Intelligent Agents

Chapter 2

Outline

- Agents and Environments
- Rationality
- PEAS (Performance Measure, Environment, Actuators, Sensors)
- Environment Type
- Agent Types

Agent and Environment

An AI system is composed of an agent and its environment. The agents act in their environment. The environment may contain other agents. (Competitive, Cooperative)

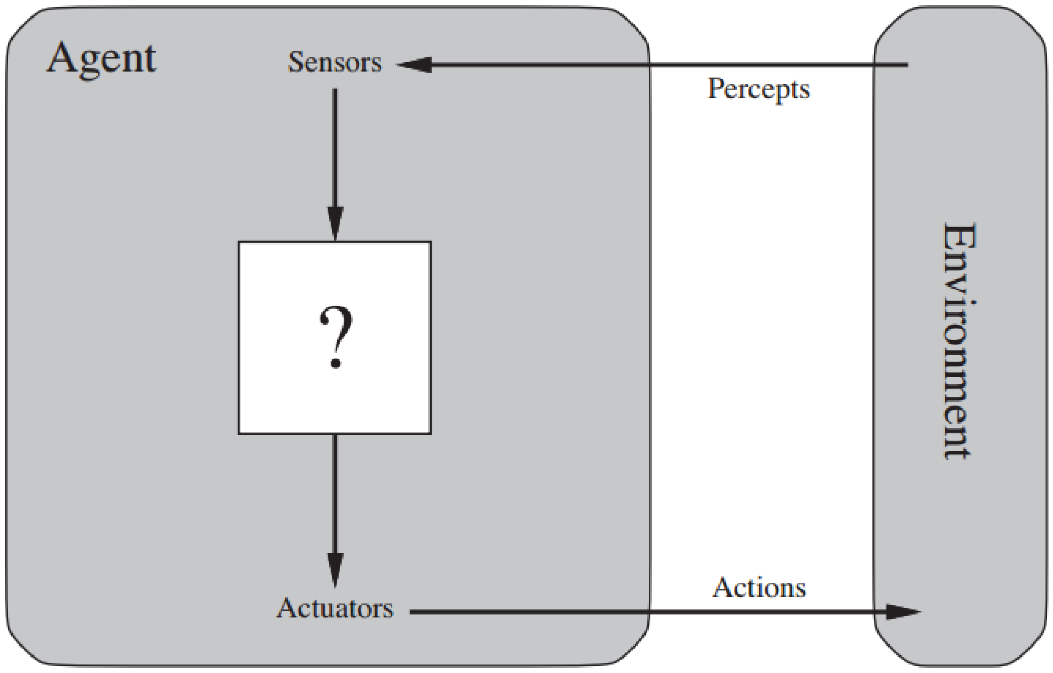
An **agent** is anything that can perceive its environment through sensors and acts upon that environment through actuators.

A **human agent** has sensory organs such as eyes, ears, nose, tongue and skin parallel to the sensors, and other organs such as hands, legs, mouth, for actuators.

A **robotic agent** replaces cameras and infrared range finders for the sensors, and various motors for actuators.

A **software agent** has encoded bit strings as its programs and actions.

Agent and Environment



Agents include humans, robots, softbots, thermostats, etc. The agent function maps from percept sequence to actions:

$f : P^* \rightarrow A$

The agent program runs on the physical architecture to produce *f*

Agents Terminology

Performance Measure of Agent − It is the criteria, which determines how successful an agent is.

Behavior of Agent − It is the action that agent performs after any given sequence of percepts.

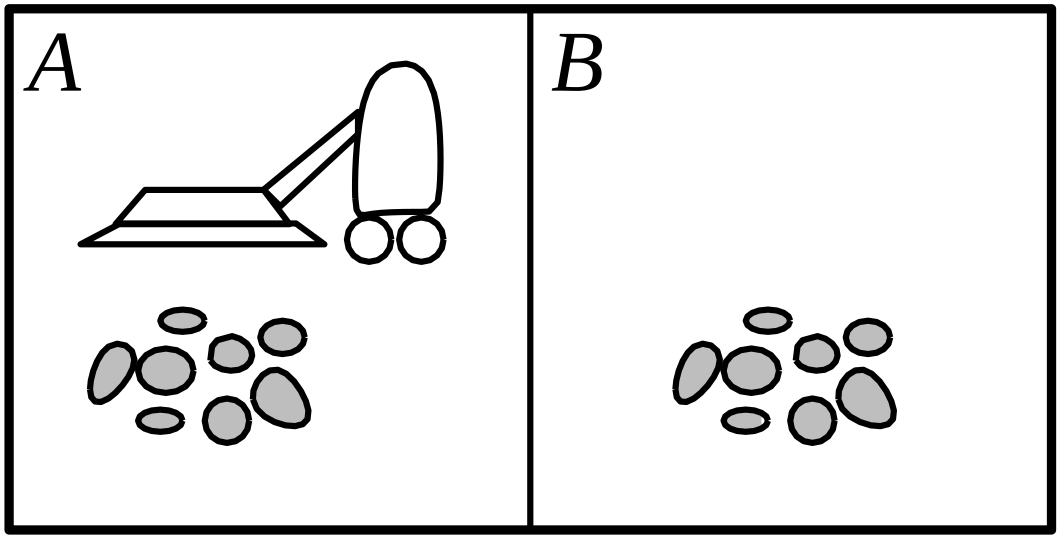
Percept − It is agent’s perceptual inputs at a given instance.

Percept Sequence − It is the history of all that an agent has perceived till to date.

Agent Function – It is a map from the precept sequence to an action.

Agent Program – Agent Function Implementation

Vacuum-cleaner world



Percepts: location and contents, e.g., [*A, Dirty*]

Actions: *Left*, *Right*, *Suck*, *NoOp*

A vacuum-cleaner 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*
.	.

```
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
```

What is the right function?
Can it be implemented in a small agent program?

Rationality

Rationality is nothing but status of being reasonable, sensible, and having good sense of judgment.
Rationality is concerned with expected actions and results depending upon what the agent has perceived.
Action for the purpose of modifying future percept is called information gathering and it is an important part of rationality.
A Rational Agent not only gathers information but also learns as much as possible.
A Rational Agent must be autonomous that is it relies on info obtained by gathering and learning rather than info provided by designer

Rationality is maximizing expected performance
Perfection is maximizing actual performance

Rationality vacuum Cleaner

Fixed performance measure evaluates the environment sequence

- one point per square cleaned up in time T ?
- one point per clean square per time step, minus one per move?
- penalize for $>$ k dirty squares?
- Penalize for > m units of electricity consumed per time step
- Penalize for amount of noise generated

Rationality = Do the right thing

Rational Action: The action that maximizes the expected value of the performance measure given the percept sequence to date

Rationality

- Rational = Best(to the best of its knowledge)
- Rational = Optimal (to the best of its ability incl. constraints)
- Rational ≠ Omniscience (not know all the knowledge all-knowing with infinite knowledge) – action outcomes may not be as expected)
- Rational ≠ Clairvoyant (Percepts may not be complete: *Pertaining to the ability of clear-sightedness; it is the paranormal ability to see persons and events that are distant in time or space. It is divided into roughly three classes:*
  - *precognition: the ability to perceive or predict future events,*
  - *retrocognition: the ability to see past events,*
  - *remote viewing: the perception of contemporary events happening outside of the range of normal perception.*)
- Rational ≠ Successful (Success is not guaranteed)
- Rational ≠Perfection (max expected not actual outcome)

What is Ideal Rational Agent?

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.

An ideal rational agent is capable of doing expected actions to maximize its performance measure, on the basis of −

- Its percept sequence
- Its built-in knowledge base

Rationality of an agent depends on the following four factors −

- The **performance measures**, determine the degree of success.
- Agent's **Percept Sequence** till now.
- The agent's **prior knowledge about the environment**.
- The **actions** that the agent can carry out.

A rational agent always performs right action: the action that causes the agent to be most successful in the given percept sequence.

The problem that agent solves is characterized by Performance Measure, Environment, Actuators, and Sensors (PEAS).

PEAS for various Agents

To design a rational agent, we must specify the task environment

Consider, e.g., the task of designing an automated taxi:

Performance measure?? safety, destination, profits, legality, comfort, . . .

Environment?? US streets/freeways, traffic, pedestrians, weather, . . .

Actuators?? steering, accelerator, brake, horn, speaker/display, . . .

Sensors?? video, accelerometers, gauges, engine sensors, keyboard, GPS, . . .

PEAS for various Agents

Internet Shopping Agents

Performance measure?? price, quality, appropriateness, efficiency

Environment?? current and future WWW sites, vendors, shippers

Actuators?? display to user, follow URL, fill in form

Sensors?? HTML pages (text, graphics, scripts)

Medical Diagnosis Agent

Performance measure?? Healthy patient, reduce costs

Environment?? Patients, Hospital Staff

Actuators?? Display of questions, tests, diagnosis , treatments, referrals

Sensors?? Keyboard entry of symptoms, findings, patients responses

PEAS for various Agents

Satellite Image Analysis Agents

Performance measure?? Correct Image categorization

Environment?? Downlink from satellite

Actuators?? Display of scene categorization

Sensors?? Color Pixel arrays

Part Picking Robot Agent

Performance measure?? Percentage of parts in correct bin

Environment?? Conveyor belt with parts, Bins

Actuators?? Jointed arm and hand

Sensors?? Camera, Joint angle sensors

PEAS for various Agents

Refinery Controlling Agents

Performance measure??	Purity, yield, safety
Environment??	Refinery Operators
Actuators??	Valves, Pumps, heaters, displays
Sensors??	Temp, Pressure, Chemical sensors

Interactive English Tutoring Agent

Performance measure??	Students scores on the test
Environment??	Sets of students and testing agency
Actuators??	Display of exercises, suggestions and corrections
Sensors??	Keyboard entry

The Structure of Intelligent Agents

Agent’s structure can be viewed as −

Agent = Architecture + Agent Program

Architecture = the machinery that an agent executes on.

Agent Program = an implementation of an agent function.

Environment Properties

Fully vs. partially observable: Environment sensors provide access to complete state of the relevant environment at each point in time

Eg: Chess with clock: Fully observable,
Automated Taxi: Partially observable

Deterministic vs. stochastic : If next state completely determined by current and action, then deterministic, otherwise stochastic. If deterministic except for actions of other agents, then strategic

Eg: Crossword Puzzle: Deterministic
Automated Taxi: Stochastic
Chess with clock: Strategic

Episodic vs. sequential : Episodic, action choice depends only on current state. Sequential if current action affects future actions

Eg: Part Picking Robot: Episodic
Image Analysis, Chess and Automated Taxi: Sequential

Environment Properties

Static vs. dynamic : Dynamic, environment can change while agent is still deliberating

Eg: Crossword puzzle: Static

Chess with clock: Semi-dynamic

Automated Taxi: Dynamic

Discrete vs. continuous : Applied to state, time, percepts, or actions

The way the information is represented

Eg: Chess and Crossword: Discrete

Automated Taxi and Refinery controller: Continuous

Single agent vs. multiagent: How distinguish agent from environment? if other's behavior maximizes its performance based on agent, then it is multiagent

Eg: Crossword Puzzle, Medical diagnosis, Refinery Controller: Single Agent

Chess with clock and Automated Taxi: Multiagent

Environment Examples

	Observable	Agent	Deterministic	Episodic	Static	Discrete
Crossword Puzzle	Fully	Single	Deterministic	sequential	Static	Discrete
Chess with Clock	Fully	Multi	Deterministic	sequential	Semi	Discrete
Poker	Partially	Multi	Stochastic	sequential	Static	Discrete
Backgammon	Fully	Multi	Stochastic	sequential	Static	Discrete
Taxi driving	Partially	Multi	Stochastic	sequential	Dynamic	Continuous
Medical Diagnosis	Partially	Single	Stochastic	sequential	Dynamic	Continuous
Image Analysis	Fully	Single	Deterministic	Episodic	Semi	Continuous
Part Picking Robot	Partially	Single	Stochastic	Episodic	Dynamic	Continuous
Refinery Controller	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Interactive Eng. Tutor	Partially	Multi	Stochastic	Sequential	Dynamic	Discrete

The environment type largely determines the agent design

The real world is (of course) partially observable, stochastic, sequential, dynamic, continuous, multi-agent

Agent Functions and Programs

An agent is completely specified by the agent function mapping percept sequences to actions
One agent function (or a small equivalence class) is rational

Aim: find a way to implement the rational agent function concisely

Table-lookup agent

```
Function Table-driven-agent (percept)
    Returns an action
     append percept to the end of percepts
     action ← Lookup (percepts, table)
    Return action
```

Drawbacks:

Huge table

Take a long time to build the table

No autonomy

Even with learning, need a long time to learn the table entries

Agent Types

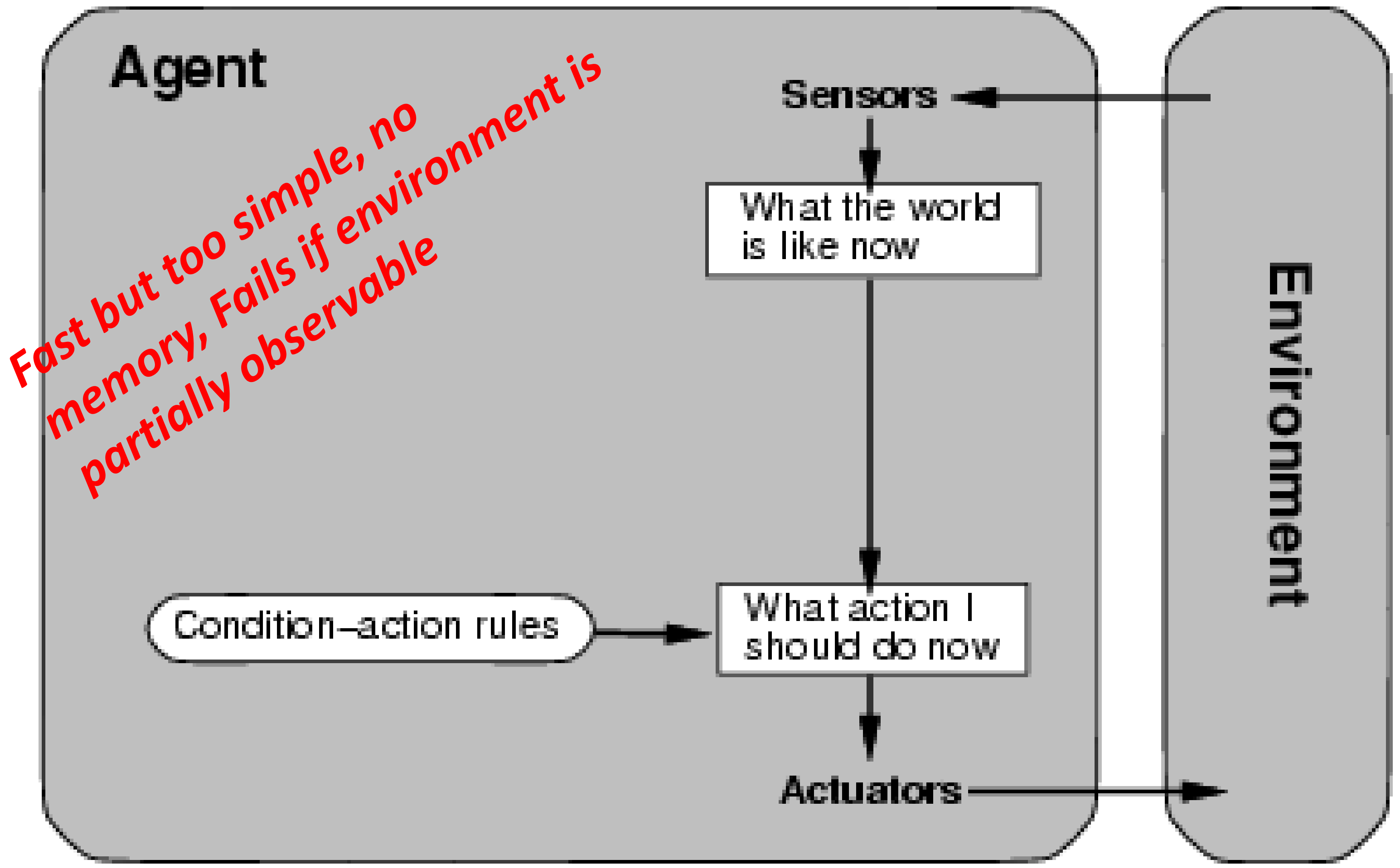
Four basic types in order of increasing generality:

- Simple reflex agents
- Model-Based reflex agents
- Goal-based agents
- Utility-based agents

All these can be turned into learning agents

```
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 Agents


Agent
Fast but too simple, no memory, Fails if environment is partially observable
Sensors
What the world is like now
Condition–action rules
What action I should do now
Actuators
Environment

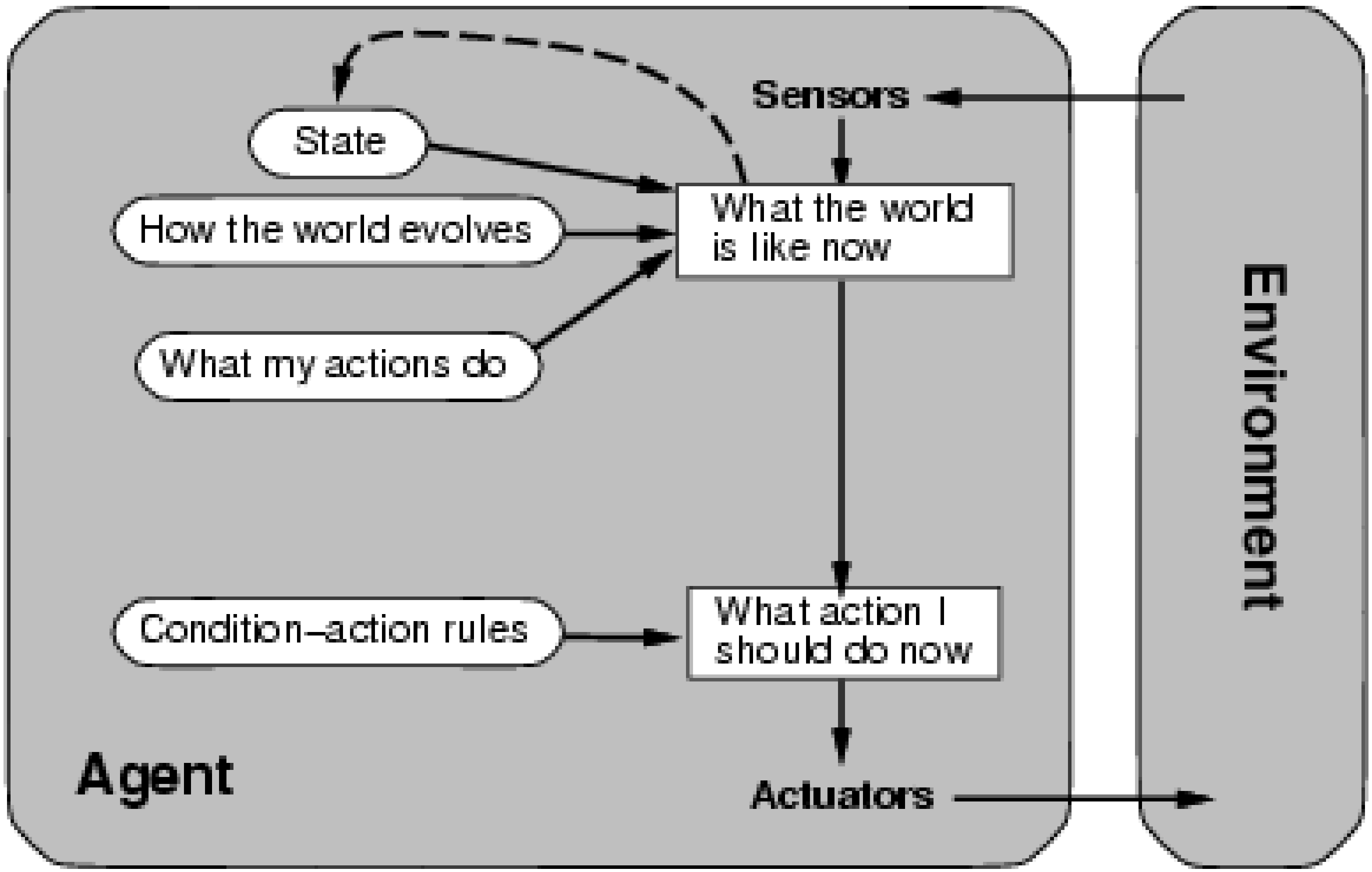
Action selection is based on the current percept (assumes fully observable environment) Condition-action rules (e.g., if car-in-front-is-breaking then initiate-breaking) represent both innate and learned reflexes. Use of simple ” if then” rules Can be short sighted

Simple Reflex Agents

```
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
```

A simple reflex agent. It acts according to a rule whose condition matches the current state, as defined by the percept.

Model-Based Reflex Agents



Model the state of the world by: modeling how the world changes, how it's actions change the world, This can work even with partial information. It is unclear what to do without a clear goal

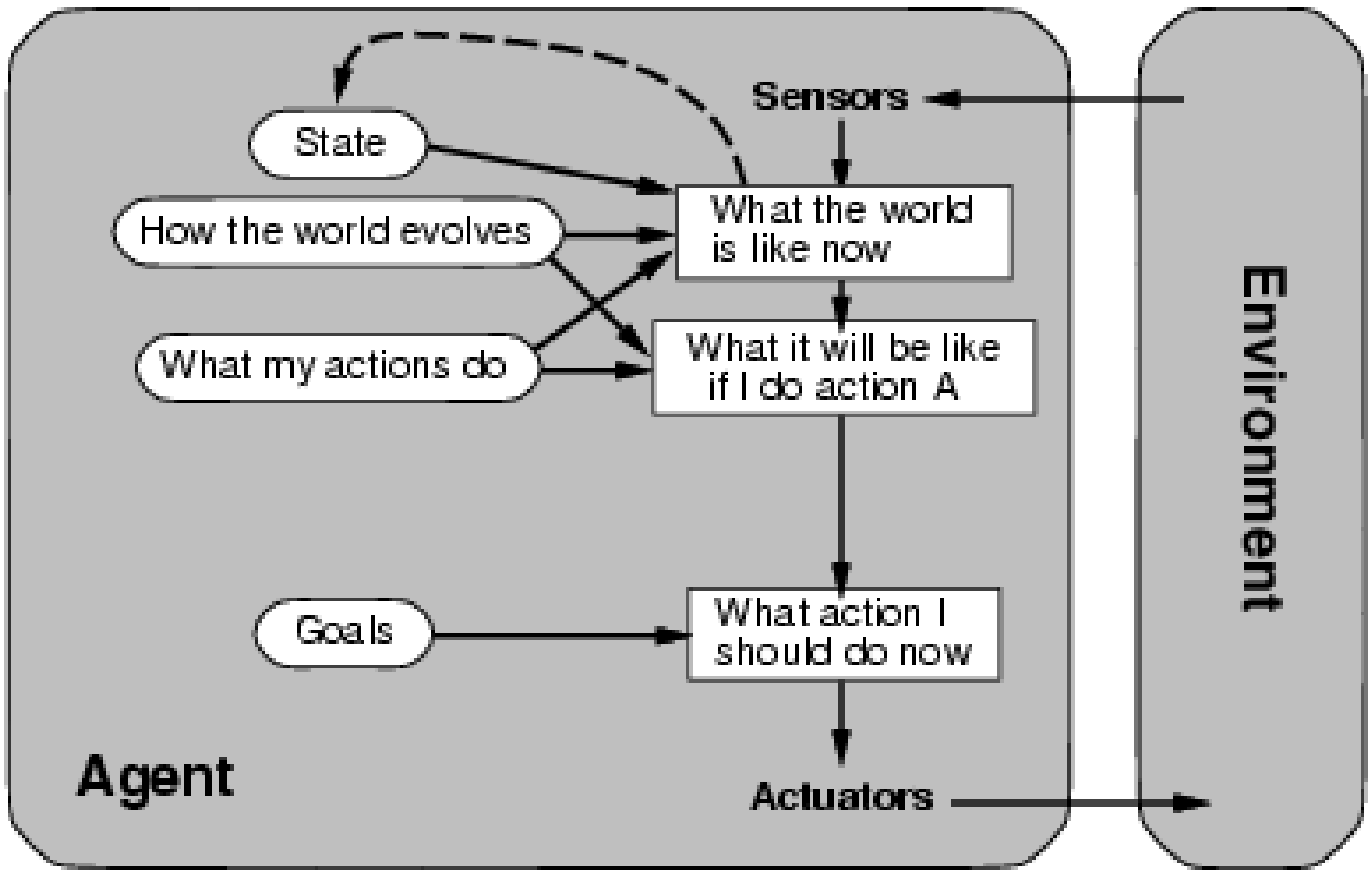
Model-Based Reflex Agents

```
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
```

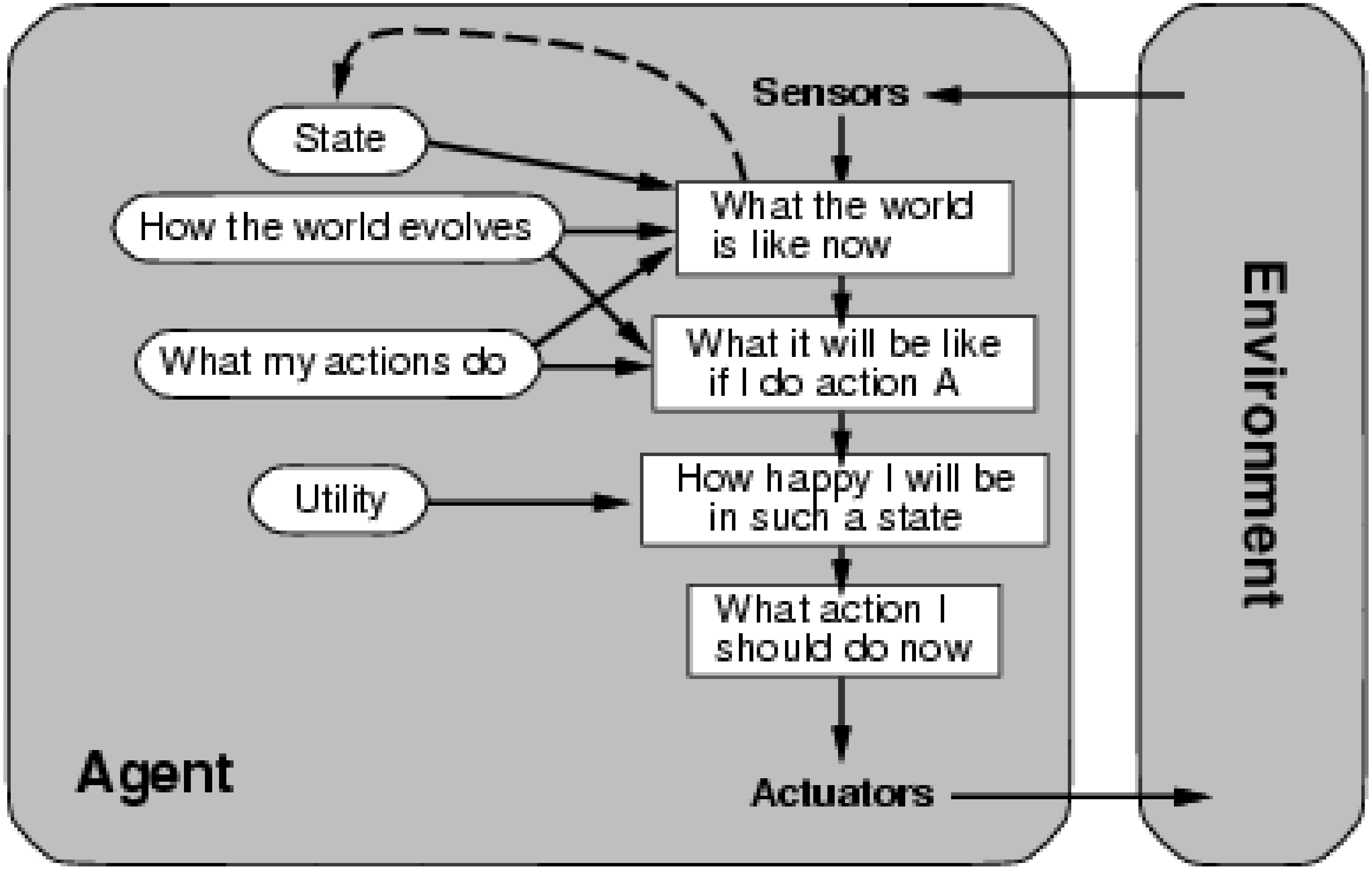
A model-based reflex agent. It keeps track of the current state of the world, using an internal model. It then chooses an action in the same way as the reflex agent.

Goal-Based Agents


Sensors
State
What the world is like now
How the world evolves
What my actions do
What it will be like if I do action A
Goals
What action I should do now
Agent
Actuators
Environment

Goals provide reason to prefer one action over the others. We need to predict the future: we need to plan & search

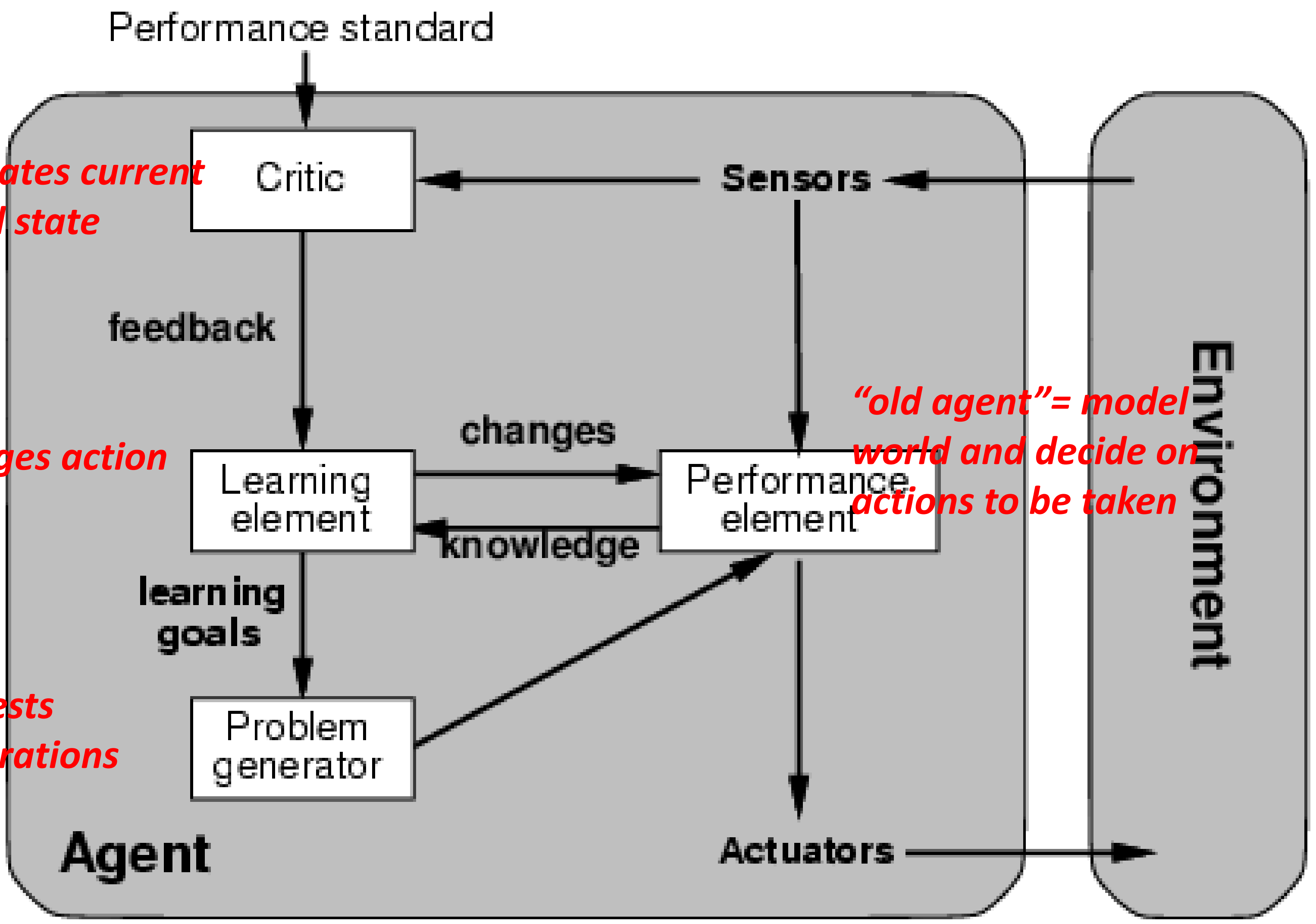
Utility-Based Agents



Sensors
State
How the world evolves
What my actions do
Utility
What the world is like now
What it will be like if I do action A
How happy I will be in such a state
What action I should do now
Agent
Actuators
Environment

Some solutions to goal states are better than others. Which one is best is given by a utility function. Which combination of goals is preferred?

Learning agents



Evaluates current world state

Changes action rules

suggests explorations

"old agent"= model world and decide on actions to be taken

How does an agent improve over time? By monitoring it's performance and suggesting better modeling, new action rules, etc.

Summary

Agents interact with environments through actuators and sensors

The agent function describes what the agent does in all circumstances The

performance measure evaluates the environment sequence

A perfectly rational agent maximizes expected performance

Agent programs implement (some) agent functions

PEAS descriptions define task environments Environments

are categorized along several dimensions:

observable? deterministic? episodic? static? discrete? single-agent?

Several basic agent architectures exist:

reflex, reflex with state, goal-based, utility-based