Artificial Intelligence

Unit-4 (Structural Knowledge Representation)

Artificial Intelligence 01CE0702

Marwadi University

Department of Computer Engineering

Shilpa Singhal

What is Knowledge?

- Knowledge is a progression that starts with **data** which is of limited utility.
- By organizing or analyzing the data, we understand what the data means, and this becomes **information.**
- The interpretation or evaluation of information yield **knowledge.**
- An understanding of the principles embodied within the knowledge is **wisdom.**

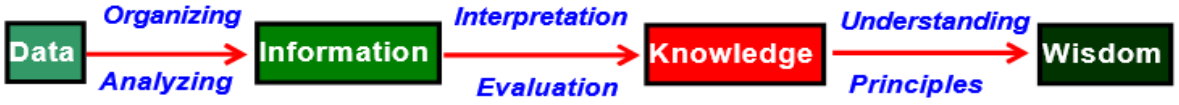


Fig 1 Knowledge Progression

Knowledge Progression

- **Data** is viewed as collection of ***disconnected facts.***

  Example : It is raining.

- **Information** emerges when ***relationships among facts*** are established and understood;Provides answers to *"who", "what", "where", and "when".*

  Example : The temperature dropped 15 degrees and then it started raining**.**

- **Knowledge** emerges when ***relationships among patterns*** are identified and understood;Provides answers as *"how"* .

  Example : If the humidity is very high and the temperature drops substantially,then atmospheres is unlikely to hold themoisture, so it rains.

- **Wisdom** is the pinnacle ofunderstanding, uncovers the ***principles of relationships thatdescribe patterns*** **;** Provides answers as *"why"* .

  Example : Encompasses understanding of all the interactions that happen between raining, evaporation, air currents, temperature gradients, changes, and raining.

Types of Knowledge?

Knowledge is categorized into two major types: **Tacit** and **Explicit.**

- term **"Tacit"** corresponds to informal or implicit type of knowledge,
- term **"Explicit"** corresponds to formal type of knowledge.

Tacit knowledge	Explicit knowledge
Exists within a human being; it is embodied.	Exists outside a human being; it is embedded.
Difficult to articulate formally.	Can be articulated formally.
Difficult to share/communicate.	Can be shared, copied, processed and stored.
Hard to steal or copy.	Easy to steal or copy
Drawn from experience, action, subjective insight.	Drawn from artifact of some type as principle, procedure, process, concepts.

Knowledge Topology Map

The map shows that, **Tacit knowledge** comes from experience, action, subjective insight and **Explicit knowledge** comes from principle, procedure, process, concepts, via transcribed content or artifact of some type.

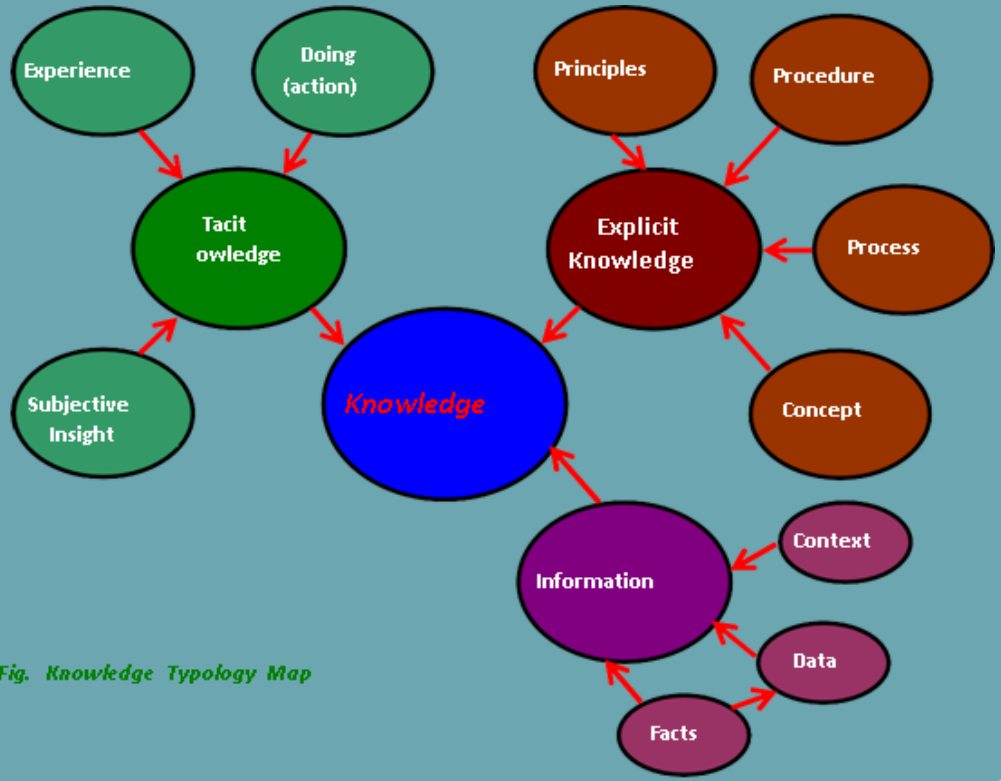


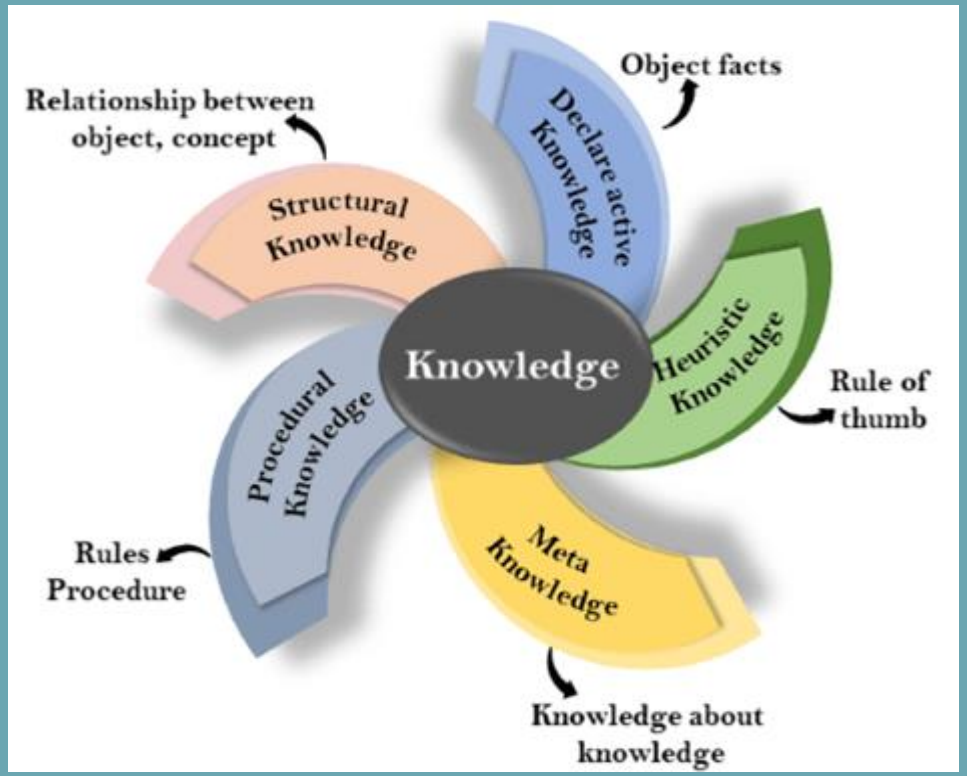
Fig. Knowledge Typology Map

Knowledge Representation

- Humans are best at understanding, reasoning, and interpreting knowledge.
- Human knows things, which is knowledge and as per their knowledge they perform various actions in the real world.
- But how machines do all these things comes under knowledge representation and reasoning. Hence we can describe Knowledge representation as following:
  - Knowledge representation and reasoning (KR, KRR) is the part of Artificial intelligence which concerned with AI agents thinking and how thinking contributes to intelligent behavior of agents.
  - It is responsible for representing information about the real world so that a computer can understand and can utilize this knowledge to solve the complex real world problems such as diagnosis a medical condition or communicating with humans in natural language.
  - It is also a way which describes how we can represent knowledge in artificial intelligence. Knowledge representation is not just storing data into some database, but it also enables an intelligent machine to learn from that knowledge and experiences so that it can behave intelligently like a human.

- Following are the kind of knowledge which needs to be represented in AI systems:
  - **Object:** All the facts about objects in our world domain. E.g., Guitars contains strings, trumpets are brass instruments.
  - **Events:** Events are the actions which occur in our world.
  - **Concepts :** are class of items, words, or ideas that are known by a common name and share common features.
  - **Processes :** are flow of events or activities that describe how things work rather than how to do things.
  - **Procedures :** are series of step-by-step actions and decisions that result in the achievement of a task.
  - **Principles :** are guidelines, rules, and parameters that govern; principles allow to make predictions and draw implications; principles are the basic building blocks of theoretical models (theories).
  - **Performance:** It describe behavior which involves knowledge about how to do things.
  - **Meta-knowledge:** It is knowledge about what we know.
  - **Facts:** Facts are the truths about the real world and what we represent.
  - **Knowledge-Base:** The central component of the knowledge-based agents is the knowledge base. It is represented as KB. The Knowledgebase is a group of the Sentences (Here, sentences are used as a technical term and not identical with the English language).

Types of Knowledge

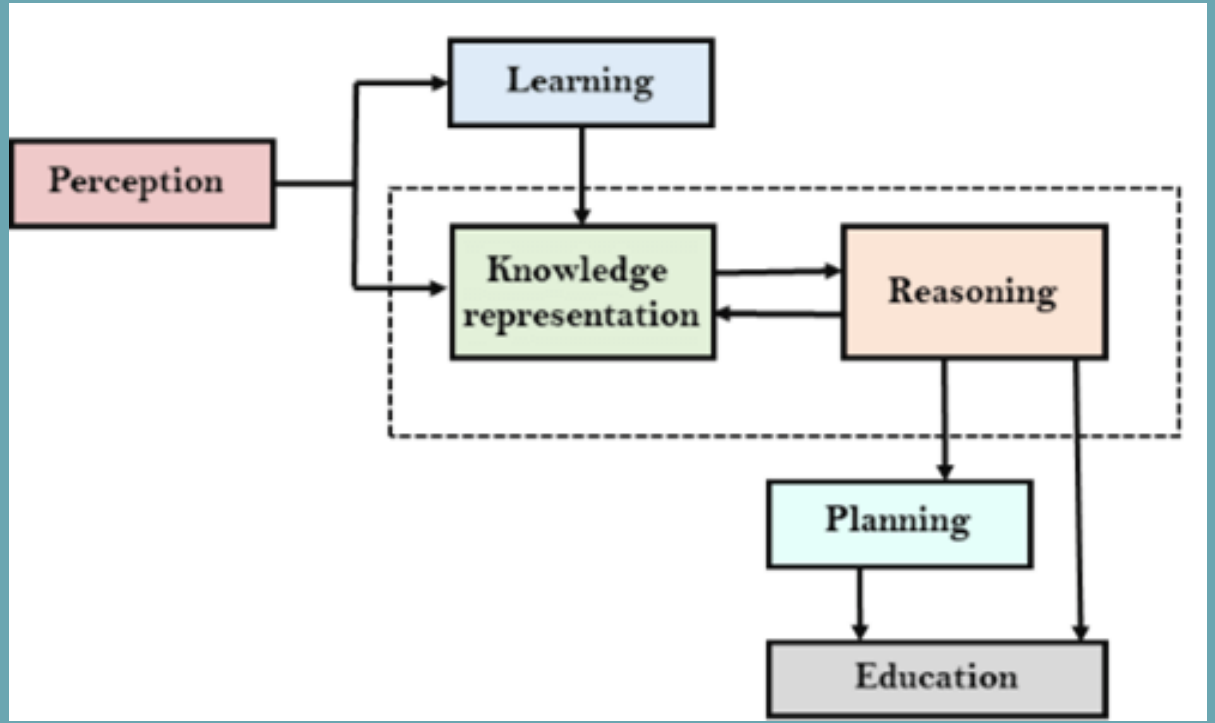


- **Declarative Knowledge:**
  - Declarative knowledge is to know about something.
  - It includes concepts, facts, and objects.
  - It is also called descriptive knowledge and expressed in declarativesentences.
  - It is simpler than procedural language.
- **Procedural Knowledge**
  - It is also known as imperative knowledge.
  - Procedural knowledge is a type of knowledge which is responsible for knowing how to do something.
  - It can be directly applied to any task.
  - It includes rules, strategies, procedures, agendas, etc.
  - Procedural knowledge depends on the task on which it can be applied.

Types of Knowledge

- **Meta-knowledge:**
  - Knowledge about the other types of knowledge is called Meta-knowledge.
- **Heuristic knowledge:**
  - Heuristic knowledge is representing knowledge of some experts in a filed or subject.
  - Heuristic knowledge is rules of thumb based on previous experiences, awareness of approaches, and which are good to work but not guaranteed.
- **Structural knowledge:**
  - Structural knowledge is basic knowledge to problem-solving.
  - It describes relationships between various concepts such as kind of, part of, and grouping of something.
  - It describes the relationship that exists between concepts or objects.

AI knowledge cycle



Marwadi University

S I I P

Schemes to knowledge representation

1. Simple relational knowledge
2. Inheritable knowledge
3. Inferential knowledge
4. Procedural knowledge

Simple relational knowledge

- Relational Knowledge is used to **associate elements of one domain with the elements of another domain or set of design constrains.**
- Relational knowledge is made up of objects consisting of attributes and their corresponding associated values.
- The results of this knowledge type is a mapping of elements among different domains.
- This approach of knowledge representation is famous in database systems where the relationship between different entities is represented.

Table - Simple Relational Knowledge

Player	*Height*	*Weight*	*Bats - Throws*
Aaron	6-0	180	Right - Right
Mays	5-10	170	Right - Right
Ruth	6-2	215	Left - Left
Williams	6-3	205	Left - Right

Simple relational knowledge

Table - Simple Relational Knowledge

Player	Height	Weight	Bats - Throws
Aaron	6-0	180	Right - Right
Mays	5-10	170	Right - Right
Ruth	6-2	215	Left - Left
Williams	6-3	205	Left - Right

The table below shows a simple way to store facts.

− The facts about a set of objects are put systematically in columns.

− This representation provides little opportunity for inference.

‡ Given the facts it is not possible to answer simple question such as :

" Who is the heaviest player ? ".

‡ But if a procedure for finding heaviest player is provided, then these facts will enable that procedure to compute an answer.

Inheritable knowledge

- This approach contains inheritable knowledge which shows a relation between instance and class, and it is called instance relation.
- In this approach **elements inherit attributes from their parents.**
- Every individual frame can represent the collection of attributes and its value.
- Within the hierarchy, elements inherit attributes from their parents, but in many cases, not all attributes of the parent elements be prescribed to the child elements.
- The KR in hierarchical structure, shown below, is called “semantic network "or a collection of “frames” or “slot-and-filler structure". It shows property inheritance and way for insertion of additional knowledge.
- The **directed arrows** represent attributes (isa, instance, and team) originating at the object being described and terminating at the object or its value.
- The **box nodes** represents objects and values of the attributes.

Inheritable knowledge

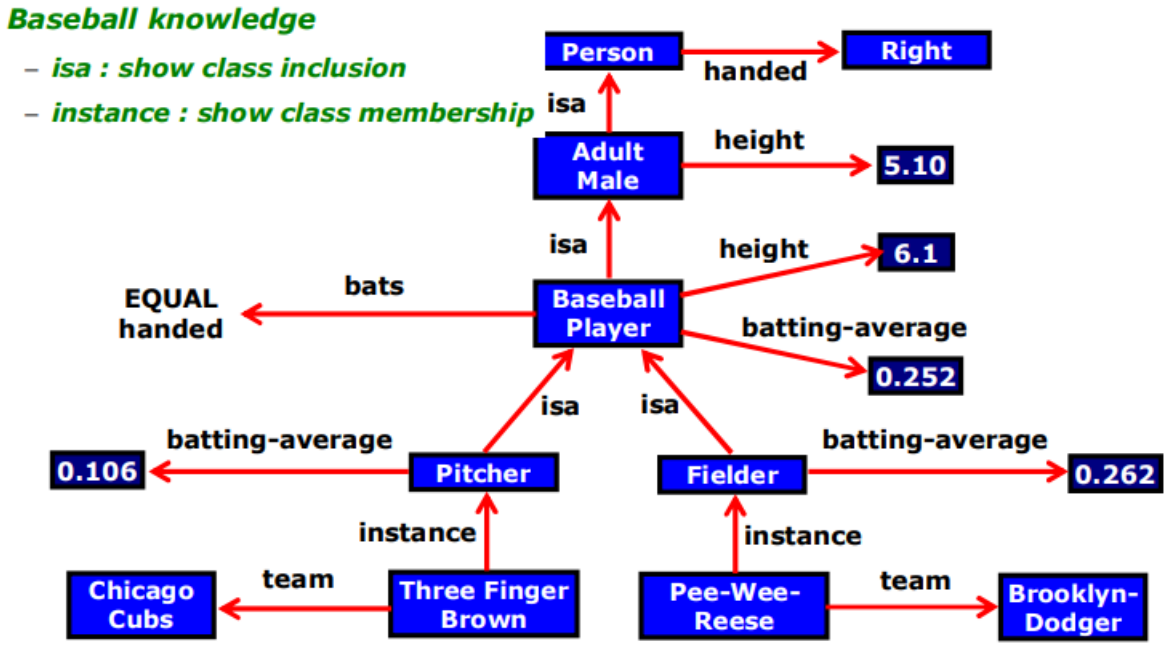


Fig. Inheritable knowledge representation (KR)

Inheritable knowledge

Viewing a node as a frame: Baseball-player

Node	**Baseball-player**
isa :	Adult-Male
Bates :	EQUAL handed
Height :	6.1
Batting-average :	0.252

Inferential knowledge

- Inferential knowledge approach represents knowledge in the form of formal logics.
- This approach can be used to derive more facts.
- Generates new information from the given information. This new information does not require further data gathering form source, but does require analysis of the given information to generate new knowledge.
- It guaranteed correctness.
- It require analysis of the given information to generate new knowledge.
- Given a set of relations and values, one may infer other values or relations.

  − In addition to algebraic relations, a predicate logic (mathematical deduction) is used to infer from a set of attributes.

  − Inference through predicate logic uses a set of logical operations to relate individual data.

Inferential knowledge

The symbols used for the logic operations are :
" → " (implication), " ¬ " (not), " V " (or), " Λ " (and), " ∀ " (for all), " ∃ " (there exists).

Examples of predicate logic statements :

1. Wonder is a name of a dog : ***dog (wonder)***
2. All dogs belong to the class of animals : ***∀x : dog (x) → animal(x)***
3. All animals either live on land or in water : ***∀x : animal(x) → live (x, land) V live (x, water)***

We can infer from these three statements that :

" Wonder lives either on land or on water."

As more information is made available about these objects and their relations, more knowledge can be inferred.

Procedural Language	Declarative Language
Basic, C++, Cobol, etc.	SQL
Most work is done by interpreter of the languages	Most work done by Data Engine within the DBMS
For one task many lines of code	For one task one SQL statement
Programmer must be skilled in translating the objective into lines of procedural code	Programmer must be skilled in clearly stating the objective as a SQL statement
Requires minimum of management around the actual data	Relies on SQL-enabled DBMS to hold the data and execute the SQL statement .
Programmer understands and has access to each step of the code	Programmer has no interaction with the execution of the SQL statement
Data exposed to programmer during execution of the code	Programmer receives data at end as an entire set
More susceptible to failure due to changes in the data structure	More resistant to changes in the data structure
Traditionally faster, but that is changing	Originally slower, but now setting speed records
Code of procedure tightly linked to front end	Same SQL statements will work with most front ends Code loosely linked to front end.
Code tightly integrated with structure of the data store	Code loosely linked to structure of data; DBMS handles structural issues
Programmer works with a pointer or cursor	Programmer not concerned with positioning
Knowledge of coding tricks applies only to one language	Knowledge of SQL tricks applies to any language using SQL

Declarative/Procedural knowledge

Semantic Net

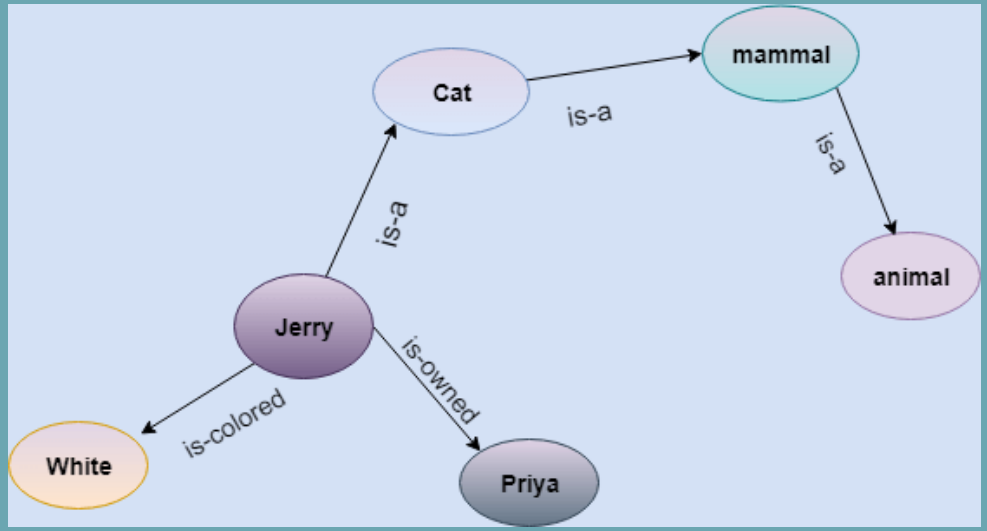
- Semantic networks are alternative of predicate logic for knowledge representation.
- In Semantic networks, we can represent our knowledge in the form of graphical networks.
- This network consists of nodes representing objects and arcs which describe the relationship between those objects.
- Semantic networks can categorize the object in different forms and can also link those objects. Semantic networks are easy to understand and can be easily extended.
- This representation consist of mainly two types of relations:
  - IS-A relation (Inheritance)
  - Kind-of-relation

Semantic Net

Example:

Statements:

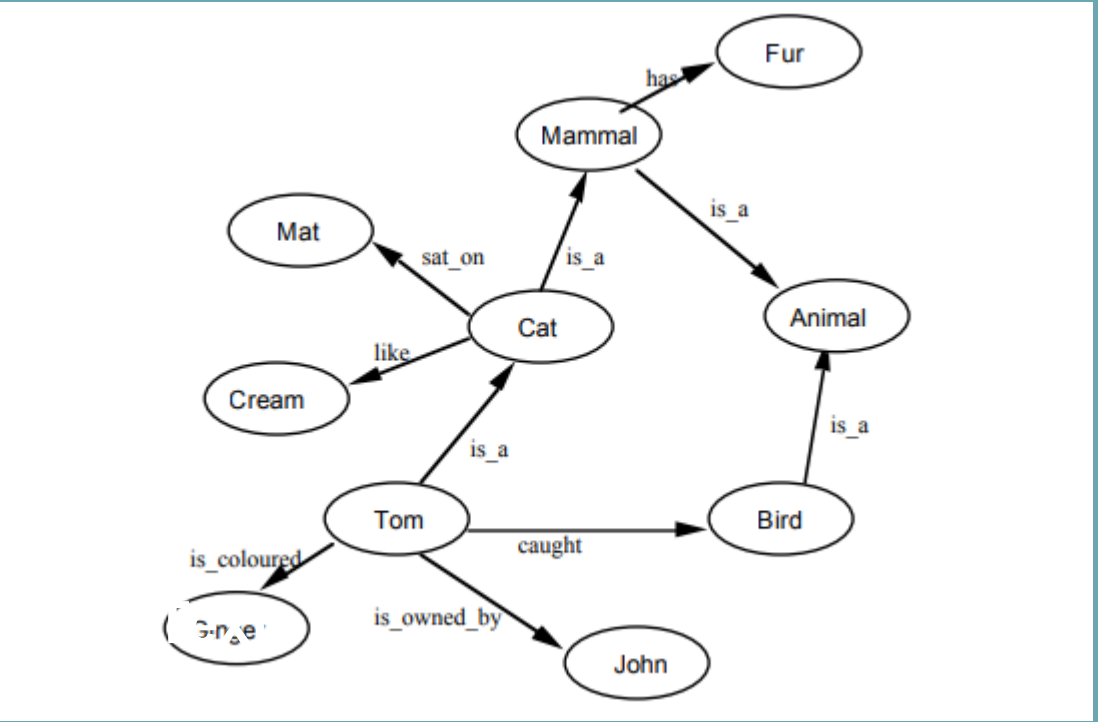
- Jerry is a cat.
- Jerry is a mammal
- Jerry is owned by Priya.
- Jerry is white colored.
- All Mammals are animal



Semantic Net

Example:

- Tom is a cat.
- Tom caught a bird.
- Tom is owned by John.
- Tom is ginger in colour.
- Cats like cream.
- The cat sat on the mat.
- A cat is a mammal.
- A bird is an animal.
- All mammals are animals.
- Mammals have fur.



Components Of Semantic Networks

LSSP

- **Lexical component:-**nodes denoting physical objects or links are relationships between objects; labels denote the specific objects and relationships
- **Structural component:-** the links or nodes from a diagram which is directed.
- **Semantic component:-** Here the definitions are related only to the links and label of nodes, whereas facts depend on the approval areas.
- **Procedural part:-** constructors permit the creation of the new links and nodes. The removal of links and nodes are permitted by destructors.

Drawbacks in Semantic representation:

- Semantic networks take more computational time at runtime as we need to traverse the complete network tree to answer some questions. It might be possible in the worst case scenario that after traversing the entire tree, we find that the solution does not exist in this network.
- Semantic networks try to model human-like memory (Which has 1015 neurons and links) to store the information, but in practice, it is not possible to build such a vast semantic network.
- These types of representations are inadequate as they do not have any equivalent quantifier, e.g., for all, for some, none, etc.
- Semantic networks do not have any standard definition for the link names.
- These networks are not intelligent and depend on the creator of the system.

Advantages in Semantic representation:

- The semantic network is more natural than the logical representation;
- The semantic network permits using of effective inference algorithm (graphical algorithm)
- They are simple and can be easily implemented and understood.
- The semantic network can be used as a typical connection application among various fields of knowledge, for instance, among computer science and anthropology.
- The semantic network permits a simple approach to investigate the problem space.
- The semantic network gives an approach to make the branches of related components.

Frames

- A frame is a record like structure which consists of a collection of attributes and its values to describe an entity in the world.
- Frames are the AI data structure which divides knowledge into substructures by representing stereotypes situations.
- It consists of a collection of slots and slot values. These slots may be of any type and sizes.
- Slots have names and values which are called facets.
- Facets:
  - The various aspects of a slot is known as Facets. Facets are features of frames which enable us to put constraints on the frames.
  - Example: IF-NEEDED facts are called when data of any particular slot is needed. A frame may consist of any number of slots, and a slot may include any number of facets and facets may have any number of values.
  - A frame is also known as slot-filter knowledge representation in artificial intelligence.

Frames

- Frames are derived from semantic networks and later evolved into our modern-day classes and objects.
- A single frame is not much useful. Frames system consist of a collection of frames which are connected.
- In the frame, knowledge about an object or event can be stored together in the knowledge base.
- The frame is a type of technology which is widely used in various applications including Natural language processing and machine visions.

Frames

Example:

Slots	Filters
Title	Artificial Intelligence
Genre	Computer Science
Author	Peter Norvig
Edition	Third Edition
Year	1996
Page	1152

Advantages of frame representation:

- The frame knowledge representation makes the programming easier by grouping the related data.
- The frame representation is comparably flexible and used by many applications in AI.
- It is very easy to add slots for new attribute and relations.
- It is easy to include default data and to search for missing values.
- Frame representation is easy to understand and visualize

Disadvantages of frame representation:

- In frame system inference mechanism is not be easily processed.
- Inference mechanism cannot be smoothly proceeded by frame representation.
- Frame representation has a much generalized approach.

Agents

What is an Agent?

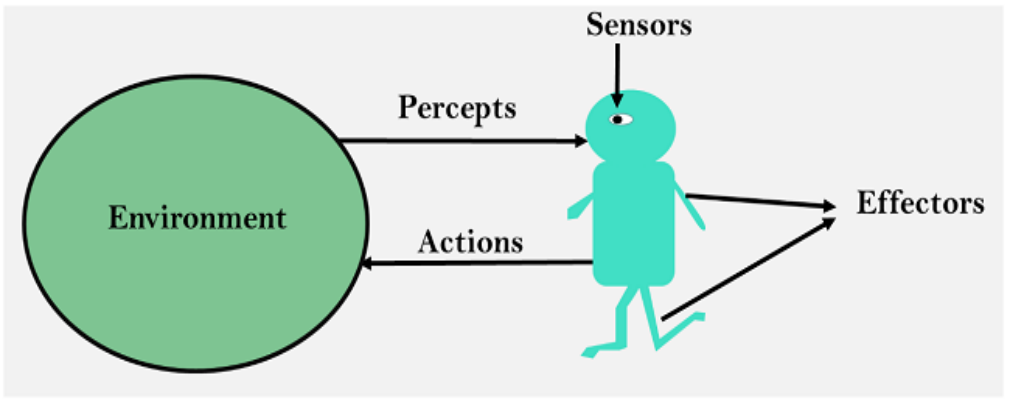
An agent can be anything that perceive its environment through sensors and act upon that environment through actuators. An Agent runs in the cycle of **perceiving**, **thinking**, and **acting**.

An agent can be:

- **Human-Agent:** A human agent has eyes, ears, and other organs which work for sensors and hand, legs, vocal tract work for actuators.
- **Robotic Agent:** A robotic agent can have cameras, infrared range finder, NLP for sensors and various motors for actuators.
- **Software Agent:** Software agent can have keystrokes, file contents as sensory input and act on those inputs and display output on the screen.

Agents

- **Sensor:** Sensor is a device which detects the change in the environment and sends the information to other electronic devices. An agent observes its environment through sensors.
- **Actuators:** Actuators are the component of machines that converts energy into motion. The actuators are only responsible for moving and controlling a system. An actuator can be an electric motor, gears, rails, etc.
- **Effectors:** Effectors are the devices which affect the environment. Effectors can be legs, wheels, arms, fingers, wings, fins, and display screen.



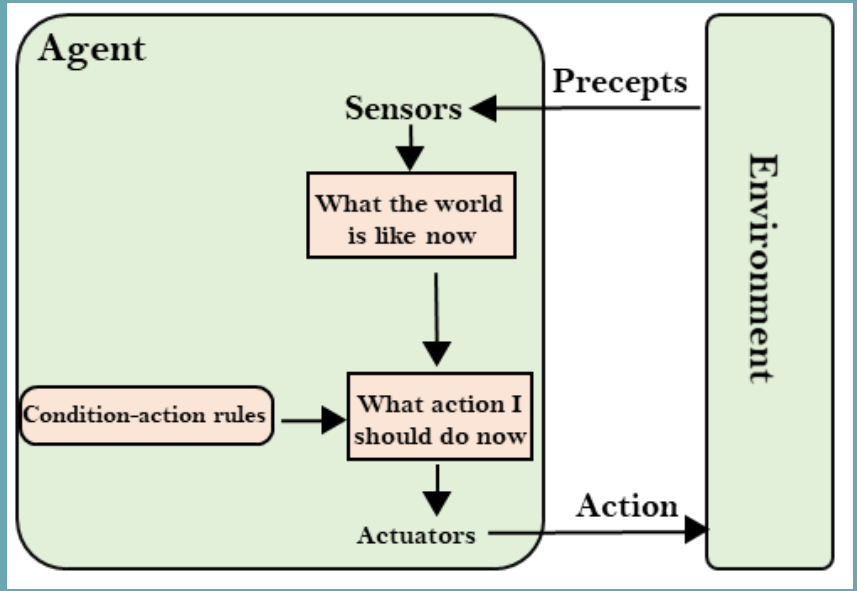
Types of Agents

Agents can be 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

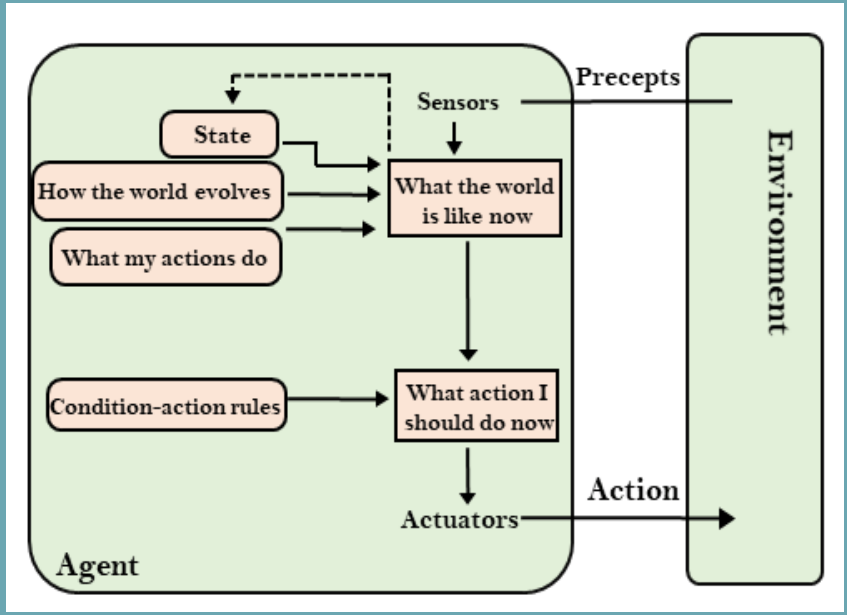
Simple Reflex Agents

- The Simple reflex agents are the simplest agents. These agents take decisions on the basis of the current percepts and ignore the rest of the percept history.
- These agents only succeed in the fully observable environment.
- The Simple reflex agent does not consider any part of percepts history during their decision and action process.
- The Simple reflex agent works on Condition-action rule, which means it maps the current state to action. Such as a Room Cleaner agent, it works only if there is dirt in the room.
- Problems for the simple reflex agent design approach:
  - They have very limited intelligence
  - They do not have knowledge of non-perceptual parts of the current state
  - Mostly too big to generate and to store.
  - Not adaptive to changes in the environment.



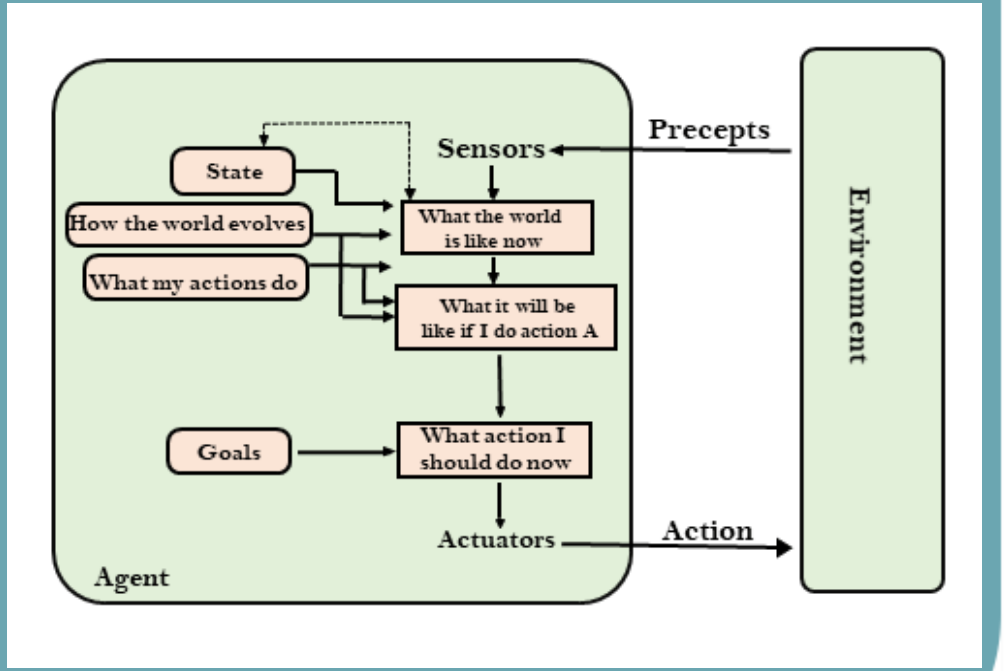
Model-Based Agents

- The Model-based agent can work in a partially observable environment, and track the situation.
- A model-based agent has two important factors:
  - **Model:** It is knowledge about "how things happen in the world," so it is called a Model-based agent.
  - **Internal State:** It is a representation of the current state based on percept history.
- These agents have the model, "which is knowledge of the world" and based on the model they perform actions.
- Updating the agent state requires information about:
  - How the world evolves
  - How the agent's action affects the world.



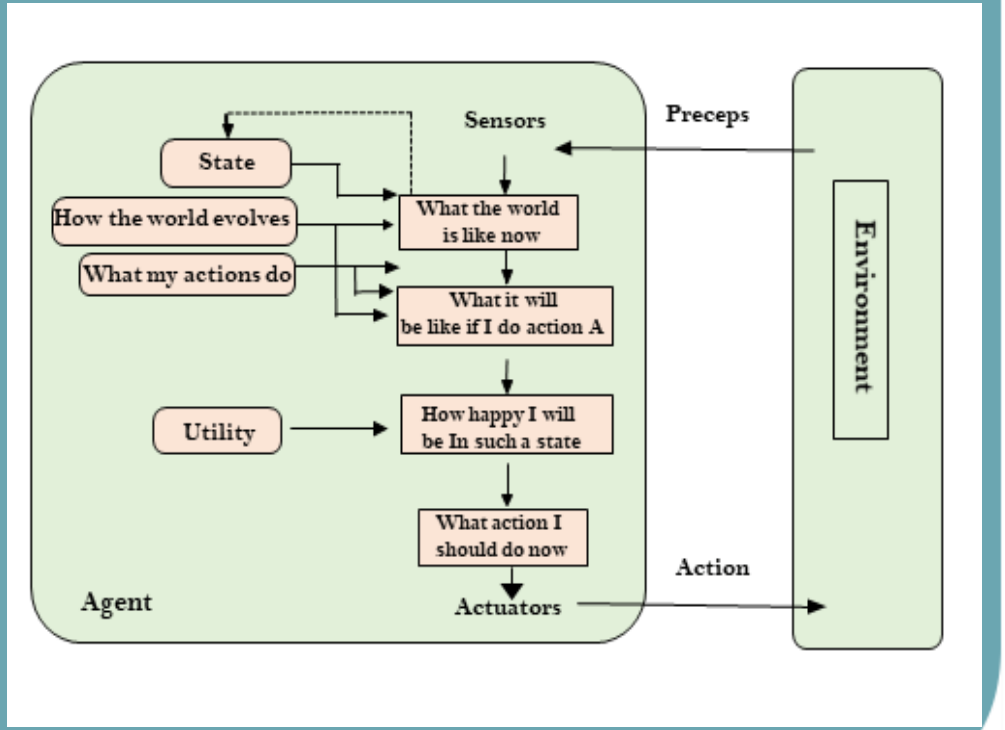
Goal-Based Agents

- These kinds of agents take decisions based on how far they are currently from their goal(description of desirable situations).
- Their every action is intended to reduce its distance from the goal.
- This allows the agent a way to choose among multiple possibilities, selecting the one which reaches a goal state.
- The knowledge that supports its decisions is represented explicitly and can be modified, which makes these agents more flexible.
- They usually require search and planning. The goal-based agent's behavior can easily be changed.



Utility-Based Agents

- These agents are similar to the goal-based agent but provide an extra component of utility measurement which makes them different by providing a measure of success at a given state.
- Utility-based agent act based not only goals but also the best way to achieve the goal.
- The Utility-based agent is useful when there are multiple possible alternatives, and an agent has to choose in order to perform the best action.
- The utility function maps each state to a real number to check how efficiently each action achieves the goals.



Learning Agents

- A learning agent in AI is the type of agent which can learn from its past experiences, or it has learning capabilities.
- It starts to act with basic knowledge and then able to act and adapt automatically through learning.
- A learning agent has mainly four conceptual components, which are:
  - **Learning element:** It is responsible for making improvements by learning from environment
  - **Critic:** Learning element takes feedback from critic which describes that how well the agent is doing with respect to a fixed performance standard.
  - **Performance element:** It is responsible for selecting external action
  - **Problem generator:** This component is responsible for suggesting actions that will lead to new and informative experiences.
- Hence, learning agents are able to learn, analyze performance, and look for new ways to improve the performance.