

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# Introductory Chapter: Enhanced Expert System - A Long-Life Solution

*Petrică Vîzureanu*

## 1. Introduction

What is the definition of an expert system? An expert system belongs to a field of artificial intelligence, and it is a computer program or a software, which can do the same task of a human expert. It can give reliable advice in a specific area of expertise (its domain) and get new conclusions about difficult activities to examine [1]. An expert system can explain its reasoning everytime and is able to interact with a user in the same way that you might consult a human expert. Also, an expert system can be defined as a software program that can outline reasoned conclusions from an amount of knowledge in a specific domain and aims to develop “smart” programs and applications.

The human experts are not 100% reliable in different domains, which can be taken into consideration the advantages and benefits of all accomplished things, but they may disagree with each other or forget to take into account a crucial parameter before making a decision. A human expert can have unsurpassed knowledge in the field and can gain as much knowledge as possible, but be hopeless explaining that to someone else. Human experts cannot be available all the time; i.e., (i) in a small company, the expert on some area can be ill or on holiday; (ii) your doctor may not be able to see you until next week; and (iii) a human expert can move to another company (taking his expertise with him) or retire.

An expert system does not get tired. An expert system (properly programmed) should be 100% reliable and can combine the expertise of several experts. An expert system should be realized to explain and justify all advices it gives. Although an expert system can be expensive to develop, once it is there, its running costs should be low, so there will be economic benefits for the company. An expert system is always available. You can take it with you if you have a notebook or Internet connection, so you could consult an expert system over the Internet. Once the knowledge has been programmed/inserted into the system, it will not be lost if the human expert leaves or dies [2].

Problems with expert systems: limited domain; systems are not always up to date and do not learn, has no “common sense” and experts are needed to setup and maintain the system.

Many applications of expert systems are very well known: Prospector—used by geologists to identify sites for drilling or mining; PUFF—medical system for diagnosis of respiratory conditions; Design Advisor—gives advice to designers of processor chips; MYCIN—medical system for diagnosing blood disorders (first used in 1979); LITHIAN—gives advice to archeologists examining stone tools; and DENDRAL—used to identify the structure of chemical compounds (first used in 1965).

Main components of an expert system: (i) the knowledge base is the collection of facts and rules which describe all the knowledge about the problem domain; (ii) the inference engine is the part of the system that chooses which facts and rules to apply when trying to solve the user’s query; and (iii) the user interface is the part of the system that takes in the user’s query in a readable form and passes it to the inference engine. It then displays the results to the user [3].

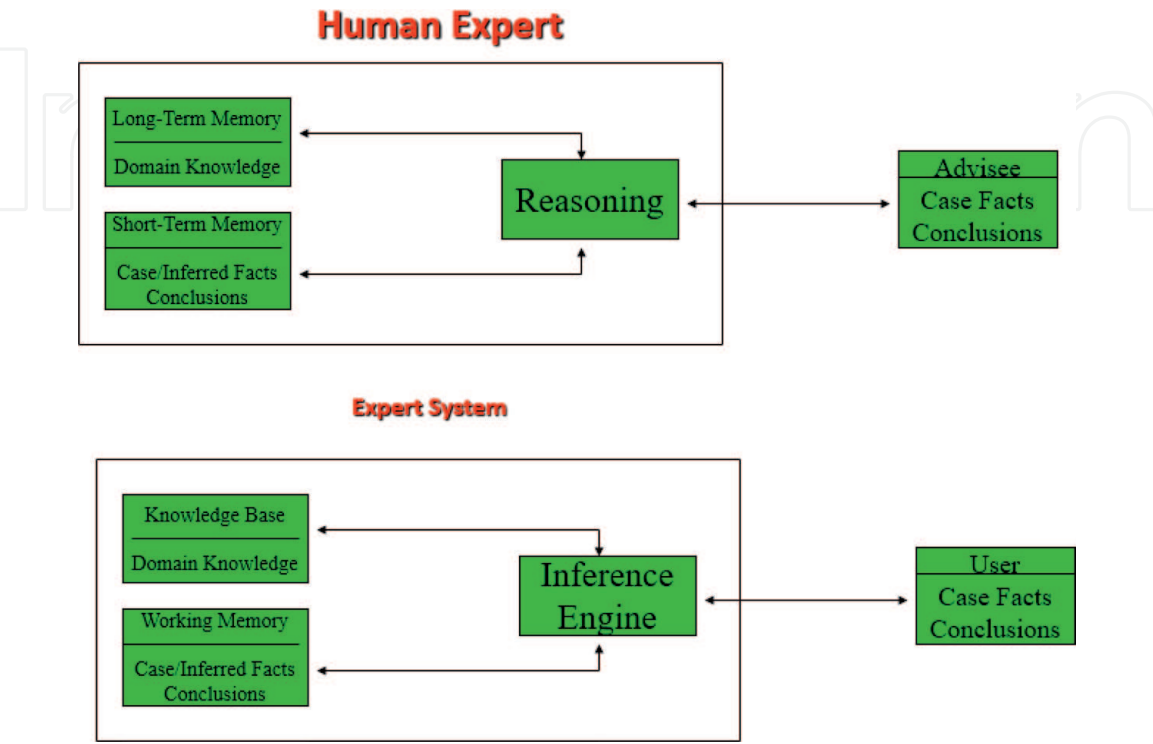


Figure 1.  
General description for human expert vs. expert system.

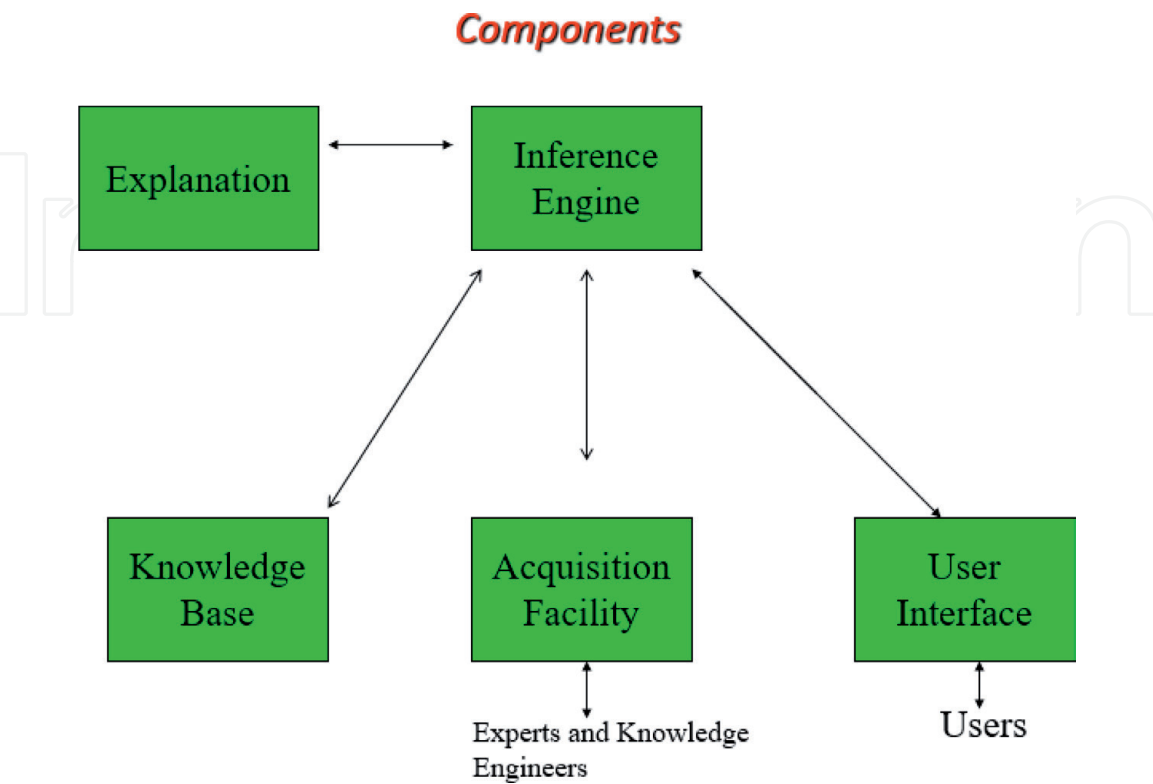
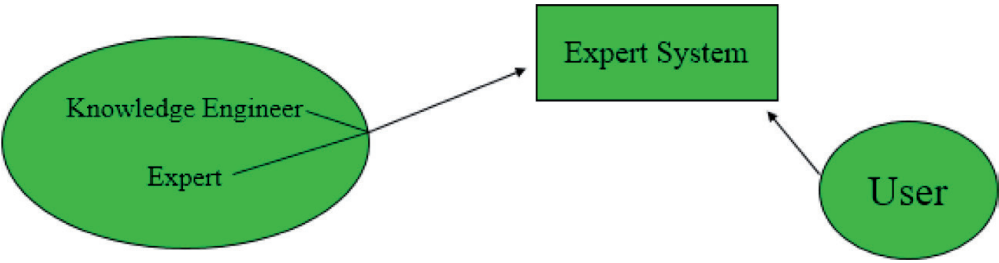
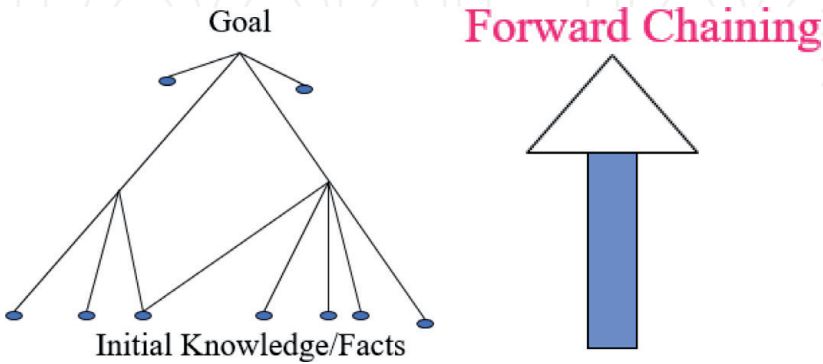


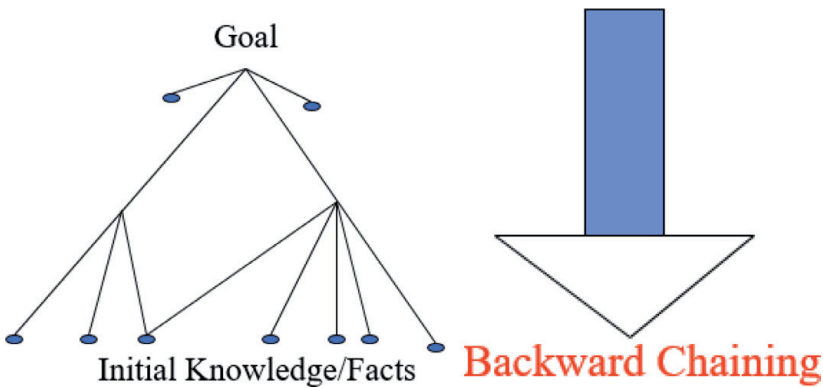
Figure 2.  
Main elements of an expert system.



**Figure 3.**  
*Logical flow inside an expert system.*



**Figure 4.**  
*Forward-chaining technique for an expert system.*



**Figure 5.**  
*Backward-chaining technique for an expert system.*

An expert system behaves similar to a human expert in a field or area. Also, they can be used to solve problems in various fields or disciplines, and can assist in problem solving, but the goal is not to replace the experts, but to provide users an effective tool thereby relieving experts of routine tasks.

An expert system has some capabilities: (i) superior problem solving—only solvable problems, (ii) ability to save and apply knowledge and experience to problems, (iii) reduced time for complex problems, and (iv) looks at problems from a variety of perspectives.

An expert system has five *basic components*: knowledge base, inference engine, explanation component, user interface, and acquisition component.

**Knowledge base** contains the factual and empirical knowledge of experts in a particular subject area and all the facts, rules and procedures, which are important for problem solving.

**Inference engine** simulates the problem-solving strategy of a human expert, represents the logical unit by means of which conclusions are drawn from the knowledge base according to a defined problem-solving method, controls the

execution, which questions to ask and in what order, and simulates the problem-solving process of human experts. Functions of the inference engine are to determine which actions are to be executed between individual parts of the expert system, how they are executed, and in what sequence, to determine how and when the rules will be processed and to control the dialog with the user [4].

**Explanation component** has the role to explain the problem-solving strategy to the user. The solutions must be reproducible by the user and engineer but can only be verified by the human expert: which facts were asked for? why? which facts were vital? and can go as far as, how would the conclusion change if some facts would change?

**User interface** employs natural language for dialogs with the user whenever possible. Questions posed such as “how should questions be answered by the user?”, “how will system responses to these questions will be formulated?”, and “what info is to be graphical?” must be easy to use, erroneous errors kept to a minimum, and questions and answers must be understandable.

**Acquisition component** provides support for structuring and implementation of the knowledge in the knowledge base; hence, it is very important because it allows the engineer to concentrate less on the programming. Knowledge data should be easy to enter, easy to understand methods of representing all info in knowledge base, syntax checks, and access to programming language.

People involved with expert systems are **domain expert** (a person who possesses the skill and knowledge to solve a specific problem in a manner superior to others), **knowledge engineer** (a person who designs, builds, and tests an expert system), and **end-user** (an individual or group who will be using the expert system).

Two main inference techniques are used: forward chaining and backward chaining.

Forward chaining is about the knowledge base searched for rules that match the known facts, and the action part of these rules is performed. The process continues until a goal is reached and puts the symptoms together to reach a conclusion that an ex. doctor has diagnosed in a patient.

Backward chaining starts from a goal, the conclusion. All the rules that contain this conclusion are then checked to determine whether the conditions of these rules have been satisfied. Ex. doctor has an end idea of what is wrong with the patient, though they must prove it by going from the diagnosis and finding the symptoms.

Expert systems have applications, in example, for automatic mapping profiles in software metrics for temporal analysis of programming learning and scoring rubrics. Also, these can be dedicated to a new knowledge representation model, providing convergence of classical operations research and modern knowledge engineering or to a mathematical model that graphically and numerically represents the probabilistic relationships between random variables.

IntechOpen

IntechOpen

### **Author details**

Petrică Vizureanu  
Technical University of Iasi, Romania

\*Address all correspondence to: [peviz2002@yahoo.com](mailto:peviz2002@yahoo.com)

### **IntechOpen**

© 2019 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

## **References**

[1] <https://torry.aberdeen.sch.uk>

[2] <http://www.duncanrig.s-lanark.sch.uk>

[3] <http://www.test-me.co.uk>

[4] Kalogirou SA. Artificial intelligence for the modeling and control of combustion processes: A review. *Progress in Energy and Combustion Science*. 2003;29(6):515-566