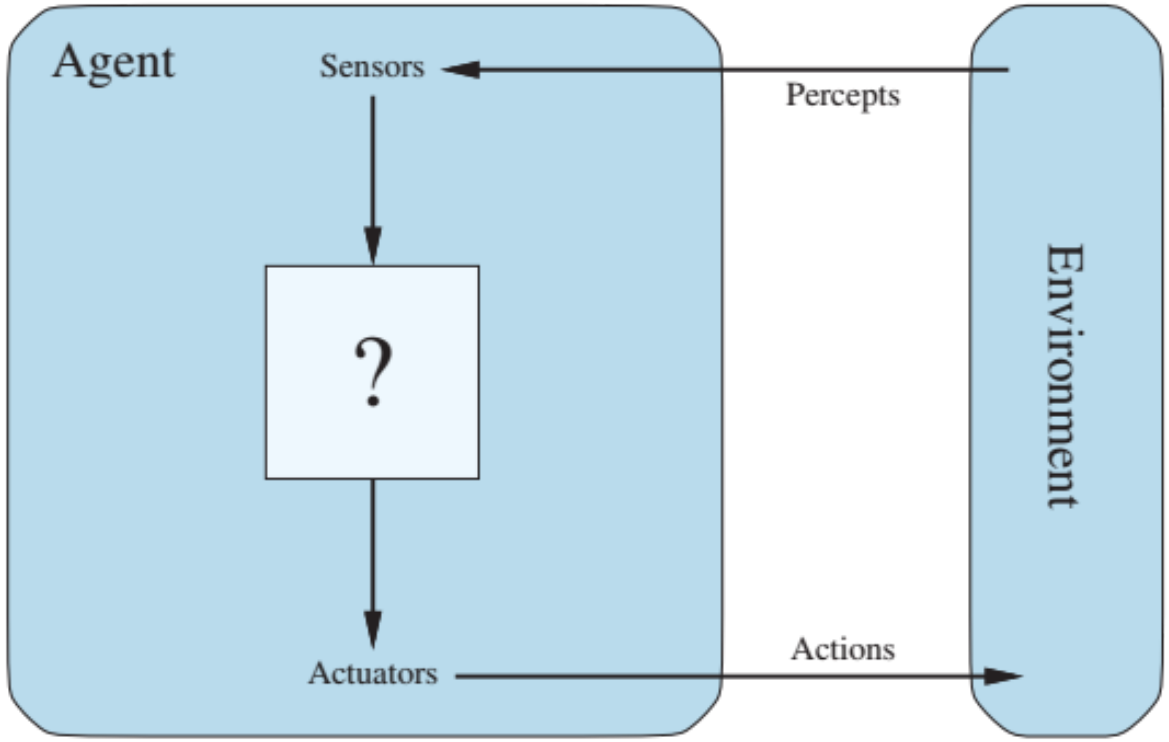
Lecture 2 - Intelligent Agents

Gleb Sizov
Norwegian University of Science and Technology



Agent

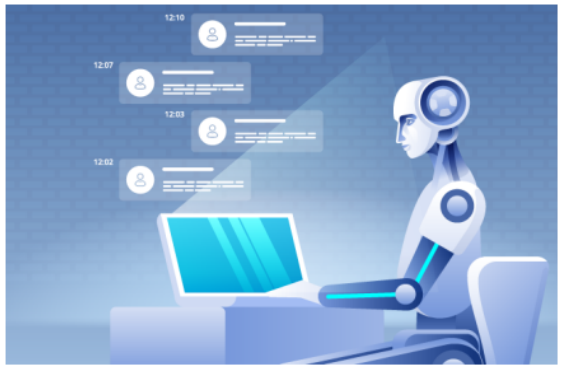
Human agent

hardware agent

software agent

(a human driver)

(a robot driver)

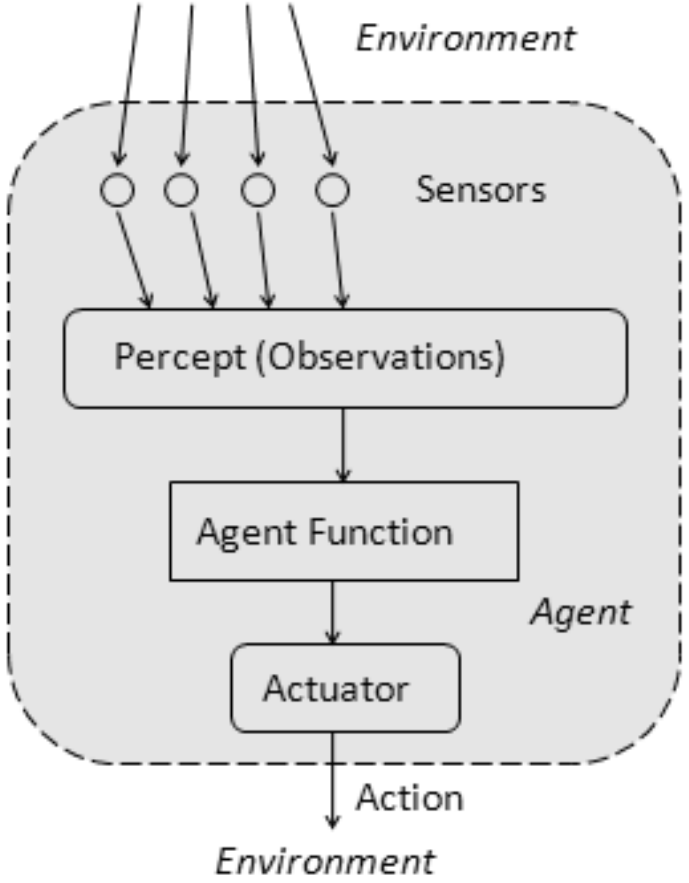


(a chatbot. www.callcentrehelper.com)

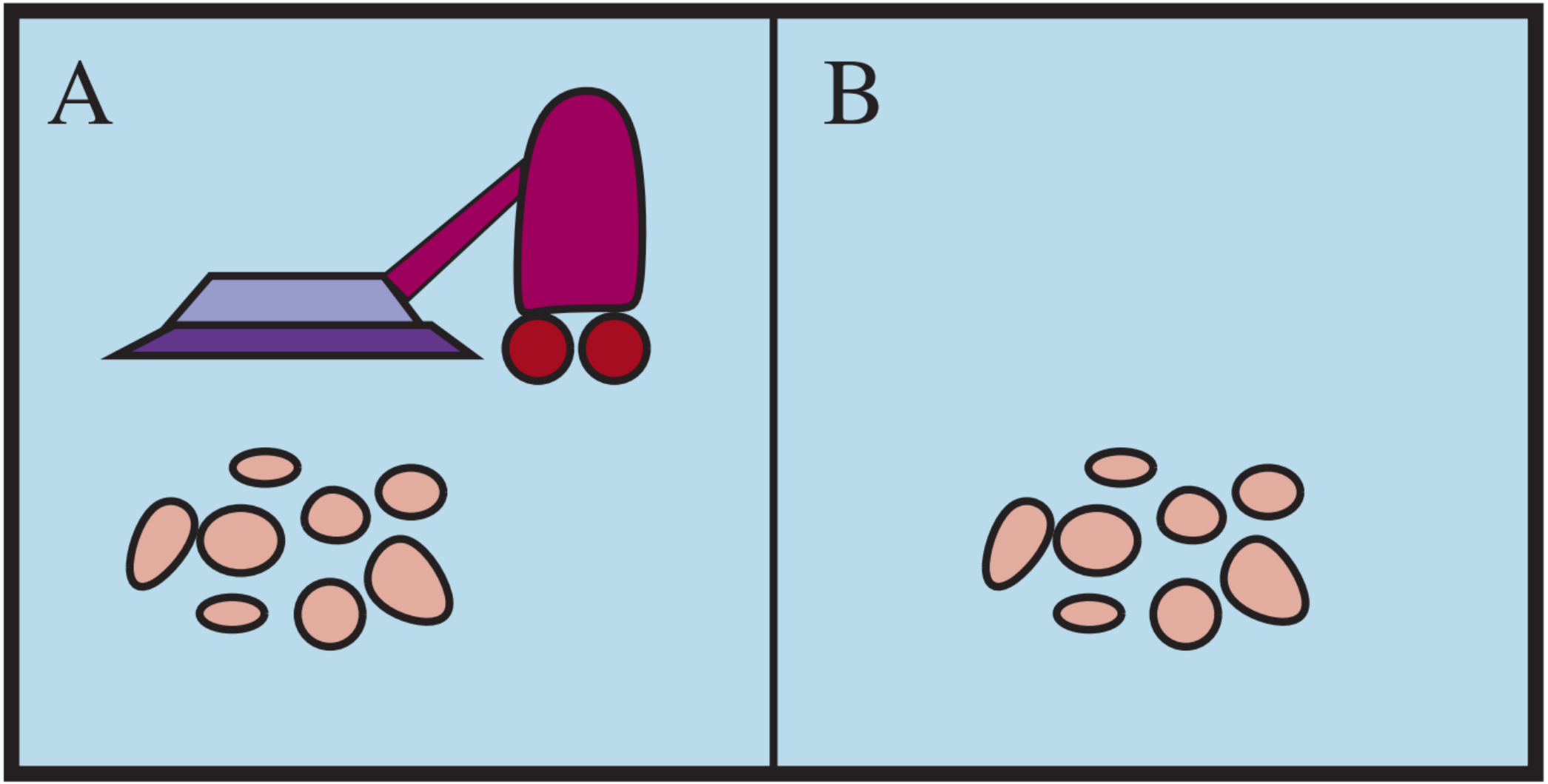
Agent behaviour

Agent function - math description of behaviour, maps percept sequence to an action

Agent program - concrete implementation of agent function, runs within a physical system



Example: Vacuum-cleaner world



Tabulation of agent function

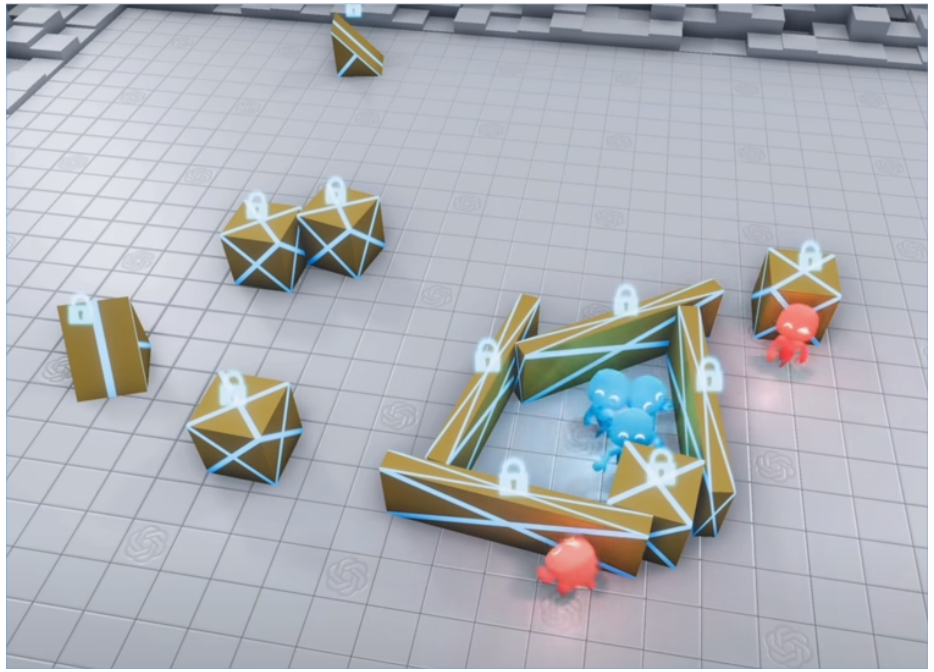
Percept sequence	Action
$[A, Clean]$	$Right$
$[A, Dirty]$	$Suck$
$[B, Clean]$	$Left$
$[B, Dirty]$	$Suck$
$[A, Clean], [A, Clean]$	$Right$
$[A, Clean], [A, Dirty]$	$Suck$
$\vdots$	$\vdots$
$[A, Clean], [A, Clean], [A, Clean]$	$Right$
$[A, Clean], [A, Clean], [A, Dirty]$	$Suck$
$\vdots$	$\vdots$

> What is the right way to fill out the table?

Doing the right thing - rational agent

Consequentialism - evaluate agents's behaviour by its consequences.
Performance measure - evaluates sequence of environment states

> Better to design performance measures according to what one wants to achieve, rather than how the agent should behave.



OpenAI Plays Hide and Seek…and Breaks The Game!

Rational agent maximizes its expected performance

Depends on:

- Performance measure
- The agent’s prior knowledge of the environment.
- The actions that the agent can perform.
- The agent’s percept sequence to date.

Rationality vs omniscient vs perfect

Rational $\neq$ *omniscient* agent knows the actual outcome of its actions

Rational $\neq$ *perfect* agent maximizes actual performace

Rationality requires information gathering and exploration

Doing actions in order to modify future percepts.

Rationality requires learning and autonomy

Learning agent modifies knowledge of the environment from experience.

Autonomous agents learns what it can to compensate for partial or incorrect prior knowledge.

Task environment

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

Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi driver	Safe, fast, legal, comfortable trip, maximize profits, minimize impact on other road users	Roads, other traffic, police, pedestrians, customers, weather	Steering, accelerator, brake, signal, horn, display, speech	Cameras, radar, speedometer, GPS, engine sensors, accelerometer, microphones, touchscreen

Task environmnet - Medical diagnosis

Agent Type	Performance Measure	Environment	Actuators	Sensors
Medical diagnosis system	Healthy patient, reduced costs	Patient, hospital, staff	Display of questions, tests, diagnoses, treatments	Touchscreen/voice entry of symptoms and findings

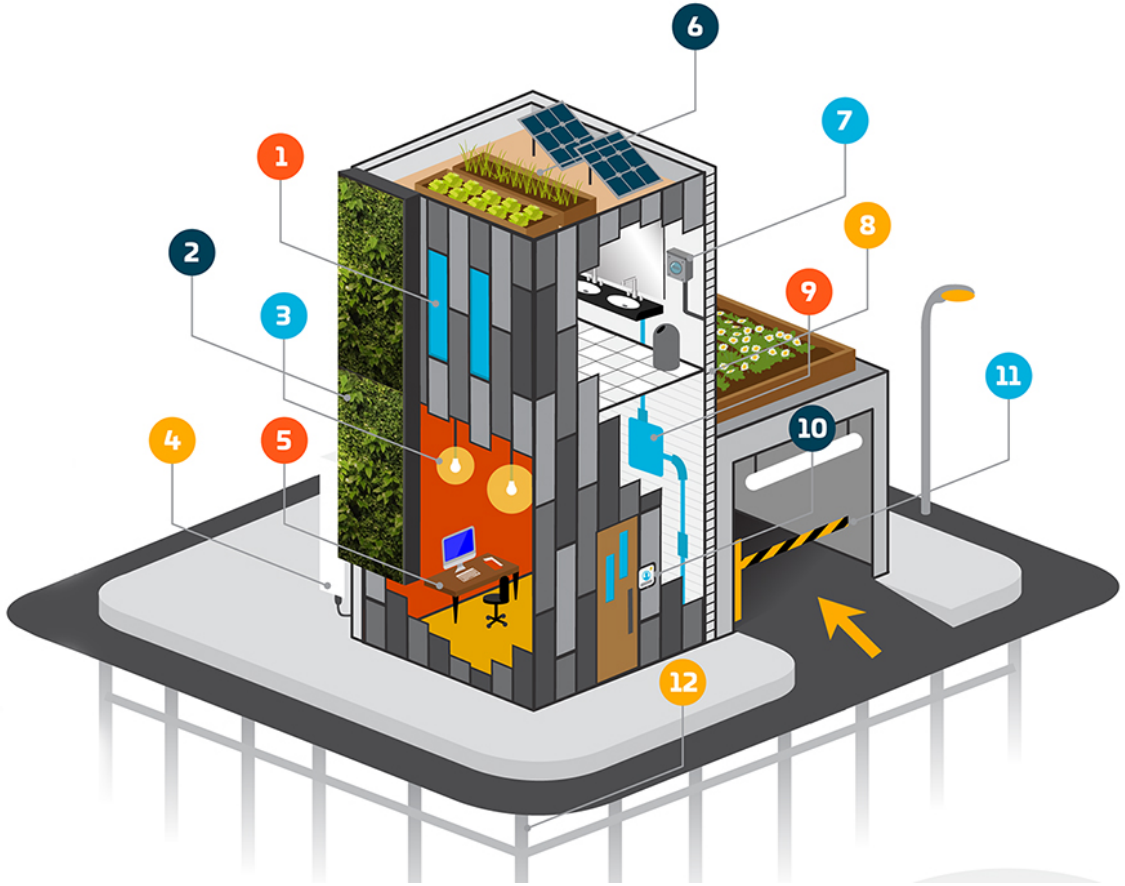
Task environment - Smart building

Performance measure?

Environment?

Actuators?

Sensors?



Environment properties

Fully observable

- Relevant parts of the state of the environment can be sensed
- No need to maintain any internal state to keep track of the world
- Examples: chess, image analysis,

Partially observable

- Parts of the environment cannot be sensed
- Agent must make informed guesses about world
- Examples: poker, taxi driving, medical diagnosis

Environment properties

Single agent

- No other agents - or my agent's performance is not influenced by them
- Examples: medical diagnosis, image analysis

Multi-agent

- My agent's performance depends on behaviour of other agents and vice versa
- Competitive and cooperative interactions
- Examples: poker, chess, taxi driving

Environment properties

Deterministic

- Any action has a single guaranteed effect, and no uncertainty/failure.
- Examples: chess, image analysis

Nondeterministic

- Given the current state and action, there is some uncertainty about the next state/
- Multiple outcome alternatives, quantified in terms of probabilities
- Examples: poker, taxi driving, medical diagnosis

Environment properties

Episodic

- The agent’s experience is divided into atomic episodes.
- Each episode consists of the agent perceiving and then performing a single action
- The choice of action in each episode depends only on the episode itself
- Examples: image analysis

Sequential

- The current decision could affect all future decisions
- Examples: poker, chess, taxi driving, medical diagnosis

Environment properties

Discrete

- Finite number of distinct states, percepts and actions
- Examples: chess, poker

Continuous

- Continuous time/state/actions
- Examples: taxi driving, medical diagnosis, image analysis

Environment properties

Dynamic

- May change while an agent is deliberating
- Examples: taxi driving, medical diagnosis

Static

- The environment does not change
- Examples: poker

Semidynamic

- The world does not change but the agent’s performance score may
- Examples: chess with clock, image analysis

Environment properties

Known

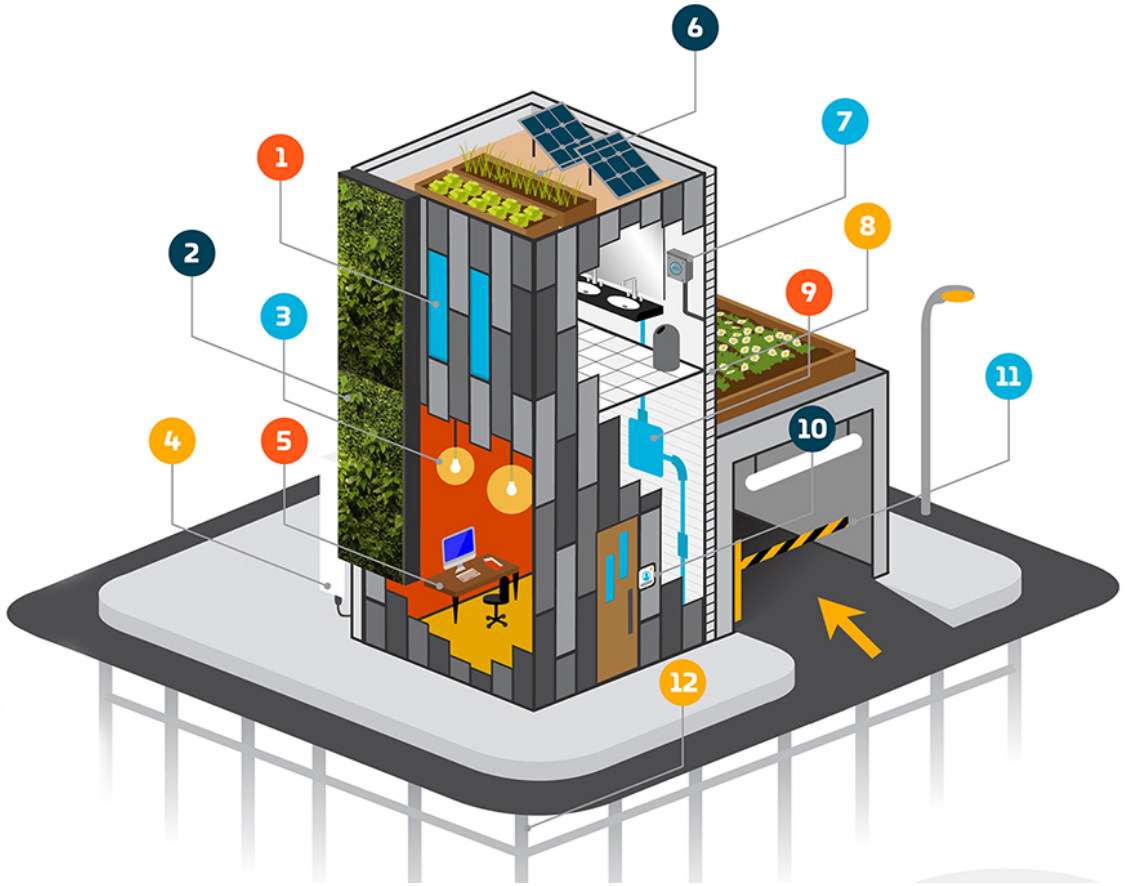
- The agent’s knowledge about how the environment works/evolves
- Note that a known environment (i.e., the agent knows all the rules that apply) may be only partially observable

Unknown

- The agent will have to learn how it works
- An unknown environment can be fully observable.

Properties of the environment - Smart building

1. Fully vs. partially observable?
2. Single agent vs. multiagent?
3. Deterministic vs. nondeterministic?
4. Episodic vs. sequential?
5. Static vs. dynamic?
6. Discrete vs. continuous?
7. Known vs unknown?



Agent = program + architecture

> The job of AI is to design an agent program that implements the agent function - the mapping from of percept sequence to actions

Agent architecture - computing device that runs the agent program with physical sensors and actuators.

Table-driven agent

Percept sequence	Action
$[A, Clean]$	$Right$
$[A, Dirty]$	$Suck$
$[B, Clean]$	$Left$
$[B, Dirty]$	$Suck$
$[A, Clean], [A, Clean]$	$Right$
$[A, Clean], [A, Dirty]$	$Suck$
⋮	⋮
$[A, Clean], [A, Clean], [A, Clean]$	$Right$
$[A, Clean], [A, Clean], [A, Dirty]$	$Suck$
⋮	⋮

Table-driven agent

```
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
```

> The key challenge for AI is to find out how to write programs that, to the extent possible, produce rational behavior from a smallish program rather than from a vast table.

Simple reflex agent

function REFLEX-VACUUM-AGENT([*location*,*status*]) **returns** an action

if *status* = *Dirty* **then return** *Suck*
else if *location* = *A* **then return** *Right*
else if *location* = *B* **then return** *Left*

Simple reflex agent

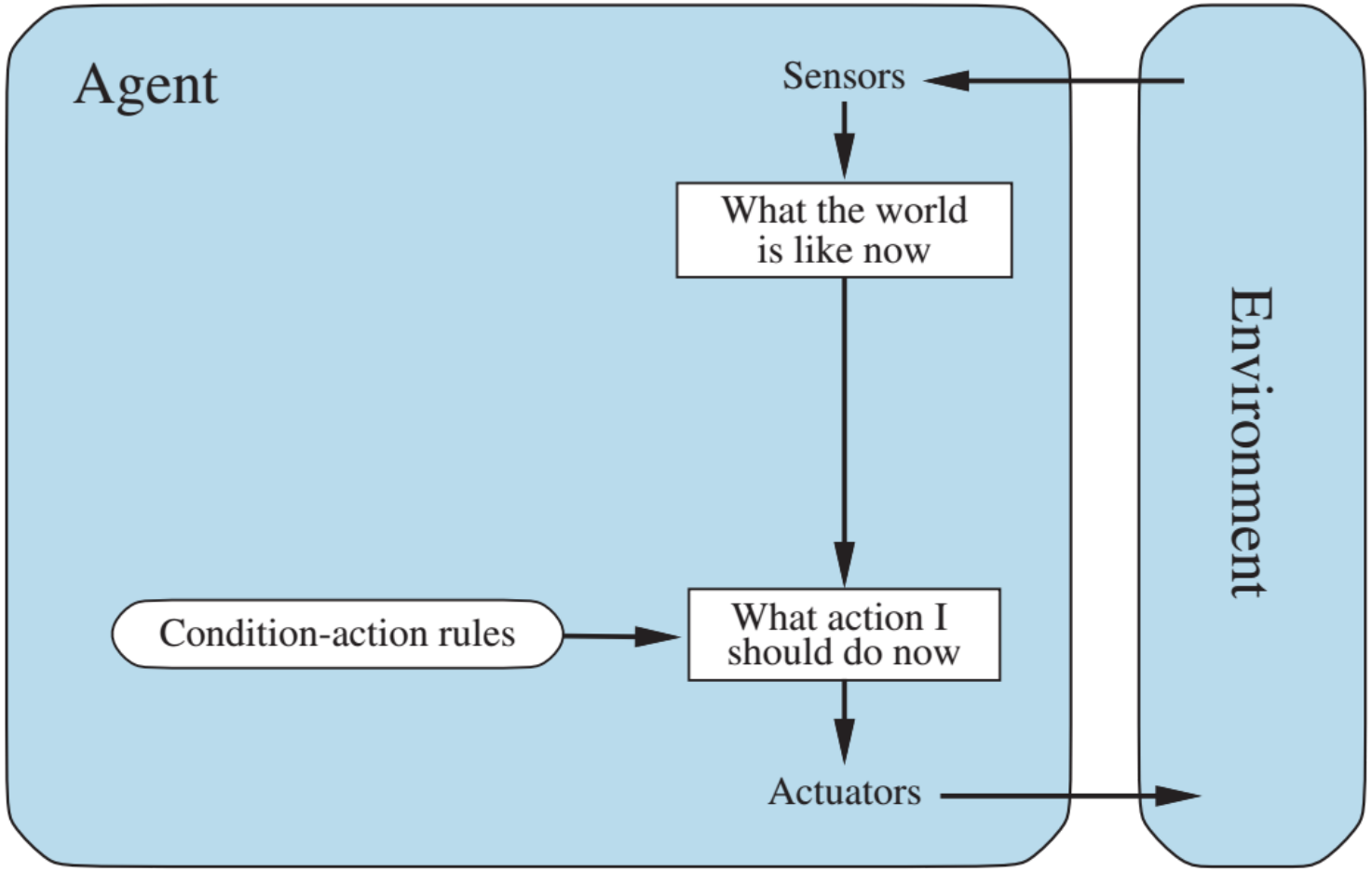
Acts according to a rule matching the current state.

Handles in fully observable environments.

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*

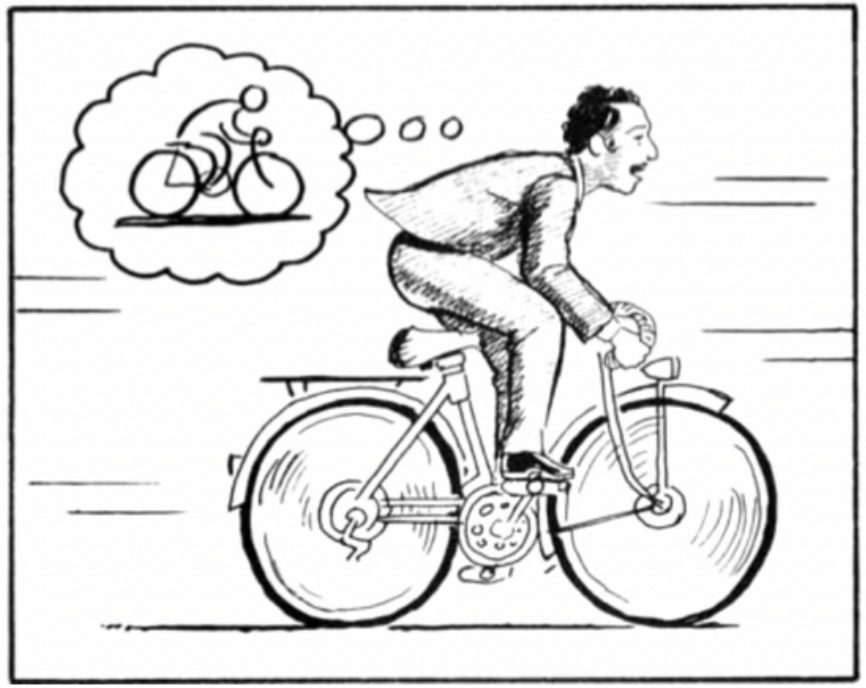
Simple reflex agent



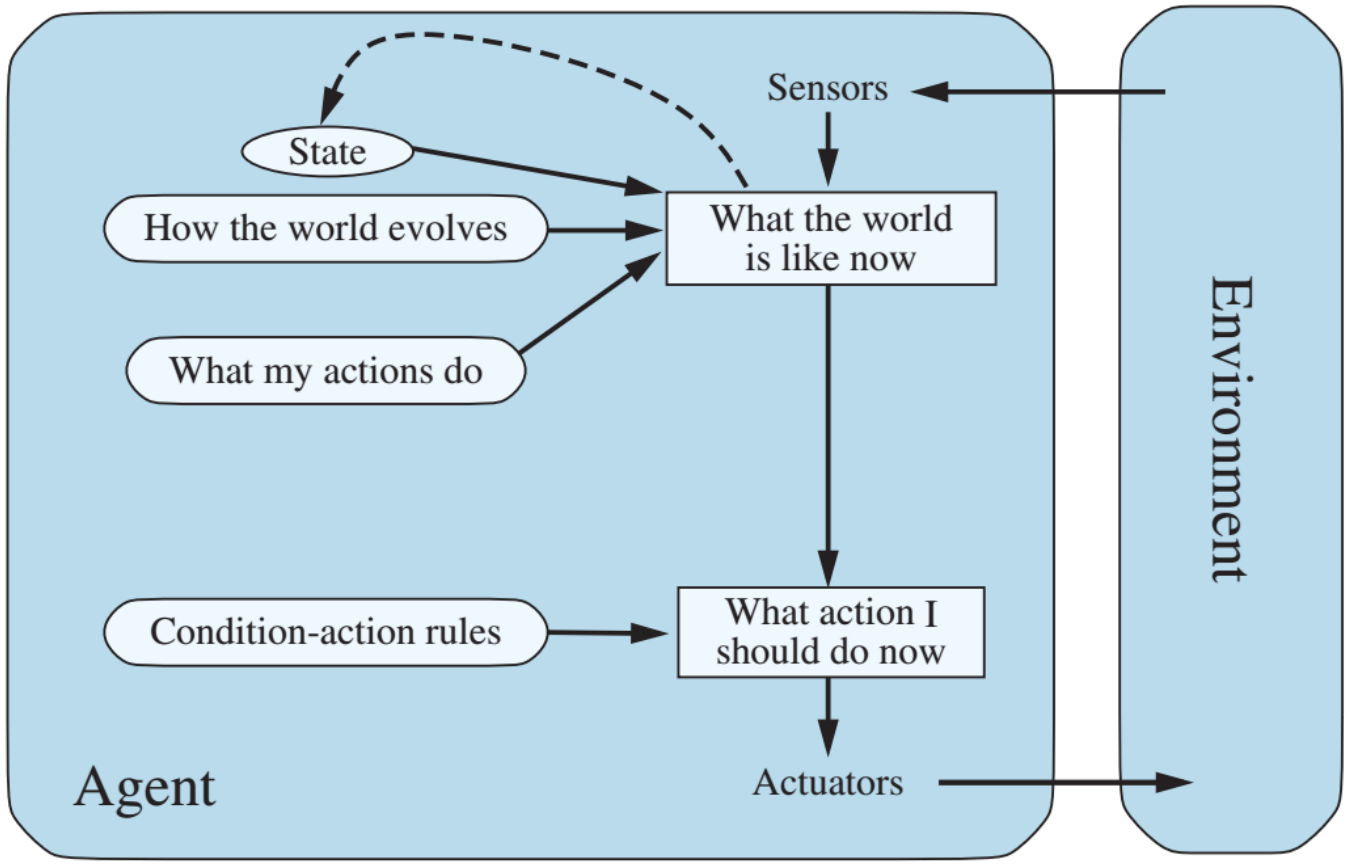
Will it handle partially observable environment?

Model-based reflex agent

- Maintains internal state
- Has a model of how the world works: transition and sensor models
- Handles partially observable environments



Model-based reflex agent



Model-based reflex agent

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

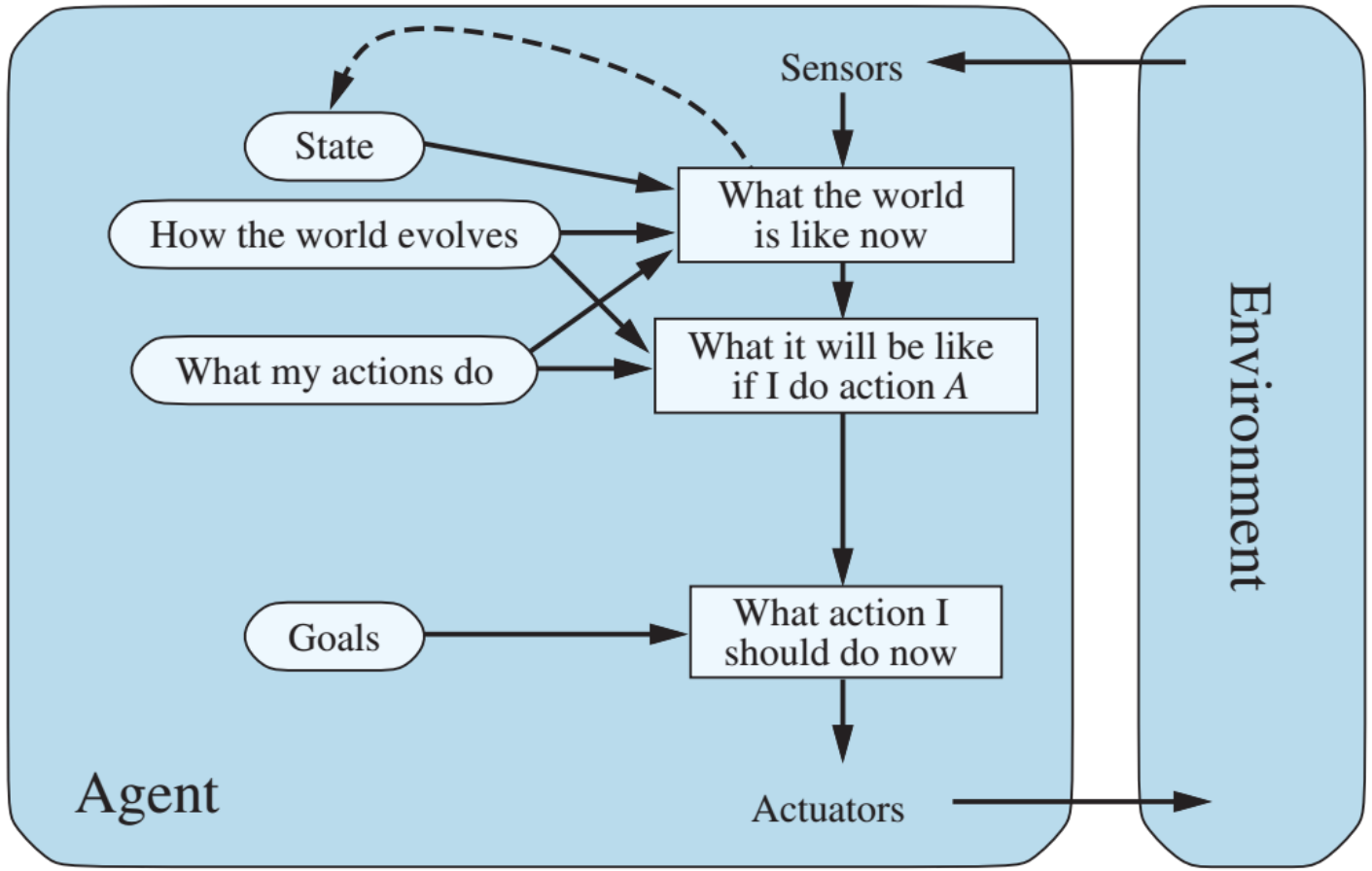
  state ← UPDATE-STATE(*state*, *action*, *percept*, *transition_model*, *sensor_model*)
  rule ← RULE-MATCH(*state*, *rules*)
  action ← *rule*.ACTION
  return *action*

Goal-based agent

- Goal identifies desirable states.
- What will happen if I do *X*? Will *X* make me happy?
- More flexible, general than reflex agent
- Handles sequential environments, e.g. taxi driver, chess

Goal-based agent

Chooses an action that will (eventually) lead to the achievement of its goals

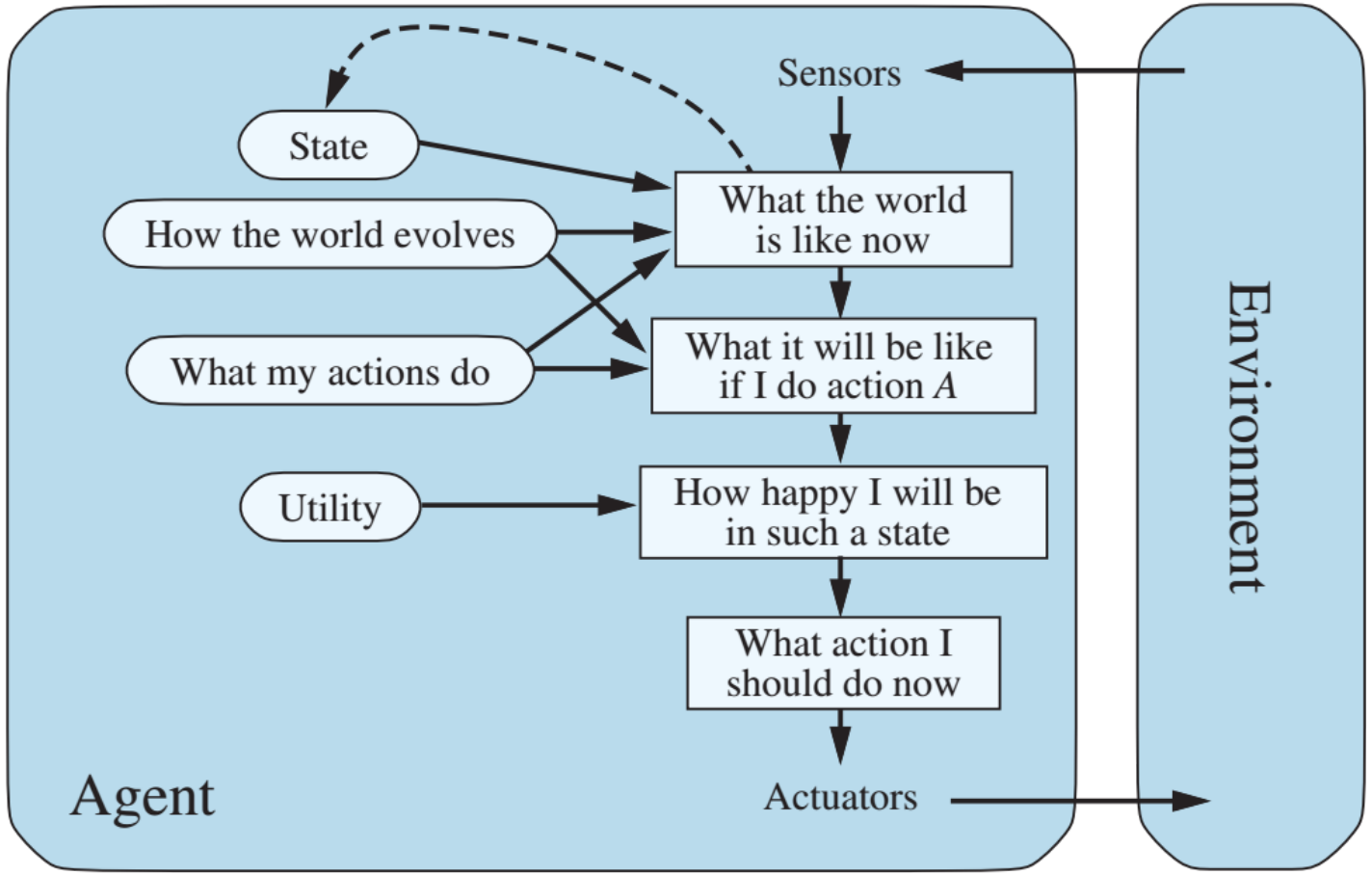


Utility function

- Utility provides hapiness level for (sequence of) states.
- Same goal may be achieved in more than one way, with different utility values.
- **Utility function** - internalization of the performance measure
- **Expected utility** - average utility weighted by state probabilities.
- Handles nondeterministic and partially observable environments, e.g. poker

Utility-based agent

Chooses an action to maximize the expected utility

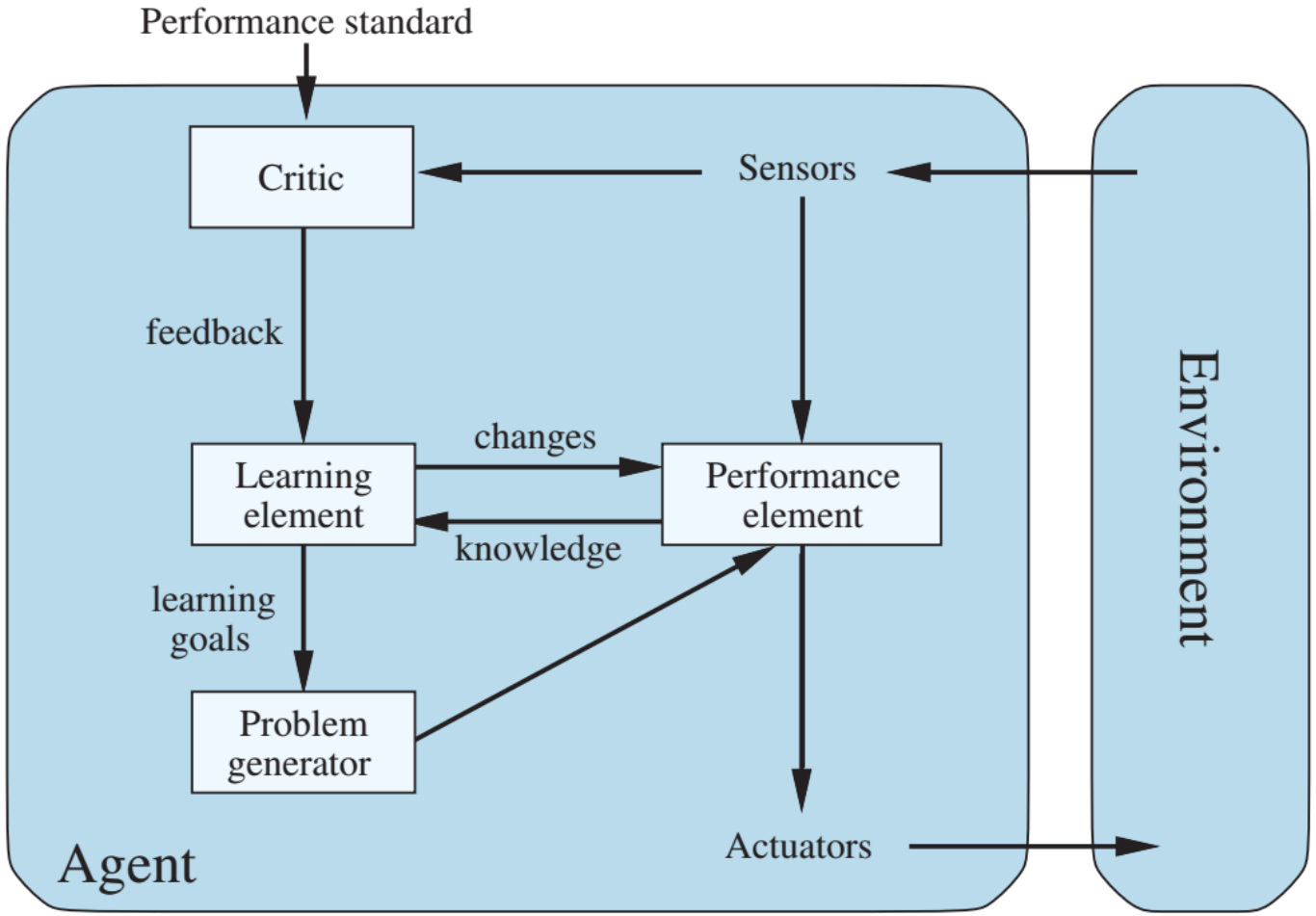


Learning agent

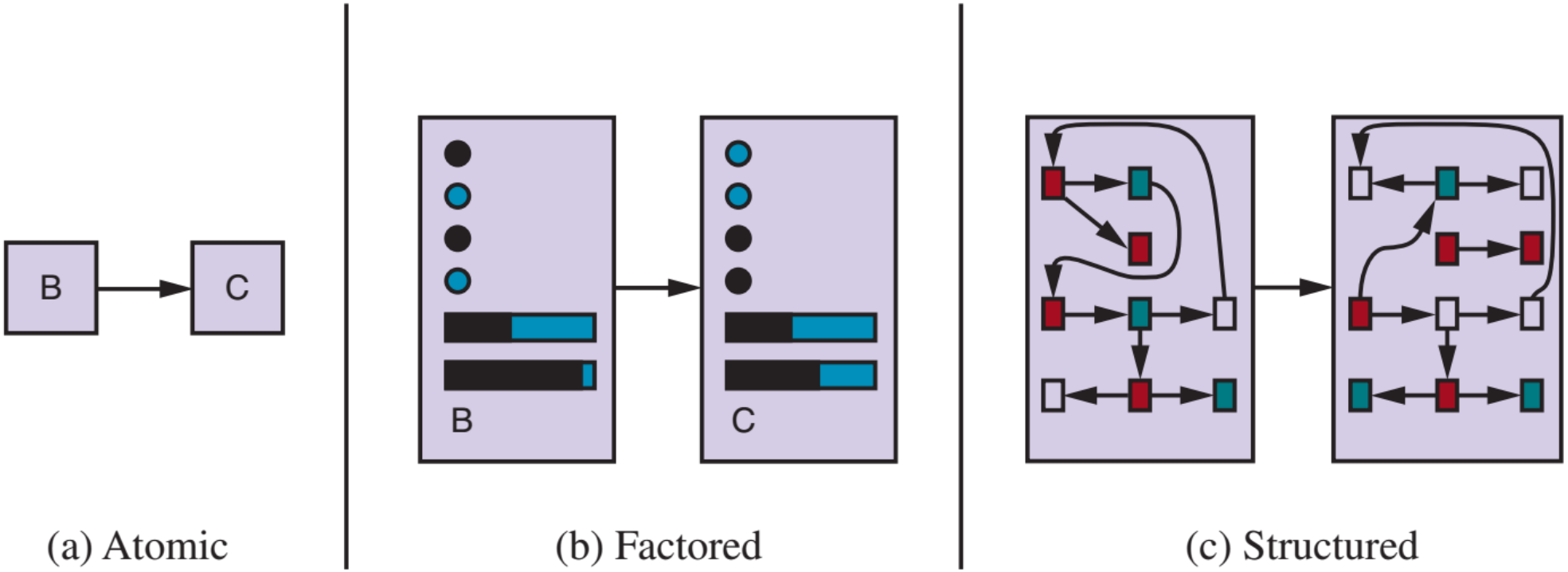
- Becomes more competent than its initial knowledge allows.
- Modifies its components to bring them into closer agreement with the feedback.
- Does some suboptimal actions to explore.
- Handles well unknown and dynamic environemnts.

Learning agent

Chooses the best action given what it knows and/or leads to new and informative experiences.



Representations of states and transitions



Questions?

True/False?

1. An agent that senses only partial information about the state cannot be rational.
2. There exist task environments in which no pure reflex agent can behave rationally.
3. There exists a task environment in which every agent is rational.
4. It is possible for a given agent to be rational in two distinct task environments.
5. Every agent is rational in an unobservable environment.
6. A rational poker-playing agent never loses.