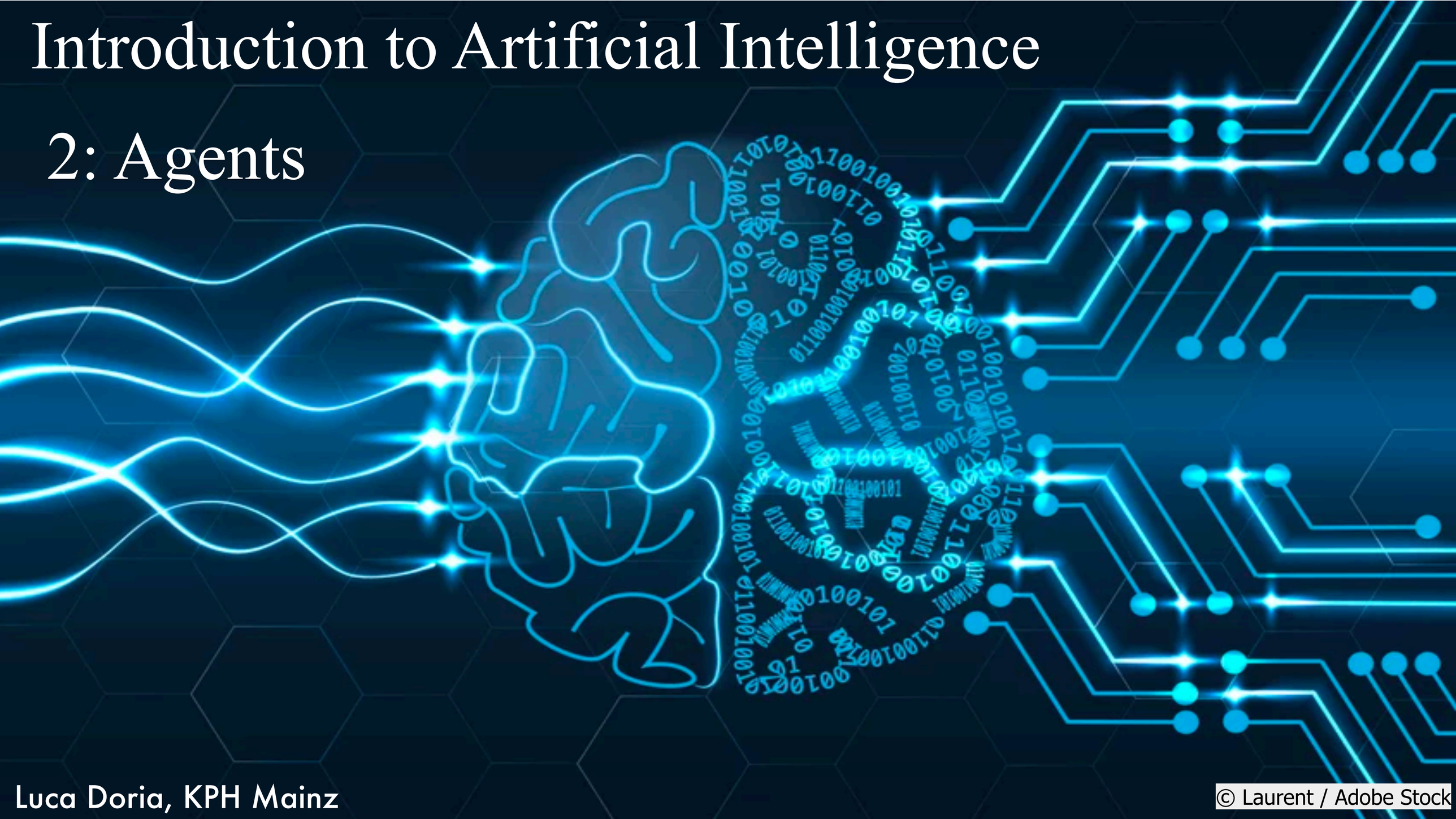
Introduction to Artificial Intelligence

2: Agents



Luca Doria, KPH Mainz

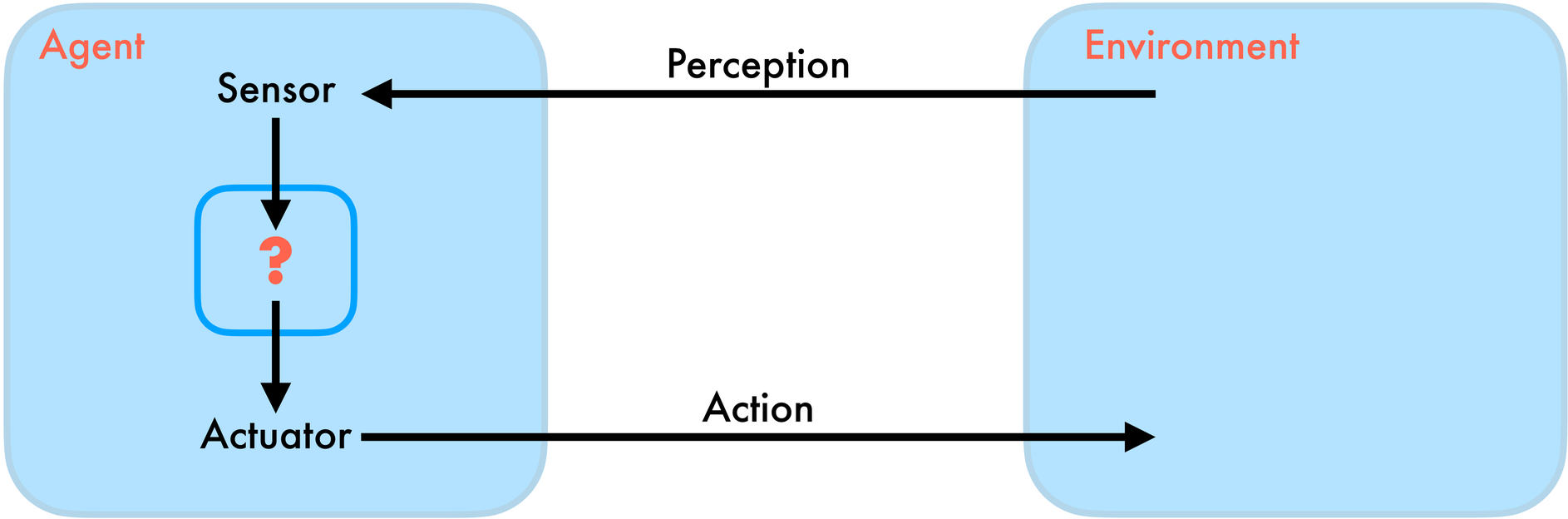
(Rational) Agents

- The “Agent”: a model for AI.
- What does it mean “rational” or “intelligent”?
- Agents act/interact in/with and environment.

- Develop a set of abilities an agent should have for being “intelligent”.

- Types of Agents.
- Types of Environments.

Interaction with the Environment



The agent perceives the environment through sensors.
Acts on the environment though actuators.
Note that the agent is not independent from the environment!

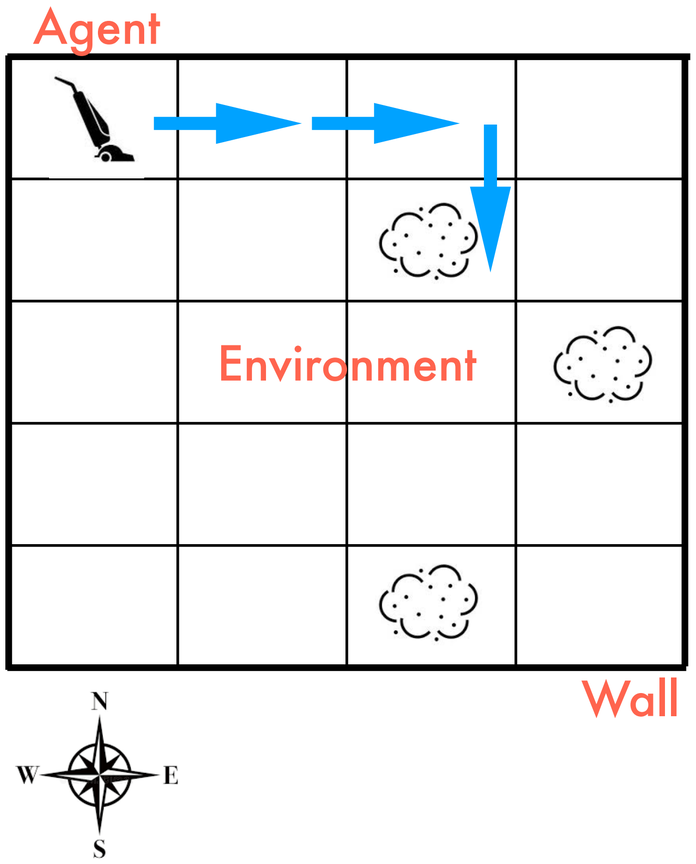
Intelligent Agents

Consequentialism: evaluate the agent by the consequences of its actions.

Design of a performance measure:
- Should evaluate what is achieved in the environment,
- Should not evaluate what we think should be the correct agent's behaviour.

The "rationality" of an agent depends on:
- The performance measure that defines the success of the actions,
- The prior knowledge of the environment,
- The actions that can be performed,
- The agent percept sequence at the moment of the evaluation.

The “vacuum-cleaner” example



Action (actuator): move (N,S,W,E), on, off.
Perception (sensor): dirty, clean, wall

Is this agent “intelligent”? It depends (see previous criteria).

We need:

- Performance measurement:
  - count of the cleaned squares,
  - energy use,
  - squares cleaned/hour, ….
- Knowledge of the environment: know where the “walls” are,
- Available actions: move, on off,
- Correct Perception: know where it is in the environment and know of the square is dirty or clean

Limits of Rationality and learning

Omniscient agent: an agent which knowns the outcomes of its actions.
—> Impossible in reality.

A rational agent will try “only” to maximise the expected performance given the results of the percepts and the previous experience.

Example: crossing the street even without incoming cars in sight seems an OK strategy. Imagine to be hit by a meteorite: would you say that your action was irrational?

A rational agent should not only be capable of gather informations form the environment, it should also be able to learn.

Limits of Rationality and learning

Summary: The Ideal Rational Agent

For each possible percept sequence, a rational agent should choose an action which is expected to maximise a performance measure, given the data provided by the percept sequence and its the built-in knowledge.

Specifications

The PEAS specification for an agent:

Performance: set an observable goal for the agent

Environment: how is the environment made?

Actuators: how does the agent actively interact with the environment?

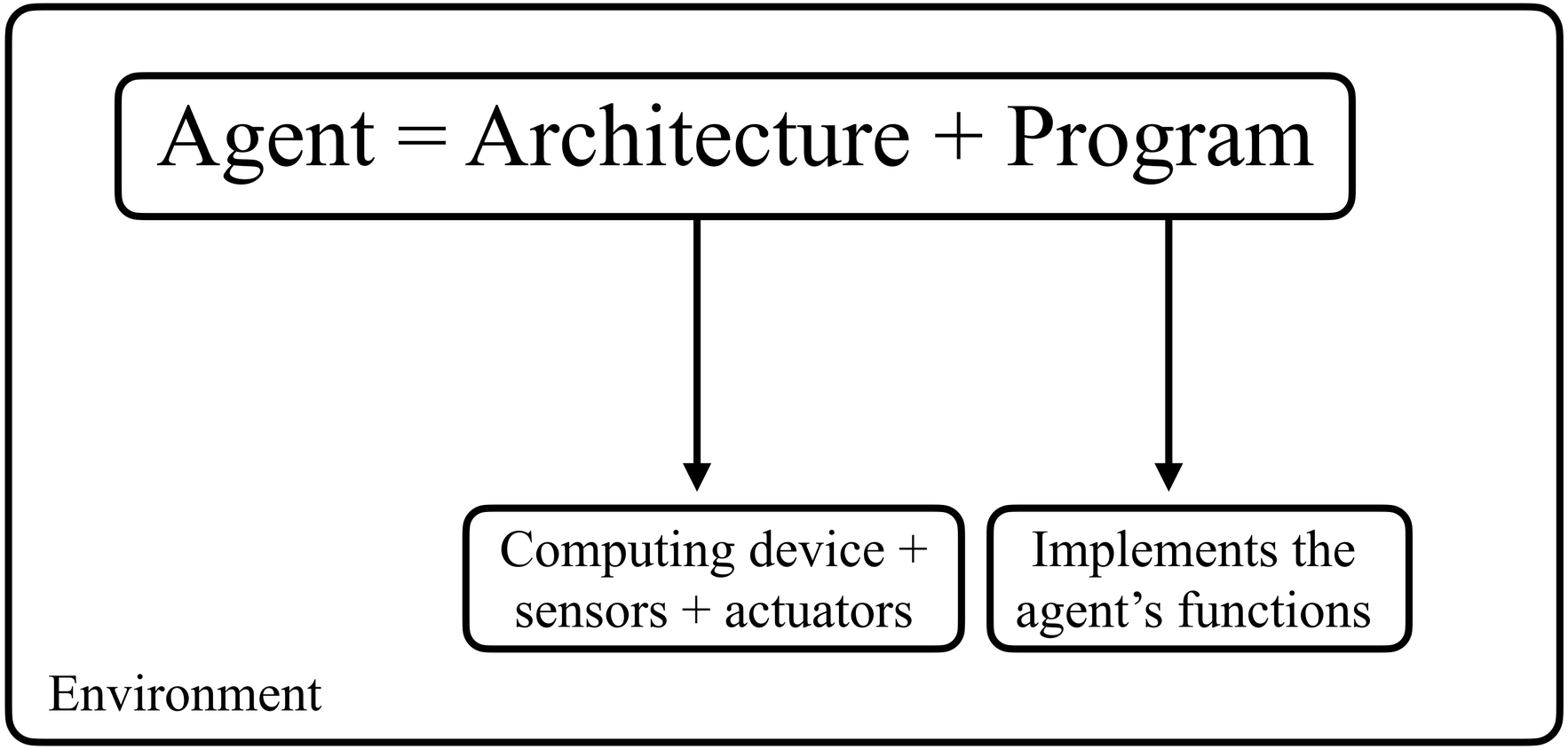
Sensors: how does the agent experience the environment?

EXERCISE: let's describe the PEAS for a self-driving car

Examples of Rational Agents

Agent Type	Performance Measure	Environment	Actuators	Sensors
Mediacal Diagnosis System	Healty Patient, reduced costs	Patient, hospital, staff	Display of questions, tests, diagnoses, treatments.	Touchscreen, voice, entry of symptoms and findings
Satellite Image analysis system	Correct categorisation of objects, terrain	Orbiting satellite, downlink, weather	Display of scene categorization	High-resolution digital camera
Part-picking robot	Percentage of parts in correct bins	Conveyor belt with parts, bins	Jointed arm and hand	Camera, tactile and point angle sensors
Refinery Controller	Purity, yield, safety	Refinery, raw materials, operators	Valves, pumps, heaters, stirrers, displays	Temperature, pressure, flow, chemical sensors
Interactive English Tutor	Student's score on test	Students, testing agency	Display of exercises, feedback, speech	Keyboard, voice

The Structure of Rational Agents



The agent functions are implemented in a program which runs on an architecture.
The architecture provides an interface with the environment.

Types of Rational Agents

-Table Agents

-Simple Reflex Agents

-Model-based Reflex Agents

-Goal-based Agents

-Utility-based Agents

-Learning Agents

Table-Driven Agents

- Simplest kind of agent.
- Takes 1 input, consults a table which gives the appropriate action.
- Always works as intended, as long as the table is correct!

```
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(precept, table)
return action
```

Table-Driven Agents

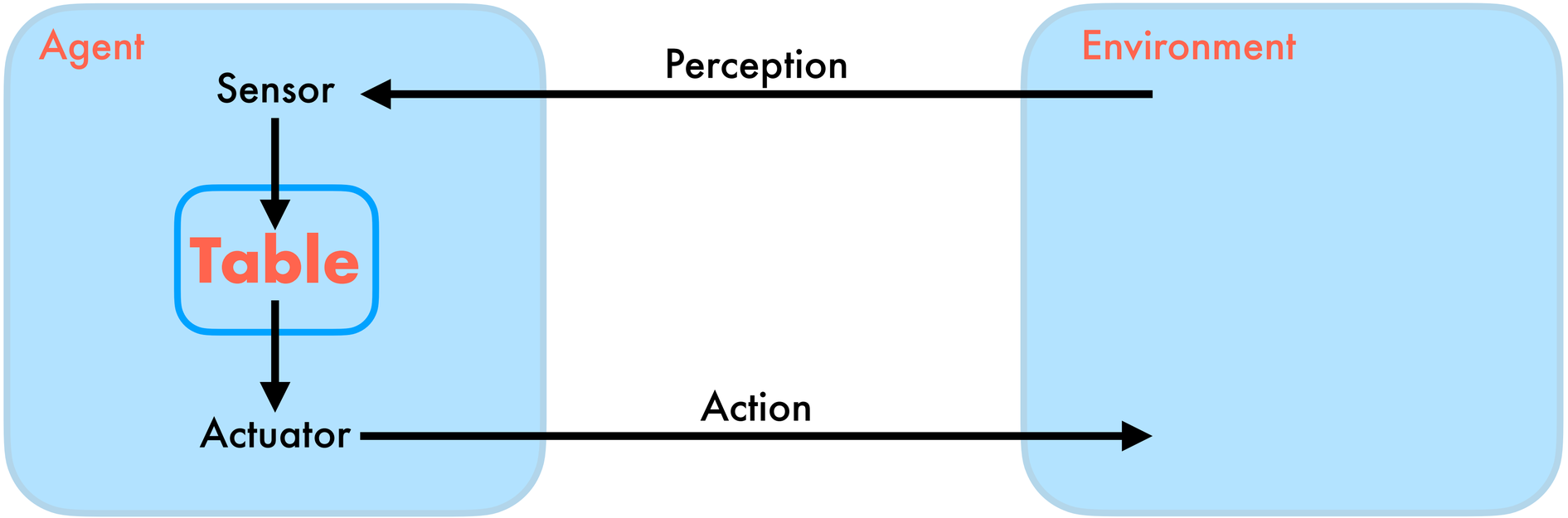


Table-Driven Agents: are they practical?

We have to prepare a table containing every action to undertake for every possible input. How large should be the table?

P: Number of possible percepts
T: Total number of percepts the agent will receive (number of "time steps").

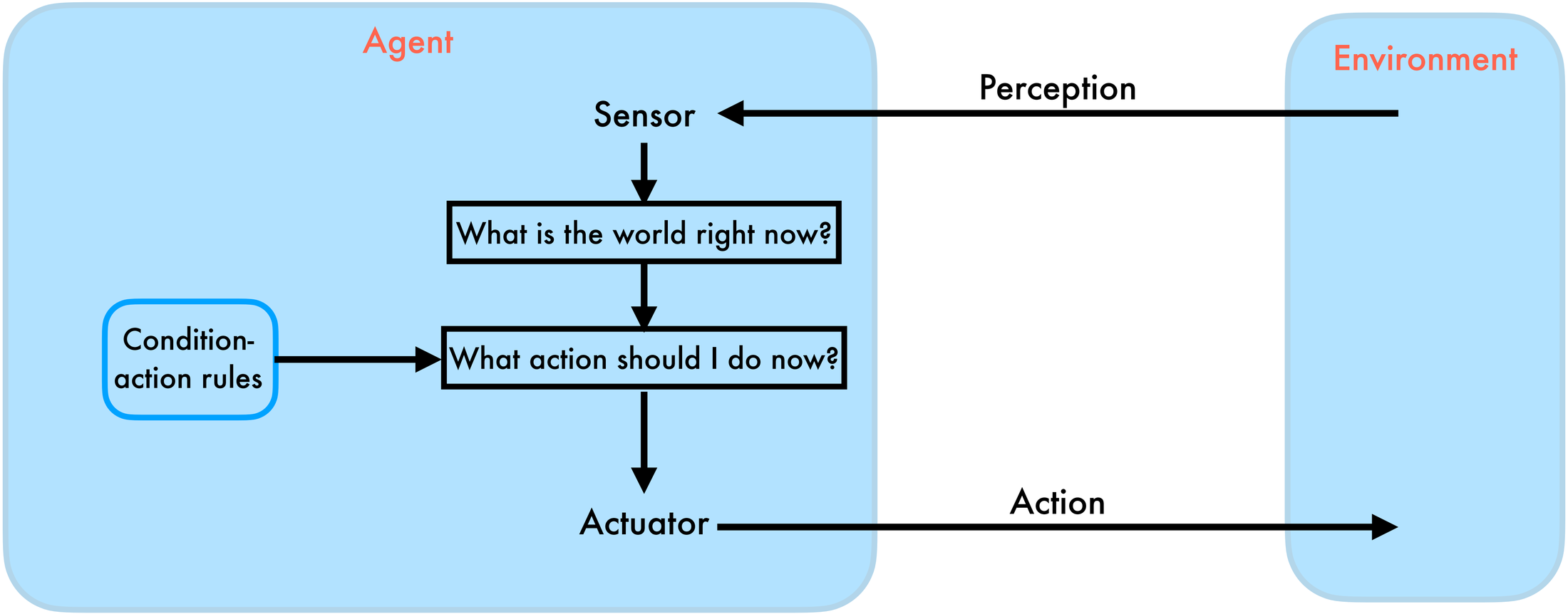
Table size for every possible percept sequence: $\#\text{Table} = \sum_{i=1}^{i=T} P^i$ (!)

Vacuum-cleaner example: P=2 (clean/dirty), T=20 (squares):

#Table = $2^1+2^2+\ldots+2^{20} = 2.097.150$

If e.g. P=10, #Table ~ 10^{10}

Simple Reflex Agents



Notation:
Rectangles: Agent's internal state of the decision process
Ovals: Background information used in the process

- Action decided only on current percept (no history).
- Often input not directly usable: interpretation/preprocessing needed.

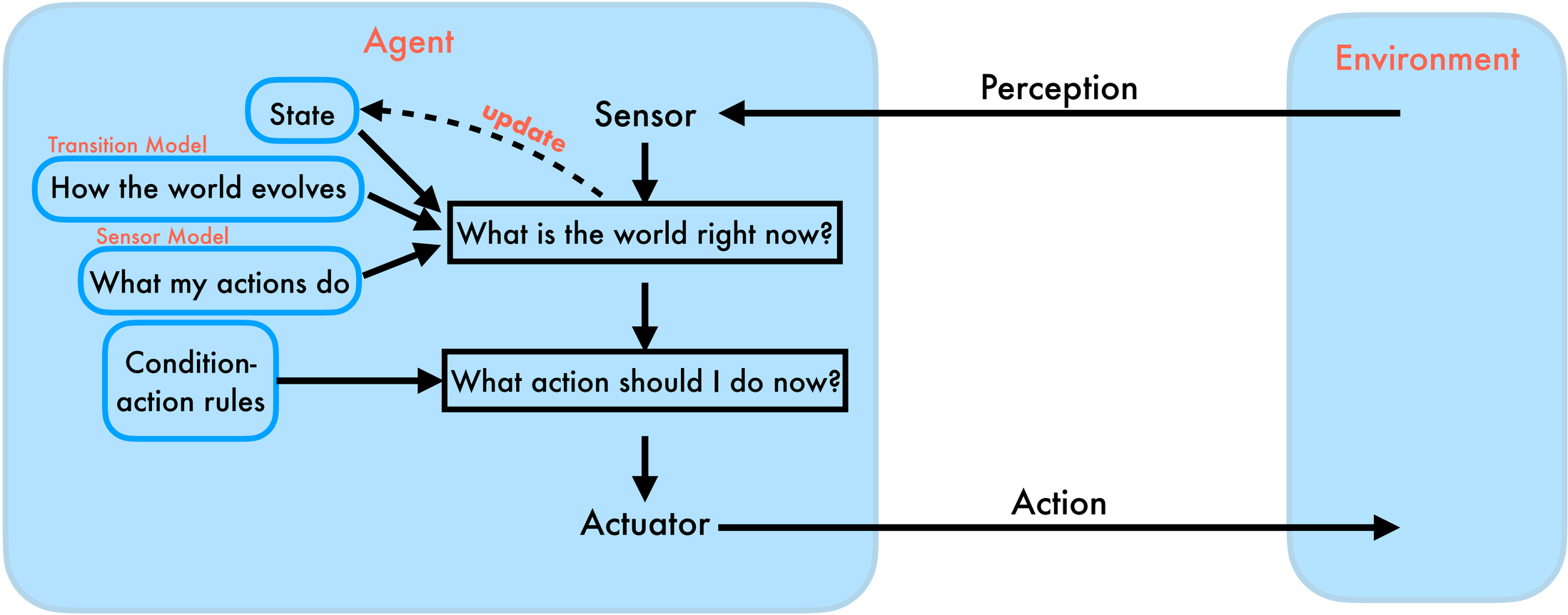
Simple Reflex Agents

```
function SIMPLE-REFLEX-AGENT(percept) returns an action
    persistent:
        rules, a set of condition-action rules
state <—- INTERPRET-INPUT(precept)
rule <—- RULE-MATCH(state, rules)
action <—- rule.ACTION
return action
```

Simple Reflex Agents can be easily trapped in infinite loop-states in cases of partial observability.
We need an improvement for coping with partial observability —> Model-based Agents

Imagine the vacuum-cleaner agent without location sensor (only a “dirt” sensor)

Model-Based Reflex Agents



- Internal state: encodes a (partial) history of what's happened before.
- The behaviour of the agent depends also from the state, not only on the present percept.
- The construction of the internal state needs a suitable representation.

Model-Based Reflex Agents

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

persistent:

state, the agent's current conception of the world state

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

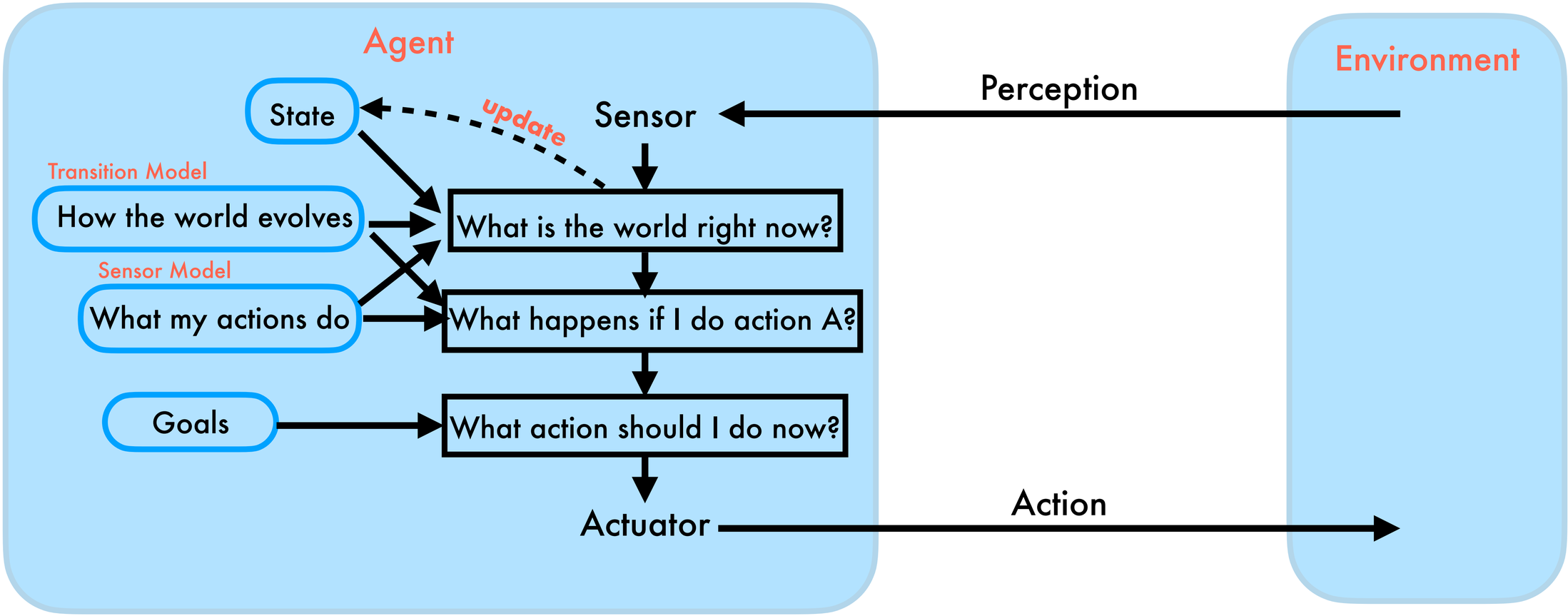
sensor_model: a description of how the current world state is reflected in the agent's percepts.

rules: a set of condition-action rules.

action: the most recent action (initially, none).

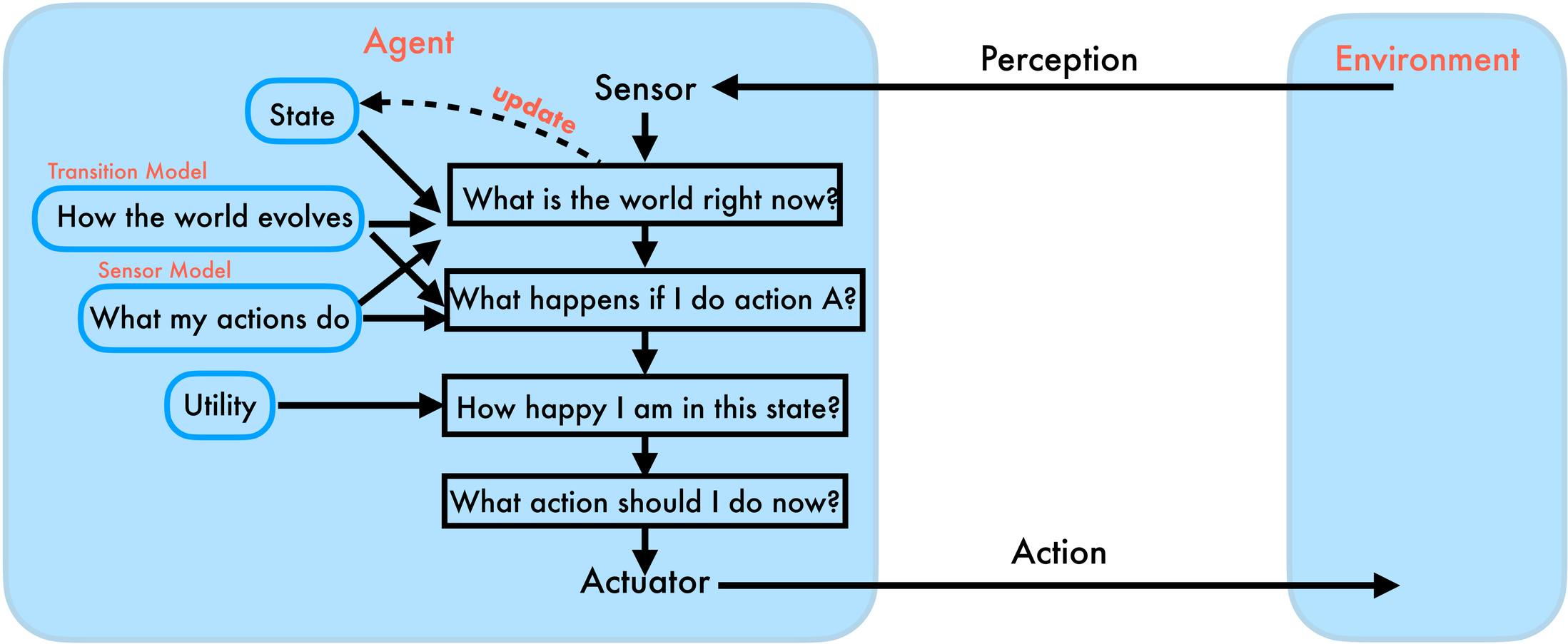
state <—- UPDATE_STATE(state, action, percept, transition_model, sensor_model)
rule <—- RULE-MATCH(state, rules)
action <—- rule.ACTION
return action

Model/Goal-Based Reflex Agents



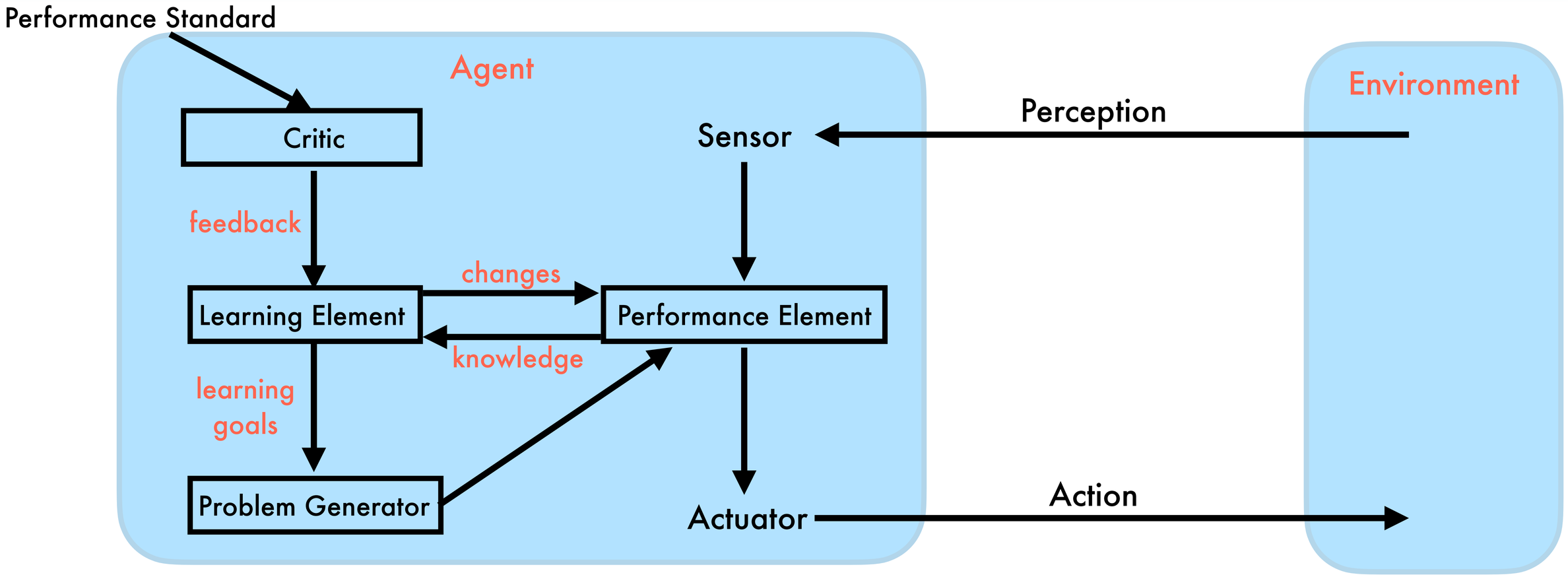
- Percepts alone might not be sufficient to decide the action.
- This is because the correct action depends on the goals.
- The action selection might require **search** or **planning** algorithms.

Model/Utility-Based Reflex Agents



- A utility function maps a state (or a sequence of states) onto a real number.
- The agent can also use these numbers to weigh the importance of competing goals.

Learning Agents



- Can improve over time.
- Can start with an initially zero-knowledge base.
- Can operate in initially unknown environments.

Learning element: makes improvements

Performance element: selects actions

Critic: determines the performance

Problem Generator: Suggests action for formative experience

Summary

- An agent is an object that can perceive and act.
- It consists of an architecture and a program
- An ideal rational agent always takes the best action for maximising its performance given the percept sequence and its knowledge if the environment.
- The program maps from a percept to an action.
- Many agent designs can be implemented:
- Reflex, Goal-Based, Utility-Based, Learning…
- Environments can be very different in terms of demands to the agents.
- Environments can be fully/partially observable, deterministic/stochastic, strategic, static/dynamic, discrete/continuous, single/multi-agent…