Dr. Ambedkar Institute of technology
Department of Electronics and Instrumentation engineering

21EIT7034- Artificial Intelligence in Industrial Applications

Unit 1: INTELLIGENT AGENT

Prepared by : Ms. Nirmala Bai L
Assistant professor
Dept. of EIE
Dr. AIT

Artificial intelligence

INTELLIGENCE

The ability to learn and solve problems.

Intelligence is composed of:

- **Reasoning**
- **Learning**
- **Problem Solving**
- **Perception**
- **Linguistic Intelligence**

The Seven Types of Intelligence

- **Linguistic--**Enjoy writing, reading, telling stories or doing crossword puzzles.
- **Logical-Mathematical--** Interested in patterns, categories and relationships. ...
- **Bodily-kinesthetic---** Process knowledge through bodily sensations. ...
- **Spatial.--**Think in images and pictures. ...
- **Musical. ...**
- **Interpersonal. ...**
- **Intrapersonal.**

Artificial intelligence

(AI) is the simulation of human **intelligence** processes by machines, especially computer systems.

"to make computers intelligent so that they can act intelligently!",

The question is how much intelligent? How can one judge the intelligence?

as intelligent as humans.

- If the computers can, somehow, solve real-world problems, by improving on their own from the past experiences, they would be called "intelligent".
- Thus, the AI systems are more generic(rather than specific), have the ability to "think" and are more flexible.

Artificial intelligence

- AI refers to the simulation of human intelligence in machines that are programmed to think like humans and mimic their actions.
- The term may also be applied to any machine that exhibits traits associated with a human mind such as learning and problem-solving

Definitions of AI

Systems that think like humans	Systems that think rationally
Systems that act like humans	Systems that act rationally

Approach #1: Acting Humanly

- AI is: "The art of creating machines that perform functions that require intelligence when performed by people" (Kurzweil)

- Ultimately to be tested by the Turing Test
- Some capacities are,
- Natural lang processing
- Knowledge representation
- Automated reasoning

Turing Test

- Human beings are intelligent
- To be called intelligent, a machine must produce responses that are indistinguishable from those of a human

Turing Test in Artificial Intelligence

- Imagine a game of three players having two humans and one computer,
- an interrogator(as human) is isolated from other two players.
- The interrogator job is to try and figure out which one is human and which one is computer by asking questions from both of them.
- To make the things harder computer is trying to make the interrogator guess wrongly.
- In other words computer would try to indistinguishable from human as much as possible.

Approach #2:Thinking Humanly

Cognitive Science

- AI is: "[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning…" (Bellman)
- Goal is to build systems that function internally in some way similar to human mind

Workings of the human mind

- Traditional computer game players typically work much differently than human players
  - Massive look-ahead, minimal "experience"
- People think differently in experience, "big picture", etc.
- *Cognitive science* tries to model human mind based on experimentation
- Cognitive modeling approach tries to act intelligently while actually internally doing something similar to human mind

Approach #3: Thinking rationally

Laws of Thought

- AI is: "The study of the computations that make it possible to perceive, reason, and act" (Winston)
- Approach firmly grounded in logic
- I.e., how can knowledge be represented logically, and how can a system draw deductions?
- Uncertain knowledge? Informal knowledge?
- *"I think ....."*

Approach #4: Acting rationally

- AI is: "The branch of computer science that is concerned with the automation of intelligent behavior" (Luger and Stubblefield)
- The intelligent agent approach
- An *agent* is something that perceives and acts
- Emphasis is on behavior

Definitions of AI

if our system can be more rational than humans in some cases, why not?

focus on action avoids philosophical issues such as "is the system conscious" etc.

Systems that think like humans	Systems that think rationally
Systems that act like humans	Systems that act rationally

We will follow "act rationally" approach

- acting rationally/like a human presumably requires (some sort of) thinking rationally/like a human,
- humans much more rational anyway in complex domains

Intelligent Agents

- Agents
- Environments
- Good behavior
- Concept of rationality
- Nature of environments
- Structure of agents

Intelligent Agent

- What is an agent ?
  - An **agent** is anything that **perceive** its environment through **sensors** and **acts** upon that environment through **actuators.**
  - Example:
    - Human is an agent
    - A robot is also an agent with cameras and motors
    - A thermostat detecting room temperature.

Intelligent Agent

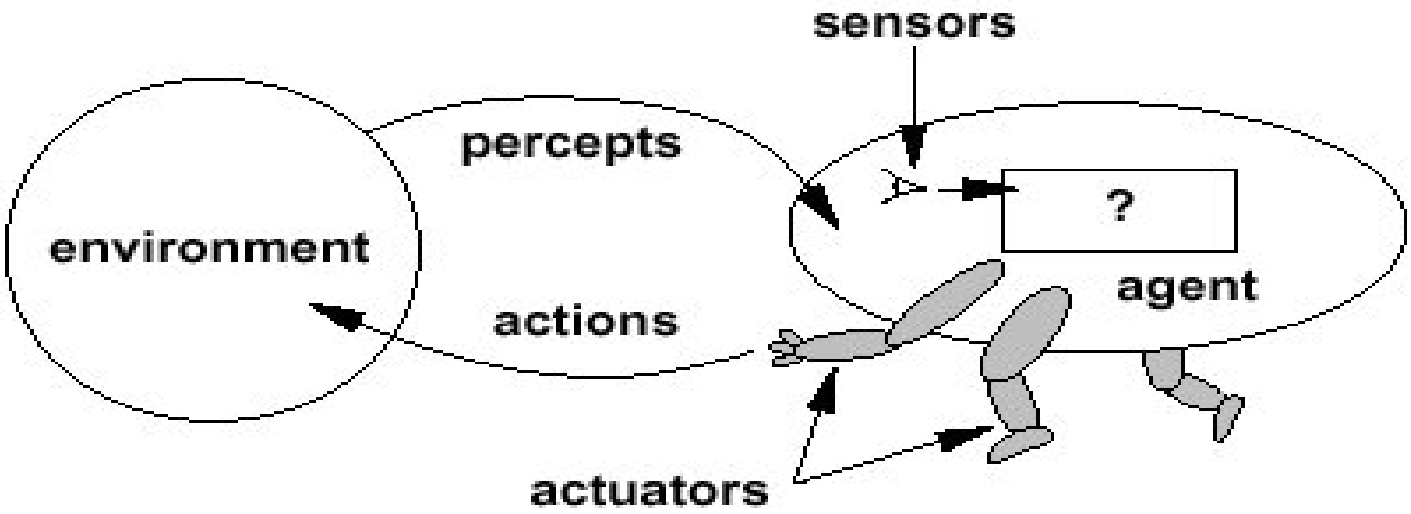
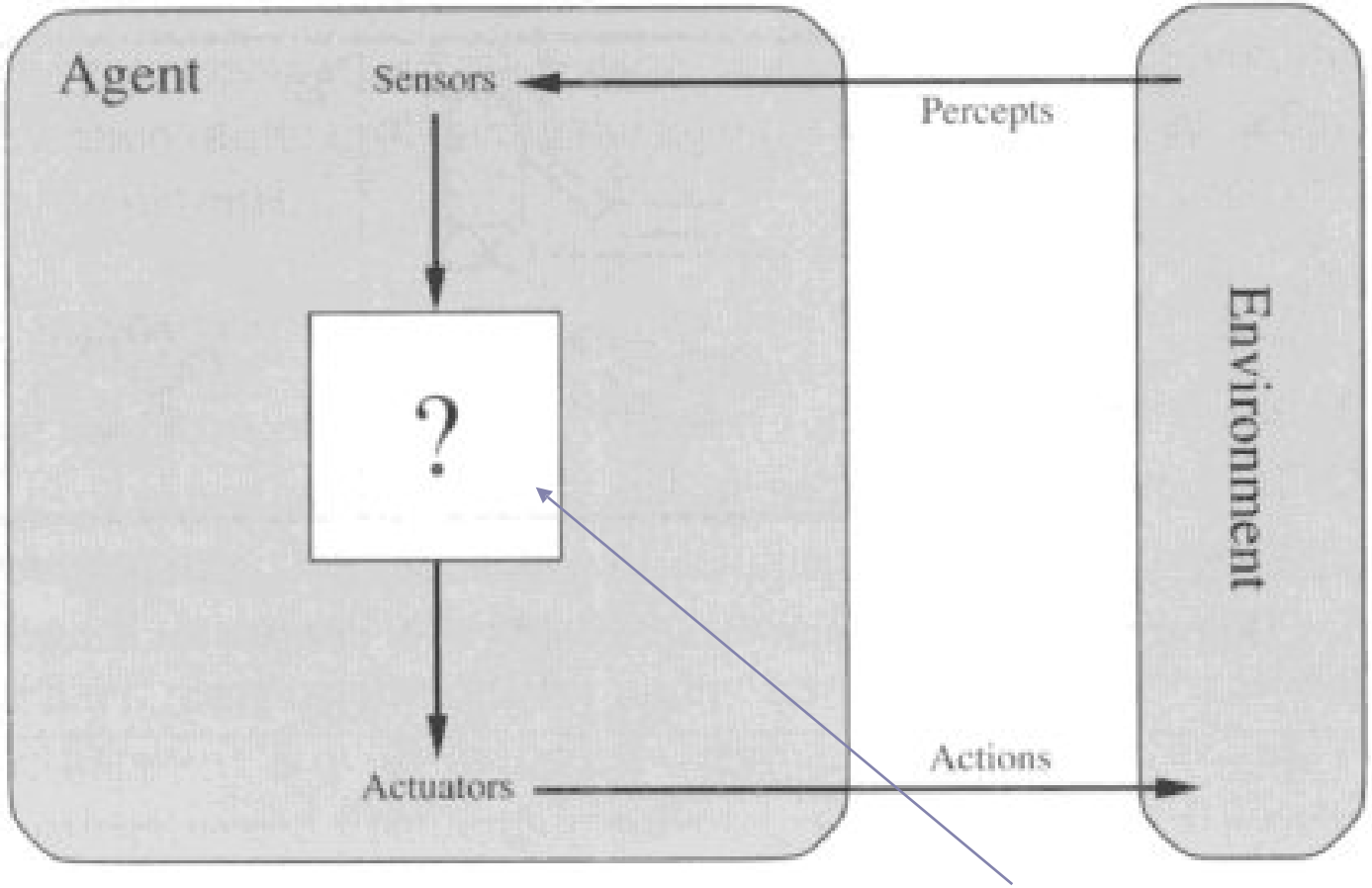


Diagram of an agent



What AI should fill

Examples of agents in different types of applications

Agent type	Percepts	Actions	Goals	Environment
Medical diagnosis system	Symptoms, findings, patient's answers	Questions, tests, treatments	Healthy patients, minimize costs	Patient, hospital
Satellite image analysis system	Pixels of varying intensity, color	Print a categorization of scene	Correct categorization	Images from orbiting satellite
Part-picking robot	Pixels of varying intensity	Pick up parts and sort into bins	Place parts in correct bins	Conveyor belts with parts
Refinery controller	Temperature, pressure readings	Open, close valves; adjust temperature	Maximize purity, yield, safety	Refinery
Interactive English tutor	Typed words	Print exercises, suggestions, corrections	Maximize student's score on test	Set of students

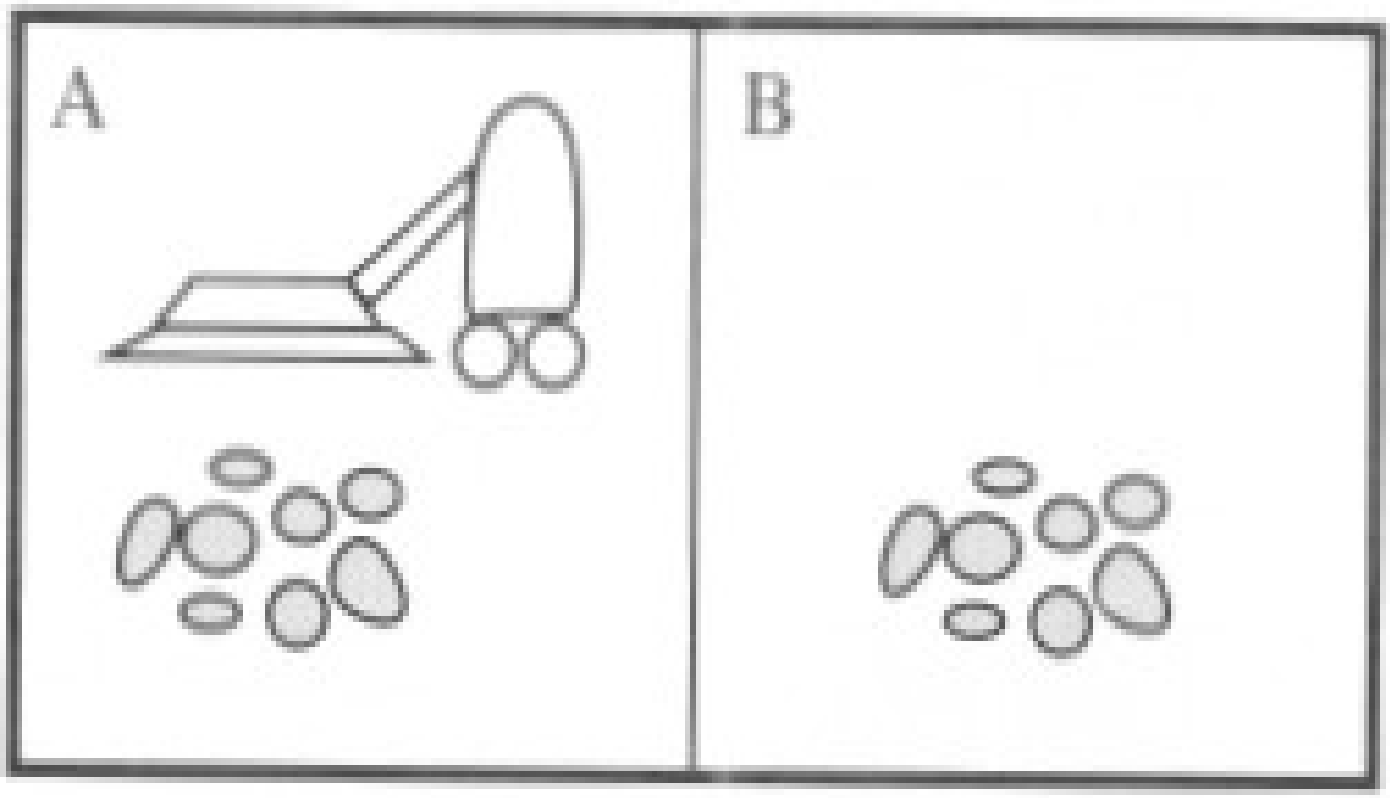
Simple Terms

- Percept
  - Agent's perceptual inputs at any given instant
- Percept sequence
  - Complete history of everything that the agent has ever perceived.

Agent function & program

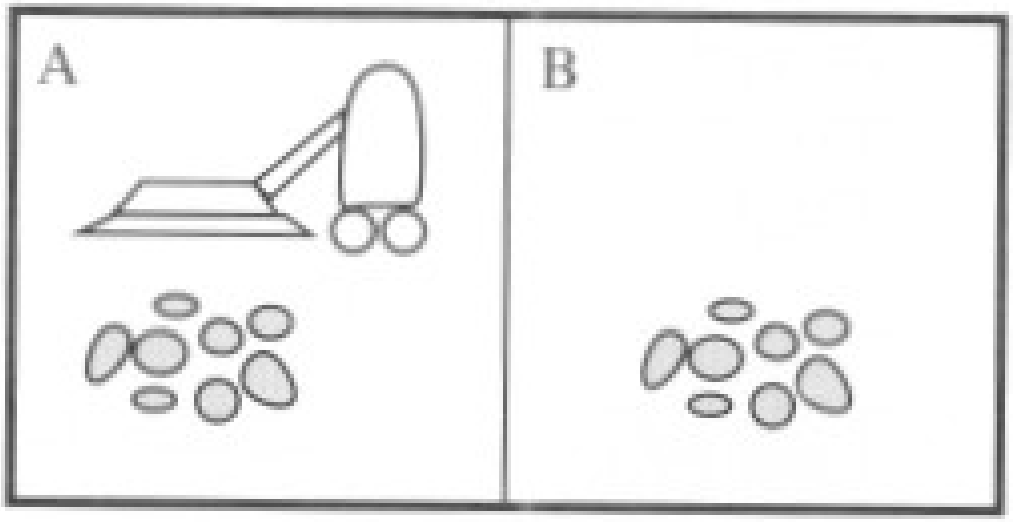
- Agent's behavior is *mathematically* described by
  - **Agent function**
  - A function mapping any given percept sequence to an action
- *Practically* it is described by
  - An **agent program**
  - The real implementation

Vacuum-cleaner world



Vacuum-cleaner world

- Perception: Clean or Dirty? where it is in?
- Actions: Move left, Move right, suck, do nothing



Vacuum-cleaner world

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
⋮	⋮

Figure 2.3 Partial tabulation of a simple agent function for the vacuum-cleaner world shown in Figure 2.2.

Program implements the agent function tabulated in Fig. 2.3

Function Reflex-Vacuum-Agent([*location*,*status*]) return an action

If *status* = *Dirty* **then return** *Suck*

else if *location* = *A* **then return** *Right*

else if *location* = *B* **then return** *left*

Concept of Rationality

- Rational agent
  - One that does the right thing = every entry in the table for the agent function is correct (rational).
- What is correct?
  - The actions that cause the agent to be most successful
  - So we need ways to measure success.

Performance measure

- Performance measure
  - An objective function that determines

    How the agent does successfully
    - E.g., 90% or 30% ?
- An agent, based on its percepts
  - ☾ action sequence :

  if desirable, it is said to be performing well.
  - No universal performance measure for all agents

Performance measure

- A general rule:
  - Design performance measures according to
    - What one actually wants in the environment
    - Rather than how one thinks the agent should behave
- E.g., in vacuum-cleaner world
  - We want the floor clean, no matter how the agent behave
  - We don't restrict how the agent behaves

Rationality

- What is rational at any given time depends on four things:
  - The performance measure defining the criterion of success
  - The agent's prior knowledge of the environment
  - The actions that the agent can perform
  - The agents's percept sequence up to now

Rational agent

- For each possible percept sequence,
  - an rational agent should select
    - an action expected to maximize its performance measure, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has
- E.g., an exam
  - Maximize marks, based on

the questions on the paper & your knowledge

Example of a rational agent

- Performance measure
  - Awards one point for each clean square
    - at each time step, over 10000 time steps
- Prior knowledge about the environment
  - The geography of the environment
  - Only two squares
  - The *effect* of the actions

Example of a rational agent

- Actions that can perform
  - Left, Right, Suck and NoOp
- Percept sequences
  - Where is the agent?
  - Whether the location contains dirt?
- Under this circumstance, the agent is rational.

Omniscience

- An omniscient agent
  - Knows the *actual* outcome of its actions in advance
  - No other possible outcomes
  - However, impossible in real world
- An example
  - crossing a street but died of the fallen cargo door from 33,000ft ☾ irrational?

Omniscience

- Based on the circumstance, it is rational.
- As rationality maximizes
  - *Expected* performance
- Perfection maximizes
  - *Actual* performance
- Hence rational agents are not ***omniscient***.

Learning

- Does a rational agent depend on only current percept?
  - No, the past percept sequence should also be used
  - This is called **learning**
  - After experiencing an episode, the agent
    - should adjust its behaviors to perform better for the same job next time.

Autonomy

- If an agent just relies on the prior knowledge of its designer rather than its own percepts then the agent lacks ***autonomy***

A rational agent should be autonomous- it should learn what it can to compensate for partial or incorrect prior knowledge.

- E.g., a clock
  - No input (percepts)
  - Run only but its own algorithm (prior knowledge)
  - No learning, no experience, etc.

Software Agents

Sometimes, the environment may not be the real world

- E.g., flight simulator, video games, Internet
- They are all artificial but very complex environments
- Those agents working in these environments are called
  - Software agent (softbots)
  - Because all parts of the agent are software

Task environments

- Task environments are the problems
  - While the rational agents are the solutions
- Specifying the task environment
  - PEAS description as fully as possible
    - Performance
    - Environment
    - Actuators
    - Sensors

  In designing an agent, the first step must always be to specify the task environment as fully as possible.
- Use automated taxi driver as an example

Characterizing a Task Environment

- Must first specify the setting for intelligent agent design.
- **PEAS: Performance measure, Environment, Actuators, Sensors**
- **Example:** the task of designing a self-driving car
  - **Performance measure : Safe, fast, legal, comfortable trip**
  - **Environment : Roads, other traffic, pedestrians**
  - **Actuators : Steering wheel, accelerator, brake, signal, horn**
  - **Sensors : Cameras, LIDAR (light/radar), speedometer, GPS, odometer engine sensors, keyboard**

Task environments

- Performance measure
  - How can we judge the automated driver?
  - Which factors are considered?
    - getting to the correct destination
    - minimizing fuel consumption
    - minimizing the trip time and/or cost
    - minimizing the violations of traffic laws
    - maximizing the safety and comfort, etc.

Task environments

- Environment
  - A taxi must deal with a variety of roads
  - Traffic lights, other vehicles, pedestrians, stray animals, road works, police cars, etc.
  - Interact with the customer

Task environments

- Actuators (for outputs)
  - Control over the accelerator, steering, gear shifting and braking
  - A display to communicate with the customers
- Sensors (for inputs)
  - Detect other vehicles, road situations
  - GPS (Global Positioning System) to know where the taxi is
  - Many more devices are necessary

Task Environments

- A sketch of automated taxi driver

Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi driver	Safe, fast, legal, comfortable trip, maximize profits	Roads, other traffic, pedestrians, customers	Steering, accelerator, brake, signal, horn, display	Cameras, sonar, speedometer, GPS, odometer, accelerometer, engine sensors, keyboard

Figure 2.4 PEAS description of the task environment for an automated taxi.

Task Environments

Write the PEAS descriptions for the following:

1. Medical diagnosis system
2. Part picking robot
3. Interactive English teacher

Task Environments

Agent type	Performance measure P	Environment E	Actuators A	Sensors S
Medical diagnosis system	Patients health,cost,accuracy	Hospital ,patient, doctors,staff	Tests ,treatments,	Keyboard entries of the symtoms,
Part picking robot	No of parts picked, placed in correct boxs	Conveyer belt	Robot hand	Camera,sensors
Interactive English teacher	Students performance, score	Students, testing	Exercises,assignments,display like monitor,black board	Monitor for entries

Properties of task environments

1. Fully observable vs. Partially observable

- If an agent's sensors give it access to the complete state of the environment at each point in time then the environment is effectively fully observable
  - if the sensors detect all aspects
  - That are relevant to the choice of action

- **Partially observable**

An environment might be Partially observable because of noisy and inaccurate sensors or because parts of the state are simply missing from the sensor data.

Example:

- A local dirt sensor of the cleaner cannot tell
- Whether other squares are clean or not

Properties of task environments

2. Deterministic vs. stochastic

- next state of the environment Completely determined by the current state and the actions executed by the agent, then the environment is deterministic, otherwise, it is Stochastic.
- Strategic environment: deterministic except for actions of other agents

-Cleaner and taxi driver are:

- Stochastic because of some unobservable aspects ☾ noise or unknown

Properties of task environments

3. Episodic vs. sequential

- An episode = agent's single pair of perception & action
- The quality of the agent's action does not depend on other episodes
  - Every episode is independent of each other
- Episodic environment is simpler
  - The agent does not need to think ahead

- **Sequential**
  - Current action may affect all future decisions

-Ex. Taxi driving and chess.

Properties of task environments

4. Static vs. dynamic

- A dynamic environment is always changing over time
  - E.g., the number of people in the street
- While static environment
  - E.g., the destination

Semidynamic

- environment is not changed over time
- but the agent's performance score does

Properties of task environments

5. Discrete vs. continuous

- If there are a limited number of distinct states, clearly defined percepts and actions, the environment is discrete

E.g., Chess game

- Continuous: Taxi driving

Properties of task environments

6. Single Agent vs. Multi Agent

- Playing a crossword puzzle – single agent
- Chess playing – two agents
- Competitive multi agent environment
  - Chess playing
- Cooperative multi agent environment
  - Automated taxi driver
  - Avoiding collision

Properties of task environments

7. Known vs. unknown

This distinction refers not to the environment itslef but to the agent's (or designer's) state of knowledge about the environment.

-In known environment, the outcomes for all actions are given. (example: solitaire card games).

- If the environment is unknown, the agent will have to learn how it works in order to make good decisions.(example: new video game).

Examples of task environments

Task Environment
Crossword puzzle Chess with a clock
Poker Backgammon
Taxi driving Medical diagnosis
Image-analysis Part-picking robot
Refinery controller Interactive English tutor

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	Strategic	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

Figure 2.6 Examples of task environments and their characteristics.

Structure of agents

- Agent = architecture + program
  - Architecture = some sort of computing device (sensors + actuators)
  - (Agent) Program = some function that implements the agent mapping = "?"
  - Agent Program = Job of AI

Agent programs

- Input for Agent Program
  - Only the current percept
- Input for Agent Function
  - The entire percept sequence
  - The agent must remember all of them
- Implement the agent program as
  - A look up table (agent function)

Agent programs

- Skeleton design of an agent program

```
function TABLE-DRIVEN-AGENT(percept) returns action
  static: percepts, a sequence, initially empty
          table, a table, indexed by percept sequences, initially fully specified

  append percept to the end of percepts
  action ← LOOKUP(percepts, table)
  return action
```

Agent Programs

- P = the set of possible percepts
- T= lifetime of the agent
  - The total number of percepts it receives
- Size of the look up table $\sum_{t=1}^{T} |P|^t$
- Consider playing chess
  - P =10, T=150
  - Will require a table of at least 10^{150} entries

Agent programs

- Despite of huge size, look up table does what we want.
- The key challenge of AI
  - Find out how to write programs that, to the extent possible, produce rational behavior
    - From a small amount of code
    - Rather than a large amount of table entries
  - E.g., a five-line program of Newton's Method

V.S. huge tables of square roots, sine, cosine, …

Types of agent programs

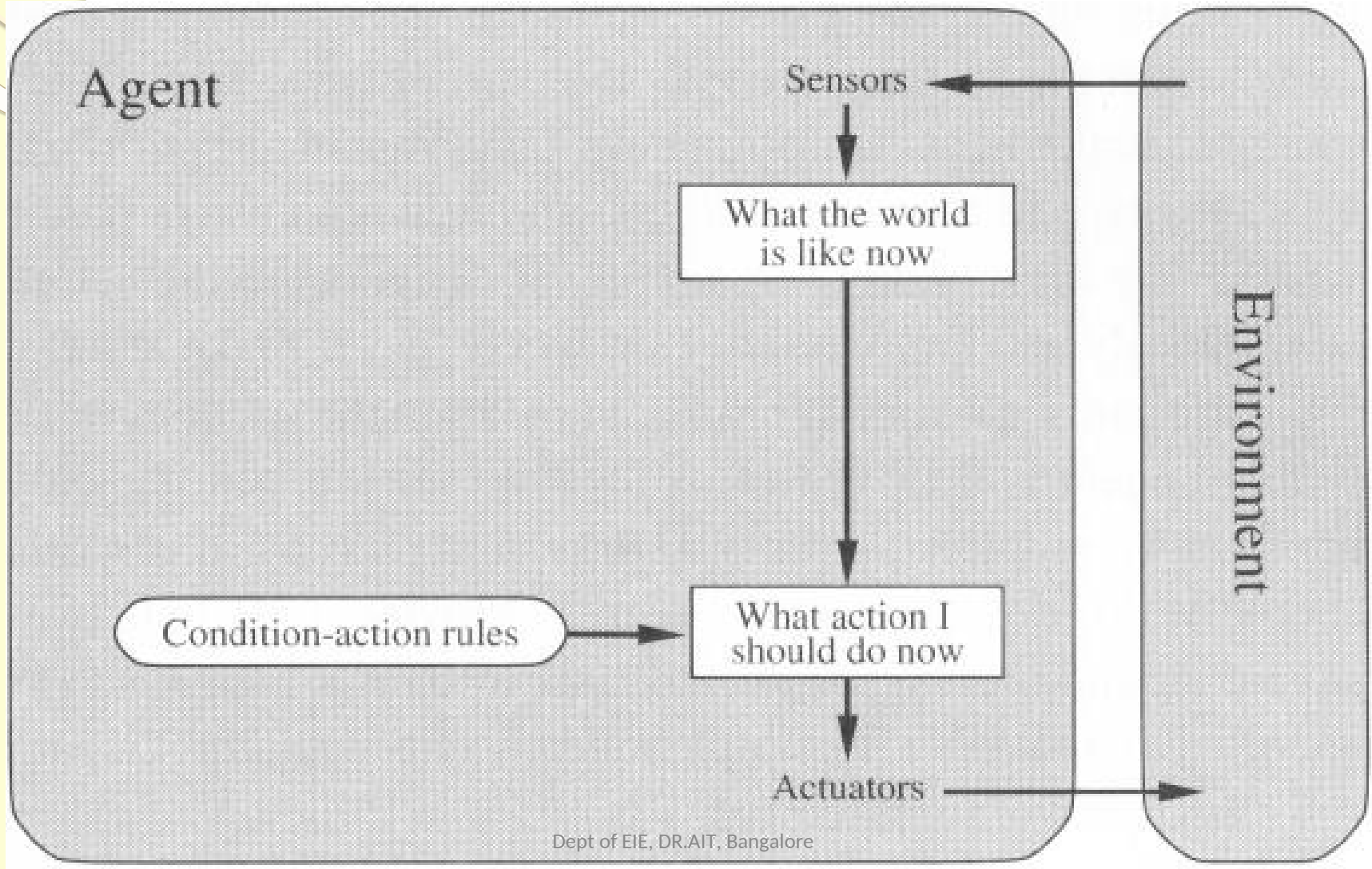
Four types

- **Simple reflex agent**
- **Model-based reflex agent**
- **Goal-based agent**
- **Utility-based agent**

Simple Reflex Agent

- It uses just ***condition-action rules***
  - The rules are like the form "if … then …"
  - efficient but have narrow range of applicability
  - Because knowledge sometimes cannot be stated explicitly
  - Work only
    - if the environment is fully observable

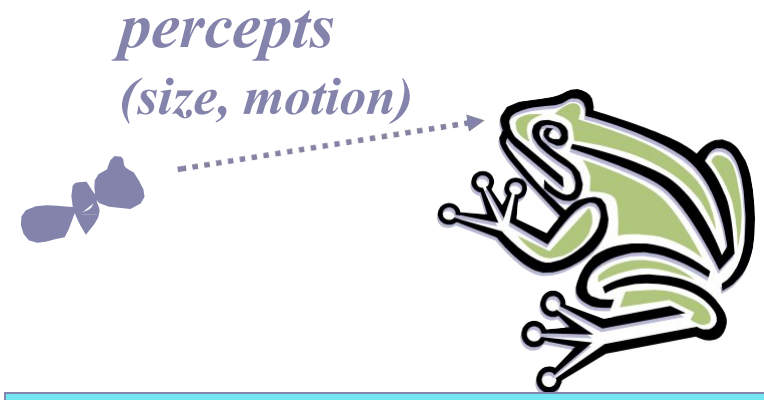
Simple Reflex Agent



Simple Reflex Agent

```
function SIMPLE-REFLEX-AGENT(percept) returns action
    static: rules, a set of condition-action rules

    state ← INTERPRET-INPUT(percept)
    rule ← RULE-MATCH(state, rules)
    action ← RULE-ACTION[rule]
    return action
```

A Simple Reflex Agent in Nature



RULES:

(1) If small moving object,
then activate SNAP

(2) If large moving object,
then activate AVOID and inhibit SNAP

ELSE (not moving) then NOOP

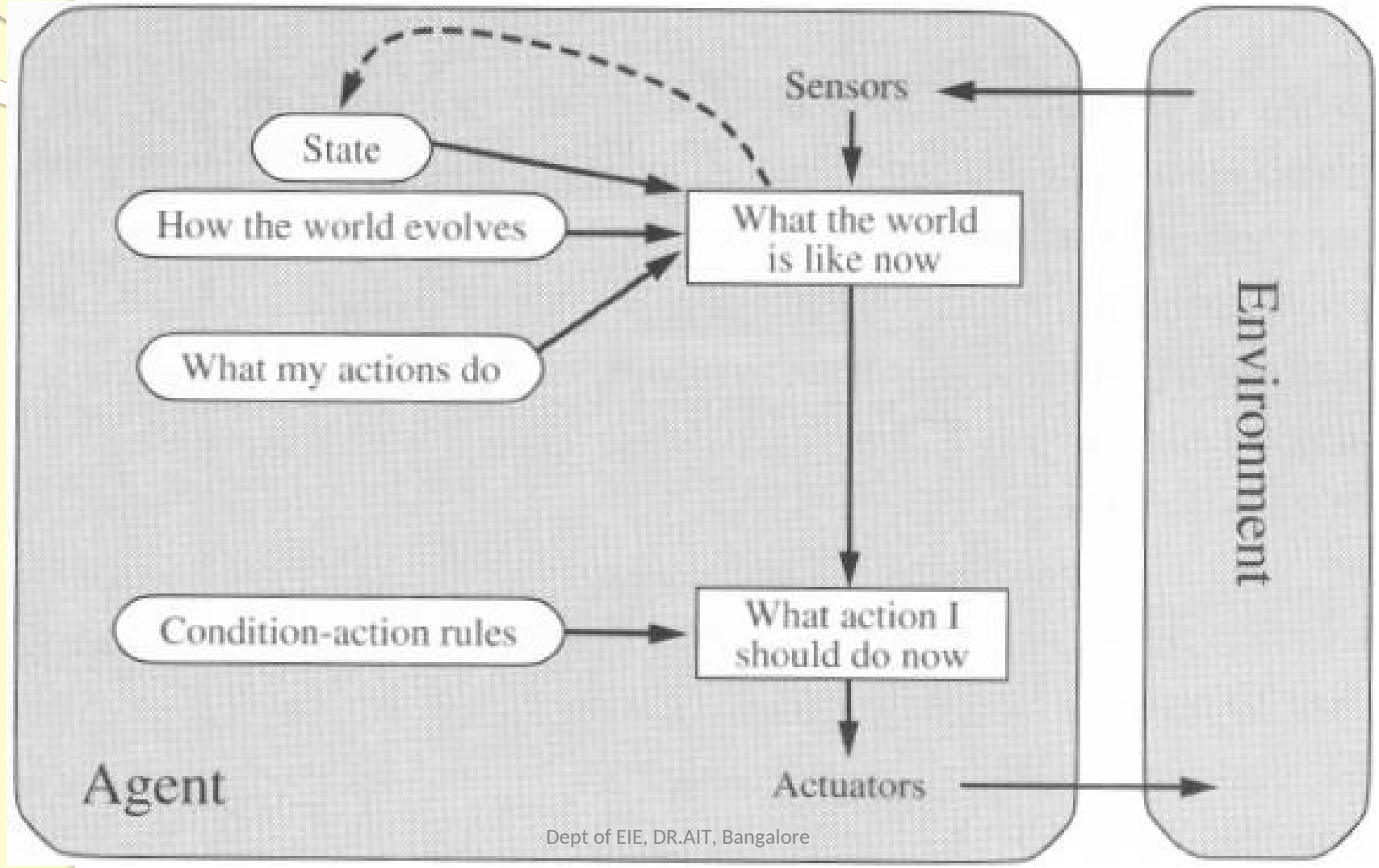
needed for completeness

Action: SNAP or AVOID or NOOP

Model-based Reflex Agents

- For the world that is partially observable
  - the agent has to keep track of an internal state
    - That depends on the percept history
    - Reflecting some of the unobserved aspects
    - E.g., driving a car and changing lane
- Requiring two types of knowledge
  - How the world evolves independently of the agent
  - How the agent's actions affect the world

Model-based Reflex Agent



Model-based Reflex Agent

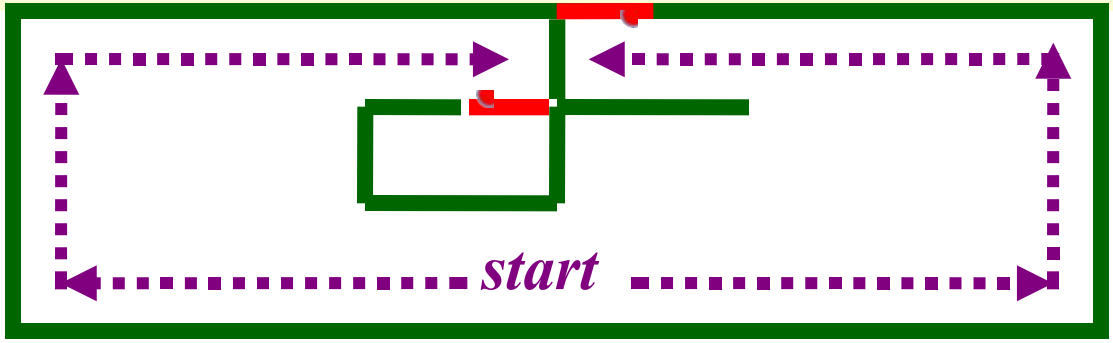
```
function REFLEX-AGENT-WITH-STATE(percept) returns action
  static: state, a description of the current world state
          rules, a set of condition-action rules

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

The agent is with memory

Example Table Agent With Internal State

IF	THEN
Saw an object ahead, and turned right, and it's now clear ahead	Go straight
Saw an object Ahead, turned right, and object ahead again	Halt
See no objects ahead	Go straight
See an object ahead	Turn randomly

Example Reflex Agent With Internal State: **Wall-Following**



Actions: left, right, straight, open-door

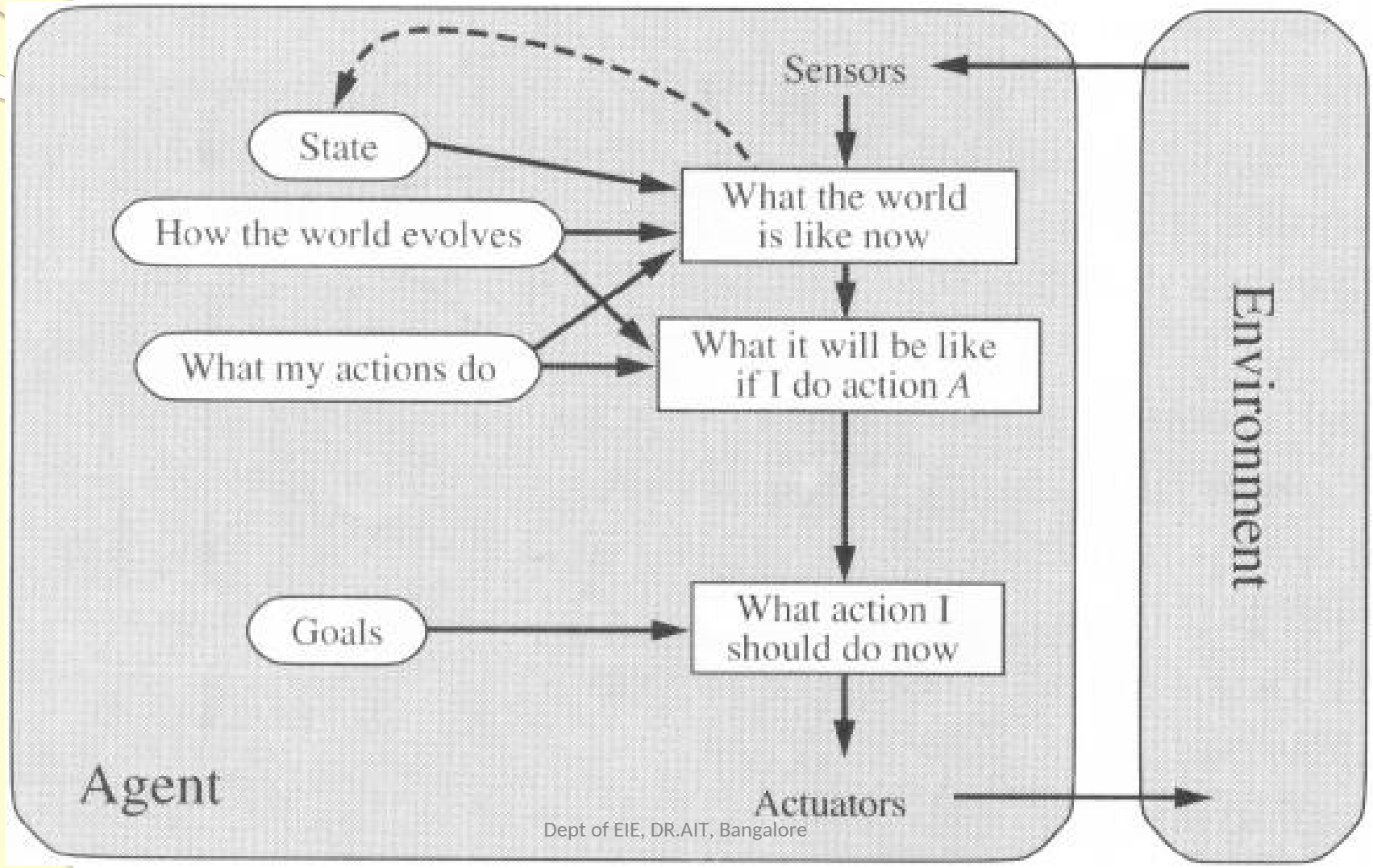
Rules:

1. If open(left) & open(right) and open(straight) then choose randomly between right and left
2. If wall(left) and open(right) and open(straight) then straight
3. If wall(right) and open(left) and open(straight) then straight
4. If wall(right) and open(left) and wall(straight) then left
5. If wall(left) and open(right) and wall(straight) then right
6. If wall(left) and door(right) and wall(straight) then open-door
7. If wall(right) and wall(left) and open(straight) then straight.
8. (Default) Move randomly

Goal-based Agent

- Current state of the environment is always not enough
- The goal is another issue to achieve
  - Judgment of rationality / correctness
- Actions chosen ☾ goals, based on
  - the current state
  - the current percept

Goal-based Agent



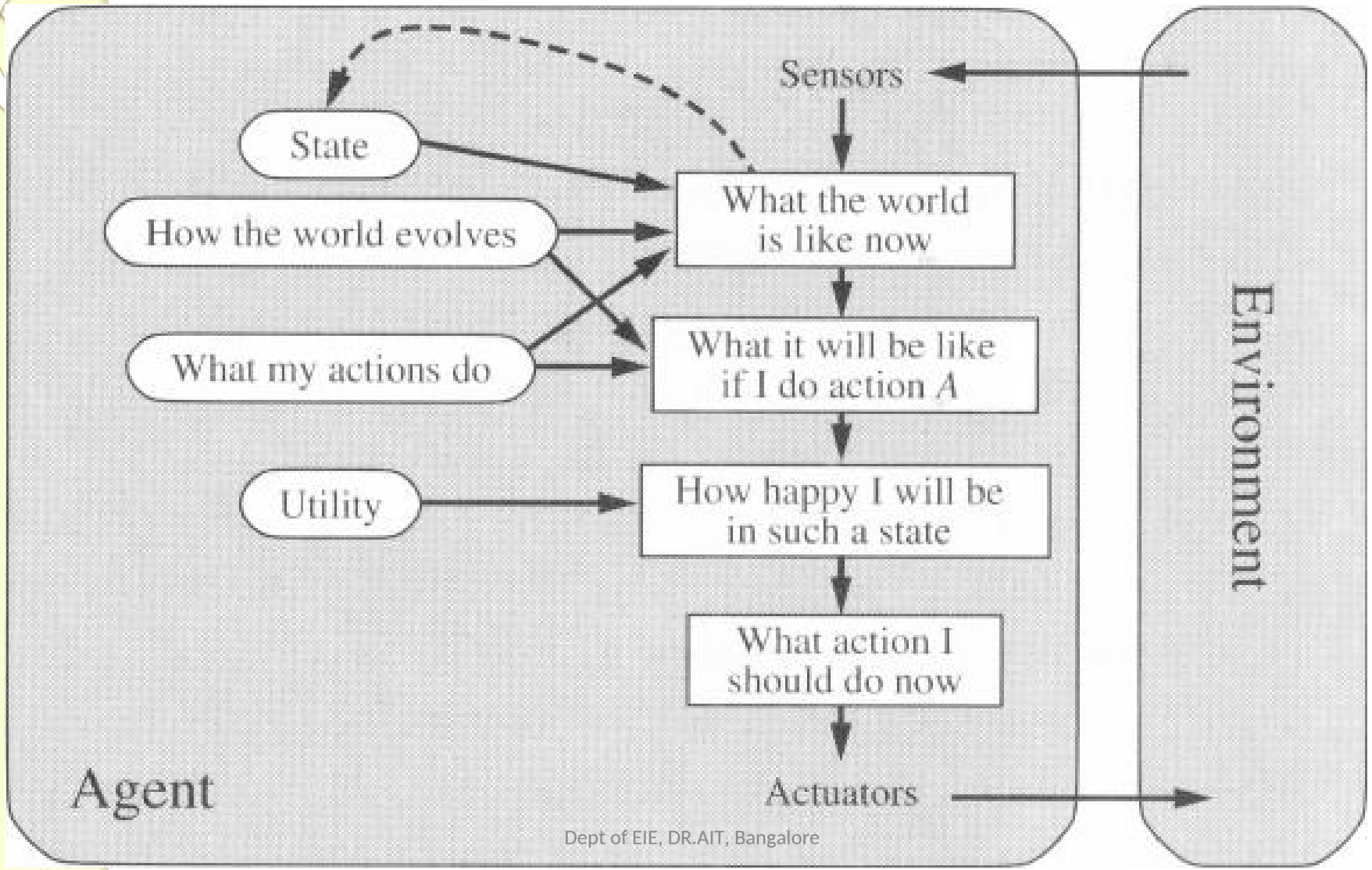
Goal-based Agents

- Conclusion
  - Goal-based agents are less efficient
  - but more flexible
    - Agent ≈ Different goals ≈ different tasks
  - Search and planning
    - two other sub-fields in AI
    - to find out the action sequences to achieve its goal

Utility-based Agent

- Goals alone are not enough
  - to generate **high-quality** behavior
  - E.g. meals in Canteen, good or not ?
- Many action sequences ☾ the goals
  - some are better and some worse
  - If **goal** means success, then **utility** means the degree of success (how successful it is)

Utility-based agents



Utility-based Agent

- it is said state A has higher utility
  - If state A is more preferred than others
- Utility is therefore a function
  - that maps a state onto a real number
  - the degree of success

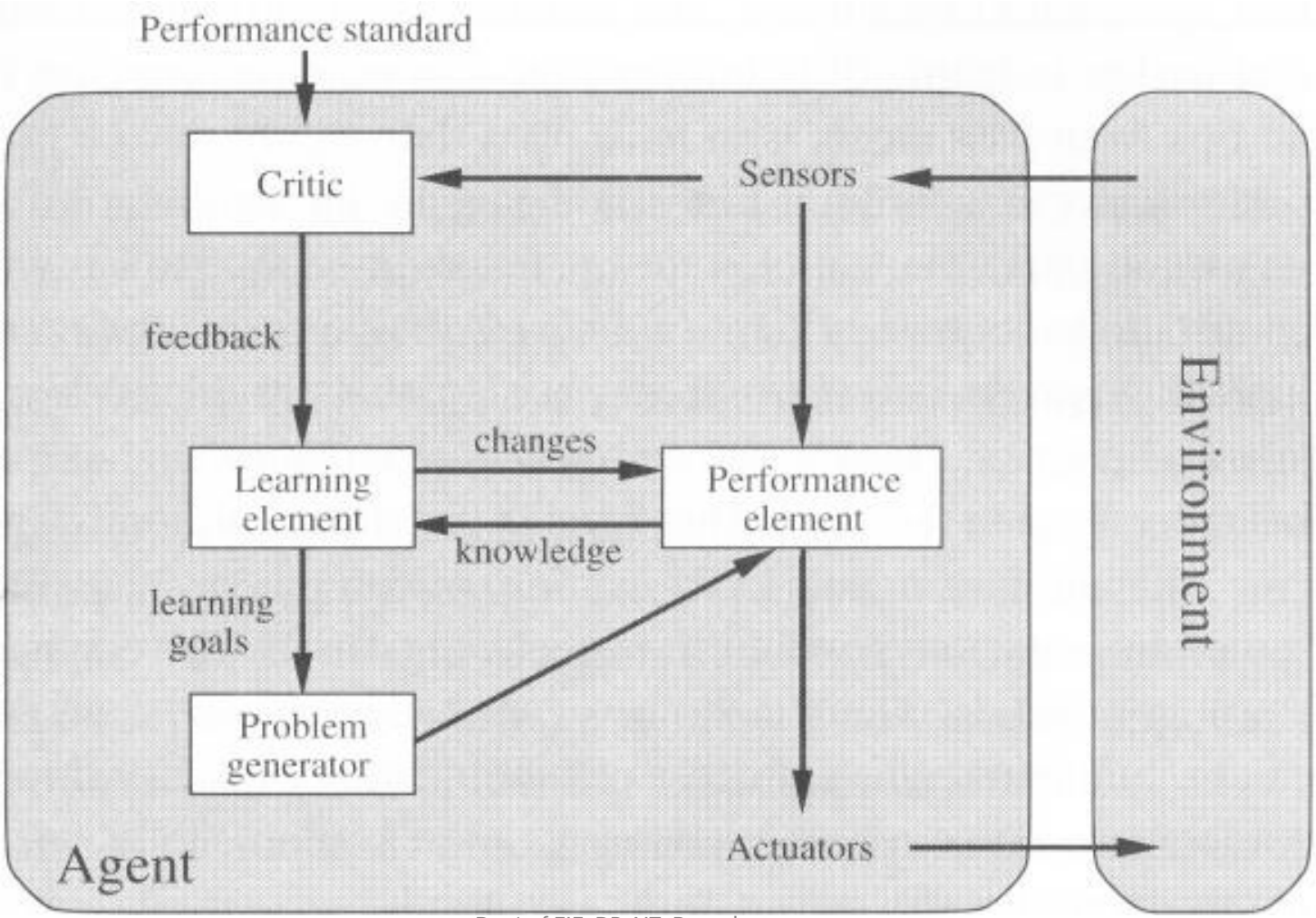
Utility-based Agent

- Utility has several advantages:
  - When there are conflicting goals,
    - Only some of the goals but not all can be achieved
    - utility describes the appropriate trade-off
  - When there are several goals
    - None of them are achieved **certainly**
    - utility provides a way for the decision-making

Learning Agent

- After an agent is programmed, can it work immediately?
  - No, it still need teaching
- In AI,
  - Once an agent is done
  - We teach it by giving it a set of examples
  - Test it by using another set of examples
- We then say the agent learns
  - A learning agent

Learning Agents



Learning Agent

- Four conceptual components
  - Learning element
    - Making improvement
  - Performance element
    - Selecting external actions
  - Critic
    - Tells the Learning element how well the agent is doing with respect to fixed performance standard.

    (Feedback from user or examples, good or not?)
  - Problem generator
    - Suggest actions that will lead to new and informative experiences.

LET'S RE VISIT….

- 4 Approaches to AI
- Act as human
- Think as human
- Act rationally
- Think rationally
- Intelligent agent--
  - Sensor
  - Mathematical Function
  - Actuator

Main terminology used

- Percept
- Percept sequence

Ex: vacuum cleaner agent,with pseudo code

Variations in agent:

1. Rational agent
2. Autonomous agent
3. Omniscient agent
4. Software agent

Task environment

Main characteristics of task environment

P-Performance E-environment

A-actuator S-sensor

Properties Of task Environment

1. **Fully observable vs. Partially observable**
2. **Discrete vs. continuous**
3. **Episodic vs. sequential**
4. **Dynamic vs. static**
5. **Deterministic vs. stochastic**
6. **Single vs multi agent**
7. **Known vs. unknown**

Structure Of Agent

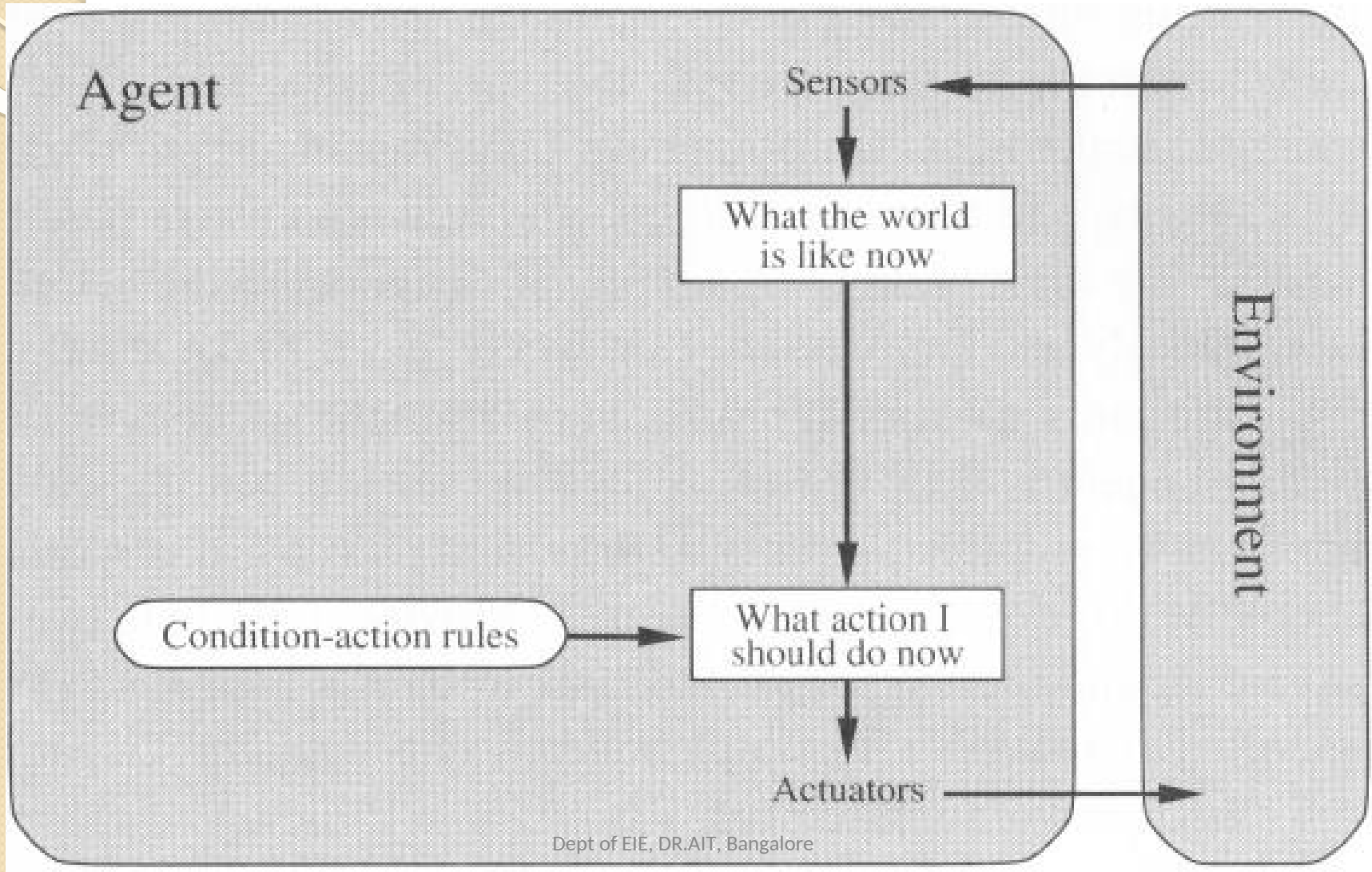
Agent = architecture + program

Ex:Table driven agent with pseudo code

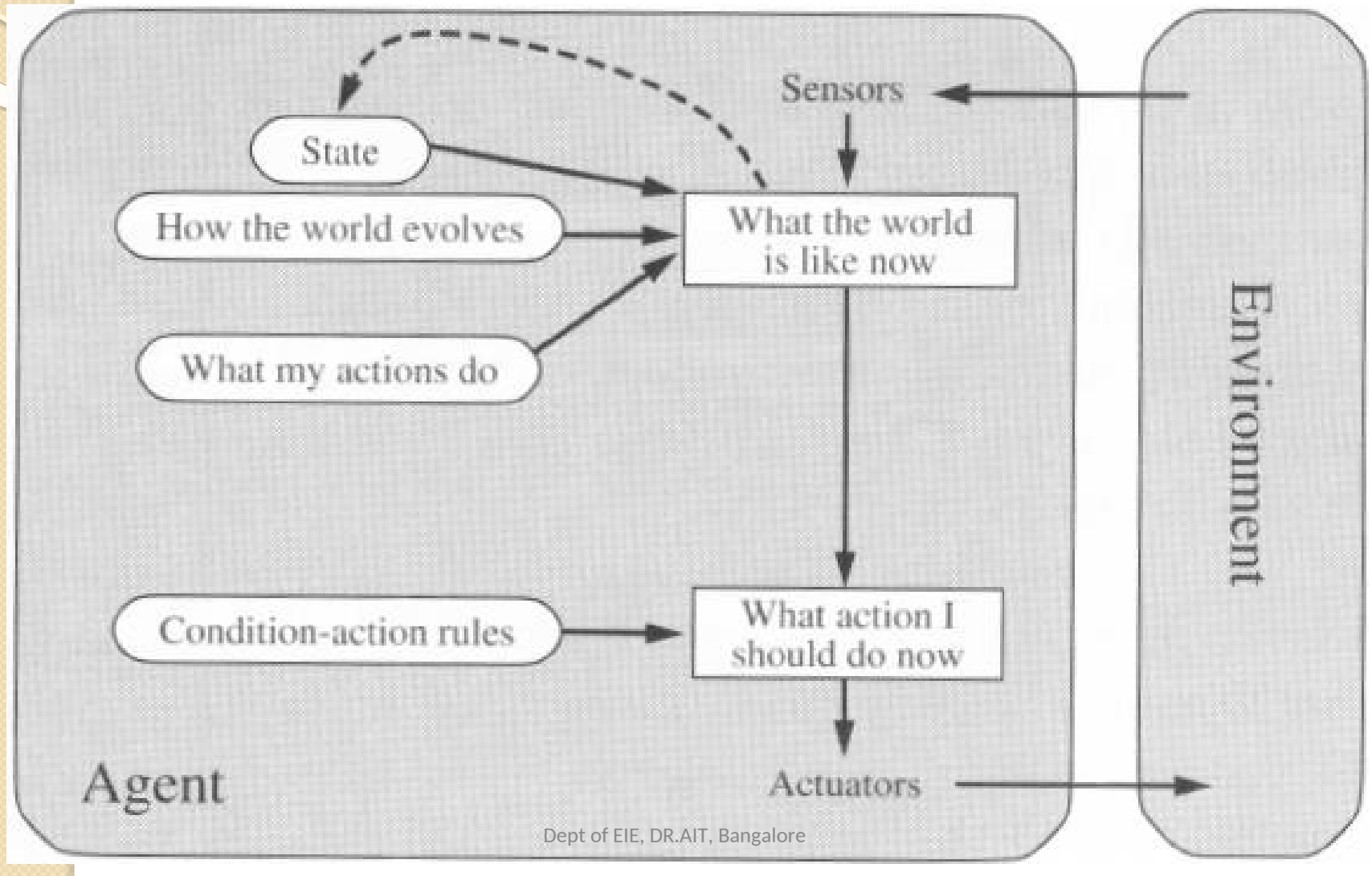
Types of agent with pseudo code

- **Simple reflex agent**
- **Model-based reflex agent**
- **Goal-based agent**
- **Utility-based agent**
- **Learning agent**

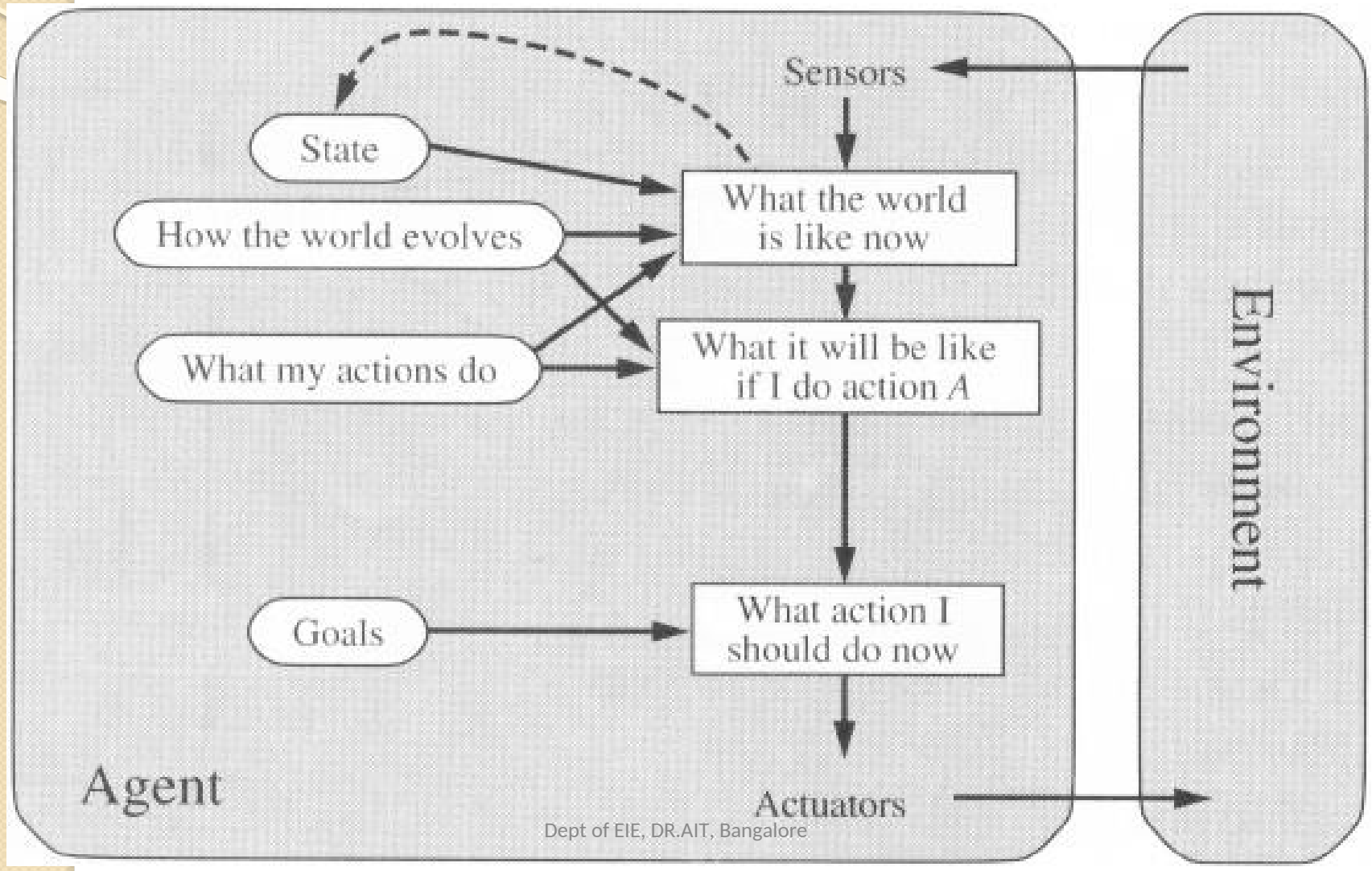
Simple Reflex Agent



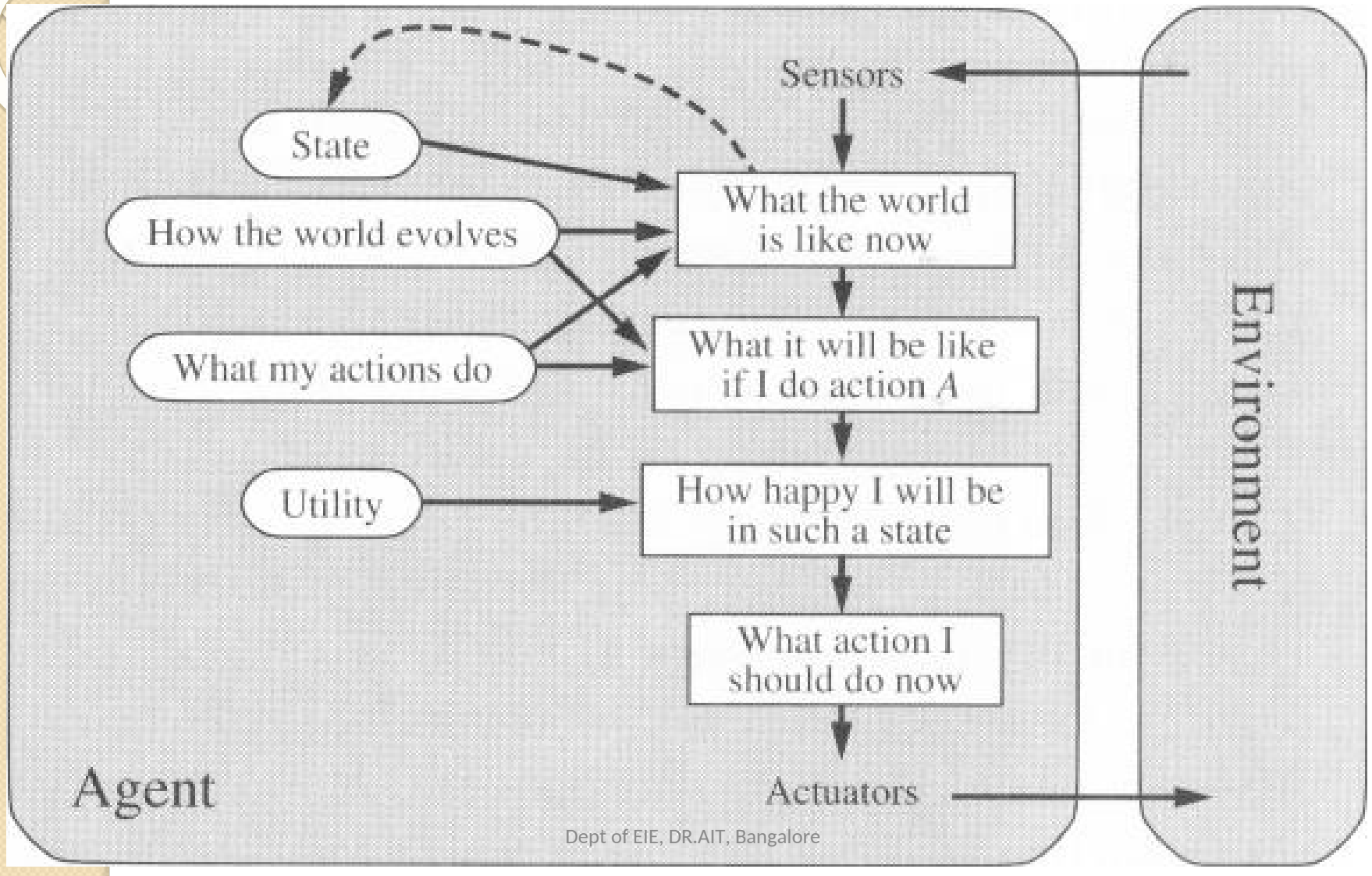
Model-based Reflex Agent



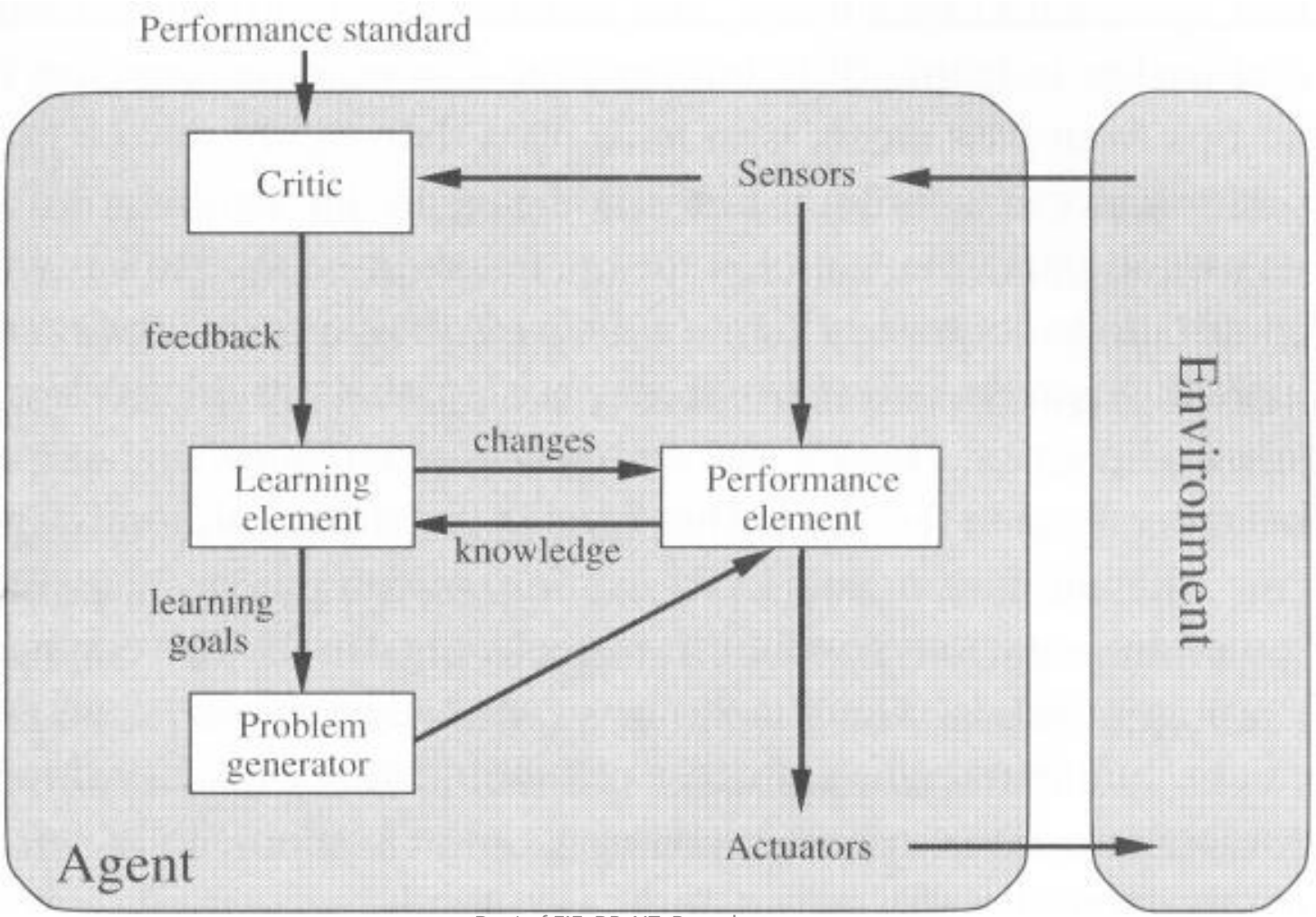
Goal-based Agent



Utility-based agents



Learning Agents



Thank you