6. Applications of Artificial Intelligence

(Artificial Intelligence and Neural Networks)

Expert System

Dr. Udaya Raj Dhungana
Assist. Professor
Pokhara University, Nepal
Guest Faculty
Hochschule Darmstadt University of Applied Sciences, Germany
E-mail: udaya@pu.edu.np and udayas.epost@gmail.com

Overview

- Introduction
- Early expert system
- Structure of expert system
- Participant in expert system
- Knowledge Acquisition
- Knowledge Elicitation
- Forward and backward chaining
- Advantages over human expert
- Benefits, limitations and Problems of expert system
- ELIZA
- A simple expert system- Guessing Animals

Expert system

- Definition:
  - According to Darlington:

    “An expert system is a program that attempts to mimic human expertise by applying inference methods to a specific body of knowledge.”

  - According to Turban:

    “An expert system is a system that employs human knowledge captured in a computer to solve problems that ordinarily require human expertise.”

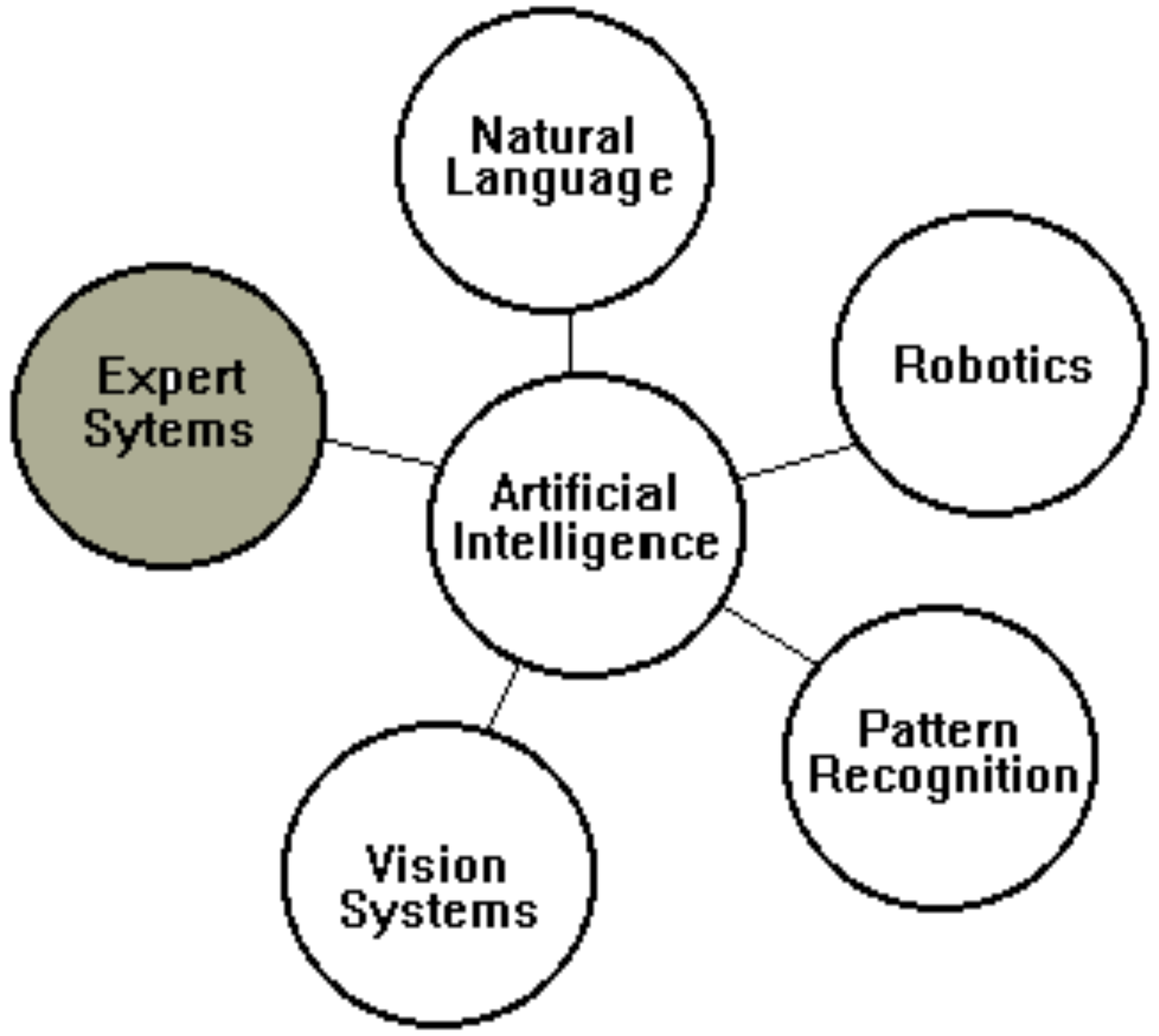
Expert system

- Definition:
  - Edward Feigenbaum
    - “An intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solutions.” [Giarratano & Riley 1998]

- A computer system that emulates the decision-making ability of a human expert in a restricted domain.

Expert system

Historically Expert Systems grew with AI, and can be regarded as a branch of AI.



Expert system

- Introduction
  - Initially, attempts were made to make computers ‘intelligent’ in a *general* way.
  - In 1960s Allan Newell and Herbert Simon wrote computer programs to test the hypothesis that intelligent behavior resulted from heuristic search.
    - GPS (**General Problem Solver**)
      - is aimed at finding general principles of intelligent problem solving.
      - solves simple problems such as the Towers of Hanoi

Expert system

- Early Expert Systems

Pioneer work			
1955	Herbert Simon, Allen Newell	Carnegie Mellon	Logic Theorist: proved theorems using propositional logic
1960	Marvin Minsky John McCarthy Claude Shannon	MIT Dartmouth Bell Labs	LISP

Expert system

- Early Expert Systems

1960	DENDRAL	Feigenbaum & Buchanan (Stanford)	Identify chemical constituents
1970	MYCIN	Stanford	diagnosis of infectious diseases
1970	MACSYMA	MIT	Math expert system
1970	HEARSAY	Carnegie Mellon	Speach recognition
1977	PROSPECTOR	Stanford Research Inst Duda, Hart, Barnett	Mineral diagnosis

Expert system

- The term ***expert system*** is used in a seminal paper by Alan Turing in 1937 related to a study in AI.

- ***An Expert System (ES)*** is a computer program that reasons using knowledge to solve complex problems. (Feigenbaum, 1992)

Expert system

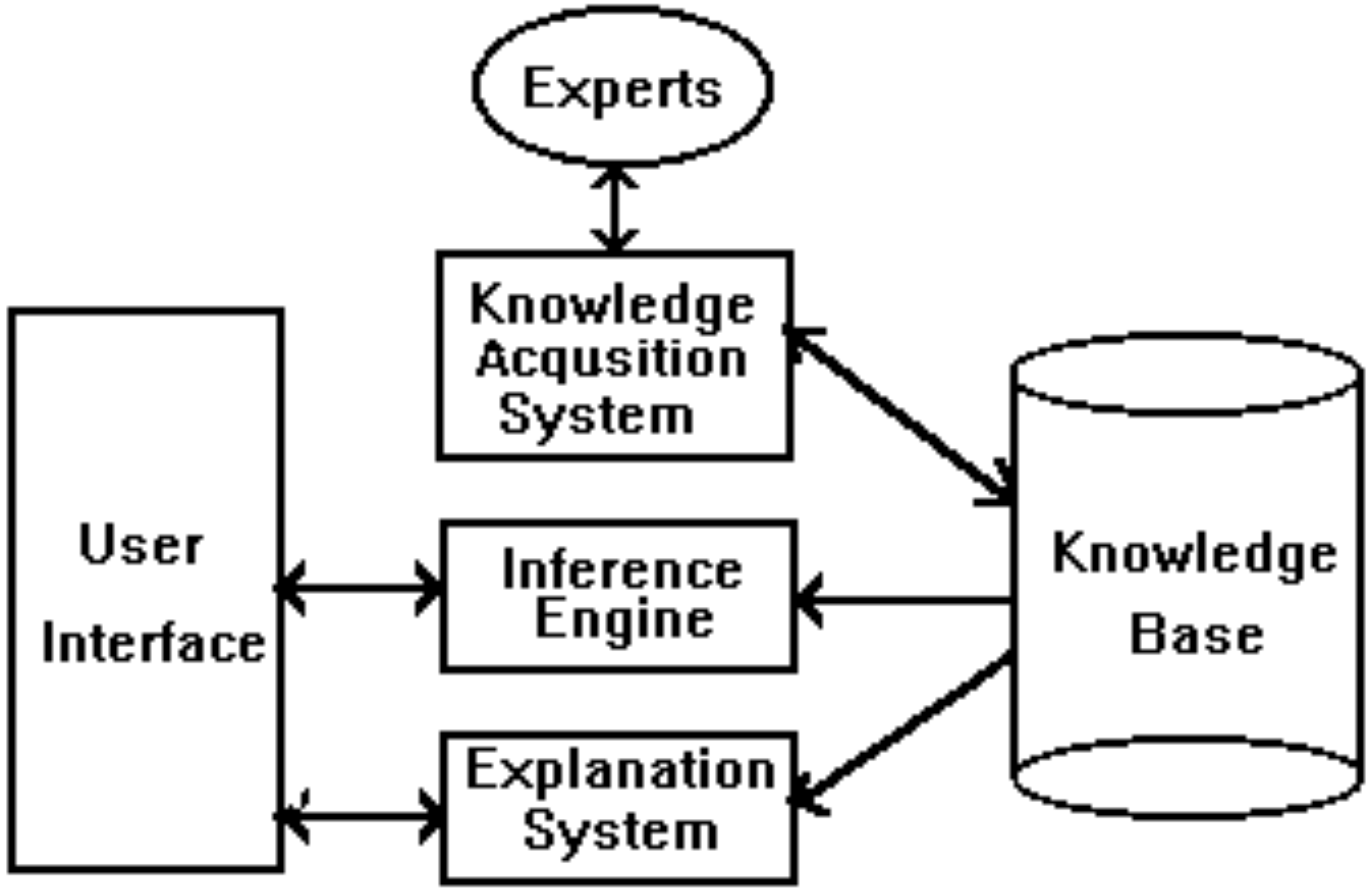
- **Structure of Expert System**
  - An ES will normally have two aspects:
    - A development environment
      - is used by the system builder to modify the system
    - A consultation environment
      - is used by the non-expert to obtain knowledge or advice

Expert system

- **Structure of Expert System**
  - An ES is a program with various components:
    1. Knowledge acquisition subsystem
    2. Knowledge base
    3. Inference engine
    4. User interface
    5. Explanation subsystem

Expert system

- **Architecture of Expert System**



Expert system

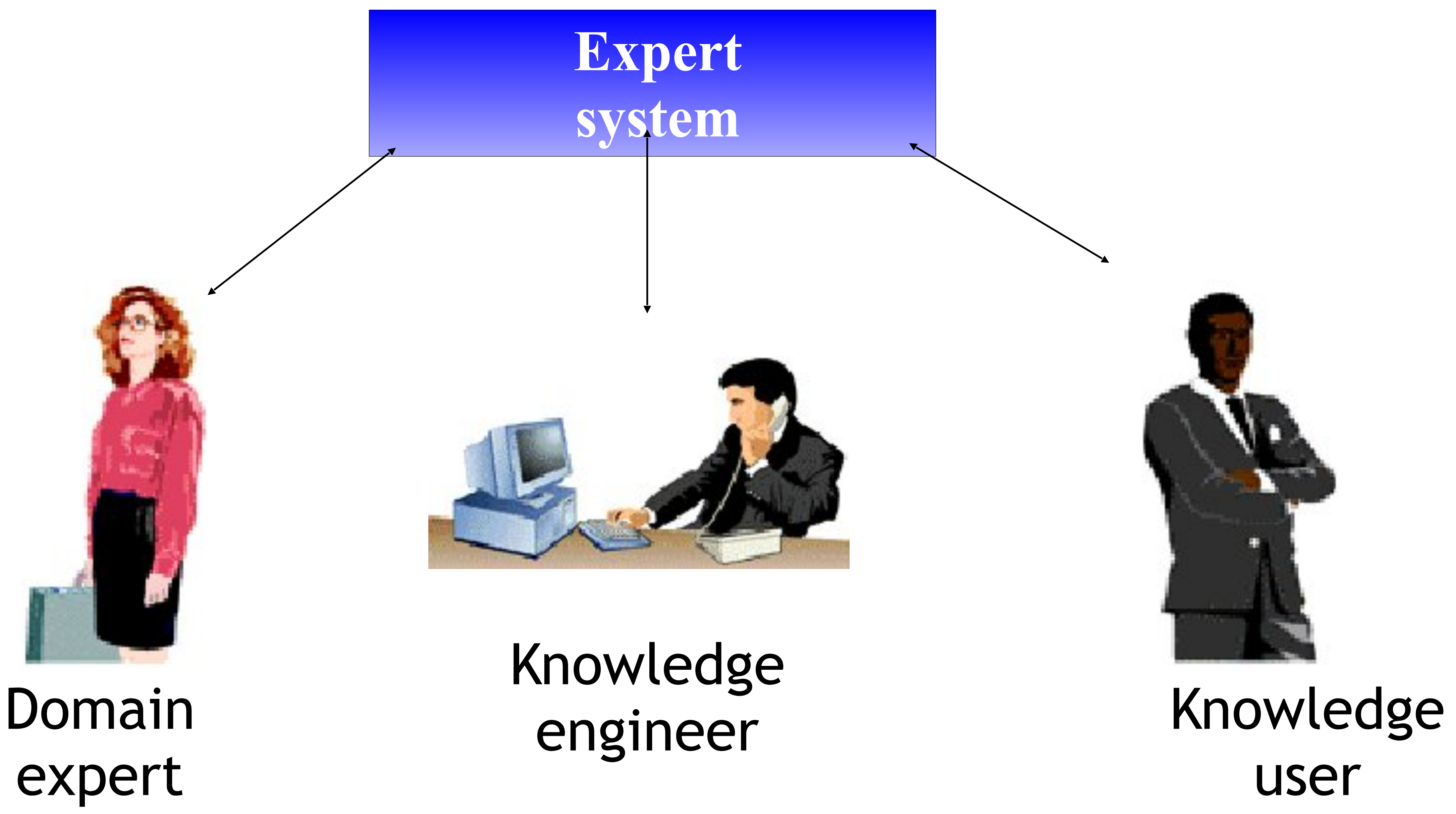
- **Architecture of Expert System**
  - user interface
    - interaction with end users
    - development and maintenance of the knowledge base
  - knowledge base
    - contains essential information about the problem domain
    - often represented as facts and rules
  - inference engine
    - mechanism to derive new knowledge from the knowledge base and the information provided by the user
    - often based on the use of rules

Expert system

- **Architecture of Expert System**
  - knowledge acquisition
    - transfer of knowledge from human experts to computers
    - sometimes knowledge can be acquired directly from the environment
      - machine learning, neural networks
  - Explanation
    - A part of the expert system that allows a user or decision maker to understand how the expert system arrived at certain conclusions or results

Expert system

- Participants in Expert Systems Development



Expert system

Domain expert

Knowledge engineer

Knowledge user

Expert system

- Participants in Expert Systems Development
  - Domain expert
    - The individual or group whose expertise and knowledge is captured for use in an expert system
  - Knowledge user
    - The individual or group who uses and benefits from the expert system
  - Knowledge engineer
    - Someone trained or experienced in the design, development, implementation, and maintenance of an expert system

Expert system

- Forward and Backward Chaining
  - different methods of reasoning and rule activation
    - forward chaining (data-driven)
      - reasoning from facts to the conclusion
      - as soon as facts are available, they are used to match antecedents of rules
      - a rule can be activated if all parts of the antecedent are satisfied
      - often used for real-time expert systems in monitoring and control
      - examples: CLIPS, OPS5
    - backward chaining (query-driven)
      - starting from a hypothesis (query), supporting rules and facts are sought until all parts of the antecedent of the hypothesis are satisfied
      - often used in diagnostic and consultation systems
      - examples: EMYCIN

Expert system

- An ideal ES should include:
  - Extensive specific knowledge from the domain of interest.
  - Extensive database interfaces
  - Application of several techniques.
  - Support for heuristic analysis.
  - Capacity to infer new knowledge from existing knowledge.
  - Decisions under uncertainty
  - Decisions with unknowns
  - Symbolic processing.
  - An ability to explain its own reasoning.

Knowledge Acquisition

- Knowledge acquisition is the process of
    - extracting,
    - structuring and
    - organizing knowledge

  from one source, usually ***human experts***, so it can be used in Expert Systems.
- This is often the major obstacle in building an Expert System.

Knowledge acquisition is the process used to define the rules and ontologies required for a knowledge-based system. The phrase was first used in conjunction with expert systems to describe the initial tasks associated with developing an expert system, namely finding and interviewing domain experts and capturing their knowledge via rules, objects, and frame-based ontologies.

Knowledge Acquisition

- Knowledge acquisition focuses on the extraction of knowledge of the problem domain from experts and other reference sources and the transfer of the expertise to a computer program.
- During expert system development, this process continues until enough problem-solving knowledge is extracted to permit the proposed system to achieve expert performance.

Knowledge Acquisition

- There are three main topic areas central to knowledge acquisition that require consideration in all ES projects:
  - the domain must be evaluated to determine if the type of knowledge in the domain is suitable for an ES
  - the source of expertise must be identified and evaluated to ensure that the specific level of knowledge required by the project is provided
  - if the major source of expertise is a person, the specific knowledge acquisition techniques and participants need to be identified.

Knowledge Acquisition

- The knowledge engineer should consider the following questions:
  1. How can the knowledge necessary for expert-level performance be represented as symbolic data structures for computer use?
  2. How can one achieve flexibility in adding and changing knowledge in the development of a *Knowledge Base*?
  3. How can the knowledge in the specific field be systematically acquired?

https://www.eolss.net/sample-chapters/c08/E3-19-04-02.pdf

Knowledge Acquisition

- The knowledge engineer should consider the following questions:

4. Can such knowledge be discovered by an autonomously acting program?

5. What designs are available for the inference procedure to be used by the intelligent artifact?

6. How can one achieve efficient and accurate performance as the extension of the problem space increases?

https://www.eolss.net/sample-chapters/c08/E3-19-04-02.pdf

Knowledge Elicitation

- Assuming, that we have considered our domain of interest carefully and defined the boundaries of the expert system,
- our first and most crucial stage is knowledge acquisition.
- Knowledge acquisition is the process of
  - getting information out of the head of the expert or from the chosen source and
  - representing it into the form required by the expert system.

http://web.cs.wpi.edu/~jburge/thesis/kematrix.html#_Toc417957386

Knowledge Elicitation

- We can, thus identify two phases of knowledge acquisition process;
  - **knowledge elicitation**, where the knowledge is extracted from the expert, and
  - **knowledge representation,** where the knowledge is put into the expert system.

http://web.cs.wpi.edu/~jburge/thesis/kematrix.html#_Toc417957386

Knowledge Elicitation

- The knowledge engineer is probably not an expert in the domain of interest.
- The engineer's first task is therefore to become familiar with the domain through talking to domain expert and reading relevant background material.
- Once the engineer has a basic level of understanding of domain he or she can begin knowledge extraction.
- It is the job of the knowledge engineer to spot gaps in the knowledge which is being offered and fill them.

http://web.cs.wpi.edu/~jburge/thesis/kematrix.html#_Toc417957386

Knowledge Elicitation

- Knowledge required for the development of an expert system can be extracted (for a domain) from
  - experts,
  - books,
  - magazines and
  - journals.
- When the knowledge extraction is based on the interviews between the domain expert and the knowledge engineer or between the domain expert and the system itself then knowledge extraction is called **Knowledge Elicitation.**

http://web.cs.wpi.edu/~jburge/thesis/kematrix.html#_Toc417957386

Knowledge Elicitation Methods

- Knowledge Elicitation (KE) methods:
  - KE Methods by Interaction Type
  - KE Methods by Knowledge Type Obtained

http://web.cs.wpi.edu/~jburge/thesis/kematrix.html#_Toc417957386

Knowledge Elicitation Methods

- KE Methods by Interaction Type
  1. Interviewing
  2. Case Study
  3. Protocols
  4. Critiquing
  5. Role Playing
  6. Simulation
  7. Prototyping
  8. Teachback
  9. Observation
  10. Goal Related
  11. List Related
  12. Construct Elicitation
  13. Sorting
  14. Laddering
  15. Document Analysis

http://web.cs.wpi.edu/~jburge/thesis/kematrix.html#_Toc417957386

Knowledge Elicitation Methods

- Interviewing:
  - Interviewing consists of asking the domain expert questions about the domain of interest and how they perform their tasks.
  - Interviews can be
    - **unstructured** (open and exploratory; no fixed questions)
    - **semi-structured**
    - **structured** (involve the use of questionnaires to ensure focus)

http://web.cs.wpi.edu/~jburge/thesis/kematrix.html#_Toc417957386

Knowledge Elicitation Methods

- **Case study:**
  - In Case Study methods different examples of problems/tasks within a domain are discussed.
- **Protocols:**
  - Protocol analysis involves asking the expert to perform a task while "thinking aloud."
  - The intent is to capture both the actions performed and the mental process used to determine these actions.

http://web.cs.wpi.edu/~jburge/thesis/kematrix.html#_Toc417957386

Knowledge Elicitation Methods

- **Critiquing:**
  - In Critiquing, an approach to the problem/task is evaluated by the expert.
  - This is used to determine the validity of results of previous KE sessions.
- **Role Playing:**
  - In Role Playing, the expert adapts a role and acts out a scenario where their knowledge is used
  - The intent is that by viewing a situation from a different perspective, information will be revealed that was not discussed when the expert was asked directly.

http://web.cs.wpi.edu/~jburge/thesis/kematrix.html#_Toc417957386

Knowledge Elicitation Methods

- **Simulation:**
  - In Simulation methods, the task is simulated using a computer system or other means.
  - This is used when it is not possible to actually perform the task.
- **Prototyping:**
  - The expert is asked to evaluate a prototype of the proposed system being developed.
  - This is usually done iteratively as the system is refined.

http://web.cs.wpi.edu/~jburge/thesis/kematrix.html#_Toc417957386

Knowledge Elicitation Methods

- **Teachback:**
  - In Teachback, the knowledge engineer attempts to teach the information back to the expert, who then provides corrections and fills in gaps.
- **Observations:**
  - The knowledge engineer observes the expert performing a task.
  - This prevents the knowledge engineer from inadvertently interfering in the process, but does not provide any insight into why decisions are made.

http://web.cs.wpi.edu/~jburge/thesis/kematrix.html#_Toc417957386

Knowledge Elicitation Methods

- **Goal related:**
  - In Goal Related methods, focused discussion techniques are used to elicit information about goals and subgoals. .
- **List related:**
  - The expert is asked to provide lists of information, usually decisions.

http://web.cs.wpi.edu/~jburge/thesis/kematrix.html#_Toc417957386

Knowledge Elicitation Methods

- **Construct Elicitation:**
  - This method is used to obtain information about how the expert discriminates between entities in the problem domain.
  - e.g. Repertory Grid Analysis: the domain expert is presented with a list of entities and is asked to describe the similarities and differences between them
  - These similarities and differences are used to determine the important attributes of the entities.
- **Sorting:**
  - The domain entities are sorted to determine how the expert classifies their knowledge. (**Classification**)

http://web.cs.wpi.edu/~jburge/thesis/kematrix.html#_Toc417957386

Knowledge Elicitation Methods

- **Laddering:**
  - a hierarchical structure of the domain is formed by asking questions designed to move up, down, and across the hierarchy.
- **Document Analysis:**
  - The Document analysis involves gathering information from existing documentation. May or may not involve interaction with a human expert to confirm or add to this information.

http://web.cs.wpi.edu/~jburge/thesis/kematrix.html#_Toc417957386

Knowledge Elicitation Methods

- KE Methods by Knowledge Type Obtained
  - Procedures
  - Problem Solving Strategy
  - Goals/Subgoals
  - Classification
  - Dependencies/Relationships
  - Evaluation

http://web.cs.wpi.edu/~jburge/thesis/kematrix.html#_Toc417957386

Knowledge Elicitation Methods

- **Procedures:**
  - These are methods that can be used to determine the steps followed to complete a task.
- **Problem Solving Strategy:**
  - These methods attempt to determine how the expert makes their decisions
- **Goals/Subgoals:**
  - These are methods that are concerned with extracting the goals and subgoals for performing the task.
  - These methods are listed separately from procedures since ordering is not necessarily provided.

http://web.cs.wpi.edu/~jburge/thesis/kematrix.html#_Toc417957386

Knowledge Elicitation Methods

- Classification:
  - These methods are used to classify entities within a domain.
- Dependencies/Relationships:
  - relationships between domain entities are obtained.
- Evaluation:
  - evaluation of prototypes or other types of KE session results are done

http://web.cs.wpi.edu/~jburge/thesis/kematrix.html#_Toc417957386

Expert system

- Advantages over human experts
  - Always and instantly available and performs the same level of expertise.
  - Has direct and instantaneous access to the necessary databases and is not bound to the limited, biased and imperfect recollections of the human.
  - It is logical, objective and consistent.
  - It doesn’t forget or make mathematical errors.
  - It multiplies the expertise of the firm by being accessible to the other divisions.
  - It is a repository for the storage of the knowledge of those experts from whose input it was developed.

Expert system

- Benefits of expert systems
  - Scarce expertise made available.
  - Integration of expertise from different sources.
  - Improved quality (e.g. where an ES assists in design).
  - Ability to work with incomplete information.
  - Reduced system downtime (ES monitors or finds faults).
  - Training (users gain expertise from the ES).
  - Makes expertise available in remote locations.
  - ES can work faster than people.
  - Reliability (ES will not get tired or bored).

Expert system

- Limitations of expert systems compared to human experts:
  - A human expert is aware of the cultural factors.
  - Human experts are generally aware of the scope and limitations of their knowledge.
  - When faced with a new situation a human expert may develop a new and brilliant approach to solving the problem.
  - People wish to communicate with other people.
  - Human experts are more flexible.

Expert system

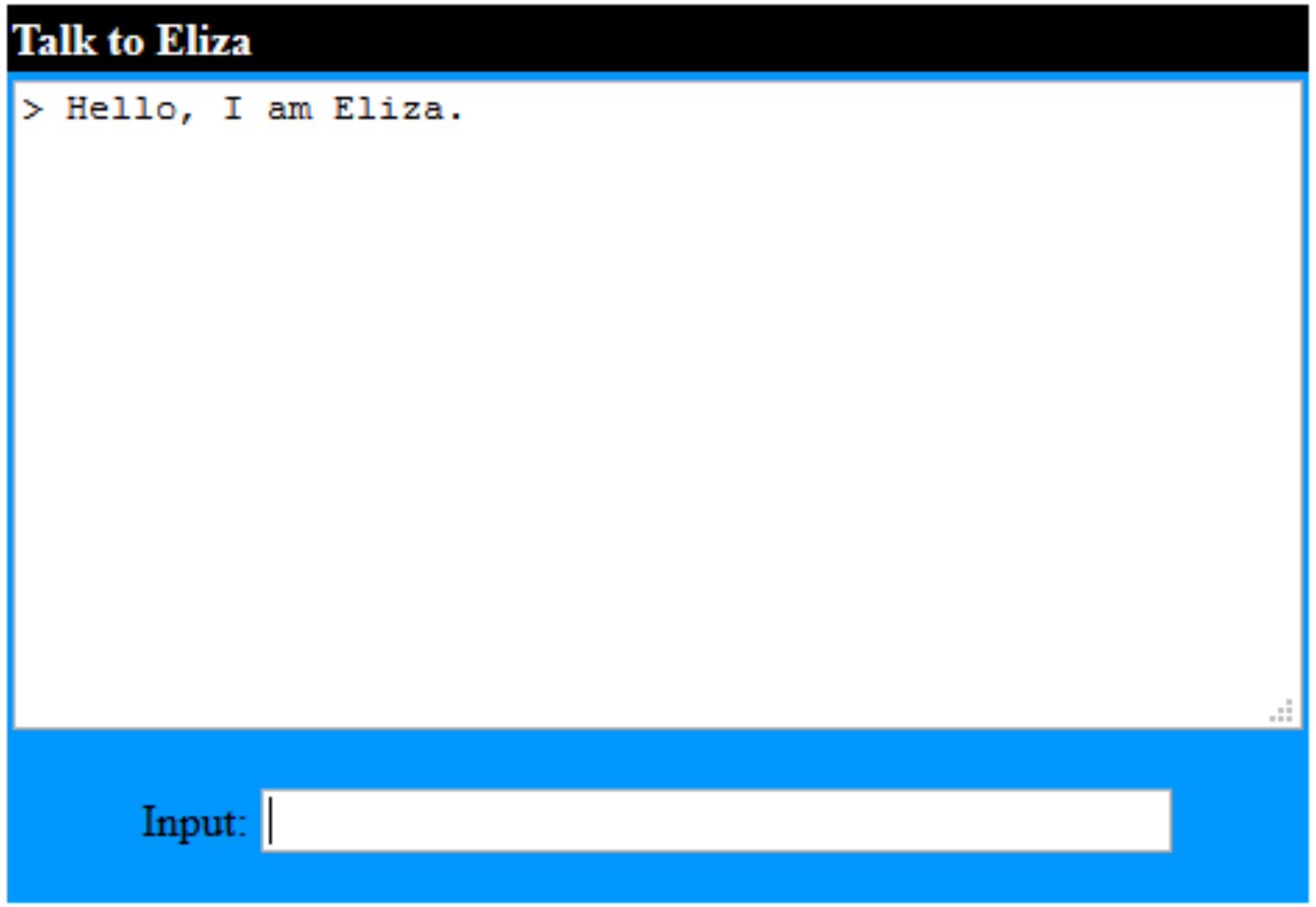
- Problems with expert systems
  - Expert systems are difficult and expensive to develop and maintain.
  - Like all software, ES may contain errors. But unlike other software systems, ES may be designed to cope with incomplete or inconsistent information.
  - If an ES gives a wrong conclusion, it may be difficult to know whether this was caused by an error in the system or by an error in the information given to it.

Expert system

- Problems with expert systems
  - ES are designed to be used by non-experts. As above, they are designed not to fail, so errors may show only in wrong conclusions, and a user without expertise may not be in a position to recognize a wrong conclusion.

Expert system

- ELIZA
  - http://psych.fullerton.edu/mbirnbaum/psych101/Eliza.htm



Talk to Eliza

```
> Hello, I am Eliza.
```

Expert system

- ELIZA
  - an early natural language processing computer program created from 1964 to 1966 at the MIT Artificial Intelligence Laboratory by Joseph Weizenbaum
  - Created to demonstrate the superficiality of communication between humans and machines
  - Eliza simulated conversation by using a 'pattern matching' and substitution methodology

Expert system

- ELIZA
  - While ELIZA was capable of engaging in discourse, ELIZA could not converse with true understanding
  - However, many early users were convinced of ELIZA’s intelligence and understanding

Expert system

- A Simple Expert System - Guessing Animal

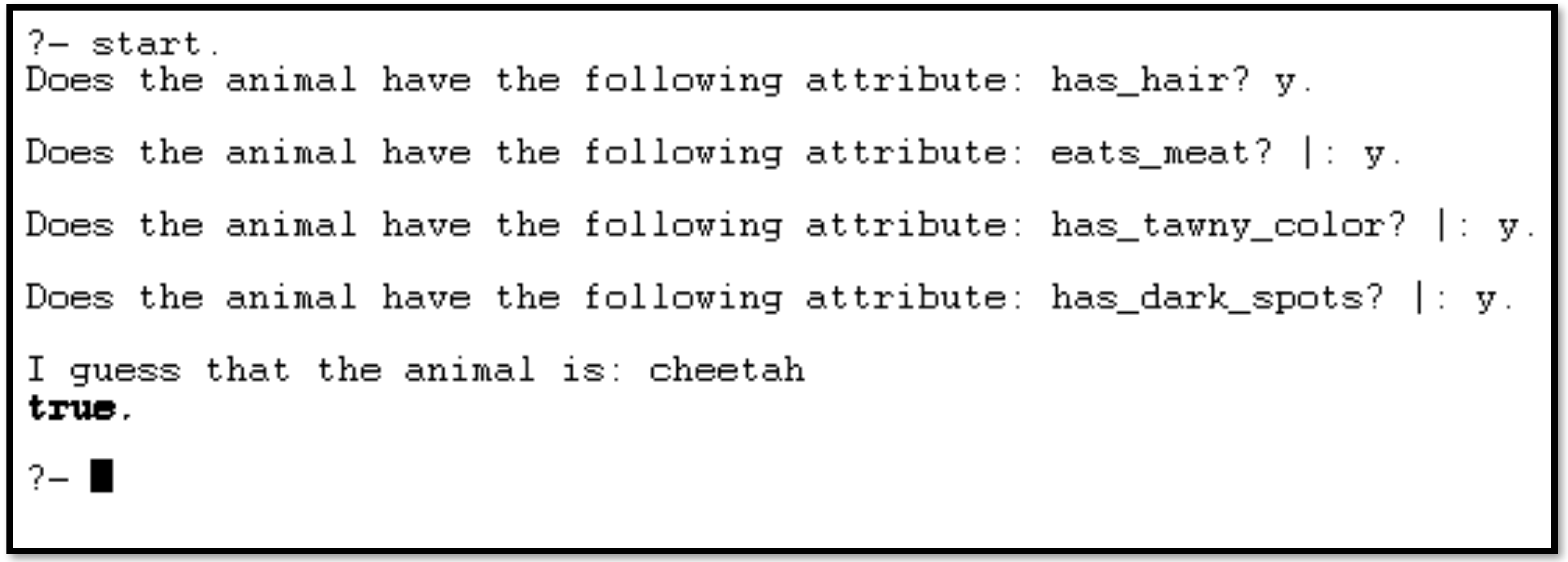
```
start :- hypothesize(Animal),
    write('I guess that the animal is: '),
    write(Animal),
    nl,
    undo.

/*hypotheses to be tested */
hypothesize(cheetah)       :- cheetah, !.
hypothesize(tiger)         :- tiger, !.
hypothesize(giraffe)       :- giraffe, !.
hypothesize(zebra)         :- zebra, !.
hypothesize(unknown).   /* no diagnosis */


/*animal identification */
cheetah  :-  mammal,
             carnivore,
             verify(has_tawny_color),
             verify(has_dark_spots).
tiger    :-  mammal,
             carnivore,
             verify(has_tawny_color),
             verify(has_black_stripes).
giraffe  :-  ungulate,
             verify(has_long_neck),
             verify(has_long_legs).
zebra    :-  ungulate,
             verify(has_black_stripes).
```

```
/*classification rules*/
mammal    :- verify(has_hair),  !.
mammal    :- verify(gives_milk).
carnivore :- verify(eats_meat), !.
carnivore :- verify(has_pointed_teeth),
             verify(has_claws),
             verify(has_forward_eyes).
ungulate  :- mammal,
             verify(has_hooves), !.
ungulate  :- mammal,
             verify(chews_cud).

/*how to ask question */
ask(Question) :-
    write('Does the animal have the
following attribute: '),
    write(Question),
    write('? '),
    read(Response),
    nl,
    ((Response == yes ; Response ==
y)
    ->
      asserta(yes(Question)) ;
      asserta(no(Question)), fail).

:- dynamic(yes/1,no/1).
```

```
/*how to verify something */
verify(S)  :-
    (yes(S)
    ->
    true ;
    (   no(S)
    ->
    fail ;
    ask(S) )).

/* verify(S) :-
    ask(S).  */



/*undo all yes/no assertions*/
undo  :- retract(yes(_)), fail.
undo  :- retract(no(_)), fail.
undo.
```

Expert system

- A Simple Expert System - Guessing Animal



```
?- start.
Does the animal have the following attribute: has_hair? y.

Does the animal have the following attribute: eats_meat? |: y.

Does the animal have the following attribute: has_tawny_color? |: y.

Does the animal have the following attribute: has_dark_spots? |: y.

I guess that the animal is: cheetah
true.

?- █
```

Questions

- What is an Expert system? Write some examples.
- What are the different components of expert system?
- Discuss the participants in expert system.
- Forward chaining vs. backward chaining
- What are the advantages of expert system?
- What are the benefits of expert system?
- What is knowledge acquisition? What is knowledge elicitation and what are its different methods? Explain in brief.

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

THANK YOU

End of Chapter