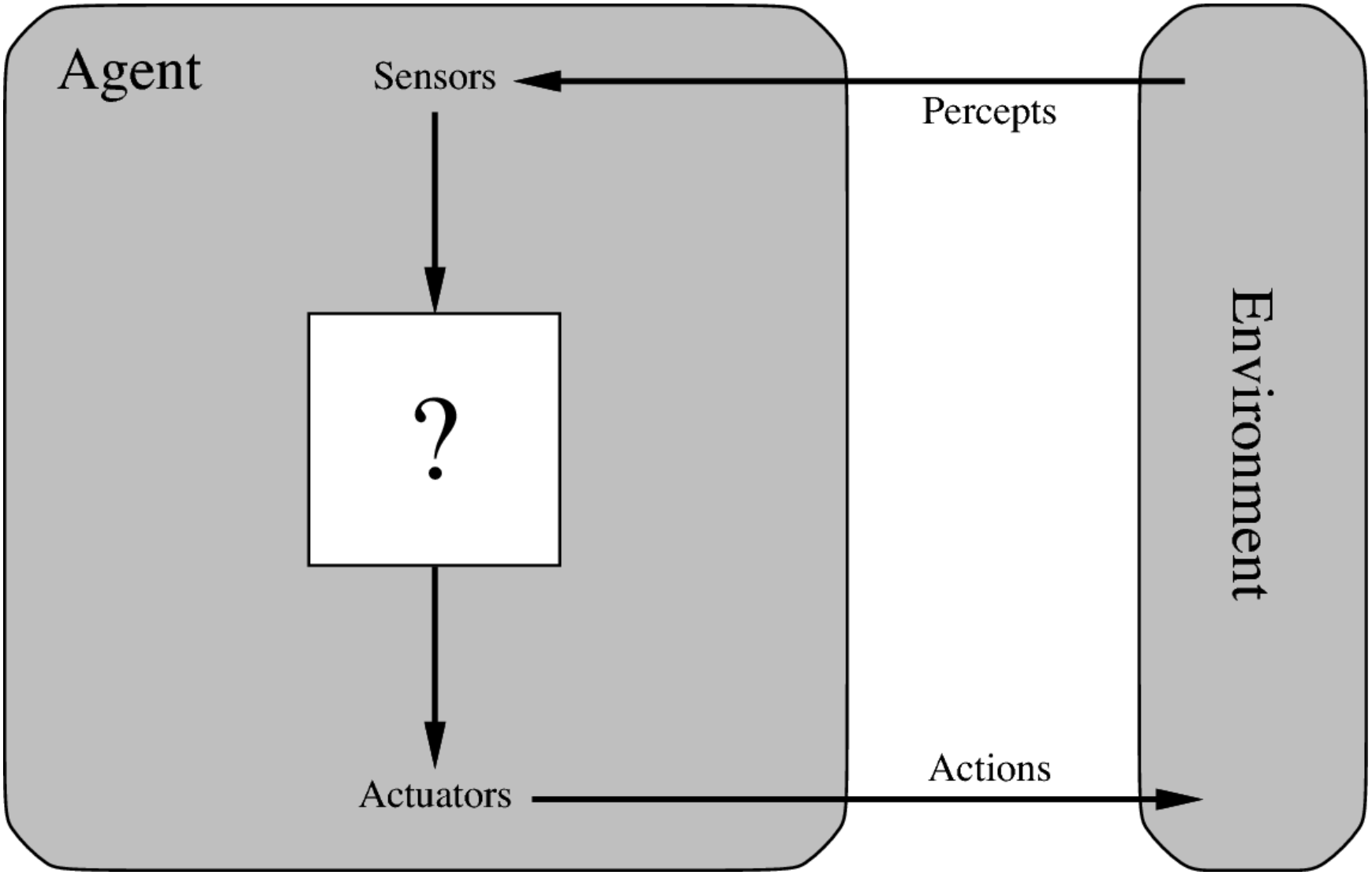
Artificial Intelligence

Information Systems and Machine Learning Lab (ISMLL)
Tomáš Horváth

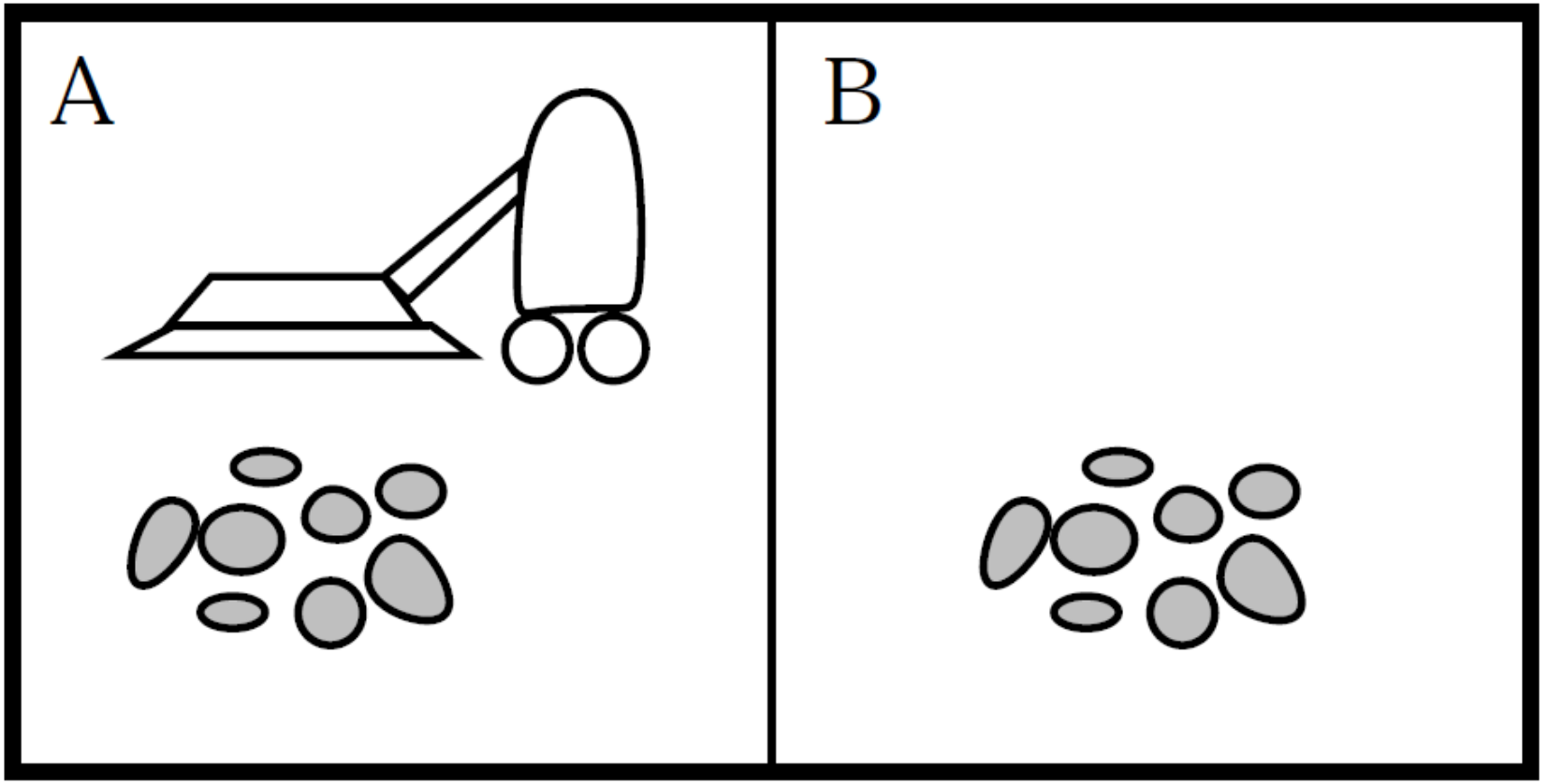
2nd November, 2011

Intelligent Agents

What is an Agent?



An example



An example

- Percepts
  - dirty
  - clean
- Action
  - move
    - left
    - right
  - suck up the dirt
  - do nothing

Tabulation of agent functions

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
…	…

- What is the right way to fill this table?
  - i.e. what makes the agent to be good, efficient, etc.

A rational agent

- the one that does the right thing
  - every entry in the table is filled correctly
  - **What does it mean to do a right thing?**
    - the right action is the one that makes an agent to be most successful...
  - How can we measure success?

Performance measure

- criteria for the success of an agent's behavior
  - objectivity
  - e.g. the amount of dirt cleaned up in one shift
    - **What will the rational agent do?**
  - or having a clean floor
    - What is the exact notion of clean floor?

- it is better to design performance measures
  - according to what we actually want
  - not how we think an agent should behave

Rationality

- For each possible percept sequence a rational agent should select an action that is expected to maximize the performance measure, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has.

- **What things the rationality depends on?**

Rationality

- depends on the
  - performance measure that defines the criterion of success
  - agent's prior knowledge of the environment
  - actions that the agent can perform
  - agent's percept sequence to date

An omniscient agent

- Rationality
  - maximizes the expected performance
- Perfection
  - maximizes the actual performance
  - such an agent would need all the informations about the complete state of the environment
    - not possible in real life

Learning and autonomy

- rationality is not an omniscience!
  - the rational choice depends only on the percept sequence to date
  - the prior knowledge of an agent about the environment should be modified as the agent gains experience – LEARNING
  - the rational agent should learn what it can to compensate for partial or incorrect prior knowledge and not just rely on the prior knowledge of its designer - AUTONOMY

The task environment

- Performance measure
  - safe, fast, legal, comfortable trip; maximum profit...
- Environment
  - roads, other traffic, pedestrians, customers, …
- Actuators
  - steering, accelerator, brake, signal, horn, display, …
- Sensors
  - cameras, sonar, GPS, accelerometer, engine sensors, ...

Properties of the task environment

- fully vs. partially observable
  - relevance of the information observed depends on the performance measure
- deterministic vs. stochastic
  - strategic, if the environment is deterministic except for the actions of other agents
- episodic vs. sequential
  - the next episode does not depend on the previous episodes vs. the current decision could affect all future decisions
- static vs. dynamic
  - semi-dynamic, if the environment doesn't change but the agent's performance score does
- discrete vs. continuous
- single agent vs. multiple agents
  - competitive vs. cooperative
  - partially cooperative (traffic)
  - partially competitive (one parking place for one car)

Task environment properties

Task environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
crossword puzzle	fully	deterministic	sequential	static	discrete	single
chess with a clock	fully	strategic	sequential	semi	discrete	multi
poker	partially	stochastic	sequential	static	discrete	multi
backgammon	fully	stochastic	sequential	static	discrete	multi
taxi driving	partially	stochastic	sequential	dynamic	continuous	multi
medical diagnosis	partially	stochastic	sequential	dynamic	continuous	single
image analysis	fully	deterministic	episodic	semi	continuous	single
part-picking robot	partially	stochastic	episodic	dynamic	continuous	single
refinery controller	partially	stochastic	sequential	dynamic	continuous	single
interactive English tutor	partially	stochastic	sequential	dynamic	discrete	multi

The structure of agents

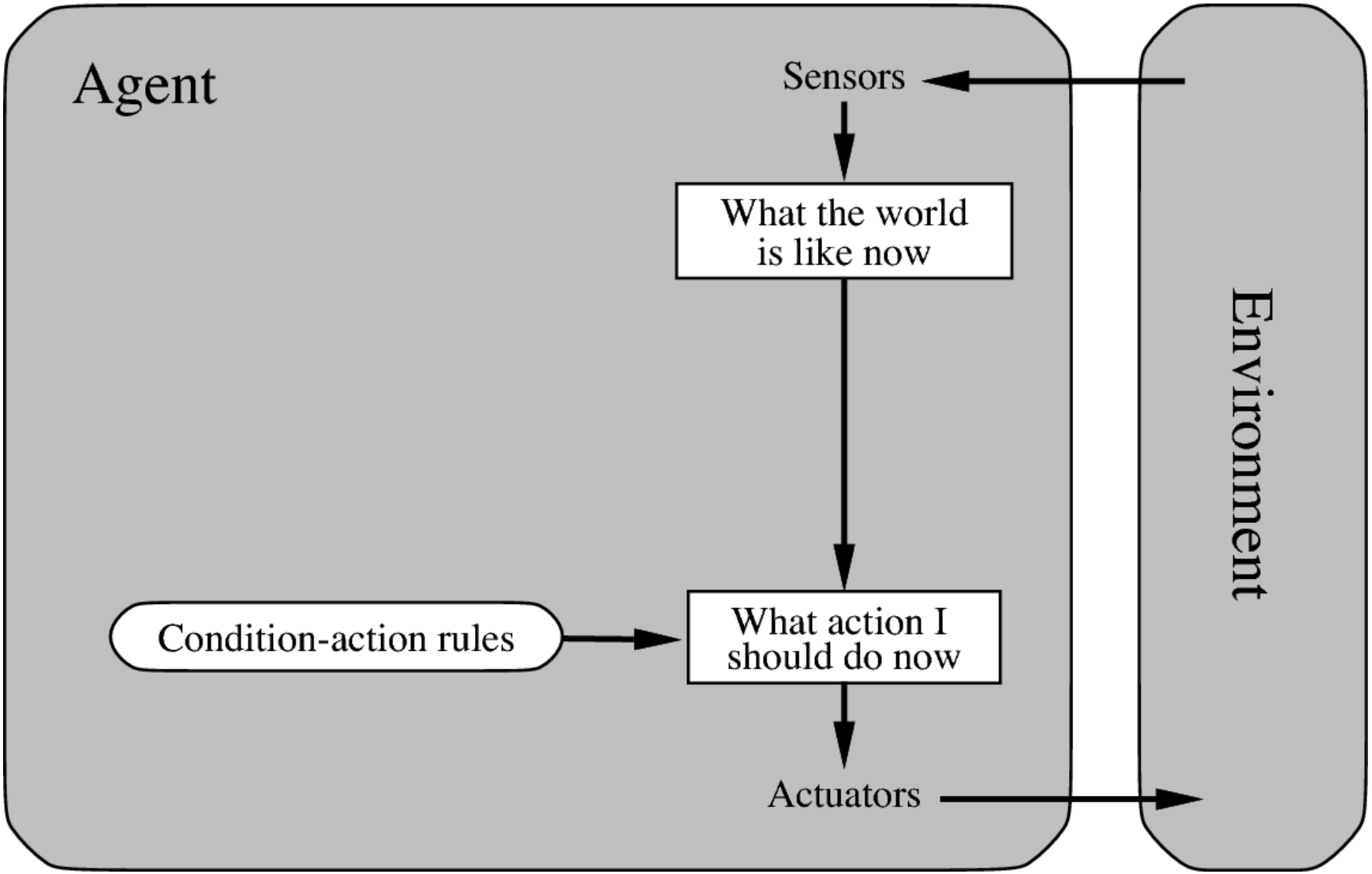
- agent = architecture + program

```
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
```

- how much entries will the lookup table contain?

```
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



Simple reflex agent

- will work only if the environment is fully observable
  - a little bit of unobservability can cause troubles
    - **can You tell an example?**
    - **are there some other problems?**

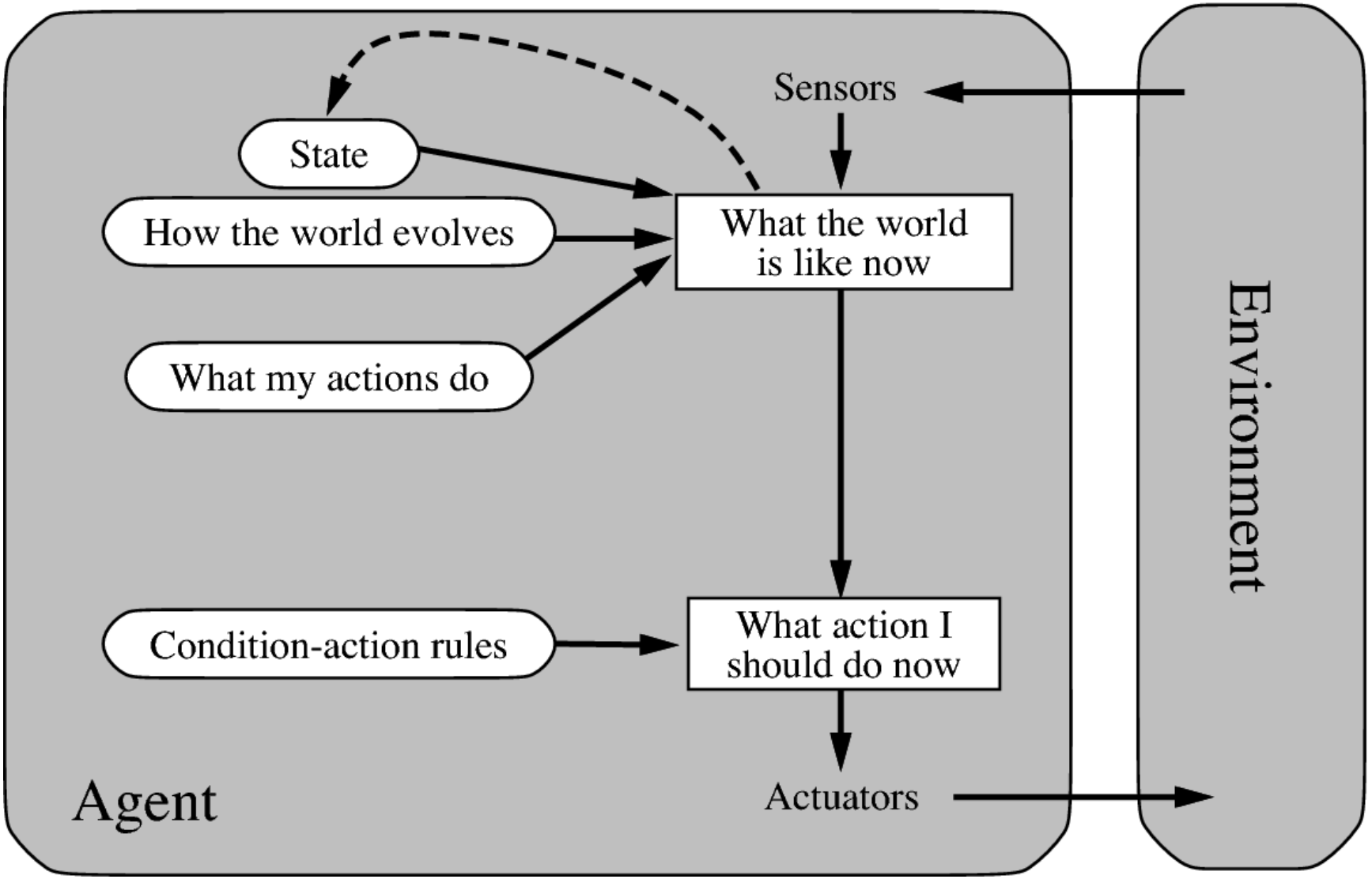
function SIMPLE-REFLEX-AGENT(*percept*) **returns** an action
  persistent: *rules*, a set of condition–action rules

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

Model-based reflex agent

- to keep track of the part of the world it can't see now is an effective way to handle observability
  - requires two kinds of knowledge
    - how the world evolves independently from the agent
      - an overtaking car accelerates, thus will be closer, ...
      - THE MODEL OF THE WORLD
    - how the agent's actions affect the world
      - what happens if the driver turns the wheel, ...

Model-based reflex agent



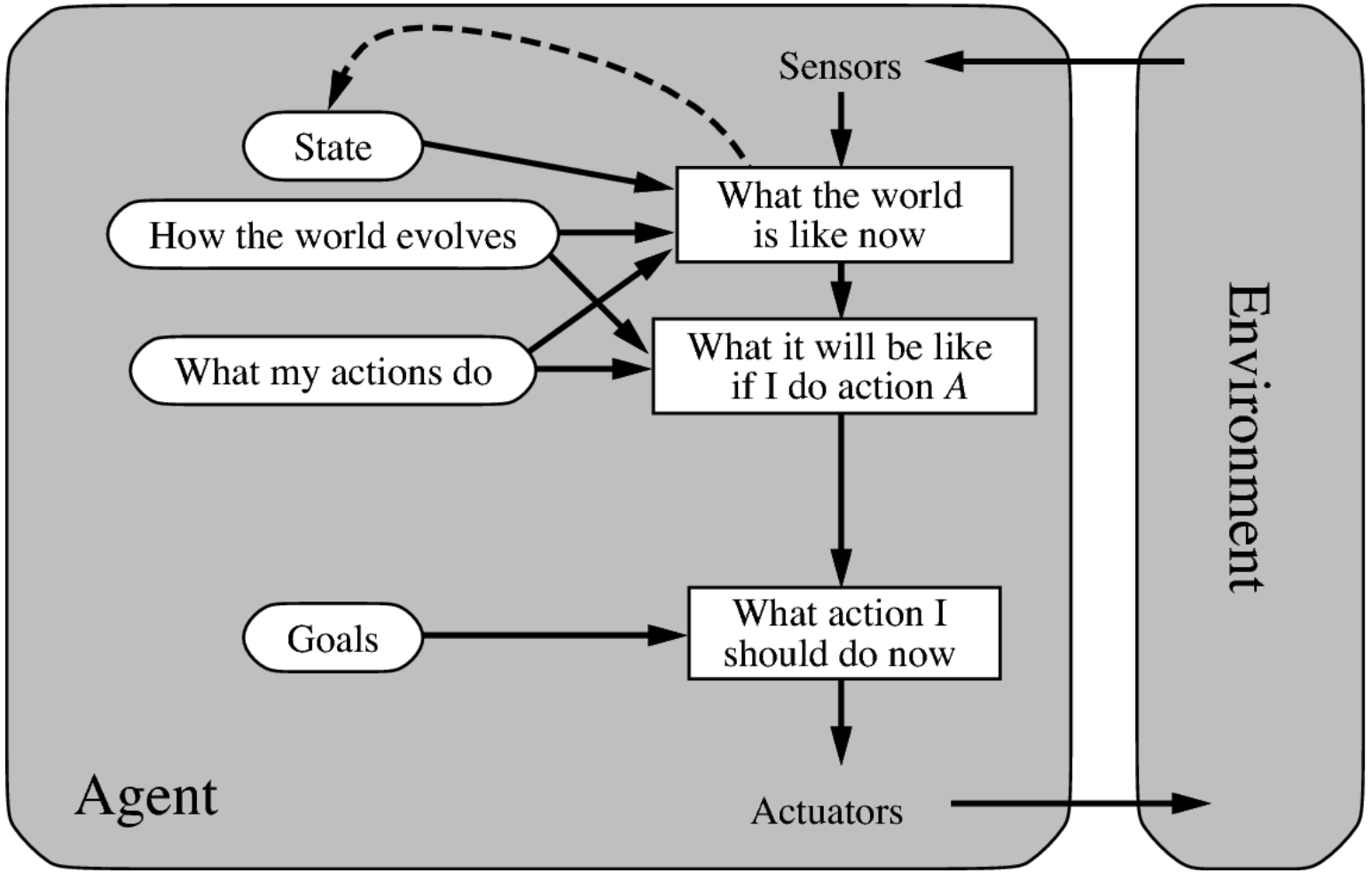
Model-based reflex agent

function MODEL-BASED-REFLEX-AGENT(*percept*) **returns** an action
  persistent: *state*, the agent's current conception of the world state
    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*

- **what is the lack of this type of an agent?**

Goal-based agent

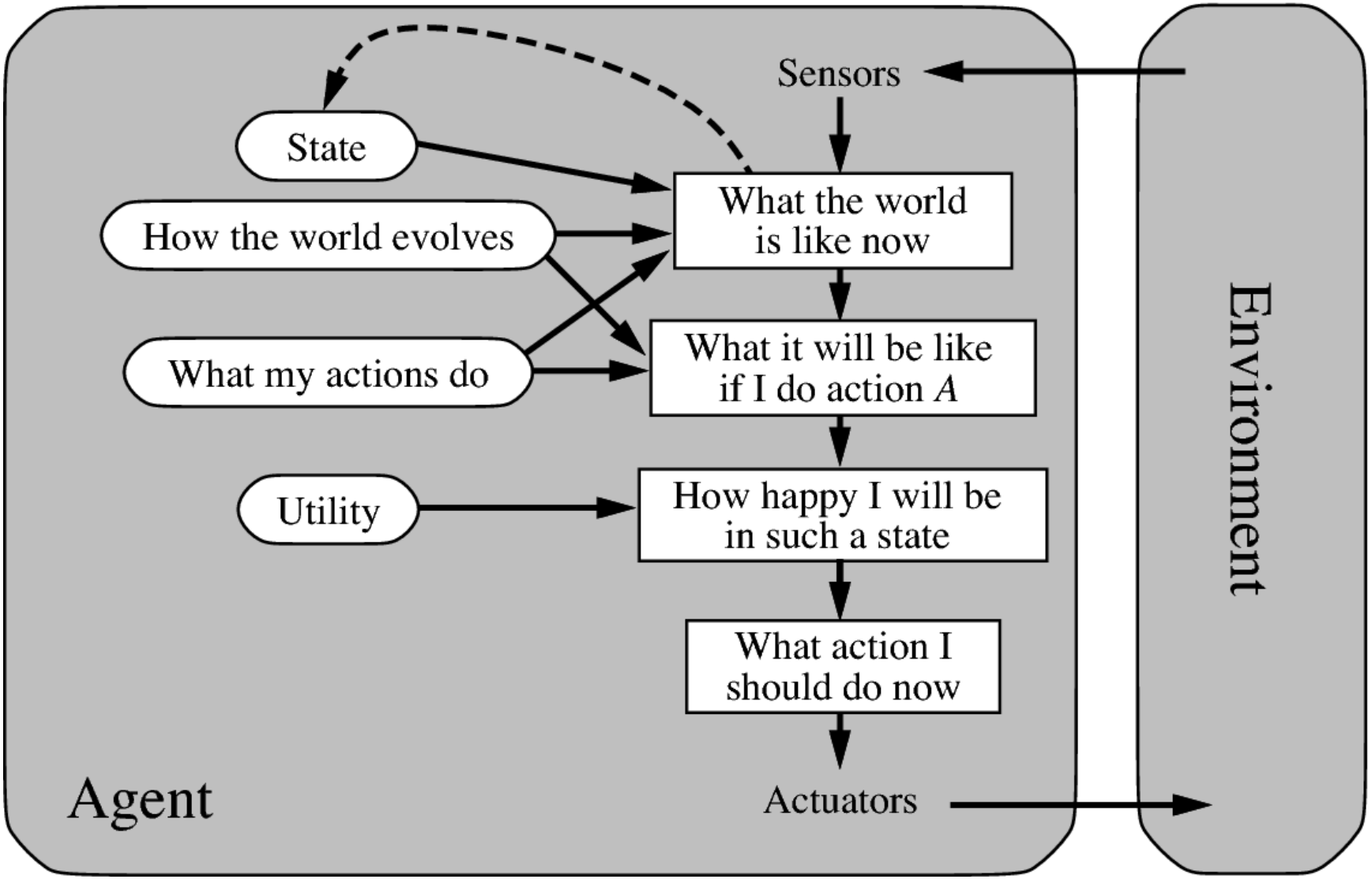


- **what is the lack of this type of an agent?**

Utility-based agent

- goal-based agent
  - binary distinction between “happy” and “unhappy”
  - many ways to reach the goal
    - which is the best/optimal?
- utility function
  - maps a state onto a real number
  - trade-off between the conflicting goals
  - likelihood of success is weighted up against the importance of several goals

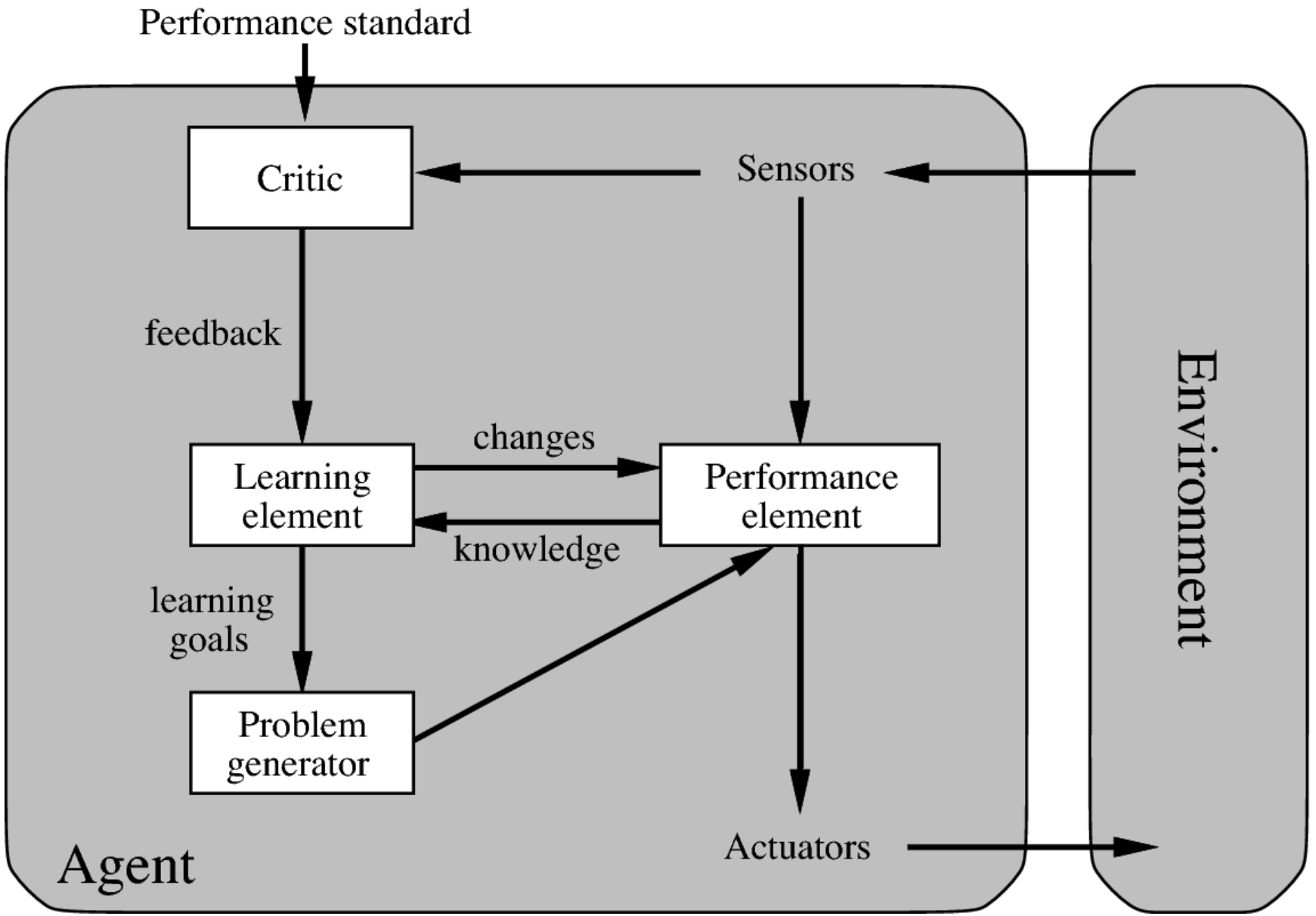
Utility-based agent



Learning agent

- learning allows to agent to operate in an unknown environment
- Main parts
  - learning element – making improvements
  - performance element – selecting external actions (previously was considered to be an entire agent)
  - critic – gives feedback to the learning element
    - important because the percepts themselves provide no indications about agent's success
  - problem generator – suggesting actions leading to new and informative experiences (exploring)

Learning agent



Thanks for Your attention!

Questions?