BCSE306L	Artificial Intelligence		L	T	P	C
			3	0	0	3
Pre-requisite	**NIL**	**Syllabus version**				
		1.0				

Course Objectives

1. To impart artificial intelligence principles, techniques and its history.
2. To assess the applicability, strengths, and weaknesses of the basic knowledge representation, problem solving, and learning methods in solving engineering problems
3. To develop intelligent systems by assembling solutions to concrete computational problems

Course Outcomes

On completion of this course, student should be able to:

1. Evaluate Artificial Intelligence (AI) methods and describe their foundations.
2. Apply basic principles of AI in solutions that require problem-solving, inference, perception, knowledge representation and learning.
3. Demonstrate knowledge of reasoning, uncertainty, and knowledge representation for solving real-world problems
4. Analyse and illustrate how search algorithms play a vital role in problem-solving

Course Contents - CAT I Syllabus

Module:1	Introduction	6 hours
Introduction- Evolution of AI, State of Art -Different Types of Artificial Intelligence-Applications of AI-Subfields of AI-Intelligent Agents- Structure of Intelligent Agents-Environments		
Module:2	**Problem Solving based on Searching**	**6 hours**
Introduction to Problem Solving by searching Methods-State Space search, Uninformed Search Methods – Uniform Cost Search, Breadth First Search- Depth First Search-Depth-limited search, Iterative deepening depth-first, Informed Search Methods- Best First Search, A* Search		
Module 3	**Local Search and Adversarial Search**	**5 hours**
Local Search algorithms – Hill-climbing search, Simulated annealing, Genetic Algorithm, Adversarial Search: Game Trees and Minimax Evaluation, Elementary two-players games: tic-tac-toe, Minimax with Alpha-Beta Pruning.		

Module 1 -3 upto Genetic Algorithm

Course Contents - CAT II Syllabus

Module 3	**Local Search and Adversarial Search**	**5 hours**
Local Search algorithms – Hill-climbing search, Simulated annealing, Genetic Algorithm, Adversarial Search: Game Trees and Minimax Evaluation, Elementary two-players games: tic-tac-toe, Minimax with Alpha-Beta Pruning.		
Module:4	**Logic and Reasoning**	**8 hours**
Introduction to Logic and Reasoning -Propositional Logic-First Order Logic-Inference in First Order Logic- Unification, Forward Chaining, Backward Chaining, Resolution.		
Module:5	**Uncertain Knowledge and Reasoning**	**5 hours**
Quantifying Uncertainty- Bayes Rule -Bayesian Belief Network- Approximate Inference in Bayesian networks		

Module 3 (Starting from Adversarial search) - 5

Course Contents - FAT Syllabus

Module:6	**Planning**	**7 hours**
Classical planning, Planning as State-space search, Forward search, backward search, Planning graphs, Hierarchical Planning, Planning and acting in Nondeterministic domains – Sensor-less Planning, Multiagent planning		
Module:7	**Communicating, Perceiving and Acting**	**6 hours**
Communication-Fundamentals of Language -Probabilistic Language Processing -Information Retrieval- Information Extraction-Perception-Image Formation- Object Recognition.		
Module:8	**Contemporary Issues**	**2 hours**

Module 6 - 8

Text/Reference Books

Text Book	
1.	Russell, S. and Norvig, P. 2015. Artificial Intelligence - A Modern Approach, 3rd Edition, Prentice Hall.

Reference Books	
1.	K. R. Chowdhary, Fundamentals of Artificial Intelligence, Springer, 2020.
2	Alpaydin, E. 2010. Introduction to Machine Learning. 2nd Edition, MIT Press.

Course Evaluation Plan - Theory

- Quiz -1 (10 Marks)
- Quiz-2 (10 Marks)
- Assignment (10 Marks)
- CAT 1 – (15 Marks)
- CAT2 – (15 Marks)
- Final Assessment Test - (40 Marks)
- No Additional Learning

Course Assessment Configuration - View

Course Code	Course Title	Course Type	ClassNbr	Slot	Allotted Program	Course Mode	Course System
BCSE306L	Artificial Intelligence	TH	VL2024250105795	E2/TE2	ALL	CBL	CBCS

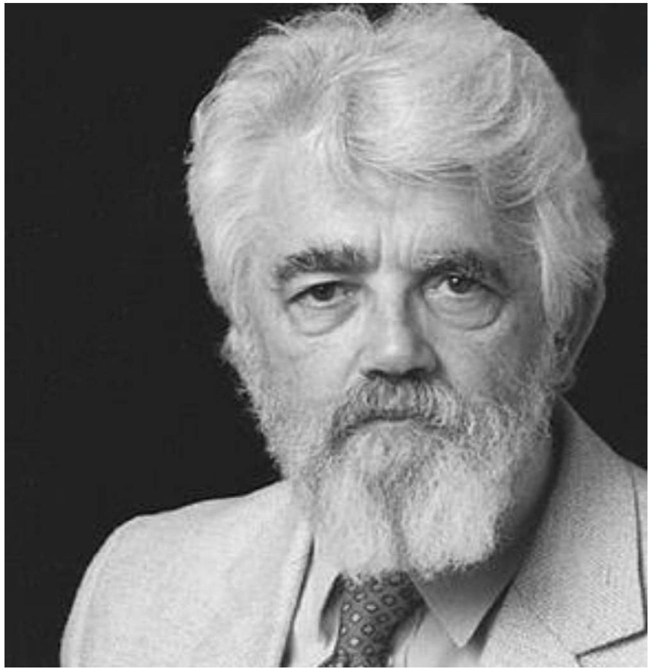
Kindly Note:
This includes participation in technical events like Hack-a-Thon, completion of on-line courses, publication of articles in scientific journals, and any other related activity.

As.No.	Assessment Title	Question Upload	Answer Upload	Due Date	Activity Date	Max. Mark	Weightage %
1	Digital Assignment - Digital Assignment - I	Mandatory	Mandatory	15-Nov-2024	-	10	10
2	QUIZ - Quiz - I	Not Applicable	Not Applicable	-	-	10	10
3	QUIZ - Quiz - II	Not Applicable	Not Applicable	-	-	10	10
4	CAT - Continuous Assessment Test - I	Not Applicable	Not Applicable	-	-	50	15
5	CAT - Continuous Assessment Test - II	Not Applicable	Not Applicable	-	-	50	15
6	FAT - Final Assessment Test	Currently info not available	Currently info not available	-	-	100	40
						Total Weightage Mark	100

INTRODUCTION
TO
ARTIFICIAL INTELLIGENCE

ARTIFICIAL INTELLIGENCE

- AI – John Mc Carthy 1956.
- Make Computers to do things.
- Developing Intelligent Computer Programs.
- Classes of AI
  - Theorem proving – requires a logic (syntax), a set of axioms and inference rules.
  - Perception
  - Robotics
  - Natural Language Processing
  - Common Sense Reasoning
  - Game Playing



What is AI?

Four Categories of Systemic Definitions

1. Think like humans 2. Act like humans

3. Think *rationally* 4. Act *rationally*

Thinking Like Humans

***Machines with minds* (Haugeland, 1985)**

Automation of "decision making, problem solving, learning…" (Bellman, 1978)

Acting Like Humans

***Functions* that *require intelligence when performed by people* (Kurzweil, 1990)**

Making computers do things *people currently do better* (Rich & Knight, 1991)

Thinking Rationally

Computational models of mental faculties (Charniak & McDermott, 1985)

Computations that make it possible to *perceive*, *reason*, and *act* (Winston, 1992)

Acting Rationally

Explaining, emulating int. behavior via computation (Schalkoff, 1990)

Branch of CS: automating intelligent behavior (Luger, 2005)

Various Definitions of AI

- AI is the automation of activities that we associate with human thinking, activities such as decision making, problem solving, learning… …

  -Bellman, 1978

- AI is concerned with designing intelligent computer systems which exhibit the characteristics we associate with intelligence in human behaviour

  -Barr and Feignenbaum, 1981

- AI is the exciting new effort to make computers think… … machines with minds, in the full and literal sense

  - Haugeland, 1985

Various Definitions of AI Cont…

- AI is the study of mental faculties through the use of computational models

  -Charniak McDermott, 1985

- AI is the art of creating machines that perform functions that require intelligence when performed by people

  -Kurzwell, 1990

- AI is the study of how to make computers do things at which, ay the moment, people are better

  -Rich and Knight, 1991

Various Definitions of AI Cont…

- AI is the study of the combinations that make it possible to perceive, reason and act

  -Winston, 1992

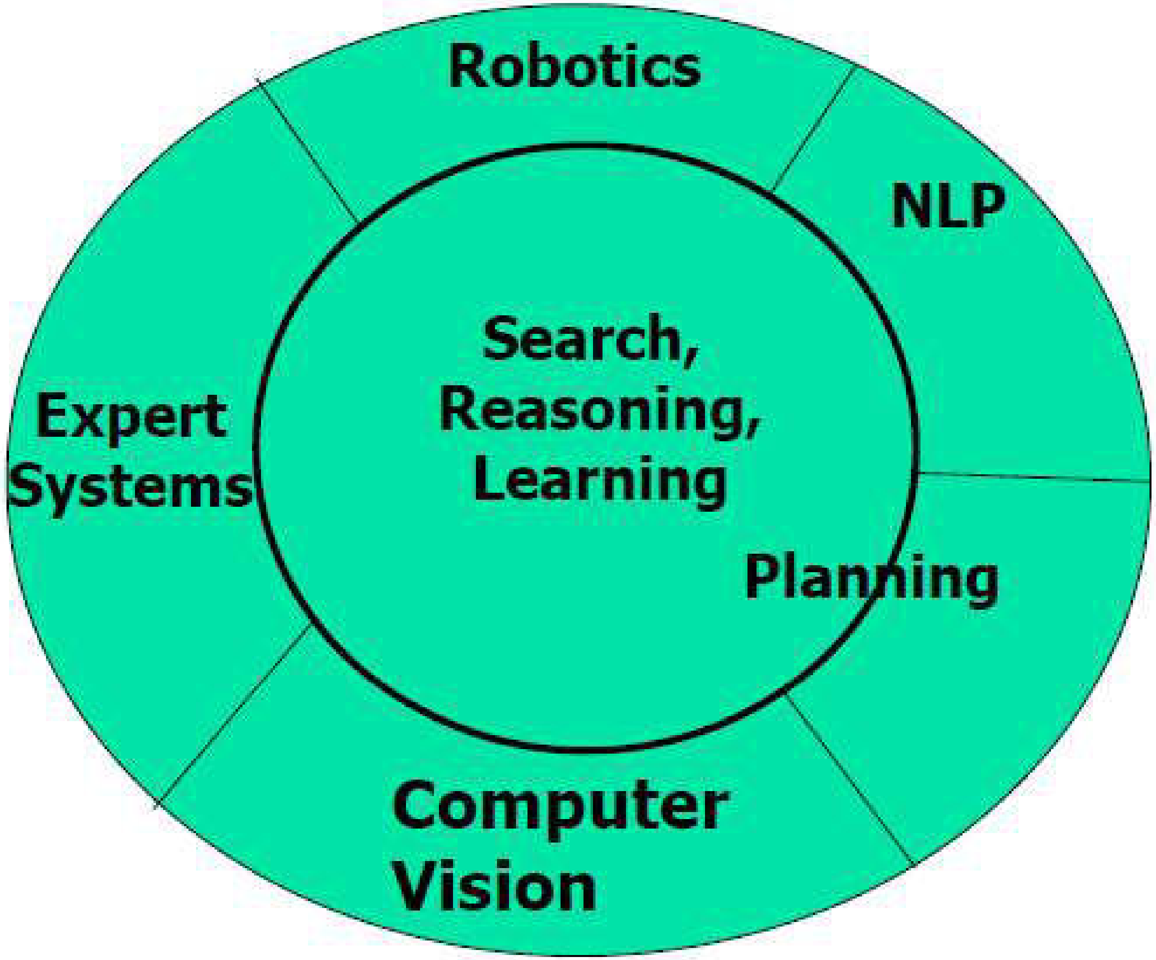
- Computational intelligence is the study of the design of intelligent agents

  -Poole et al., 1998

- AI… is concerned with intelligent behavior in artifacts

  -Nilsson, 1998

- Artificial Intelligence is not an easy science to describe
- It’s a fuzzy borders with the following disciplines:
  - **Mathematics**
  - **Computer science**
  - **Philosophy**
  - **Psychology**
  - **Statistics**
  - **Physics**
  - **Biology and other disciplines**


Robotics
NLP
Search,
Reasoning,
Learning
Expert
Systems
Planning
Computer
Vision

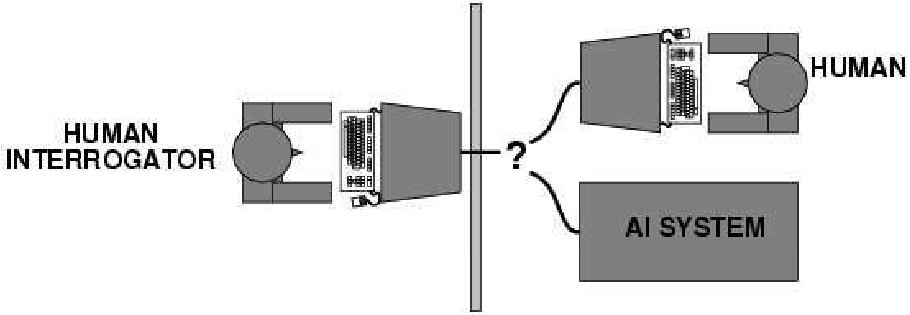
What is AI?

Views of AI fall into four categories:

Thinking humanly	Thinking rationally
Acting humanly	Acting rationally

Acting humanly: Turing Test

- Turing (1950) "Computing machinery and intelligence":
- "Can machines think?" → "Can machines behave intelligently?"
- Operational test for intelligent behavior: the Imitation Game



- Predicted that by 2000, a machine might have a 30% chance of fooling a lay person for 5 minutes
- Anticipated all major arguments against AI in following 50 years
- Suggested major components of AI: knowledge, reasoning, language understanding, learning

Acting humanly: Turing Test cont….

- The computer would need to possess the following capabilities:
    - **Natural Language Processing (NLP)**
    - **Knowledge representation (KR)**
    - **Automated reasoning**
    - **Machine Learning**
    - **Computer vision**
    - **Robotics**

Thinking humanly: cognitive modeling

- The goal is not just to produce human-like behavior but to produce a sequence of steps of the reasoning process, similar to the steps followed by a human in solving the same task.
- 1960s "cognitive revolution": information-processing psychology
- Requires scientific theories of internal activities of the brain
- -- How to validate? Requires

  1) Predicting and testing behavior of human subjects (top-down) or

  2) Direct identification from neurological data (bottom-up)
- Both approaches (roughly, Cognitive Science and Cognitive Neuroscience) are now distinct from AI

Thinking rationally: "laws of thought"

- The goal is to formalize the reasoning process as a system of logical rules and procedures of inference.
- Aristotle: what are correct arguments/thought processes?
- Several Greek schools developed various forms of *logic*: *notation* and *rules of derivation* for thoughts; may or may not have proceeded to the idea of mechanization
- Direct line through mathematics and philosophy to modern AI
- Problems:
  1. Not all intelligent behavior is mediated by logical deliberation
  2. What is the purpose of thinking? What thoughts should I have?

Ex: Thinking Rationally

All computers use energy. Using energy always generates heat. Therefore, all computers generate heat.

Acting rationally: rational agent

- Rational behavior: doing the right thing
- The right thing: that which is expected to maximize goal achievement, given the available information
- Doesn't necessarily involve thinking – e.g., blinking reflex – but thinking should be in the service of rational action

Rational agents

- An agent is an entity that perceives and acts
- This course is about designing rational agents
- Abstractly, an agent is a function from percept histories to actions:

$$[f: \mathcal{P}^\star \rightarrow \mathcal{A}]$$

- For any given class of environments and tasks, we seek the agent (or class of agents) with the best performance
- Caveat: computational limitations make perfect rationality unachievable

→ design best program for given machine resources

Perception in AI implies **the ability of machines to use input data from sensors.**

AI prehistory

• Philosophy	Logic, methods of reasoning, mind as physical system foundations of learning, language, rationality
• Mathematics	Formal representation and proof algorithms, computation, (un)decidability, (in)tractability, probability
• Economics	utility, decision theory
• Neuroscience	physical substrate for mental activity
• Psychology	phenomena of perception and motor control, experimental techniques
• Computer engineering	building fast computers
• Control theory	design systems that maximize an objective function over time
• Linguistics	knowledge representation, grammar

Generic Techniques Developed

- Forward/backward chaining (reasoning)
- Resolution theorem proving (reasoning)
- Proof planning (reasoning)
- Constraint satisfaction (reasoning)
- Davis-Putnam method (reasoning)
- Minimax search (games)
- Alpha-Beta pruning (games)
- Case-based reasoning (expert systems)
- Knowledge elicitation (expert systems)
- Neural networks (learning)
- Bayesian methods (learning)
- Explanation based (learning)
- Inductive logic programming (learning)
- Reinforcement (learning)
- Genetic algorithms (learning)
- Genetic programming (learning)
- Strips (planning)
- N-grams (NLP)
- Parsing (NLP)
- Behaviour based (robotics)
- Cell decomposition (robotics)

Representations/Languages Used

- First order logic
- Higher order logic
- Logic programs
- Frames
- Production Rules
- Semantic Networks
- Fuzzy logic
- Bayes nets
- Hidden Markov models
- Neural networks
- Strips

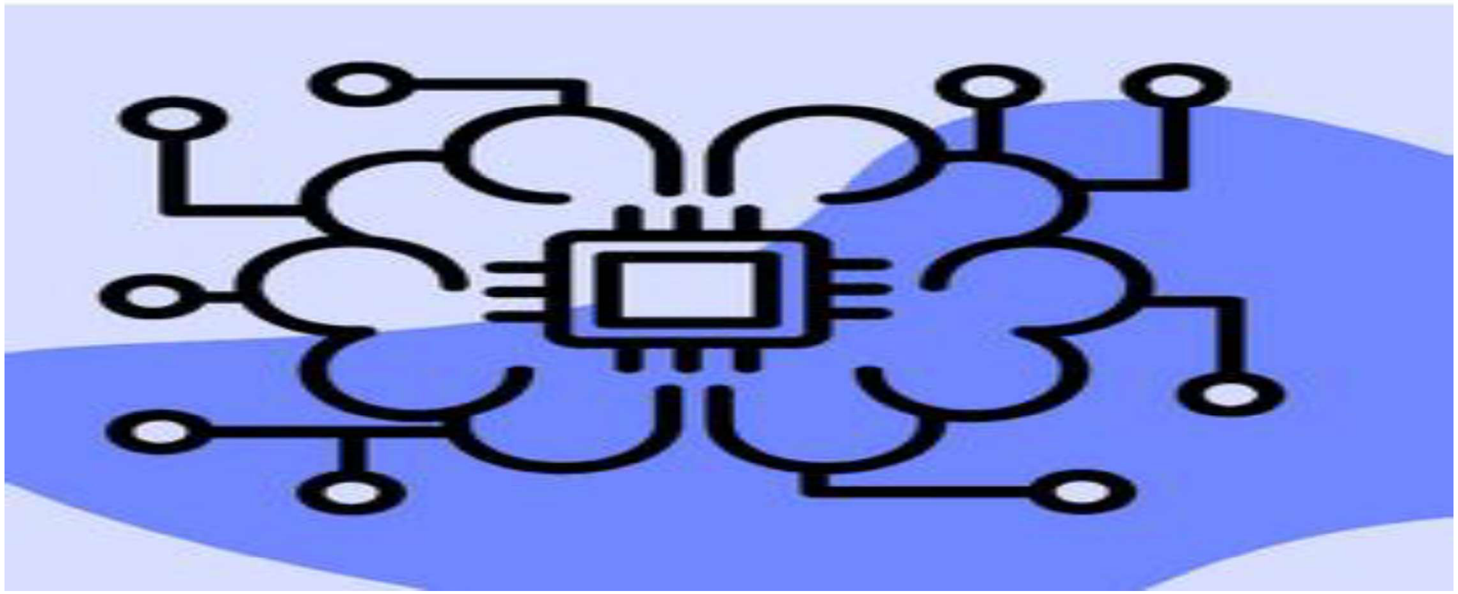
- Some standard AI programming languages have been developed in order to build intelligent programs efficiently and robustly. These include:
  - Prolog
  - Lisp (LISt Processing)
  - ML (Metalanguage)
- Other languages are also used extensively to build AI programs, including:
  - Perl
  - C++
  - Java
  - C

Application Areas

- Agriculture
- Architecture
- Art
- Astronomy
- Bioinformatics
- Email classification
- Engineering
- Finance
- Fraud detection
- Information retrieval
- Law
- Mathematics
- Military
- Music
- Scientific discovery
- Story writing
- Telecommunications
- Telephone services
- Transportation
- Tutoring systems
- Video games
- Web search engines

ARTIFICIAL INTELLIGENCE

INTRODUCTION



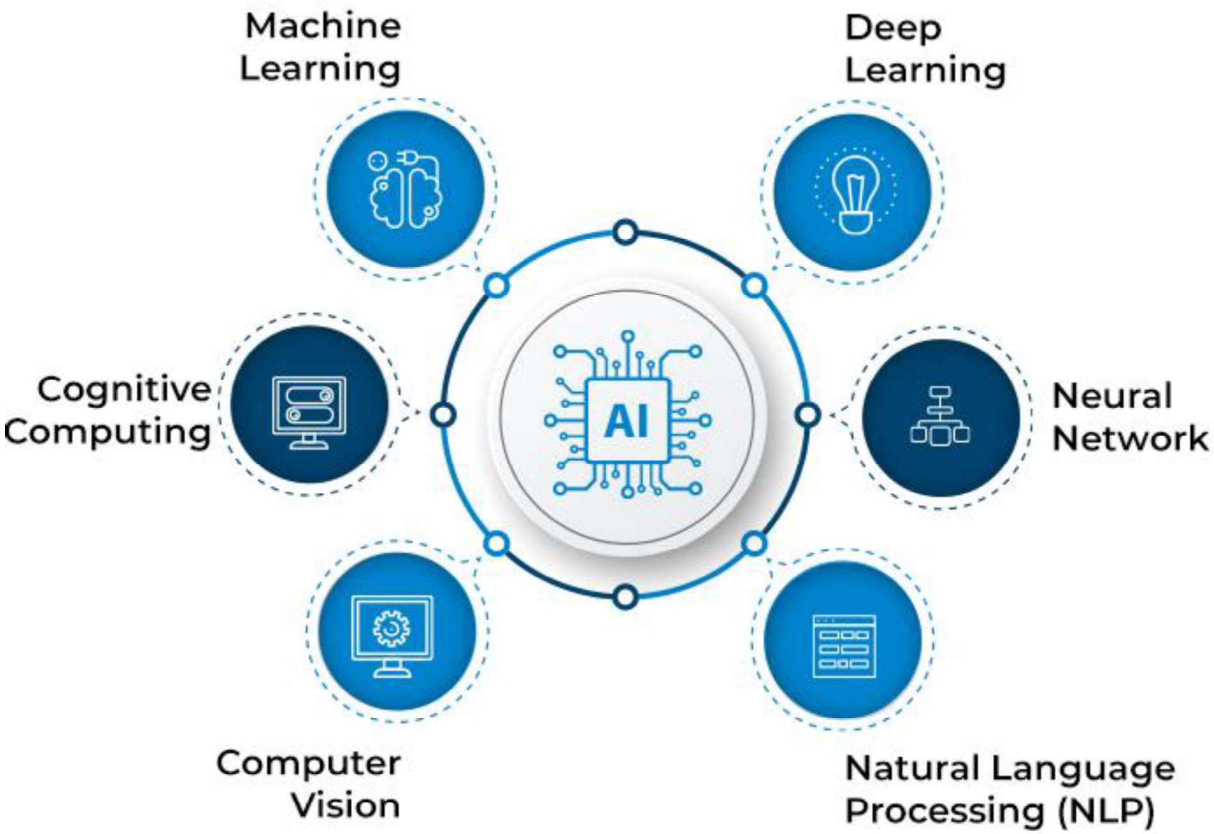
What is Artificial Intelligence?

- “Artificial intelligence is a computerized system that exhibits behavior that is commonly thought of as requiring intelligence.”
- “Artificial Intelligence is the science of making machines do things that would require intelligence if done by man.”
- The founding father of AI, Alan Turing, defines this discipline as:
  - “AI is the science and engineering of making intelligent machines, especially intelligent computer programs.”

Thinking humanly “’ Machines that think like humans “	Thinking rationally “ Machines that think Rationally”
Acting humanly “Machines that behave like humans”	Acting rationally “Machines that behave Rationally”

AI fall into four categories

Key components of AI



Key components of AI

- Machine learning: Machine learning is an AI application that automatically learns and improves from previous sets of experiences without the requirement for explicit programming.
- Deep learning: Deep learning is a subset of ML that learns by processing data with the help of artificial neural networks.
- Neural network: Neural networks are computer systems that are loosely modelled on neural connections in the human brain and enable deep learning.
- Cognitive computing: Cognitive computing aims to recreate the human thought process in a computer model. It seeks to imitate and improve the interaction between humans and machines by understanding human language and the meaning of images.

Key components of AI

- Natural language processing (NLP): NLP is a tool that allows computers to comprehend, recognize, interpret, and produce human language and speech.
- Computer vision: Computer vision employs deep learning and pattern identification to interpret image content (graphs, tables, PDF pictures, and videos)

Goals of Artificial Intelligence

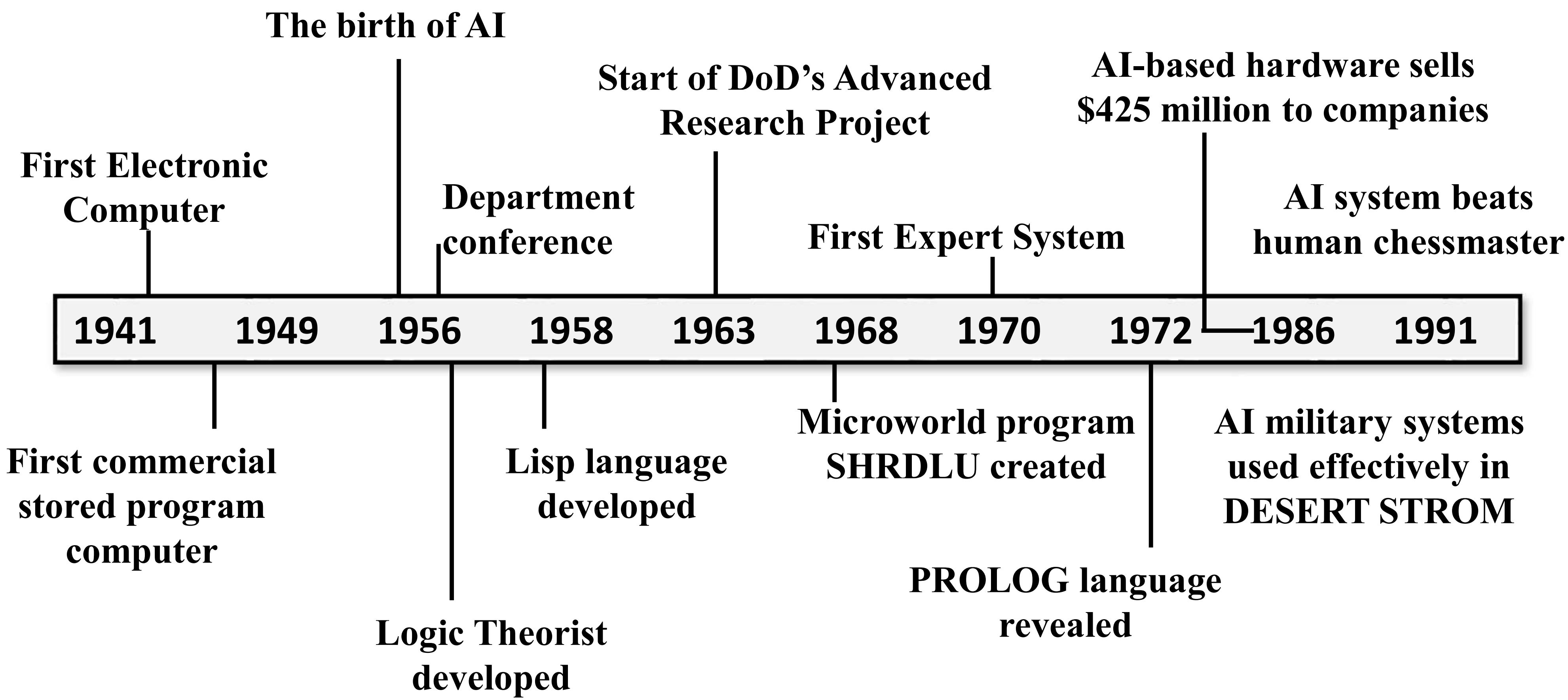
- Generalized learning
  - Generalization allows humans to recognize the similarities in knowledge acquired in one circumstance, allowing for the transfer of knowledge onto new situations.
- Reasoning
  - To derive new information from existing information using logical rules and principles.
- Problem-solving
  - Problem-solving is a process of generating solutions from observed data.

Applications of Artificial Intelligence

- Personalized Online Shopping
- Smart Cars
- Marketing
- Surveillance
- Agriculture
- Customer Service
- Video Games
- Healthcare
- Banks
- Smart Homes
- Virtual Assistance
- Space Exploration
- Chatbots

HISTORY OF AI

Timeline of major AI events



INTRODUCTION

- The origin of AI can be seen in Turing's work in his paper on Computing Machinery and Intelligence published in 1950.

- McCarthy's invention of the LISP programming language in 1950s

- Newell and Simon's GPS concept (showed the general search knowledge could used to solve a range of problems (Missionaries and cannibals problems)
- Missionaries and cannibals problems - **three missionaries and three cannibals must cross a**

Concepts of AI (1943-1955)

- Warren McCulloch and Walter Pitts proposed a model of artificial neurons (1943)

- Alan Turing evolving and exhibiting complete vision of AI in his research article in 1950

- Marvin Minsky and Dean Edmonds built the first neural network computer is known as SNARC (1951)

BIRTH OF AI (1956)

- John McCarthy introduce the word AI

Adolescence of AI (1952-1969)

- McCarthy published a paper entitled “programs with common sense” in the year of 1958
- In 1959, Herbert Gelernter built the Geometry Theorem Prover that could prove theorems which were difficult & tedious for normal mathematics

Youthfulness of AI (1969-1979)

- System which can able to solve simple problems with the help of AI concepts is called **weak AI method**
- The system can solve all the problems (Difficult & complex) of a particular area using a powerful knowledge base in a limited domain is known as **strong AI method**

Youthfulness of AI (1969-1979)

- DENDRAL was an influential project in artificial intelligence (AI) of the 1960s, and the computer software expert system that it produced. Its primary aim was **to study hypothesis formation and discovery in science**.
- E.g: ***DENDRAL*** developed at Stanford (1969) by Ed Feigenbaum, Bruce Buchanan & Joshua Lederberg (molecular structure based estimation)
- Newell and Simon to devise their famous physical symbol system hypothesis in 1976

Youthfulness of AI (1969-1979)

- In 1970s expert systems were developed which embodied as a set of rules the knowledge of an expert (MYCIN Medical expert system)
- At the same time systems were developed to understand language, of which the most famous was Winograd's SHRDLU system
- In1980s saw the development of Neural Networks as a method of learning examples.

- ***MYCIN*** developed in medicine field by Feigenbaum, Buchanan and Edward Shortliffe
- Natural Language understanding based research was started in this period
- ***SHRDLU*** system to understand nature language
- Most important out come of this era is knowledge was utmost importance and it would be wiser to go for intelligent systems
- Programs were developed to solve real world problems and this increased the demand for workable knowledge acquisition and knowledge representation scheme (**PROLOG**)

Maturity and Commercialization of AI (1980-present)

- In 1988 AI department of Digital Equipment Corporation was able to develop about 40 expert systems
- Japanese announced “Fifth Generation” project in 1981 to built expert systems running prolog
- Same period U.S formed Microelectronic and Computer Technology Corporation(MCC) established AI based chips (IC’s)
- In late 1970s strong AI methods overtaken the weak AI problems
- In mid 1980s neural-net models of memory revival happened

- In 1988 Judea Pearl’s published a paper in “Probabilistic Reasoning in Intelligent Systems”
- It deals with uncertainty
- The Bayesian network formalism was invented to allow efficient representation of uncertainty knowledge which now dominates the AI research on uncertain reasoning and expert systems
- Today, the field of AI has become so big and diversified that has been divided into different specialization and the areas such as vision and robotics etc.

Abridged history of AI

- 1943 McCulloch & Pitts: Boolean circuit model of brain
- 1950 Turing's "Computing Machinery and Intelligence"
- 1956 Dartmouth meeting: "Artificial Intelligence" adopted
- 1952—69 Look, Ma, no hands!
- 1950s Early AI programs, including Samuel's checkers program, Newell & Simon's Logic Theorist, Gelernter's Geometry Engine
- 1965 Robinson's complete algorithm for logical reasoning
- 1966—73 AI discovers computational complexity
  Neural network research almost disappears
- 1969—79 Early development of knowledge-based systems
- 1980-- AI becomes an industry
- 1986-- Neural networks return to popularity
- 1987-- AI becomes a science
- 1995-- The emergence of intelligent agents

Knowledge Based Systems

- Domain Expert
- Knowledge Engineering
- Knowledge Engineer
- Knowledge = Facts + Rules
- Knowledge Base
- Knowledge-Based Systems
- Database Systems Vs Knowledge Base Systems.

Knowledge Base Vs Database

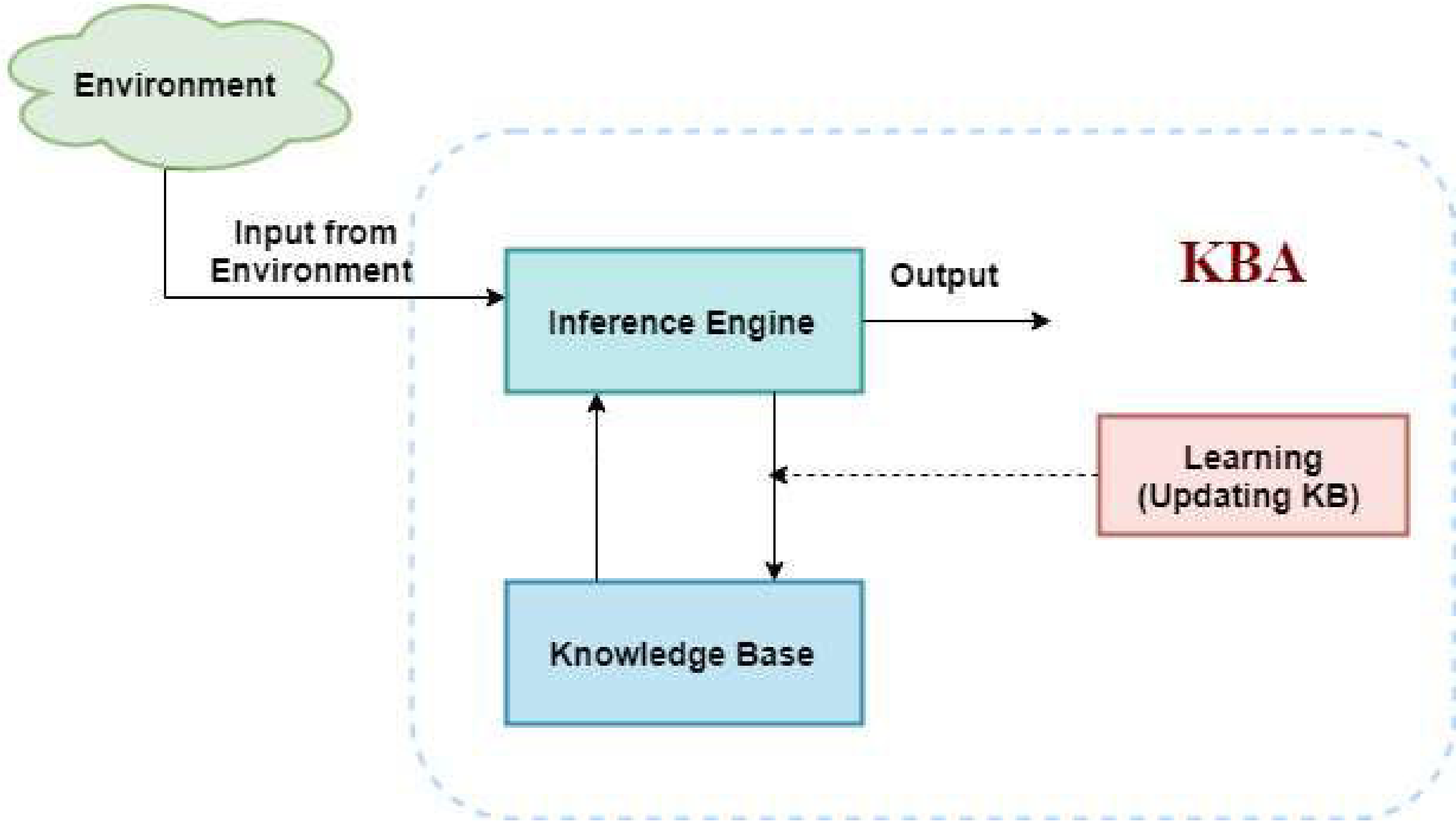
More Rules	Less Rules
Facts Less	More Facts
Explicit Rules and Facts	Explicit Facts and Implicit Rules
Experts update	Clerks update
Main Memory Based	Disk Based


Expert Systems –
Apply expert
knowledge to
difficult,
Real world problems.
KBS – Make domain knowledge
explicit
AI Programs - Exhibit Intelligent Behavior by skillful
application of heuristics.

Expert Systems

- Knowledge Base = Facts + Rules.
- Inference Engine (Interpreter + Scheduler)
- Deduction
- Reasoning
  - Temporal Reasoning
  - Reasoning by Analogy
  - Chaining (Forward & Backward)
- Representation and Manipulation.

ARCHITECTURE OF KNOWLEDGE-BASED AGENT



LOGIC

- First Order Logic
  - Predicate Logic
  - Propositional Logic
- Higher Order Logics
  - Situational Logic
  - Fuzzy Logic
  - Temporal Logic
  - Modal Logic
  - Epistemic Logic

Knowledge Representation Techniques

- Tables
- Rules
- Semantic Networks
- Frames
- Conceptual Dependency
- Scripts

SEARCHING

- Depth First Search
- Breadth First Search
- Best First Search
- Hill Climbing
- A^*
- AO^*
- Minimax

CONCLUSION

- Decision Making
- Complete and Incomplete Information
- Intelligent Behavior
- Knowledge Representation
- Logics
- Searching Techniques

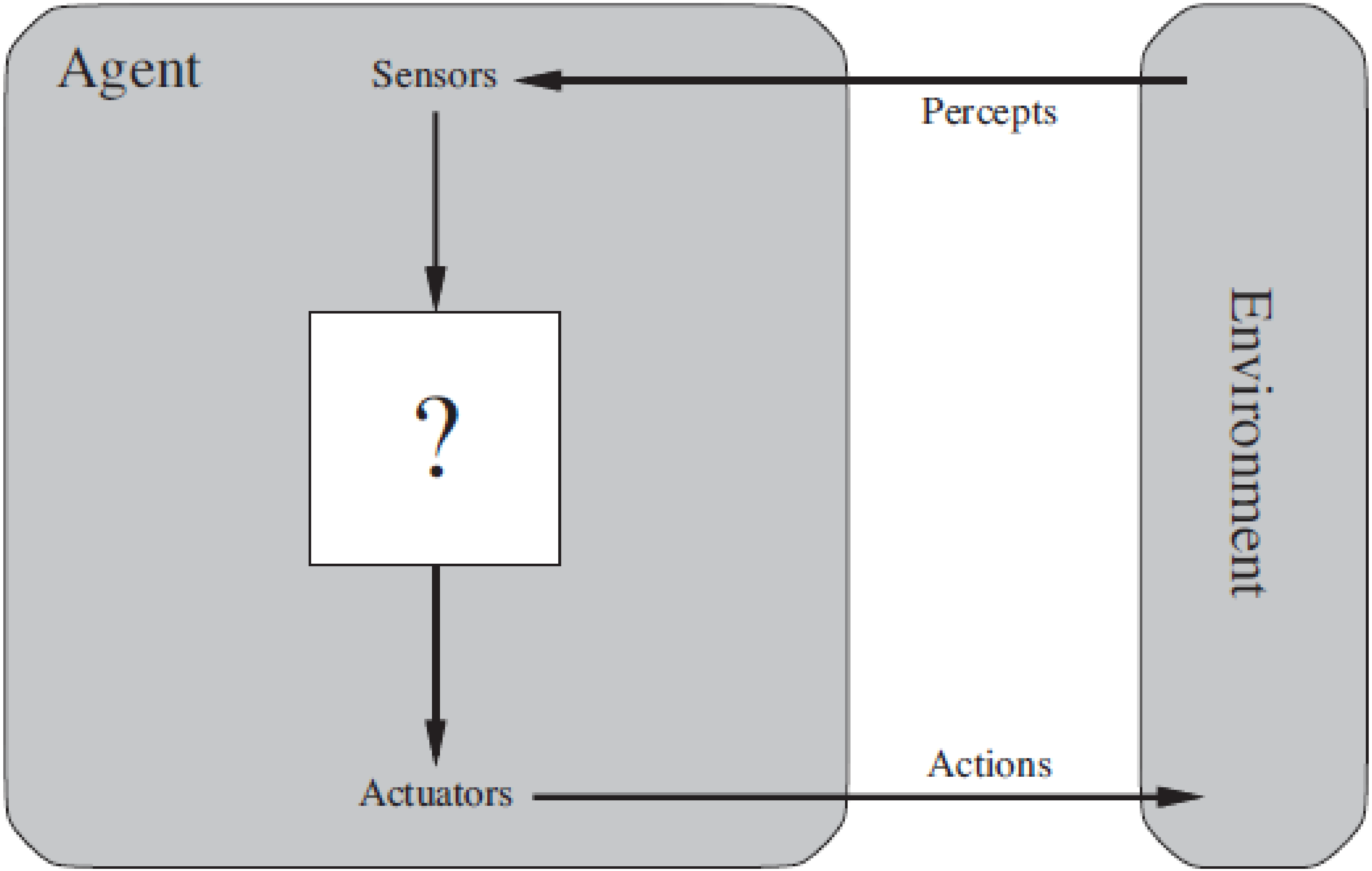
INTELLIGENT AGENTS

OUTLINE

- Introduction to Intelligent Agent (IA)
- Definition of Agent
- Rationality
- PEAS (Performance measure, Environment, Actuators & Sensors)
- Environment types
- Agent types

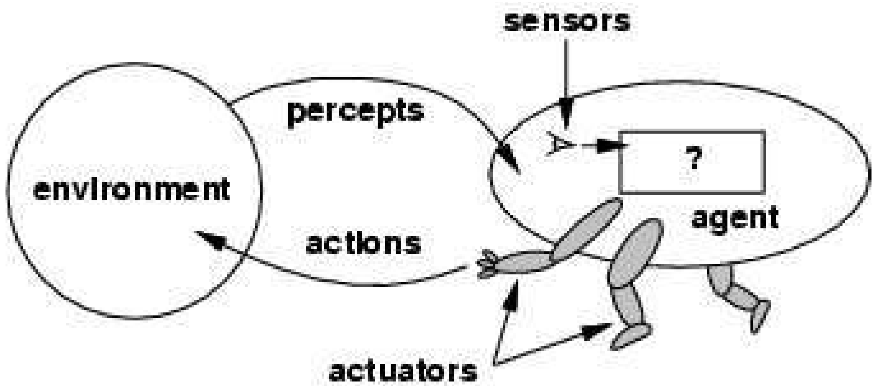
AGENTS

- An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators.
- **Human agent**: eyes, ears, and other organs for sensors and other body parts such as hands, legs, mouth, for actuators.
- **Robotic agent**: cameras and infrared range finders for sensors; various motors for actuators



Agents interact with environments through sensors and actuators.

AGENTS AND ENVIRONMENTS

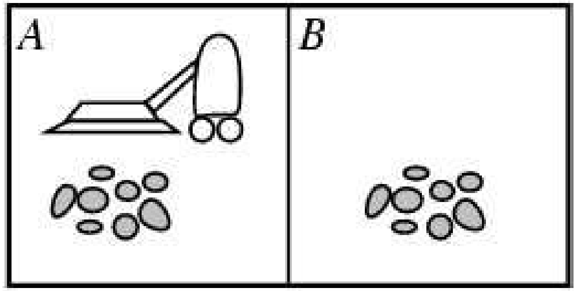


- The agent function maps from percept histories to actions:

$$[f: P^* \rightarrow A]$$

- The agent program runs on the physical architecture to produce *f*
  - » agent = architecture + program

VACUUM-CLEANER WORLD



- **Percepts**: location and contents, e.g., [A,Dirty]
- **Actions**: *Left*, *Right*, *Suck*, *NoOp*

A VACUUM-CLEANER AGENT

- input{tables/vacuum-agent-function-table}

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*
⋮	⋮

RATIONAL AGENTS

- An agent should strive to "do the right thing", based on what it can perceive and the actions it can perform. The right action is the one that will cause the agent to be most successful.

- Performance measure: An objective criterion for the success of an agent's behavior.

- E.g., performance measure of a vacuum-cleaner agent could be ***amount of dirt cleaned up***, ***amount of time taken***, ***amount of electricity consumed***, ***amount of noise generated***, etc.

RATIONALITY

- **Four factors affecting rational** at any given time is:
  - **The performance measure that defines the criterion of success**
  - **The agent's prior knowledge of the environment**
  - **The actions that the agent can perform**
  - **The agent's percept sequence to date**

RATIONAL AGENTS

- 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.

RATIONAL AGENTS

- Rationality is distinct from omniscience (all-knowing with infinite knowledge)
- Agents can perform actions in order to modify future percepts so as to obtain useful information (information gathering, exploration)
- An agent is autonomous if its behavior is determined by its own experience (with ability to learn and adapt)

NATURE OF ENVIRONMENT

- Environment:
  - It is the task environment of the agent.
  - Must decide and think about “Task environment”
- Task environments are essentially the “problems” to which rational agents are the “solution”
- **SPECIFYING THE TASK ENVIRONMENT**
  - It involves grouping the following measuring factors together under the heading of the “Task environment”.
  - PEAS is an AI agent representation system that focuses on evaluating the performance of the environment, sensors, and actuators.
  - PEAS (Performance, Environment, Actuator, Sensors)
  - PEAS define the task environment for an intelligent agent

PEAS

- PEAS: **P**erformance measure, **E**nvironment, **A**ctuators, **S**ensors
- Must first specify the setting for intelligent agent design
- Consider, **e.g**., **the task of designing an automated taxi driver**:

  – Performance measure
  – Environment
  – Actuators
  – Sensors`

PEAS

- Must first specify the setting for intelligent agent design
- Consider, e.g., 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

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

PEAS description of the task environment for an automated Taxi

PEAS

- **Agent**: Medical diagnosis system
- **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

- **Agent**: 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

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

Agent Type	Performance Measure	Environment	Actuators	Sensors
Medical diagnosis system	Healthy patient, reduced costs	Patient, hospital, staff	Display of questions, tests, diagnoses, treatments, referrals	Keyboard entry of symptoms, findings, patient's answers
Satellite image analysis system	Correct image categorization	Downlink from orbiting satellite	Display of scene categorization	Color pixel arrays
Part-picking robot	Percentage of parts in correct bins	Conveyor belt with parts; bins	Jointed arm and hand	Camera, joint angle sensors
Refinery controller	Purity, yield, safety	Refinery, operators	Valves, pumps, beaters, displays	Temperature, pressure, chemical sensors
Interactive English tutor	Student's score on test	Set of students, testing agency	Display of exercises, suggestions, corrections	Keyboard entry

Examples of agent types and their PEAS descriptions

Environment types (or) Properties of Task environment

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

- Ex:
  - chess – the board is fully observable, as are opponent’s moves.
  - Driving – what is around the next bend is not observable.

Environment types (or) Properties of Task environment

- **Deterministic** (vs. stochastic) / environment : If the next state of the environment is completely determined by the current state and the action executed by the agent, then we say the environment is deterministic; otherwise, it is stochastic.
- Stochastic:
  - the next state has some uncertainty associated with it .
  - Uncertainty could come from randomness,
  - lack of a good environment model, or lack of complete sensor coverage.

Environment types (or) Properties of Task environment

Ex:

- Stochastic environment: physical world, Ludo (any game that involve dice)
  - **Self-Driving Cars-** the actions of a self-driving car are not unique, it varies time to time.

- Deterministic environment: Robot on Mars, Tic tac toe.
  - **Chess –** there would be only a few possible moves for a coin at the current state and these moves can be determined.

Environment types (or) Properties of Task environment

- 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.
- Episodic environment, an agent's current action will not affect a future action,
- Sequential if current decisions affect future decisions, or rely on previous ones
- Examples of episodic are expert advice systems – an episode is a single question and answer
- Ex: Episodic - Part picking robot.
- Sequential - Chess

Environment types

- Static (vs. dynamic): The environment is unchanged while an agent is deliberating.
- Dynamic: the environment may change over time.
- Static if nothing (other than the agent) in the the environment changes
- E.g. – Playing football, other players make it dynamic.
- Expert systems are usually static (unless knowledge changes)
- Ex: Static - Crossword Puzzle.

Environment types

- Discrete (Vs. continuous): A limited number of distinct, clearly defined percepts and actions.
- Continuous : The environment in which the actions performed can not be numbered ie., is not discrete, is said to be continuous.
- **Discrete vs. continuous:**
  - Discrete = time moves in fixed steps, usually with one measurement per step (and perhaps one action, but could be no action). E.g. a game of chess
  - Continuous = Signals constantly coming into sensors, actions continually changing. E.g. self driving a car

Environment types

- Single agent (Vs. multi-agent): An agent operating by itself in an environment.
- An agent operating by itself in an environment is single agent!
- Multi agent is when other agents are present!
- Other agents - anything that changes from step to step.
- Competitive or co-operative - Multi-agent environments
- Ex: Single Agent - Maze, cars in a taxi drive.
- Multi Agent – foot ball

Environment types

	Deterministic vs Stochastic	Episodic vs Sequential	Static vs Dynamic	Discrete vs Continuous	Fully observable vs Partially observable	Single agent vs Multi-agent
Playing Soccer	Stochastic	Sequential	Dynamic	Continuous	Partially observable	Multi-agent
Brushing your teeth	Stochastic	Sequential	Static	Continuous	Fully observable	Single agent
Playing a tennis match	Stochastic	Sequential	Dynamic	Continuous	Partially observable	Multi-agent
Playing tennis against the wall	Stochastic	Sequential	Dynamic	Continuous	Fully observable	Single agent
Deciding what item to take at hostel lunch	Deterministic	Episodic	Static	Discrete	Fully observable	Single agent

Environment types

Reasons for "Playing Soccer"

i. **Stochastic** – For a given current state and action executed by agent, the next state or outcome cannot be exactly determined, for e.g., if agent kicks the ball in a particular direction, then the ball may or may not be stopped by other players, or the soccer field can change in many different ways depending on how players move.

ii. **Sequential** – The past history of actions in the game can affect the next action in the game.

iii. **Dynamic** – The environment can change while the agent is making decision, for e.g., soccer field (environment) changes when a player moves.

iv. **Continuous** – Location of the ball or players is continuous. The speed or the direction (angle) at which the agent hits the ball is continuous.

v. **Partially observable** – An agent cannot detect all the things on soccer field that can affect its action, for e.g. it cannot determine what other players are thinking.

vi. **Multi-agent** – There are many agents involved in soccer game.

Environment types

	Chess with a clock	Chess without a clock	Taxi driving
Fully observable	Yes	Yes	No
Deterministic	Strategic	Strategic	No
Episodic	No	No	No
Static	Semi	Yes	No
Discrete	Yes	Yes	No
Single agent	No	No	Yes

- The environment type largely determines the agent design

- The real world is (of course) partially observable, stochastic, sequential, dynamic, .continuous, multi-agent.
- Strategic environment - actions of other agents.

Environment types

Task Environment	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle	Fully	Single	Deterministic	Sequential	Static	Discrete
Chess with a 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. English tutor	Partially	Multi	Stochastic	Sequential	Dynamic	Discrete

Figure 2.6 Examples of task environments and their characteristics.

STRUCTURE OF AGENTS

- How it (Intelligent Agents) inside work
- Main job of AI is to design the **agent program** that implements agent function mapping percepts to actions
- Program will run on some sort of computing device with physical sensors & actuators called **architecture**
  - ***Agent = architecture+program***

Implement agent function

The computing device that we using
with sensors and actuators

Agent functions and programs

- The Agent program is one which can take the current percept as input from the sensors and return an action to the actuators
- The agent function can take the entire percept history.
- An agent is completely specified by the <u>agent function</u> mapping percept sequences to actions
- One agent function (or a small equivalence class) is <u>rational</u>
- Aim: find a way to implement the rational agent function concisely.

Table-Driven Agent

- Trivial agent program that keeps track of the percept sequence and then uses it to index into a table of action to decide what to do.

- The table represents explicitly the agent function that the agent program embodies.

- In order to build a rational agent, the designers must construct a table that contains the appropriate action for every possible percept sequence.

```
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
```

The TABLE-DRIVEN-AGENT program is invoked for each new percept and returns an action each time.

Table-Driven Agent

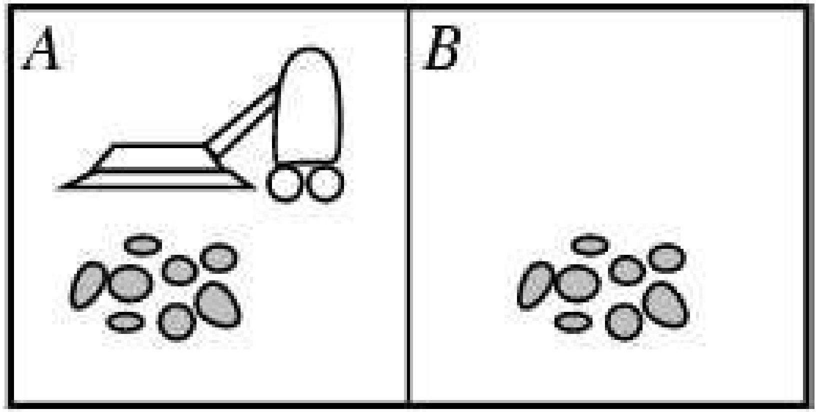
- **Drawbacks**:
  - Huge table
  - Take a long time to build the table
  - No autonomy
  - Even with learning, need a long time to learn the table entries

vacuum-cleaner 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*



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

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

Agent types

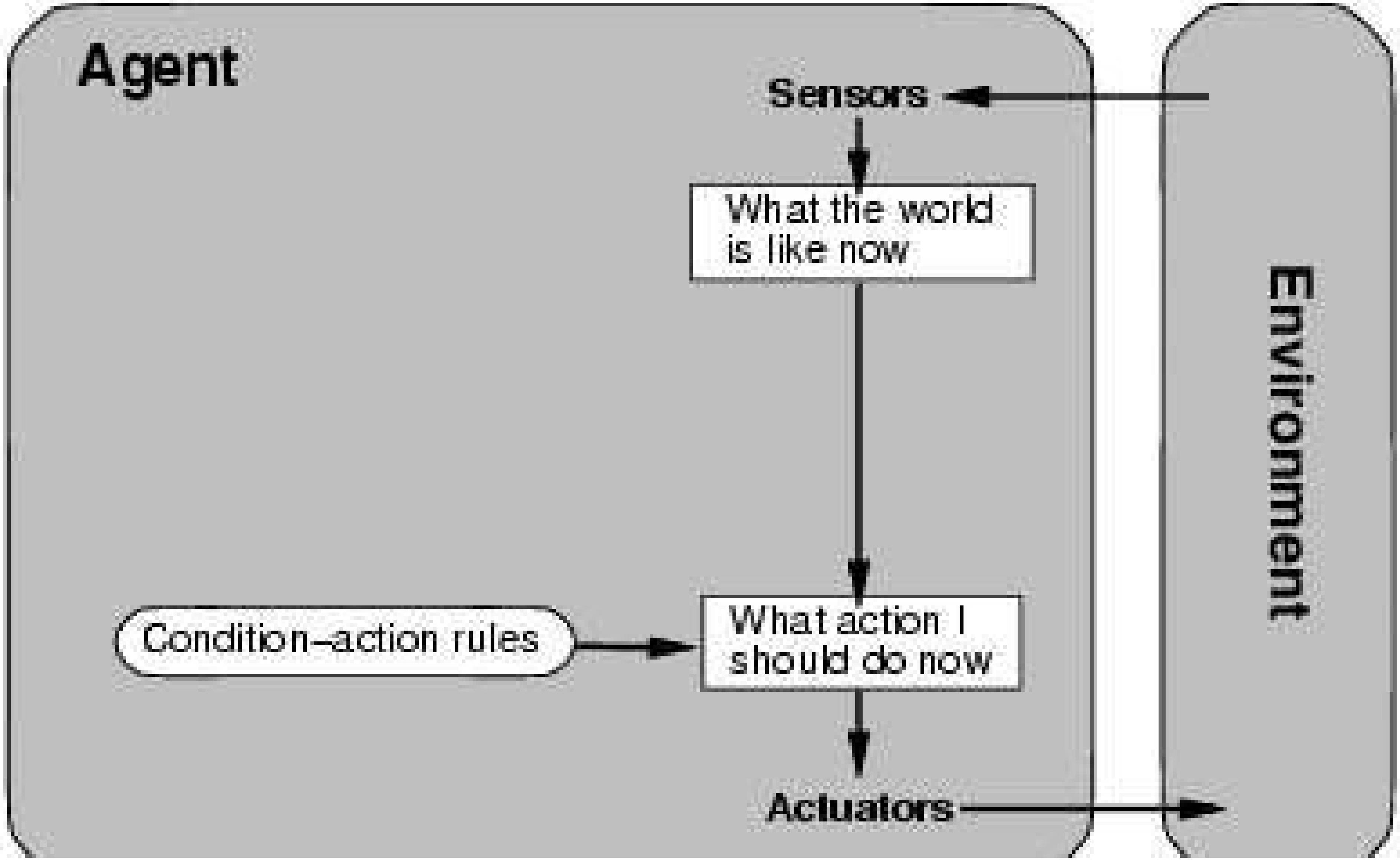
- Five basic types in order of increasing generality:
  - » **Simple reflex agents**
  - » **Model-based reflex agents**
  - » **Goal-based agents**
  - » **Utility-based agents**
  - » **Learning agent**

Simple reflex agents

- It select actions on the basis of the current percept, ignoring the rest of the percept history

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*
⋮	⋮

Simple reflex agents



```
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
```

- The INTERPRET-INPUT function generates an abstracted description of the current state from the percept.
- RULE-MATCH function returns the first rule in the set of rules that matches the given state description.

- Simple reflex agents have the admirable property of being **simple**, but they turn out to be of very limited intelligence (***The correct decision can be made on the basis of only the current percept i.e., environment is fully observable***)

- Even a little bit of unobservability can cause serious trouble

- Advantages:
  - Easy to implement
  - Uses much less memory than the table-driven agent
- Disadvantages:
  - Will only work correctly if the environment is fully observable
  - Infinite loops

For simple reflex agents operating in partially observable environments, infinite loops are often unavoidable. It may be possible to escape from infinite loops if the agent can randomize its actions.

Model-based reflex agents

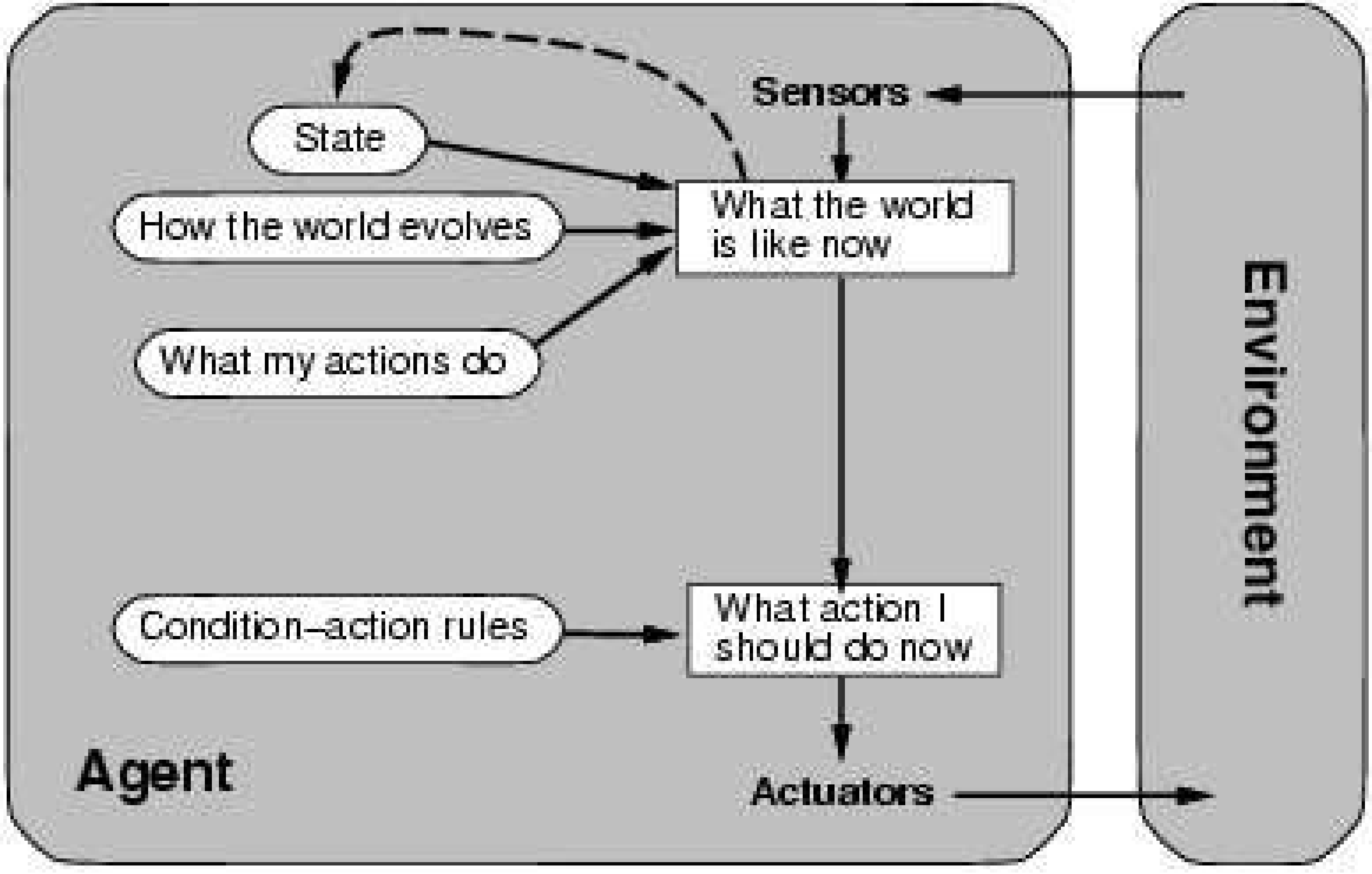
- The agent should maintain some sort of ***internal state*** that depends on the percept history and thereby reflects at least some of the unobserved aspect of the current state.
- Updating this internal state information as requires two kinds of knowledge to be encoded in the agent program
  - » **Information about how the world evolves independently of the agent**
  - » **Information about how the agent's own actions affect the world**

- The above said state information is implemented in simple Boolean circuits or in complete scientific theories called **model** of the world
- An agent that uses such a model is called a **model-based 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
```

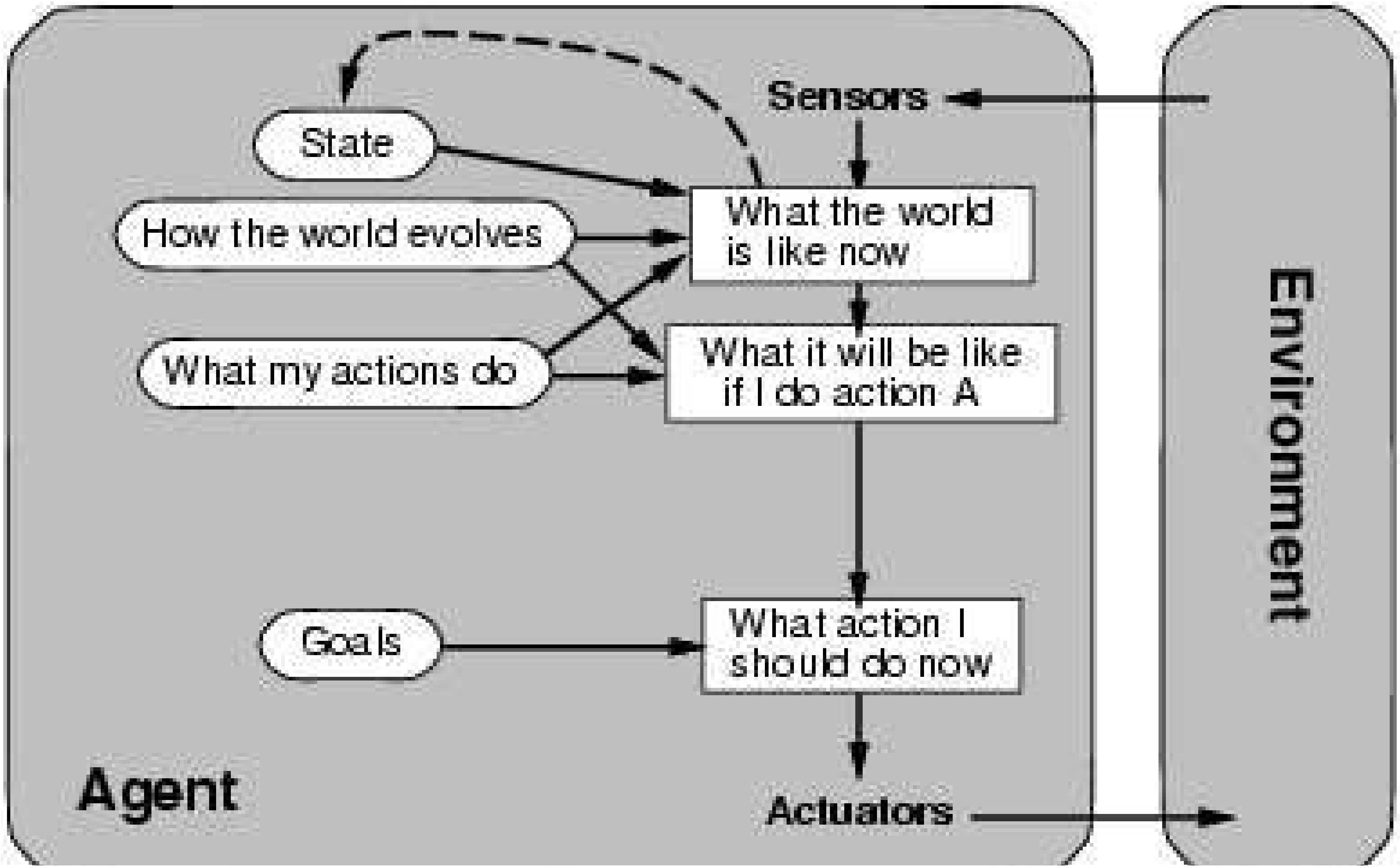
Model-based reflex agents



Goal-based agents

- Goal information guides agent’s actions (looks to the future)
- Sometimes achieving goal is simple eg. from a single action
- Other times, goal requires reasoning about long sequences of actions
- Flexible: simply reprogram the agent by changing goals

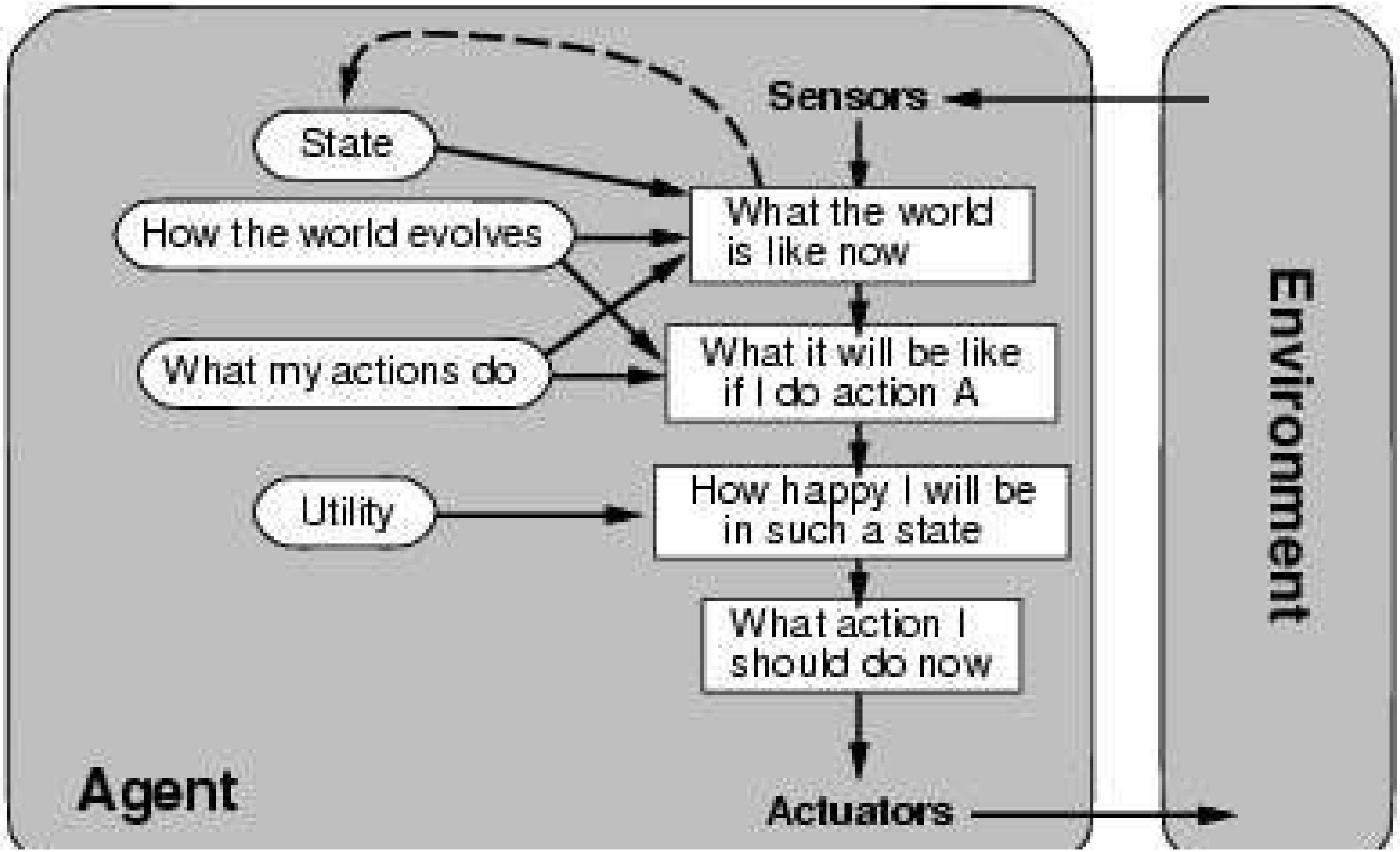
Goal-based agents



Utility-based agents

- What if there are many paths to the goal?
- Utility measures which states are preferable to other state
- Maps state to real number (utility or "happiness")

Utility-based agents



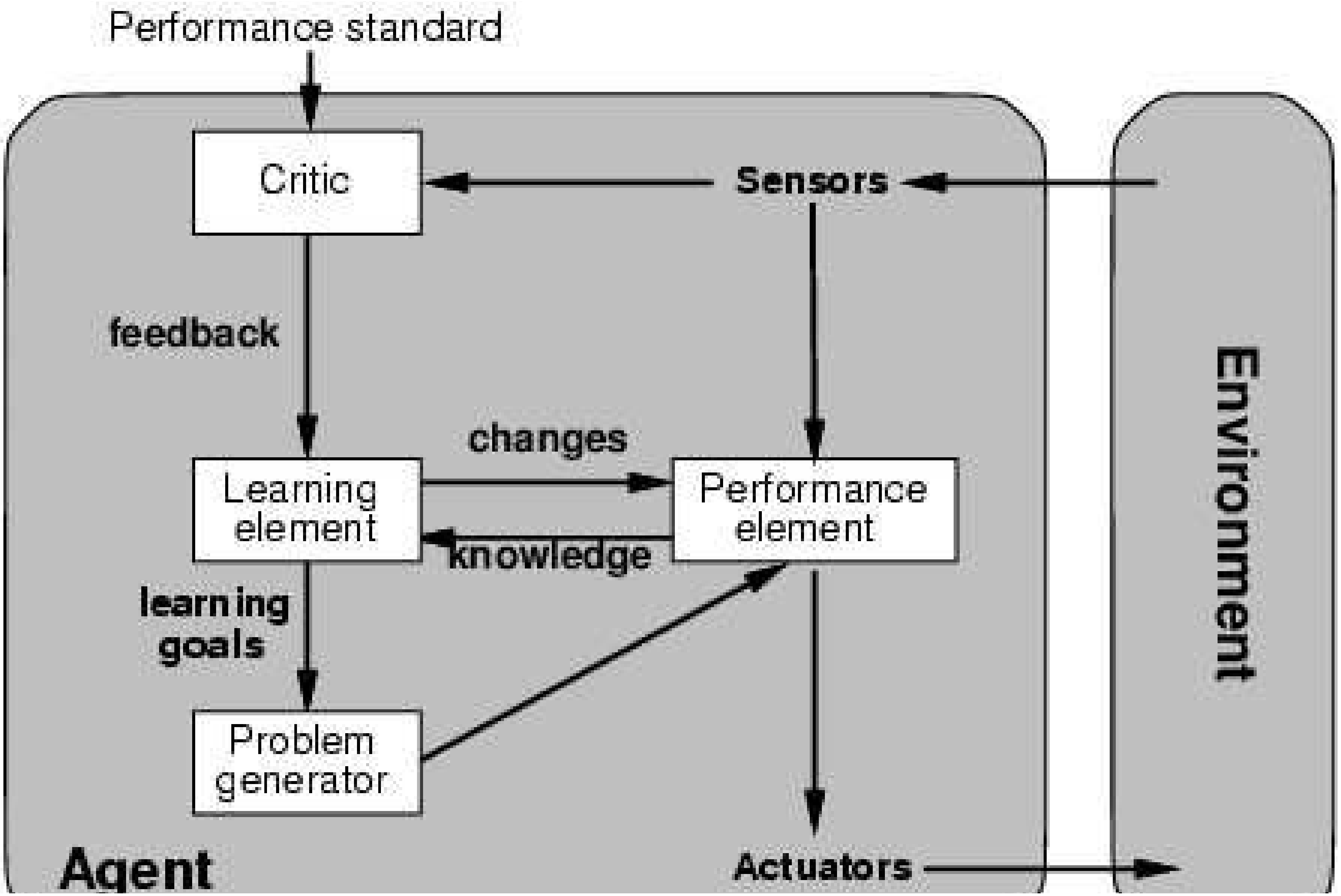
Learning agents

- Learn from its past experience or it has learning capabilities.
- It starts to act with basic knowledge and then able to act and adapt automatically through learning.

Four Conceptual Components

- Critic – How well the agent is doing with respect to fixed performance.
- Learning element – responsible for making improvements.
- Problem Generator – allows the agent to explore.
- Performance element – responsible for selecting external actions.

Learning agents



Learning agents