

Knowledge Management

Prepared by Dr. Mohamed Eassa

IT 383
Knowledge Management

Lecture 8



Knowledge Engineering

Lecture 8

IT 383

Knowledge Management

Prepared by
Dr. Mohamed Eassa



Knowledge Engineering



What is knowledge Engineering?

A set of intensive activities involved in building, maintaining and using knowledge-based systems.

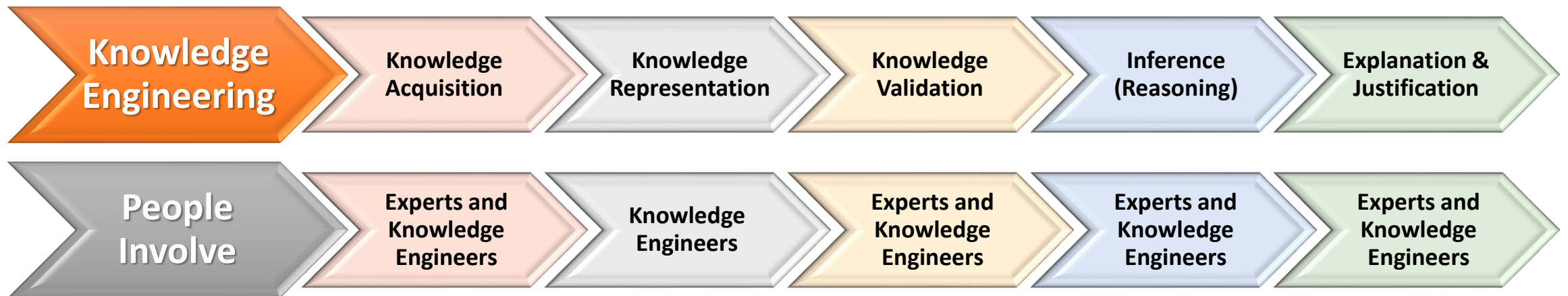


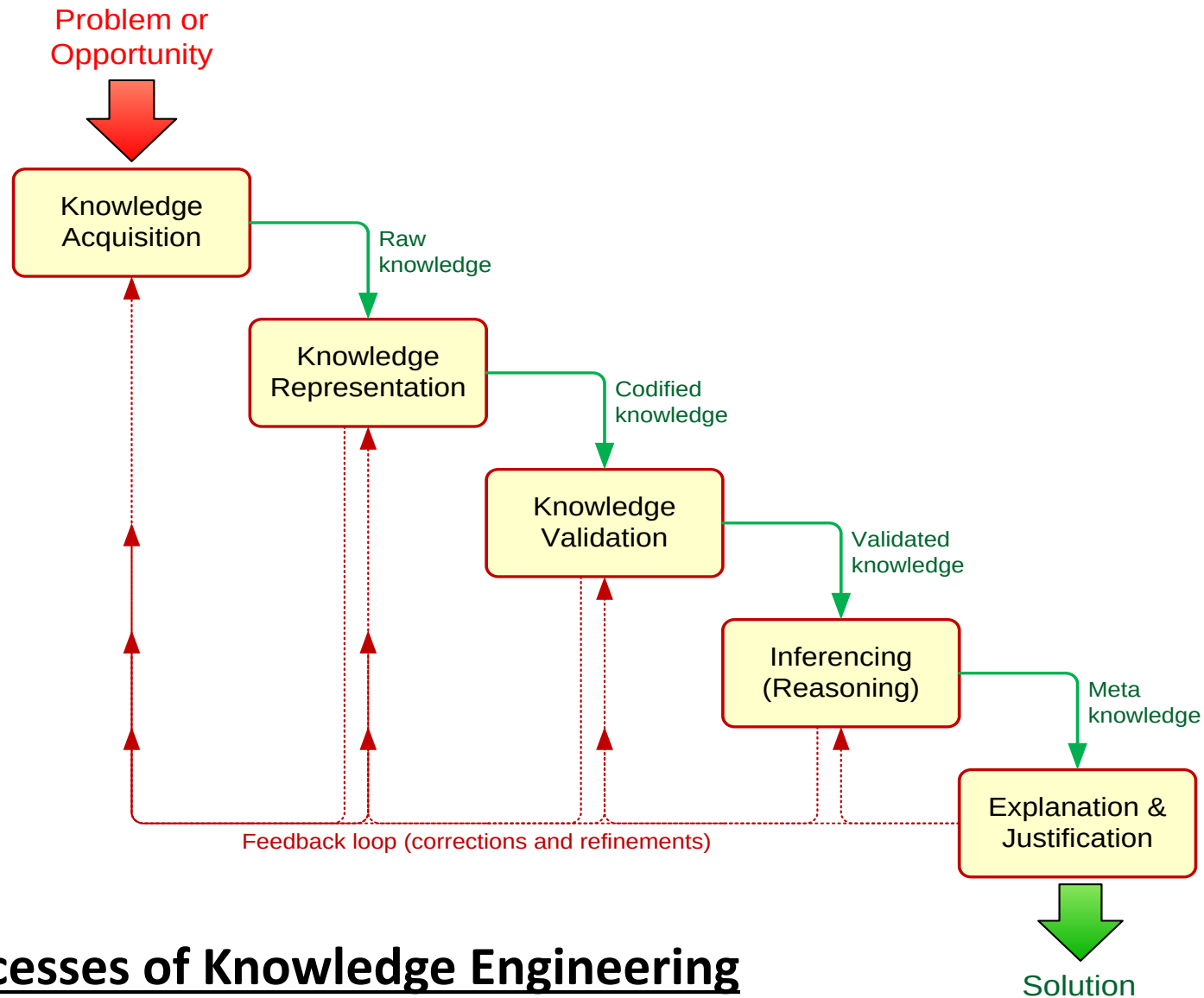
Lecture Objectives

- **Identify Knowledge Engineering Activities (Total View).**
- **Understanding Knowledge Acquisition.**
- **Understanding Knowledge Representation.**
- **Understanding Knowledge Validation.**
- **Understanding Knowledge Inference.**
- **Understanding Knowledge Explanation & Justification.**

➤ What is knowledge Engineering?

A set of intensive activities involved in building, maintaining and using knowledge-based systems.







Knowledge Acquisition



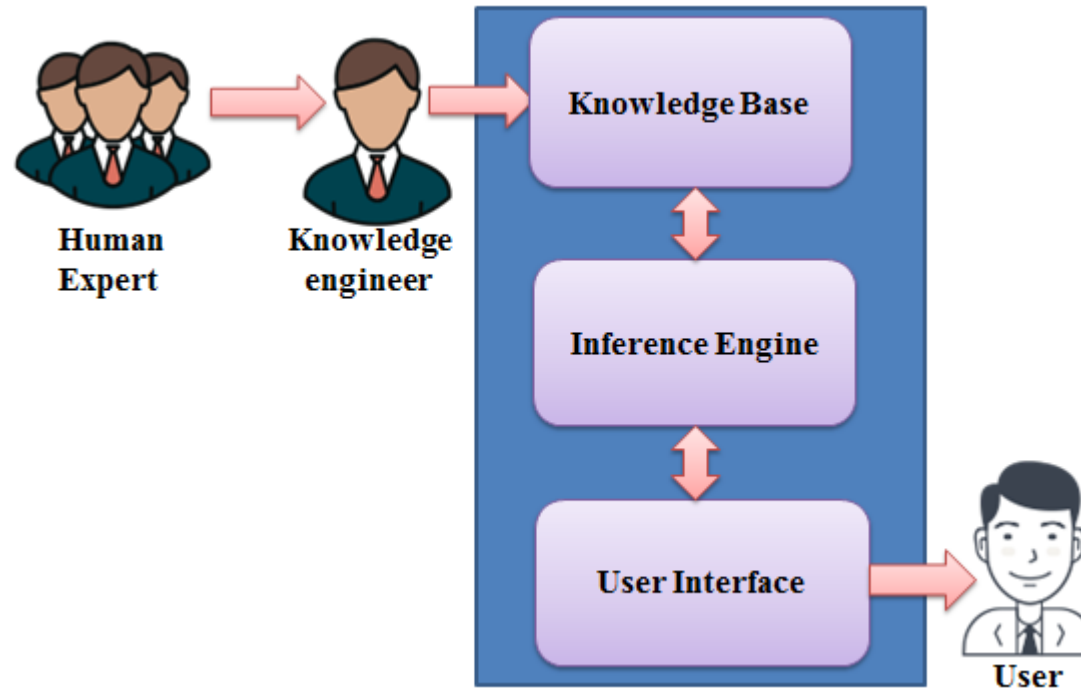
knowledge Acquisition

The tools and techniques used to capture knowledge from human experts (and other information sources).

Knowledge Acquisition

➤ knowledge Acquisition

The tools and techniques used to capture knowledge from human experts (and other information sources).



➤ Role of the Knowledge Engineer:

- Identifies the problem domain.
- Captures the knowledge, he/she should be familiar with knowledge capture tools, conceptual thinker, tolerance for ambiguity, ability to work with other professionals and experts.
- Codified the knowledge.
- Validate the knowledge along with the experts.
- Coordinates the entire project from beginning to end. He/she can make or break the project.

➤ Roles of selecting the experts:

- **They Have the Experience**, their experiences and knowledge set them apart and enable them to serve as resources to others within the organization.
- **They Understand User Needs**, they understand the problem domain and can offer the solutions.
- **They Implement Best Practices**, their passion motivates them to reexamine best practices and shape new techniques.
- **They are Proactive**, their experiences enable others to learn and produce results quickly and accurately.

Knowledge Acquisition

➤ Knowledge acquisition tools:

➤ Interview:

- Commonly used in the **early stages of tacit knowledge capture** and verify its validity (Validity of a given question means its logically correct and is understood the same way by different people)
- Interviewing as a tool requires **training and preparation**, “The knowledge engineer must be well **trained to observe and analyze motions, facial movements, body language**, etc. and **create a friendly, non-threatening atmosphere** and reveal any fear or criticism **to experts**.”

➤ Types of Interviews:

➤ Structured Interviews:

Questions and responses are definitive. Used when specific information is wanted, varieties of structured questions:

➤ Semi-structured Interviews:

Predefined questions are asked but allow expert some freedom in expressing the answers.

➤ Unstructured Interviews:

Neither the questions nor their responses specified in advance. Used when exploring an issue.

➤ Guide to a Successful Interview:

➤ Set the stage and establish rapport.

➤ Properly phrase the questions, Question construction is important.

➤ Listen closely and avoid arguments.

➤ Evaluate session outcomes.

➤ Things to Avoid:

- Taping a session without advance permission from the expert.
- Converting the interview into an interrogation.
- Interrupting the expert.
- Asking questions that put the domain expert on the defensive.

- **Losing control of the session.**
- **Pretending to understand an explanation when you actually don't.**
- **Bring items not on the agenda.**
- **Try to be specific, rather than general.**
- **Stay away from using technical terms just to show off.**

➤ On-Site Observation:

- Process of observing, interpreting, and recording problem-solving behavior while it takes place by experts.
- In addition, the knowledge engineer asks the expert questions about the problem solving process.
- The protocol of observation is more listening than talking.
- Avoid giving advices to expert while observing.

- In comparison to the interview, observation places the knowledge engineer closer to the actual process, steps, and procedures used to solve the domain problem.
- The problem here is that some experts don't like to be observed. Experts fear of 'giving away' their experience in a quick look.

➤ Brainstorming:

- Unstructured approach to generating ideas about a problem; invites two or more experts into a session in which discussion are carried out.
- The primary goal of brain storming is to think up creative solutions to problems.
- All possible solutions are considered equally and everything is valued.
- Questions can be raised for clarification, but no evaluation is made at the moment.
- Idea generation, followed by idea evaluation.

➤ Protocol Analysis:

- Think-aloud method; Expert keeps talking, speaking out loud while solving a problem.
- Unlike other techniques, knowledge engineer never interrupt or ask questions.
- Provides wealth of information that is very useful to knowledge capture and representation.



Knowledge Representation

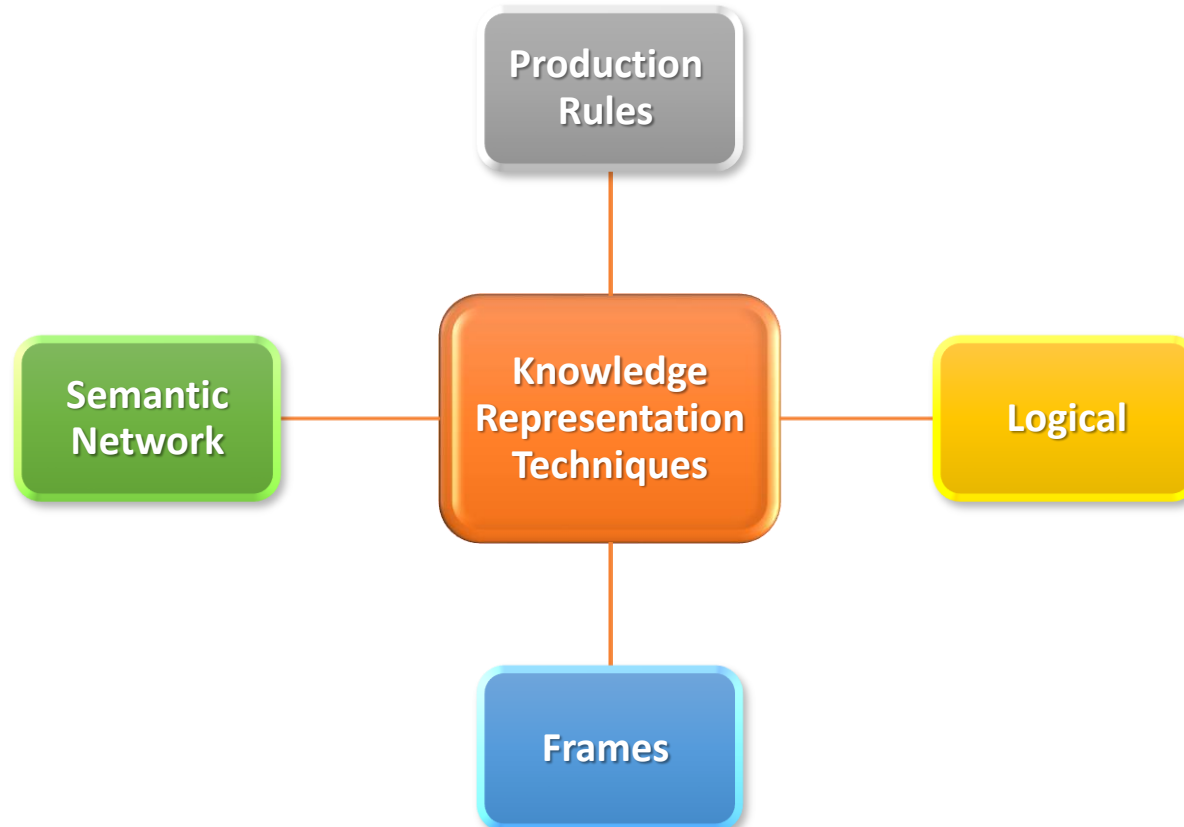


knowledge Representation

Representing the captured knowledge in the form that can be understood by computers then utilized this knowledge to solve the complex problems.

➤ What is knowledge Representation?

Representing the captured knowledge in the form that can be understood by computers then utilized this knowledge to solve the complex problems.



Production Rules

➤ Production Rules:

A set of **if-then rules** - typically state that if certain conditions hold, then some action should be taken.

➤ The rule's structure connects one or more **premises** contained in the **IF part**, to one or more **conclusions** contained in the **THEN part**.

➤ A rule is a form of procedural knowledge.

➤ A rule describes how to solve a problem.

Production Rules

- A rule may have multiple premises joined with **AND** statements, **OR** statements, **NOT** statements or a combination of all.
- In a rule based expert system, domain knowledge is captured in a set of rules and entered in the system's knowledge base.
- the system uses these rules along with information contained in the working memory to solve a problem.
- When the IF portion of the rule matches the information contained in the working memory, the system performs the actions specified in the THEN part of the rule.

Production Rules

- When this happened, the rule fires and its THEN statements are added to the working memory.
- The new statements added to the working memory can also cause other rules to fire.

Suppose we have **three facts**: **Raining**, **Outside(Omar)**, $\neg$ **HasUmbrella(Omar)**.

And we have a rule in KB: **IF Raining $\wedge$ Outside(x) $\wedge$ $\neg$ HasUmbrella(x) THEN GetWet(x)**

If **x=Omar** matches the facts, so we are able to infer **GetsWet(Omar)**:

IF Raining $\wedge$ Outside(Omar) $\wedge$ $\neg$ HasUmbrella(Omar) THEN GetWet(Omar)

This means we now have **four facts** in our knowledge base:

Raining, **Outside(Omar)**, $\neg$ **HasUmbrella(Omar)**, and **GetWet(Omar)**

$\neg A$	NOT A (negation)
$A \wedge B$	A AND B (Conjunction)
$A \vee B$	A OR B (Disjunction)
$A \rightarrow B$	If A then B "A implies B" (Implication)
$A \leftrightarrow B$	A if and only if B (Equivalence)
$\exists A$	There exist A
$\forall A$	For all A (universal quantifier)

Logical Representation

➤ **Logical Representation:**

Logical representation is a language with some concrete rules that has no ambiguity in representation. It consists of precisely defined sentences. Each sentence can be translated into logics using syntax and semantics.

First-Order Logic

Propositional Logic

Predicate Logic

Logical Representation

➤ Basics:

$\neg A$	NOT A (negation)
$A \wedge B$	A AND B (Conjunction)
$A \vee B$	A OR B (Disjunction)
$A \rightarrow B$	If A then B “A implies B” (Implication)
$A \leftrightarrow B$	A if and only if B (Equivalence)
$\exists A$	There exist A
$\forall A$	For all A (universal quantifier)

Logical Representation

➤ Propositional logic:

It is the branch of logic that studies ways of joining and/or modifying entire statements or sentences to form more complicated statements or sentences, as well as the logical relationships and properties that are derived from these methods of combining or altering statements.

Example:

A: It is cold

B: It is snowing

C: It is winter

D: It is sunny

To represent the following rules using propositional logic;

R1: “if it is snowing then it is cold”

R2: “if it is cold and it is snowing then it is not sunny”

R1: $B \rightarrow A$

R2: $A \wedge B \rightarrow \neg D$

Logical Representation

➤ Predicate logic:

It is a formal system that uses variables and quantifiers ($\forall, \exists$) to formulate propositions.

Example 1:

1- All humans are mortal.

2- Omar is a human. (Nada is a human, Ahmed is a human,...etc.)

Therefore;

3- Omar is mortal

To represent the previous statements using predicate logic;

(Omar, Nada, and Ahmed are variables, may represent by variable "X"), so,

$\forall X [is_a_human(X) \rightarrow is_mortal(X)]$

Where X is a variable = (Omar, Nada, or Ahmed)

Frame Representation

➤ Frame Representation:

A data structural for representing all knowledge about a particular object.

Content of the frame:

- ❖ It has a name
- ❖ Slots: a set of attributes
- ❖ Facets:
 - attribute values
 - Default
 - Range
 - If needed
 - If changed

Frame Name

Bird

Slots:

color	unknown
Eats	Worms
No_Wings	2
Flies	True
Hungry	Unknown
Activity	Unknown

Semantic Network Representation

➤ Semantic Networks Representation:

Provides a graphical view of a problem's important objects, properties and relationships.

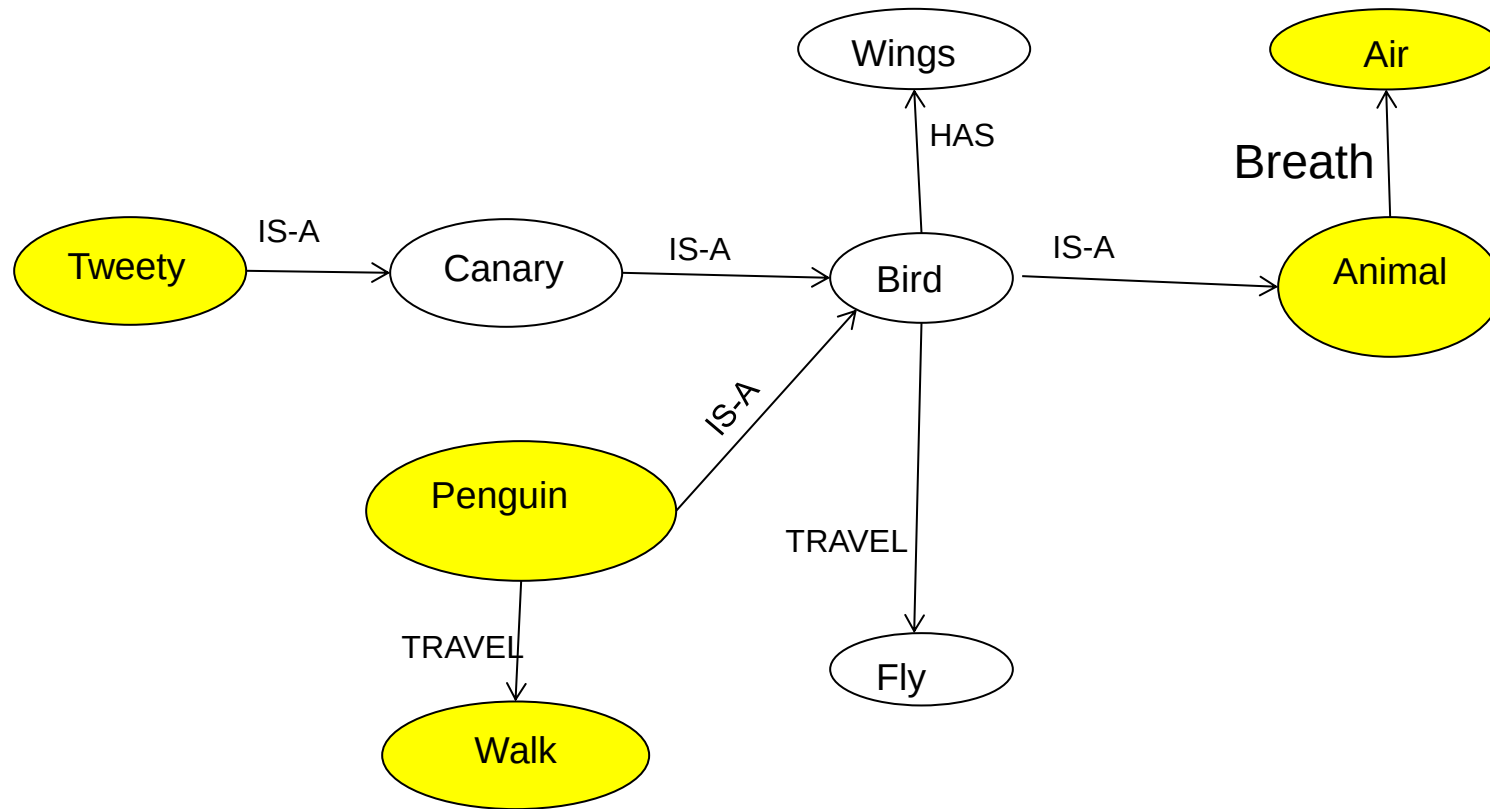
➤ It contains nodes and arcs that connect the nodes that represent the relationships between the nodes.

➤ The node can represent:

- Objects
- Object properties
- Property value

➤ Both the nodes and arcs have labels that clearly describe the objects represented and their natural relationships.

Semantic Network Representation



➤ **Ontology:**

It is a data model that represents knowledge as a set of concepts within a domain and the relationship between these concepts.

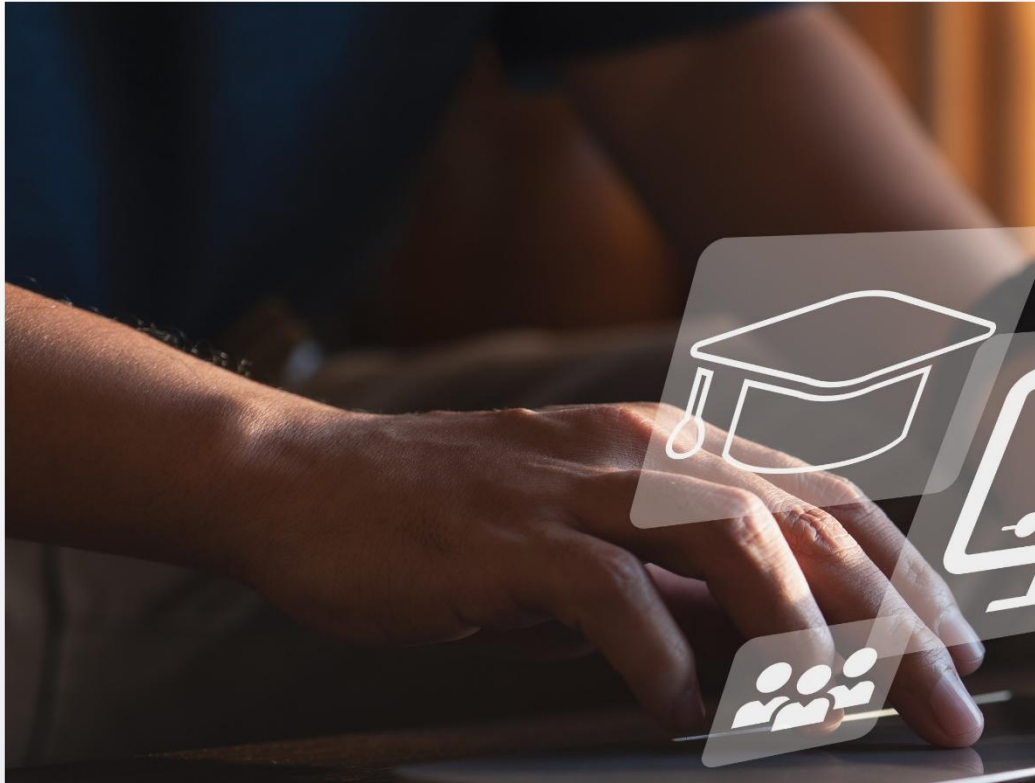
The main components of an ontology are:

Concepts: types of classes, objects, or kinds of things.

Relations: ways in which classes and individuals can be related to one another

Axioms: including rules in a logical form that together comprise the overall theory that the ontology describes in its domain of application.

Ontologies are commonly encoded using ontology languages (OWL, XML,...).



Knowledge Validation

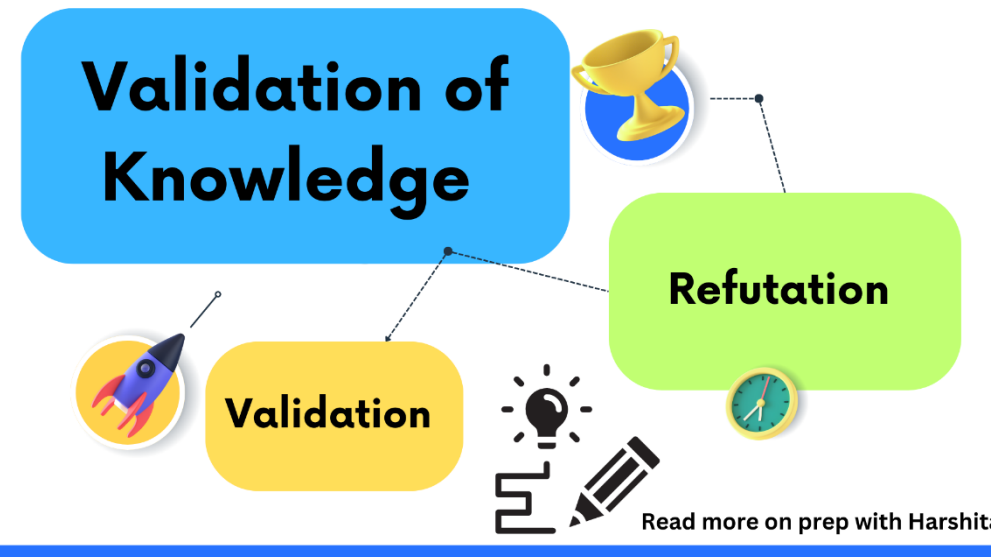


knowledge Validation

Knowledge validation is the process of ensuring that knowledge is accurate, credible, and relevant for decision-making and problem-solving. It involves evaluating the quality and correctness of knowledge within an organization or system.

➤ What is knowledge Validation?

Knowledge validation is the process of ensuring that knowledge is accurate, credible, and relevant for decision-making and problem-solving. It involves evaluating the quality and correctness of knowledge within an organization or system.



Why Knowledge Validation is Important

Accuracy: Inaccurate knowledge can lead to poor decisions, mistakes, and inefficiencies.

Reliability: Validated knowledge ensures that stakeholders can depend on it when making strategic or operational decisions.

Why Knowledge Validation is Important

Trust: Employees, stakeholders, and users must trust the knowledge provided for it to be used effectively in daily operations.

Continuous Improvement: Knowledge validation is an ongoing process that fosters continual improvement within an organization or community.

Challenges in Knowledge Validation

Challenges in Knowledge Validation

Volume of Information: The sheer amount of knowledge available in modern organizations makes it difficult to manually validate all of it.

Evolving Knowledge: As knowledge and best practices change over time, ensuring that everything stays up-to-date can be complex.

Subjectivity: In some fields, especially in areas like social sciences or management, validating knowledge can be more subjective, as different interpretations may be valid.

Challenges in Knowledge Validation

Challenges in Knowledge Validation

Expert Availability: Not all knowledge areas have readily available experts, or experts may be scarce, which hinders the validation process.

Bias and Error: Knowledge validation systems may themselves be biased, leading to flawed conclusions.



Inference



Inference

Inferencing is the process of using the rules in knowledge base along with the known facts to draw conclusions. **Inferencing requires a logic embedded in the computer program to access the stored knowledge and executes the inferencing rules, this program is called Inference Engine.**

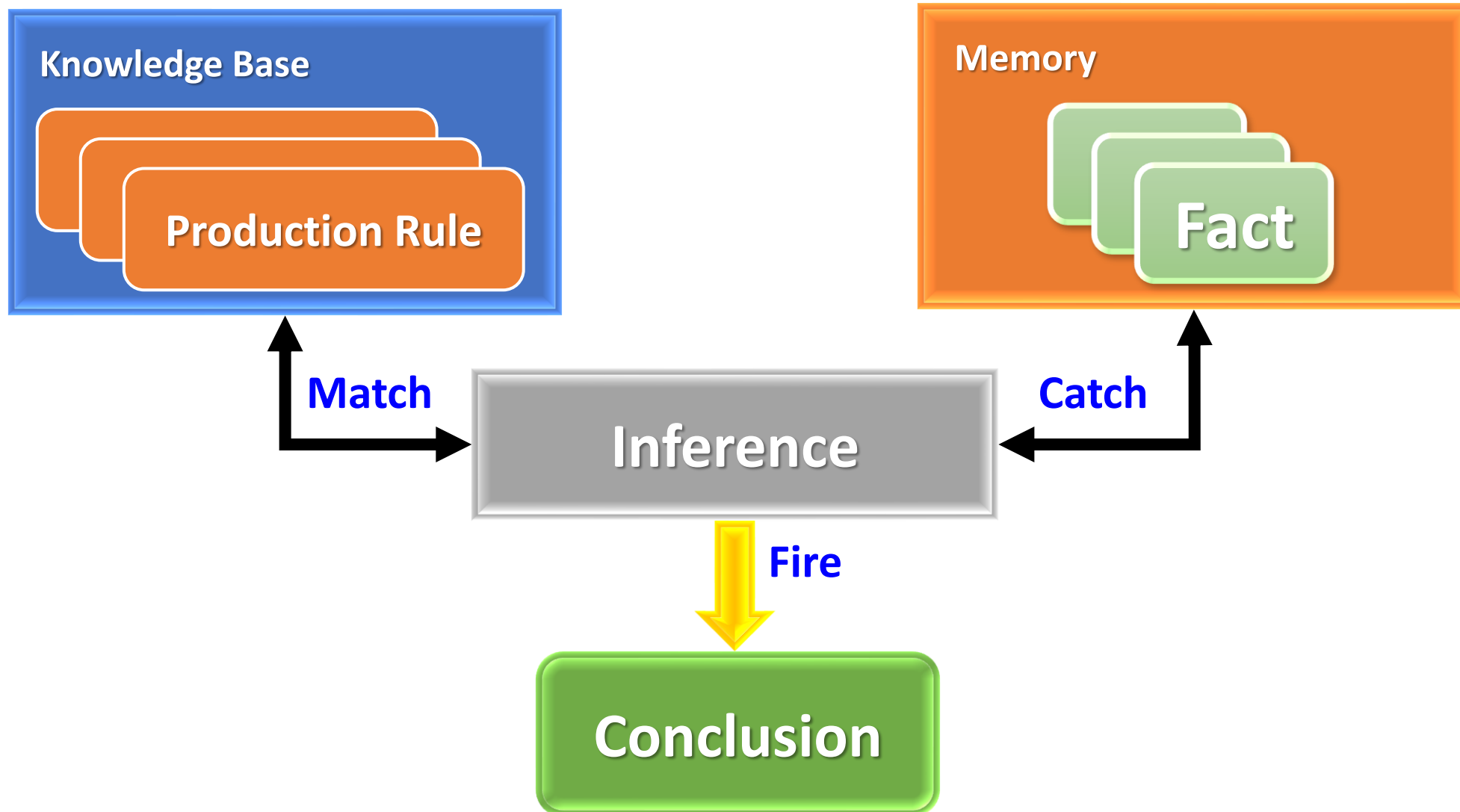
➤ What is inference?

Inferencing is the process of using the rules in knowledge base along with the known facts to draw conclusions. Inferencing requires a logic embedded in the computer program to access the stored knowledge and executes the inferencing rules, this program is called Inference Engine.

➤ The inference engine checks every rule in the knowledge base to identify those that can be fired based on the collection of known facts and keep doing so until the goal is achieved.

➤ The most popular mechanisms for rule-based system are forward and backward chain.

Inference



➤ **Forward chain:**

A data-driven approach, starting from available information (facts) then trying to draw conclusions .

➤ **Backward chain:**

A goal-driven approach, starting from the expectation of what is going to happen then seeking evidence that support that expectation.

➤ **Uncertainty:**

Uncertainty is defined as the lack of the exact knowledge that would enable us to reach a perfectly reliable conclusion.

➤ **Information can be incomplete, inconsistent, uncertain, or all three. In other words, information is often unsuitable for solving a problem.**

➤ **Classical logic permits only exact reasoning.**

➤ **Unknown data:**

When the data is incomplete or missing, the only solution is to accept the value “unknown” and proceed to an approximate reasoning with this value.

➤ **Combining the views of different experts:**

Unfortunately, experts often have contradictory opinions and produce conflicting rules. To resolve the conflict, the knowledge engineer has to attach a weight to each expert and then calculate the composite conclusion. But no systematic method exists to obtain these weights.

➤ **Weak implications:**

Domain experts and knowledge engineers have the painful task of establishing concrete correlations between IF (condition) and THEN (action) parts of the rules.

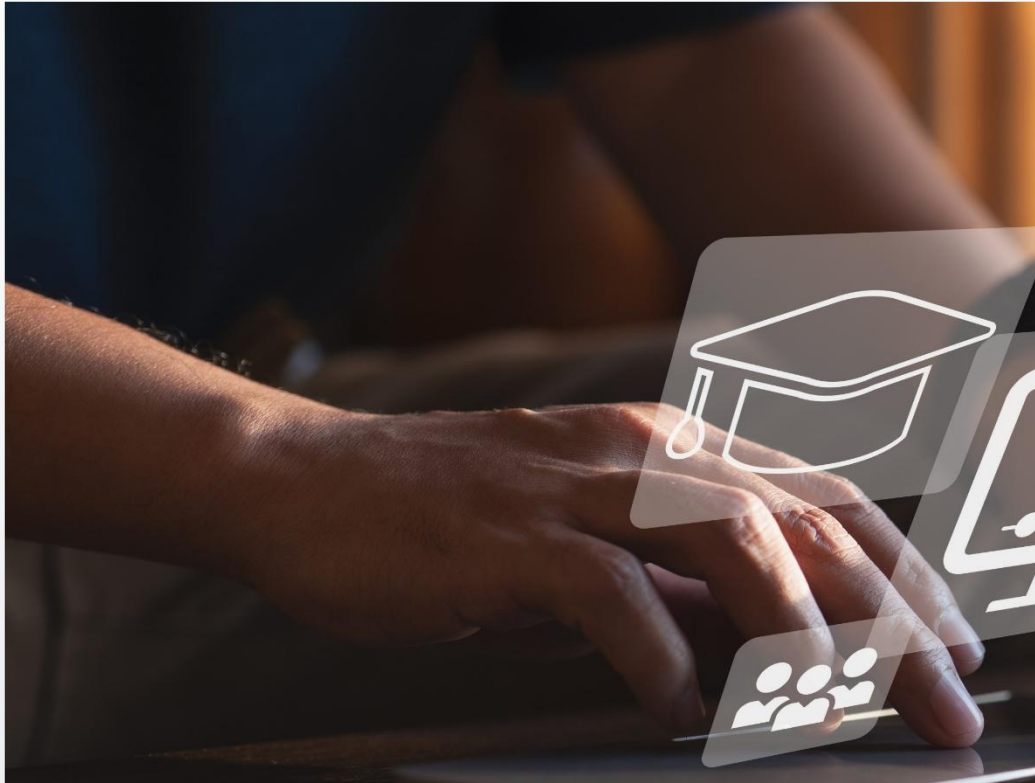
➤ **Certainty factor:**

Relies on judgmental measures of belief. Uses measure of belief (MB) and measure of disbelief (MD), **$Cf = MB - MD$** , and take on values from **-1 to 1**.

$$CF(\text{Rule}) = CF(\text{Then})/CF(\text{IF})$$

R1= IF A THEN B.

$$CF(\text{B})= CF(\text{R1}) \times CF(\text{A})$$



Explanation & Justification

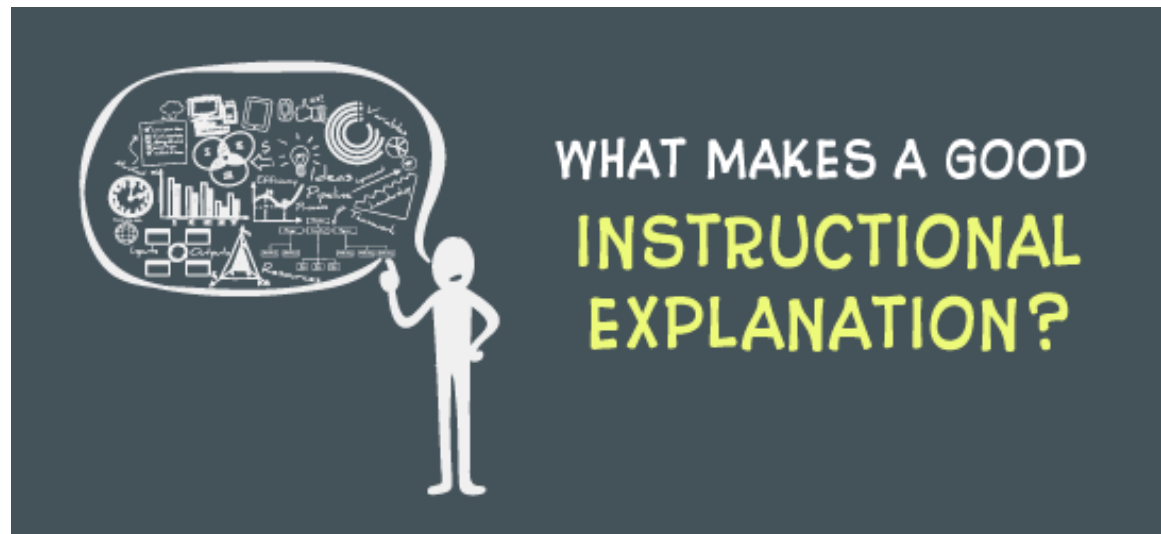


Explanation & Justification

Justification in Knowledge Engineering (KE) is the process of providing structured reasoning for decisions made by a system. Unlike **Explanation**, which focuses on "how" a decision was made, justification emphasizes "why" the decision is valid.

➤ Explanation in Knowledge Engineering:

Explanation is the process of clarifying how a system arrived at a particular conclusion or decision. It helps users interpret and validate the reasoning of the system.



Tools and techniques for Explanation in Knowledge Engineering

Rule-Based Explanation

Rule-based explanation techniques are widely used in knowledge-based systems (KBS) to provide transparent and interpretable reasoning. These techniques use predefined rules (if-then statements) to explain decisions systematically.

Rule-based explanation techniques rely on explicit logic and knowledge representations to describe how a system reaches conclusions. These systems consist of rules, facts, and an inference engine that applies logical reasoning to derive results.

These techniques These use explicit rules to explain the reasoning process.

➤ Justification in Knowledge Engineering:

Justification in Knowledge Engineering (KE) is the process of providing structured reasoning for decisions made by a system. Unlike explanation, which focuses on "how" a decision was made, justification emphasizes "why" the decision is valid.

Justification provides a deeper, more structured reasoning for a decision, often including the sources of knowledge used. While explanations focus on the "how," justifications focus on the "why" behind a decision.

Tools and techniques for Justification in Knowledge Engineering

Case-Based Reasoning (CBR) is a problem-solving approach in knowledge engineering where **past cases** (previous experiences) are used to solve new problems. Instead of relying on predefined rules like rule-based systems, CBR retrieves and adapts **similar past cases** to justify decisions.

Case-Based Reasoning (CBR)

CBR explains decisions by comparing them with past cases.

Example: A **medical diagnosis** system may diagnose a new patient by comparing their symptoms with past patient cases stored in a knowledge base.

Summary

- **knowledge Engineering Definition.**
- **Understanding the knowledge Acquisition.**
- **Understanding The roles of knowledge engineer.**
- **Understanding The roles of experts.**
- **Understanding the knowledge Representation.**
- **Understanding the knowledge Validation.**
- **Understanding the knowledge Inference (Reasoning).**
- **Understanding the knowledge Explanation & Justification.**



Thank You