Chapter 4: Knowledge Application Systems: Systems that Utilize Knowledge

- **Knowledge Application Systems** support the process through which some individuals utilize **knowledge** possessed by other individuals without actually acquiring, or learning, that **knowledge**.
- **The Systems** apply knowledge to solve specific problems.
- are typically enabled by intelligent technologies.
- knowledge application depends on direction and routines.
- Both mechanisms and technologies can support knowledge application systems by facilitating the knowledge management processes of routines and direction.
- Mechanisms and technologies can facilitate knowledge application through direction and routines either within or across organizations.
- **Knowledge application** technologies, which support direction and routines includes: fault diagnosis (or troubleshooting) **systems**.
- Mechanisms facilitating **direction** include hierarchical relationships, help desks, and support centers; whereas mechanisms facilitating **routines** include organizational policies, work practices, and standards.
- Technologies supporting direction and routines include expert systems, decision support, advisor systems, fault diagnosis (or troubleshooting) systems, and help desk systems.
- These technologies may support direction, as in the case of a field service technician seeking to troubleshoot a particular product; or may support routines, as in the case of a customer service representative who may need to identify alternative product delivery mechanisms while preparing the shipment of an order.

Technologies for Applying Knowledge

Artificial Intelligence(AI)

- **AI** refers to enabling computers to perform tasks that resemble human thinking ability.
- It is the area of computer science that deals with the design and development of computer systems that exhibit human-like cognitive capabilities.
- Much like KM and human intelligence, AI is associated with knowledge.
- Definitions for AI range from: systems that act like humans, systems that think like humans, systems that think rationally, to systems that act rationally.
- Systems that act like humans refer to those that pass the Turing Test, which refers to a computer passing a test by a human interrogator, who cannot tell whether the responses came from a person or not. Systems that think like humans refer to a computer program whose input to output behavior matches those of humans, for example when solving problems, like playing chess or performing a medical diagnosis.

- Systems that think rationally refer to those that follow a specific logic to solve a problem.
- Finally, systems that act rationally refer to those computer agents that are expected to have specific characteristics that enable them to operate autonomously within their environments, and even adapt to change in the face of uncertainty.

Rule-Based Systems

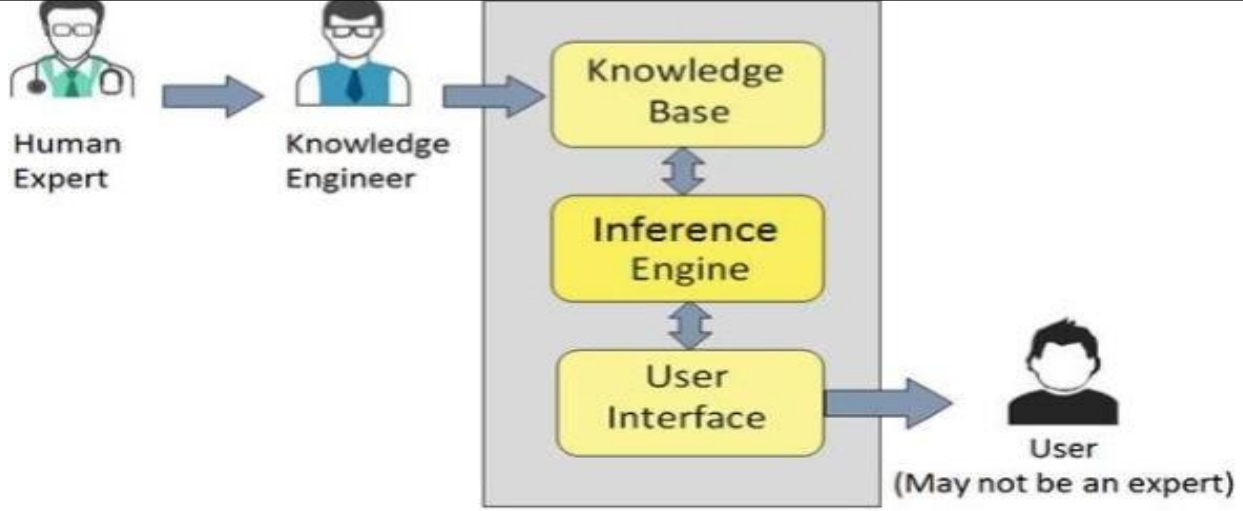
- A **rule**-**based system** is a set of "if-then" statements that uses a set of assertions, to which **rules** on how to act upon those assertions **are** created. In software development, **rule**-**based systems can** be used to create software that **will** provide an answer to a problem in place of a human expert.
- A classic **example** of a **rule**-**based system** is the domain-specific expert **system** that uses **rules** to make deductions or choices. For **example**, an expert **system** might help a doctor choose the correct diagnosis **based** on a cluster of symptoms, or select tactical moves to play a game.
- Traditionally, the development of knowledge-based systems had been based on the use of rules or models to represent the **domain knowledge**.
- The development of such systems requires the collaboration of a subject matter expert with a knowledge engineer, the latter being responsible for the elicitation and representation of the expert's knowledge.
- The process of developing knowledge application systems requires eliciting the knowledge from the expert and representing it a form that is usable by computers.
- This process is called **knowledge engineering**. Knowledge engineers typically build knowledge application systems by first interviewing in detail the **domain expert** and representing the knowledge more commonly in a set of heuristics, or rules-of-thumb.

- Experts develop these rules-of-thumb over years of practical experience at solving problems. In order for the computer to understand these rules-of-thumb, we represent them as *production rules* or *IF-THEN* statements. For example: *IF* the number of employees is less than 500, *THEN* the firm is a small business
- Applicable when the domain knowledge can be defined by a manageable set of rules or heuristics.
- Rules are the most commonly used knowledge representation paradigm, perhaps due to their intuitive implementation.
- The IF portion is the *condition* (also *premise* or *antecedent*), which tests the truth-value of a set of assertions.
- If the statement is true, the THEN part of the rule (also *action, conclusion,* or *consequence*) is also inferred as a fact.
- In addition to rules, other paradigms to represent knowledge include **frames**, predicates, associative networks, and objects. Rule-based systems have posed some disadvantages.
- One is that in many circumstances, the number of rules that may be needed to properly represent the domain may be quite large.
- Expert systems with such a large number of rules offer many disadvantages, namely

(1) difficulty in coding, verifying, validating, and maintaining the rules; and

(2) reduction in the efficiency of the inference engine executing the rules.

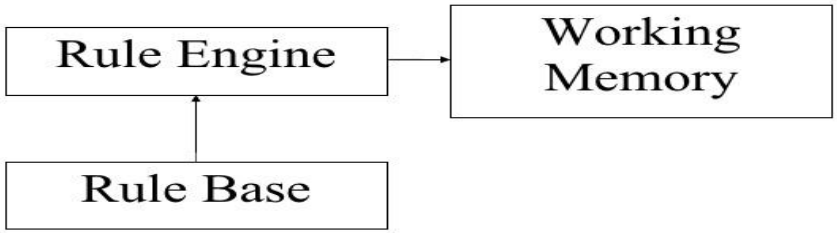
- As an alternative, we consider the use of cases as a method to represent knowledge.



Difference between artificial intelligence and expert systems

OrangeScape
the origin of innovation

Components of a rule engine



[Rule Based System]

Case-Based Reasoning (CBR)Systems

- **Case-based reasoning** (**CBR**) is the process of solving new problems based on the solutions of similar past problems.
- It is an experience-based approach to solving new problems by adapting previously successful solutions to similar problems.
- **It** is an artificial intelligence technique designed to mimic human problem solving.
- Addressing memory, learning, planning and problem solving, CBR provides a foundation for a new technology of intelligent computer systems that can solve problems and adapt to new situations.
- In CBR, the "intelligent" reuse of knowledge from already-solved problems, or cases, relies on the premise that the more similar two problems are, the more similar their solutions will be.
- Its goal is to mimic the way humans solve problems. When faced with a new problem, humans search their memories for past problems resembling the current problem and adapt the prior solution to "fit" the current problem.
- It is is based on the paradigm of human thought in cognitive psychology that contends that human experts derive their knowledge from solving numerous cases in their problem domain.

- CBR is a method of analogical reasoning that utilizes old cases or experiences in an effort to solve problems, critique solutions, explain anomalous situations, or interpret situations.
- Case-based reasoning is a prominent type of analogy solution making.
- An auto mechanic who fixes an engine by recalling another car that exhibited similar symptoms is using case-based reasoning.
- Applicable in weak-theory domains, that is, where an expert either doesn't exist or does not fully understand the domain. Also applicable if the experience base spans an entire organization, rather than a single individual.

Case-based reasoning has been formalized for purposes of computer reasoning as a four-step process:

- **Retrieve:** Given a target problem, retrieve from memory cases relevant to solving it. A case consists of a problem, its solution, and, typically, annotations about how the solution was derived. For example, suppose Fred wants to prepare blueberry pancakes. Being a novice cook, the most relevant experience he can recall is one in which he successfully made plain pancakes. The procedure he followed for making the plain pancakes, together with justifications for decisions made along the way, constitutes Fred's retrieved case.

- **Reuse:** Map the solution from the previous case to the target problem. This may involve adapting the solution as needed to fit the new situation. In the pancake example, Fred must adapt his retrieved solution to include the addition of blueberries.
- **Revise:** Having mapped the previous solution to the target situation, test the new solution in the real world (or a simulation) and, if necessary, revise. Suppose Fred adapted his pancake solution by adding blueberries to the batter. After mixing, he discovers that the batter has turned blue – an undesired effect. This suggests the following revision: delay the addition of blueberries until after the batter has been ladled into the pan.
- **Retain:** After the solution has been successfully adapted to the target problem, store the resulting experience as a new case in memory. Fred, accordingly, records his new-found procedure for making blueberry pancakes, thereby enriching his set of stored experiences, and better preparing him for future pancake-making demands.
- CBR developed its roots in the work of artificial intelligence theorist and cognitive psychologist, Roger Schank, and his students at Yale in the late 20th century. The researchers studied the problem-solving ability of humans and found that most people assemble solutions based on earlier experiences with similar situations.

- With applications spanning fields ranging from machine learning to medicine to law, CBR is accomplished by gathering case histories and implemented by identifying significant features that describe a case. CBR systems can “learn” by acquiring new knowledge as cases. This, along with the application of database techniques, makes it easier to maintain large volumes of information.

Four step process for CBR

- In general, the case-based reasoning process entails:
- Retrieve- Gathering from memory an experience closest to the current problem.
- Reuse- Suggesting a solution based on the experience and adapting it to meet the demands of the new situation.
- Revise- Evaluating the use of the solution in the new context.
- Retain- Storing this new problem-solving method in the memory system.

A typical case-based knowledge application system will consist of the following processes:

1. Search the case library for similar cases. This implies utilizing a search engine that examines only the appropriate cases and not the entire case library, as it may be quite large.

2. Select and retrieve the most similar case(s). New problems are solved by first retrieving previously experienced cases. This implies having a means to compare each examined case to the current problem, quantifying their similarity, and somehow ranking them in decreasing order of similarity.

3. Adapt the solution for the most similar case. If the current problem and the most similar case are not similar enough, then the solution may have to be adapted to fit the needs of the current problem. The new problem will be solved with the aid of an old solution that has been adapted to the new problem.

4. Apply the generated solution and obtain feedback. Once a solution or classification is generated by the system, it must be applied to the problem. Its effect on the problem is fed back to the CBR system for classification of its solution (as success or failure).

5. Add the newly solved problem to the case library. The new experience is likely to be useful in future problem solving. This step requires identifying if the new case is worth adding to the library and placing it in the appropriate location in the case library

There are several variants of CBR, such as **exemplar-based reasoning**, **instance based reasoning**, and **analogy-based reasoning**.