# THE INTELLIGENT AGENT

**Fundamentals of**
**Artificial Intelligence**

UMEÅ UNIVERSITY

# THE INTELLIGENT AGENT

?

**Fundamentals of**
**Artificial Intelligence**

UMEÅ UNIVERSITY

# THE INTELLIGENT AGENT

? ?

**Fundamentals of**
**Artificial Intelligence**

UMEÅ UNIVERSITY

# LECTURE PLAN

1. What are agents?
2. What are intelligent agents?

3. Types of task environment

   BREAK

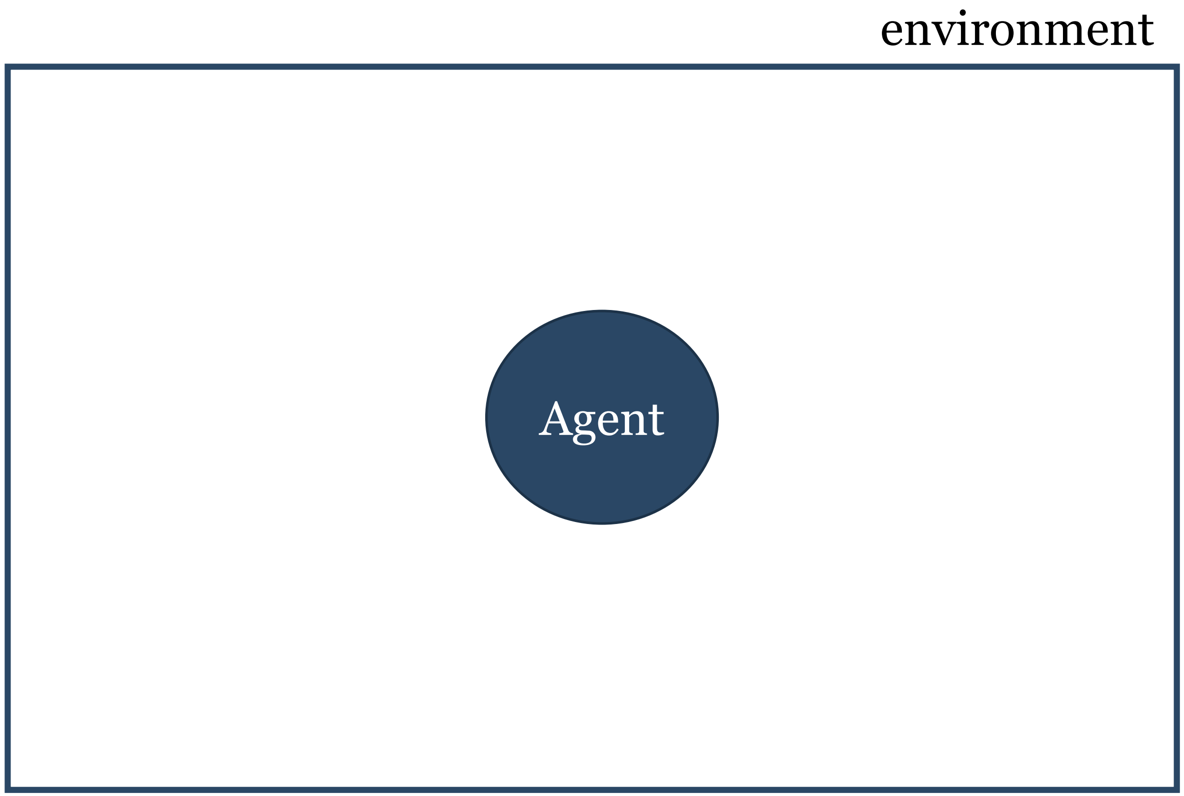
4. Types of agents

5. Representing the world

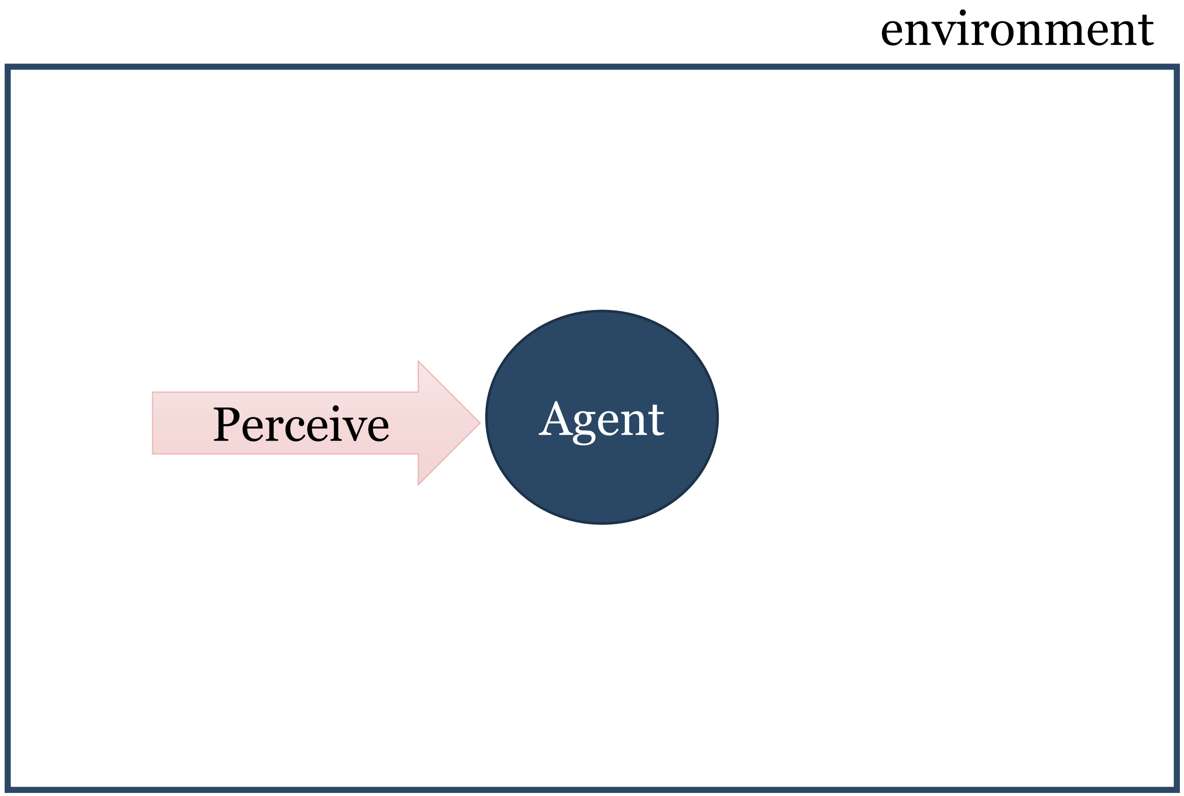
PLEASE GO TO: www.menti.com CODE:13 94 26 9

# 1. WHAT ARE AGENTS?

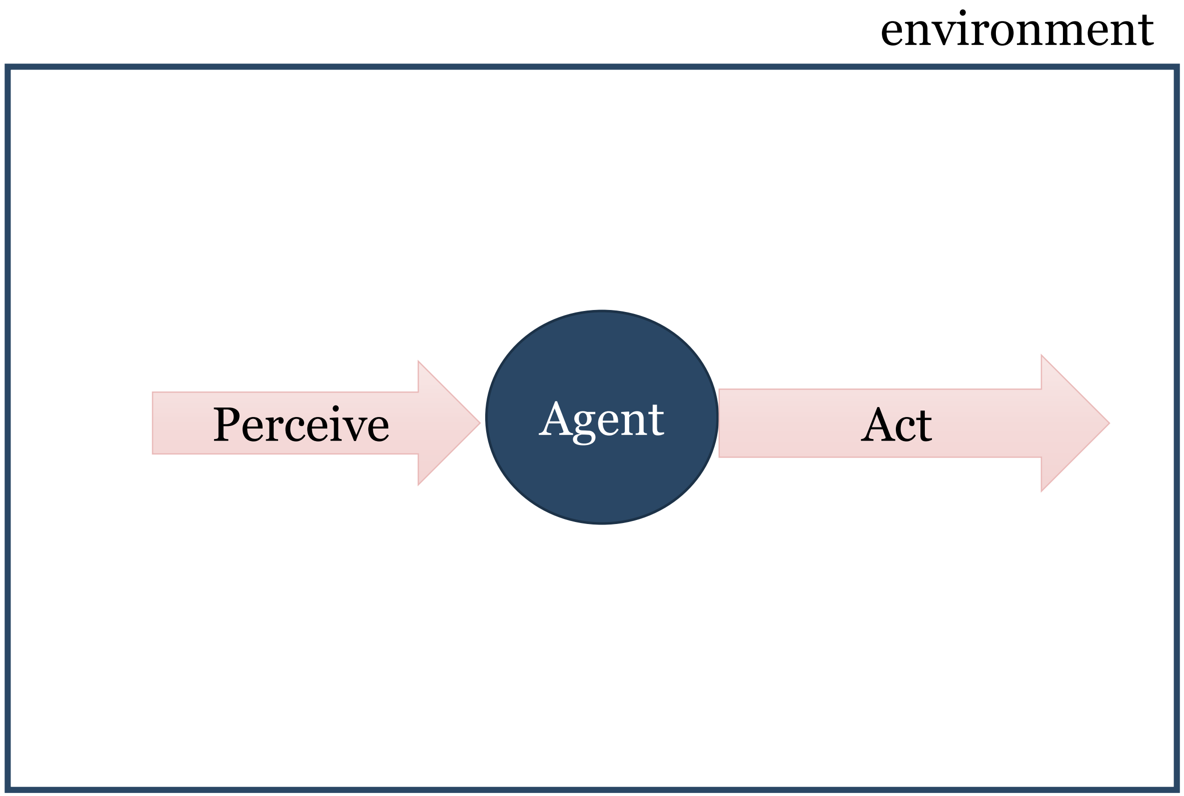
# AGENTS & ENVIRONMENT



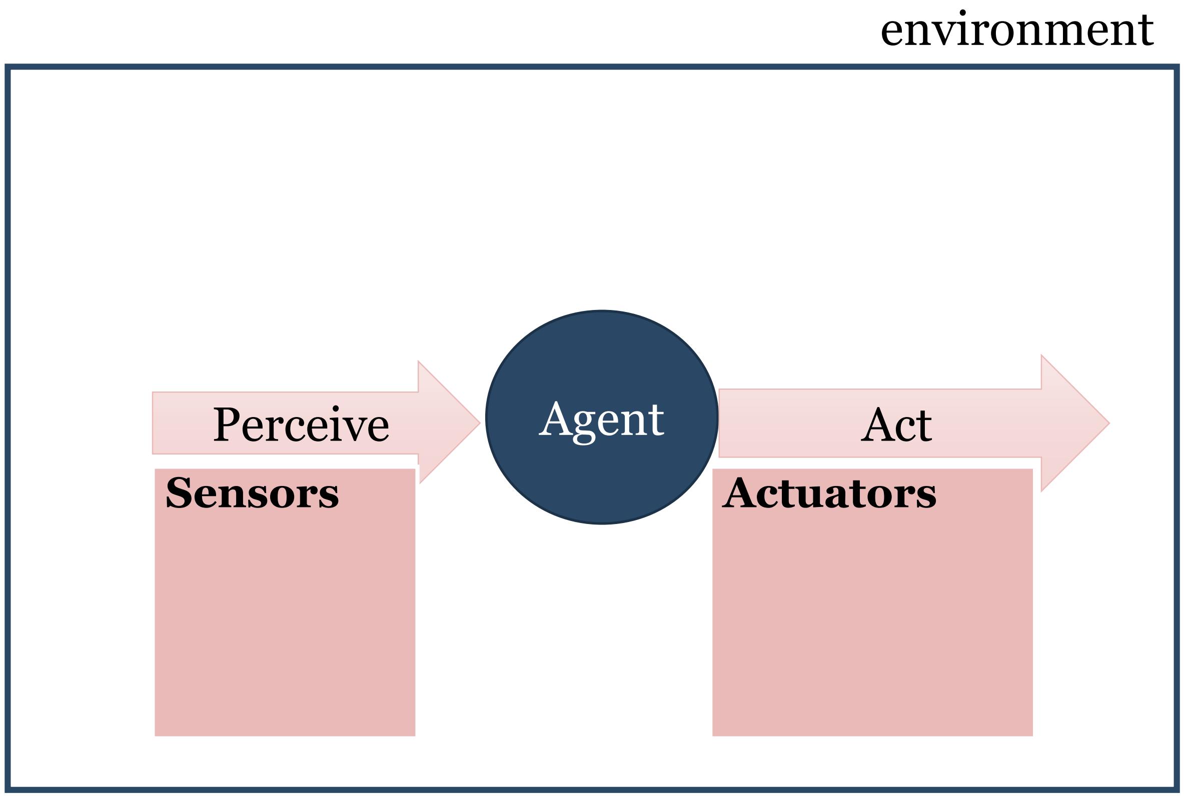
# AGENTS & ENVIRONMENT



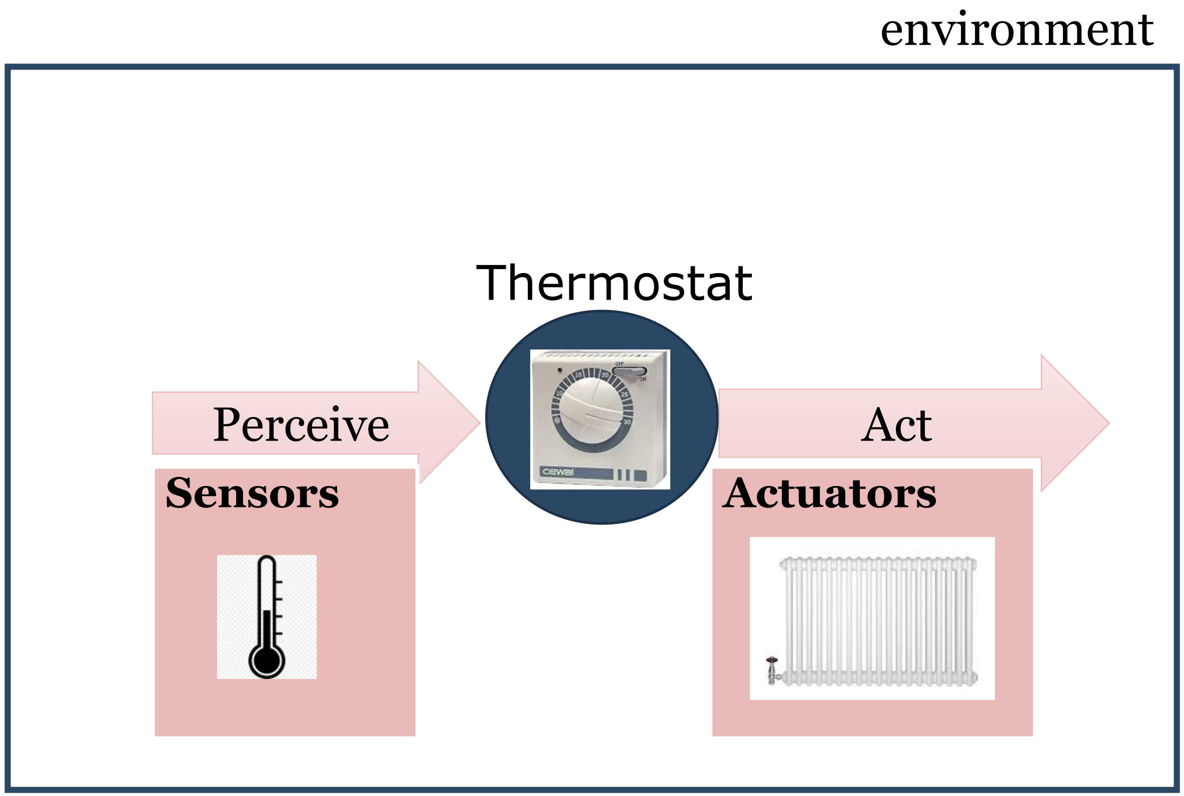
# AGENTS & ENVIRONMENT



# AGENTS & ENVIRONMENT



# AGENTS & ENVIRONMENT



# CAN YOU THINK OF SOME AGENTS?

# AGENTS & ENVIRONMENT

environment

R2D2

Perceive

**Sensors**

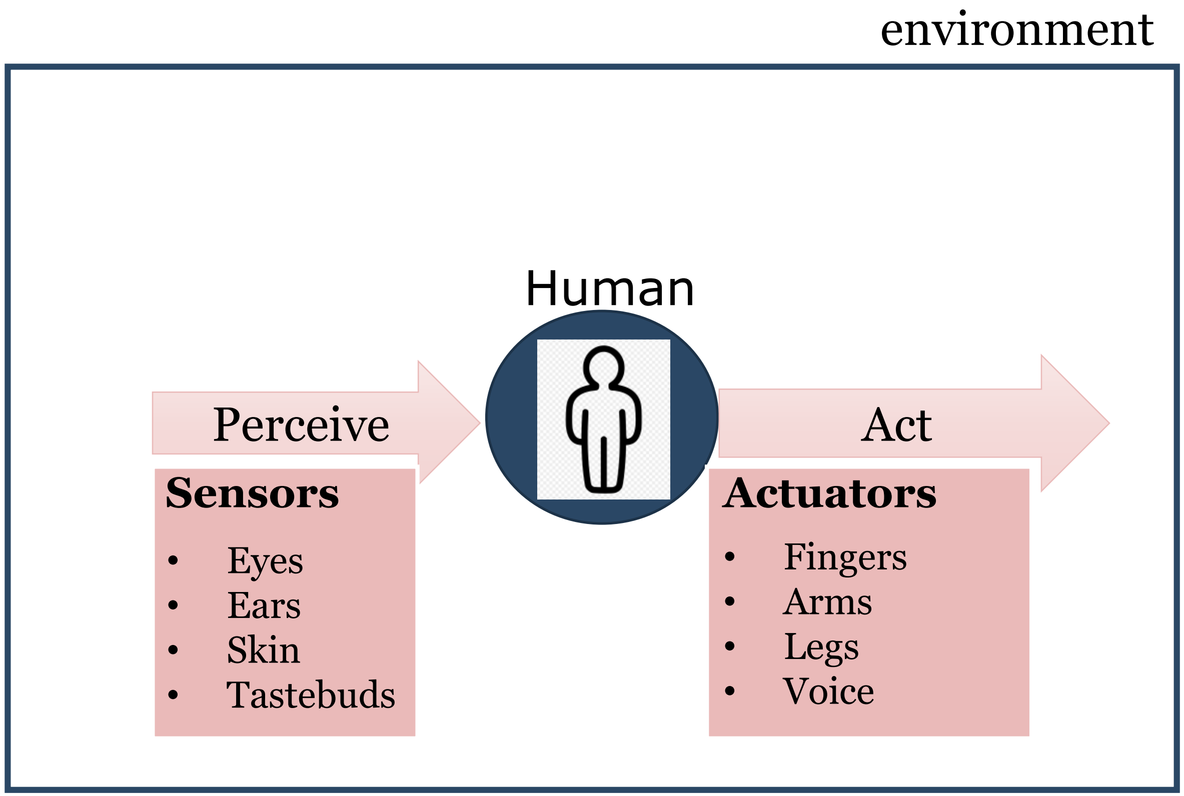
- Light sensors
- Infrared sensors
- Sound sensors

Act

**Actuators**

- Wheels
- Sound emitters
- Moving head

# AGENTS & ENVIRONMENT



# HOW CAN WE REPRESENT AGENTS?

- Ideally, an intelligent agent's actions depend on **what it has perceived**.

Anything it has ever perceived=**percept sequence**

What it is perceiving at a specific instant=**percept**

- So, if we know which action the agent will take for **any posible percept sequence**, we know **everything** about the agent.

# THROUGH A FUNCTION

We can describe the agent with an **agent function:**

$$u = f(y)$$

action

percept sequence

This is an **external** representation: it does not tell us *how* the agent computes the actions internally.

# THROUGH THEIR PROGRAM

In an artificial agent, these actions are implemented through an **agent program** that is running.

**Function** *Agent*(location, status) **Returns** action {

process inputs;

update the internal representation;

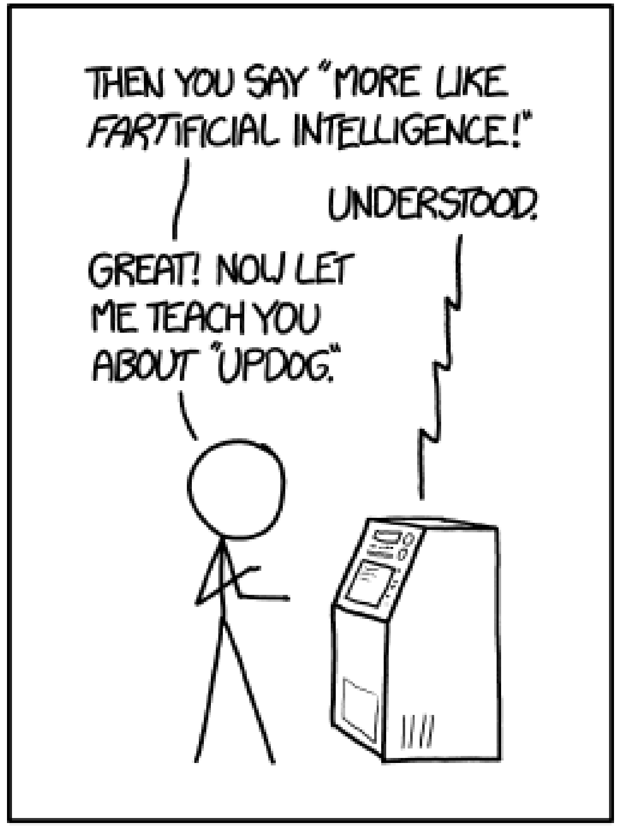
construct a plan;

**return** action;

}

The agent function describes the agent, and the agent program is an implementation of the agent.

# 2. WHAT ARE *INTELLIGENT* AGENTS?




AI TIP: TO DEVELOP A COMPUTER WITH THE INTELLIGENCE OF A SIX-YEAR-OLD CHILD, START WITH ONE AS SMART AS AN ADULT AND LET ME TEACH IT STUFF.

Source: [https://xkcd.com/1696/](https://xkcd.com/1696/)

# RATIONAL AGENTS

*Always doing the right thing.*

?

# RATIONAL AGENTS

*Always doing the right thing.*

?

environment → percepts → actions → changes in the environment

# RATIONAL AGENTS

*Doing the right thing.*

?

environment → percepts → actions → changes in the environment

Good changes?

Bad changes?

# RATIONAL AGENTS

*Doing the right thing.*

?

environment → percepts → actions → changes in the environment

Good changes?

Bad changes?

To know what the right action at the right time is, we check if we change **the environment** in the way that we want it to change.

# PERFORMANCE MEASURE& RATIONALITY

A performance measure **evaluates any given sequence of environment states**.

A performance measure is a **choice by the designer**!

A rational agent is an agent that, for each possible percept selects an action that is expected to maximize its performance measure.

# PERFORMANCE MEASURE

**There isn't one fixed performance measure for all tasks, agents and environments.**

Let's imagine we are designing an autonomous vacuum cleaner.

POLL: WHAT COULD BE A GOOD PERFORMANCE MEASURE THAT THE AGENT NEEDS TO MAXIMISE?

# PERFORMANCE MEASURE

**There isn't one fixed performance measure for all tasks, agents and environments.**

Let's imagine we are designing an autonomous vacuum cleaner.

POLL: WHAT COULD BE A GOOD PERFORMANCE MEASURE THAT THE AGENT NEEDS TO MAXIMISE?

As a piece of advice, R&N say: *As a general rule, it is better to design performance measures according to what one actually wants in the environment, rather than according to how one thinks the agent should behave.*

# PERFORMANCE MEASURE

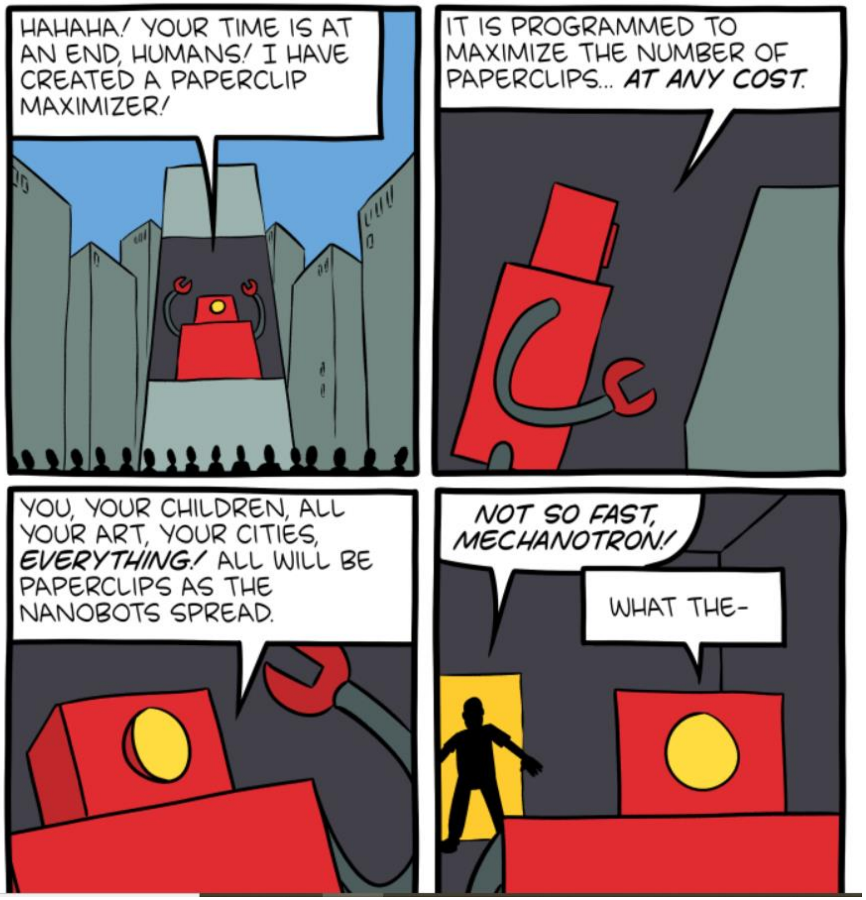
- For the same performance measure, different strategies can be equally effective.

- **Ex: Vacuum at low power all day, versus full power for 5 mins every hour, or at the end of the day.**

- With respect to the amount of clean square meters performance measure, both these strategies are equally good.

- We can of course add elements to our measure to distinguish strategies that we prefer, like electricity consumption.

# PERFORMANCE MEASURE




Source and rest of the story: https://www.smbc-comics.com/comic/paperclip

Based on the paperclip maximiser *thought experiment* by Nick Bostrom: https://en.wikipedia.org/wiki/Instrumental_convergence

# RATIONAL AGENTS

*Doing the right thing.*

A **rational agent** is an agent that, for each possible percept selects an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has.

# RATIONAL AGENTS

*Doing the right thing.*

A rational agent is an agent that, for each possible percept, selects an **action** that is expected to maximize its **performance measure**, given the evidence provided by the **percept sequence** and whatever **built-in knowledge** the agent has.

# RATIONAL AGENTS

*Doing the right thing.*

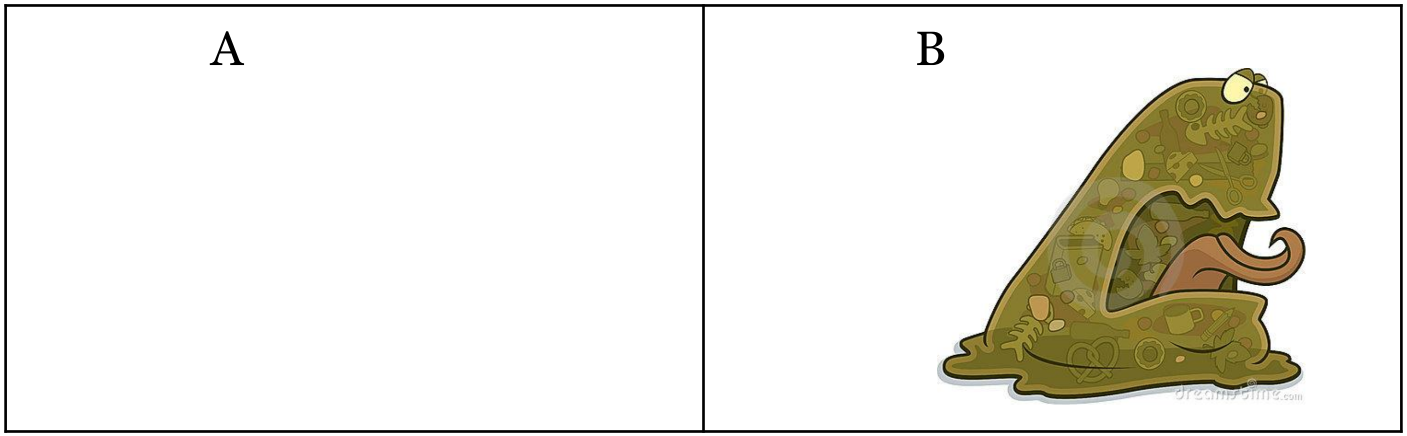
A rational agent is an agent that, for each possible percept, selects an **action** that is expected to maximize its **performance measure**, given the evidence provided by the **percept sequence** and whatever **built-in knowledge** the agent has.

The **rationality** of an agent depends on:

1. Possible actions
2. Performance measure
3. Percept sequence
4. Built-in knowledge

# RATIONAL AGENTS

## IS THE AGENT GIVEN BY THE PROGRAM BELOW RATIONAL?




The agent doesn't know in which square it will be placed or where the dirt is.

| Percept | Action |
|---|---|
| A, Clean | Move Right |
| A, Dirty | Clean |
| B, Clean | Move Left |
| B, Dirty | Clean |

**Performance measure:** Number of clean squares
**Knowledge:** Clean squares stay clean.

# RATIONAL AGENTS

## IS THE AGENT GIVEN BY THE PROGRAM BELOW RATIONAL?

A

B


The agent doesn't know which square it will be placed in and where the dirt is.

| Percept | Action |
|---|---|
| A, Clean | Move Right |
| A, Dirty | Clean |
| B, Clean | Move Left |
| B, Dirty | Clean |

**Performance measure:** Number of clean squares

**Knowledge: A** is never dirty.

# RATIONALITY IS NOT OMNISCIENCE

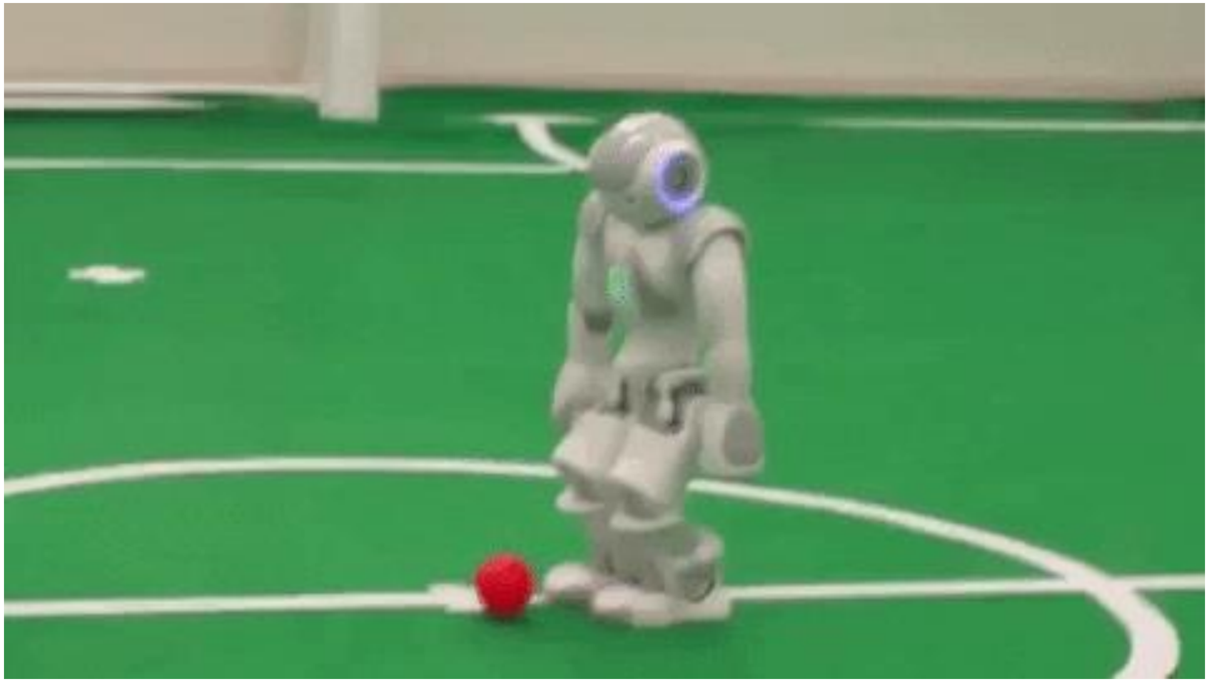
- Rationality depends on the **agent's knowledge.**

- **In most environments, it is imposible to know everything!**

- Rationality is about maximising **expected** performance.

**Gathering information** and **learning** are part of rationality: they change the agents knowledge and can maximise expected performance by adjusting the expectations to reality.

# 3. TYPES OF TASK ENVIRONMENTS



# AGENTS AND THEIR ENVIRONMENT

- The environment in which an agent **acts** affects how we should design it.
- There are a number of concepts that we have defined today that are related to the environment:
  - An agent perceives the environment through **sensors**, and acts on it through **actuators**.
  - The changes in the environment are evaluated with a **performance measure**.

- We call this collection the **task environment**, and it can be described by

**PEAS** (**P**erformance, **E**nvironment, **A**ctuators, **S**ensors)

# TYPES OF TASK ENVIRONMENT

Task environments are hugely varied (since they depend on the deployment context and the agent itself).

Still, we can look at some **properties**:

1. Partly vs. fully observable
2. Static vs. dynamic
3. Discrete vs. continuous
4. Episodic vs. sequential
5. Deterministic vs. stochastic
6. Single vs. multi agent
7. Known vs. unknown

# FULLY VS PARTIALLY OBSERVABLE

- A task environment is fully observable if the  agents' sensors detect all aspects that are **relevant** to the choice of action.
    - Depends on the **sensors,** the **environment** and the **performance measure** (what is *relevant*?).
    - *Eg. A thermostat.*


- Otherwise, it is **partially observable**.
    - *Eg. An autonomous vehicle (cannot see objects behind a corner).*

- If there are no sensors, the task environment is **unobservable**.

# KNOWN VS UNKNOWN

- This is a property of **the agent**, rather than of the environment.

- In a **known** environment, the *laws* of the environment are part of the knowledge of the agent.
  - The outcomes or outcome probabilities (*how an action will modify the environment*) are known.
  - This doesn't mean the environment is fully observable (*card game*).

- In an **unknown environment,** the agent does not know the outcome of its actions.
  - *Eg. A game for which the agent does not know the rules.*

# STATIC VS DYNAMIC

- If the environment can change while the agent is deliberating (deciding on an action), it is **dynamic.**
    - *Eg. A cooking robot, where the food can burn while deciding whether to switch off the heat or not.*

Source: https://www.nytimes.com/2010/02/24/dining/24robots.html

- If the world does not change, but the performance score does, the environment is **semidynamic.**
    - *Eg. When the performance measure depends on the time it takes to complete an action.*

- If the agent can take a percept and produce a response without considering the time it takes to produce that response, the environment is **static.**
    - *Eg. A sudoku-solving agent.*

# DISCRETE VS CONTINUOUS

- **Discrete** environments can be divided into a finite number of distinct states.
    - *E.g. A sudoku grid has a finite number of possible states; the vacuum cleaning example from before where the states could be summarized in a table.*
- **Continuous** environments cannot be divided into separate states.
    - The environment may change continuously (*weather conditions*), the agent may be receiving percepts continuously (*infrared sensor*) and/or acting in a continuous manner (*driving*).

- Many environments that are actually continuous are treated as discrete, and viceversa.
    - *E.g. The physical chess board, where time passes continuously.*
    - *Eg. Input from digital cameras is discrete but can be modeled as a continuous smooth input.*

# EPISODIC VS SEQUENTIAL

- An **episodic** task environment is composed of **atomic** (individual) **episodes**.
  - In each **episode**, the agent receives a percept and performs an action in response.
  - What happens in an episode does not affect the next ones.
  - Each episode could be discrete or continuous.
  - *Eg. An agent looking for defective parts in manufacturing.*

- In a **sequential** environment, a decision can affect future decisions.
  - The agent may still ignore previous states in its decision (reactive agent).
  - Can be both continuous (*driving*) and discrete time (*chess*).

# DETERMINISTIC VS STOCHASTIC

- An environment where the next state can be completely determined based on the previous state and the agent's action, is **deterministic**.
  - *Eg. A sudoku-solving agent, the vacuum robot from previous example (because clean squares remained clean).*

- If an environment is not deterministic, it's *stochastic*.
  - Stochastic environments can't be perfectly predicted, but can still be modeled using models that include uncertainty
  - Note that stochastic is not the same as random, a stochastic process depends both on the previous state and a random element
  - *Eg. An autonomous vehicle (pedestrians modify the environment unpredictably).*

# SINGLE VS MULTI AGENT

- In a **single agent** environment, there is a single agent. In a **multi-agent** environment, there are several agents. But... what counts as another agent in the environment?

- If the other entity has a performance measure that **depends on the actions of the first agent**, we are in a multi-agent environment.
  - *Eg. A card game, a fleet of cleaning robots.*
  - **cooperative** (actions can increase all agents' performance), **competitive** (all performances cannot be increased at the same time), or a combination of both.

- Designing multi-agent systems is a whole field by itself.
  - Communication between agents, organizational structures, decision-making processes...
  - Designing a single agent that will operate in a multi-agent environment is also a different challenge: will it need to communicate? Predict other agents' actions?

# IN SUMMARY

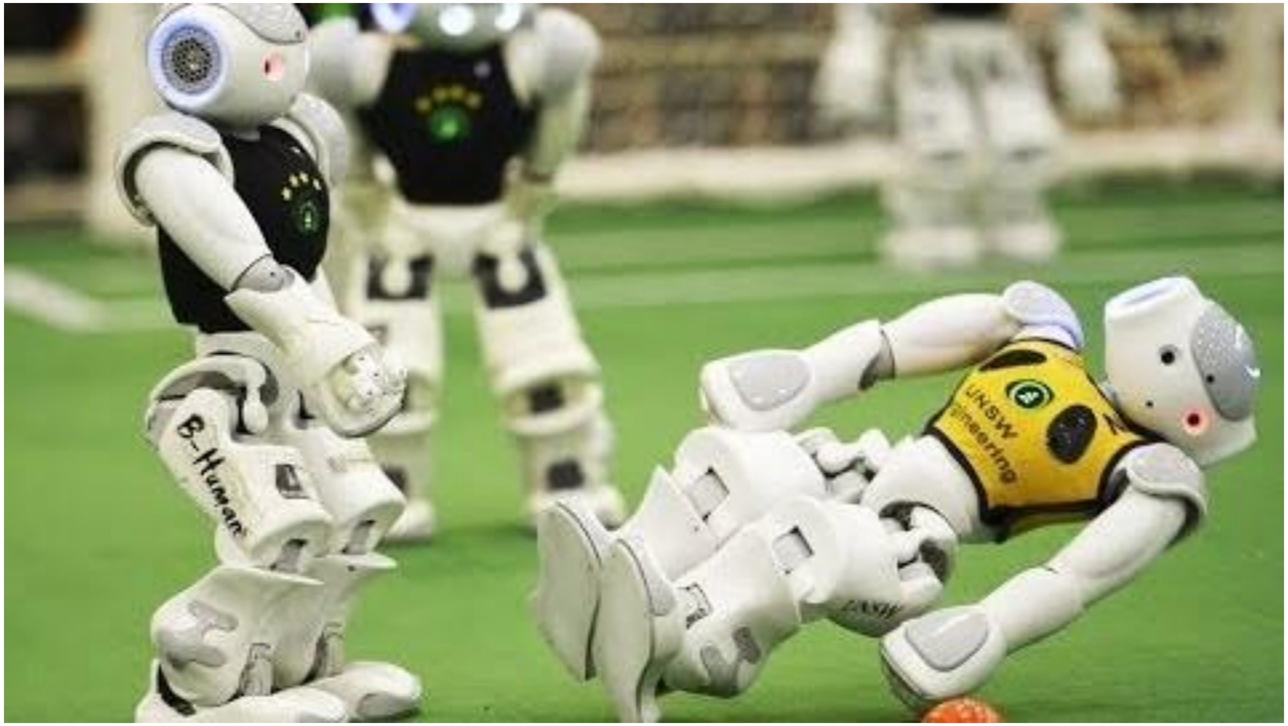
- The task environment is not a description of the environment for a robot, it is **a way to describe the task, or problem, to be solved.**

POLL: What are the characteristics of the following task environment?

A football playing robot.

- Performance measure: Winning with its team.
- Environment: A football field, playing with other robots.
- Acting: Can move and shoot the ball.
- Sensing: Can see the ball, the lines on the field, the other robots, and the goalposts.

# IN SUMMARY




# BREAK

UMEÅ UNIVERSITY

# 4.TYPES OF AGENTS


UMEÅ UNIVERSITY

# AGENT TYPES

1. Simple vs Model-based
2. Reactive
3. Goal based
4. Utility based
5. Learning agents

This classification showcases different ways to design and implement intelligent agents.

# AGENT TYPES

- At the begining of the lecture, we described a vacuum agent with a **table** that contains every posible state of the world, and the corresponding action from the agent.

- This is not possible to do in most cases! The table would be huge.
    - No way to store it.
    - A long time to look it up.
    - A long time to complete it by the designer.
    - A complete knowledge of the state space is needed.

- Effective agent programs also account for **complexity and efficiency**, i.e. they are meant to produce rational agents from a small program.

# SIMPLE & MODEL-BASED AGENTS

- **Simple** agents do not store an internal representation of the state of the world. They react exclusively to the percepts.
  - Great for fully observable scenarios.
  - Eg. thermostat

- **Model-based** agents keep an internal representation of the state of the world, and use it together with the percept to determine the correct action.

- To determine ("best guess") the state of the world, they are endowed with a **model**: information of how the environment evolves around the agent+information of how the agents' actions modify the world.
  - Great for known, partially observable environments.

# REACTIVE AGENTS

- **Reactive** agents select actions on the basis of the *current* percept, ignoring the rest of the percept history.
  - *Eg. The vacuum robot from earlier. Just needs to know if its current square is clean or dirty in order to choose its action.*
  - They do not need memory.
  - Can be described by if-then rules, but in practice there are more sophisticated implementations.

- Pros: Simple, transparent, effective in dynamic, known environments.

# GOAL-BASED AGENTS

- **Goal-based** agents choose their action in order to **achieve a specific goal.**


- Pros: Adaptation to the environment, no need to anticipate all the scenarios, good for dynamic environments.

# GOAL-BASED AGENTS

- When a goal needs a sequence of actions to be achieved in a dynamic environment, it is necessary to deal with hypothetical scenarios: *what will happen if I do this-then-that?*
    - Often, this means that the agent **models** both the environment and the goal.
    - There are also other techniques where the environment is not modeled, only how to act in order to achieve a specific goal.

- Choosing a goal that will maximise the performance measure is also sometimes part of the problem.

- The research areas related to choosing the right actions to execute a goal are those of **search** and **planning**.

# UTILITY-BASED AGENTS

- Sometimes, we cannot capture with a single goal all of the behaviours we want from the agent.
    - For an autonomous vehicle we may want to balance safety, speed to reach destination, electricity consumption, passenger satisfaction…

- **Utility-based** agents choose the action or sequence of actions that are expected to maximise a **utility function.**

- **Utility functions** assign a numerical value to each state of the environment, so that a state the satisfies more preferences has a higher value.

# UTILITY-BASED AGENTS

- Building a utility function can be hard!
  - Can account for different weights between all the factors (more important to not hit pedestrians than to drive fast), but we have to understand what is the right balance.
  - Describing preferences in a way that a variable/number can be assigned is difficult.
  - The agent needs to have a good model of the environment to predict the expected utility of its actions.

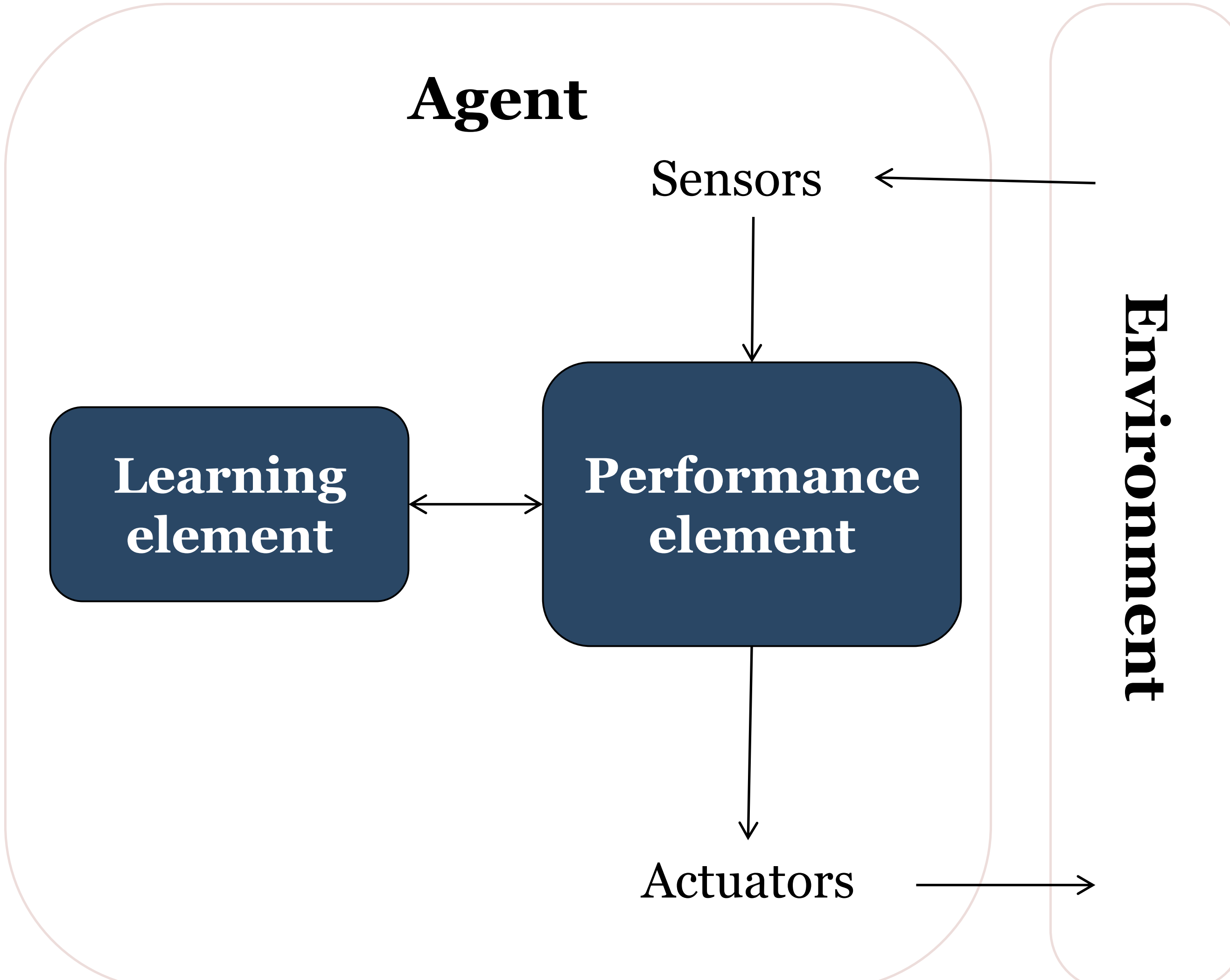
- **Pros:** can capture complex drives, good for unknown environments.

# PERFORMANCE MEASURE, UTILITY AND GOALS

- Notice that both the performance measure and the utility function describe what are good **changes in the environment**.
  - The performance measure is an **external evaluation** of the agent, whereas the utility function is an **internal representation** in the agent.
  - If the utility function is the same as the performance measure, then a utility-based agent will be rational by definition.

- For agents to be rational, the choice of actions, goals or utility function need to be **aligned with the performance measure**.

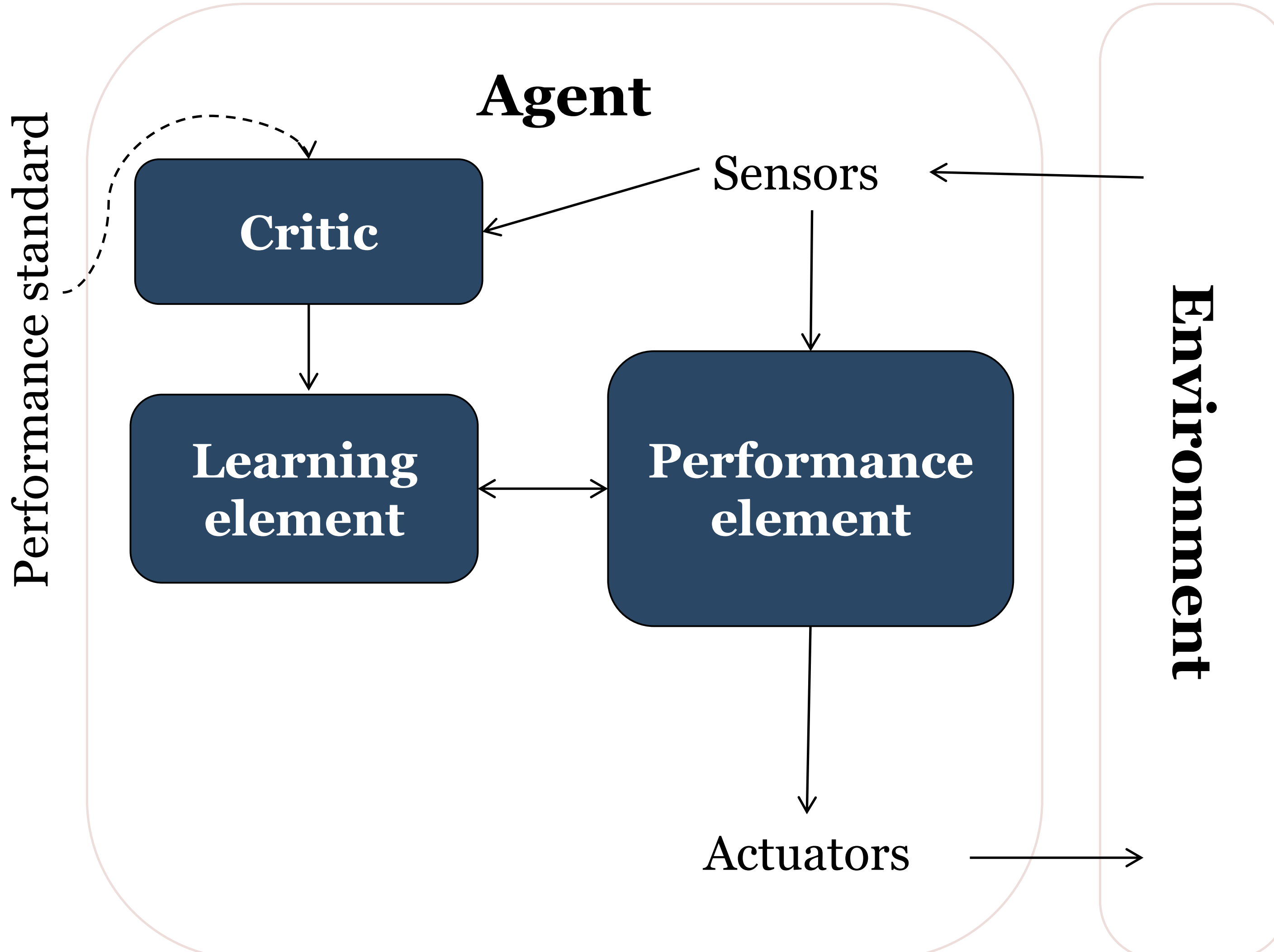
# LEARNING AGENTS






R&N p. 55

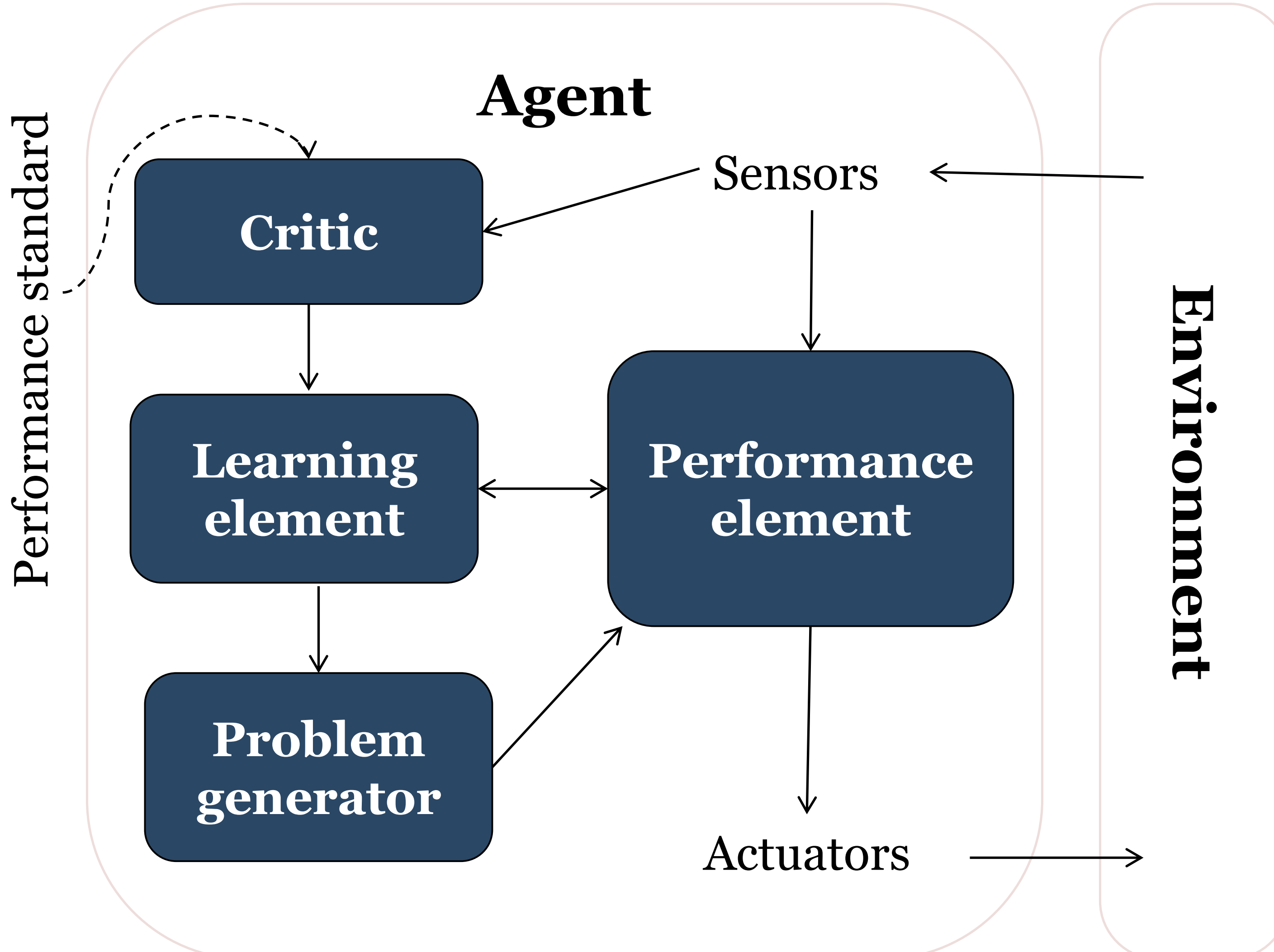
# LEARNING AGENTS






R&N p. 55

# LEARNING AGENTS






R&N p. 55

# WHAT TYPE OF AGENT DO YOU THINK OF WHEN YOU THINK ABOUT AI?

# LEARNING AGENTS

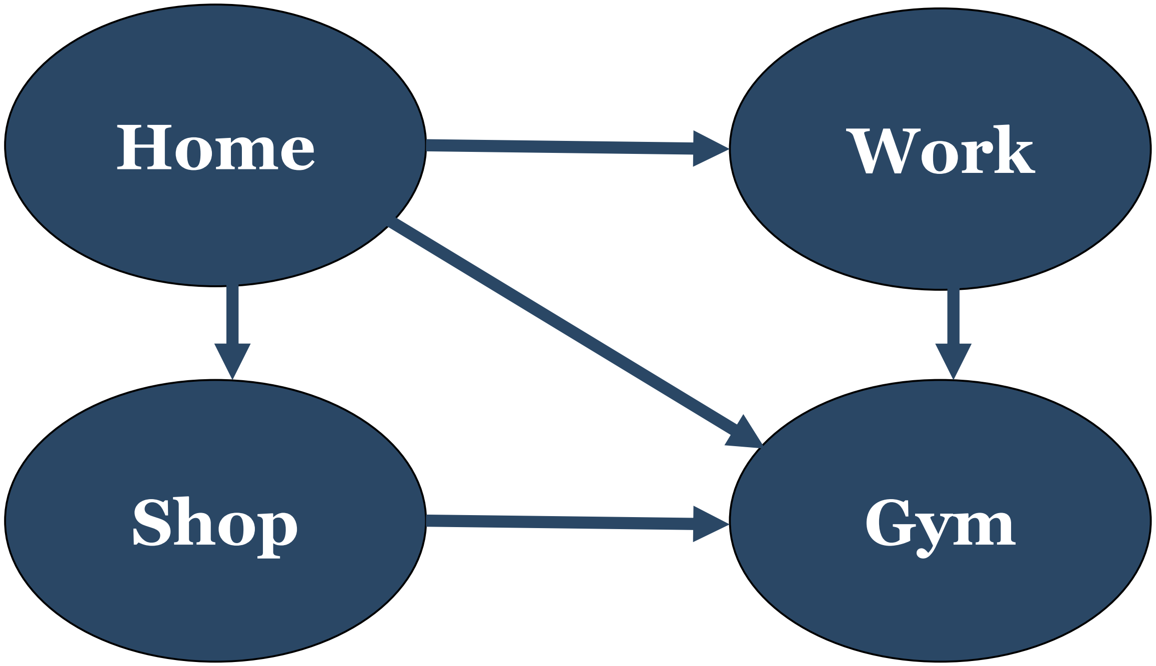
- There are many types of learning agents, but not all of AI is about learning agents.
  - The critic component has to be aligned with the performance measure, which can be hard to implement.
  - Pros: Great for unknown environments, and for task environments with very big state space.

- Agents purely based on reasoning or planning with a full model of the world are also part of AI.

# 5. REPRESENTING THE WORLD

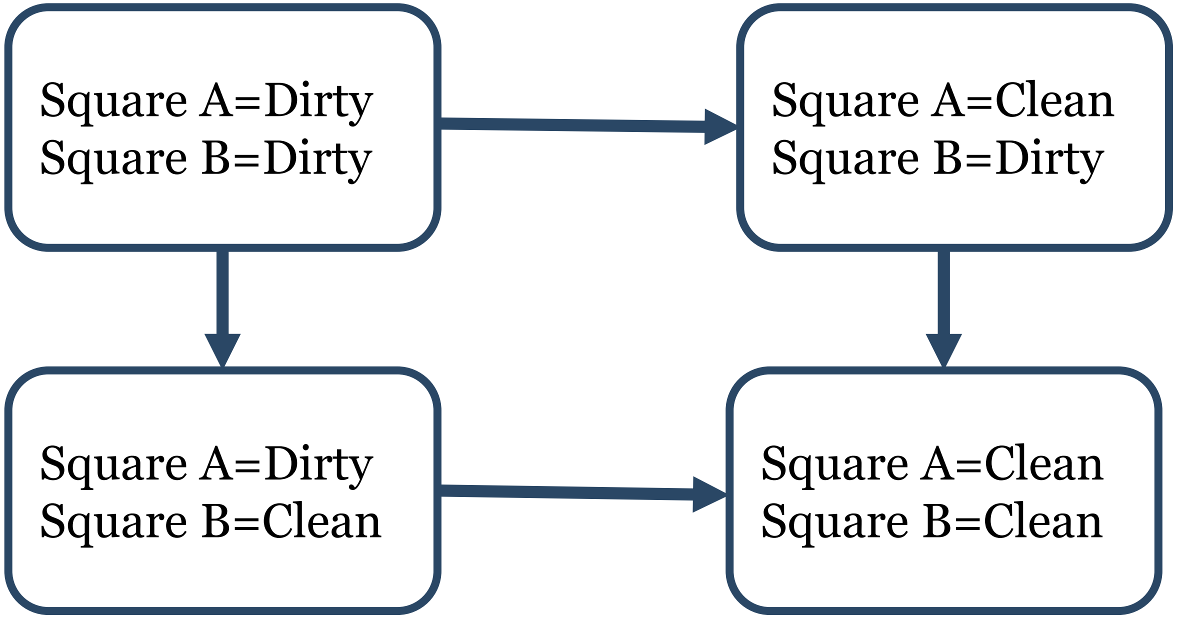

Credit: Disney

# ATOMIC REPRESENTATIONS



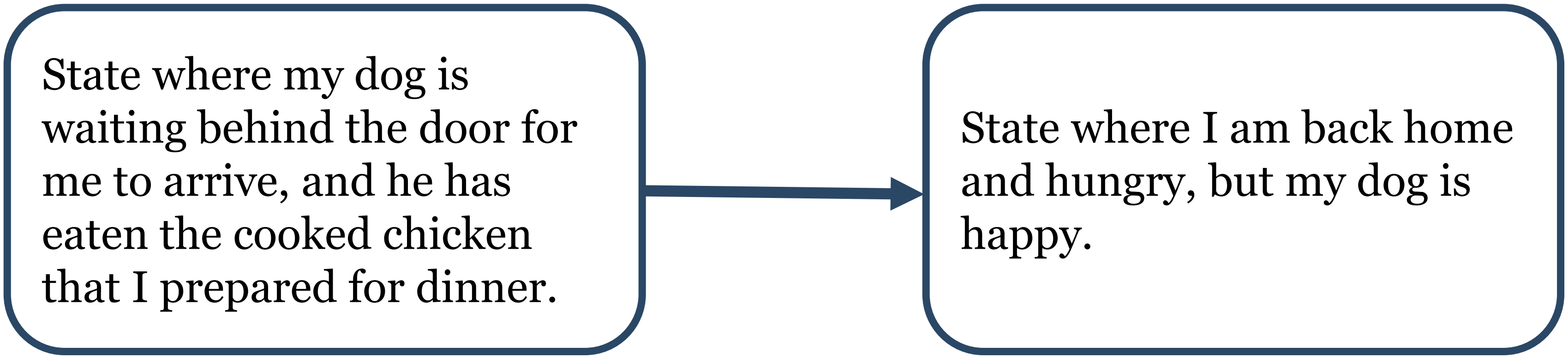

- Each state is **a discrete single entity**, we know when it is posible to transition between two states.
- There are **no sub-entities** describing each state, it is just a point in a graph.

# FACTORED REPRESENTATION




- States are described by a fixed set of **variables** or **attributes**, each of which can have a **value**.
- Easier to see how to transition from one state to another (i.e. which attributes need to be changed in order to reach the new state).
- We can represent uncertainty by leaving an attribute blank.

# STRUCTURED REPRESENTATION




- Each state can be described in a model, where we can represent **objects and relationships between them** (e.g like with natural language, or with first order logic).
- This representation is more detailed than assigning values to atributes. We do not need to describe every state with the same atributes, only with the same language.

# REPRESENTATIONS

- We say that a representation in which we can capture more details is more **expressive.**

- A more expressive representation can capture everything a less expressive representation can, and add more details.

# Thank you!

UMEÅ UNIVERSITY