Intelligent Agents

Prof. Dr. habil. Jana Koehler

Artificial Intelligence - Summer 2019

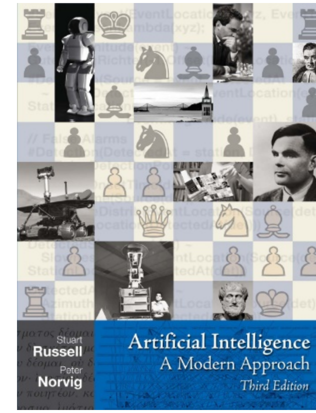
Agenda

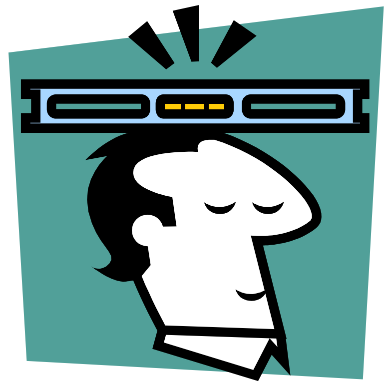
- The Rational Agent
  – perception – cognition – action

- Properties of Environments

- Types of Agents
  – **Simple Reflex** agents respond immediately to percepts
  – **Model-based Reflex Agent** is 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

Metaphor of the Rational Agent

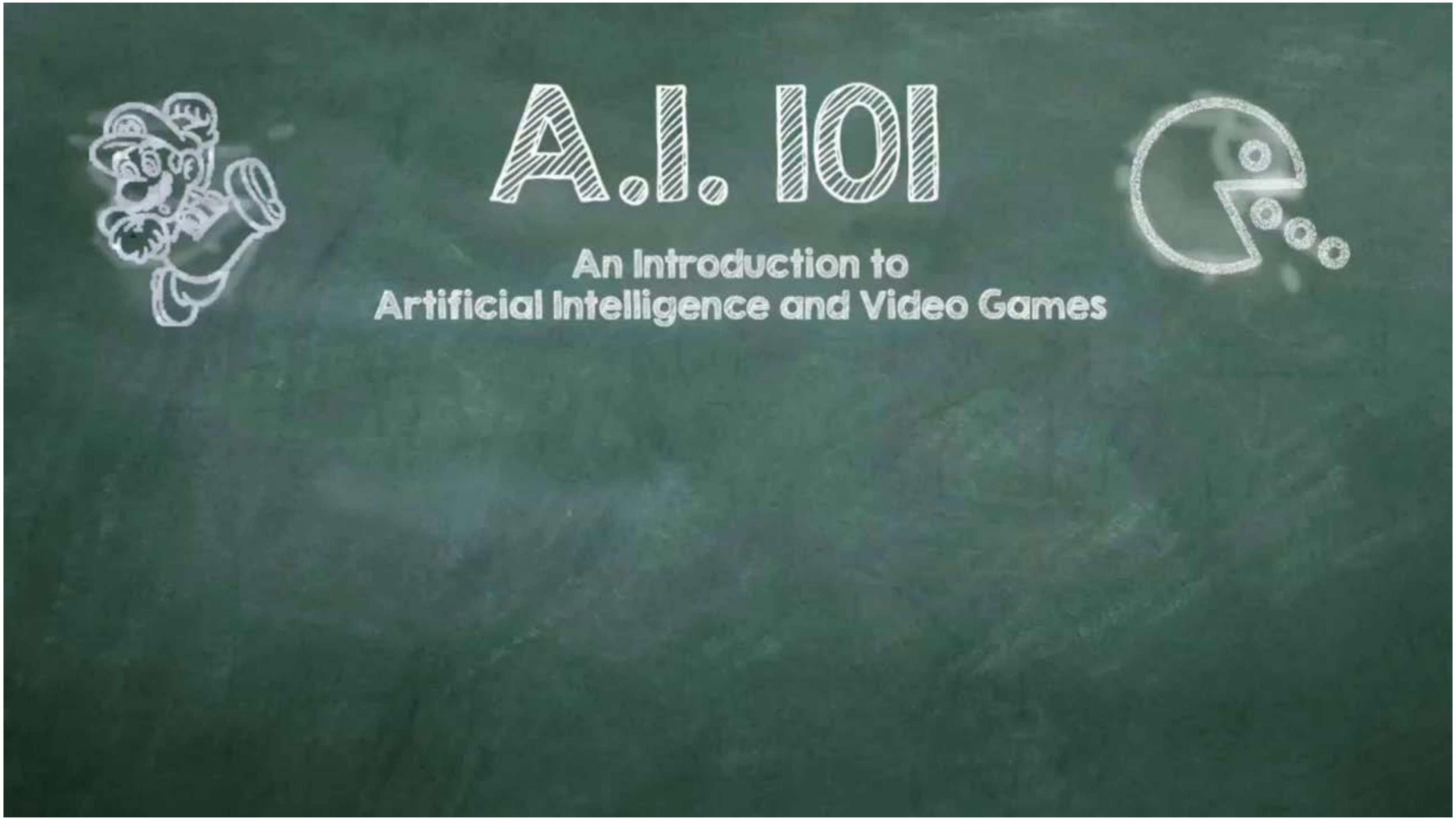




A central aspect of intelligence is the ability to act successfully in the world

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

What is an Agent?



https://www.youtube.com/watch?v=IK7l4ZLm55I

Agent

(1) Ability to perceive environment

(2) Observations used to make Decisions

(3) Decisions will result in actions

(4) Decision must be RATIONAL

    (1) best possible action the agent can take

The Rational Agent

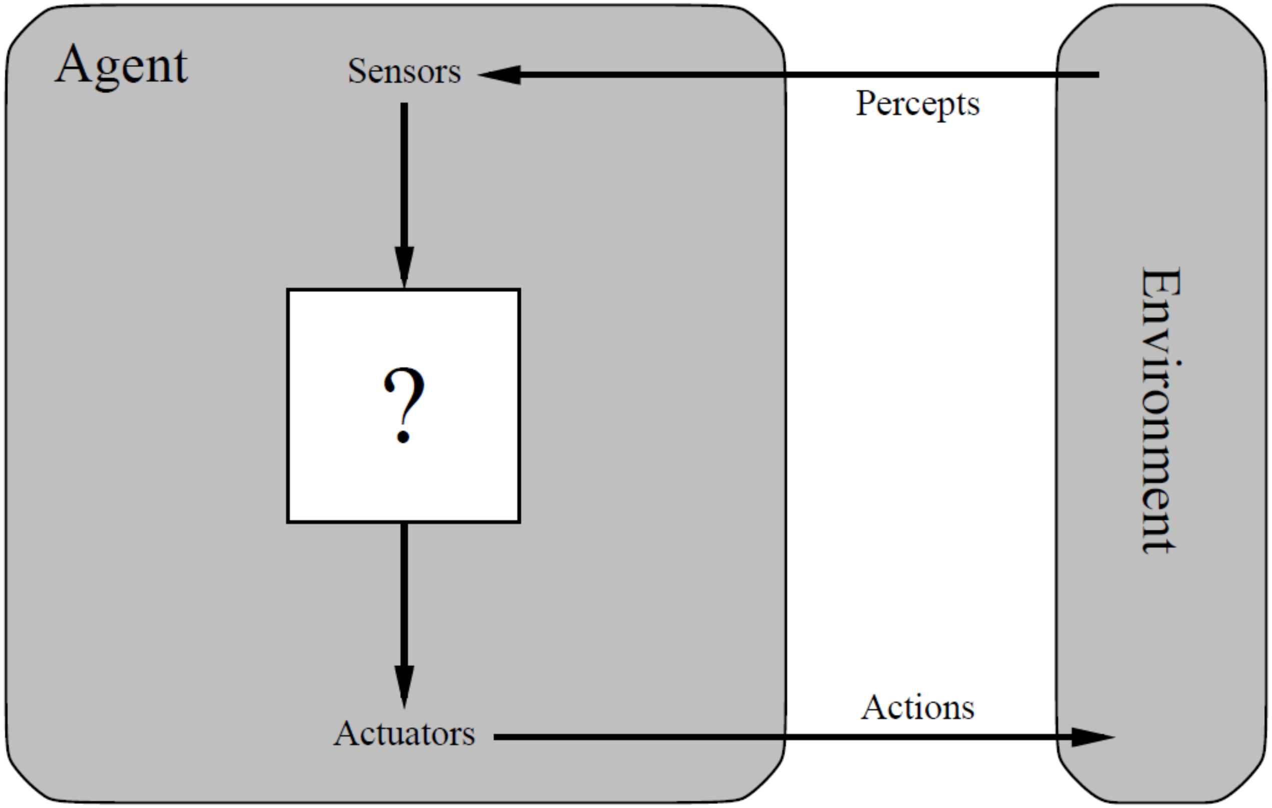
- AGENT = something that acts (latin *agere, do, make …)*
- RATIONAL AGENT = *acts so as to achieve the best expected outcome*
- Rational thinking is one possible mechanism to achieve rational behavior
- Perfect rationality cannot be achieved in complex environments
  – *LIMITED RATIONALITY = acting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

Rationality vs. Omniscience

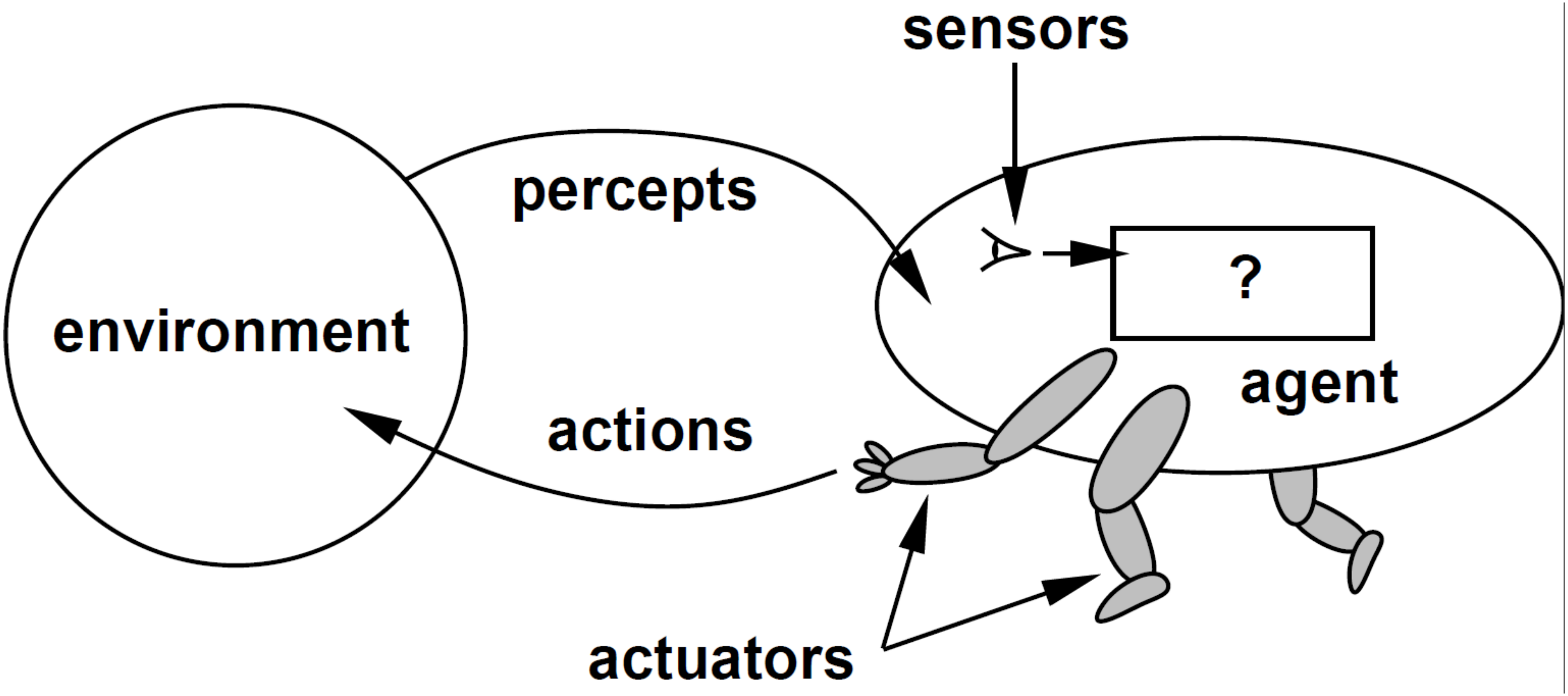
- An omniscient agent knows the actual effects of its actions
- In comparison, a rational agent behaves according to its percepts and knowledge and attempts to <u>maximize the expected performance</u>
- Example: If I look both ways before crossing the street, and then as I cross I am hit by a meteorite, 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)



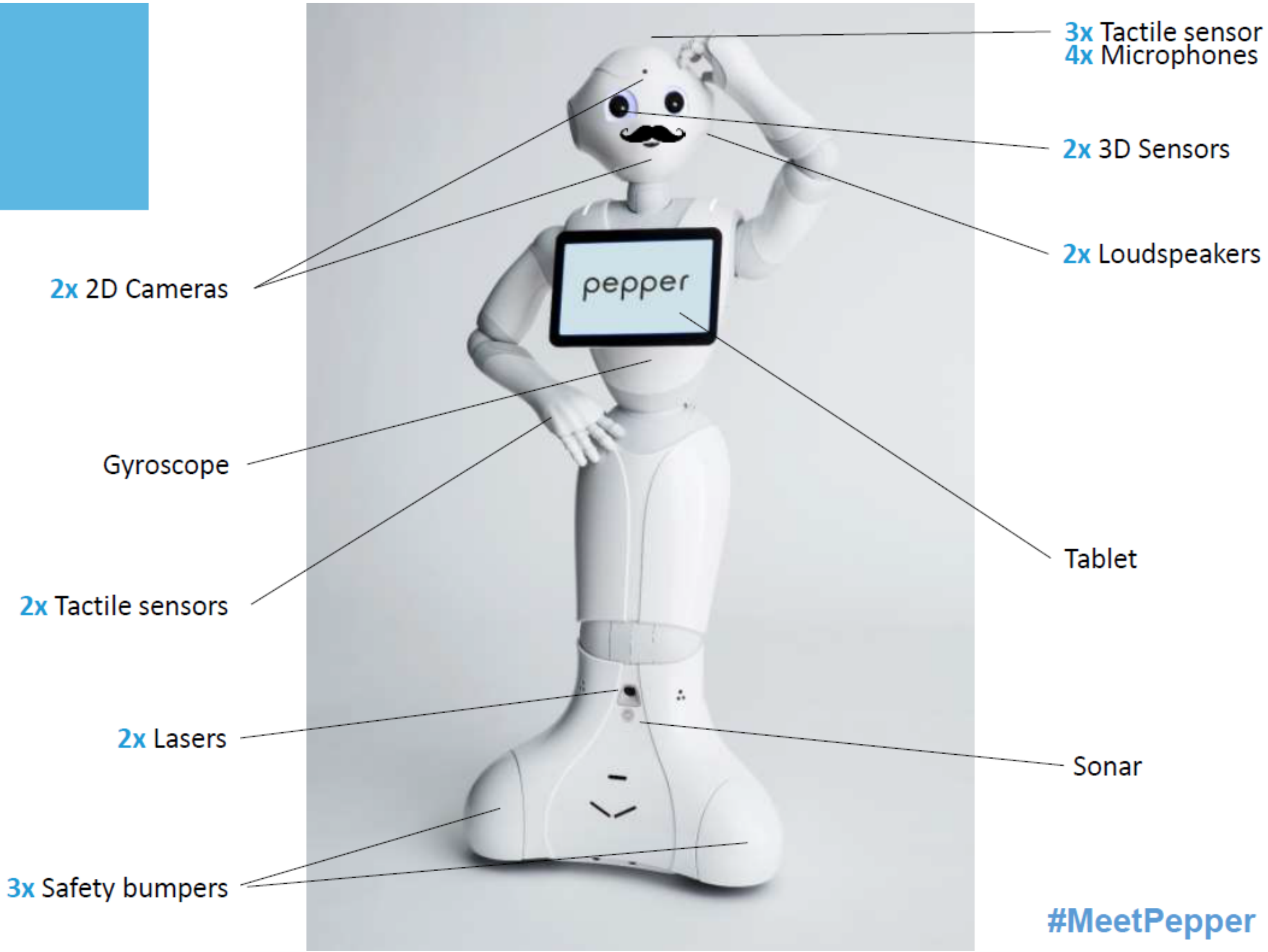
A Simplified Human Agent



Actuators and Sensors in Pepper

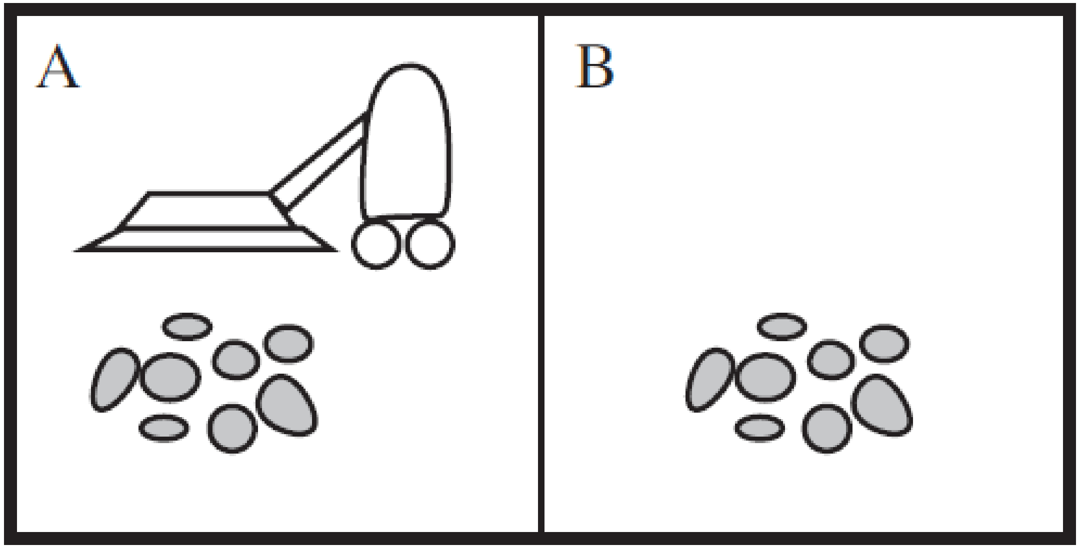
HARDWARE OVERVIEW

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



#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**
  - implementation of the agent function

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."

Formalization

Mapping your input to the best possible output:

$$\text{Performance measure} \times \text{Percepts} \times \text{Knowledge} \rightarrow \text{Action}$$

- An agent has a performance measure M and a set A of possible actions. Given a percept sequence P, as well as knowledge K about the world, it selects an action $a \in A$.
- The action a is optimal if it maximizes the expected value of M, given the evidence provided by P and K. The agent is rational if it always chooses an optimal a.

$\rightarrow$ If the vacuum cleaner bumps into the Hamster, then this can be rational in case the percept does not allow to recognize the hamster.

$\rightarrow$ Note: If observation actions are required, they are elements of A, i.e., the agent must *perceive actively*. Example: *"truck-approaching"* $\notin P$ *but I didn't look to check* $\Longrightarrow$ *I am NOT being rational!*

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 <u>by the agent</u> to evaluate the desirability of a state of the world

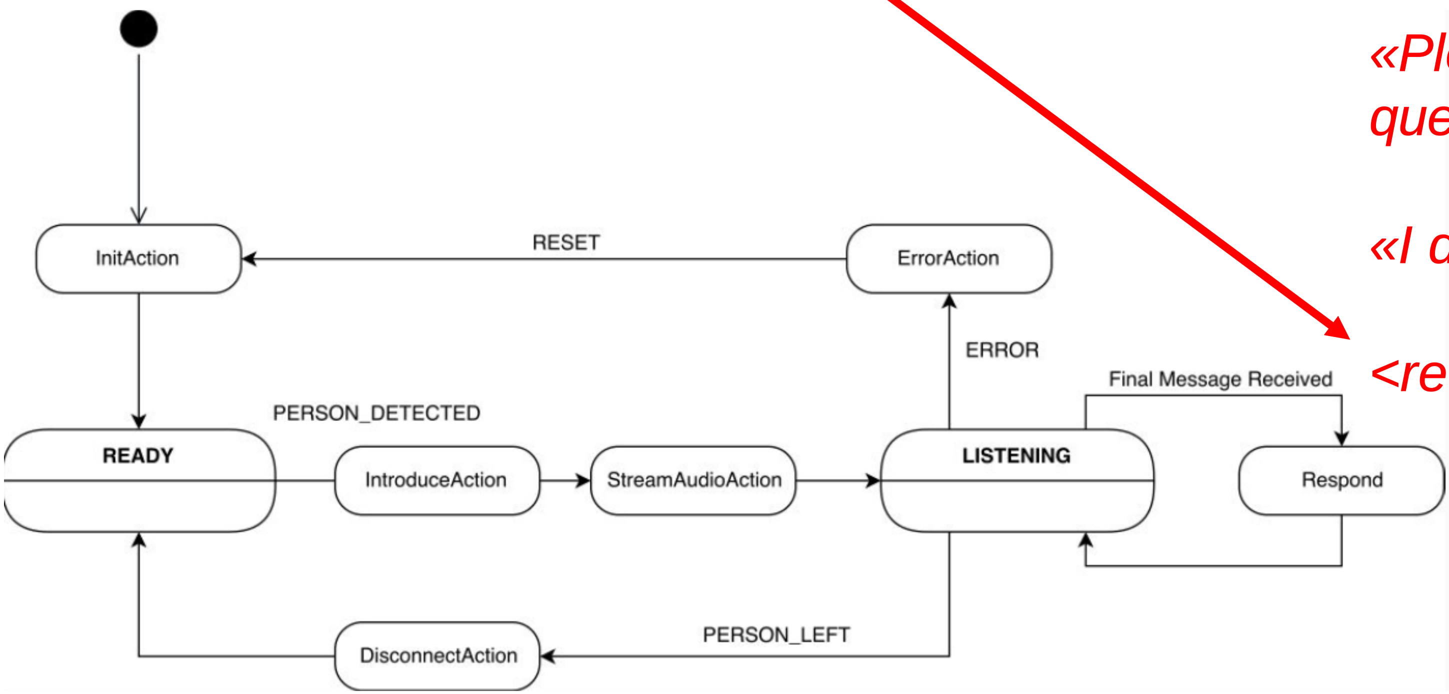
- A utility function maps a state (or a sequence of states) onto an evaluation value (usually a real number)

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

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
How much does the agent **know** about the environment initially?	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
How can we **model** the environment?	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*	Sequential**
Can we model the AI system with only one agent?	Single-Agent	Multi-Agent

- *episodic memory = short-term memory
  - the agent does not remember the past to make good decsions
- **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 Architectures

We also need to realize the agent, through:

- an agent program, executed on
- an architecture which also provides an interface to the environment (percepts, actions).

$\rightarrow$ Agent $=$ Architecture $+$ Program.

Practical limitations:

- Our definition captures limitations on percepts and knowledge.
- It does not capture computational limitations (often, determining an optimal choice would take too much time/memory).

  $\rightarrow$ In practice, we often merely *approximate* the rational decision.

Types of Agents

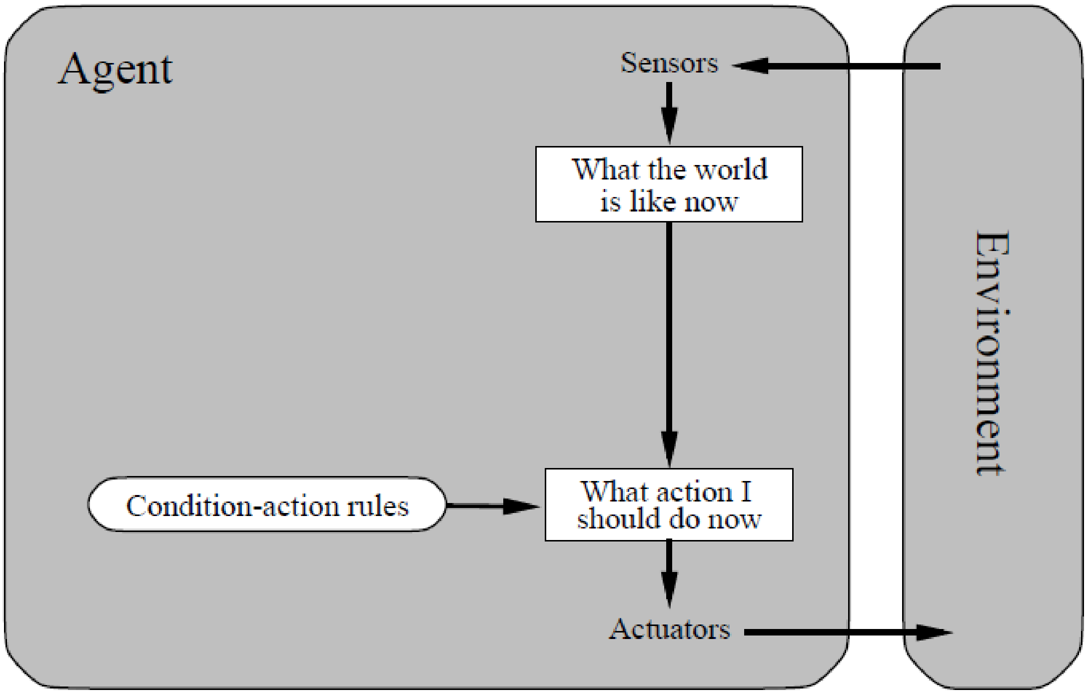
Agents differ in their capabilities

- **Exploration**: explorative actions for information gathering
- **Learning**: as much as possible from the perceipts
- **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)

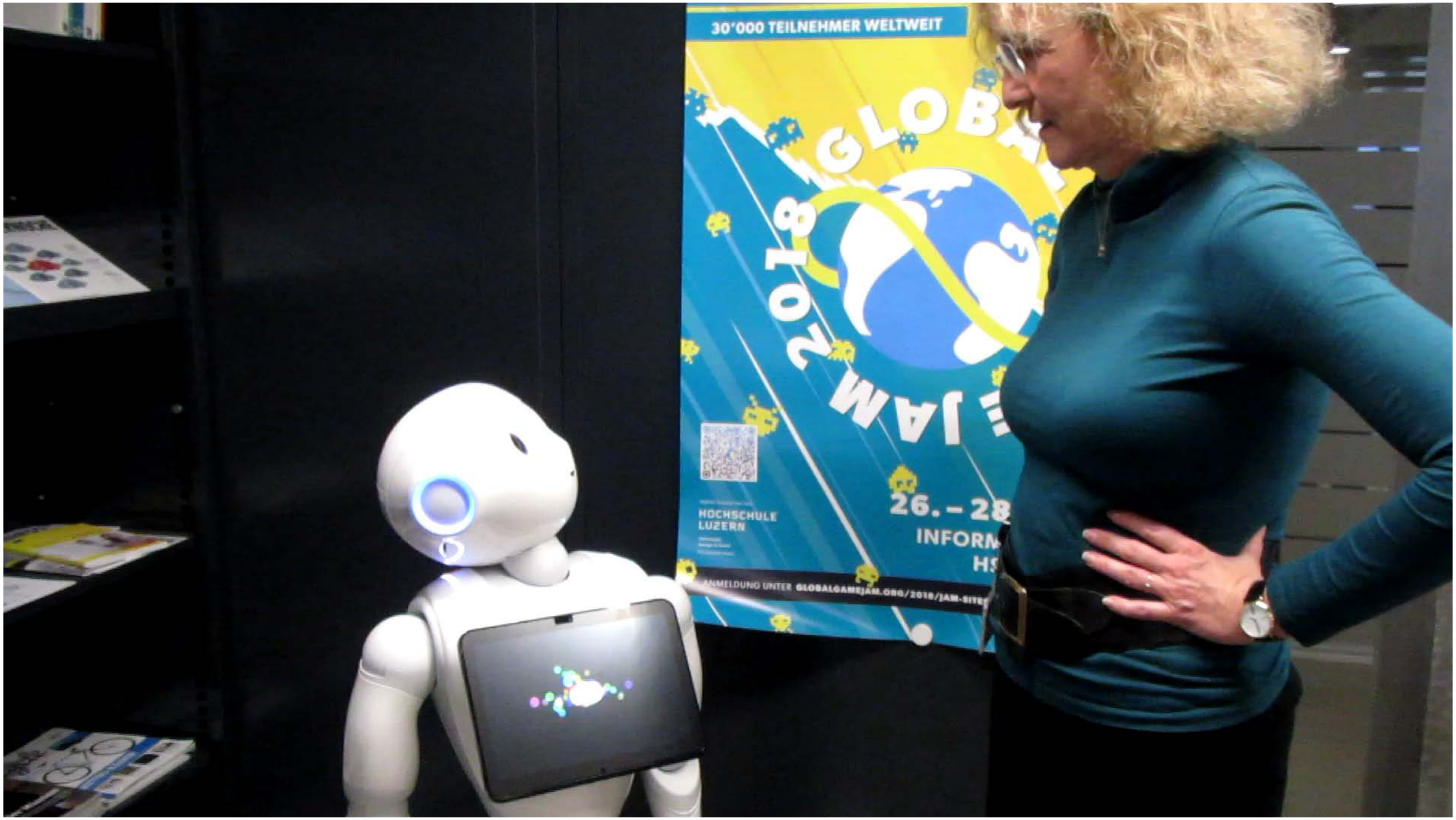
"**If** *car-in-front-is-braking* **then** *initiate-braking*."

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 agent without a location sensor and in a clean room A: moving left will fail forever
- Vacuum agent with a coin flip to choose a move

Simple Reflexes in Pepper

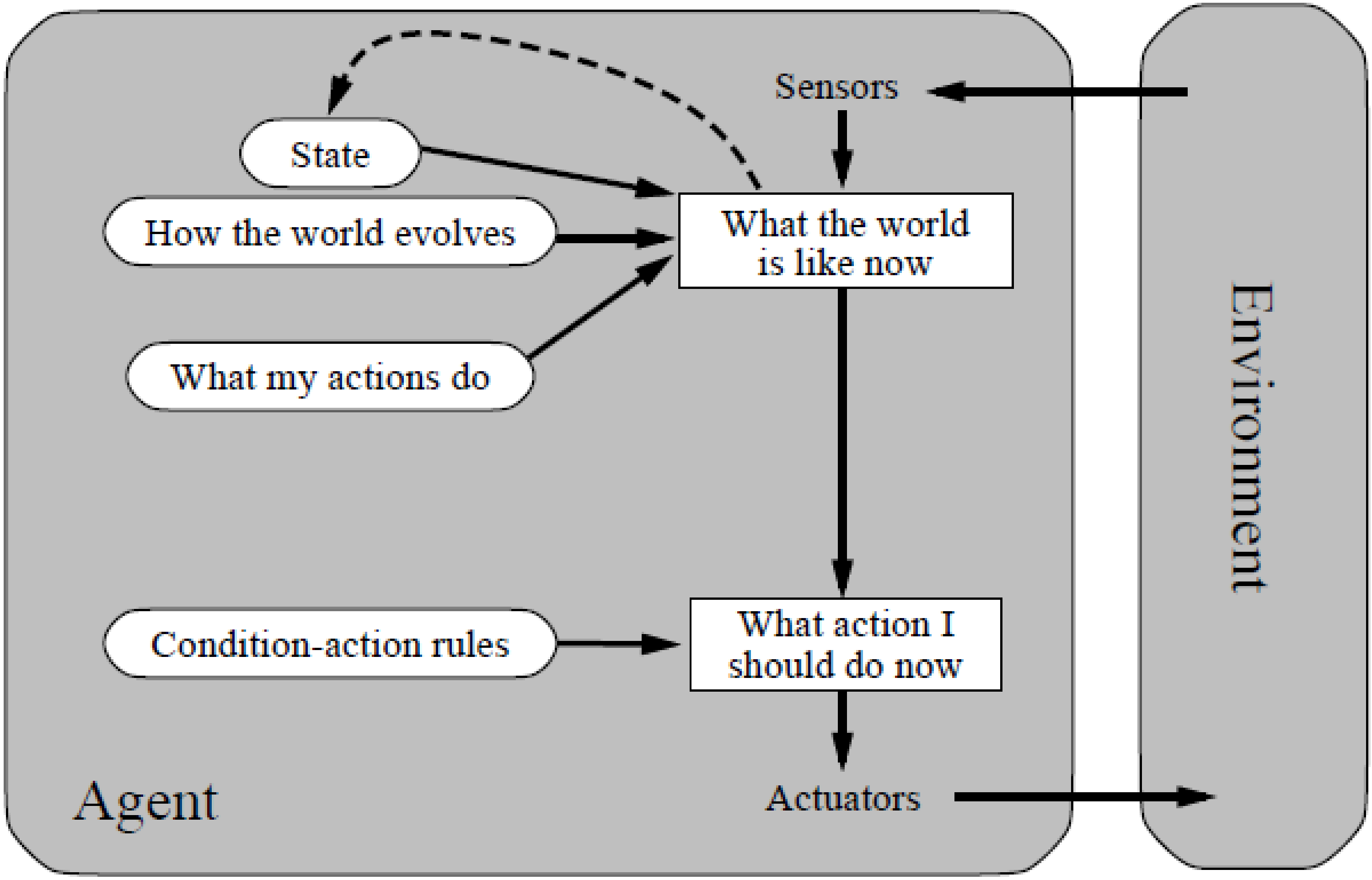


Model-based Reflex Agent

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

- Uncertainty about the world state is unavoidable
  - model represents the agent's "best guess"

Model-based Reflex Agent

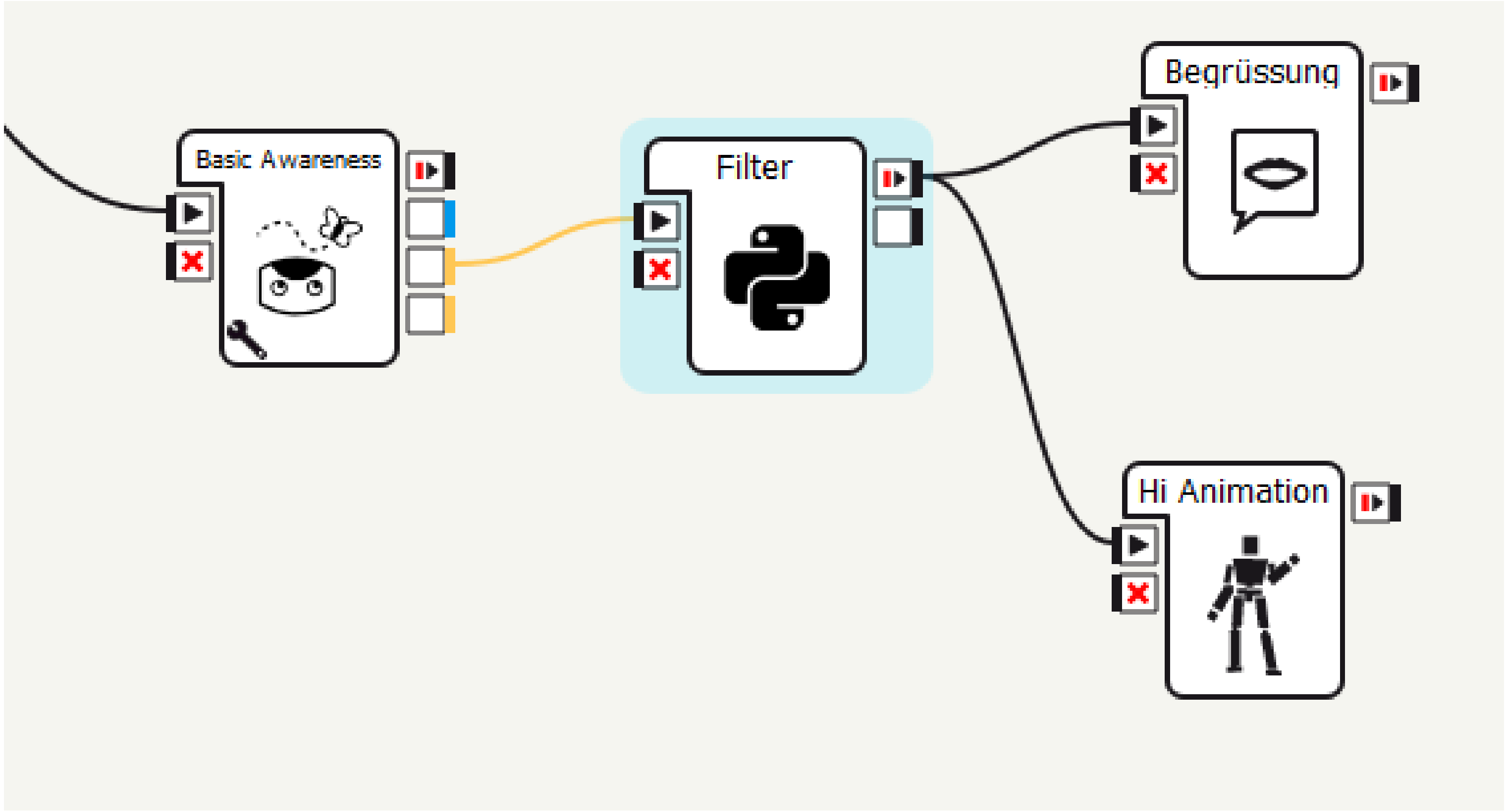


Function of the Model-based Agent

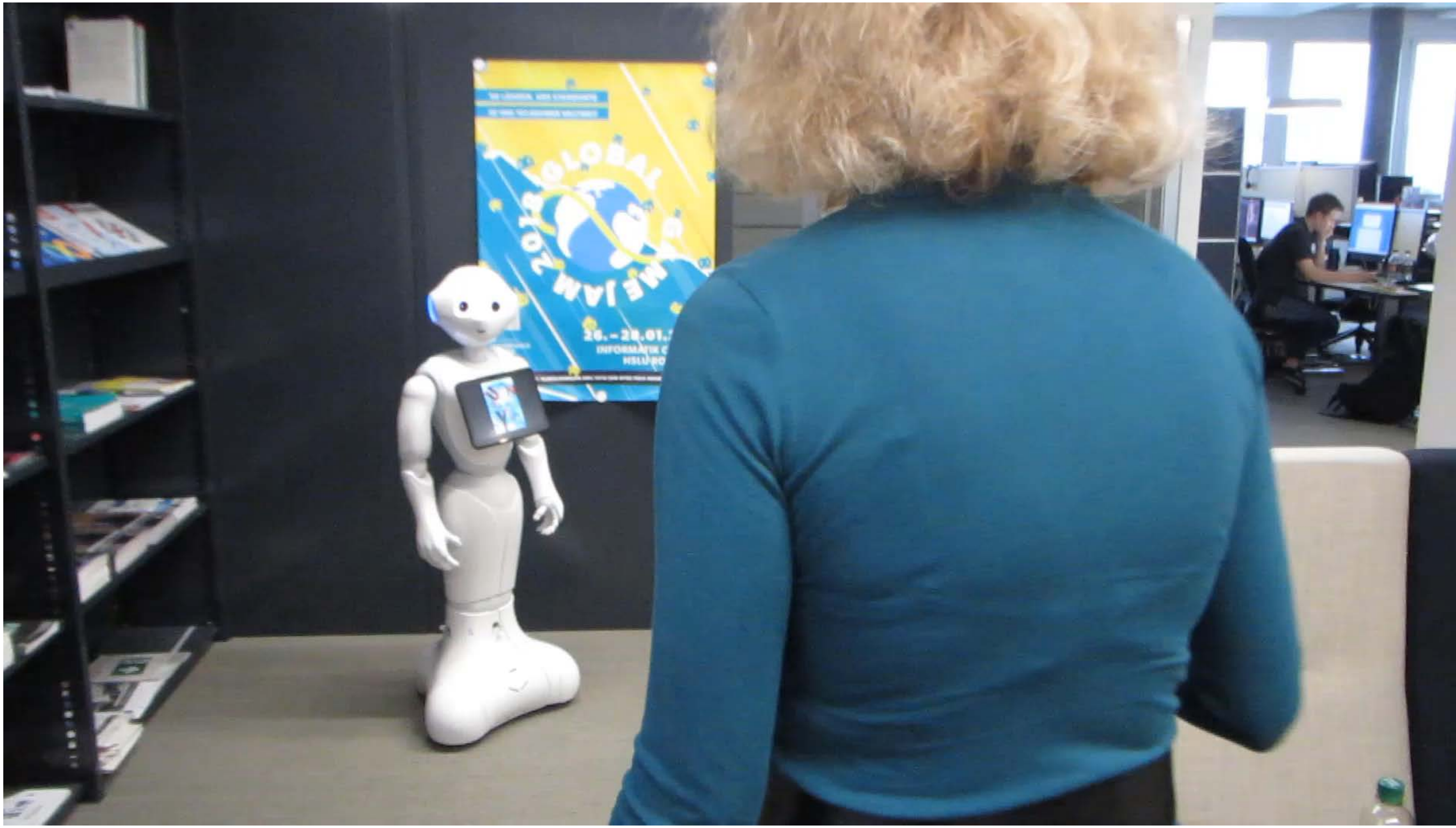
function MODEL-BASED-REFLEX-AGENT(*percept*) **returns** an action
  persistent: *state*, the agent's current conception of the world state
    model, a description of how the next state depends on current state and action
    rules, a set of condition–action rules
    action, the most recent action, initially none

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

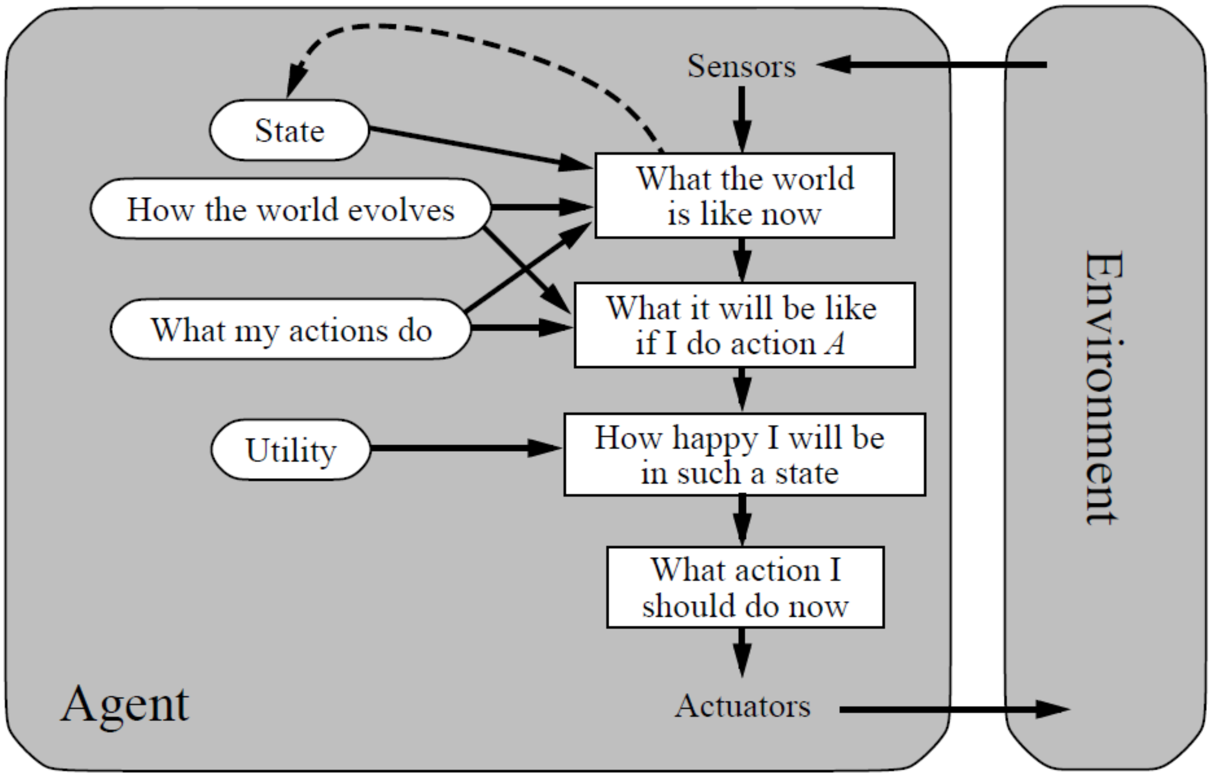
Model-based Reflexes in Pepper



Model Based Pepper

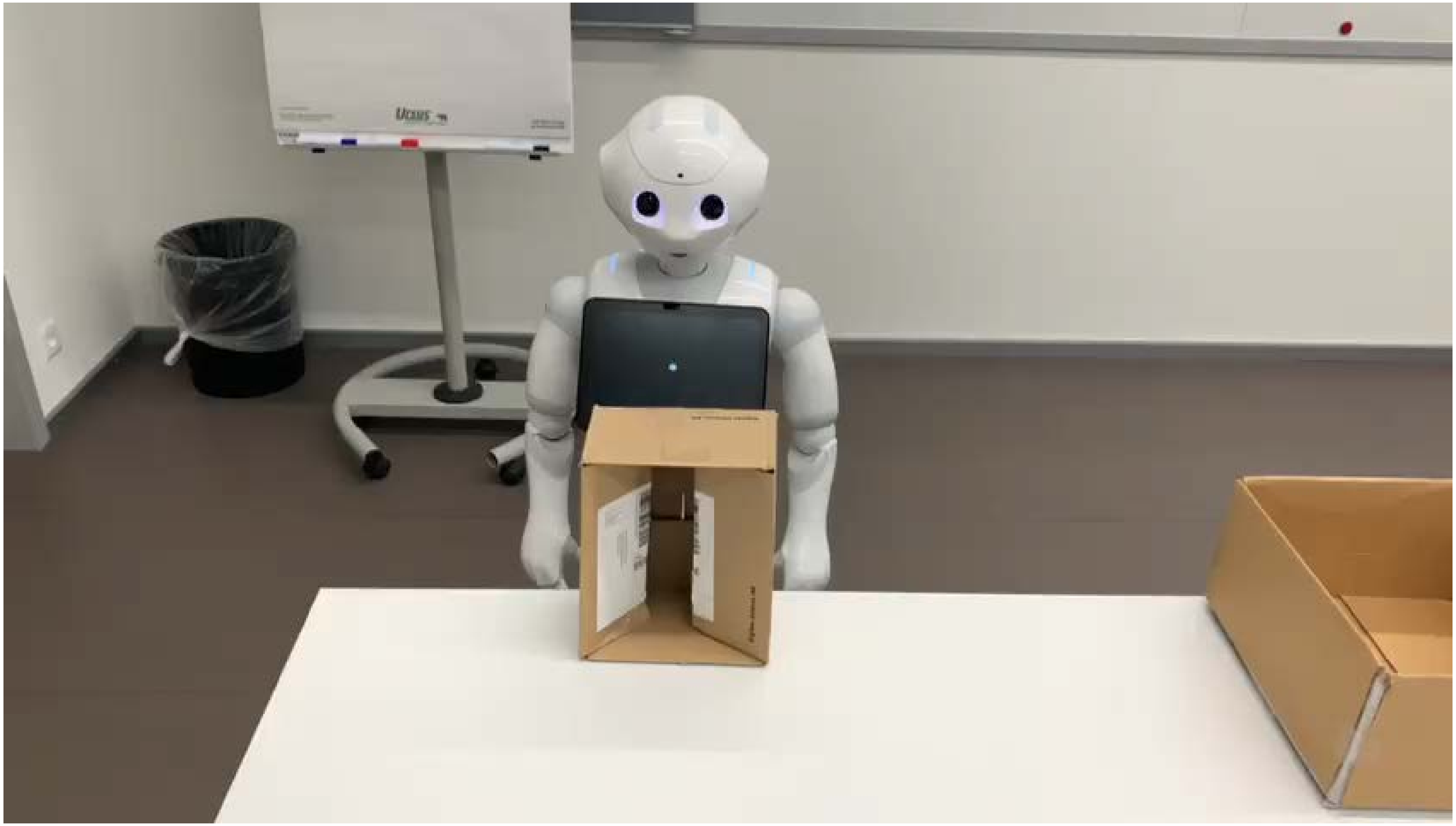


Utility-based Agent

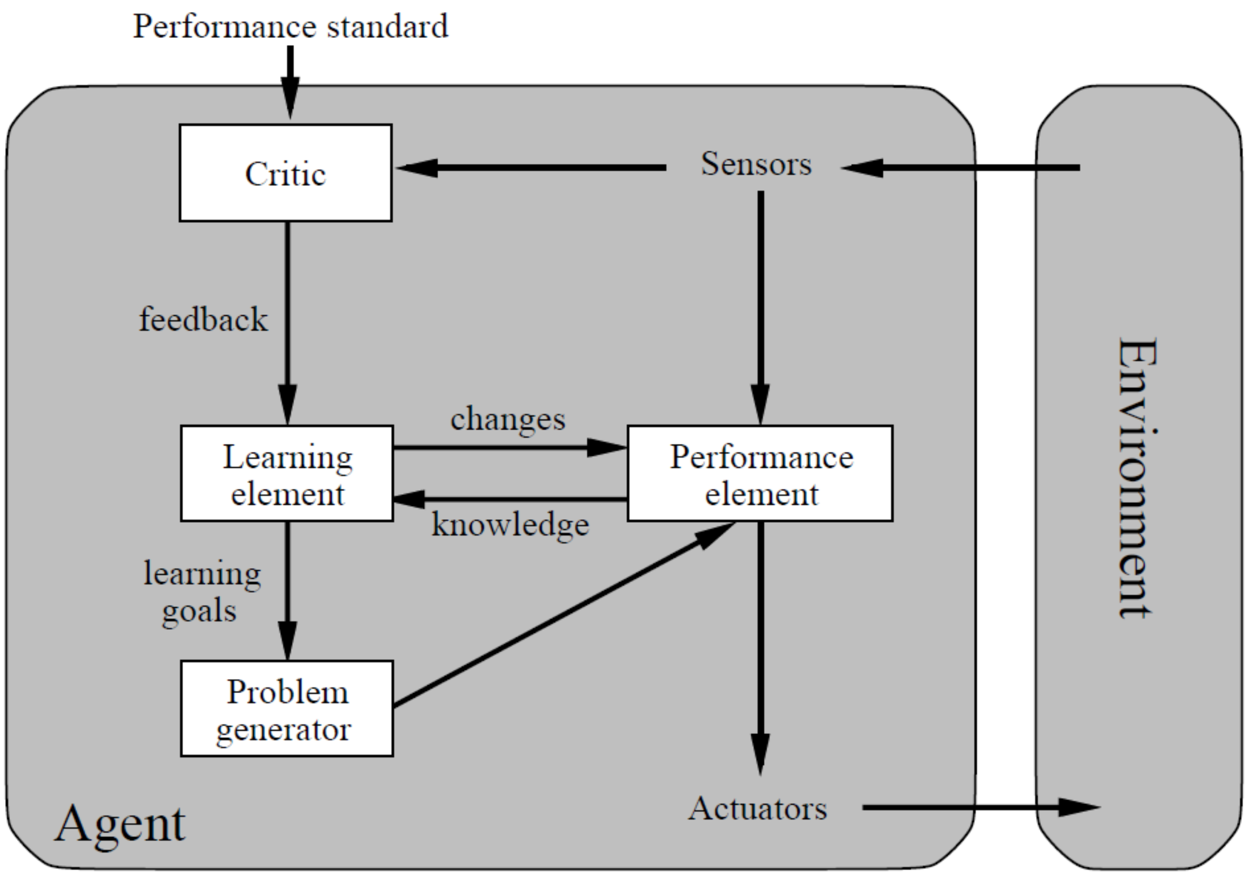


- If several actions are possible in a state, this agent can evaluate their utility and make a deliberate choice
  - reaches the goal via "useful" intermediate states

Utility-based Pepper

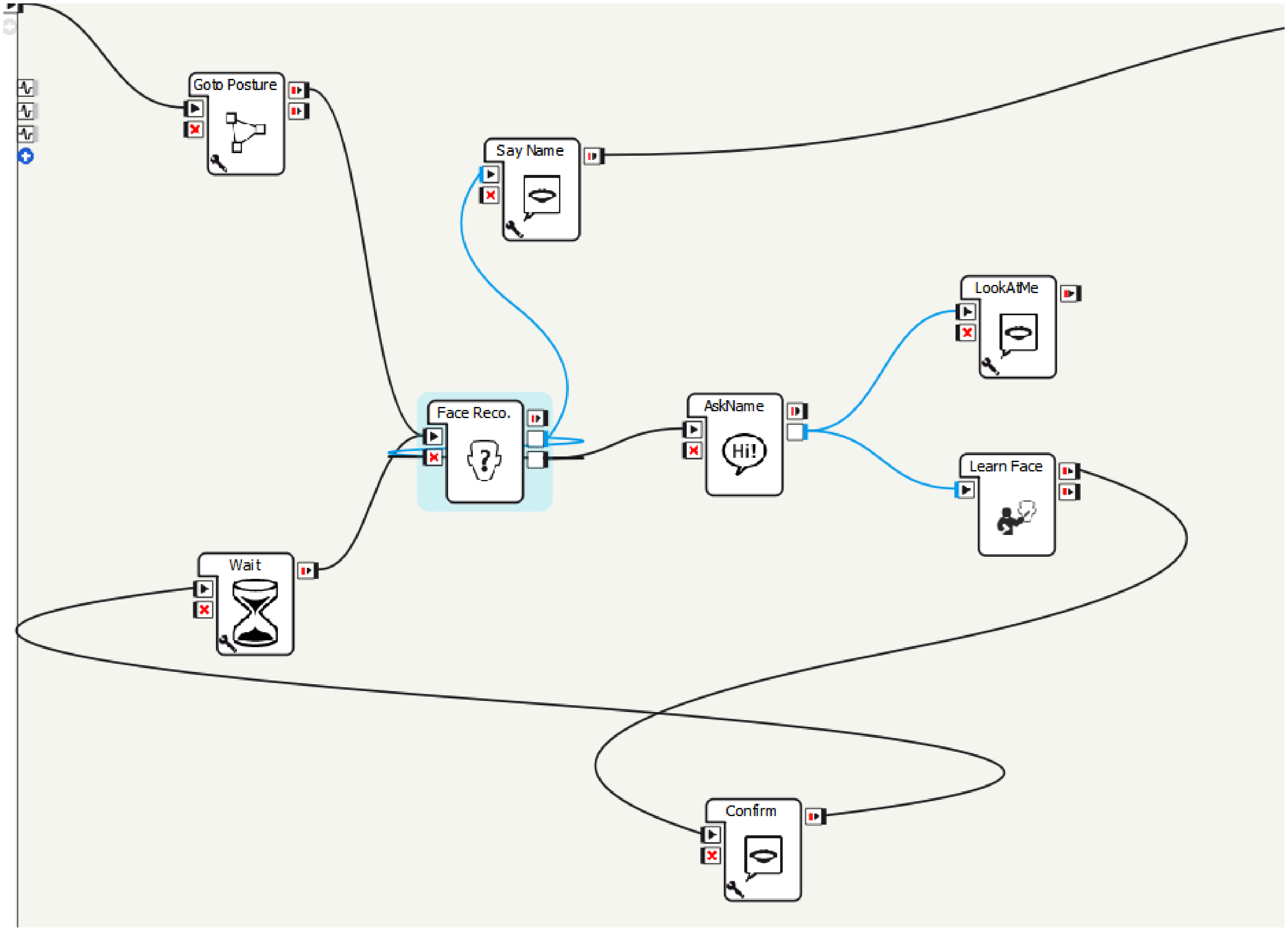


Learning Agent

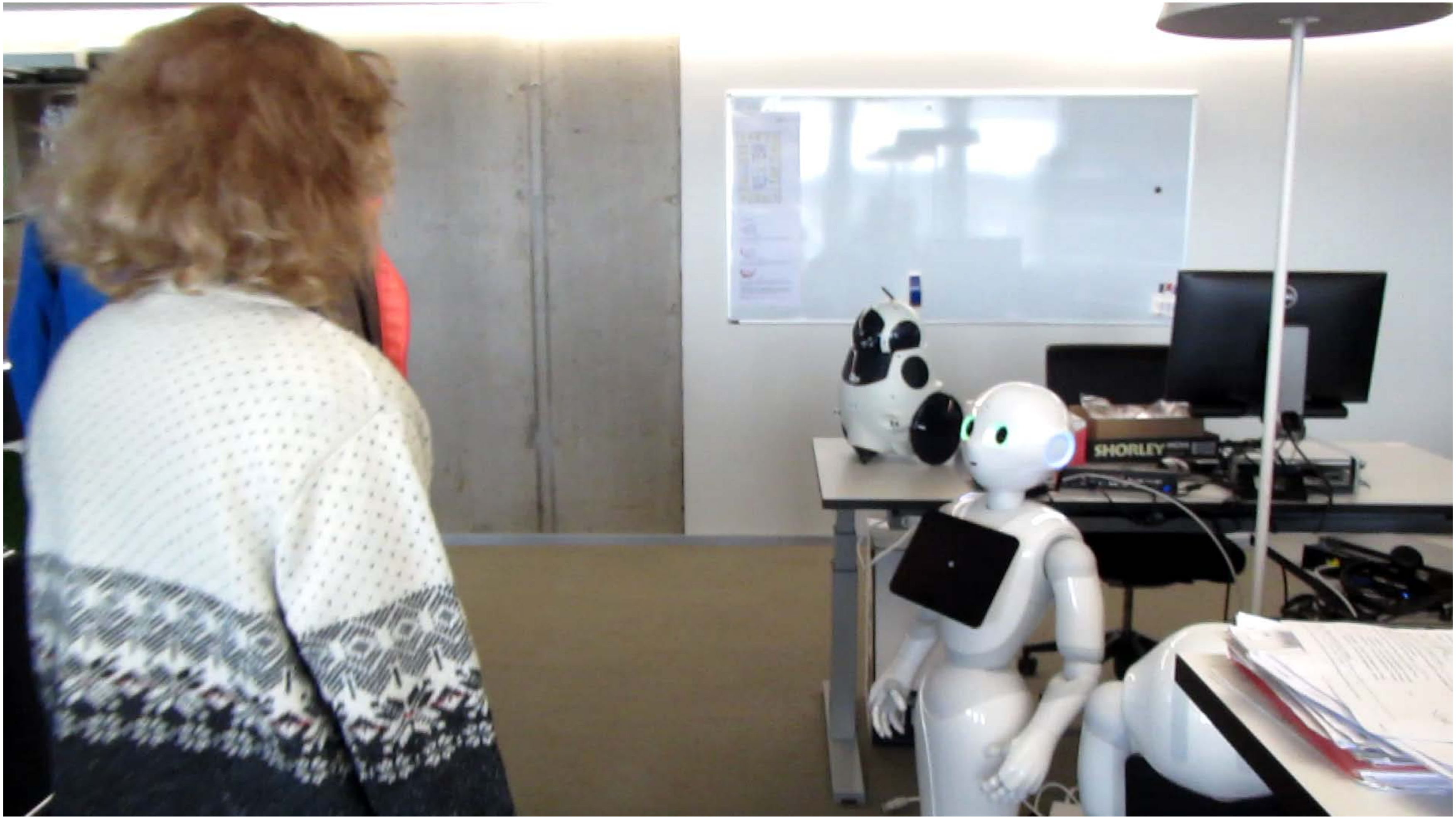


- This agent can acquire new skills and reflect on it's own performance to improve over time

Learning Faces in Pepper



(Simple) Learning Pepper

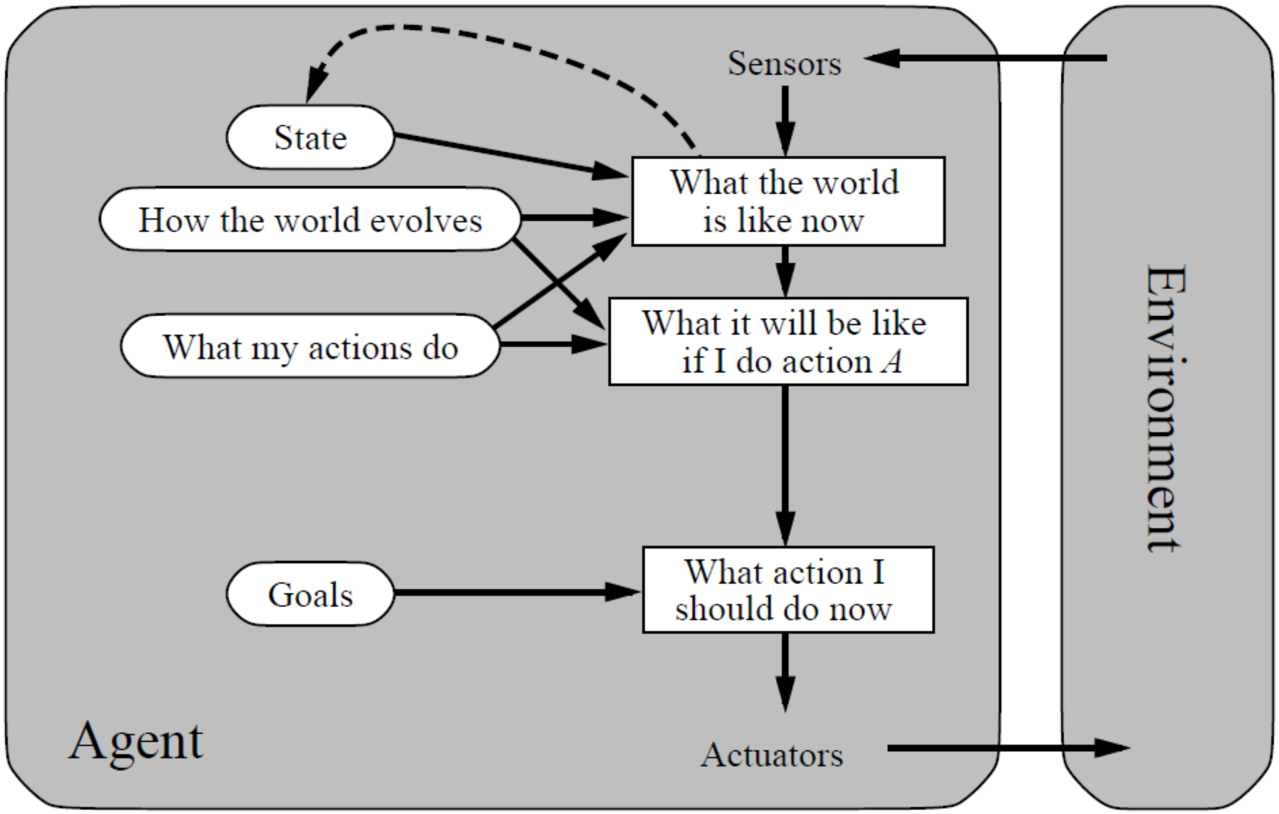


Learning Agent

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

- Responsibilites of its components
  - performance element: shows the agent how well it succeed 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

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 the PEAS specification for your chosen behavior?
5. What agent type(s) do you need to build to achieve the behavior?

- cognitive services https://cs-workshop.enterpriselab.ch

Further Reading

- Chapter 2 in Russel/Norvig
- AAMAS - International Conference on Autonomous Agents and Multiagent Systems is the top address to understand the current research in this field
  - http://ifaamas.org/
- MIT Moral Machine Experiment http://moralmachine.mit.edu/
- Apply this in another course? Watch out for Cognitive Robotics Lab seminar in Fall 2019