UNIVERSITÄT DES SAARLANDES

Artificial Intelligence

Intelligent Agents

Podcasts are available with these slides:
Login to the CMS, go to
Information > Materials > Lecture Podcasts

Prof. Dr. habil. Jana Koehler

Summer 2020

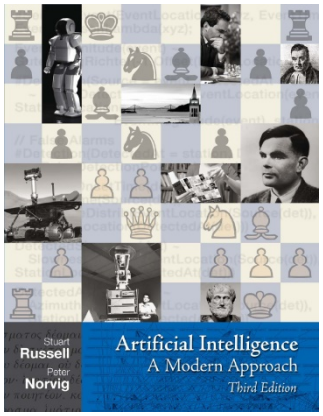
Agenda

- The Rational Agent
  - perception – cognition – action
- Properties of Environments
- Types of Agents
  - **Simple Reflex** agents respond immediately to percepts
  - **Model-based Reflex** agents are aware of action effects
  - **Goal-based** agents work towards goals
  - **Utility-based** agents try to maximize their reward
  - **Learning** agents improve their behavior over time

Recommended Reading

- AIMA Chapter 2: Intelligent Agents

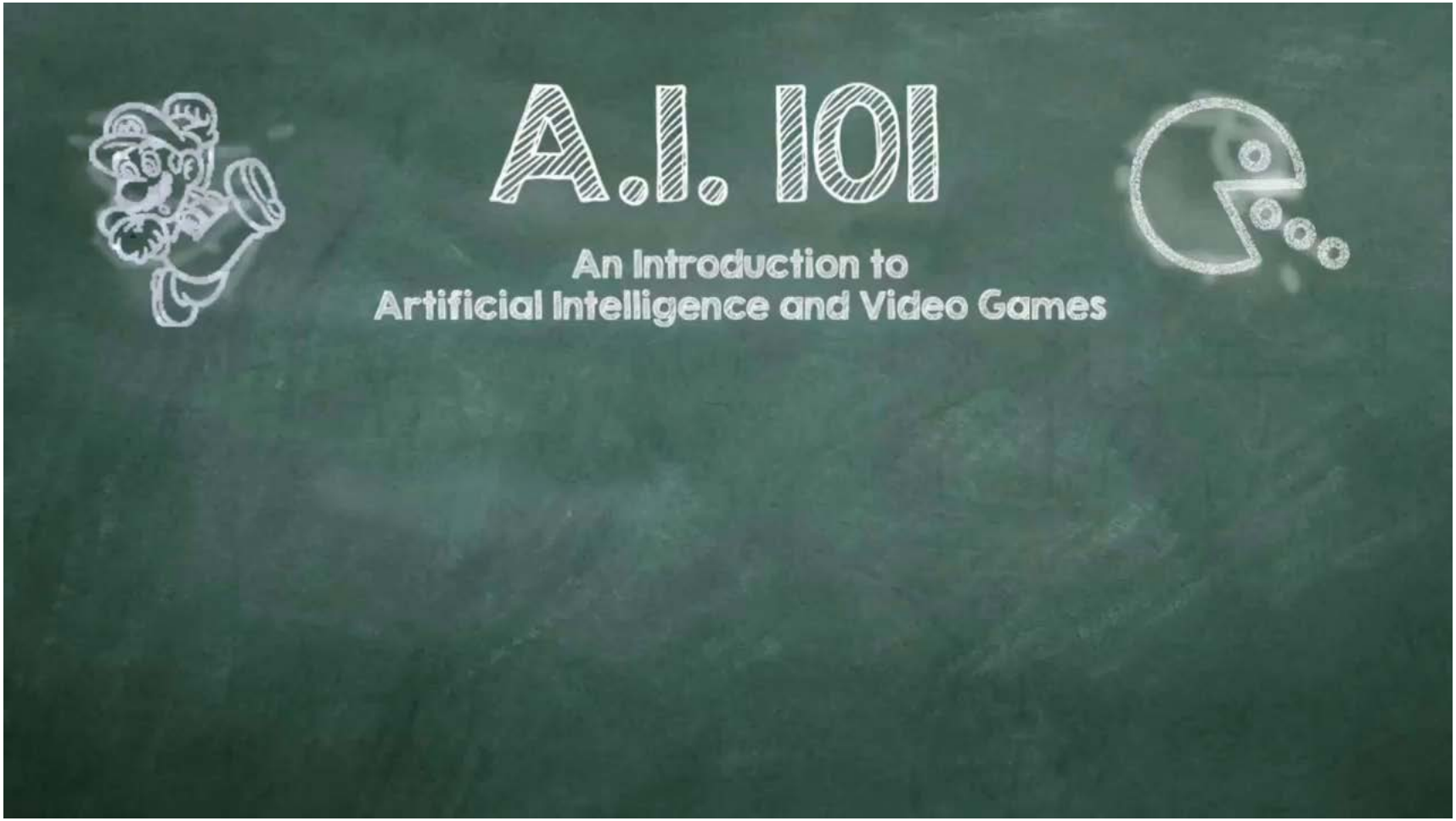
Metaphor of the Rational Agent



A central aspect of intelligence is the ability
to
act successfully in the world

- **What does it need for an agent to act at all?**
- **How can the agent act successfully?**

What is an Agent?



Watch video in CMS: Information > Materials > Videos
Youtube version was removed recently https://www.youtube.com/watch?v=IK7l4ZLm55I

Agent

1) Ability to perceive environment

2) Perceptions are used to make Decisions

3) Decisions will result in actions

If the agent is rational, then

4) Decisions must be RATIONAL
   - ➢ Must lead to best possible action the agent can take

The Rational Agent

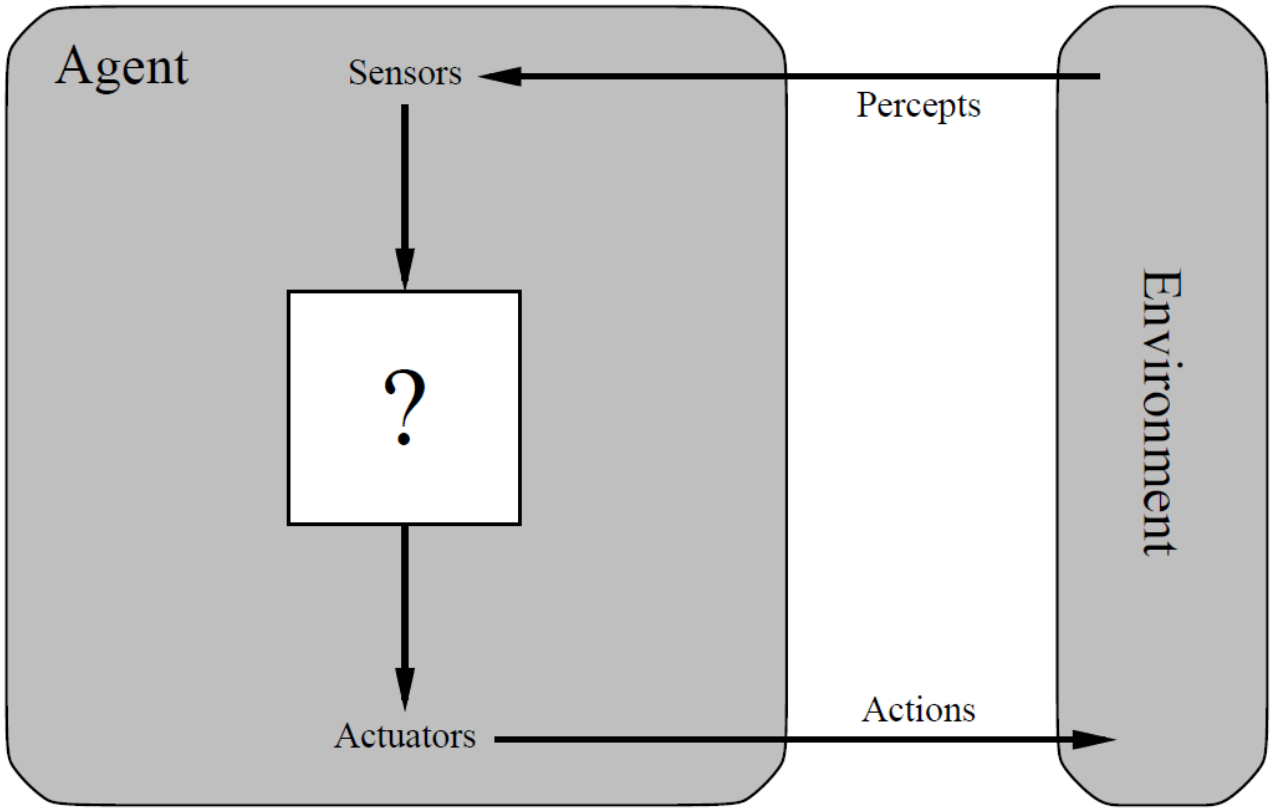
- AGENT = something that acts (latin *agere, do, make …)*
- RATIONAL AGENT = *acts so as to achieve the best expected outcome*
- Rational thinking is one possible mechanism to achieve rational behavior
- Perfect rationality cannot be achieved in complex environments
  - *LIMITED RATIONALITY = acting appropriately*

Rationality vs. Omniscience

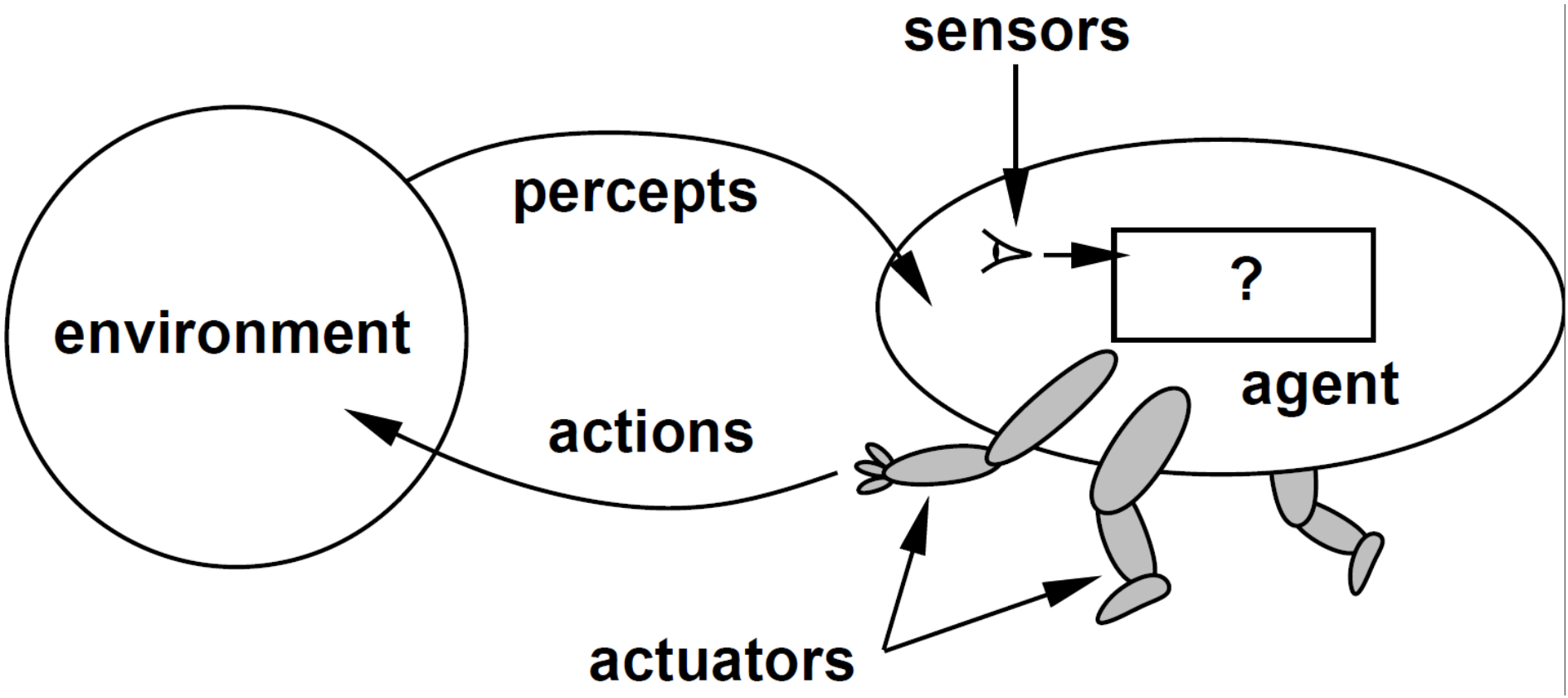
- An omniscient agent knows the actual effects of its actions
- In comparison, a rational agent behaves according to its percepts and knowledge and attempts to maximize the expected performance
- Example: If I look both ways before crossing the street, and then as I cross I am hit by a meteorite, I can hardly be accused of lacking rationality
- **Rationality maximizes *expected* performance**, perfection maximizes *actual* performance based on its knowledge about the environment

Agent

- Perceive the environment through sensors (= percepts)
- Act upon the environment through actuators (= actions)



A Simplified Human Agent



Actuators and Sensors in Pepper

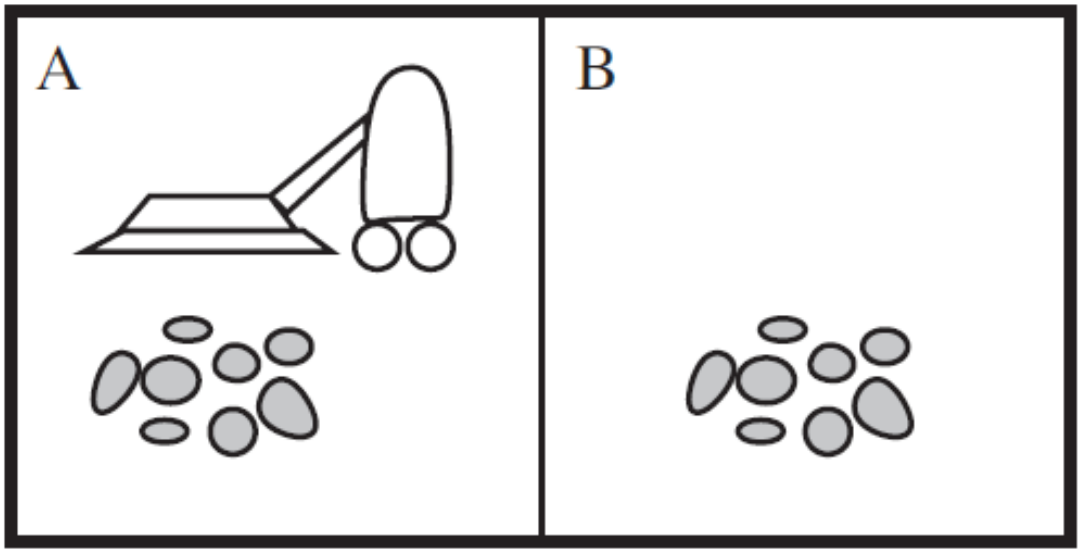
HARDWARE OVERVIEW

BayTrail processor
Wifi / Ethernet
20 motors
120cm / 4 feet
28kg / 62 lb
12h battery

3x Tactile sensor
4x Microphones
2x 3D Sensors
2x Loudspeakers
2x 2D Cameras
Gyroscope
Tablet
2x Tactile sensors
2x Lasers
Sonar
3x Safety bumpers

#MeetPepper

The Vacuum Cleaner Agent



- Percepts
  - Am I in square A or B?
  - Is the square dirty?
- Actions
  - move left, move right
  - suck
  - do nothing

Modeling the Agent

- **Percept sequence**
  - complete history of what the agent has perceived to date
- **Agent function**
  - a function that maps any given percept sequence to an action

function TABLE-DRIVEN-AGENT(*percept*) **returns** an action
  persistent: *percepts*, a sequence, initially empty
    table, a table of actions, indexed by percept sequences, initially fully specified

  append *percept* to the end of *percepts*
  action ← LOOKUP(*percepts*, *table*)
  return *action*

- **Agent program**
  - takes the current percept as input from the sensors and returns an action to the actuators

An Agent Table

percept sequence	action
[A, clean]	right
[A, dirty]	suck
[B, clean]	left
[B, dirty]	suck
[A, clean], [A, clean]	right
[B, clean], [B,clean]	left
…	…

"If the current square is dirty, then suck else move to the other square."

Performance of Rational Agents

- . . . do the "right thing"!

➢ In order to evaluate their performance, we have to define a performance measure

- Vacuum cleaner agent
  - m^2 per hour, level of cleanliness, energy usage, noise level, safety (behavior towards hamsters/small children)

➢ Optimal behavior is often unattainable
  - Not all relevant information is perceivable
  - Complexity of the problem is too high

The Performance Measure

- Each action of the agent takes the world to another state

- If the sequence of world states is desirable for an external observer, the agent has performed well

- The performance measure evaluates the STATES of the ENVIRONMENT independent of the AGENT!
  - otherwise an agent could achieve perfect rationality simply by deluding itself that its performance was perfect
  - You get the behavior you reward
  - Vacuum cleaner: amount of dirt collected – suck, release, suck, release, suck

Ideal Rational Agent

- Rational behavior is dependent on
  - performance measures (goals)
  - percept sequences
  - knowledge of the environment
  - possible actions

Ideal 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 whatever *built-in knowledge* the agent has.

PEAS Descriptions of Agents in Environments

Performance Measure	Number of correctly answered questions
Environment	Rigi Kaltbad Station Shop
Actuators	loud speaker
Sensors	camera, microphones

Other PEAS Examples

Agent Type	Performance Measure	Environment	Actuators	Sensors
Medical diagnosis system	healthy patient, costs, lawsuits	patient, hospital, stuff	display questions, tests, diagnoses, treatments, referrals	keyboard entry of symptoms, findings, patient's answers
Satellite image analysis system	correct image categorization	downlink from orbiting satellite	display categorization of scene	color pixel arrays
Part-picking robot	percentage of parts in correct bins	conveyor belt with parts, bins	jointed arm and hand	camera, joint angle sensors
Refinery controller	purity, yield, safety	refinery, operators	valves pumps, heaters displays	temperature, pressure, chemical sensors
Interactive English tutor	student's score on test	set of students, testing agency	display exercises, suggestions, corrections	keyboard entry

Utility Function of Agent

- The utility function is used by the agent to evaluate the desirability of a state of the world

- A utility function maps a state (or a sequence of states) onto an evaluation value (usually a real number)

- The agent can use the evaluation
  - to select an action (sequence)
  - to weigh the importance of competing goals

More formally ....

$$\text{Utility} \times \text{Percepts} \times \text{Knowledge} \rightarrow \text{Action}$$

- A selected action is *optimal* if it takes the agent to a state of maximum expected utility given available percepts and knowledge

- The agent is rational if it always chooses optimal actions

Utility Function of Rigibot

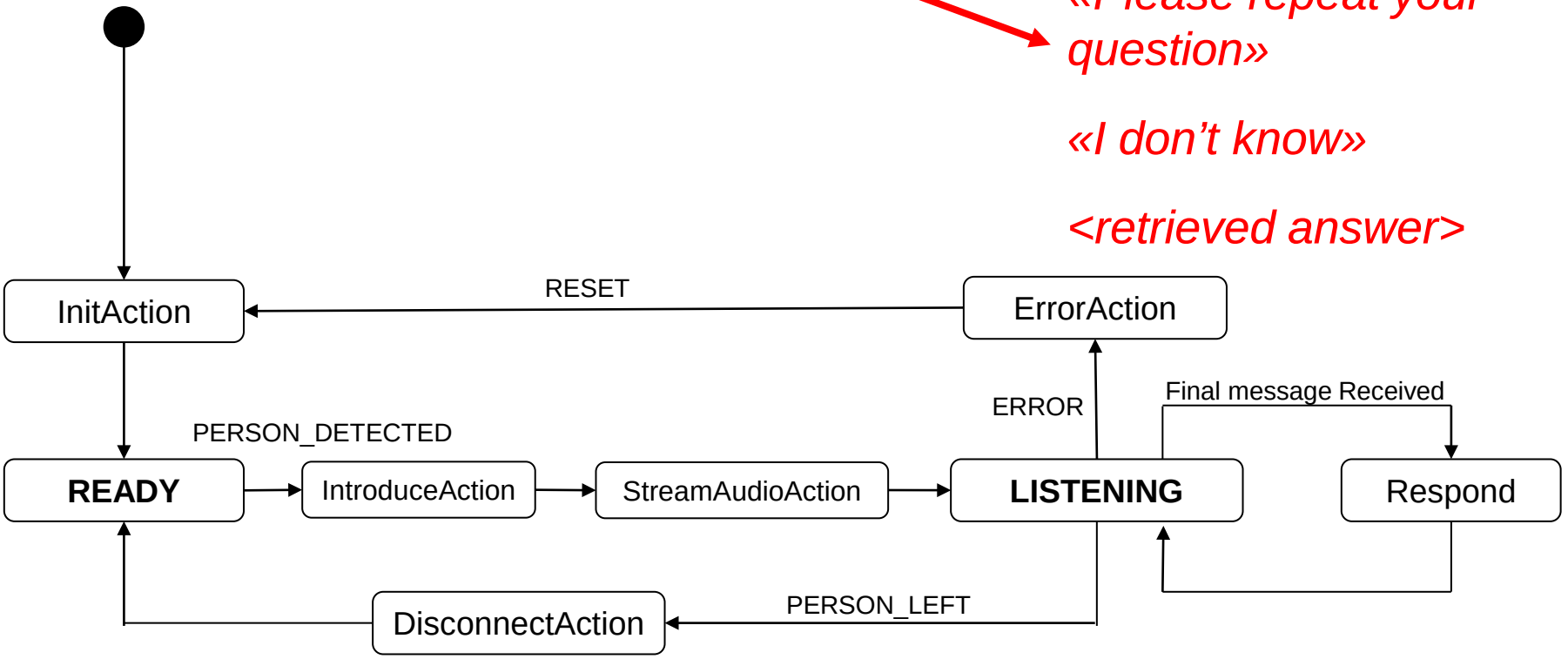
Confidence Values

- Google Speech to Text
- Microsoft QnA Maker

«Please repeat your question»

«I don't know»

<retrieved answer>



Properties of Environments

Environments that are unknown, partially observable, nondeterministic, stochastic, dynamic, continuous, and multi-agent are the most challenging.

Properties of Environments

Question	Yes	No
Does the agent initially have complete **knowledge** about the environment?	Known	Unknown
Can the agent **observe** all relevant aspects of the environment with its sensors?	Accessible	Inaccessible
Is the environment **changing** while the agent is deliberating?	Dynamic	Static
Are time, percepts and actions of the **model** discrete?	Discrete	Continuous

Properties of Agent Actions

Question	Yes	No
Do effects of actions happen as planned by the agent?	Deterministic	Stochastic
Do effects of actions depend only on the current state and the action, but not on the action history?	Episodic[1)]	Sequential[2)]
Can we model the AI system with only one agent?	Single-Agent	Multi-Agent

[1)] episodic memory = short-term memory

- the agent does not need to remember the past to make good decisions

[2)] sequential environments require a long-term memory

- good decisions depend on taking the right action in the past
  → actions have long-term effects
- for example, making a move in a chess game

Simple and Difficult Environments

Property	Simple	Difficult
Knowledge	Known	Unknown
Observability	Accessible	Inaccessible
Dynamics of Changes	Static	Dynamic
Detail of Models	Discrete	Continuous
Short-term Action Effects	Deterministic	Stochastic
Long-term Action Effects	Episodic	Sequential
Number of Agents	Single	Multiple

- ***The key in designing successful AI applications is to understand how we can make environments simpler for the agent!***

Example Environments

- Playing basketball, GO, Robocup soccer
- Moving the king in a chess game
- Controlling the movement of a walking robot
- Drive of an autonomous car
- Breaking on a dry road
- Breaking on an icy road
- Exploring an unkown city
- Throwing a dice
- Classifying an object

Agent Architecture

- Agent = Architecture + Program
- The agent program is implemented on some architecture that determines the components on which the program can run
- The architecture also defines interfaces of the agent tothe environment
  - implement the ability to receive percepts and to execute actions
- In practice, limitations of the architecture (including runtime and memory limitations) force the agent to approximate the rational decision

Types of Agents

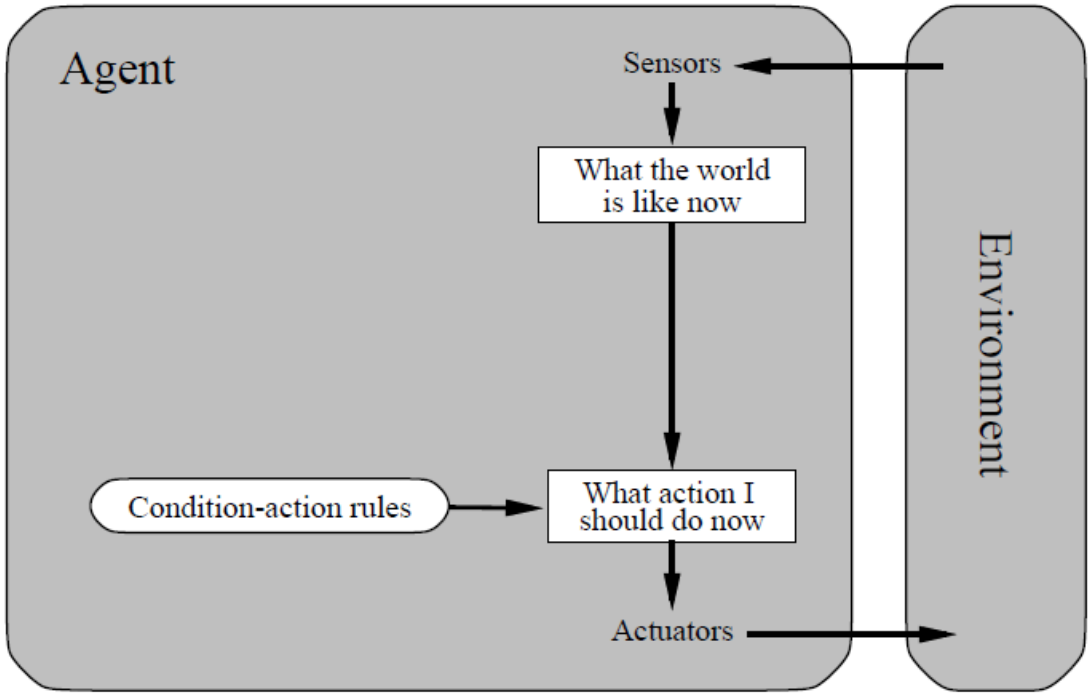
Agents differ in their capabilities

- **Exploration**: explorative actions for information gathering
- **Learning**: as much as possible from the percepts
- **Autonomy**: improve partial or incorrect knowledge

5 Types

- **Simple Reflex Agent**
- **Model-based Reflex Agent**
- **Goal-based Agent**
- **Utility-based Agent**
- **Learning Agent**

Simple Reflex Agent



- Senses the world and responds immediately based on simple rules that interpret the sensor input and link it to actions
  - no explicit world model, no "memory"

Function of the Reflex Agent

function SIMPLE-REFLEX-AGENT(*percept*) **returns** an action
  persistent: *rules*, a set of condition–action rules

  state ← INTERPRET-INPUT(*percept*)
  rule ← RULE-MATCH(*state*, *rules*)
  action ← *rule*.ACTION
  return *action*

Condition-action-rule (productions)

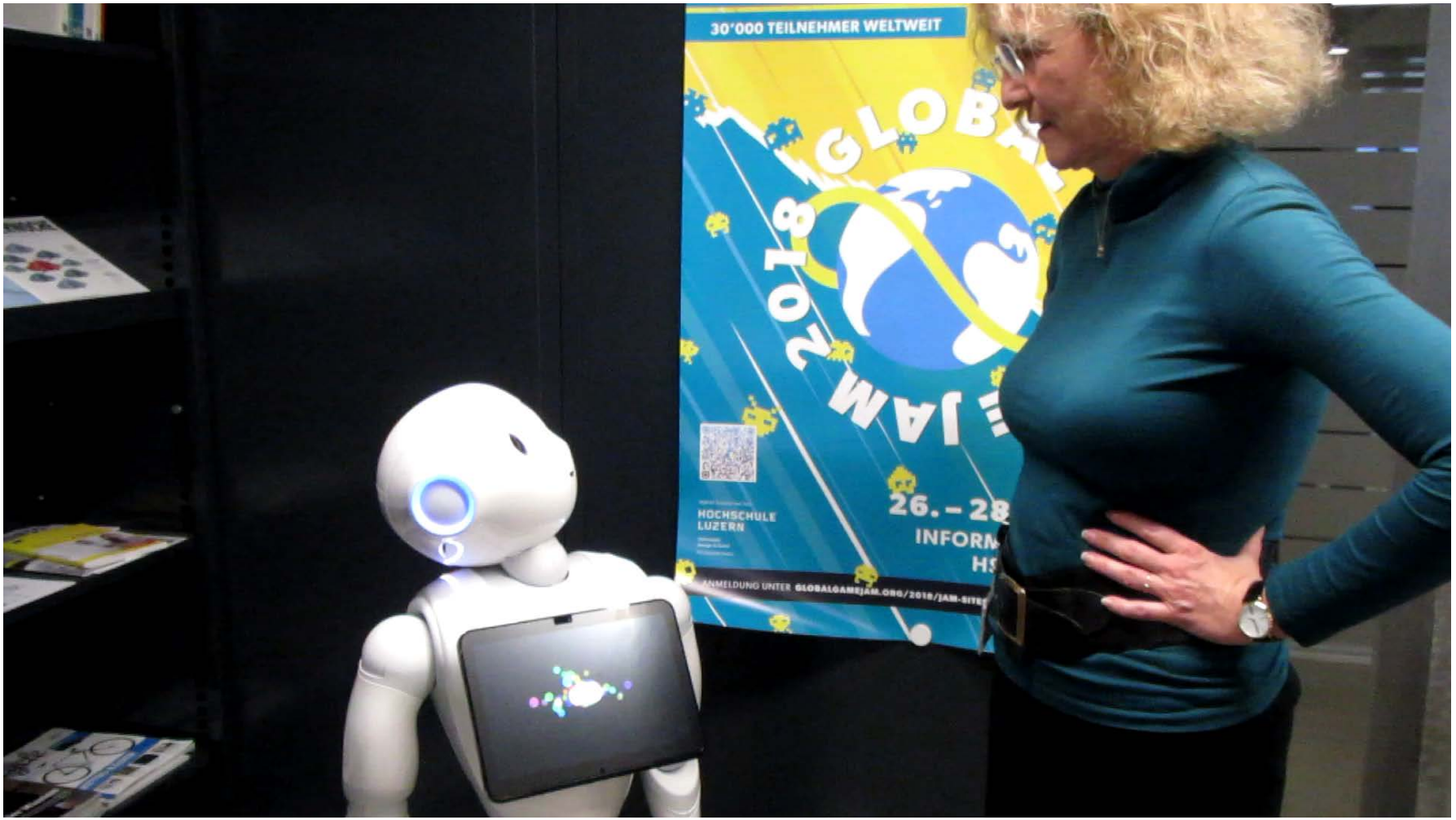
"**If** *car-in-front-is-braking* **then** *initiate-braking*."

When do Simple Reflex Agents work?

- Correct decision is made based on current percepts only
  - environment must be fully observable, otherwise infinite loops can occur
  - escape from infinite loops through randomization

- Vacuum cleaner agent without a location sensor and in a clean room A: moving left will fail forever
- Vacuum cleaner agent with a coin flip to choose a move can randomly escape dead-end situations

Simple Reflexes in Pepper



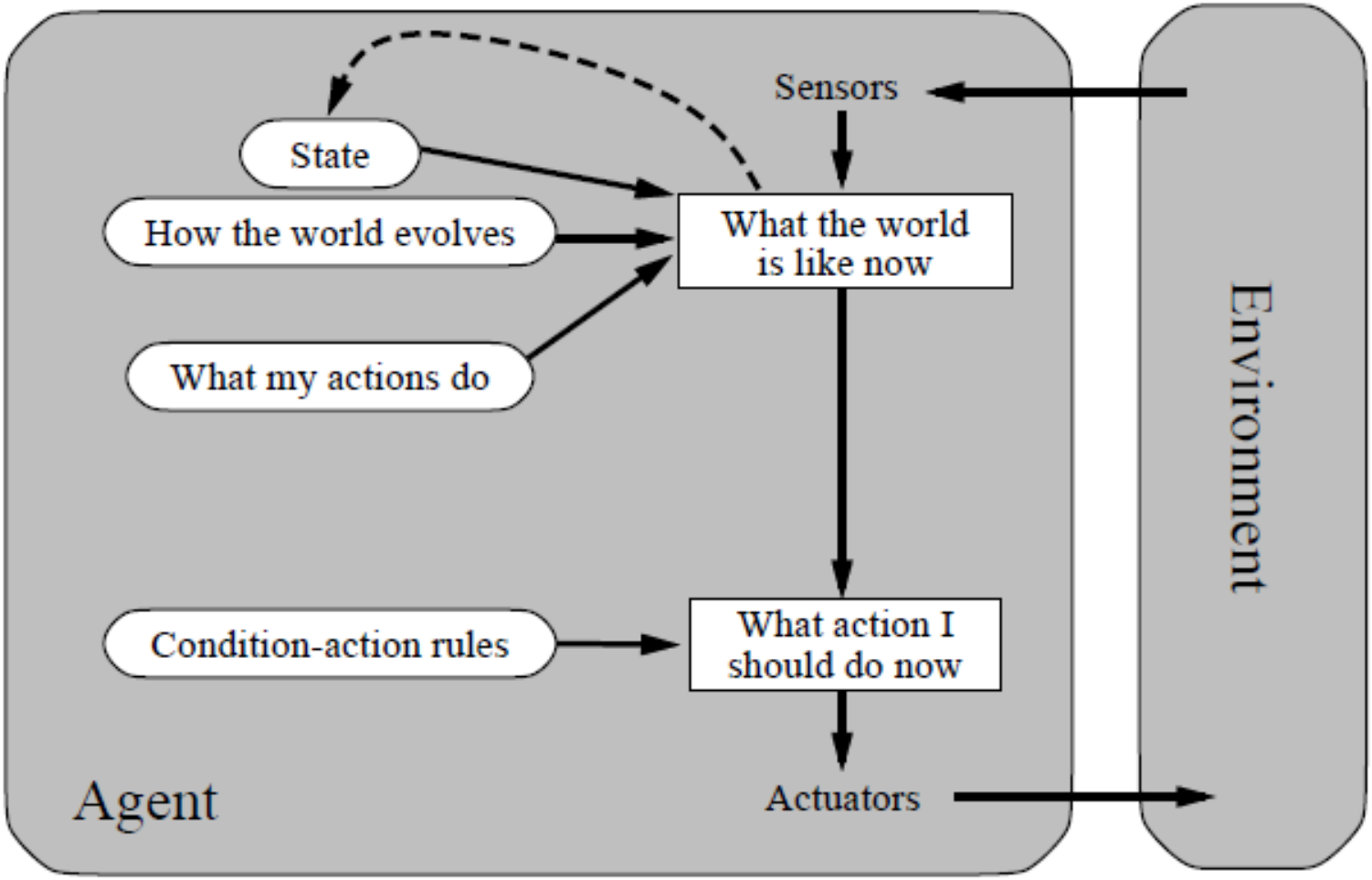
Watch videos about example agents in the CMS: Information > Materials > Videos

Model-based Reflex Agent

- Keep part of the world the agent cannot perceive right now
  - internal world model (agent state) that depends on the agent's percept history
- Model can serve to answer questions
  - what are the effects of agent actions?
  - how does the world evolve (independently of the agent)?

- Uncertainty about the world state is unavoidable because of limited sensing capabilities and limited world models
  - model represents the agent's "best guess“ of the world state, the evolution of the world and the effects of its actions

Model-based Reflex Agent



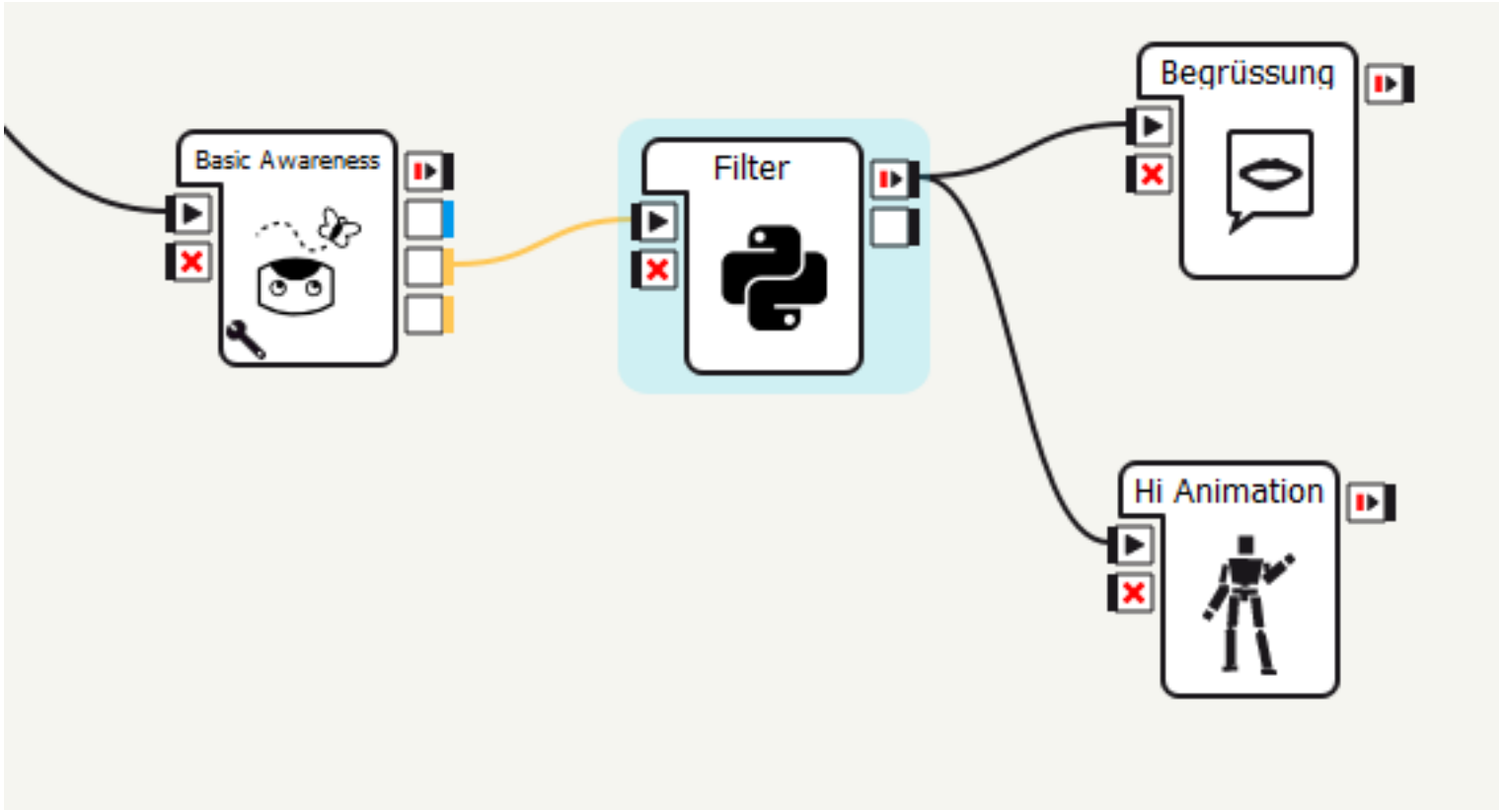
Function of the Model-based Agent

function MODEL-BASED-REFLEX-AGENT(*percept*) **returns** an action

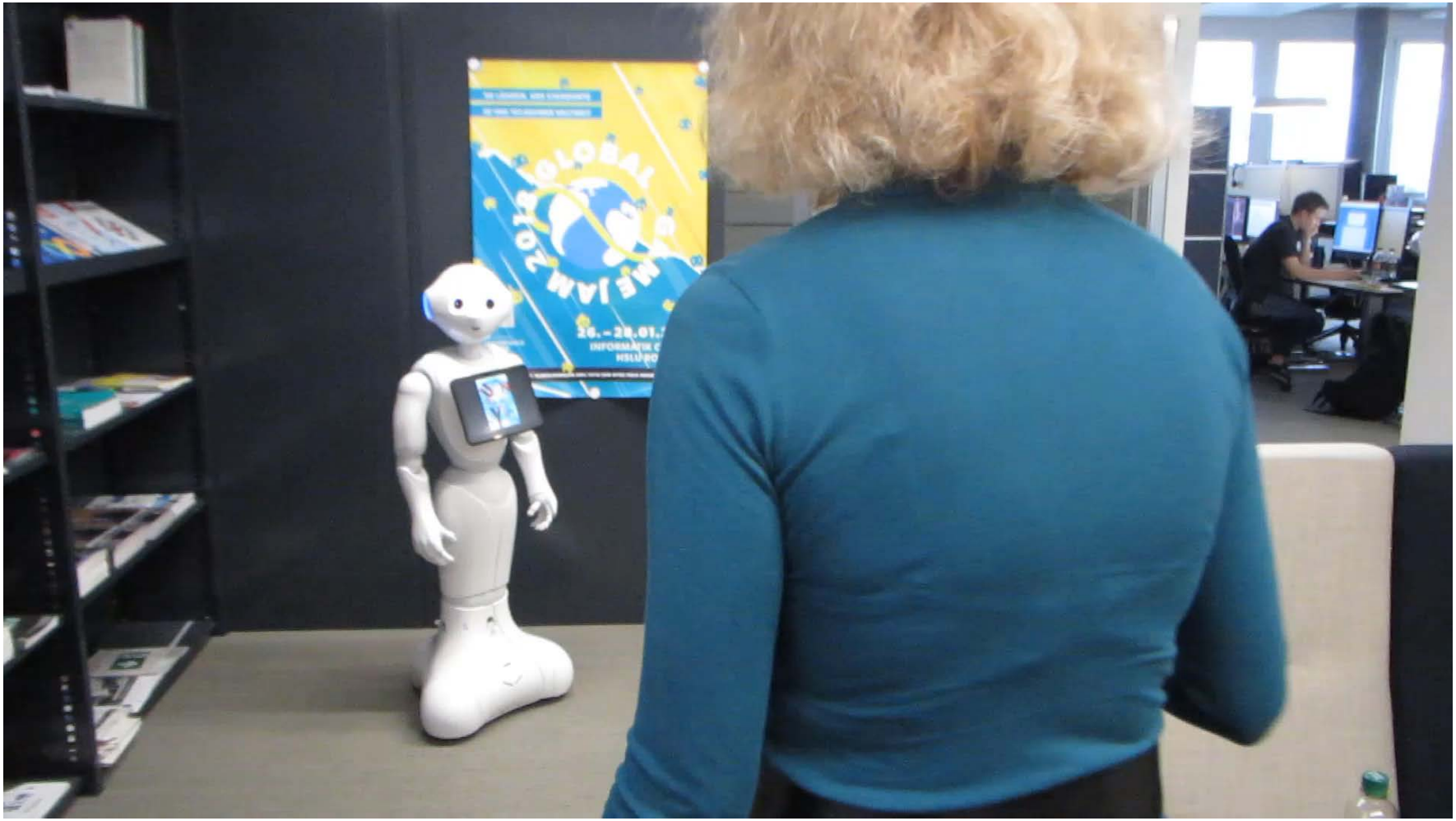
persistent: *state*, the agent's current conception of the world state
model, a description of how the next state depends on current state and action
rules, a set of condition–action rules
action, the most recent action, initially none

state ← UPDATE-STATE(*state*, *action*, *percept*, *model*)
rule ← RULE-MATCH(*state*, *rules*)
action ← *rule*.ACTION
return *action*

Model-based Reflexes in Pepper

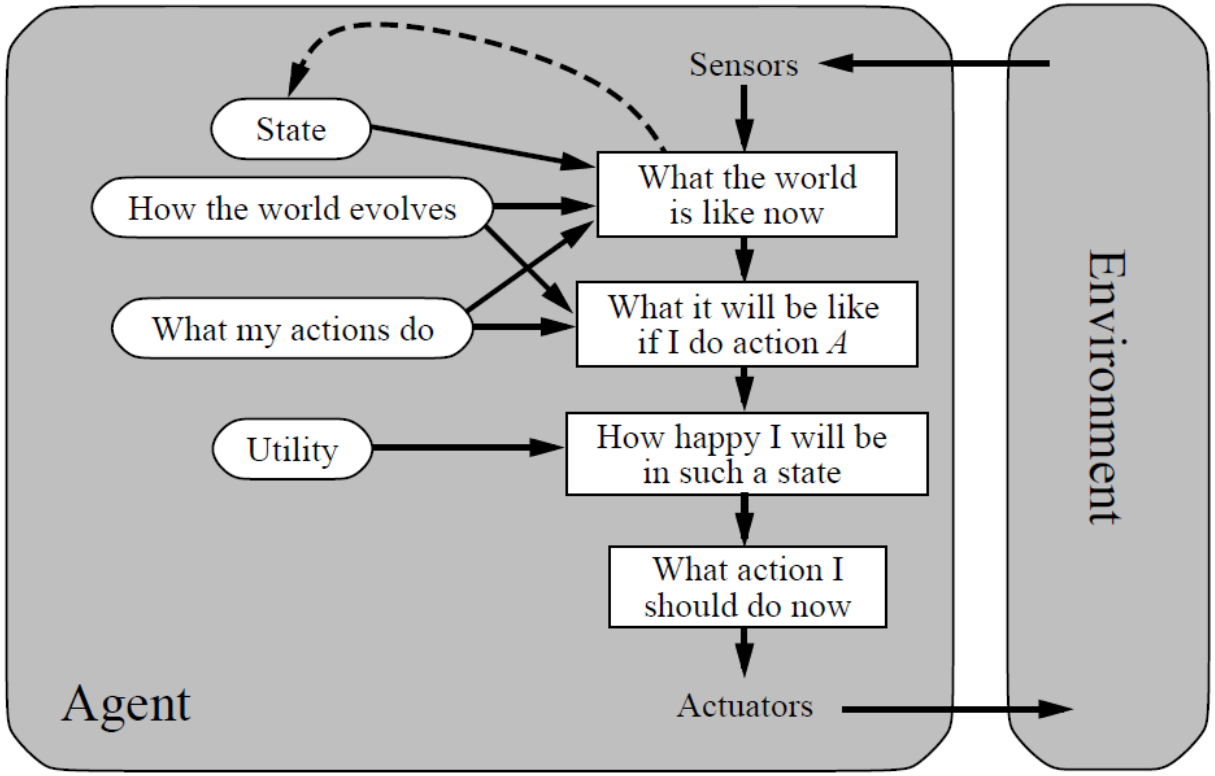


Model Based Pepper



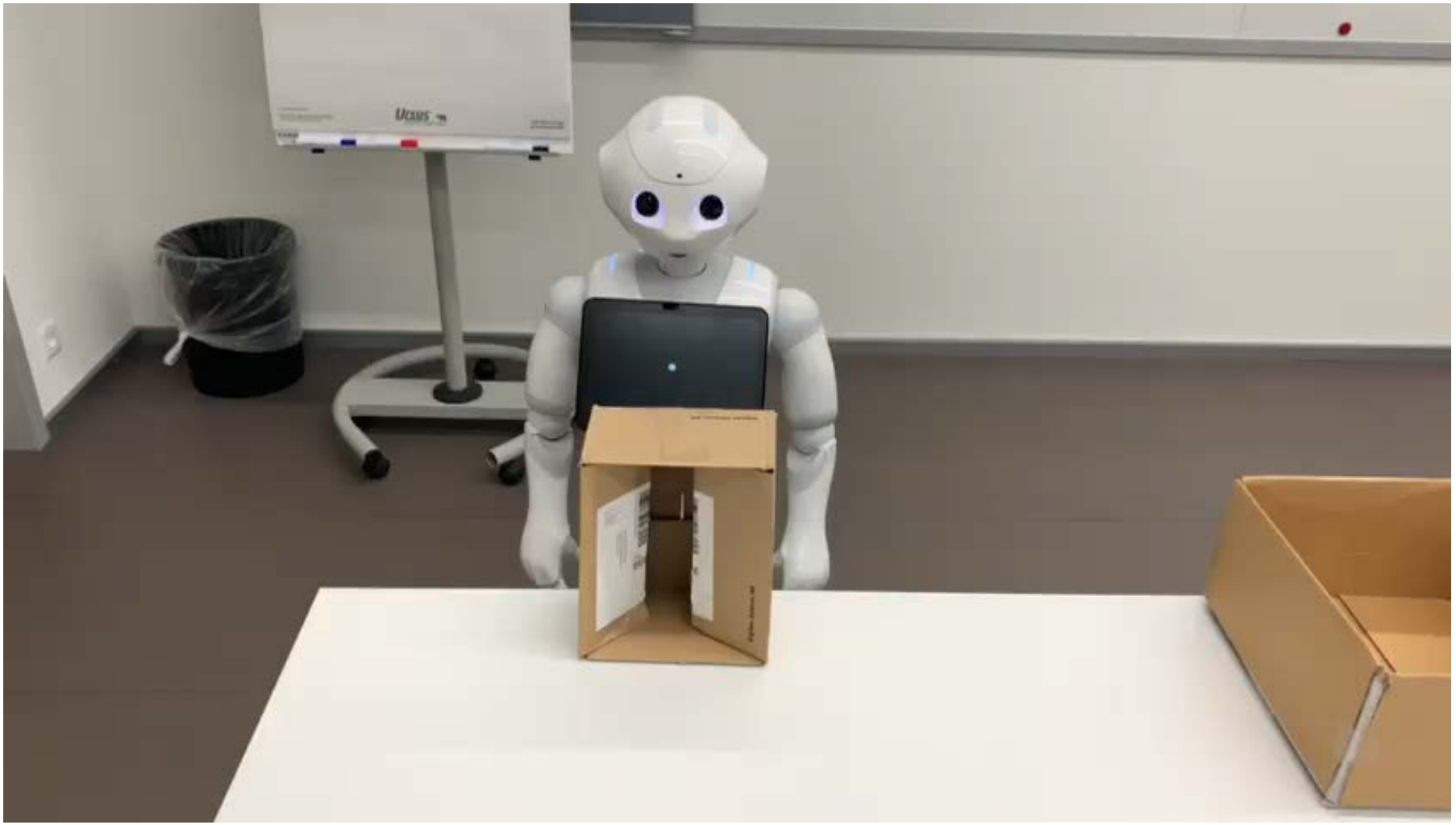
Watch videos about example agents in the CMS: Information > Materials > Videos

Utility-based Agent



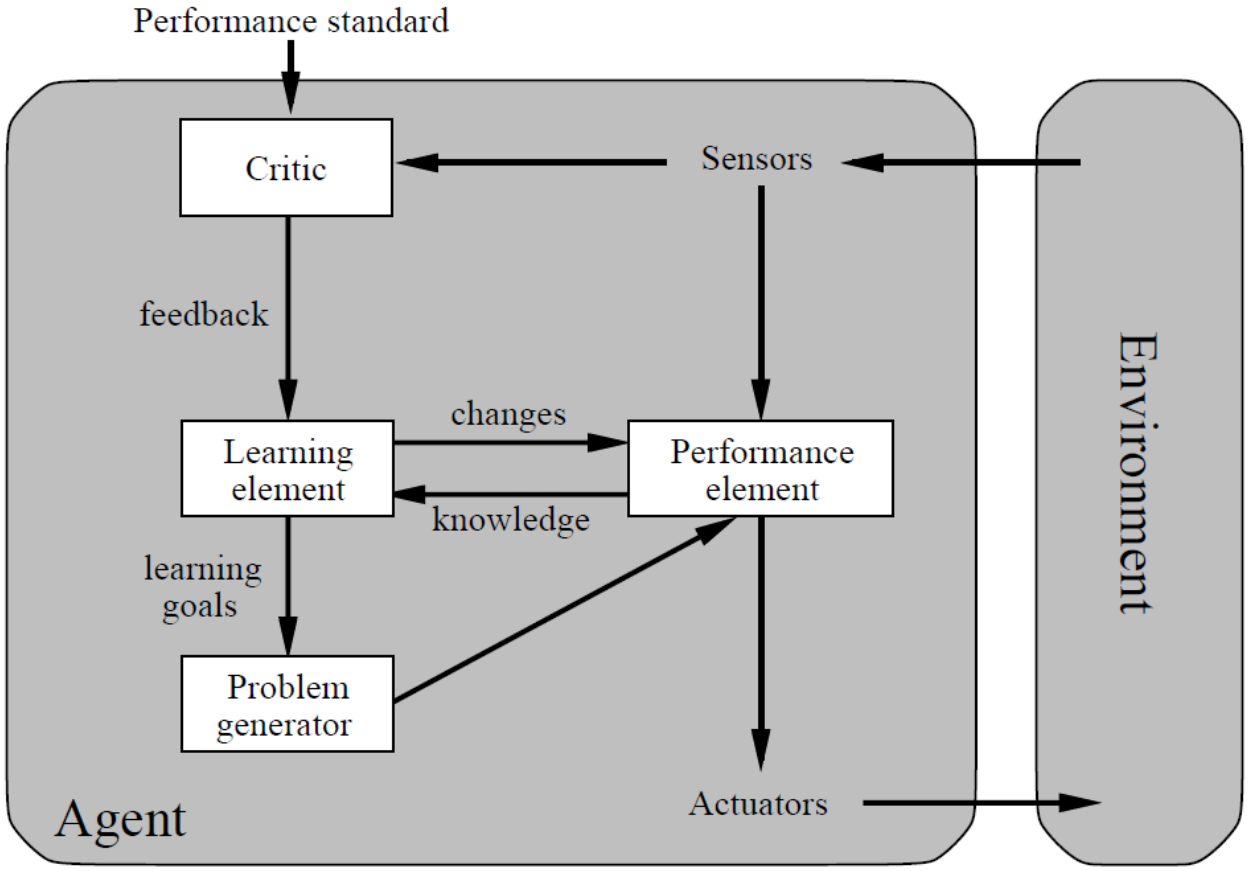
- If several actions are possible in a state, this agent can evaluate their utility and make a deliberate choice
  - reaches the goal via "useful" intermediate states

Utility-based Pepper



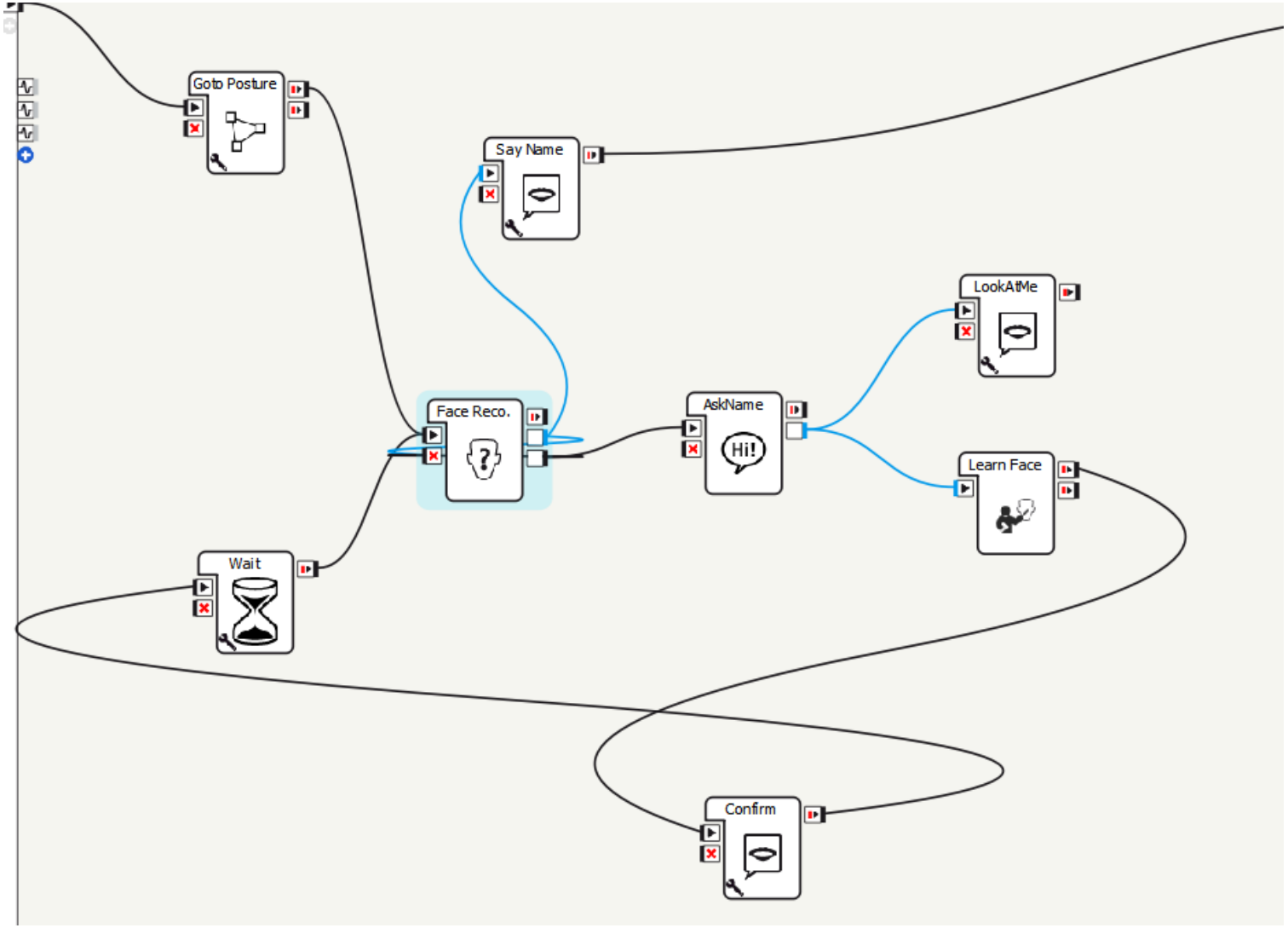
Watch videos about example agents in the CMS: Information > Materials > Videos

Learning Agent

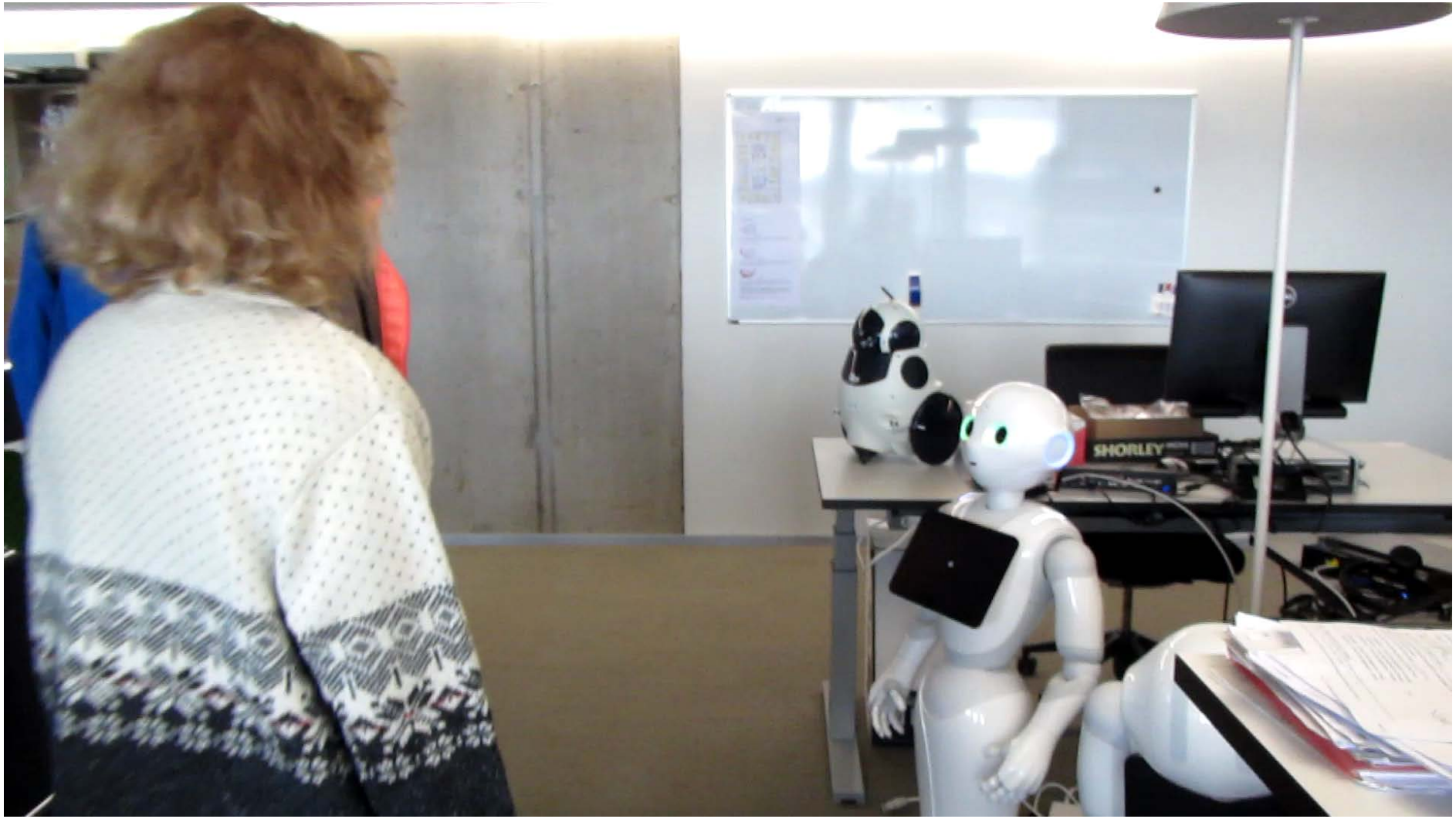


- This agent can acquire new skills and reflect on its own performance to improve over time

Learning Faces in Pepper



(Simple) Learning Pepper



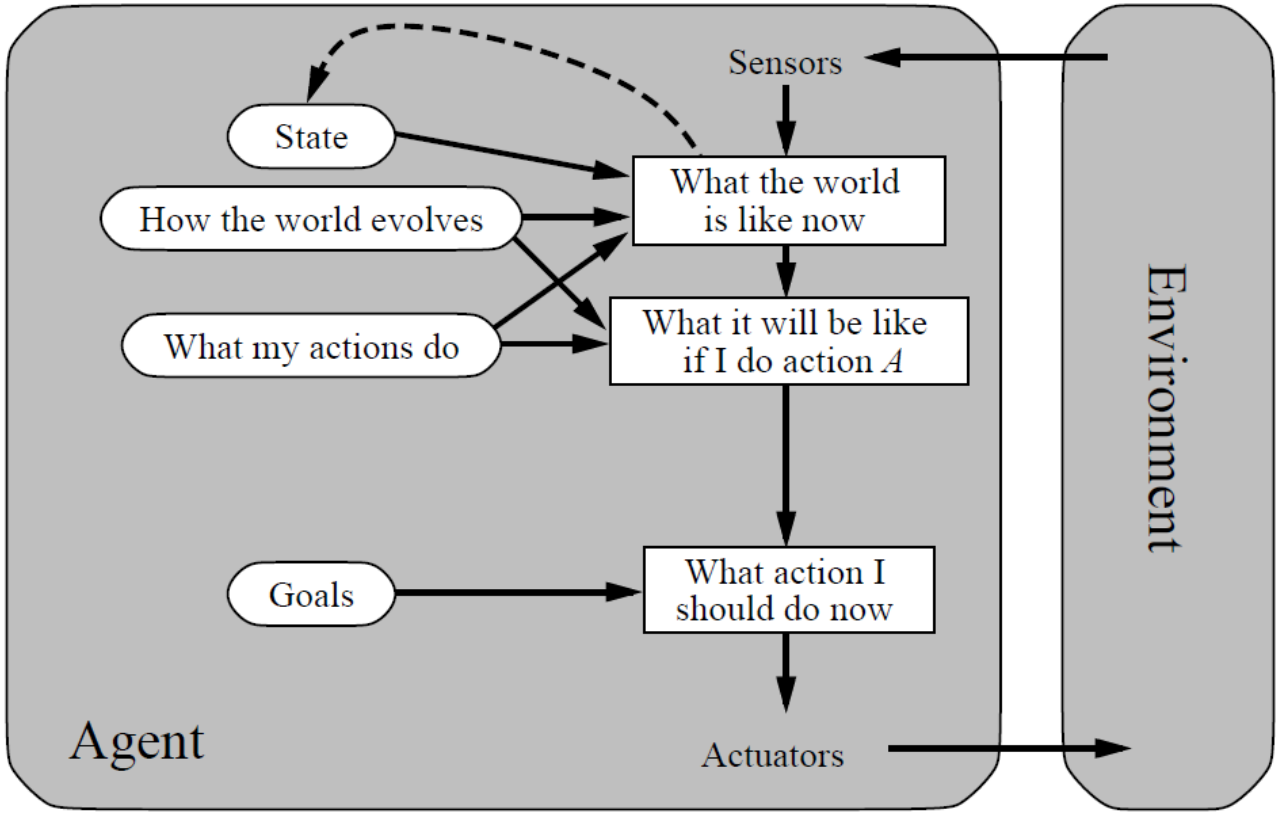
Watch videos about example agents in the CMS: Information > Materials > Videos

Learning Agent

- Learning agents can become more competent over time
- Can start with an initially empty knowledge base
- Can operate in initially unknown environments

- Responsibilities of its components
  - performance element: shows the agent how well it succeeded in the environment
  - learning element: improves the performance element by posing new tasks
  - critic: evaluates the behavior of the agent based on its performance and gives the evaluation as feedback to the learning element
  - problem generator: suggests actions that will lead to informative experiences

Goal-based Agent



- Builds a model of the world and uses an explicit representation of goals
- Evaluates the effects of actions on the world model before selecting an action

Goal-Driven Spot

https://www.youtube.com/watch?v=wXxrmussq4E

Summary

- AI research often assumes the metaphor of the rational agent, which maximizes expected outcome
- Humans are not rational agents and sacrifycing outcome for the benefit of other is important for the human society to work
- Understanding properties of the environment and the boundary between agent and environment is successful for building successful AI applications
- More complex environments usually require more complex agents, AI distinguishes different agent architectures
- Ethical and risk considerations are important when designing powerful agents, see e.g. MIT Moral Machine Experiment http://moralmachine.mit.edu/

Working Questions

1. What is a rational agent?
2. How can we characterize environments?
3. What types of rational agents do we distinguish?
4. What is a PEAS description of an agent/environment pair?
5. What agent type(s) do you need to build to achieve a certain behavior?