Introduction to Artificial Intelligence

Subject 5: Introduction to Expert System

Dr. Mohammed Shambour

knowledge-based system (KBS) & Knowledge

- **What is a knowledge-based system?**
  - A knowledge-based system (KBS) is essentially a **computer program** designed to solve complex problems by relying on a **specific set of human expertise**, rather than relying solely on traditional procedural code.

- **What is Knowledge?**
  - Knowledge is the **information** that people use to *solve problems* or *make decisions*. Knowledge can be broken down into several elements:
    - **Facts**: Basic truths or confirmed data (e.g., "The capital of Jordan is Amman").
    - **Concepts**: Generalized ideas or categories (e.g., "Algorithm," "Linked List").
    - **Procedures**: Step-by-step instructions for performing a task (e.g., "Steps for performing a tree expanding").
    - **General rules or empirical knowledge**: The knowledge comes from **experience and observation** rather than pure theory.
    - **Sample Cases**: Specific cases or instances that help illustrate, explain, or learn a concept or solution. (*e.g., A computer that failed to boot because the RAM was not properly seated*).

knowledge-based system (KBSs) & Knowledge

In short, a **Knowledge-Based System** is a computer program that has a "**brain**" filled with human expert knowledge, allowing it to make smart decisions and solve problems in a specific area, just like a human expert would. ☺

Expert systems

What is an expert system?

A particular kind of knowledge-based system

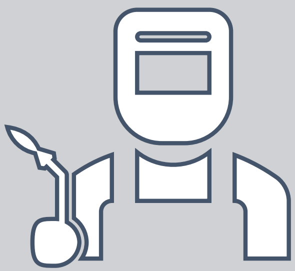
To some extent, an expert system can act as a substitute for the expert from whom the knowledge was taken.

Knowledge engineering

Knowledge engineering is essential in expert systems because it enables the system to actually **think**, **reason**, and **solve problems** like a human expert.

Without knowledge engineering, an expert system would have **no organized expertise** and **no ability to make informed decisions.**

Knowledge engineering



Knowledge engineering refers to the complete process of:

1.Designing the architecture,

2. Building the content, and

3. Deploying and Maintaining the System

https://www.16personalities.com/ar

Knowledge engineering Process: 1.Designing the architecture

Here are specific examples for each phase, using the context of building an **Automated Medical Diagnostic System**.

1. Designing the architecture (Conceptualizing the System)

This phase defines what the system will do and how it will operate.

Design Task	Example in Medical Diagnostic System
Knowledge Acquisition Strategy	How the system will *collect* expert knowledge. e.g. Deciding to use **structured interviews** with five cardiologists and ten years of patient treatment records to extract the necessary diagnostic rules.
Representation Scheme	How the system will *store* and *organize* knowledge. The system uses If–Then rules to represent medical knowledge. e.g. **IF** the patient's temperature is above 38.5°C **AND** the patient has body aches **THEN** the system suggests a high probability of influenza.
Inference Mechanism	How does the system *handle* stored knowledge? The system will use Forward Chaining to process patient symptoms (facts) and derive a final diagnosis. e.g. A patient enters the following facts: Temperature = 39°C, cough, and Shortness of breath. The system uses forward chaining to match the patient's symptoms to rules and produce the most likely diagnosis.
User Interface (UI) Design	How the user will interact with the expert system. e.g. The user interface guides nurses through simple symptom questions, shows diagnostic progress, and confidently presents the final result.

Knowledge engineering Process: 2. Building the content

2. Building the content

This phase involves converting human expertise into a working system.

Building Task	Example in Medical Diagnostic System
Knowledge Base (KB) Population	Input facts and rules into the system. e.g. **IF** CBC shows low hemoglobin **AND** low serum ferritin **THEN** diagnosis = Iron-Deficiency Anemia.
Testing and Validation	Check system accuracy using real cases. e.g. Example: Run 100 historical patient cases; **refine rules** when diagnoses don't match physician results.
Handling Uncertainty	Incorporate probabilities or confidence levels. e.g. Assign a certainty factor of 0.8 for Flu if symptoms partially match.
Component Integration	Connect all internal parts. e.g. Link a Prolog inference engine to a SQL-based knowledge base.

Knowledge engineering Process: 3. Deploying and Maintaining the System

3.Implementing the expert system

This phase brings the system into real-world use and ensures it stays accurate over time.

Building Task	Example in Medical Diagnostic System
Deployment	Install and integrate the final system. e.g. Deploy the diagnostic tool on the hospital server and link it to the electronic medical records system.
Training	Teach staff how to use the system effectively. e.g. Train nurses and doctors to enter symptoms and interpret the explanation ("Why") feature.
Maintenance and Evolution	Update the system as knowledge changes. e.g. Review accuracy metrics monthly and update rules to reflect new medical guidelines or emerging pathogens.
Acceptance Testing	Run final checks in a controlled environment. e.g. Compare system diagnoses with real doctors for several weeks before full adoption.

Experts

- **An expert** is an experienced practitioner in his/her particular field. More than that, he/she is a highly effective problem-solver and decision-taker in that field.
- Experts have three qualities:
  - They make good decisions.
  - They make those decisions quickly.
  - They are able to cope with a wide range of problems.

As a result, they are highly valued, receive high wages, and tend to be overworked.

Experts and expert systems

- Note that:
  - The task that an expert system performs will generally be **regarded as difficult.**
  - An expert system almost always operates in a rather narrow field of knowledge. The field of knowledge is called the knowledge domain of the system.
  - There are many fields where expert systems can usefully be built.
  - There are many fields where they can’t.

Fields Where Expert Systems Can Be Usefully Built

These tasks are **rule-based**, **knowledge-intensive**, and **narrow in scope, such as:**

1. Medical Diagnosis (Specific Specialties)
- Diagnosing heart disease, skin conditions, or eye infections.

2. Technical Troubleshooting
- Diagnosing car engine failures
- Finding faults in computer networks

3. Finance and Banking
- Credit risk assessment
- Loan approval recommendations
- Fraud detection using rule patterns

4. Agriculture
- Identifying crop diseases
- Suggesting fertilizer or irrigation strategies

5. Customer Support Chatbots (Rule-Based)
- Providing structured answers in telecom, banking, or IT support

These domains share two traits:
✓ They are **difficult problems**
✓ But they are **narrow**, with clear rules or expert empirical knowledge

Fields Where Expert Systems Cannot Be Effectively Built

These tasks require common sense, broad knowledge, or creativity—things rule-based systems cannot handle, **such as:**

1. Open-Ended Medical Practice

- Making all medical decisions in all specialties
- Covering every possible rare or unknown disease

2. Creative Work: Creativity cannot be reduced to If–Then rules.

- Writing novels, composing music, painting artwork

3. Human Emotion Understanding

- Detecting complex emotional states
- Giving psychological counseling

4. General Conversation

- Conducting a natural and free human dialogue

5. Real-World Robotics

- Navigating unpredictable environments

These domains share two traits:
✓ They are **difficult problems**
✓ They require broad judgment, experience, ethics, or creativity, which cannot be reduced to fixed rules or narrow domain knowledge.

Experts and expert systems

- Note also that an expert can usually
  - explain

  and
  - justify

  his/her decisions.

Reasons for building an expert systems

- One might build an expert system for any or all of the following reasons:
  - To archive an expert's knowledge, to insure against the day when he/she leaves, or retires, or dies.
  - To disseminate his/her knowledge, so that it is available in more (possibly many more) places than the location of the expert.
  - To ensure uniformity of advice/decisions.
  - As a basis for training other specialists.

Advantages of expert systems

- An expert systems have the following advantages over human experts:
  - ❑ The knowledge is permanent (e.g., A human doctor, however, may forget rare symptoms over time).
  - ❑ The knowledge is easily replicated (e.g., If a bank creates a loan-approval expert system, it can run the same system in many branches).
  - ❑ The knowledge is represented explicitly, and can be evaluated (e.g., A rule might say: “IF temperature > 38°C AND cough → THEN possible influenza”. Humans cannot always explain their reasoning, but the expert system’s reasoning can be reviewed and corrected.
  - ❑ The system is consistent - whereas human practitioners have bad days, computers don’t (e.g., Unlike a human tax officer, a tax expert system will always calculate the same tax amount for the same data.
  - ❑ Once built, running costs are low (e.g., A plant-disease expert system installed on a farmer’s mobile phone runs for years at minimal cost).

Disadvantages of expert systems

- Developing an expert system usually costs a great deal of time & money
- Historically, there has been a high failure rate in expert systems projects:
  - The project may fail during development—most commonly in the "knowledge acquisition" phase. In many cases, the knowledge transferred from human experts to system engineers is incomplete. This happens because experts often rely on intuitive, experience-based insights that are difficult to explain, leading to gaps in the knowledge that the system needs to perform effectively.
- The development may succeed, but the organisation may fail to accept and use the finished system. This can happen due to:
  - **Lack of Trust:** Users may be hesitant to rely on a computer system over a human expert.
  - **Poor Integration:** The system may not integrate well with existing organizational workflows.
  - **Perceived Threat:** Experts may view the system as a threat to their job security, leading to resistance or non-cooperation.

Disadvantages of expert systems

A human expert can update his/her knowledge in the light of

- Common sense
- Knowledge derived from other domain
- Contacts with other experts.

An expert system can’t.

Choosing an Expert System project

It is important that E.S. projects are carefully chosen, because of:

1. Cost

❑ Developing an expert system requires **significant resources**:

- Collecting and structuring expert knowledge takes time.
- Hiring **knowledge engineers** and **domain experts** can be expensive.
- Maintaining and updating the knowledge base is ongoing work.

2. Danger of failure

❑ Expert systems can fail if:

- The **knowledge base is incomplete** or inaccurate.
- Users **don’t trust or understand** the system’s recommendations.

3. Project Selection Criteria

❑ Organizations should only seek expert systems projects that:

- Have **clear, well-defined problems** suitable for rule-based or knowledge-driven reasoning.
- Are **feasible** in terms of time, cost, and available expertise.
- Promise **significant value** — such as improving accuracy, saving money, or preserving expert knowledge.

Guidelines for Choosing Expert System Projects

Because expert systems can be costly and risky, projects must be carefully evaluated.

1. The Expense Must Be Justified

- Development costs should be reasonable when compared to the expected benefits.
- The project should clearly add value — for example, by saving time, improving accuracy, or preserving expert knowledge. (e.g. Hospital services vs number of staff)

2. Technology Must Be Appropriate

- Expert systems are best for problems that need expert reasoning or judgment, not simple calculations.
- Example: Good use:A legal advisory system that interprets rules and exceptions based on expert lawyers' knowledge. Bad use: A payroll system that only calculates salaries.

3. The Right Expertise Must Be Involved

- Collaboration with domain experts and knowledge engineers is essential.
- The quality of the system depends on the quality and depth of the expertise captured. (e.g. Hiring senior vs junior expert for diagnosing car engine faults)

4. The Problem Should Not Be Solvable by Conventional Programming

- If the task can be easily solved using standard algorithms or traditional software, then an expert system is unnecessary (e.g., Converting temperatures from Celsius to Fahrenheit).

5. Full Support from Management and Participants

- Both management and technical participants must actively support the project.
- Success depends on commitment to knowledge sharing, testing, and system adoption (e.g., Officers hid the loan rules, because they fear the system will replace them).

Types of expertise

In knowledge-based systems and expert systems, expertise generally falls into **four main categories**, each capturing a different kind of expert knowledge:

1. Domain Expertise

- Knowledge specific to a particular field or discipline.
- Examples: medical diagnosis, civil engineering design, GIS coordinate systems.

2. Experience-based Expertise

- Experience-based rules that experts use to make decisions quickly.
- Not formally proven, but effective in practice.
- Example: “If symptoms A and B appear together, check condition X first.”

3. Meta-Expertise

- Knowledge about ***how*** to use or apply other knowledge (e.g., When should a doctor to rely on lab tests (codified knowledge) OR clinical judgment(experiential knowledge)).
- Includes strategies for selecting rules, deciding which method to apply, or resolving conflicts.

4. Control Expertise

- Control expertise = knowledge that controls how the reasoning happens.
- It guides rule selection, ordering, and optimization during inference.
- E.g., In a car diagnosis system, checking the battery (rule 1) before checking the spark plugs (rule 2) because it is more common and simpler to test.

The phone call test

A simple and practical way to decide **which problems are suitable for building an expert system** is summarized by Prof. Morris Firebaugh:
"Any problem that can be, and frequently is, solved by your in-house expert in a 10–30 minute phone call can be automated as an expert system." www.cwa.mdx.ac.uk

Meaning of the statement

This guideline suggests that the **best domains for expert systems** have the following characteristics:

1. **The problem is well-defined**
   – The expert does not need hours of investigation.
   – The reasoning steps are structured and repeatable.
2. **The solution is commonly provided through consultation**
   – Employees or clients frequently call an expert for guidance.
   – The expert typically uses a set of known rules, checks, or decision steps.
3. **The expert usually answers within 10–30 minutes**
   – This indicates that the knowledge can be broken into *rules*, *if–then structures*, *decision trees*, or *diagnostic steps*.
4. **No deep research or creativity is required**
   – The expert applies experience and known rules—not long calculations or novel solutions.

The phone call test: Examples

Example 1: IT Helpdesk Troubleshooting
A senior IT technician solves most problems over the phone:

- "Is your computer on?"
- "Is the cable connected?"
- "Do you see an error message?"
- "Restart and check again."

These steps can easily be put into an expert system that guides users through the same process.

Example 2: Medical Triage
A nurse answers phone calls:

- "Do you have a fever?"
- "How long have you had the symptoms?"
- "Are you having trouble breathing?"
- "Go to the clinic immediately" or "Rest and drink fluids."

This structured questioning can become a triage expert system.

Example 3: Car Maintenance Advice
A mechanic helps customers by phone:

- "Is the engine light on?"
- "Do you hear a knocking sound?"
- "Does the car start?"
- Recommendation: "Check the battery," "Go to service," etc.

This decision process can be turned into rules.

Example 4: Honey Quality Evaluation
The chief expert can often predict honey quality by asking junior staff:

- "What is the color?"
- "What is the scent strength?"
- "How thick is it?"
- "Does it crystallize quickly?"

This can be automated into a honey-quality expert system.

Possible expert systems - case histories

- For discussion....
- The following seven problem areas **may**, or **may not**, be suitable for computerisation as expert systems.

Possible expert systems - case histories

1. Medical Diagnosis in Remote Areas

A developing country has a large population but very few doctors, and cannot afford to train many more. The government plans to train paramedics instead and provide them with medical kits and portable computers. Each computer would contain an expert system to help diagnose and treat common diseases.

Possible expert systems - case histories

2. Housing Department Interview Support

The Amman post office currently allocates most staff time to client interviews based on a routine question sequence with limited variability. The department proposes building an expert system to guide and standardize the questioning process.

Possible expert systems - case histories

3. Honey Quality Assessment

A honey-importing company depends heavily on its chief quality expert, who can accurately judge which honey varieties will appeal to customers based on taste, color, aroma, and texture. Since she will soon retire, the company plans to create an expert system that captures her expertise so junior honey specialists can perform evaluations in the same way.

Possible expert systems - case histories

4. Teaching Young Children

An education authority faces a severe shortage of primary school teachers. It proposes developing an expert system capable of teaching five-year-old children basic English and arithmetic.

Possible expert systems - case histories

5. Automated Book-Keeping

A software company proposes developing an expert system that can perform bookkeeping tasks for small businesses.

Possible expert systems - case histories

6. Diagnosing Complex Machine Faults

A major manufacturer of electric and diesel vehicles is seeking to provide a sufficient number of qualified maintenance technicians capable of diagnosing complex mechanical and electrical faults. The company proposes building a specialized system capable of identifying faults in these machines.

Possible expert systems - case histories

7. Analyzing geological sites

A mining company wants to expand its operations and search for new metals, but it doesn’t have enough trained geologists. To solve this problem, the company plans to build an expert system capable of analyzing geological sites and estimating the likelihood of finding valuable minerals.

That’s all for Today