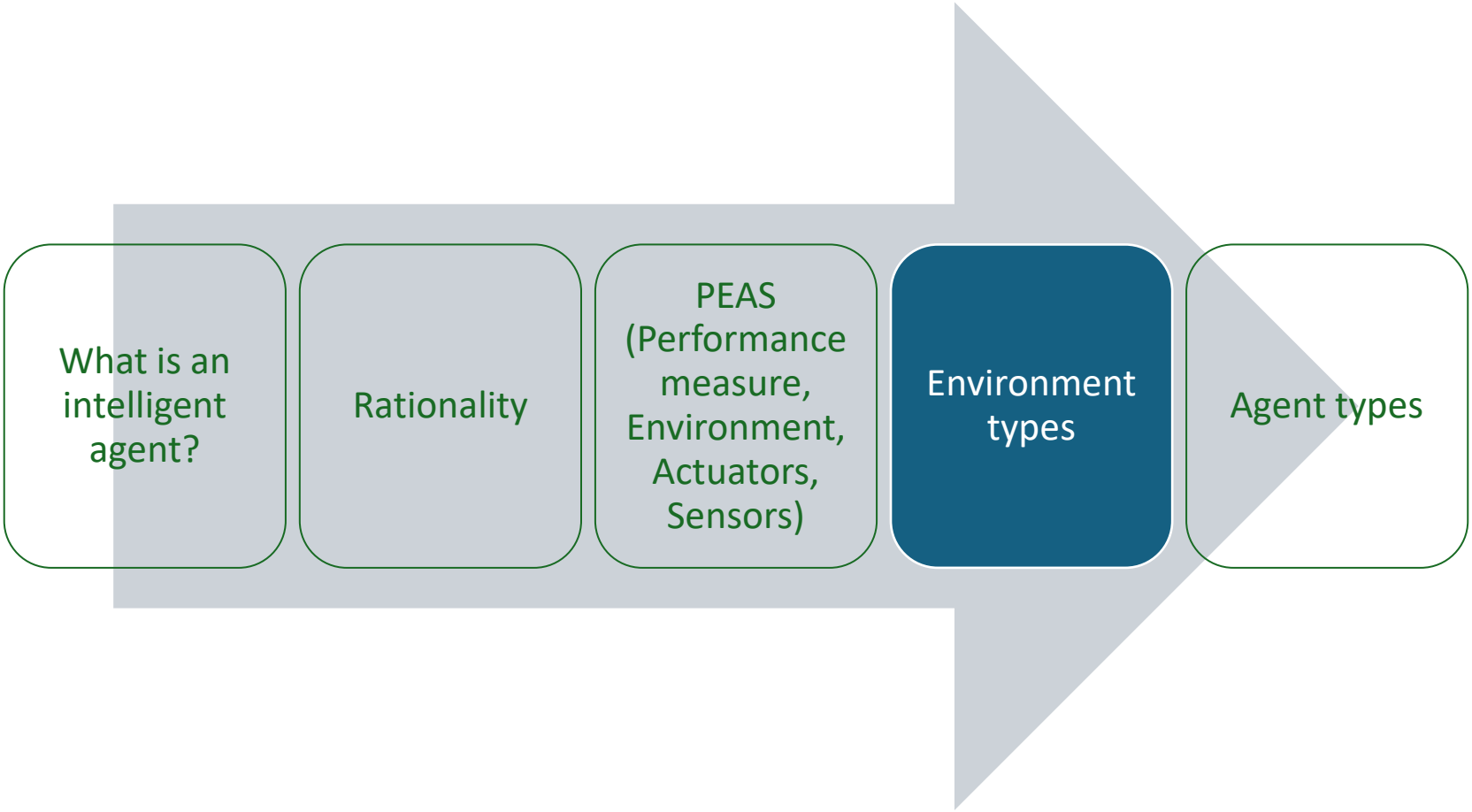
Example: Automated Taxi Driver

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

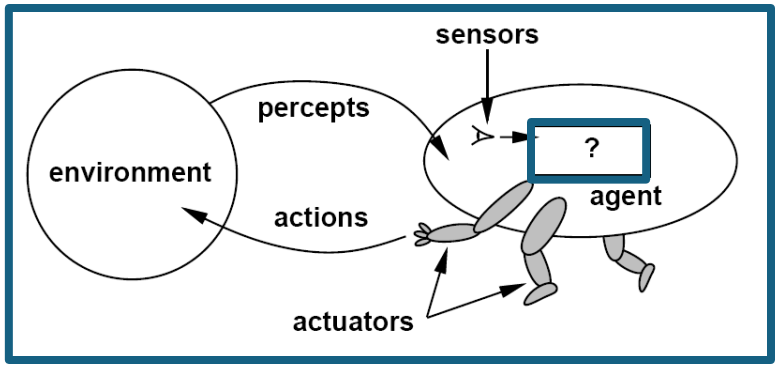
Example: Spam Filter

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

Outline: Environment Types

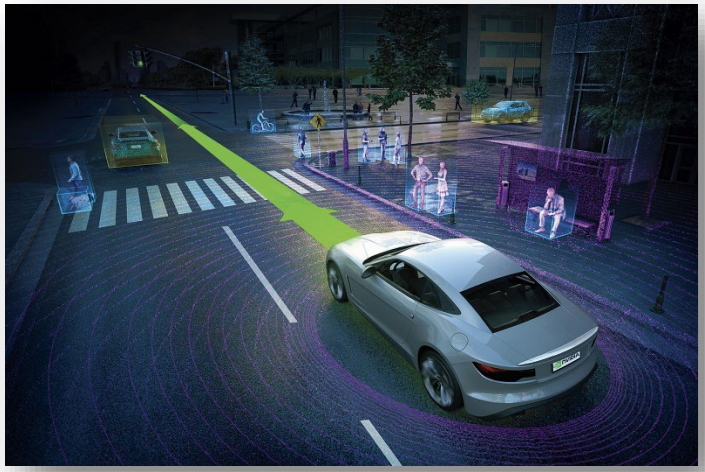


The Environment



- We typically consider everything outside the agent function (the agent's brain) as the agent's environment.
- This means that the sensors and actuators are part of the environment.
- The agent function receives already preprocessed percepts and acts by issuing high-level instructions to the actuators.

Examples:



Environment Types

Fully observable: The agent's sensors give it access to the complete state of the environment. The agent can "see" the whole environment.

vs.

Partially observable: The agent cannot see all aspects of the environment. E.g., it can't see through walls

Deterministic: Changes in the environment is completely determined by the current state of the environment and the agent's action.

vs.

Stochastic: Changes cannot be determined from the current state and the action (there is some randomness).
Strategic: The environment is stochastic and adversarial. It chooses actions strategically to harm the agent. E.g., a game where the other player is modeled as part of the environment.

Known: The agent knows the rules of the environment and can predict the outcome of actions.

vs.

Unknown: The agent cannot predict the outcome of actions.

Environment Types (cont.)

Static: The environment is **not** changing while agent is deliberating.
Semidynamic: the environment is static, but the agent's performance score depends on how fast it acts.

vs.

Dynamic: The environment is changing while the agent is deliberating.

Discrete: The environment provides a fixed number of distinct percepts, actions, and environment states. Time can also evolve in a discrete or continuous fashion.

vs.

Continuous: Percepts, actions, state variables or time are continuous leading to an infinite state, percept or action space.

Episodic: Episode = a self-contained sequence of actions. **Short episodes** for a task that the agent performs repeatedly. What the agent does in one episode does not affect future episodes.

vs.

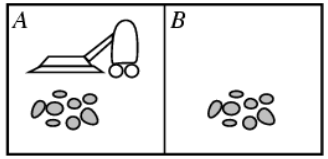
Sequential: Tasks are long, and actions taken now affect the outcomes later. The agent must consider the **long-term consequences** of its actions.

Single agent: A single agent operating in an environment.

vs.

Multi-agent: Agent cooperate or compete in the same environment.

Examples of Different Environments



Vacuum cleaner world

Chess with a clock

Scrabble

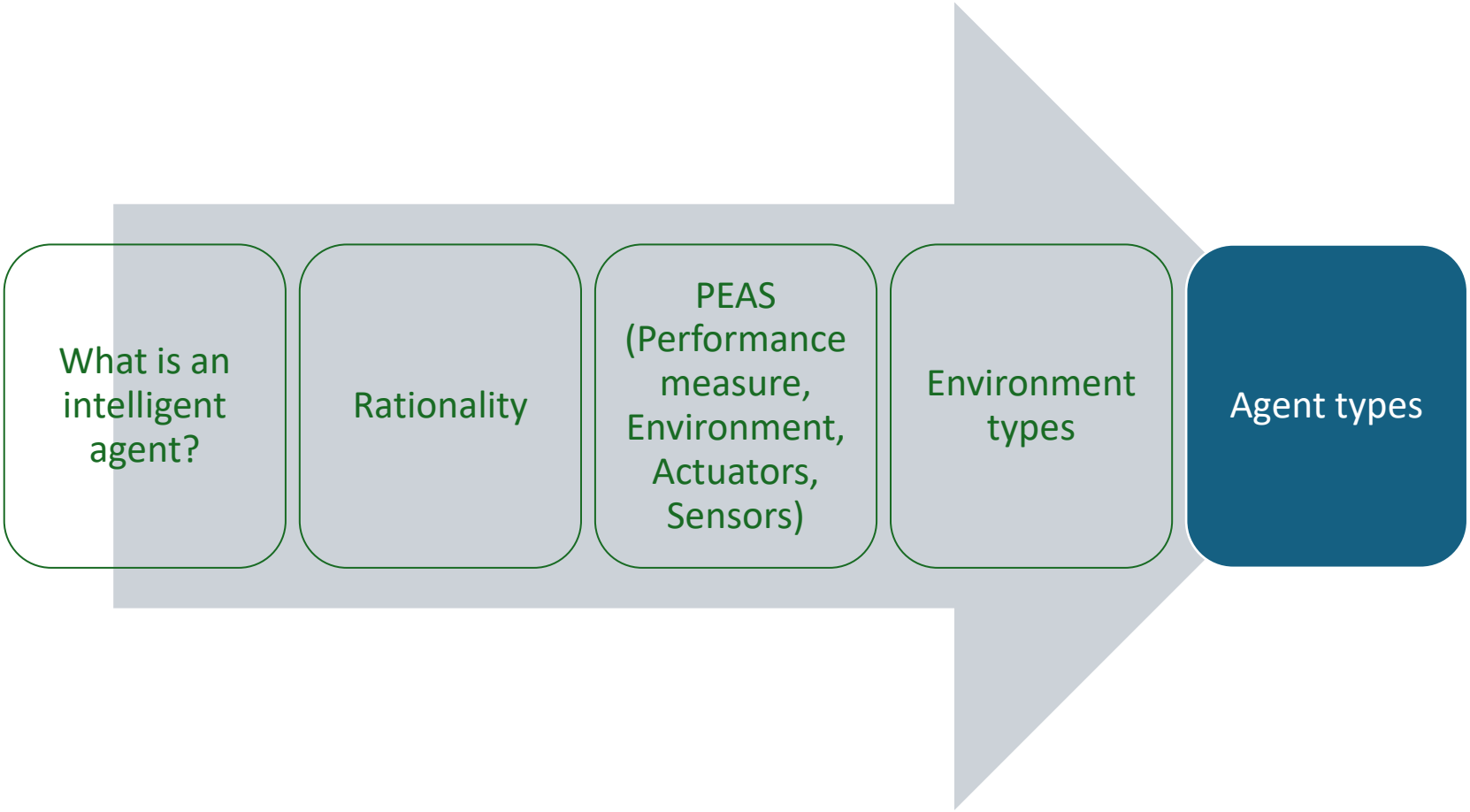
Taxi driving

Observable	Partially	Fully	Partially	Partially
Deterministic	Deterministic	Determ. game Mechanics + 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*

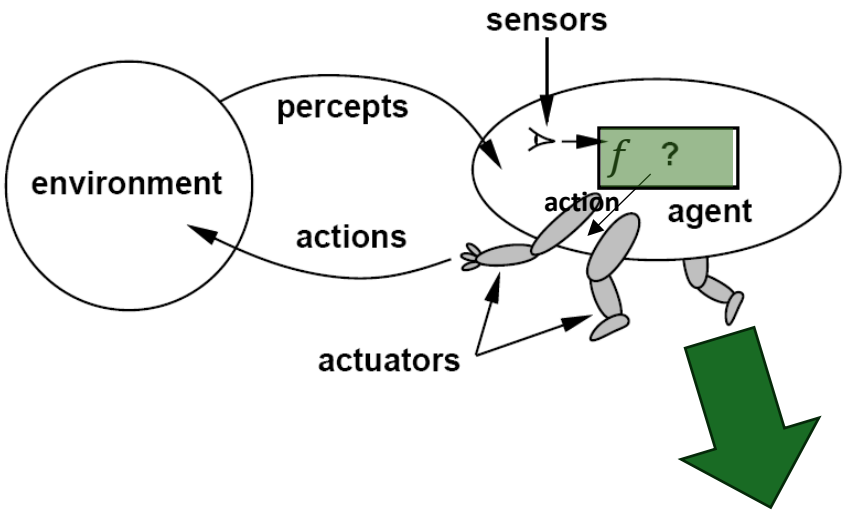
* Can be models as a single agent problem with the other agent(s) in the environment.

** A single game would be sequential environment. Multiple games could be also modeled as an episodic sequence of independent games.

Outline: Agent Types

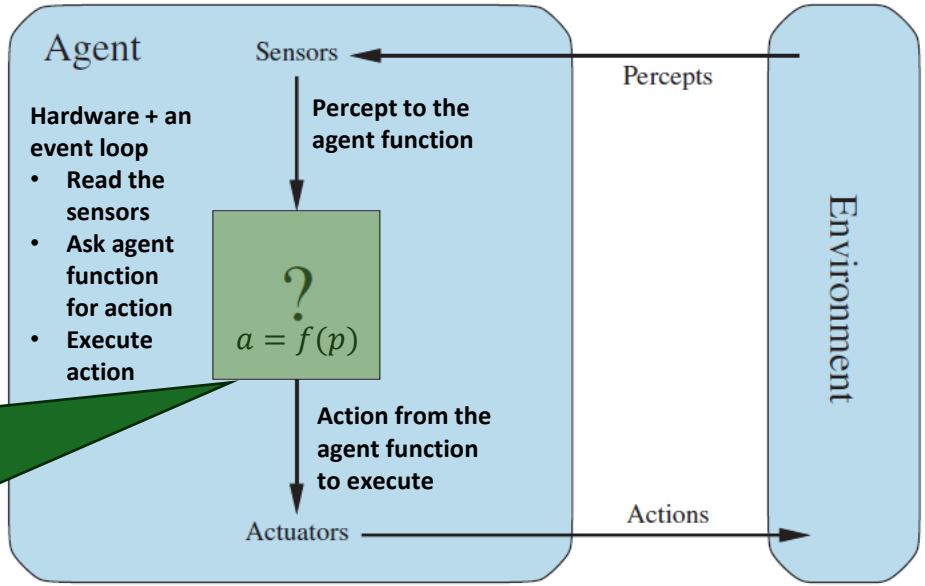


Designing a Rational Agent



Remember the definition of a rational agent:

*"For each possible percept sequence, a rational agent should select an **action** that **maximizes its expected performance measure**, given the evidence provided by the **percept sequence** and the **agent's built-in knowledge**."*

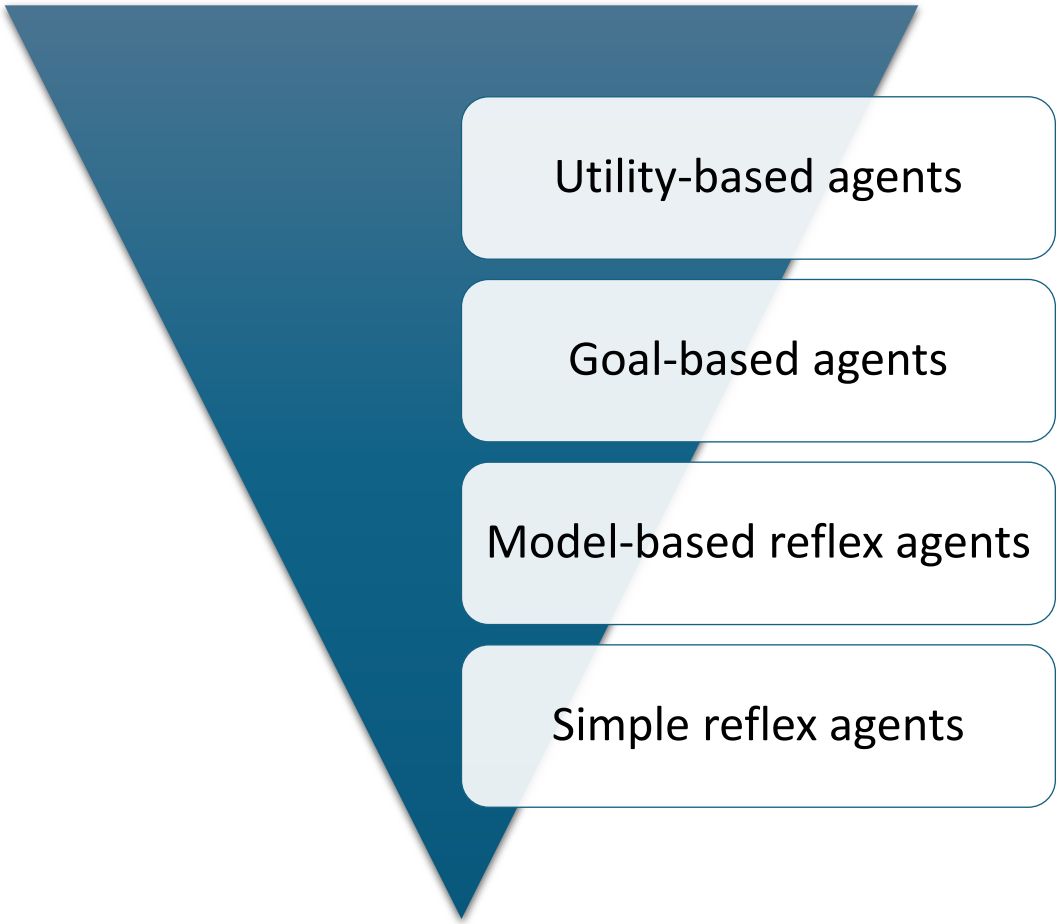


Agent Function

- Represents the "brain"
- Assess performance measure
- Remember percept sequence
- Built-in knowledge

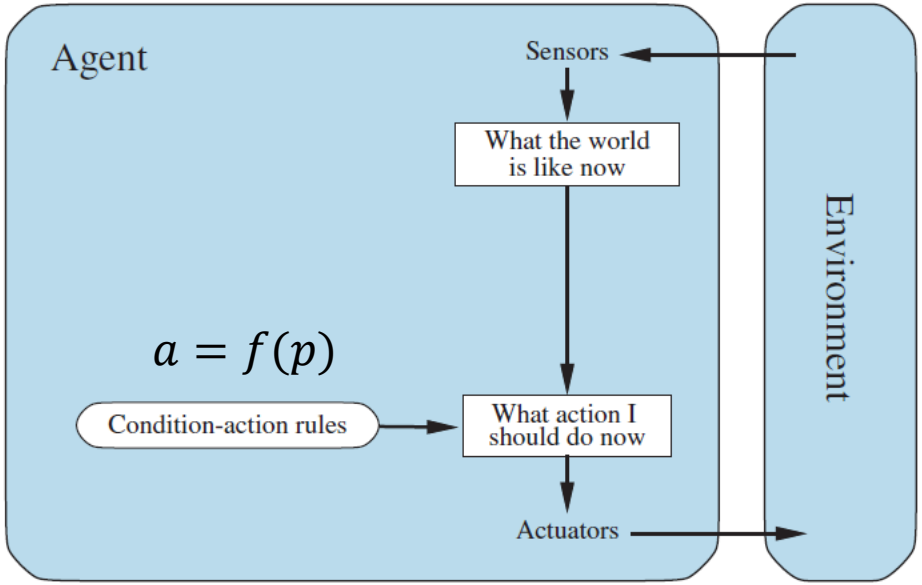
Important: Everything outside the agent function represents the environment. This includes the physical robot, its sensors and its actuators, and event loop!

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.

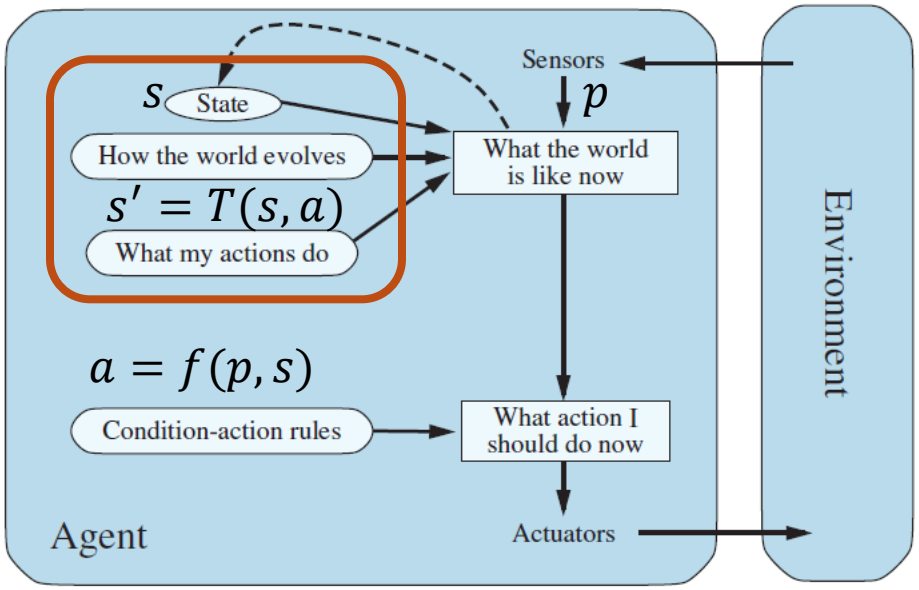


The interaction is a sequence: $p_0, a_0, p_1, a_1, p_2, a_2, \ldots p_t, a_t, \ldots$

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.
- It knows how the environment evolves over time and what its actions do (implemented as the **transition function**) to keep its state up-to-date.
- There is now **more information for the rules** to make better decisions.



The interaction is a sequence: $p_0, s_0, a_0, p_1, s_1, a_1, p_2, s_2, a_2, p_3, \ldots, p_t, s_t, a_t, \ldots$

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

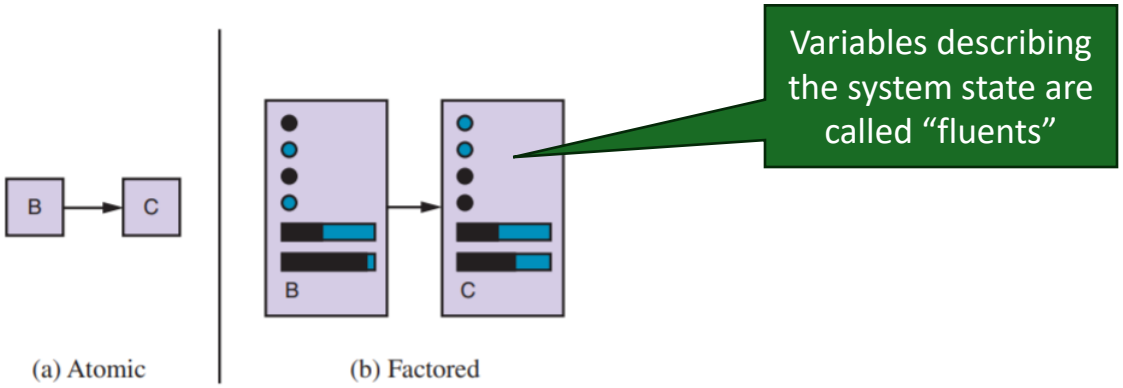
Models a snapshot of the current situation

State Representation

States help keep track of the environment and the agent in it. This is often referred to as the **system state**.

The representation can be:

- **Atomic**: Just a label for a black box. E.g., A, B
- **Factored**: A set of attribute values called fluents (because they model what can change). E.g., [location = left, status = clean, temperature = 75 deg. F]



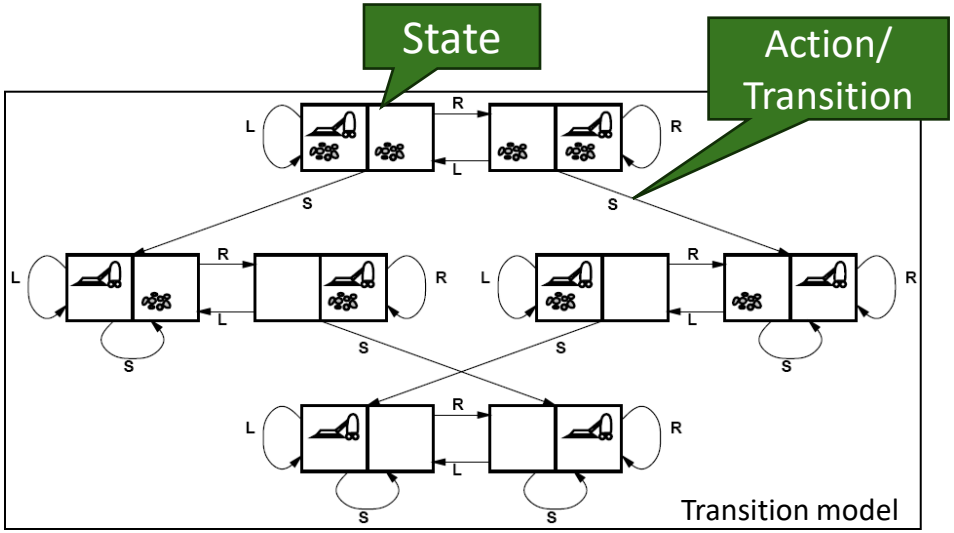
The set of all possible states is called the **state space** S. This set is typically very large!

Note: We often construct atomic labels from factored state representations. E.g.: If the agent's state is the location x = 7 and y = 3, then the atomic state label could be the string "(7, 3)". With the atomic representation, we can only compare if two labels are the same. With the factored state representation, we have more information. E.g., we can calculate the distance between the coordinates in two states.

Models how the situation changes

Transition Function

- How **the environment changes when actions are performed** is modeled as a discrete **dynamical system**.
- Example of a state diagram for the Vacuum cleaner world.



- States change because of
  a. System dynamics of the environment (the environment evolves by itself).
  b. The actions of the agent.

- Both types of changes are represented by the transition function written as

$T: S \times A \rightarrow S$ or $s' = T(s, a)$

S … set of states
A … set of available actions
$a \in A$ … an action
$s \in S$ … current state
$s' \in S$ … next state

Old-school vs. Smart Thermostat

Old-school thermostat

Percepts **States**

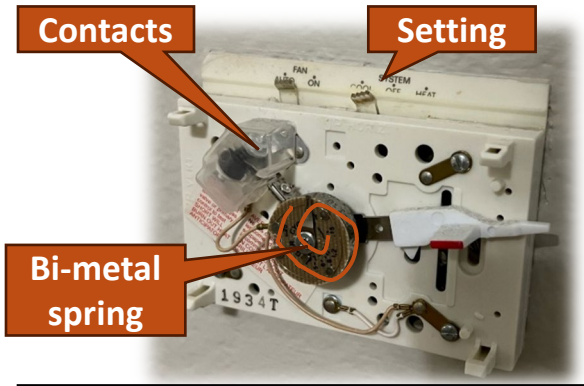
Transitions

Smart thermostat

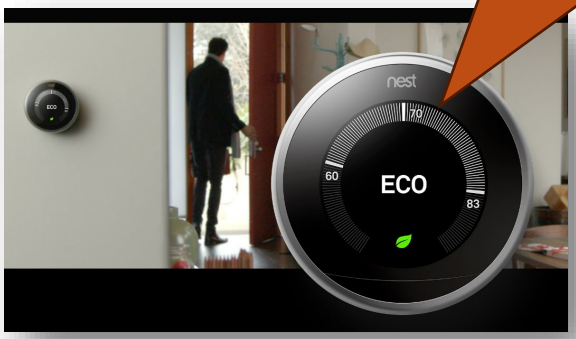
Percepts **States**

Transitions

Old-school vs. Smart Thermostat: Solution



Set target temperature



Change temperature when you are too cold/warm.

Old-school thermostat

Percepts

Setting: Cool, off, heat

Contact: Open, closed

State

The agent uses no states (only reacts to the current percepts)

Transitions

No transitions (has no states)

Smart thermostat

Percepts

Sensors
- Temp: deg. F
- Someone walking by
- Someone changes temp.

Internet
- Outside temp.
- Weather report
- Energy curtailment
- Day & time
- …

State

Fluents model current situation

Factored description
- Estimated time to cool the house
- Someone home?
- How long till someone is coming home?
- Schedule
- ….

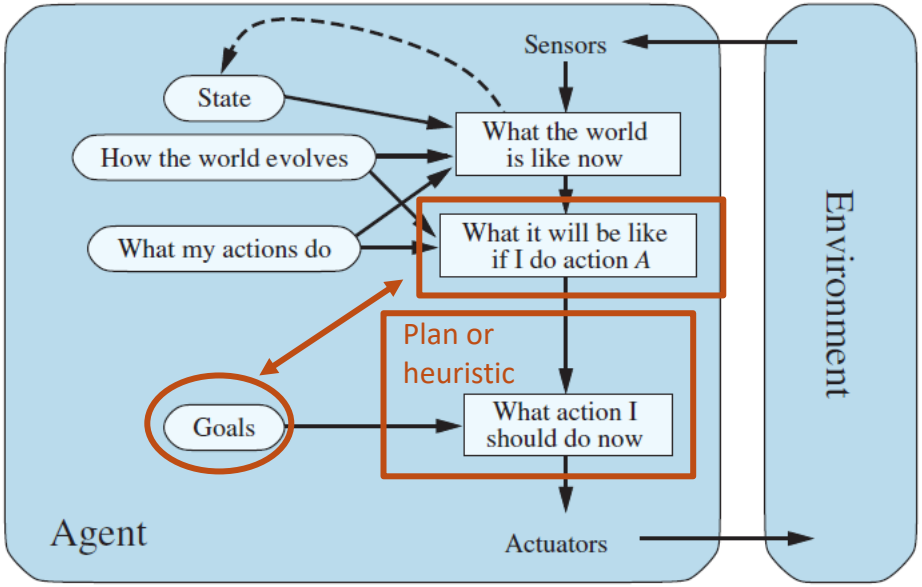
Transitions

Many: E.g., Person walks by -> someone is home.
Temperature changes -> estimated cool time changes

Actions or changes in the environment change the state

Goal-based Agent

- The agent has the task of reaching a defined **goal state**, and then it is done.
- The agent needs to choose actions to move towards the goal. Subtypes:
  - **Greedy or heuristic goal-seeking agent**: Choose the next action to move towards the goal.
  - **Planning agent**: Use **search algorithms** to plan a sequence of actions that leads to the goal.
- Performance measure: the **cost to reach the goal**.



$$a = \text{argmin}_{a_0 \in A}\left[\sum_{t=0}^{T} c_t \,\middle|\, s_T \in S^{goal}\right]$$

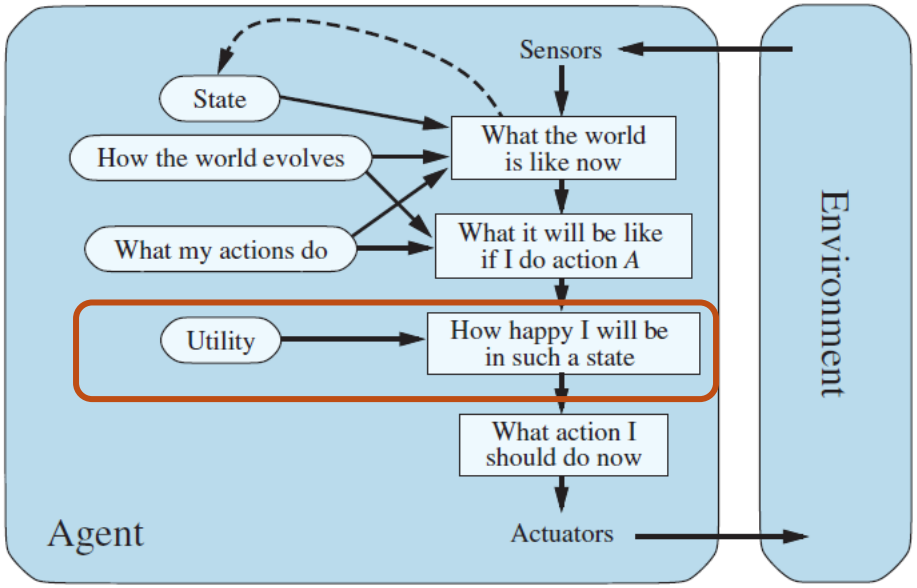
Sum of the cost of a planed sequence of actions that leads to a goal state

The interaction is a sequence: $\underbrace{p_0, s_0, a_0, p_1, s_1, a_1, p_2, s_2, a_2, \ldots, s^{goal}}_{\text{cost}}$

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 $R(s)$.
- Choose actions to stay in desirable states.
- Performance measure: The discounted sum of **expected utility over time**.



$$a = \text{arg}max_{a_0 \in \text{A}} E\left[\sum_{t=0}^{\infty} \gamma^t r_t \,|a_0\right]$$

Implements rational behavior: Utility is the expected future discounted reward

Techniques: Markov decision processes, reinforcement learning

The interaction is a sequence: $\underbrace{p_0, s_0, a_0, p_1, s_1, a_1 . p_2, s_2, a_2, \ldots}_{\text{reward}}$

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

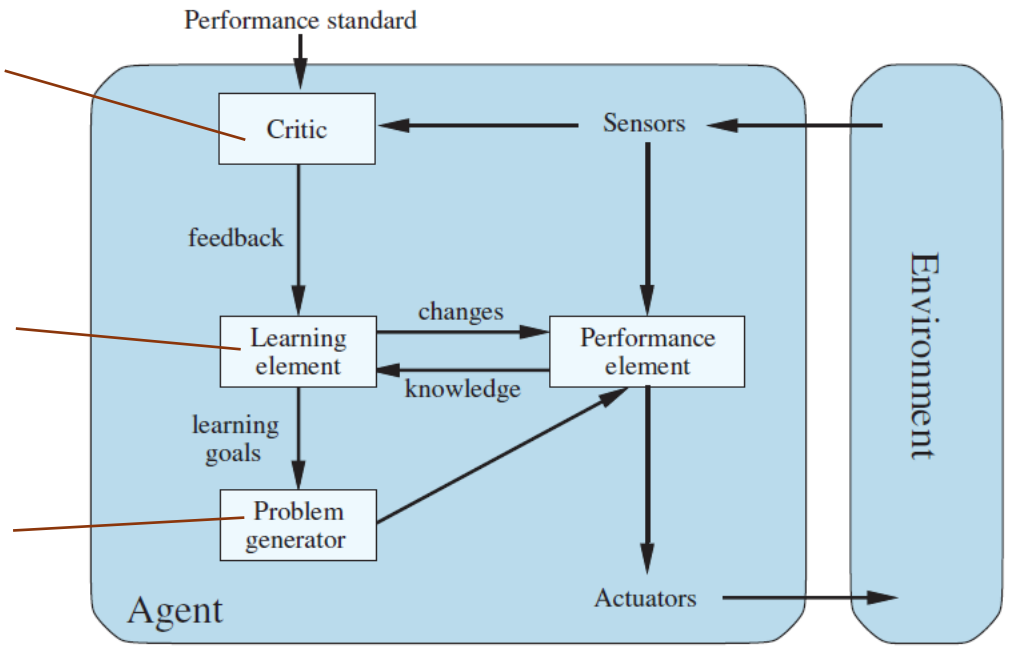
Agents that Learn

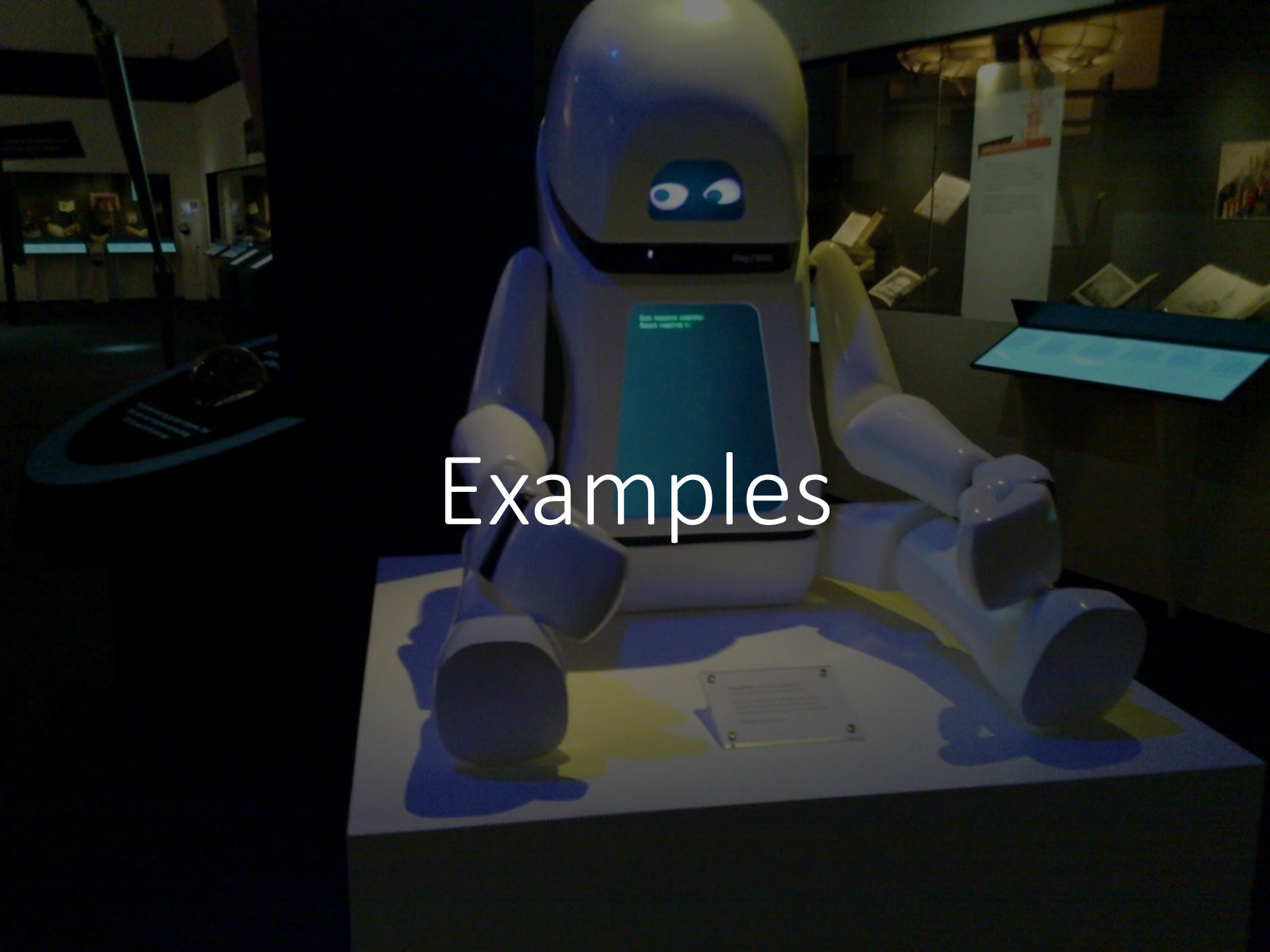
The **learning element** modifies the agent program (reflex-based, goal-based, or utility-based) to improve its performance.

How is the agent currently performing?

Updates how the performance element chooses actions.

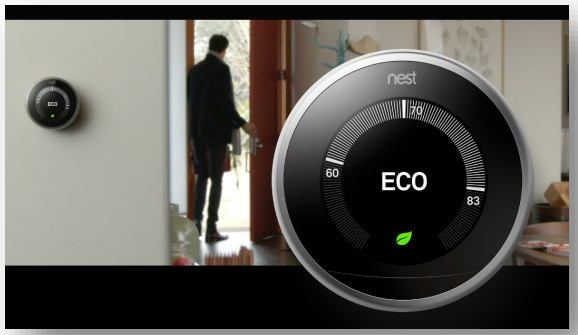
Generate actions for exploration





Examples

Smart Thermostat: What Type of Agent is it?



Change temperature when you are too cold/warm

Simple Reflex Agent?

Model-based Reflex Agent?

Goal-based?

Utility-based?

Learning?

Smart thermostat

Percepts

Sensors
- Temp: deg. F
- Someone walking by
- Someone changes temp.

Internet
- Outside temp.
- Weather report
- Energy curtailment
- Day & time
- …

States

Factored states
- Estimated time to cool the house
- Someone home?
- How long till someone is coming home?
- Schedule
- ….

Example: Modern Vacuum Robot

Features are:

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

iRobot Roomba 960

Read TechHive's review

$247.17 See It

on Amazon (Prime Eligible)

The Roomba 960's flawless navigation, stellar cleaning, and advanced features set it apart from all other robot vacuums.

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

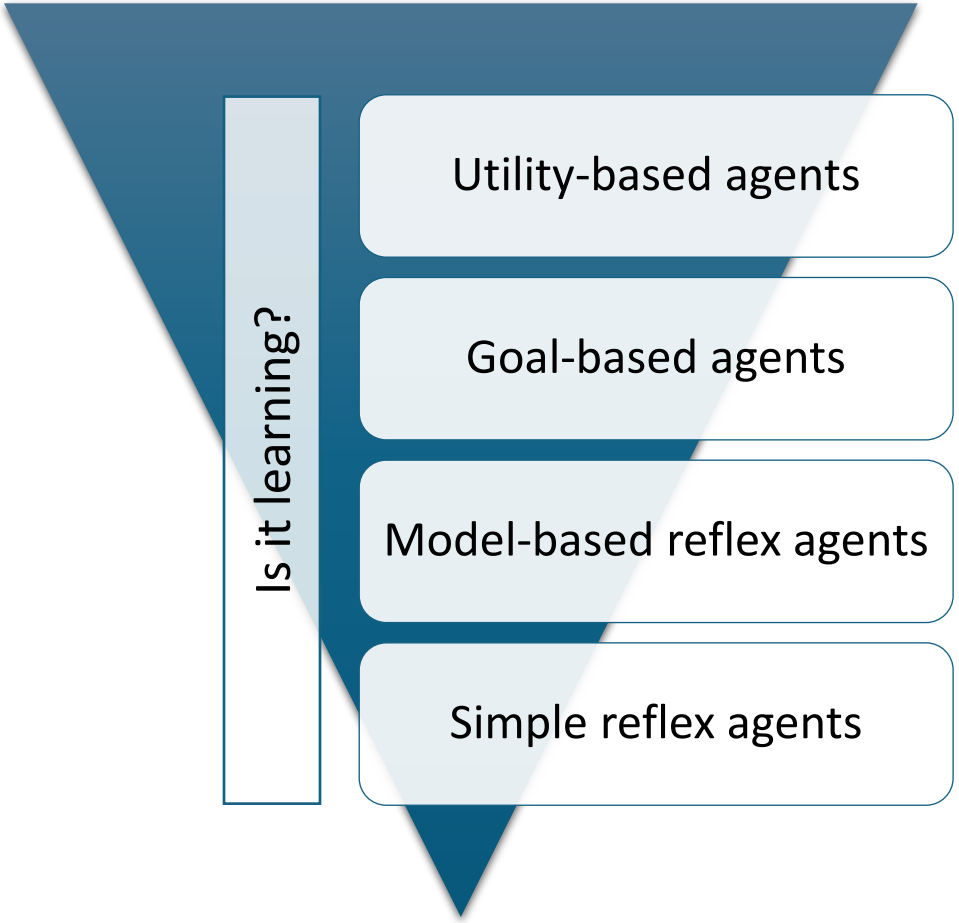
PEAS Description of a Modern Robot Vacuum

Performance measure	Environment	Actuators	Sensors

PEAS Description of a Modern Robot Vacuum: Solution

Performance measure	Environment	Actuators	Sensors
• Time to clean 95% • Does it get stuck?	• Rooms • Obstacles • Dirt • People/pets • …	• Wheels • Brushes • Blower • Sound • Communicate to server/app	• Bumper • Cameras/dirt sensor • Laser • Motor sensor (overheating) • Cliff detection • Home base locator

What Type of Intelligent Agent is a Modern Robot Vacuum?



Does it collect utility over time? How would the utility for each state be defined?

Does it actively try to reach a goal state?

Does it store state information? How would states be defined (atomic/factored)?

Does it use simple rules based only on the current percepts?

☑ Check what applies

Example: Large Language Models

Default (GPT-3.5)

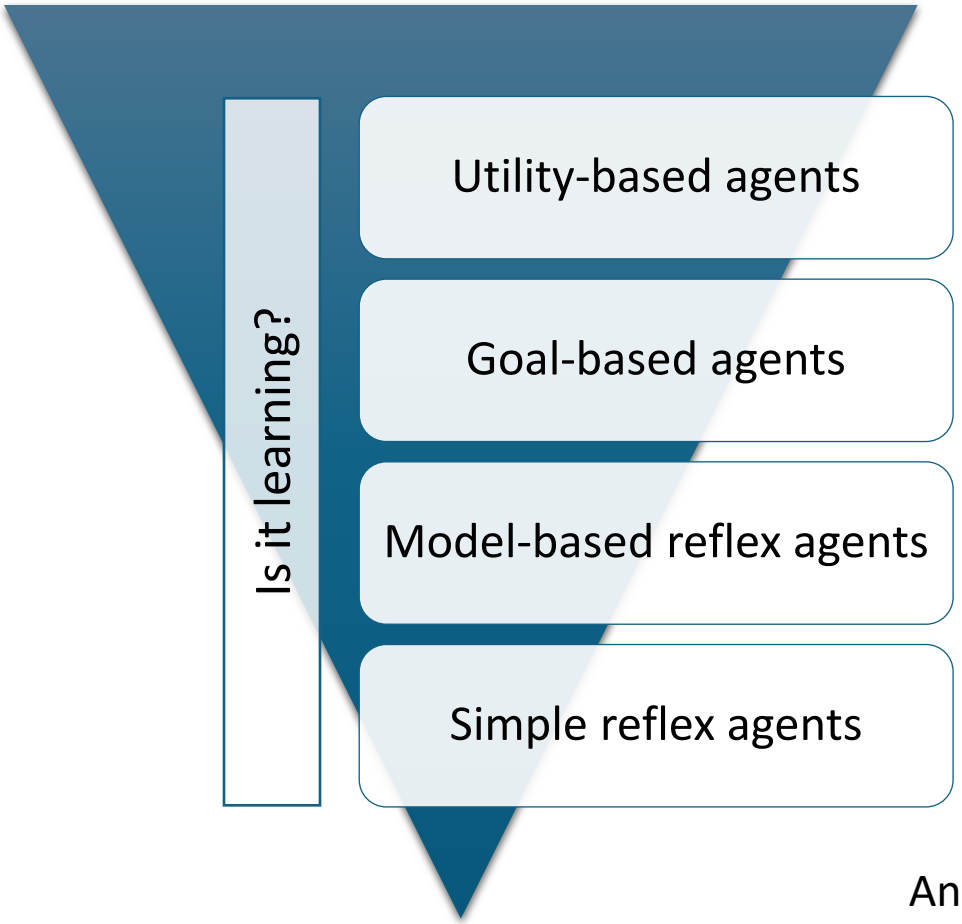
the sun is shining. It is

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

Performance measure	Environment	Actuators	Sensors

What Type of Intelligent Agent is ChatGPT?



Does it collect utility over time? How would the utility for each state be defined?

Does it actively try to reach a goal state?

Does it store state information? How would the state be defined (atomic/factored)?

Does it use simple rules based on the current percepts?

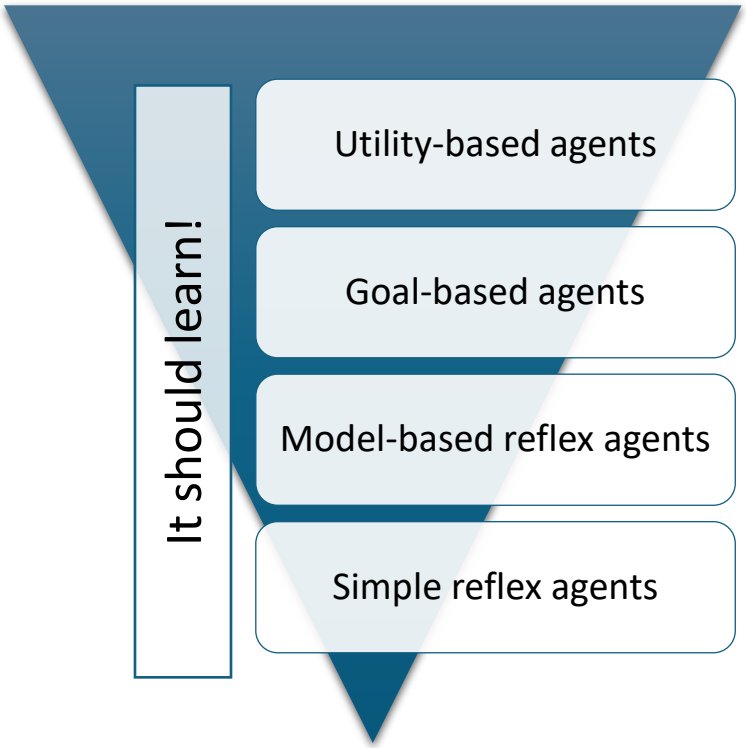
☑ Check what applies

Answer the following questions:

- Does ChatGPT pass the Touring test?
- Is ChatGPT a rational agent? Why?

We will talk about knowledge-based agents later.

Intelligent Systems a Sets of Agents: Self-driving Car



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

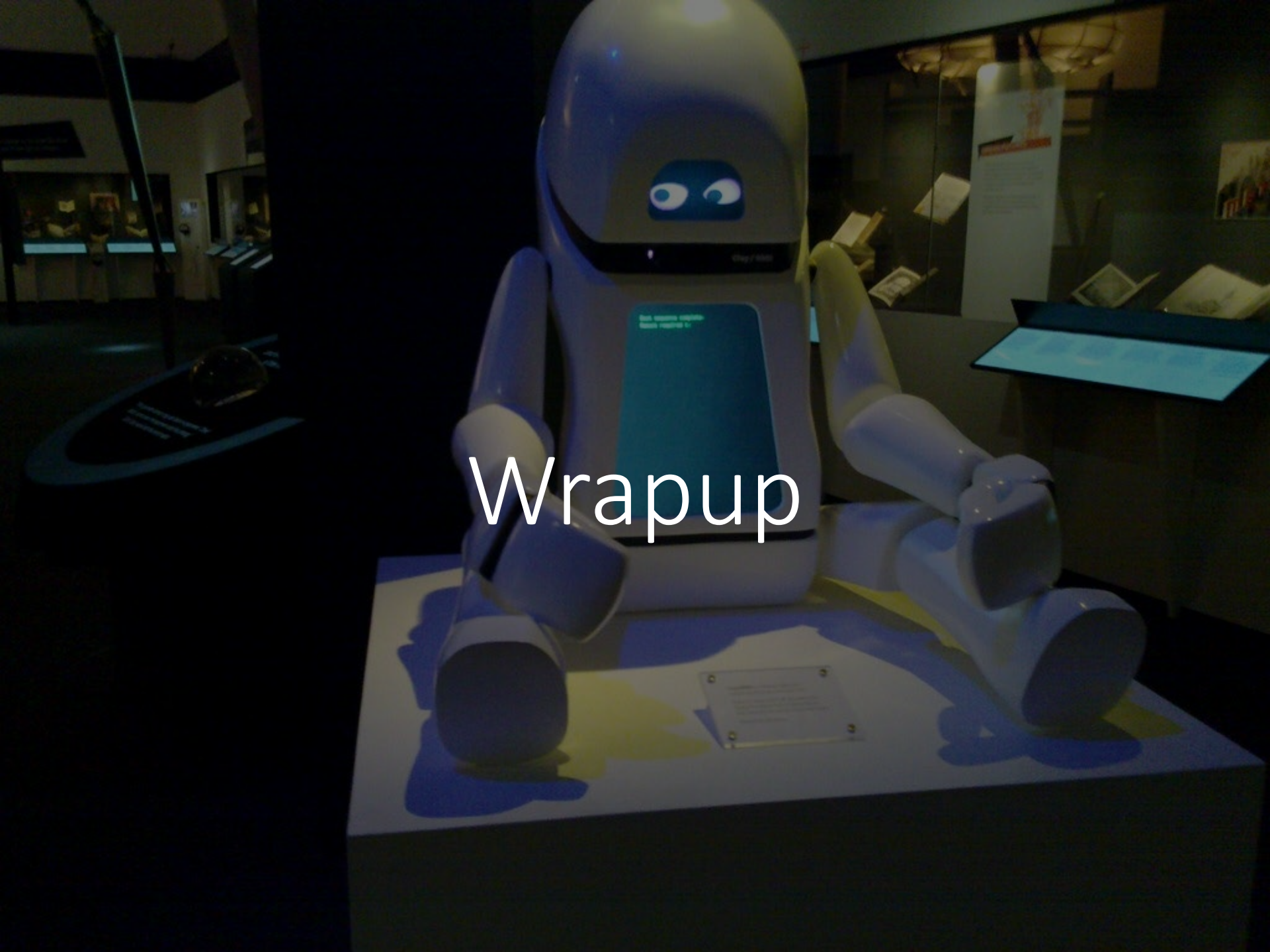
Plan the route to the destination.

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

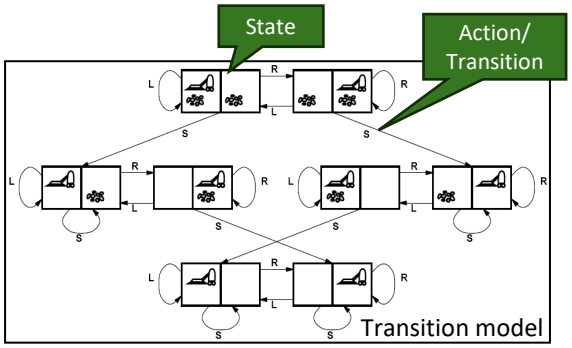
React to unforeseen issues like a child running in front of the car quickly.

High-level planning

Low-level planning


Wrapup

Important Environment Types Revisited



State

Fully observable: The agent has access to the complete current **state** of the environment.
It has deterministic percepts that are 100% reliable.

vs.

Partially observable: The agent's sensors provide incomplete or noisy information about the **state** of the environment.
Noisy information means unreliable stochastic percepts (aka a stochastic sensor model)

Transition Function

Deterministic:
Deterministic **transition function**: Changes in the environment are completely determined by the current state of the environment and the agent's action.

vs.

Stochastic:
Stochastic **transition functions** lead to belief states, transition probabilities, and a Markov process.

Known: The agent knows the **transition function**.

vs.

Unknown: The needs to **learn the transition function** by trying actions.

We will spend the whole course on discussing algorithms that can deal with environments that have different combinations of these three properties.

AI Areas

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)

What You Should Know

- What an **agent function**
  $$action = f(percepts)$$
  is and how it interacts with the environment.
- What are **states** and what is the **transition function**?
- How **environments** differ in terms of observability, uncertainty (stochastic behavior), and if the transition function is known.
- How to identify different **types of agents**.