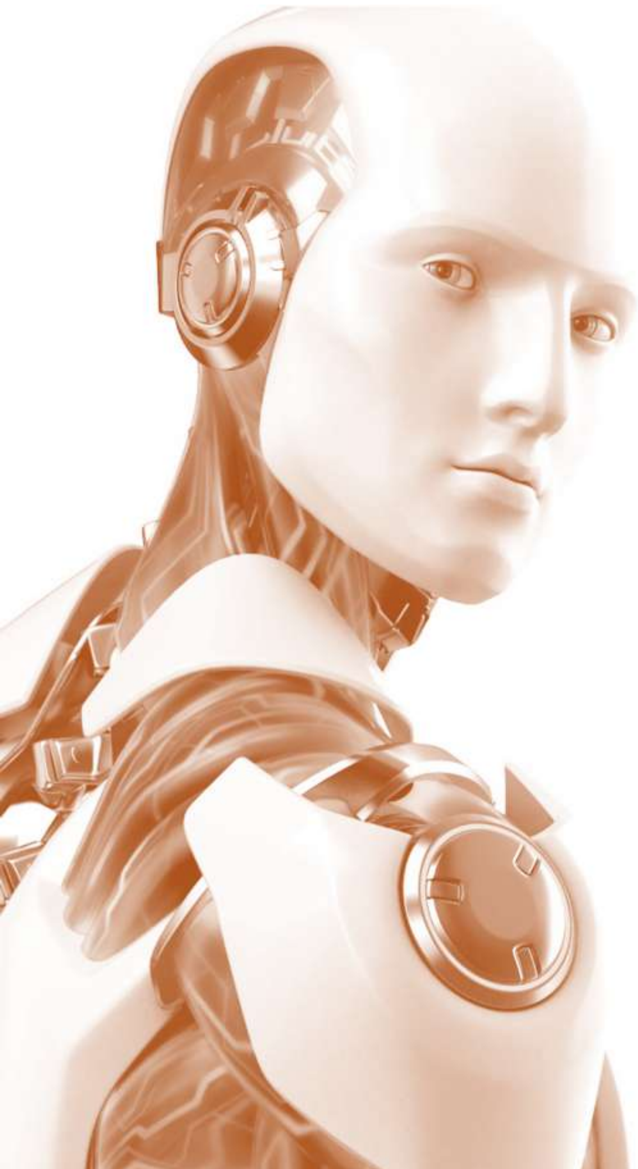


Introduction to AI
Lecture 3 - Agents

Profa. Dra. Esther Luna Colombini
esther@ic.unicamp.br

Prof. Dr. Alexandre Simões
alexandre.simoes@unesp.br

LaRoCS – Laboratory of Robotics and Cognitive Systems

LAROCS

UNICAMP

Advanced Institute for Artificial Intelligence

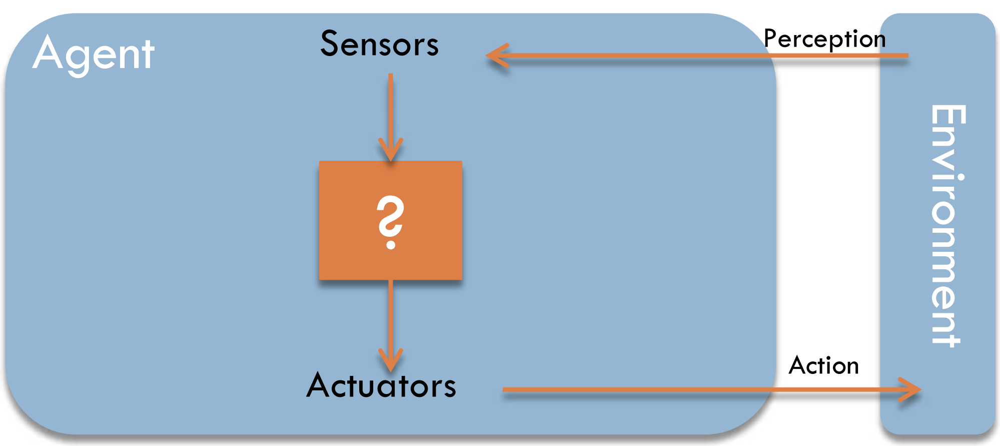
- What can a computer currently solve?
  - Play a game of ping pong
  - Driving in the center of Cairo
  - Buying groceries for a week at the market
  - Buying groceries for a week on the web
  - Play a competitive game of “truco”
  - Discover and prove new mathematical theorems
  - Writing an intentionally funny story
  - Provide suitable legal advice in a specialized area of law
  - Translate spoken English into Swedish spoken in real time
  - Perform a complex surgical operation

Summary

- Agents
  - Basic concepts: function, program, perception, actions, ...
  - Relationship between behavior and performance
  - Rationality
  - Autonomy
- Environemts
  - Types and properties
- Agents in environments
  - Software
  - Types

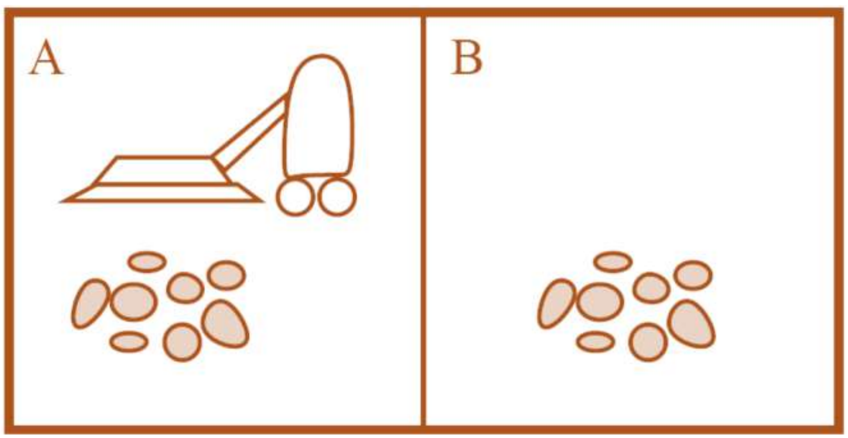
Agents

- An agent is everything that can be considered capable of perceiving an environment through sensors and acting on it through actuators.
  - Examples of agents: man, robot, software ...
- **Perception:** what does it see?
- **Action:** what can it do?
- **Agent function:** maps any specific perception sequence to an action (abstract description)
  - $f : P* \longrightarrow A$
- **Agent program:** implementation of the agent's role (concrete description)



Vacuum Cleaner World

- **Agent function:** to clean
- **Environment:** two rooms A and B (simple world: you can describe everything that happens!)
- **Agent's perception:** what room is it in and if there is dirt in the room
- **Agent actions:** move left, right, vacuum dust or do nothing (NOP)



Vacuum Cleaner World

Tabulation of the agent's role:

Sequence of perceptions:	Action
[A, Clean]	Right
[A, Dirty]	Clean
[B, Clean]	Left
[B, Dirty]	Clean
[A, Clean] [A, Clean]	Right
[A, Clean] [A, Dirty]	Aspirar
...	...
[A, Clean] [A, Clean] [A, Clean]	Right
[A, Clean] [A, Clean] [A, Dirty]	Clean
...	...

- "The choice of an agent's action at any given moment may depend on the entire sequence of perceptions observed so far"
- How to fill the table?
- What makes the agent good or bad?

What is a good agent?

- A good agent is one who does everything right

- What does it mean for the agent to do everything right?

$$\Downarrow$$

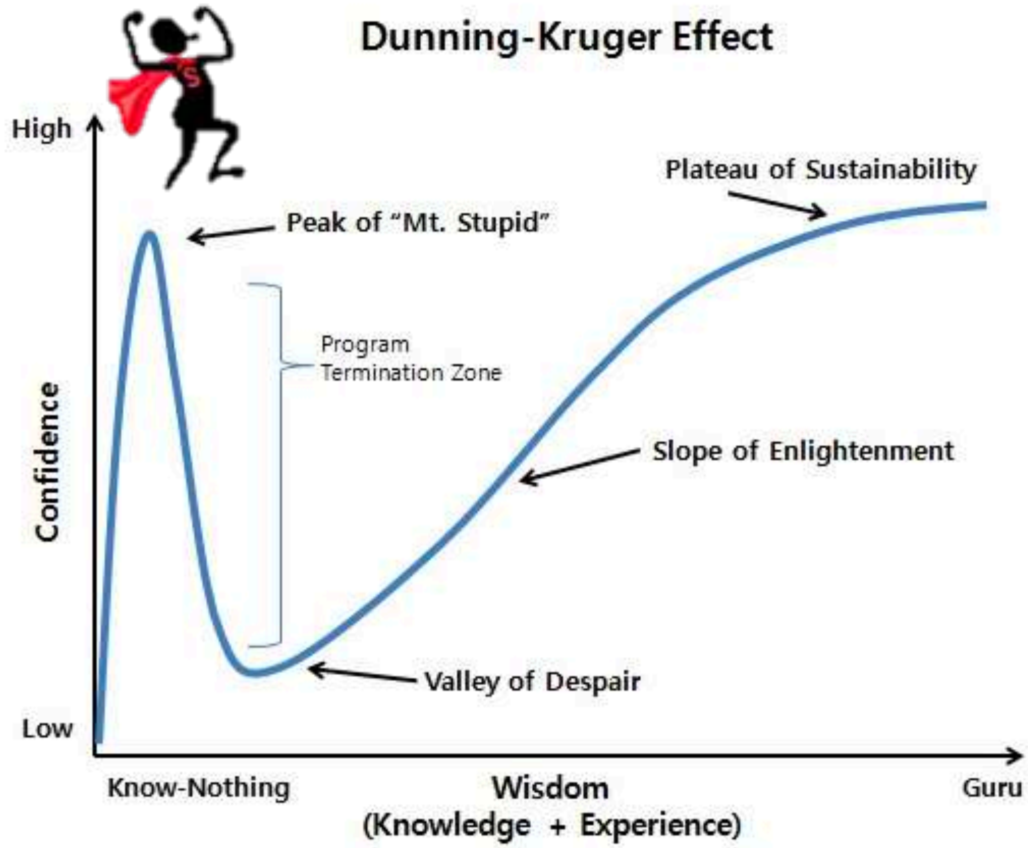
- Possible approach: the right action is the one that will make the agent more successful

$$\Downarrow$$

- Need for a method to measure success!

Performance measures

- Evaluates any given sequence of environment states and not the agent
- If the performance measure is related to the agent's level of satisfaction or if it is **based on the agent's opinion**, there may be **an illusion of perfect performance**
  - Ex: the student who leaves the test with the perception that he scored 10… but the "environment" evaluates as 5. ☹
- There is no single or fixed performance measure



Performance measures

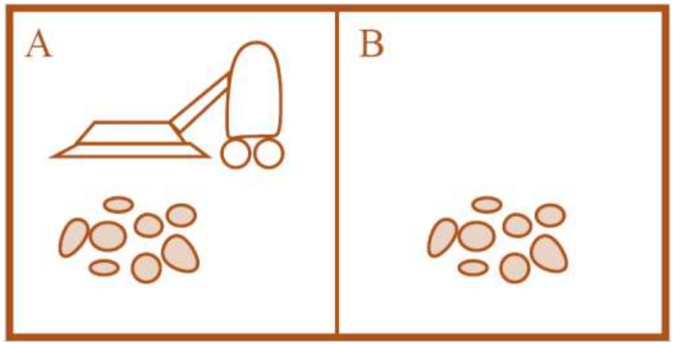
- There is no appropriate measure for all agents, tasks and situations: it is usually imposed by the designer who is building the agent
- Ex: possible performance measures for the vacuum cleaner
  - Amount of dirt removed in a single 8-hour shift (agent can clean everything, throw dirt back on the floor and clean everything)
  - Reward the agent for each clean room and penalize it for the electricity consumed and the noise generated
  - Observe "average cleanliness" over time

Performance measures

- In general, it is best to design performance measures:
  - According to the result really desired in the environment
  - Not necessarily related to the expected behavior of the agent
- What's better?
  - A fast, efficient agent that takes long breaks
  - A slow and continuously working agent
  - On average, the two perform the same!

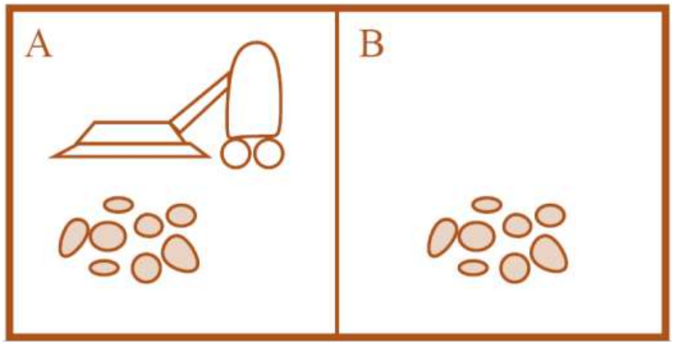
- Rational agent:
  - "For each possible sequence of perceptions, a rational agent must select an action that is expected to maximize its performance measure, given the evidence provided by the perception sequence and any internal knowledge of the agent"
- It depends on 4 factors:
  - The performance measure that defines the success criterion
  - The agent's previous knowledge of the environment
  - The actions the agent can take
  - The sequence of perceptions of the agent so far
    - Ex: can an agent that does not have an action on which it depends to solve the problem be considered rational?

Vacuum cleaner: rational agent?



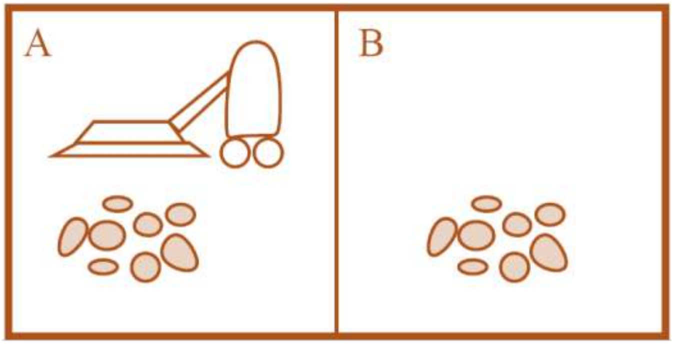
- Agent model:
  - **Environment knowledge:** the environment is known a priori, but the distribution of dirt and the initial position of the agent are not
  - **Actions**: Left, Right, Vacuum. Clean room remains clean and Vacuuming cleans the room completely (guaranteed result).
  - **Sequence of perceptions:** the agent successively perceives his position and if this position contains dirt
  - **Performance measure :** 1 point for each clean room in each time period up to 1,000 time periods
    - Note: Using the table above, after cleaning the agent swings from side to side
    - Rational Agent! (selects an action that is expected to maximize its performance measure)

Vacuum cleaner: rational agent?



- Agent model:
  - **Environment knowledge:** the environment is known a priori, but the distribution of dirt and the initial position of the agent are not
  - **Actions**: Left, Right, Vacuum. Clean room remains clean and Vacuuming cleans the room completely (guaranteed result).
  - **Sequence of perceptions:** the agent successively perceives his position and if this position contains dirt
  - **Performance measure:** 1 point for each clean room in each time period up to 1,000 time periods. 1 point penalty for each move left or right
    - **Irrational Agent !!! (with the set of possible actions, the agent does not maximize its performance measure)**

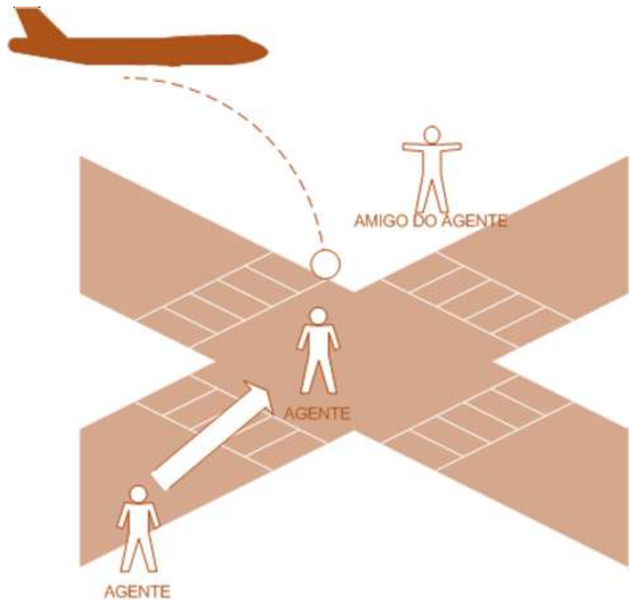
Vacuum cleaner: rational agent?



- Considering the same performance measure as the previous example:
  - 1 point for each clean room in each time period up to 1,000 time periods. 1 point penalty for each move left or right
- How to make the agent in the previous example rational?
  - Include NOP action?
  - But, how long to stay stopped?
  - What if a room gets dirty again?

Rationality x Omniscience

- Rationality is not the same as perfection !!!



- **Rationality:** maximizes the expected performance (any action given the sequence of perceptions so far)
- **Perfection:** maximizes real performance (depends on crystal ball or time travel!)

- If the environment is "fully" known:
  - The agent will always act correctly
  - Common in 1st generation robotic arms
- But, what happens if the environment changes over time? Or if the agent reaches unknown points in the environment?
  - A **rational agent** can not only collect information about its universe, but can also learn from this information in order to modify its future behavior
  - A learning agent can succeed in different environments!

Autonomy

- **Autonomous agent:** it is based on its own perceptions, and not only on the previous knowledge of its designer. Learns what can be done to compensate for partial or incorrect prior knowledge.
- **Rational agent:** must be autonomous!
  - Example: vacuum cleaner learns to predict where and when dirt will appear!

Summary

- Rationality != Omniscience
  - Perception may not provide all the necessary information
- Rational != Clairvoyant
  - The result of the action may not be as expected
- So rational != Success
  - Rational ⇒ exploration, learning, autonomy

How successful are agents?

- Successful agents divide the task of calculating the role of the agent into three distinct moments:
  - When the agent is being designed, part of the calculation is done by its designers;
  - When the agent is deliberating on his next action, the agent performs more calculations;
  - As it learns from experience, he performs even more calculations to decide how to modify his behavior

Specifying the problem:

Automated taxi (or Uber) driver

	P	E	A	S
Agent type	**Performance measure**	**Environment**	**Actuators**	**Sensors**
Taxi Driver	Safe, fast, lawful, comfortable travel, maximizing profits	Roads, other types of traffic, pedestrians, customers	Steering, accelerator, brake, signal, horn, display	Cameras, sonar speedometer, GPS, odometer, accelerometer, engine sensors, keyboard

Environments

Environments

Agent type	Performance measure	Environment	Actuators	Sensors
Medical diagnostic system	Healthy patient, correct diagnosis (FP, FN, TN, TP), minimize costs	Patient, hospital, team	Display questions, tests, diagnostics, treatments, indications	Keyboard input for symptoms, findings, patient responses
Parts selection robot	Percentage of parts in correct trays	Conveyor belt with parts; trays	Articulated arm and hand	Camera, articulated angle sensors

Environment properties

- Fully observable vs. partially observable
  - Do the agent's sensors allow access to the complete state of the environment?
    - Ex: robot football with global vision (c.o.) x local vision (p.o.)
- Deterministic vs. stochastic
  - Is the next state totally determined by the current state and the action taken by the agent?
    - Ex: vacuum cleaner (deterministic) x robot taxi driver (stochastic)
- An environment is said to be uncertain if it is not fully observable or is not deterministic!

Environment properties

- Static vs. dynamic
  - Does the environment change while the agent is deliberating?
    - Ex: chess (est.) X robot football (din.)
  - Semi-dynamic: the environment itself does not change, but the agent's performance level changes. Ex. Chess with time counting
- Discrete vs. continuous
  - Does the environment have a finite number of states? Do the sensors or actuators use discrete measurements? Is the time discreet?
    - Ex: chess (disc with respect to states and sensors) x taxi driver (cont. With respect to states and sensors)
  - Ultimately, everything that will be represented on the computer is limited by the accuracy of the system (discrete)

Environment properties

- Single agent x multiagent
  - How many agents are there in the environment? In the case of a multi-agent, do agents compete or cooperate?
    - Ex: crossword puzzles (single) x tic-tac-toe (mult.)
- Episodic vs. sequential
  - Does the next episode depend on the actions taken in the previous episodes? In the sequential environment, short-term actions can have a long-term effect!
    - Ex: Rolling a dice (ep.) X taking the next step (seq.)
- Known vs. unknown (refers to knowledge about the physics of the environment)
  - Do the agent's sensors allow access to the complete state of the environment?
    - Ex: robot football with global vision (c.o.) x local vision (p.o.)

Terror of agents

- Partly observable
- Multiagent
- Stochastic
- Sequential
- Dynamic
- Continuous
- Unknown

Terror of agents

A robor soccer match

Exercises

Environment properties

Environment	Observability	Determinísm	Episode	Static/Dynamic	Discrete	Agent
Crossword puzzle game	Completely	Deterministic	Sequential	Static	Discrete	Single
Chess with a clock	Completely	Deterministic	Sequential	Semi	Discrete	Multi
Poker	Partially	Estocástico	Sequential	Static	Discrete	Multi
Driving a taxi	Partially	Estocástico	Sequential	Dynamic	Continuous	Multi
Medical diagnostic	Partially	Estocástico	Sequential	Dynamic	Continuous	Single

- And a robotic arm playing chess? What changes?

Environment properties

Environment	Observability	Determinísm	Episode	Static/Dynamic	Discrete	Agent
Image analysis	Completely	Deterministic	Episodic	Semi	Continuous	Single
Parts selection robot	Partially	Estocástico	Episodic	Dynamic	Continuous	Single
Refinery controller	Partially	Estocástico	Sequential	Dynamic	Continuous	Single
Robot Football Game	Partially	Estocástico	Sequential	Dynamic	Continuous	Multi

Agent Structure

- AI's job is to design the agent program that implements the agent's role - which maps perceptions into actions.

- **Architecture:** computing device with physical sensors and actuators

- **Agent** = Architecture + Program.

Agent programs

- The AGENT-DIRECTED-BY-TABLE program is invoked with each new perception and returns one action at a time. It keeps track of the sequence of perceptions using its own private data structure:

AGENT-DIRECTED-BY-TABLE (perception) function **returns** an action
static variables:
perceptions: a sequence, initially empty
table: a table of actions, indexed by sequences of perceptions, at first completely specified

attach perception to the end of perceptions
action ← ACCESS (perceptions, table)
return action

$$\sum_{t=1}^{T} |P|^{T}$$

Requires a table that contains the appropriate action for all possible strings of perceptions.

- Find out how to write programs that, as far as possible, produce rational behavior from a small amount of code, rather than from a large number of table entries
  - Example: Square root table x Netwon method

0	0	31	961
1	1	32	1024
2	4	33	1089
3	9	34	1156
4	16	35	1225
5	25	36	1296
6	36	37	1369

If $f(x) = 0$ has only one root in the range $[a, b]$ and if nor $f'(x)$ nor $f''(x)$ cancel each other in that interval, chose x_0 like that none of the two numbers a and b for which $f(x)$ and $f''(x)$ have the same signal, then:

$$x_{k+1} = x_k - \frac{f x_k}{f' x_k}$$

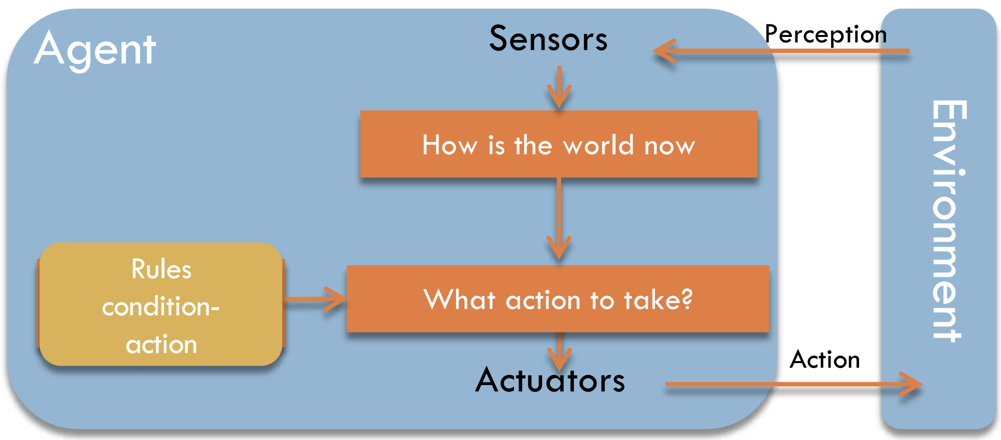
is closer to the root than x_0.

Types of agents

- Simple reactive agents
- Model-based reactive agents
- Goal-based agents
- Utility-based agents
- Learning-based agents

Simple reactive agents

- Selects actions based on current perception, ignoring the rest of the perception history

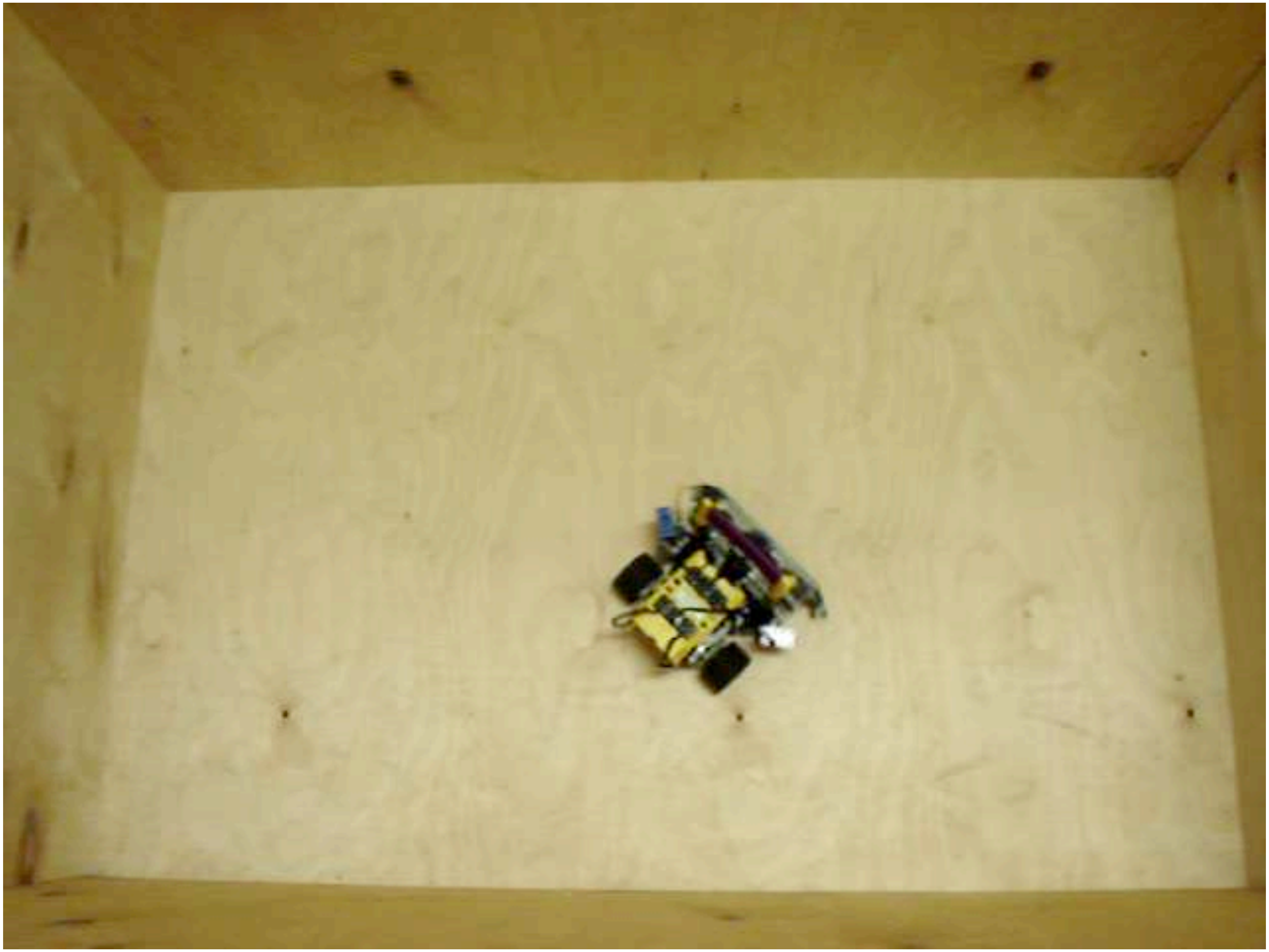


AGENT-REACTIVE-SIMPLE (perception) function **returns** an action
static variables:
rules: a set of condition-action rules

> *state* ← INTERPRETING-ENTRY(*perception*)
> *rule* ← CORRESPONDING-RULE (*state*, *rules*)
> *action* ← RULE-ACTION [*rule*]
> **return** action

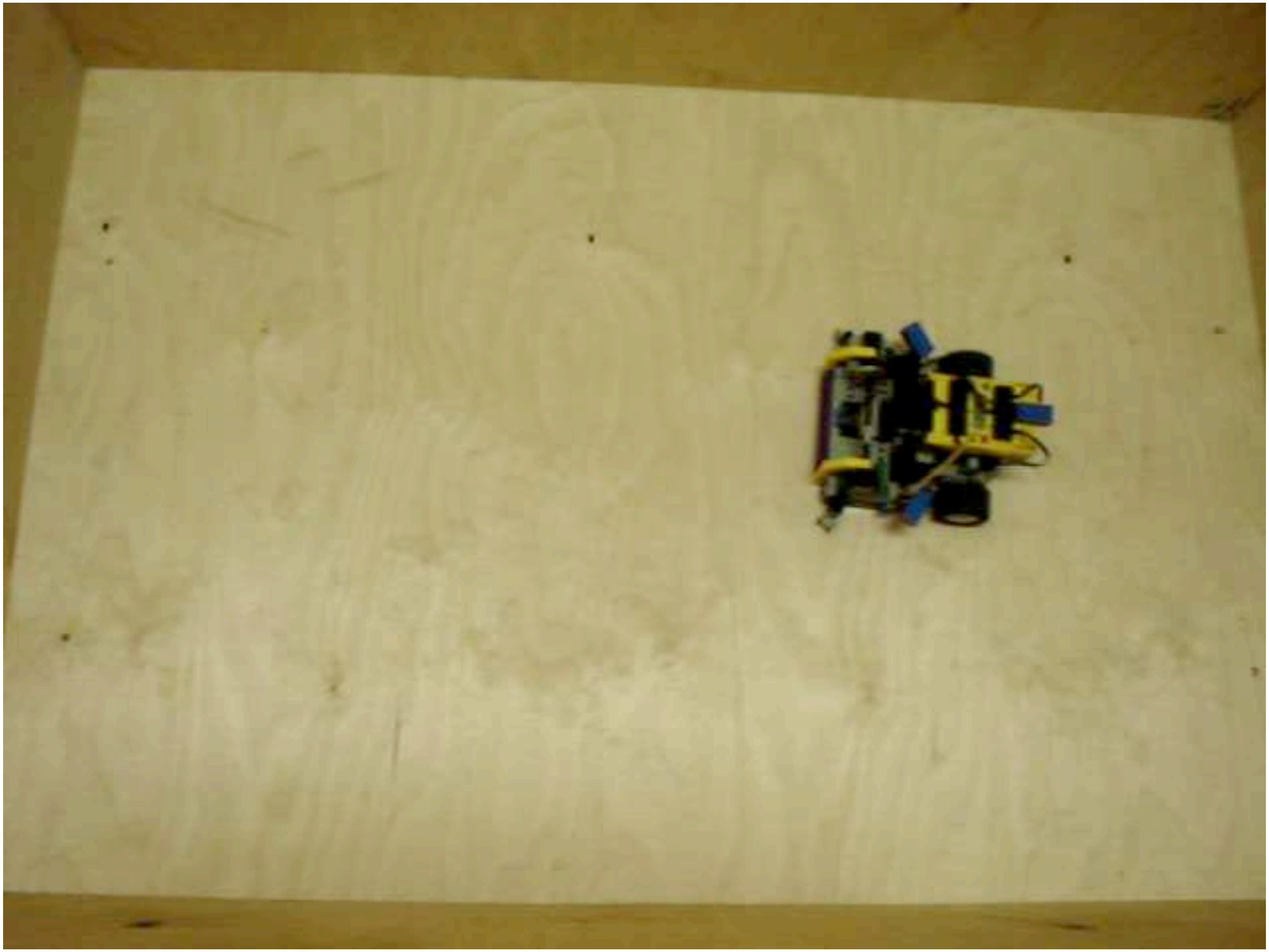
Simple reactive agents

Example 1



Simple reactive agents

Example 2



Simple reactive agents

- Example: reactive vacuum cleaner

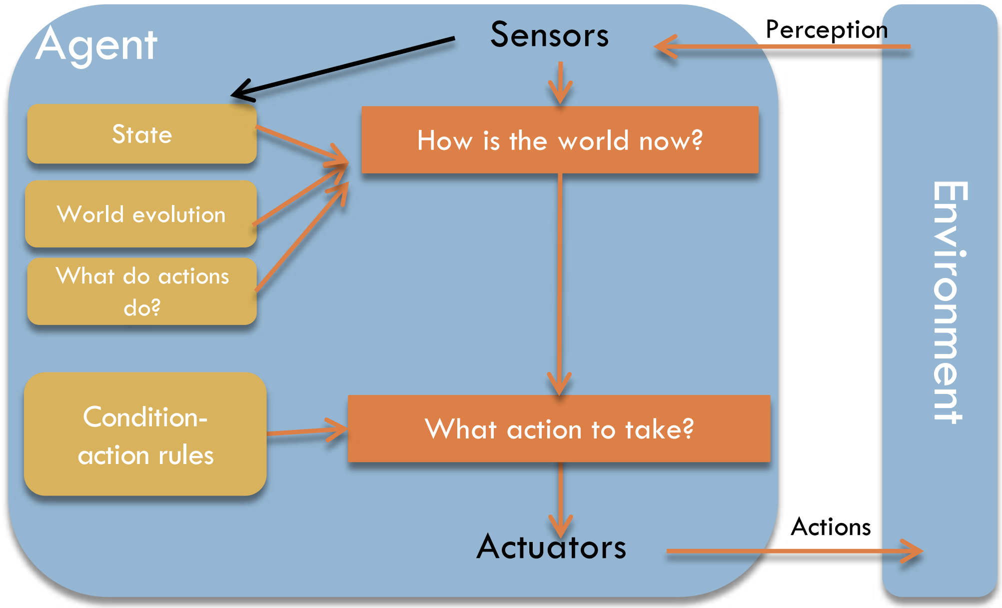
function AGENT-VACUUM-CLEANER-REACTIVE([*position*, *state*])
returns na action
if state = sujo **then return** Clean
else if *position* = A **then return** Right
else if *position* = B **then return** Left

- Problem: environment must be completely observable! Imagine the vacuum cleaner without the localization sensor!

Model-based reactive agents

- Agent controls the part of the world that he cannot see now (agent maintains an internal state that depends on the history of perceptions)
  - Ex: taxi driver will change lanes and may not momentarily see some cars around him

Model-based reactive agents



function AGENT-REACTIVE-BASEAD-ON-MODELS(*preception*) **return** an action

static variables:

state: a description of the current state of the world

model: a description of how the next state depends on the current state and the action

rules: a set of condition-action rules

action: the most recent action, initially none

state ← UPDATE-STATE (*state*, *action*, *perception*, *model*)

rule ← REGRA-CORRESPONDENTE (*state*, *rule*)

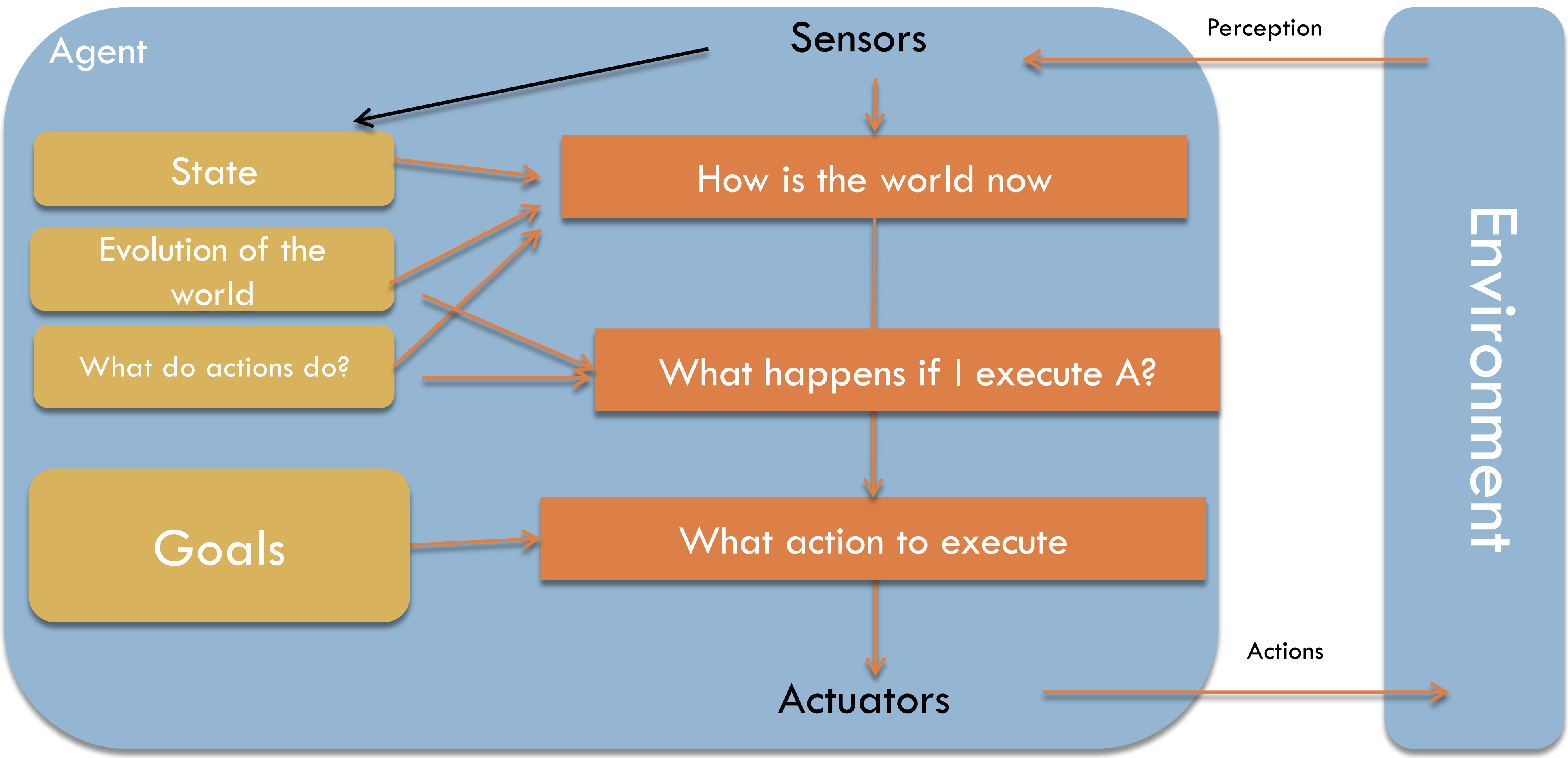
action ← AÇÃO-DA-REGRA [*rule*]

return action

Goal-based agents

- The agent combines its goal with information about the results of possible actions to choose actions that achieve the goals
  - Example:
    - Taxi at a road junction: turn left, right or go ahead?
  - Need for search and planning: A.I. subfields dedicated to finding sequences of actions that achieve the goals of the agent

Goal-based agents

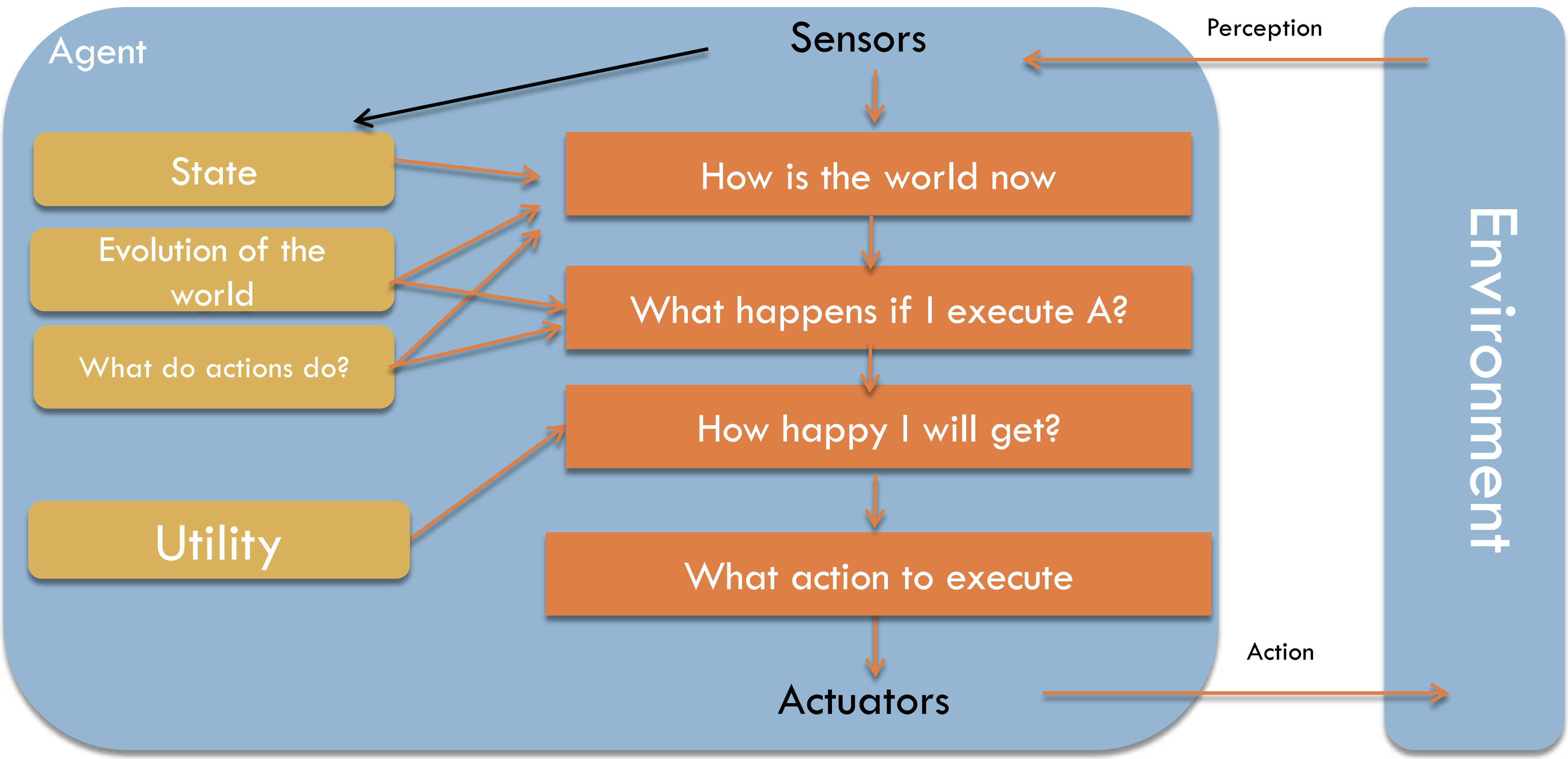


Goal-based agents

Utility-based agents

- There are many sequences of actions that lead the agent to its goal. Some are faster, safer, more economical, etc.
- Goals simply allow for a crude binary distinction between "happy" and "unhappy" states
- **Utility-based agents** use a performance measure (utility function) that allows a comparison between different states of the world, allowing you to select the sequence of actions
- Good for conflicting goals

Utility-based agents

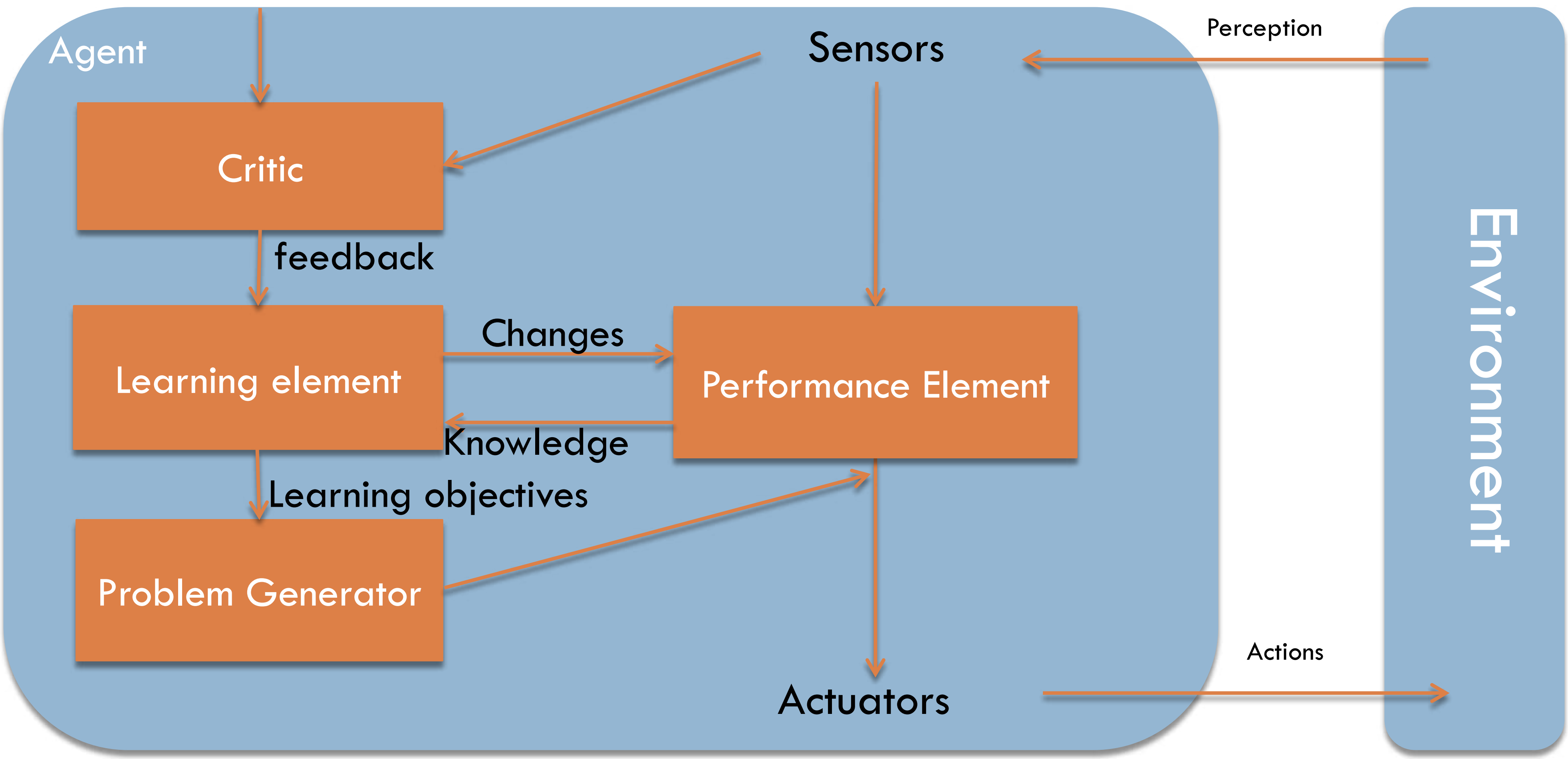


Learning agents

- A learning element uses feedback on how an agent is functioning and determines how the performance element should be modified to work better in the future

Learning agents

Performance standard



Learning agents

Example

Summary

- Agents interact with the environment through sensors and actuators
- The role of the agent describes what the agent does in all circumstances
- The performance measure assesses the sequence of the environment
- A perfect rational agent maximizes the expected performance
- Agent programs implement agent roles
- PEAS descriptions define the task environment

Extra-class activities

Lecture 3

- Activities
  - Reading:
    - RUSSELL, S. NORVIG, P. Inteligência Artificial. 3ª edição. Capítulo 2.
  - Recommended exercises:
    - 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 2.10, 2.11, 2.12

References

Lecture 3

- RUSSELL, S. NORVIG, P. Inteligência Artificial. 3ª edição.
- Simões, A. S. Slides de aula: IA para Controle e Automação.