Components of KBS

We have to build systems with specific characteristics:

- Problem solving using symbolic information
- Resolution by mean of reasoning and heuristic methods
- Capacity to explain the resolution process
- Interactivity (with a user/the environment)
- Able to adapt to the environment

A basic set of components is needed

- Reasoning subsystem
  - Knowledge storage subsystem (Knowledge base)
  - Knowledge use and interpretation subsystem (solving mechanism)
- Problem state storage subsystem (working memory)
- Explanation and resolution inspection subsystem
- Communication and interface subsystem
- Learning subsystem

KBS based on production systems

- Problems are solved using the reasoning mechanism of an inference engine
- The domain knowledge is described by means of an ontology
- The problem solving knowledge is described using production rules or an equivalent formalism
Knowledge storage subsystem

- It will store all the knowledge needed to solve problems in the domain of application.
- We can find three kinds of knowledge:
  - **Factual knowledge** (domain objects and their characteristics)
  - **Relational knowledge** (relations among the domain objects)
  - **Conditional knowledge** (Deductive knowledge about the problem)
- The two first kind of knowledge are described in the domain ontology.
- The third one represents the knowledge related to the resolution of the problem and is described by production rules.

**Rule modules**:
- Eases the development and maintaining of the system.
- Increases the efficiency of the reasoning process.
- Allows to implement strategies for the use of the knowledge (meta-knowledge, meta-rules).

Knowledge storage subsystem: Rules

- Conditional knowledge includes:
  - **Deductive knowledge** (structural): Describes the process of problem solving as a chain of deductions.
  - **Knowledge about goals** (strategic): Drives the resolution process to the solution.
  - **Causal knowledge** (supportive): Supports explanation of the problem resolution process.
- **Rule modules**: Sets of related rules.
  - Eases the development and maintaining of the system.
  - Increases the efficiency of the reasoning process.
  - Allows to implement strategies for the use of the knowledge (meta-knowledge, meta-rules).

Knowledge storage subsystem: Meta-Rules

- They describe high level knowledge about the process of solving the problem.
- They allow to take control of the resolution process:
  - Activating and deactivating rules/modules.
  - Deciding the application order of rules/modules.
  - Deciding resolution strategies, handling of exceptions, uncertainty, ...
- This knowledge is more difficult to obtain from the experts.
Knowledge use and interpretation subsystem

- Usually is an inference engine
- It applies its execution cycle to solve the problem
  - Detection of applicable rules
  - Selection of the best rule (Domain independent strategy or driven by metaknowledge)
  - Execution of the rule

Problem state storage subsystem

- Stores the problem initial facts and all the facts obtained during the resolution process
- It can store also other kinds of information needed for the control of the resolution process or other subsystems
  - Order of deduction of the facts
  - Use preferences of the facts
  - Rule that produced each fact
  - Rules recently fired
  - Backtracking points
  - ...

Explanation and resolution inspection subsystem

- The possibility to justify the decisions taken during the resolution process gives credibility to the system
- It allows also to detect wrong assumptions or decisions during the process
- A system should be able to answer why and how
- Different levels of explanation:
  - Trace: A list of the steps of the resolution is given
  - Justification: The reason why each element that appear in the trace of the resolution is given (reasoning path, questions, facts, preferences, subgoals, ...)

Notes
Learning subsystem

- Usually the set of problems that a KBS solves is closed
- In some domains it is necessary to adapt to the environment and to solve new problems
- Learning can happen:
  - During the development process of the KBS: The knowledge engineering process is substituted or complemented by inductive learning methods, building a model of the problem from examples
  - During the resolution process: Incorrect solutions are detected and corrected, new rules that make more efficient the resolution process are learned

Cased Based Reasoning (CBR)

- The solution of a problem is obtained identifying a previous similar solution
- Advantages:
  - The problem of knowledge acquisition is reduced
  - It is easier to maintain/correct/extend the system
  - The solving process is more efficient
  - It allows to obtain explanations closer to the user experience

Execution cycle

It has four phases
- Retrieval: The more similar stored cases are retrieved
- Reuse: The solution of the cases are obtained
- Revision: The retrieved solution is evaluated and adapted
- Retention: The relevance of the new case is assessed and the case is stored if it is an interesting one
Architecture of KBS - KBs based on Case Based Reasoning

**Execution cycle**

![Diagram of the execution cycle]

- **Domain Knowledge**
- **Retrieval**
- **Case Stored**
- **Case Revised**
- **Case Solved**
- **New Case**
- **Case Retrived**

**Knowledge storage subsystem**

- The knowledge is composed by **cases**
- A case is a complex structure (characteristics, solution)
- The cases are stored in the **case base** (structure, indexing)
- We will have also knowledge about:
  - How to evaluate case similarity
  - How to combine/adapt/retrieve solutions
  - How to evaluate solutions

**Knowledge use and interpretation subsystem**

- It is based on the execution cycle of Cased Based Reasoning
  - Retrieve the more similar cases from the case base
  - Obtain the solution from the cases
  - Combine/adapt the solution (procedures/reasoning)
Problem state storage subsystem

- Information of the current case
- Most similar cases retrieved
- Reasoning results from evaluating/combining/adapting the solutions

Explanation - Learning

- **Explanation**
  - It is a part of the information of a case
  - This explanation will be complemented by the reasoning process to combine/adapt the solutions
- **Learning**
  - To add new cases (easier than in production systems)
  - The new case has to be different enough (evaluation)
  - Cases could be forgotten (low usage, similar to others)

Other methodologies

- Systems based on neural networks
- Intelligent Agents/Multiagent systems
Neural networks

- Connectionist artificial intelligence
- The base element is the neuron (computational element)
- Neuron: Inputs, outputs, state, functions for the combination of inputs and state and function to compute the output
- Neurons are organized in networks with layers
- A neural network associates inputs (problem description) to outputs (solution of the problem)
- A neural network has to be trained (using solved problems) in order to learn how to solve the problem (association)
Intelligent Agents/Multiagent systems

- It supposes a collective vision of intelligent systems (instead of monolithic)
- An intelligent agent solves only a simple task
- An agent:
  - Obtains information from the environment (perception)
  - Elaborates a decision based on its perceptions and state (reasoning)
  - Performs an action (actuation)

Techniques involved: organization, cooperation, coordination, negotiation, tasks distribution, communication, reasoning about others, ...

Advantages:
- More flexible systems
- Reconfiguration/reorganization allow to solve other tasks
- Agents act as reusable components
- Fault tolerance (an agent can be substituted by other)
- Distributed computing

Related to:
- Grid computing/Cloud computing (Management of tasks and resources)
- Web services (No reasoning capabilities)