CSC 103 : INTRODUCTION TO EXPERT SYSTEM
(First Semester 2024/2025 Academic Year)
Engr. Dr. Ugorji C C

Module 1: Introduction to Expert Systems

Overview of expert systems:

1943: Post's Theorem on Computable Problems: In 1943, Emil L. Post demonstrated that any problem which can be computed can be expressed using a set of IF-THEN rules. This idea laid the groundwork for rule-based systems in AI, where logic is applied in a conditional format to solve problems. IF-THEN rules are fundamental to many AI algorithms and programming languages, and they represent one of the earliest approaches to formalizing problem-solving through computation. This concept influenced later developments in computer science and AI, providing a theoretical basis for rule-based reasoning in intelligent systems.

1961: General Problem Solver (GPS): The General Problem Solver (GPS), developed by Allen Newell and Herbert A. Simon in 1961, was an early AI program designed to mimic human problem-solving skills. GPS aimed to solve a wide range of problems by representing them in a formal language and using heuristics—rules of thumb—to guide the search for a solution. Though it was a general-purpose program, its performance was limited to specific, well-defined problems. GPS's significance lies in introducing the concept of heuristic search in AI, which remains essential for modern algorithms in areas such as path finding and optimization.

1969: DENDRAL and Domain-Specific Expertise: DENDRAL, developed in 1969 by Edward Feigenbaum, Bruce Buchanan, and Joshua Lederberg, marked a significant advancement in AI by being the first expert system to emphasize the importance of domain-specific knowledge. Unlike earlier systems that aimed for general problem-solving, DENDRAL focused on solving problems in organic chemistry, specifically in identifying molecular structures from mass spectrometry data. This approach demonstrated that integrating specialized knowledge into AI systems could significantly enhance their problem-solving abilities. DENDRAL's success underscored the value of combining AI techniques with expert knowledge, shaping the future development of domain-specific AI applications.

1970s: MYCIN and Certainty Factors:MYCIN, developed by Buchanan and Shortliffe during the 1970s, was an expert system designed for medical diagnosis, particularly for identifying bacterial infections and recommending treatments. One of MYCIN's groundbreaking contributions was the introduction of "certainty factors," which allowed the system to deal with uncertainty in medical decision-making. Unlike traditional binary logic, which would require facts to be either true or false, certainty factors enabled MYCIN to express degrees of belief in a diagnosis based on available evidence. This approach foreshadowed the development of probabilistic reasoning in AI, paving the way for modern machine learning techniques that handle uncertainty in data.

1982: R1 (XCON) - The First Commercial Expert System: In 1982, John McDermott developed R1 (also known as XCON), which was the first expert system to achieve significant commercial success. R1 was used by Digital Equipment Corporation (DEC) to configure computer systems, a task that involved selecting the appropriate components and assembling them according to specific requirements. The system's impact was profound; by 1986, it was saving DEC an estimated $40 million annually by automating the configuration process and reducing errors. R1's success demonstrated the commercial viability of expert systems and accelerated the adoption of AI in business applications, particularly in fields where specialized knowledge and rules could be codified.

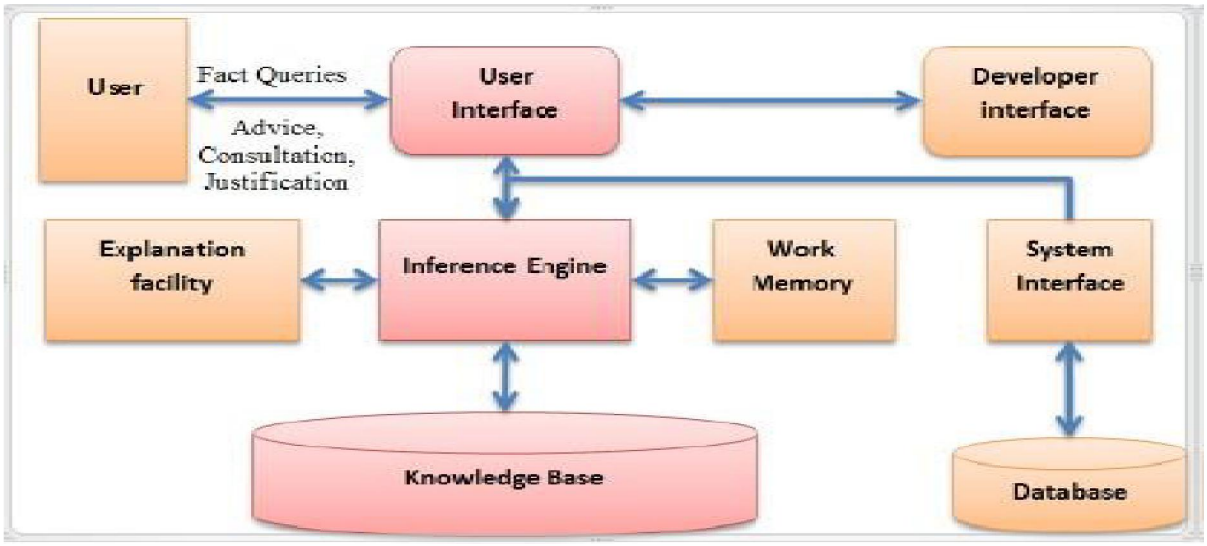
Evolution and Impact of Expert Systems: The progression from theoretical concepts, such as Post's theorem, to practical systems like R1, reflects the evolution of AI from abstract ideas to real-world applications. The development of domain-specific expert systems, the handling of uncertainty through certainty factors, and the commercialization of AI all played critical roles in shaping the trajectory of artificial intelligence. These milestones laid the foundation for modern AI systems, which continue to integrate specialized knowledge, probabilistic reasoning, and heuristic techniques to solve complex problems across diverse domains.

Definition, Component and Development of Expert Systems:

An expert is a person who has *expertise* in a certain area. That is, the expert has knowledge or special skills that are not known or available to most people.

Definition of Expert System: Is a branch of artificial intelligence, aiming to take the experience of human specialists and to transfer to a computer system. Specialty knowledge is stored in the computer, which by an execution system (inference engine) is reasoning and it derives specific conclusions for the problem.
Expert systems are usually built for specific application areas called Domain. The components of Expert System are as follows:

I. User Interface.
II. Inference Engine.
III. Knowledge base.
IV. Working memory.
V. Explanation facility



Architecture of an Expert system

1. User Interface

The user interface is the component of the system that accepts the user's query in a readable format, sends it to the inference engine, and then presents the results to the user. It serves as the communication channel between the user and the system, often featuring menus and graphical elements for user-friendly, problem-oriented interaction.

2. Knowledge Base

The knowledge base is a collection of facts and rules that encompass all the information about the problem domain. It contains the essential knowledge for understanding, formulating, and solving problems. In rule-based expert systems, knowledge is represented as a series of rules, each describing a relationship, recommendation, directive, strategy, or heuristic in an "IF (condition) THEN (action)" format. When a rule's condition is met, the rule "fires," triggering the corresponding action.

3. Inference Engine

The inference engine performs reasoning to enable the expert system to reach a solution. It connects the rules in the knowledge base with the facts from the database, determining which rules are applicable and prioritizing them. As the "brain" of the expert system, it serves as the control mechanism (rule interpreter) that applies reasoning methods. The inference engine works by selecting appropriate facts and rules from the knowledge base to address the user's query.

There are two main inference methods:

1. **Forward Chaining:** The inference engine applies rules whose conditions are met, adding conclusions to the working memory until no further rules can be applied. The order in which rules are checked is significant.

2. **Backward Chaining:** This approach starts with a goal and attempts to prove it by validating its conditions. For example, to prove a rule "IF A and B THEN C," the engine first confirms A and B, which may require further checks.

4. Working Memory

Working memory is used to store intermediate results and other information derived from the system's reasoning process.

5. Explanation Facility

The explanation facility allows the user to inquire about how the system arrived at a particular conclusion or why specific information is needed. It helps users understand the system's reasoning and justify its advice, analysis, or conclusions.

Development of Expert Systems:

Developing an expert system is a multi-step process that starts with identifying the right problem and ends with building a functional system that replicates human expert decision-making. Every phase is essential to ensure the system is accurate, dependable, and effective in addressing complex issues within its domain. From acquiring expert knowledge to evaluating the system's performance, each stage plays a vital role in making sure the system works as intended and meets user expectations. By following this systematic approach, developers can produce expert systems that enhance decision-making across various industries and applications.

Steps and processes in expert system development: The development of an expert system follows a series of structured steps, designed to ensure the system can replicate the decision-making skills of a human expert within a specific field. Below is an outline of the key stages:

1. **Problem Identification and Domain Selection:** The first step in building an expert system is to find a problem that the system will solve and choose the area where it will be used.

i. **Problem Identification:** The problem should need expert-level thinking or decision-making. Examples of such areas include medical diagnosis, financial planning, engineering design, or troubleshooting.
ii. **Domain Selection:** After choosing the problem, you need to define the specific field (domain) it will focus on, such as medical diagnosis or loan approval. The domain should be narrow enough for the system to manage but wide enough to handle different situations.
iii. **Feasibility Study:** Evaluate if the expert system can successfully solve the problem. Check if there are available experts, if the knowledge is too complex, and if the system can function in real-world situations.

2. **Knowledge Acquisition:** To collect and organize the expert knowledge needed to solve the chosen problem.

i. **Interviews and Collaboration:** The system developer, called the knowledge engineer, works closely with experts to gather their knowledge. This is done by interviewing the experts, observing how they make decisions, and studying past examples or cases.
ii. **Sources of Knowledge:** Information can come from different places, such as textbooks, research papers, past cases, and most importantly, the experience of the human experts.
iii. **Tacit vs. Explicit Knowledge:** Experts often have tacit knowledge, meaning they know things but find it hard to explain. The knowledge engineer must dig deeper to turn this unspoken knowledge into clear information that the system can use.

3. **Knowledge Representation:** To organize the collected knowledge in a way that the expert system can use to make decisions.

i**. Choosing a Representation Model:** There are different ways to represent knowledge, such as:

- **Rule-Based Representation:** Information is written as if-then rules. For example, If the patient has a high fever, then suspect an infection.
- **Frame-Based Representation:** Knowledge is arranged into frames that look like objects with attributes. For instance, a car frame might include engineand brakes.
- **Semantic Networks:** Concepts and their relationships are connected like a web. For example, a heart attack might link to symptoms like chest pain and shortness of breath.

ii.**Handling Uncertainty:** Some expert systems need to manage uncertain or incomplete information. This can be done using techniques like fuzzy logic or probability-based reasoning.

iii**. Building a Knowledge Base:** This is where all the structured knowledge (facts, rules, frames) is stored and used by the system for decision-making.

4. Designing the Inference Engine: This is the main part of the expert system that makes decisions. It uses the knowledge stored in the system to solve problems.

i. **Forward and Backward Chaining:**
   a. **Forward Chaining:** The system starts with known facts and uses rules to reach conclusions. For example, it might start with symptoms to figure out a diagnosis.
   b. **Backward Chaining:** The system starts with a guess and looks for facts that support it. For instance, it might begin with a suspected disease and check if the symptoms match.
ii. **Conflict Resolution:** Sometimes, multiple rules can apply at the same time. The system needs a way to decide which rule to use first. This can be done by giving priority to certain rules based on their reliability or using probabilities.
iii. **Uncertainty Management:** The inference engine must handle uncertainty. It can use fuzzy logic or probabilistic reasoning when dealing with outcomes that have probability values.

5. **User Interface Design**: To create an easy-to-use interface that allows users, who are usually not experts, to interact with the expert system.

i. **Designing Input and Output Mechanisms:** Users should be able to enter information easily and get back clear answers. The system might ask questions in a straightforward way and give simple, helpful advice or conclusions.
ii. **Natural Language Processing:** Some expert systems let users talk to them in everyday language, making them easier to use. For example, a user could say, "What's wrong with my car?" and the system would reply in simple terms.
iii. **Graphical User Interface (GUI):** If needed, a visual interface can be created to show decisions, processes, or diagnostics in a clear way.
iv. **Adaptability for Users:** The system should be flexible enough to meet the needs of different users, including beginners who may need more help and experts who want quick, advanced options.

6. **Implementation and Testing:** To create and check a working version of the expert system.

i. **Choosing a Development Tool or Programming Language:** Expert systems are usually built using special tools or programming languages like Prolog, LISP, or Python, especially if they include machine learning.
ii. **Building the System:** During this step, the knowledge base (where the information is stored), inference engine (the part that makes decisions), and user interface (what users see) are created based on the design. This is when coding and putting everything together happens.
iii. **Testing:** After building the system, it is thoroughly tested to make sure it works correctly. This includes:

- **Unit Testing:** Checking individual parts, like specific rules or sections of the system.
- **System Testing:** Testing the whole system with different situations to ensure it gives accurate and consistent results.
- **Expert Validation:** Having experts review the system's output to confirm that it makes decisions similar to a human expert.

7. **Evaluation and Refinement:**To keep improving the system by checking how well it works and making necessary changes.

i. **Field Evaluation:** The system is used in real-life situations by actual users, which helps gather feedback on how practical and effective it is.
ii. **Performance Metrics:** Measure how well the system performs by looking at its accuracy, speed, user satisfaction, and the quality of its decisions.
iii. **Knowledge Base Updates:** As time goes on, new information or rules may need to be added to the knowledge base. For example, in a medical expert system, new treatments or diagnostic techniques might be included.
iv. **System Refinement:** Using feedback from users and experts, the system is improved to fix any mistakes, enhance user interactions, or expand its decision-making abilities.

Key challenges in developing expert systems:

a. **Knowledge Acquisition:** Gathering the necessary expert knowledge can be difficult. Experts may find it hard to explain their thought processes or may have tacit knowledge that isn't easily communicated.
b. **Complexity of Knowledge:** The information that needs to be encoded into the system can be complex and vast. Organizing this knowledge in a way that the system can use effectively is a significant challenge.
c. **Handling Uncertainty:** Real-world situations often involve uncertainty and incomplete information. Designing the system to deal with these uncertainties, such as using fuzzy logic or probabilistic reasoning, can be complex.
d. **User Interface Design:** Creating an intuitive and user-friendly interface is essential, especially since many users may not be experts. Balancing simplicity and functionality can be challenging.
e. **Integration with Existing Systems:** The expert system often needs to work with other software or databases. Ensuring smooth integration can pose technical challenges.
f. **Maintenance and Updates:** Keeping the system up-to-date with new knowledge, rules, or changes in the domain is necessary but can be resource-intensive.
g. **Validation and Testing:** Ensuring the system produces accurate and reliable results through thorough testing and validation by domain experts can be time-consuming and complex.
h. **Acceptance and Trust:** Users must trust the expert system's recommendations. Building that trust can be a challenge, especially if users are accustomed to human experts.

The need for Expert Systems and Applications:

Organizations often face a shortage of problem-solving expertise due to the time and costs involved in training individuals, leading to a limited supply of experts. This shortage can result in delays and inconsistent decision-making, particularly when experts are unavailable or located far from where issues arise. For example, in factories, diagnosing and repairing specialized equipment usually requires experienced experts. When equipment malfunctions occur frequently or when factories operate around the clock, the demand for experts can quickly exceed their availability.

To address this challenge, developing expert systems can be an effective solution. These systems can help identify common equipment malfunctions and provide solutions, allowing routine issues to be resolved without expert intervention. This approach can minimize delays and boost productivity, as copies of the expert system can be distributed across multiple locations, ensuring access to expertise at all times.

Expert System Application Areas

Expert systems are being used in various fields where human expertise is needed.key areas where they are applied:

1. **Accounting & Finance**
   The finance sector heavily uses expert systems. They help bankers decide on loans and allow insurance companies to assess risks and set prices. They are also used in foreign exchange trading and other financial tasks like tax advice, credit analysis, and detecting insider trading.
2. **Agriculture**
   Expert systems assist with irrigation, pest control, and choosing the right crops and fertilizers. They also help manage soil, identify weeds, and plan agroforestry systems.
3. **Business**
   In business, expert systems help with tasks like developing advertising, analyzing markets, assessing sales staff, and planning careers. They are also useful for project management and evaluating product performance.
4. **Chemicals**
   These systems support chemical hazard evaluations, pollution control, and ingredient mixtures. They help with wastewater management and identifying metals and alloys.
5. **Computers**
   In the tech field, expert systems aid in diagnosing software issues, selecting new technologies, and assisting non-technical users. They help with hardware diagnostics and decision-making systems.
6. **Construction**
   Expert systems are used in construction for project scheduling, cost estimating, evaluating materials, and ensuring work zone safety.
7. **Education**
   In education, they recommend library resources, help interpret statistical data, and assist with medical training and financial aid eligibility.
8. **Engineering**
   These systems are widely used in engineering to solve complex problems related to design and manufacturing, including equipment diagnostics and scheduling power generation.
9. **Insurance**
   Expert systems assist in underwriting, rating life insurance, and managing social security benefits.
10. **Medical**
    They play a significant role in medical diagnostics, helping with admission protocols, X-ray analysis, and treatment decisions.
11. **Military, Government & Space**
    They assist in training, combat planning, and weather forecasting, as well as space mission analysis.
12. **Troubleshooting**
    Expert systems help identify faults and suggest fixes for equipment issues in areas like aviation, telecommunications, and power generation.
13. **Knowledge Publishing**
    This emerging area uses expert systems to provide relevant knowledge to users, such as grammar advice and tax guidance.

Module 2: Knowledge Representation in Expert Systems

Knowledge Representation: is a branch of artificial intelligence that focuses on capturing information about the world so it can help solve complex problems, like figuring out medical diagnoses. Knowledge and Representation are two important but different ideas in designing smart systems:

i. **Knowledge** is what the system understands about the world.
ii. **Representation** is how that knowledge is stored so that a computer can read and use it to solve problems

Knowledge begins with **data**, which is simply a collection of facts.
Data: These are individual facts.
Example: It is raining.
Information: This comes from organizing data and answering questions like "who," "what," "where," and "when."
Example:The temperature dropped 15 degrees, and then it started raining.
Knowledge: This is understanding the information and answering "how."
Example: *If the humidity is high and the temperature drops, it is likely to rain.*
Wisdom: This is knowing the principles behind the knowledge and answering "why."
Knowledge can vary in detail, ranging from specific information about a problem to more general concepts that apply to many situations.

Importance of Knowledge Representation:
Knowledge representation is vital because solving problems requires a lot of information, and it needs to be in a format that computers can understand.
To represent what we know, we look at two types of knowledge:

- **Knowing "how" to do something** (e.g., "how to drive a car" - procedural knowledge).
- **Knowing "that" something is true or false** (e.g., "that is the speed limit on a motorway" - declarative knowledge).

Two main types of Knowledge:

1. **Tacit Knowledge**: This is informal and not easily written down.It is personal, hard to share, and comes from experience.
2. **Explicit Knowledge**: This is formal and documented. It is easy to share and can be stored, copied, and is usually found in written materials or procedures.

Knowledge Typology Map: This illustrates how tacit and explicit knowledge relate. Tacit knowledge comes from personal experience, while explicit knowledge is based on principles, procedures, processes, and concepts. Good knowledge representation allows for quick and easy access to information

i. **Facts**: Specific, unique pieces of information.
ii. **Concepts**: Categories of items that share common features.
iii. **Processes**: Descriptions of how things happen.
iv. **Procedures**: Step-by-step instructions to complete a task.
v. **Principles**: Guidelines that help make predictions and decisions.

.

Knowledge Representation Methods: Knowledge in expert systems can be represented in three main ways:

1. **Production Rules**:
   These are IF-THEN statements that explain relationships and guide actions.
   Example: IF the heating element glows AND the bread is dark, THEN the toaster thermostat is broken.
   Components: Working memory (information about the current problem), rule base (general information), and interpreter (decides which rule to apply).
   Advantages: Easy to understand, modular, and can handle uncertainty.
   Disadvantages: Can be inefficient and less expressive.
2. **Semantic Nets**:
   These are visual representations that show how concepts are related using nodes (for objects) and links (for relationships).
   Example: A cat is an animal, and all animals can move.
   Advantages: Easy to visualize, space-efficient, and keeps related knowledge grouped together.
   Disadvantages: Issues can occur from improper inheritance and unclear standards for values.

3. **Frames**:
   Frames are like semantic nets but include properties. They represent objects as pairs of slots and fillers.
   Components: Frame name, attributes (slots), and values (fillers). Fillers can connect to other frames.
   Advantages: Directly reflect domain knowledge, efficient, and support complex knowledge structures.
   Disadvantages: Lack clear semantics and have some expressive limitations.

CLASSES OF EXPERT SYSTEM:

Expert systems can be divided into four main types based on how they work:

1. Rule-Based Expert Systems
2. Frame-Based Expert Systems
3. Fuzzy Logic-Based Expert Systems
4. Neural Network-Based Expert Systems

This section will give a quick look at the structure of each of these types**.**

1. Rule-Based Expert Systems

The Rule-Based Expert System is the first and most commonly used type of expert system, often used by researchers to find solutions. It relies on rules and is applied in many fields to help solve different problems using existing knowledge.

This system works by using rules that reflect expert knowledge. Most expert systems are rule-based and follow a set of guidelines that a human expert would use to diagnose issues. For example, a rule-based system might help a doctor choose the right diagnosis based on a set of symptoms or guide a player in making the best moves in a game.

Some advantages of Rule-Based Expert System:

- It uses "if-then" statements to present knowledge, making it easy for people to understand.
- It also separates knowledge and reasoning, which aligns with our everyday thinking.

Advantages:

- It tends to be slower than other types because they need to go through all the rules in the database each time a rule is applied.
- Precision is crucial. When a rule matches a condition, the wording must be exactly the same as in the database. This can create issues if there are minor differences.

2. Frame-Based Expert Systems:

A frame-based expert system uses a structure called a frame to hold information about specific objects or ideas. Marvin Minsky introduced frames in the 1970s.
A frame is a data structure with typical knowledge about a particular object or concept. Frames are used to capture and represent knowledge in a frame based expert system. Each frame has its own name and set of attributes or slots associated with it, Such as Name, weight, height and age are slots in the frame Person and Model, processor, memory and price are slots in the frame Computer. Frame-based expert systems use facets to add extra details to frame attributes. There are three types:

1. Value facets: Set default values.
2. Prompt facets: Let users input values during a session.
3. Inference facets: Pause the process if an attribute changes.

To manage knowledge, the system uses methods (commands that run as needed) and demons (IF-THEN rules that act when a condition changes)

3. Fuzzy Logic-Based Expert System:

A **Fuzzy Logic-Based Expert System** uses fuzzy logic to handle uncertain or imprecise information. It was introduced by mathematician Zadeh in 1965. Fuzziness means there are no clear boundaries between different things, and fuzzy logic helps deal with this uncertainty. Fuzzy logic works on the idea that things can be on a sliding scale, not just true or false. Unlike classical logic, which uses only "True" (1) and "False" (0), fuzzy logic uses values between 0 and 1 to show the possibility of something being true or false. In these systems, reasoning is based on **fuzzy rules** that make approximate conclusions. For example, rules like "If x is A, then y is B" help draw fuzzy judgments. The results are fuzzy sets, but a specific value is needed. The simplest way is to pick the value with the highest membership. The advantage of this system is its ability to handle complex expert knowledge and offer flexibility in reasoning. However, it struggles with learning new knowledge and may be prone to errors.

4.Neural Network-Based Expert Systems:

Neural networks, inspired by the brain's structure, help computers learn from data. Like the brain's neurons, which form complex connections, artificial neural networks (ANNs) use mathematical models to find patterns, classify, and make predictions. Unlike traditional expert systems that use explicit rules, ANNs learn automatically from examples and adjust connections to improve accuracy. Neural networks are low-level computational structure that performs well when dealing with raw data. ANNs have advantages like efficiency and fault tolerance but also have limitations. Their performance depends on the quality and quantity of training data, and they cannot explain their reasoning because they mimic the brain's neural activity.