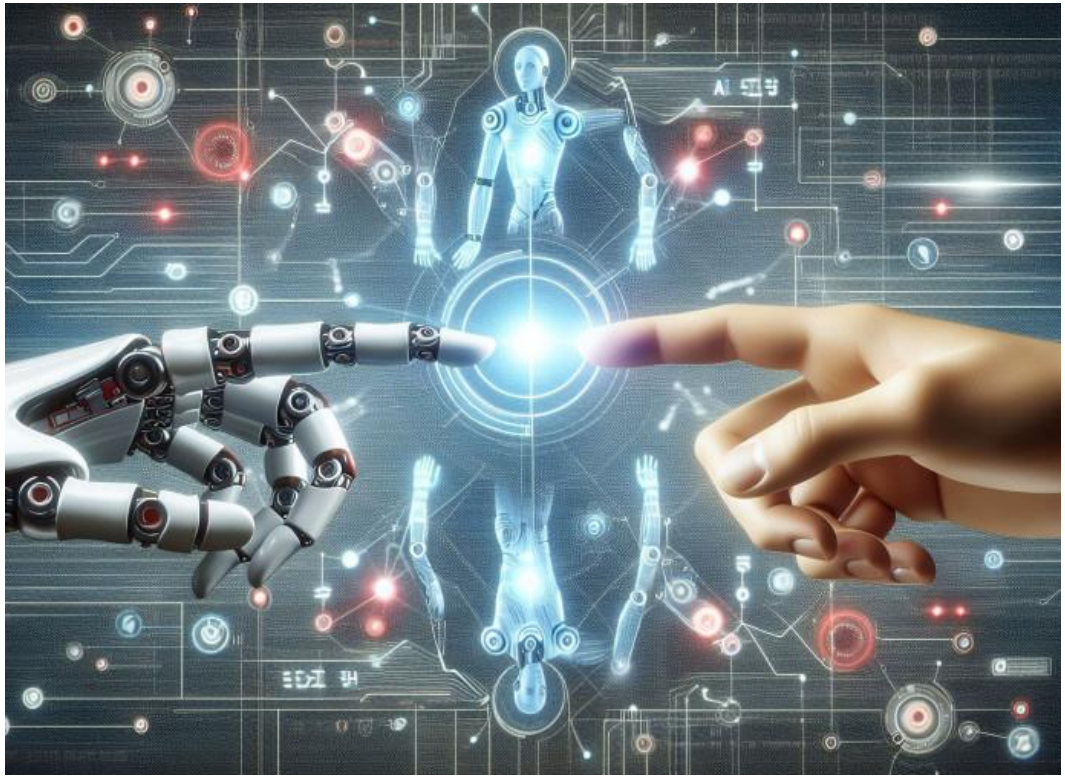
Lecture 1
Introduction to Expert Systems



Objectives

- Learn the meaning of an expert system
- Understand the problem domain and knowledge domain
- Learn the advantages of an expert system
- Understand the stages in the development of an expert system
- Examine the general characteristics of an expert system

What is an expert system?

"An expert system is a computer system that emulates, or acts in all respects, with the decision-making capabilities of a human expert."

Professor Edward Feigenbaum

Stanford University

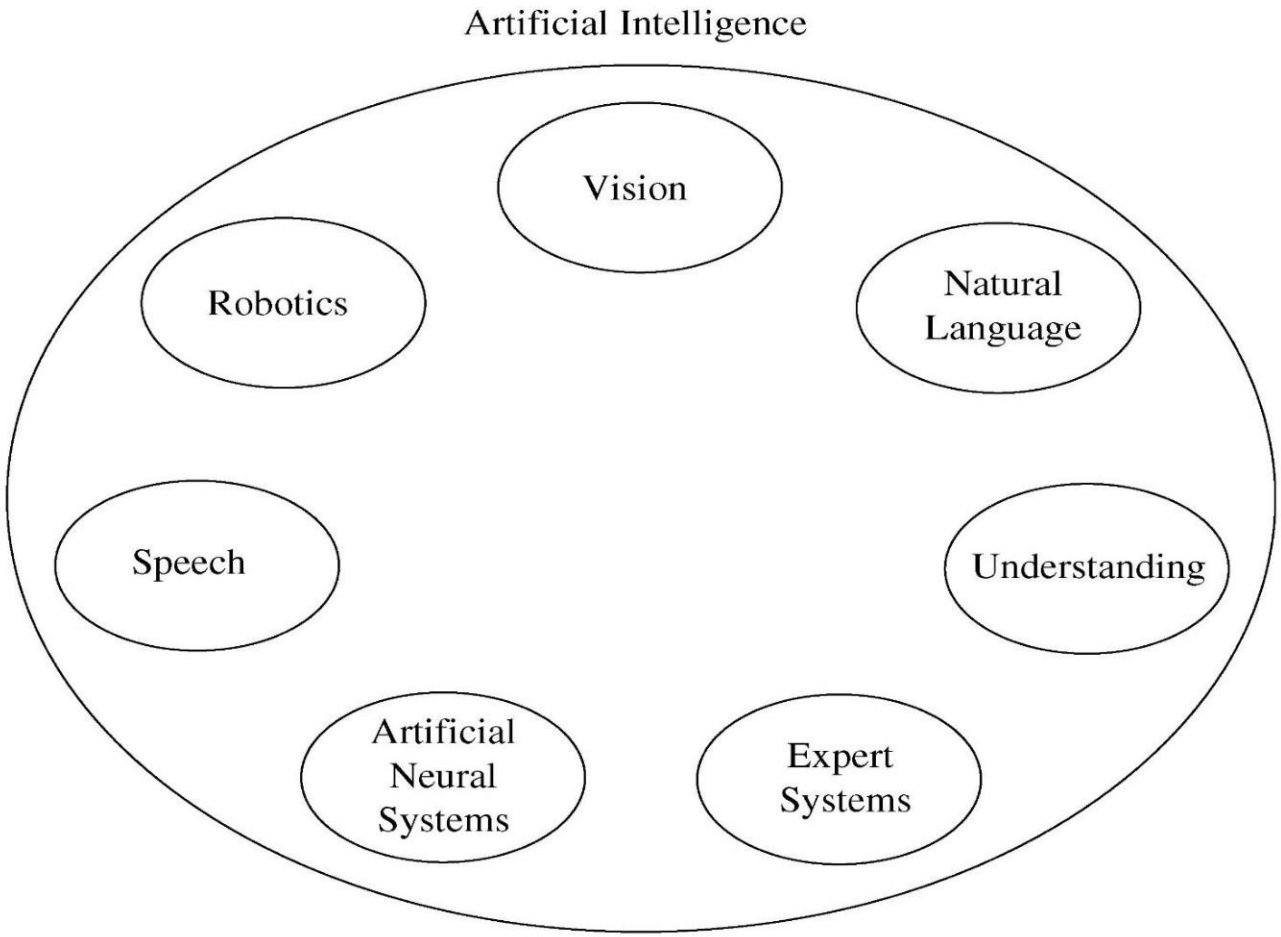
النظام الخبير هو نظام حاسوبي يحاكي قدرات صنع القرار لدى الخبير البشري في مجال معين .

هو جزء من مجال الذكاء الاصطناعي الذي يهدف إلى تمكين الحواسيب من حل المشكلات المعقدة التي تتطلب عادة خبرة بشرية.

الهدف الرئيسي **:

توفير بديل أو مساعد للخبراء البشريين في مجالات مثل الطب، والهندسة، والتمويل، والعلوم.

Fig 1.1 Areas of Artificial Intelligence



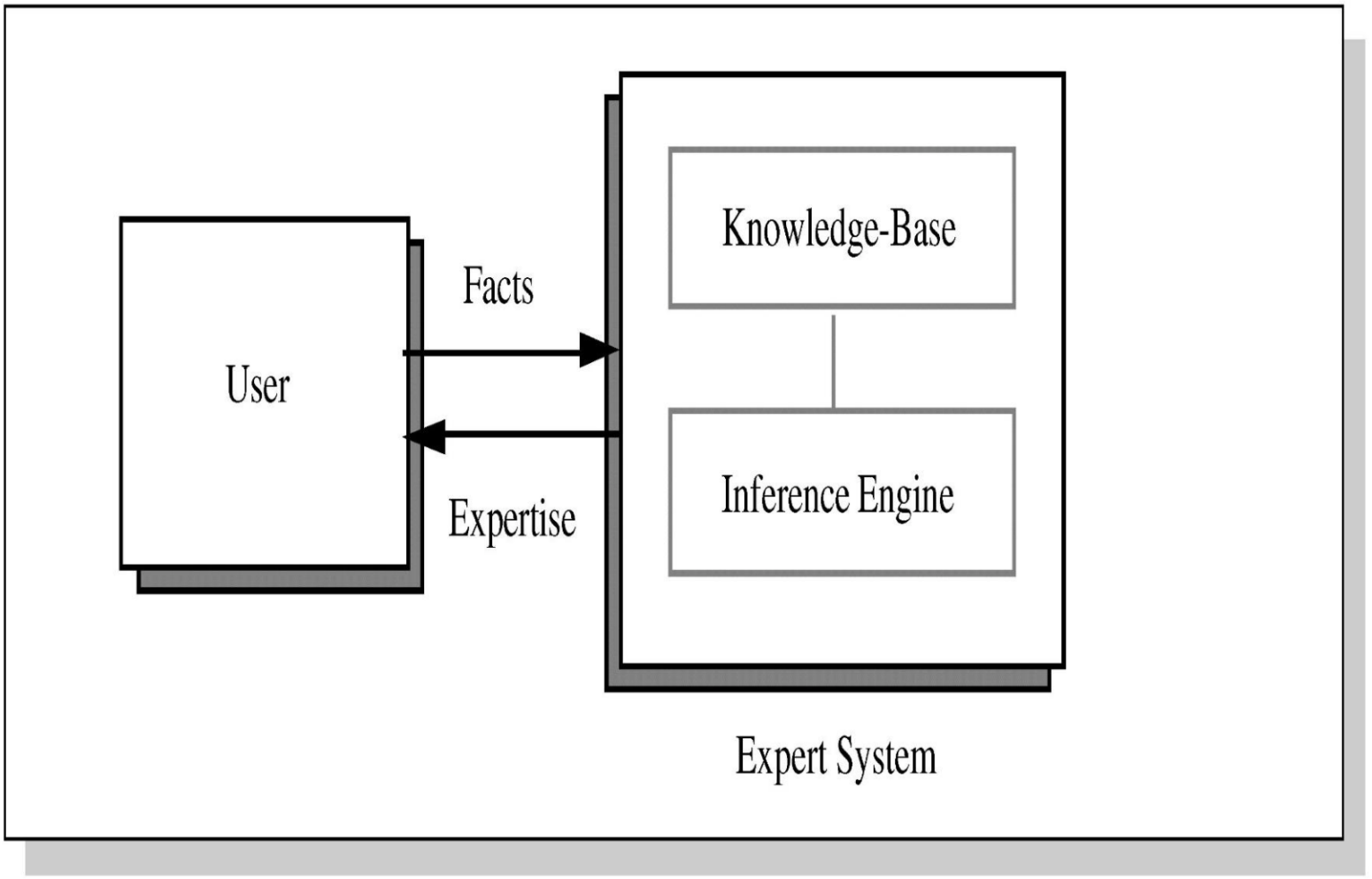
Expert system technology may include:

- Special expert system languages – CLIPS
- Programs
- Hardware designed to facilitate the implementation of those systems

Expert System Main Components

- Knowledge base – obtainable from books, magazines, knowledgeable persons, etc.
- Inference engine – draws conclusions from the knowledge base

Figure 1.2 Basic Functions of Expert Systems



Problem Domain vs. Knowledge Domain

- An expert's knowledge is specific to one problem domain – medicine, finance, science, engineering, etc.
- The expert's knowledge about solving specific problems is called the knowledge domain.
- The problem domain is always a superset of the knowledge domain.

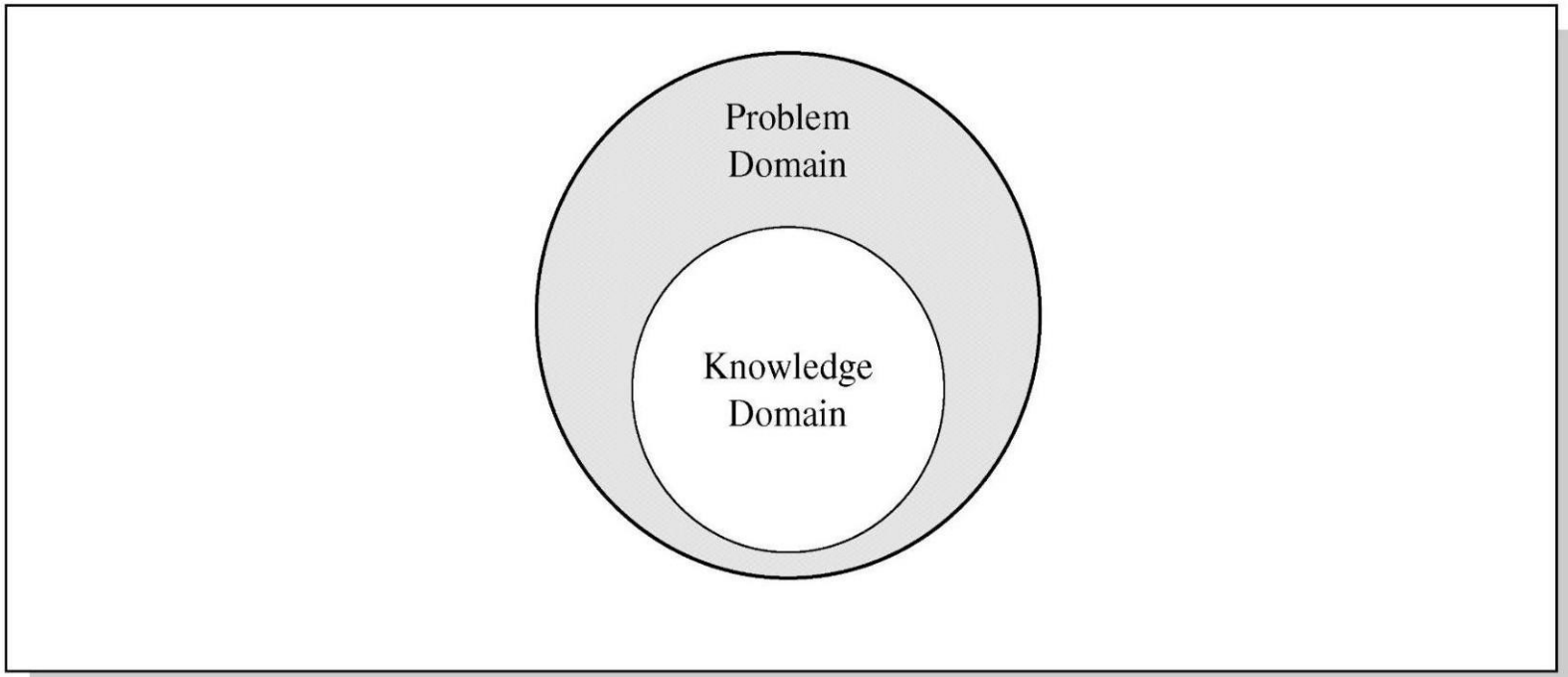
problem domain vs knowledge domain

Problem Domain: This is about **what problem you're trying to solve**. It defines the specific area of focus or application. For example, if you’re building an expert system for diagnosing car issues, the problem domain is "car repairs.“

Knowledge Domain: This is about **what information you need to solve that problem**. It encompasses all the facts, rules, and expertise required to make decisions within that problem domain. In the car repairs example, the knowledge domain would include information about car mechanics, parts, symptoms of common issues, and troubleshooting techniques.

In short: - ****Problem Domain** = What problem you're addressing** (e.g., car repairs). - ****Knowledge Domain** = What knowledge is needed to address that problem** (e.g., car mechanics and troubleshooting).

Figure 1.3 Problem and Knowledge Domain Relationship



Advantages of Expert Systems

- Increased availability
- Reduced cost
- Reduced danger
- Performance
- Multiple expertise
- Increased reliability

Advantages Continued

- Explanation
- Fast response
- Steady, unemotional, and complete responses at all times
- Intelligent tutor
- Intelligent database

مميزات النظم الخبيرة (الفوائد الرئيسية)

1Preservation of Expertise. الحفاظ على الخبرة

المعرفة والخبرة البشرية معرضة للضياع بسبب تقاعد الخبير، أو مغادرته للعمل، أو حتى وفاته. النظام الخبير يلتقط هذه المعرفة ويخزنها بشكل دائم.

مثال: إذا كان أفضل طبيب في تشخيص مرض نادر على وشك التقاعد، يمكن نقل خبرته إلى نظام خبير ليستفيد منه الأطباء الآخرون لسنوات قادمة.

2Availability. التوافر الدائم24/7

الخبير البشري يحتاج للنوم والراحة والإجازات. النظام الخبير يعمل 24 ساعة في اليوم، 7 أيام في الأسبوع، وأيام العطل.

مثال: يمكن لمهندس الصيانة استخدام النظام الخبير في منتصف الليل لتشخيص عطل مفاجئ في آلة معقدة، دون الحاجة لانتظار قدوم الخبير الرئيسي في الصباح.

3Reduced Cost. التكلفة المنخفضة

استشارة النظام الخبير أرخص بكثير من استئجار خبير بشري بدوام كامل. كما أنه يقلل من التكاليف الناتجة عن الأخطاء أو التوقف الطويل.

مثال: بدلاً من سفر خبير دولي لتشخيص مشكلة في مصنع، يمكن استخدام نظام خبير محلي بتكلفة ضئيلة.

4Consistency and Lack of Emotion. الاتساق وعدم التأثر

الخبير البشري قد يتعب، أو يمر بيوم سيء، أو يتأثر بعواطفه، مما قد يؤثر على قراراته. النظام الخبير يتعامل مع كل حالة بنفس المنطق والموضوعية، مما يضمن **اتساق** النتائج.

مثال: نظام خبير للموافقة على منح القروض سيحكم على جميع الطلبات بنفس المعايير، دون تحيز لشكل طالب القرض أو حالته المزاجية.

5Dealing with Incomplete Information. القدرة على التعامل مع المعلومات غير الكاملة

يمكن للنظام الخبير أن يعمل حتى مع وجود نقص في بعض البيانات، حيث يستخدم قواعد المنطق والاحتمالات لتقديم أفضل حل ممكن بناءً على المعلومات المتاحة.

مثال: إذا كان نظام تشخيص الأمراض يفتقد نتيجة أحد التحاليل، يمكنه أن يقدم قائمة بالأمراض المحتملة مع توصية بإجراء تحليل معين للحصول على تشخيص أدق.

6Multiple Geographic Locations. التوزيع الجغرافي

يمكن نسخ النظام الخبير وتوزيعه على عدة فروع لشركة ما في أنحاء العالم، مما يمنح الجميع إمكانية الوصول إلى نفس مستوى الخبرة في نفس الوقت.

مثال: يمكن لشركة طيران أن يكون لديها نفس النظام الخبير لصيانة الطائرات في جميع مطاراتها حول العالم.

7Training Tool. كأداة تدريب

يمكن للمبتدئين استخدام النظام الخبير كمدرب. فهم يطرحون عليه المشاكل ويرون كيف يستنتج الحل، ويتعلمون من خطوات تفكيره.

مثال: طالب طب يمكنه استخدام نظام تشخيصي لاختبار معرفته، ثم مقارنة استنتاجه مع استنتاج النظام وفهم الأسباب behindكل خطوة.

8Explanation and Documentation. القدرة على التوثيق والشرح

أحد أهم المميزات هي قدرة النظام على **شرح استنتاجه**. فهو لا يعطيك الإجابة فحسب، بل يخبرك **بسبب** وصوله إلى هذه الإجابة، قائلاً: "لقد وصلت إلى هذا التشخيص لأن الأعراض (أ، ب، ج) تشير إليه، بينما استبعدت التشخيص (د) لأن العرض (هـ) غير موجود".

مثال: هذا يزيد من ثقة المستخدم في النظام ويتيح له مراجعة المنطق المتبع.

النظم الخبيرة تحول الخبرة البشرية الثمينة والمحدودة إلى "سلعة" رقمية متاحة للجميع، في أي وقت، وبطريقة موضوعية ومتسقة، مما يعزز من كفاءة و دقة اتخاذ القرارات في المجالات التخصصية.

ملاحظة: على الرغم من هذه المميزات، تبقى النظم الخبيرة أدلة وليست بديلاً كاملاً عن العقل البشري، خاصة في المواقف التي تتطلب إبداعاً أو تعاملاً مع سياقات غير مألوفة.

Representing the Knowledge

The knowledge of an expert system can be represented in a number of ways, including IF-THEN rules:

```
IF you are hungry THEN eat
```

Knowledge Engineering

The process of building an expert system:

1. The knowledge engineer establishes a dialog with the human expert to elicit knowledge.
2. The knowledge engineer codes the knowledge explicitly in the knowledge base.
3. The expert evaluates the expert system and gives a critique to the knowledge engineer.

Knowledge Engineering

Knowledge Engineering is the **core discipline behind building Expert Systems**. It is the process of designing, building, and implementing an Expert System by **extracting knowledge from human experts** and codifying it into a form a computer can use.

In simple terms, it is the **"bridge" between a human expert's mind and a computer's reasoning system.**
The person who does this is called a **Knowledge Engineer**.

Knowledge Engineering

The Main Steps in the Knowledge Engineering Process

The process is a structured cycle, often involving the following key stages:

1. Knowledge Acquisition

What it is: This is the **most critical and difficult step**. It involves extracting the knowledge, rules, and problem-solving strategies from one or more human experts.

How it's done: The Knowledge Engineer uses techniques like:

- **Interviews** (structured and unstructured)
- **Observation** of the expert solving problems
- **Analysis of case studies and historical data**
- **Protocol analysis** (the expert "thinks aloud" while solving a problem)

Challenge: Experts often have "tacit knowledge"—knowledge so ingrained they don't consciously think about it, making it hard to articulate.

(غالبًا ما يمتلك الخبراء "معرفة ضمنية" - معرفة راسخة لدرجة أنهم لا يفكرون فيها بوعي، مما يجعل التعبير عنها أمرًا صعبًا.)

Knowledge Engineering

2. Knowledge Representation

What it is: This is the process of **structuring the acquired knowledge** into a formal format that the computer can store and manipulate. This is the "blueprint“-"المخطط" of the Expert System's brain.

Common Methods:

Production Rules (IF-THEN Rules): The most common method.

Example: IF the engine does not start AND the lights are dim THEN the problem is a dead battery (with 90% confidence).

Frames: Represent "objects" and their "attributes."

Example: A CAR frame has slots for make, model, engine-type, and symptoms.

Semantic Nets: A graphical representation of knowledge with nodes (concepts) and arcs (relationships).

Ontologies: A more modern and complex way to define the concepts and relationships in a domain.

Knowledge Engineering

3. Knowledge Validation & Verification (V&V)

Verification: "Did we build the system **right**?"

Checking for consistency, completeness, and correctness in the knowledge base and the inference engine. (Are there conflicting rules? Does the logic work?)

Validation: "Did we build the **right system**?"

Testing the system's performance and conclusions against the human expert's decisions to ensure it produces accurate and useful results for real-world problems.

4. Inference Engine Development

While the **Knowledge Base** is the "what" (the facts and rules), the **Inference Engine** is the "how" (the reasoning mechanism).

The Knowledge Engineer helps ensure the inference engine can effectively use the knowledge base. Common reasoning strategies include:

Forward Chaining: Data-driven reasoning. Starts with known facts and uses rules to extract new data until a goal is reached.

Backward Chaining: Goal-driven reasoning. Starts with a hypothesis (a goal) and works backward through rules to see if the available data supports it.

Knowledge Engineering

5. Maintenance and Updating

Knowledge is not static. The knowledge base must be updated regularly to reflect new discoveries, techniques, or changes in the domain.

The Role of the Knowledge Engineer

The Role of the Knowledge Engineer

The Knowledge Engineer is a crucial intermediary. They must have:

Excellent Communication Skills: To interact with domain experts and draw out their tacit knowledge.

Analytical and Modeling Skills: To structure the often-unstructured knowledge from experts.

Understanding of the Problem Domain: While they don't need to be experts themselves, they must understand the domain's concepts well enough to model it.

Technical AI Skills: To choose the right tools and representation methods for the project.

Analogy: Building a Smart Cookbook

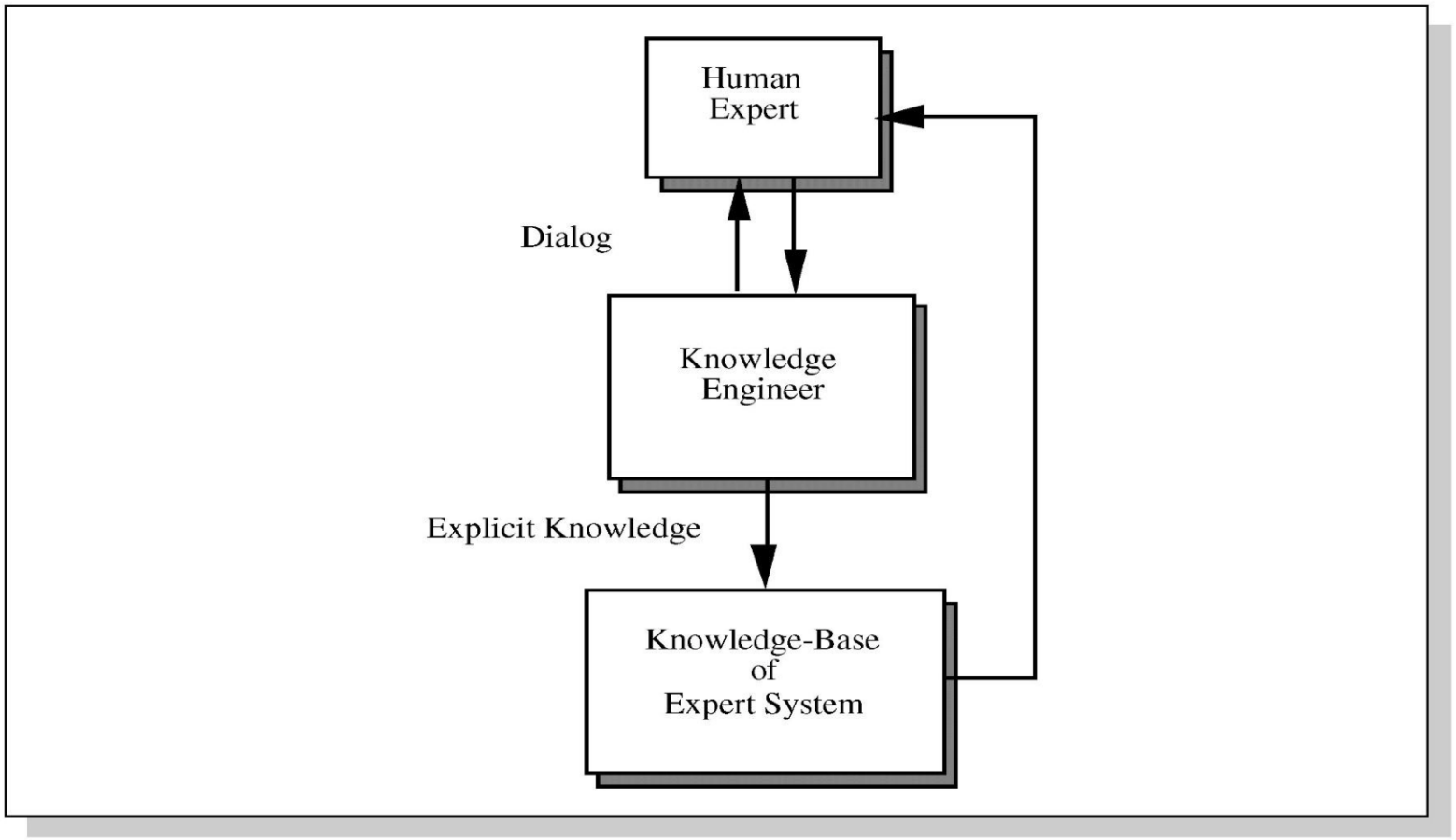
Knowledge Acquisition: A recipe writer (Knowledge Engineer) spends time with a master chef (Domain Expert), watching them cook and asking detailed questions about why they add certain ingredients at specific times.

Knowledge Representation: The writer structures this information into a formal recipe with a list of ingredients (facts) and step-by-step instructions (rules).

Inference & Validation: A home cook (User) follows the recipe (Inference Engine) to create a dish. They validate it by tasting it and comparing it to the chef's description.

In summary, **Knowledge Engineering is the foundational process that transforms human expertise into the structured, rule-based "intelligence" of an Expert System.** While modern Machine Learning automates much of this knowledge acquisition from data, Knowledge Engineering remains vital for creating transparent, reliable systems in well-defined, rule-rich domains.

Development of an Expert System



The Role of AI

- An algorithm is an ideal solution guaranteed to yield a solution in a finite amount of time.
- When an algorithm is not available or is insufficient, we rely on artificial intelligence (AI).
- Expert system relies on inference – we accept a "reasonable solution."

Differences between expert systems and artificial intelligence systems

"**Artificial Intelligence Systems**" is a broad category, and "**Expert Systems**" are a sub-category with a very specific approach. The most common contrast today is between **Expert Systems** and modern **Machine Learning/Deep Learning Systems**.

Knowledge vs. Learning

The fundamental distinction lies in *where the intelligence comes from*.

Expert Systems: Their intelligence comes from **human experts**. Programmers and domain experts painstakingly (accuratelly) translate human knowledge into a set of static rules (e.g., IF [condition] THEN [action]).

Modern AI Systems (e.g., Machine Learning): Their intelligence comes from **data**. They learn patterns and rules automatically by processing vast amounts of data. They are not explicitly programmed for the task. (لم يتم برمجتهم صراحة لهذه المهمه)

Expert System Analogy: A Smart Checklist

Think of an Expert System as an **extremely thorough, interactive checklist** built by the world's best expert.

How it works: It asks you questions, and based on your answers, it follows a pre-defined path to a conclusion.

Strength: It's reliable, logical, and you can see its work. If it says "you have a problem with X," you can trace back through the questions to see why.

Weakness: It's brittle-هش. If it encounters a situation the original expert didn't think to write a rule for, it fails completely. It cannot learn new solutions on its own.

Modern AI System Analogy: A Child's Brain

Think of a Modern AI System (like a neural network) as a **child's brain**.

How it works: You don't give it rules. You show it thousands of examples (e.g., pictures of cats and dogs). Its "neural networks" find patterns and learn the underlying rules for themselves.

Strength: It's highly adaptable and can generalize to new situations it hasn't seen before. It can discover complex patterns too subtle-too accurate for humans to describe with rules.

Weakness: It's a "black box." You can't easily ask a child *exactly how* they know something is a cat; they just do. Similarly, it's hard to know why a neural network made a specific error.

In summary: Expert Systems are a **rule-based, transparent, and specialized branch of AI** where knowledge is explicitly programmed.
Modern AI Systems are typically **data-driven, adaptive, and often opaque غامض** systems where knowledge is learned. Expert Systems are a part of AI's history and still have niche applicationsتطبيقات محدده ومتخصصه, but the field as a whole has largely moved toward machine learning.
ولكن المجال ككل تحرك إلى حد كبير نحو التعلم الآلي.