Artificial Intelligence

- **AI** is a branch of computer science which is concerned with the study and creation of computer systems that exhibit
  - some form of intelligence

    OR

  - those characteristics which we associate with intelligence in human behavior

- It has become an essential part of the technology industry.
- Research associated with artificial intelligence is highly technical and specialized.
- The core problems of artificial intelligence include programming computers for certain traits such as:
  - Knowledge
  - Reasoning
  - Problem solving
  - Perception
  - Learning
  - Planning
  - Ability to manipulate and move objects

- Knowledge engineering is a core part of AI research.
- Machines can often act and react like humans only if they have abundant information relating to the world.
- Artificial intelligence must have access to objects, categories, properties and relations between all of them to implement knowledge engineering.
- Initiating common sense, reasoning and problem-solving power in machines is a difficult and tedious approach.

Applications

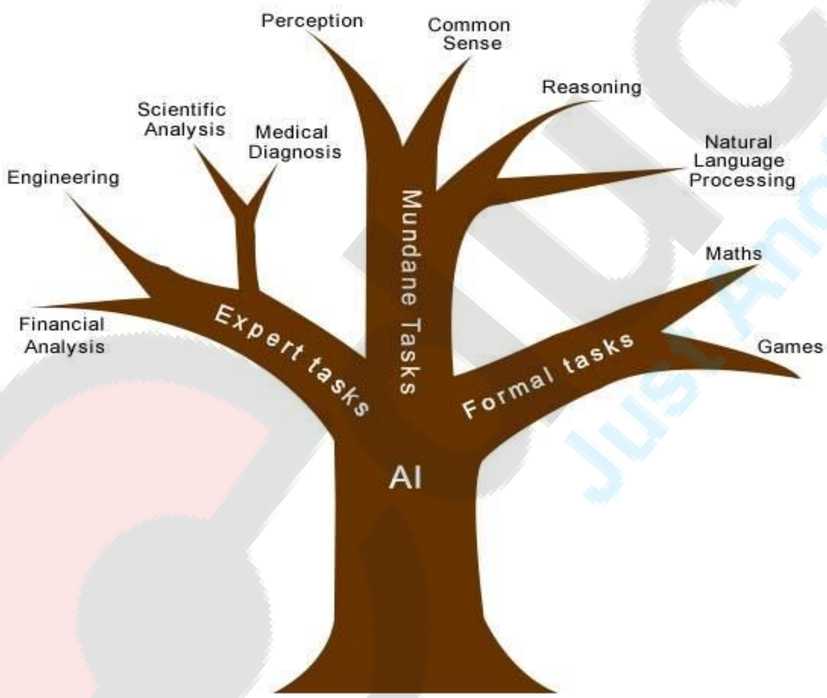
- Some of the applications are given below:
  - **Business :** Financial strategies, give advice
  - **Engineering:** check design, offer suggestions to create new product
  - **Manufacturing:** Assembly, inspection & maintenance
  - **Mining:** used when conditions are dangerous
  - **Hospital :** monitoring, diagnosing & prescribing
  - **Education :** In teaching

- **household :** Advice on cooking, shopping etc.
- **farming :** prune trees & selectively harvest mixed crops.

- Game Playing
  - Interactive computer programs
  - Made by creating human level artificial intelligent entities e.g. enemies, partners, support characters that act just like humans
  - Example: Checkers, chess
  - Deep Blue chess program won over world champion
- Speech recognition
  - A process of converting a speech signal to a sequence of words
  - Typical uses are
    - Voice dialing(call home)
    - Call routing(Collect call)
    - Speaker recognition
  - E.g. The spoken language interface PEGASUS in American Airlines' EAASY SABRE reservation system , allows user to obtain flight information and make reservations over telephone
- Understanding natural language
  - Natural language processing(NLP) does automated generation and understanding of natural human languages
  - Examples: Google Now feature, speech recognition, Automatic voice output
- Computer vision
  - Combination of concepts, ideas and techniques from: digital image processing, pattern recognition, AI and CG
  - E.g. Face recognition (programs in use by banks)
  - Autonomous driving- ALVINN system autonomously drove a van from Washington to San Diego

- **Expert Systems**
    - Examples − Flight-tracking systems, Clinical systems- MYCIN
- **Neural Networks**
    - Examples − Pattern recognition systems such as face recognition, character recognition, handwriting recognition
- **Robotics**
    - Examples − Industrial robots for moving, spraying, painting, precision checking, drilling, cleaning, coating, carving, etc
- **Fuzzy Logic Systems**
    - Examples − Consumer electronics, automobiles, etc

Task domains of AI



- **Mundane (ordinary) tasks**
    - Humans learn **mundane (ordinary) tasks** since their birth
    - They learn by perception, speaking, using language, and locomotives

- For humans, the mundane tasks are easiest to learn
- Earlier, all work of AI was concentrated in the mundane task domain
- **Example:**
    - ✓ Perception
    - ✓ Computer Vision
    - ✓ Speech, Voice
    - ✓ Natural Language Processing
    - ✓ Understanding
    - ✓ Language Generation
    - ✓ Language Translation
    - ✓ Common Sense
    - ✓ Reasoning
    - ✓ Planning
    - ✓ Robotics
    - ✓ Locomotive

➢ **Formal Tasks**

- Mathematics
- Geometry
- Logic
- Integration and Differentiation
- Games
- Go
- Chess (Deep Blue)
- Checkers
- Verification

- Theorem Proving

- **Expert tasks**
  - Expert task domain needs expert knowledge without common sense, which can be easier to represent and handle
  - **Example:**
    - ✓ Engineering
    - ✓ Fault Finding
    - ✓ Manufacturing
    - ✓ Monitoring
    - ✓ Scientific Analysis
    - ✓ Financial Analysis
    - ✓ Medical Diagnosis
    - ✓ Creativity

Task environment

- In designing an agent, the first step must always be to specify the task environment
- The problem the agent solves is characterized by Performance Measure, Environment, Actuators, and Sensors (PEAS)
- For specifying the task environment PEAS description should be as fully as possible
- For example: agent to be designed is an **automated taxi driver**
- PEAS description for an **automated taxi driver** is given below
- **Performance measures:** to specify Performance measures following points should be considered
  - ✓ 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.

- **Environment:** to specify Environment following points should be considered
  - ✓ 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

- **Actuators** (for outputs): to specify actuators following points should be considered
  - It should have control over the accelerator, steering, gear shifting and braking
  - A taxi must have display to communicate with the customers

- **Sensors** (for inputs): to specify sensors following points should be considered
  - It should be able to detect other vehicles, road situations
  - A taxi must have GPS (Global Positioning System) to know where the taxi is
  - Many more devices are necessary

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

Properties of task environment

1. **Fully observable vs. Partially observable**
    - **Fully 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 and 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
2. **Deterministic vs. stochastic**
    - If 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
    - Example: Cleaner and taxi driver are: Stochastic because of some unobservable aspects are a noise or unknown
3. **Episodic vs. sequential**
    - An episode is agent's single pair of perception & action
    - **Episodic**
        - 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**
        - In this kind of environment current action may affect all future decisions
        - Ex. Taxi driving and chess
4. **Static vs. dynamic**
    - **Dynamic**
        - A dynamic environment is always changing over time
        - E.g., the number of people in the street
    - **Static**
        - If the environment does not change while an agent is acting, then it is static
        - E.g., the destination
5. **Discrete vs. continuous**
    - **Discrete**

- If there are a limited number of distinct states, clearly defined percepts and actions, the environment is discrete
  - E.g. Chess game
- **Continuous**
  - Otherwise it is continuous
  - E.g. Taxi driving

6. Single agent VS. multi agent

- The environment may contain other agents which may be of the same or different kind as that of the 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

7. Known vs. unknown

- This distinction refers not to the environment itself but to the agent's (or designer's) state of knowledge about the environment
- **Known**
  - In known environment, the outcomes for all actions are given (example: solitaire card games)
- **Unknown**
  - If the environment is unknown, the agent will have to learn how it works in order to make good decisions(example: new video game)

➢ **Example**

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

Agent

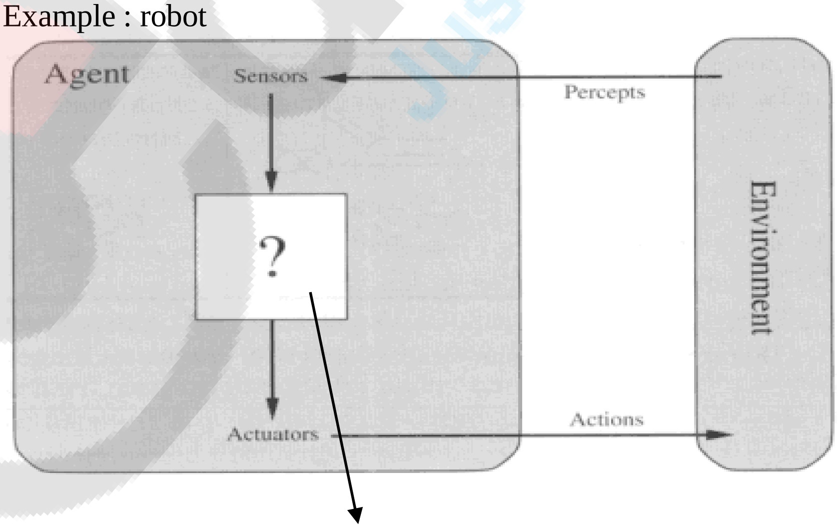
- An agent is anything that can perceive its environment through sensors and acts upon that environment through actuators/effectors
- **Example:**
  - A **human agent** has eyes, ears, and other organs for sensors, and hands, legs, mouth, and other body parts for effectors
  - A **robotic agent** substitutes cameras and infrared range finders for the sensors and various motors for the effectors
  - A **thermostat** detecting room temperature

Rational agent

- Rational agent is one that does the right thing and every entry in the table for the agent function is correct (rational)
- The rationality of the agent is measured by its performance measure, the prior knowledge it has, the environment it can perceive and actions it can perform
- 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
- A *rational* system tries to get the best possible outcome given limited knowledge
- A **chess AI** would be a good example of this

Intelligent agent

- An Intelligent agent must sense, must act, must be autonomous(to some extent), must be rational
- An agent which acts in a way that is expected to maximize to its performance measure, given the evidence provided by what it perceived and whatever built-in knowledge it has is known as Intelligent agent
- The performance measure defines the criterion of success for an agent.
- Example : robot



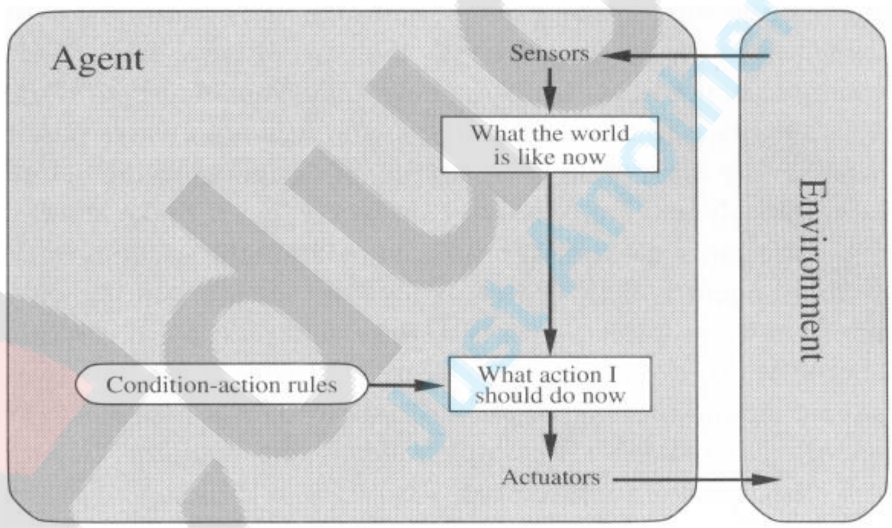
What AI should fill

Agent classification

- Agents are grouped into five classes based on their degree of perceived intelligence and capability
  - Simple reflex agents
  - Model based reflex agents
  - Goal based agents
  - Utility based agents
  - Learning agents

- **Simple reflex agents**

- Act only on the basis of the current percept, ignoring the rest of the percept history
- It uses just ***condition-action rules***
- The rules are like the form "if … then …"
- These agents are efficient but have narrow range of applicability because knowledge sometimes cannot be stated explicitly
- They Succeed when the environment is fully observable
- E.g. Vacuum cleaner

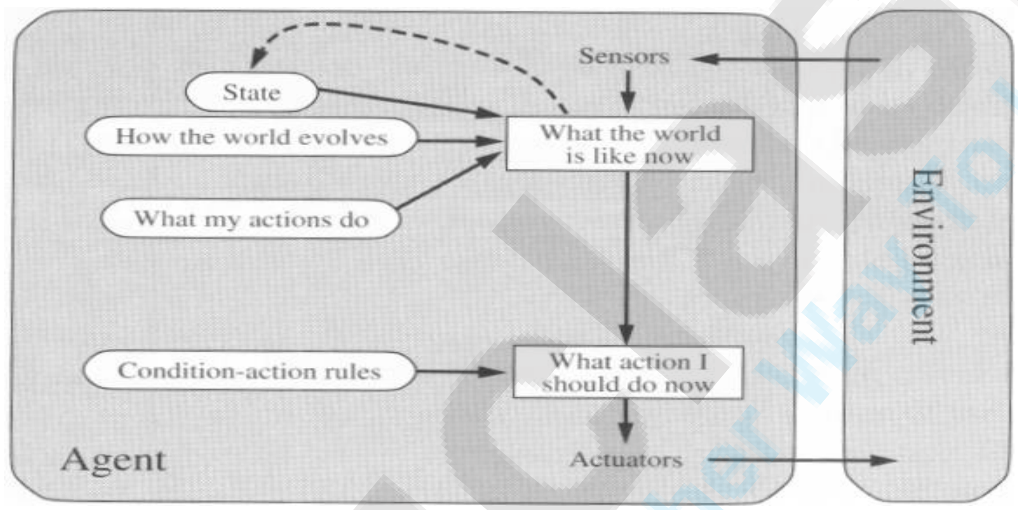


- **Model based reflex agents**

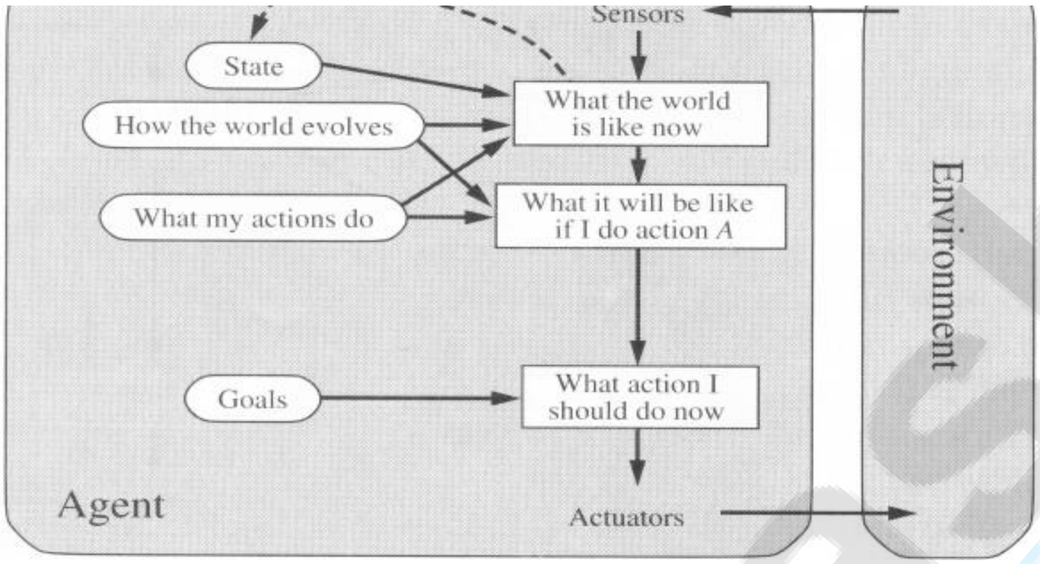
- Agents keep track of partially observable environments.
- These have an internal state depending on perception history.
- The environment/ world is modeled based on how it evolves independently from the agent, and how the agent actions affects the world.

- E.g., driving a car and changing lane

- Agents Require two types of knowledge
  - How the world evolves independently of the agent
  - How the agent’s actions affect the world
  - E.g. **self-steering mobile vision**
- The agent is with memory

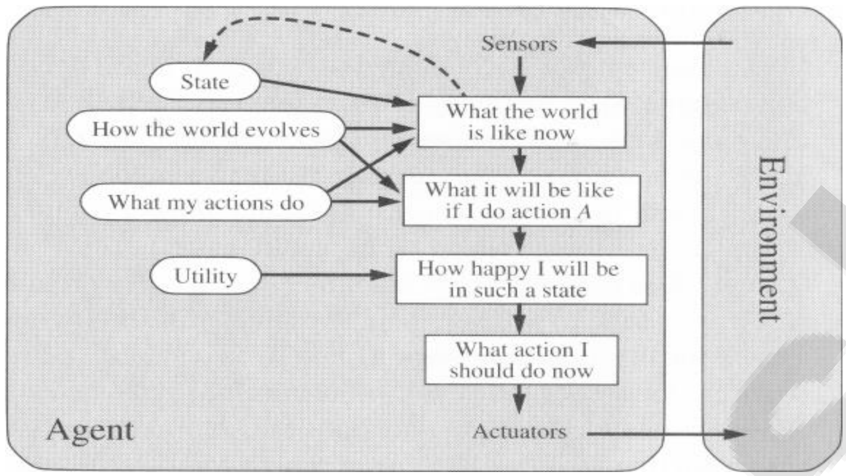


- **Goal based agents**
  - This is an improvement over model based agents, and used in cases where knowing the current state of the environment is not enough
  - Goal-based agents further expand on the capabilities of the model-based agents, by using "goal" information
  - Goal information describes situations that are desirable
  - This allows the agent a way to choose among multiple possibilities, selecting the one which reaches a goal state
  - Agents combine the provided goal information with the environment model, to choose the actions which achieve that goal.
  - Goal-based agents are less efficient but more flexible
  - E.g. **searching robots** that has initial location and want to reach a destination



- **Utility-based agents**

- Utility-based agents are an improvement over goal based agents, helpful when achieving the desired goal is not enough.
- We might need to consider a cost.
- For example, we may look for quicker, safer, cheaper trip to reach a destination.
- This is denoted by a utility function.
- A utility agent will chose the action that maximizes the expected utility
- If goal means success, then **utility** means the degree of success (how successful it is)
- E.g. **route recommendation system** which solves for the 'best' route to reach a destination



- **Learning agents**
- A general intelligent agent, also known as learning agent, was proposed by Alan Turing, and is now the preferred method for creating state-of-the-art systems in Artificial Intelligence.
- All the agents described above can be generalized into these learning agents to generate better actions.
- Four conceptual components of learning agents are:
    - Learning element: it is responsible for making improvement
    - Performance element: it is responsible for 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