LU-3: Structure of agents and Environments

Objectives

- To explain the principles involved in simple
  - Reflex agent
  - Model based agent
  - Goal based agent
  - Utility based agent

Outcomes

- Design a simple
  - Reflex agent
  - Model based agent
  - Goal based agent
  - Utility based agent

PEAS

- **PEAS**: Performance measure, Environment, Actuators, Sensors
- Must first specify the setting for intelligent agent design
- Consider, e.g., the task of designing an automated taxi driver. The following factors to be considered
  - **Performance measure**
  - **Environment**
  - **Actuators**
  - **Sensors**

PAGE – another acronym to coin PEAS
P – Percept (sensors)
A – Actions (action)
G – Goals (Performance measure)
E – Environment (Environment)

Structure of Intelligent Agents

Agent = Architecture + Program

Architecture – Device (a Plain computer or special purpose hardware

Program : A function that implements the agents mapping from percepts to actions

Software Agents (Software robots or Softbots)

- Flight simulator
- Online news scanner
- Online surveillances
- Cartoon dog

State Diagram

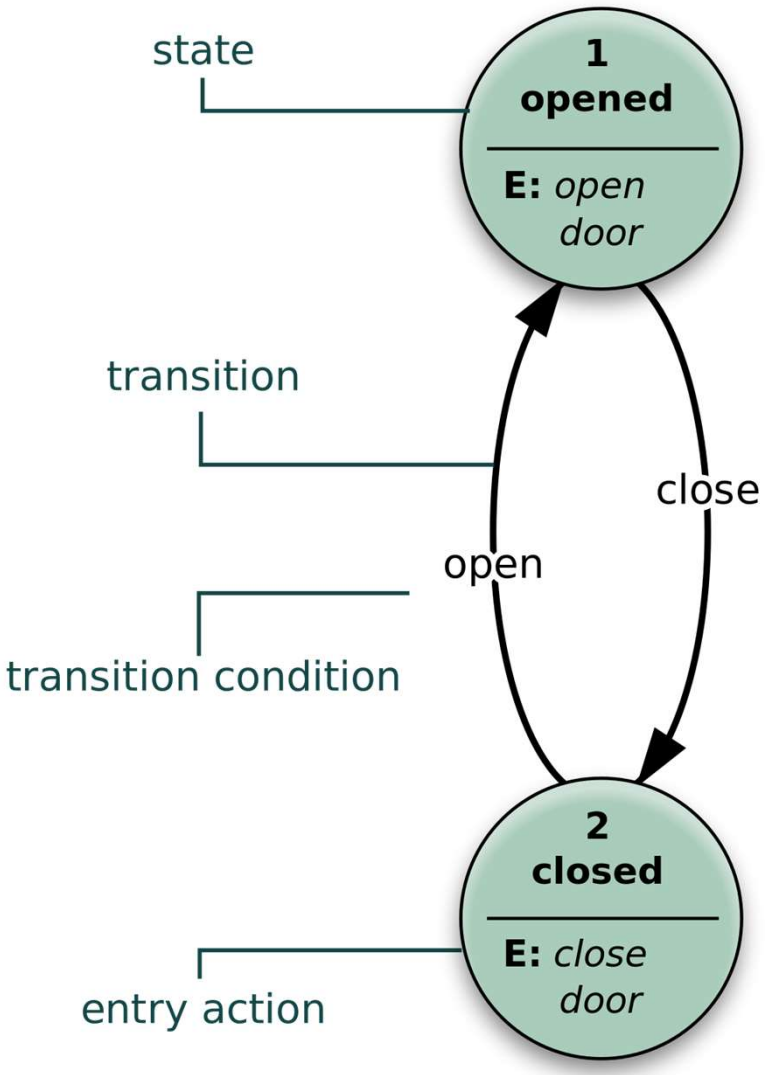
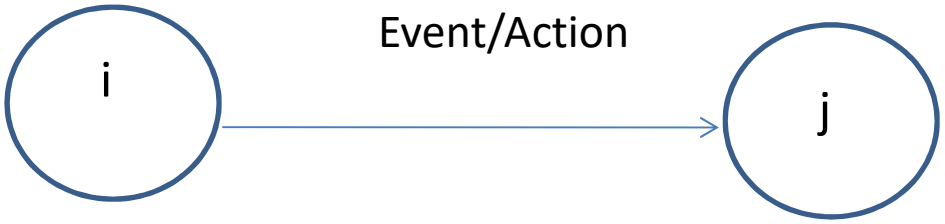


Diagram Acknowledge: Wiki

Agent Program

Function SKELETION-AGENT returns actions

static: memory, the agents memory of the world
memory ← UPDATE-MEMORY(memory, percept)
action ← CHOOSE-BEST-ACTION(memory)
memory ← UPDATE-MEMORY(memory, action)
return action



Starting State	Input/Event	Next State	Action
i	Event	J	Action

Agent using look up Table

Function TABLE-DRIVEN-AGENT (percept) returns action)

static: percept - a sequence initially empty

table – indexed by percept sequence, initially fully specified

append percept to the end of the percepts

action ← LOOKUP (percepts, table)

return action

Limitations of the lookup based Table

- Table size may be exhaustively large (simple chess program requires 35^{100} entries
- Long time for the designer to build
- No autonomy possible. If the environment gets changed, the agent would be lost
- Even with learning, need a long time to learn the table entries

PEAS – Example 1 (Automated Taxi Driver)

- Setting for intelligent agent design: Example -The task of designing an automated taxi driver:
  - Performance measure: Safe, fast, legal, comfortable trip, maximize profits
  - Environment: Roads, other traffic, pedestrians, customers
  - Actuators: Steering wheel, accelerator, brake, signal, horn
  - Sensors: Cameras, sonar, speedometer, GPS, odometer, engine sensors, keyboard

PEAS – Example 2 (Medical diagnosis system/Agent)

- Performance measure: Healthy patient, minimize costs, lawsuits
- Environment: Patient, hospital, staff
- Actuators: Screen display (questions, tests, diagnoses, treatments, referrals)
- Sensors: Keyboard (entry of symptoms, findings, patient's answers)

PEAS/PAGE – Example 3 (Part-Picking Robot)

- Performance measure: Percentage of parts in correct bins
- Environment: Conveyor belt with parts, bins
- Actuators: Jointed arm and hand
- Sensors: Camera, joint angle sensors

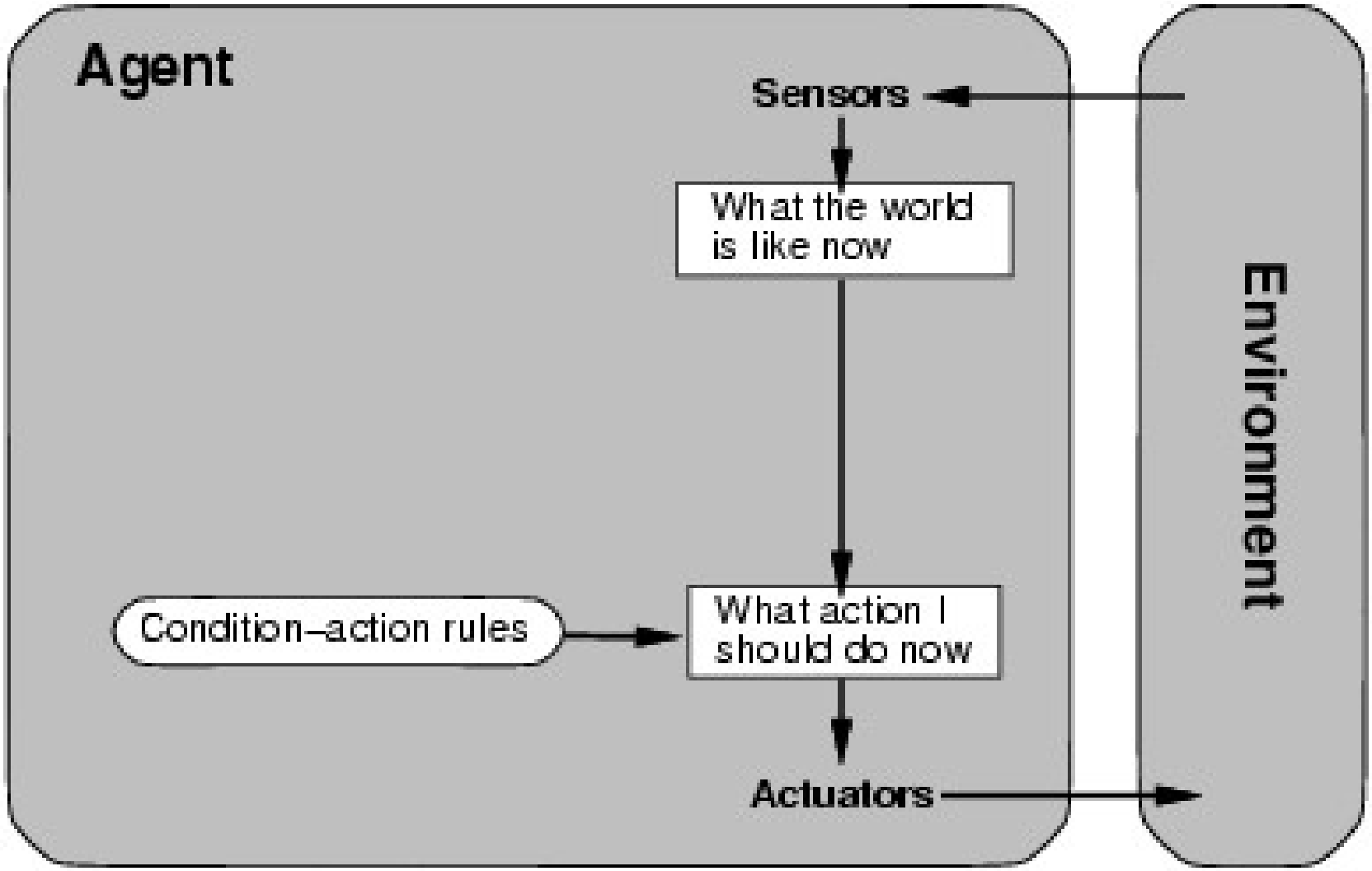
PEAS/PAGE – Example 4 (Interactive English Tutor)

- Performance measure: Maximize student's score on test
- Environment: Set of students
- Actuators: Screen display (exercises, suggestions, corrections)
- Sensors: Keyboard

Agent types (in order of increasing generality)

- Simple reflex agents
- Model-based reflex agents
- Goal-based agents
- Utility-based agents
- Learning agents
- Atomic-factored-structured

Simple reflex agents

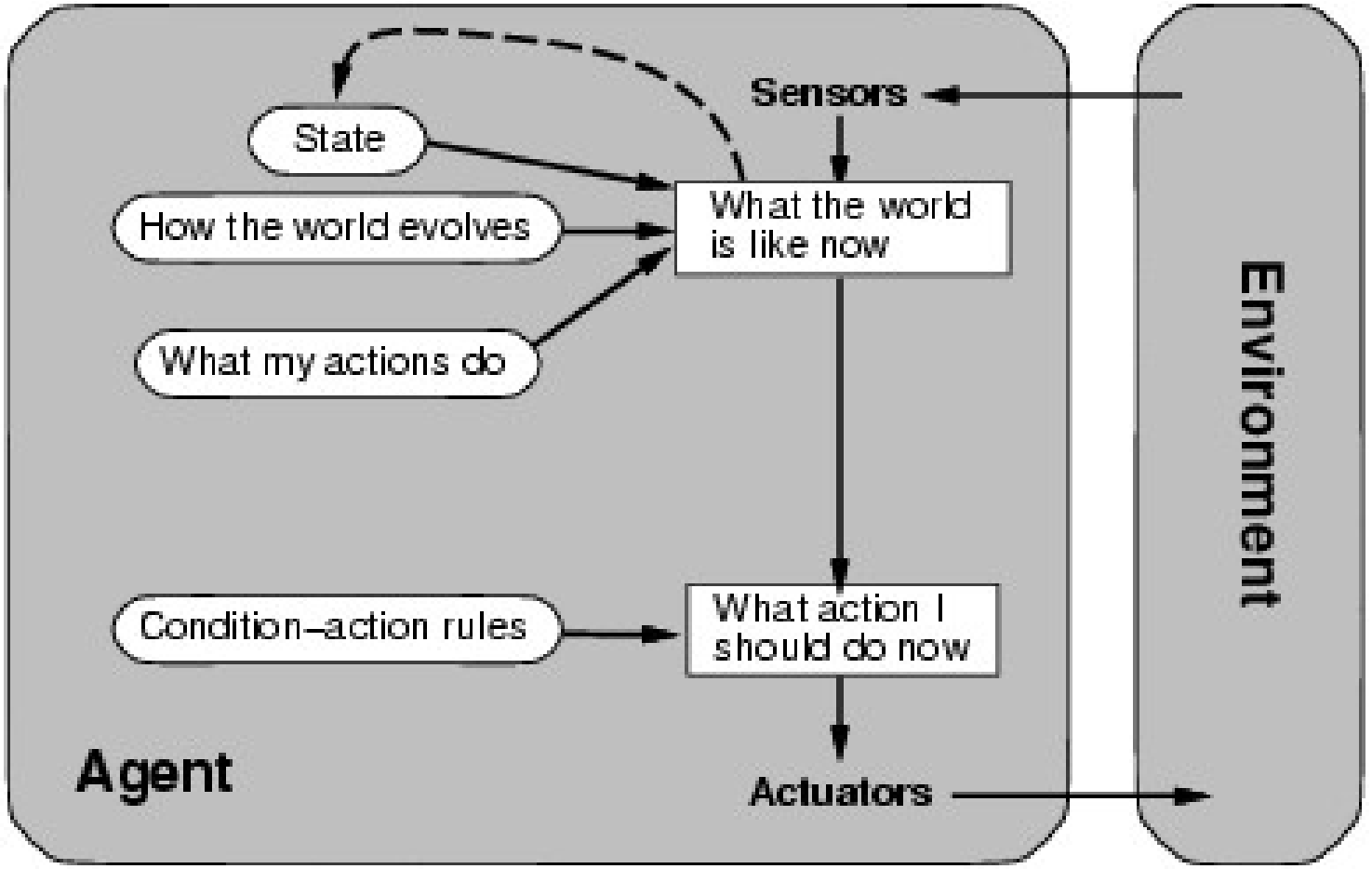


Function SIMPLE-REPLEX-AGENT(percept) returns action

static: rules – a set of condition-action rules

state ← INTERCEPT-INPUT(percept)
rule ← RULE-MATCHING (state, rules)
action ← RULE-ACTION[rule]
return action

Model-based reflex agents



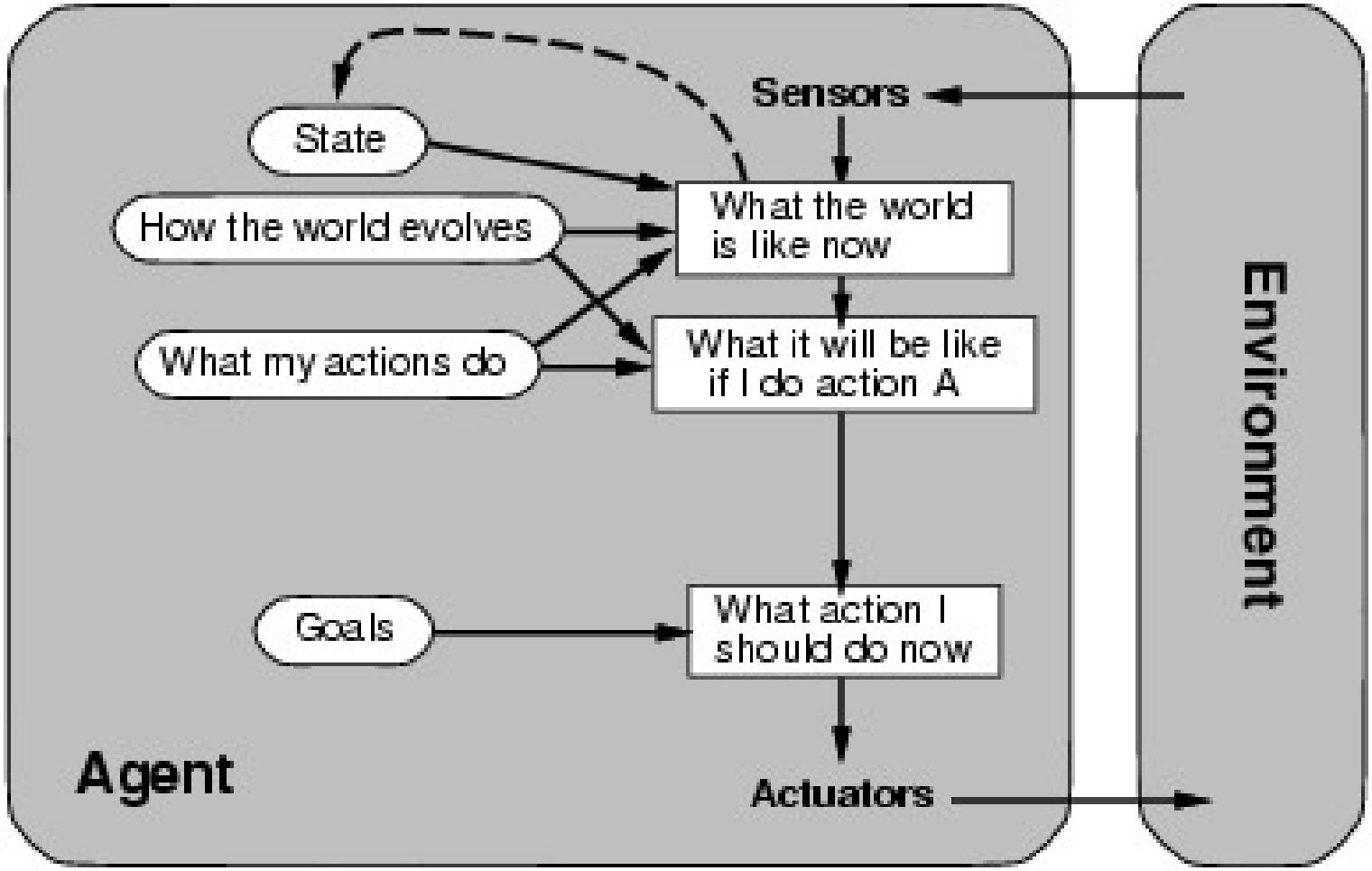
A reflex agent with an internal state

Function REPLEX-AGENT-WITH-STATE (percept) returns action

static: state – a description of the current world state
rules – a set of condition-action rules

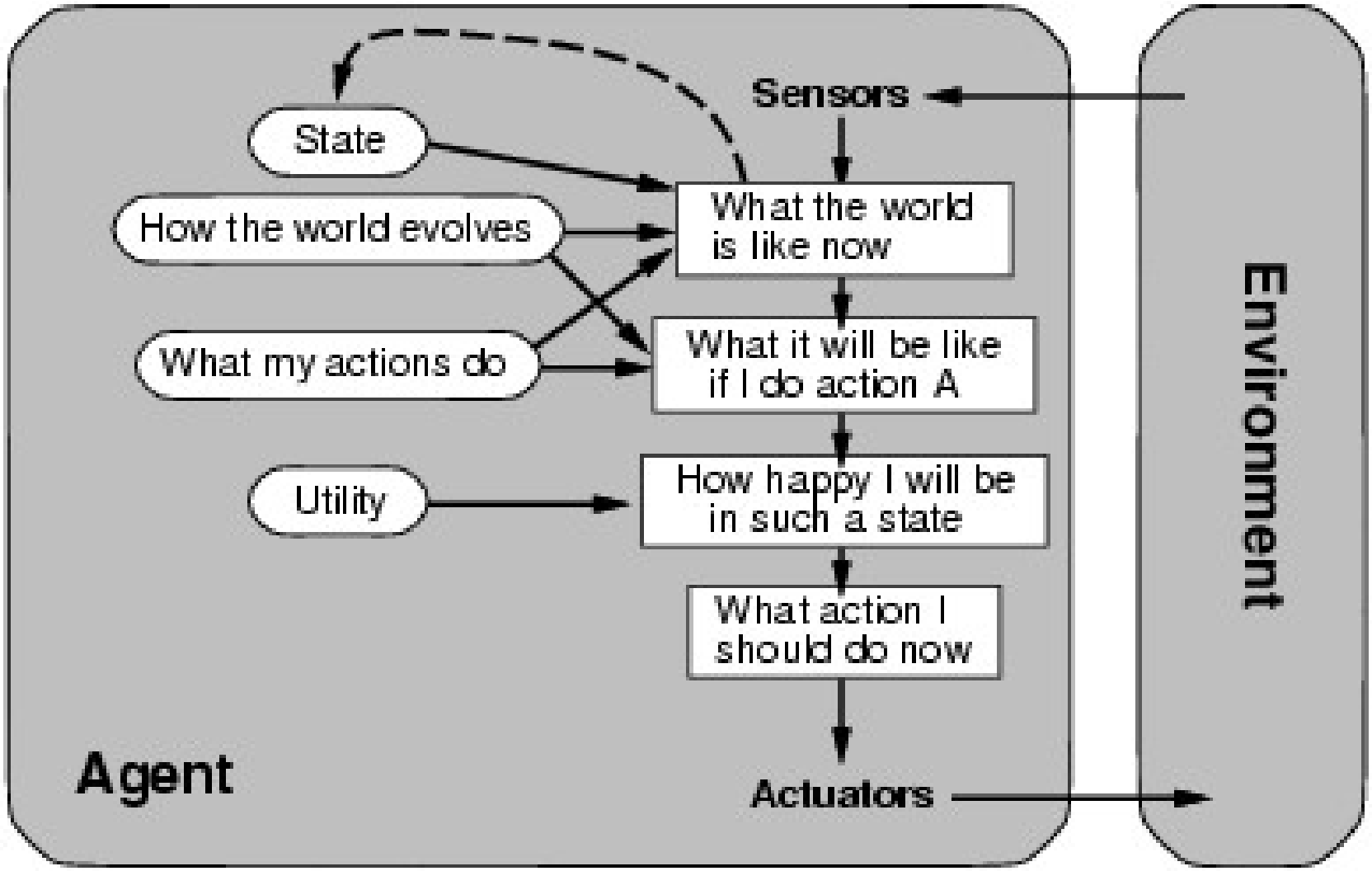
state ← UPDATE-STATE(state, percept)
rule ← RULE-MATCHING (state, rules)
action ← RULE-ACTION[rule]
state ← UPDATE(state, action)
return action

Goal-based agents

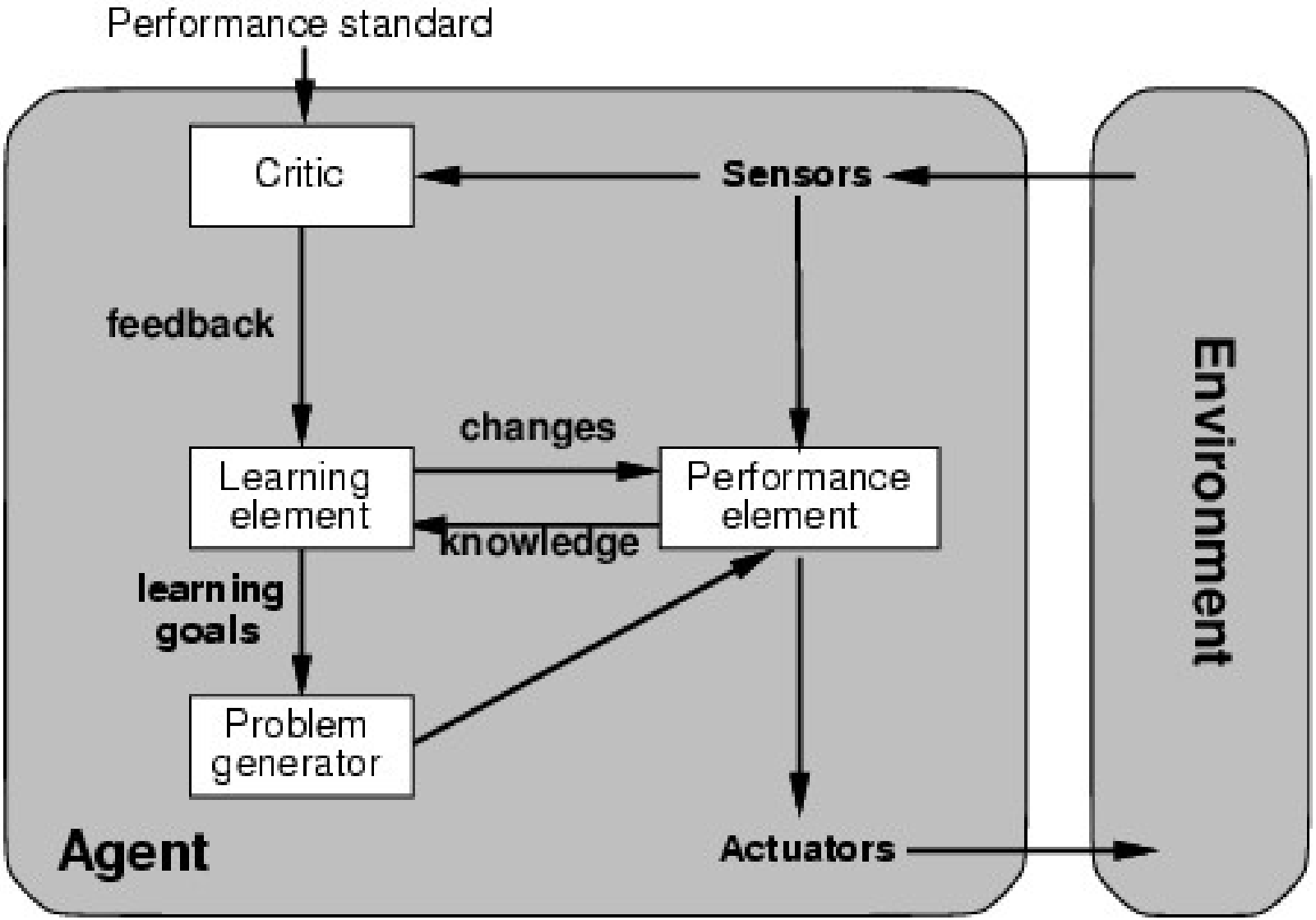


Utility defines a function that maps a state onto a real number (for example degree of happiness) may be a cost which must be traded off with the goals.

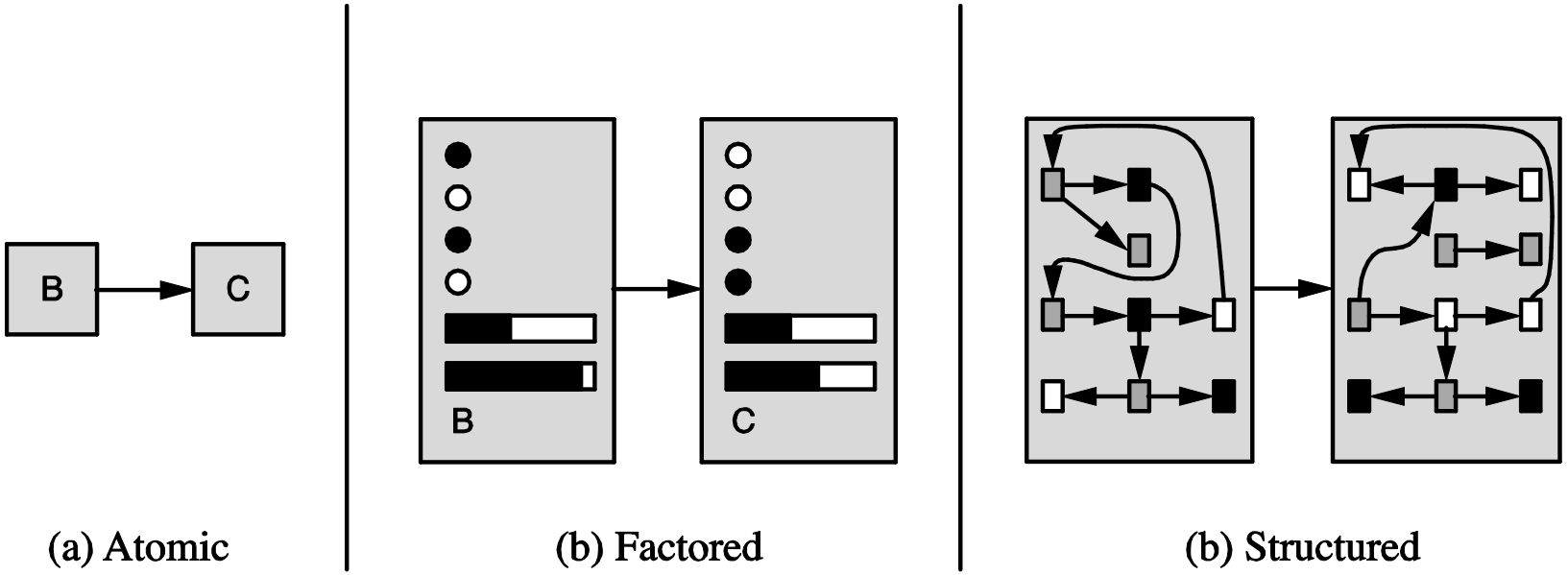
Utility-based agents



Learning agents



Atomic-factored-structured



Environment types

- **Fully observable** (vs. partially observable): An agent's sensors give it access to the complete state of the environment at each point in time.

- **Deterministic** (vs. stochastic): The next state of the environment is completely determined by the current state and the action executed by the agent. (If the environment is deterministic except for the actions of other agents, then the environment is strategic)

- **Episodic** (vs. sequential): The agent's experience is divided into atomic "episodes" (each episode consists of the agent perceiving and then performing a single action), and the choice of action in each episode depends only on the episode itself.

Environment types …contd

- Static (vs. dynamic): The environment is unchanged while an agent is deliberating. (The environment is semidynamic if the environment itself does not change with the passage of time but the agent's performance score does)

- Discrete (vs. continuous): A limited number of distinct, clearly defined percepts and actions.

- Single agent (vs. multiagent): An agent operating by itself in an environment

Environment Programs

- It gives each agent its percepts, gets an action from each agent and then updates its environment
- The program keeps track of the performance measure of each agent

Example: Playing of chess by computer with an opponent of different profile

Possible Assessment Questions -1

1. Give a PEAS description of the task environment for each of the following activities. Include detailed write-up one a ch aspect of the task environment.

PEAS = Performance Environment Actuator Sensor

(a) SSN wants to develop and deploy a **face-recognition based smart attendance system** for its employees and students. Provide a detailed PEAS description for the same.

(b) SSN wants to develop and deploy an **online autonomous proctoring system** that can monitor *n* students through a video communication channel. Provide a detailed PEAS description for the same.

(c) SSN wants to develop a team of robotic agents to participate in **Robo cup soccer competition** (https://2021.robocup.org/.Provide a detailed PEAS description for such a robotic agent.

Possible Assessment Questions -2

With a diagram explain the agent program of simple reflex agent
With a diagram explain the agent program of model based agent
With a diagram explain the agent program of goal based agent
With a diagram explain the agent program of utility based agent
Design a simple reflex agent for the vacuum environment
Can a simple reflex agent with a randomized agent function outperform a simple reflex agent? Design such a agent and measure its performance on several environment