EXPERT SYSTEMS

for Personal Computers

M Chadwick and J A Hannah

SIGMA PRESS

Expert Systems for Personal Computers

—

an introduction to artificial intelligence

M. Chadwick and J. A. Hannah

Σ
SIGMA
PRESS

First published in 1986 by

Sigma Press 98a Water Lane, Wilmslow, SK9 5BB, England.

Printed in Malta by Interprint Limited

British Library Cataloguing in Publication Data

Chadwick, Michael
Expert systems for personal computers:
an introduction to artificial intelligence.
1. Experiment systems (Computer science)
2. Microcomputers
I. Title II. Hannah, John Adrian
006.3'3 QA76.9.E96

ISBN 1-85058-044-8

Distributed by

John Wiley & Sons Ltd., Baffins Lane, Chichester, West Sussex, England.

CONTENTS

CHAPTER 1
INTRODUCTION TO EXPERT SYSTEMS

1.1 Stepping into the AI World

Artificial Intelligence (AI) is a rapidly-emerging technology which promises much for the future. It is devoted to programming computers to carry out tasks that would require intelligence if carried out by a human being.

Only a few years ago, the use of AI programs was restricted to large computers, typical of those operating in laboratories or computer centres of universities. But nowadays, personal computers and even home computers are capable of running AI programs. The pioneers of AI research were often regarded as highly-imaginative eccentrics dealing with science fiction-like robots. But a change has occurred. The results of AI research have become very important in many areas and will be available to everyone who uses a microcomputer. The Japanese are working on the fifth generation of computers that will use AI techniques extensively.

The principal AI application areas are the following:

1. Natural Language Processing

Currently, the commands given to computers must often be expressed in a very restricted form. The slightest deviation will cause an error. The computers of the future must be able to communicate in natural languages. Another important task is translation from one language to another.

2. Computer Vision

Tomorrow's computers will be able to see. They will perceive their surroundings through TV cameras and other sensors, extract significant patterns (Pattern Recognition) and will 'understand' what they see.

3. Problem Solving and Planning

Traditional computer programs solve problems following a step-by-step method that is given by the programmer. AI programs will be be able to design a solution method by themselves.

4. Expert Systems

This is probably the hottest topic in AI. An expert system makes a computer act as if it had the knowledge of a human expert in a certain domain.

Principles of AI Programming

The following basic principles are common to nearly all AI programs.

Search

One obvious and often-used method for solving a problem is the trial-and-error search. The *blind search* technique will not be feasible for solving complex problems because the number of possible combinations will grow enormously. AI programs therefore use *heuristic* (rule of thumb) search techniques.

Pattern Recognition

Reacting to changes in the environment requires the ability to recognize certain patterns. The rules used in expert systems, for example, are of the following form: *If* a certain situation occurs, *then* do the following action.

Knowledge Representation

This is one of the most important and most active areas in AI. In order to solve problems, the computer needs an internal model of the world. This model contains, for example, the description of relevant objects and the relations between these objects. All information must be stored in such a way that it is readily accessible. Various models such as property lists, semantic networks, frames, scripts and production systems have already been designed.

Planning

If someone is given a complex task, he usually works out a plan first. No-one would set off on a trip through a foreign country without first consulting a map

before driving off. In the same manner, AI programs work out a general plan. While trying to reach a goal, they often establish subgoals.

Learning
Simulating intelligent behaviour requires the ability to learn. AI programs are able to learn by modifying their internal model. Successful solution strategies can be remembered, unsuccessful ones can be deleted.

1.2 Definition and Applications of Expert Systems

There is no generally valid definition of the term ' expert system', but let us consider the following one:

> An expert system is a computer program that simulates the reasoning of a human expert in a certain domain. To do this, it uses a *knowledge base* containing facts and heuristics, and some *inference procedure* for utilizing its knowledge.

We are dealing with *rule-based* systems, where knowledge is represented by so-called *production rules*. We'll discuss this technique in detail in Chapter 2.

Since the user wants to interact with the expert system, there must be a *user interface*. It has two main functions: it gives advice and explanations to the user (explanatory module) and manages the knowledge acquisition (acquisition module). Figure 1.1 shows this basic structure.

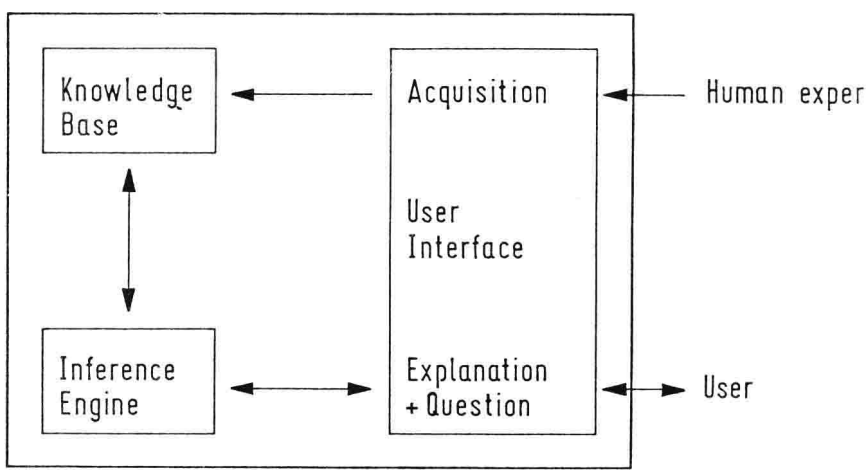

Fig. 1.1 Basic Structure of an Expert System

From a strongly simplified point of view, the following actions take place: first the knowledge of a human expert must be mapped into the knowledge base. This process is also known as *knowledge engineering*. Then the expert system is ready for use. There then follows a dialogue between the user and the system. The user responds to the the system's questions and is given advice or a final answer. The inference engine analyses situations, establishes subgoals and draws conclusions.

Applications

Since the mid-1970 s, a great variety of expert systems have emerged. In the next section we will discuss some important systems that have already proved successful. Here is a list of some problem domains suitable for expert systems:

- medical diagnosis

- electronic circuit diagnosis

- automobile engine fault diagnosis

- interpretation of physical or chemical data

- mineral exploration

- military defense

- air-traffic control

- computer-aided instruction

- computer configuration selection

- understanding speech

- photo interpretation

- intelligent information retrieval

- automatic programming

1.3 Examples of Expert Systems

The following survey presents a selection of well-known expert systems that have been installed and used successfully. For further reading, we recommend the references at the end of the book.

MYCIN

The MYCIN system was designed for diagnosis and therapy recommendation for infectious diseases. The medical knowledge is represented as a set of production rules. For example:

IF 1) the infection is primary-bacteremia
2) the site of the culture is one of the sterile sites
3) the suspected portal of entry of the organism is the gastrointestinal tract

THEN There is suggestive evidence (0.7) that the identity of the organism is bacteroides.

The system is able to explain its reasoning and can therefore be used in medical teaching. The conclusions drawn can be rated on a confidence scale of -1.0 to $+1.0$, where -1.0 means 'certainly wrong' and $+1.0$ 'means certainly right'. In the above example, the conclusion is asserted with a mild degree of certainty of 0.7.

The inference engine uses the backward-chaining technique. To achieve a certain goal it establishes one or more subgoals. We will discuss and program this method later.

For the acquisition and application of new knowledge, another system called TEIRESIAS is used. It works in conjunction with MYCIN and provides interaction with a human domain expert.

A further development of MYCIN is the emycin SYSTEM (empty MYCIN). It uses the control structure of MYCIN but has the knowledge base removed. This base can be substituted by another, and thus EMYCIN is a system for building expert systems in any domain.

DENDRAL

The DENDRAL system is capable of interpreting data from a mass spectrometer. It is widely used by research chemists and has produced a number of significant results. Actually, the DENDRAL system consists of several DENDRAL programs:

1. Heuristic DENDRAL
This part is given the mass spectrum and the atomic constituents of a molecule. The knowledge base contains production rules of the IF-THEN form. The inference engine uses the forward-chaining technique.

2. CONGEN
This part is the CONstrained GENerator. It generates a list of structures consistent with some given information.

3. Meta-DENDRAL
The task of this program is to infer rules from empirical data. It is capable of modifying existing rules and creating new ones.

The DENDRAL system is probably one of the best examples of an expert system that has left the AI research domain and entered the field of commercial application.

MACSYMA

The MACSYMA system has been designed for solving mathematical problems, for example integration, differentiation, solution of equations and systems of equations, vector algebra and matrix operations. In contrast to other systems (e.g. MYCIN), it has no explanation facilities and no reasoning under uncertainty.

PROSPECTOR

The PROSPECTOR system has been developed to assist geologists in mineral exploration. It matches given data against internal models describing important types of ore deposits. Like MYCIN, it can reason with uncertain and incomplete data. It is capable of explaining its reasoning processes.

The geological knowledge is represented in semantic networks. New knowledge is acquired through the KAS knowledge acquisition system.

PROSPECTOR uses a convenient user interface. It has proved very successful by predicting several deposits.

EXPERT

The EXPERT system is an expert system design tool. Though it was designed independently of any specific application, it is best suited for classification problems.

The user creates a file that contains statements describing a certain model. These statements are error-checked and compiled. The model may then be executed.

The model consists of three sections: hypotheses, findings and decision rules. Findings are the facts (e.g. symptoms, test results) input by the user, hypotheses are the conclusions drawn from these facts. After an assertion about a finding has been made, the corresponding rules are evaluated. This system is able to explain its reasoning. If, for example, during an automobile engine fault diagnosis the conclusion 'battery discharged' is reached, the system can explain what rule it used:

IF: Starter Data: No cranking

Simple Checks: Headlights Are Dim

THEN: Battery Discharged (0.9)

The number 0.9 is a confidence factor; the closer it is to 1, the greater the confidence.

CHAPTER 2
PRINCIPLES OF RULE-BASED EXPERT SYSTEMS

2.1 Production Rules

The origin of production systems dates back to 1943, when they were proposed by Post for the first time. Since then they have undergone various modifications, but the basic idea has remained the same: the knowledge is encoded in a declarative form which comprises a set of *rules* of the form

situation → action

These situation-action rules are also called *production rules* or *IF-THEN rules*. A system that uses this form of knowledge representation is called a *production system*.

The basic underlying idea that led to this form of representation is derived from the observation that human experts often think in IF-THEN patterns when solving a problem.

Example:

Your car won't start. Your reasoning process might be of the following form:

First I'll check the starter.

IF the starter doesn't work THEN the battery might be flat.

But there may be other reasons for the starter not working. It might be broken, while the battery is in order. To find out what's wrong you have to check for

other symptoms. For example, you combine two or more symptoms (or facts, or situations, or beliefs etc.) using the logical operator AND:

IF the starter doesn't work
AND the headlights are dim
AND the ignition and oil pressure lights don't come on
THEN the battery might be flat

Notice that you can't be absolutely sure your conclusion is right, because the battery connections might be poor. So you could state the conclusion in the following form:

THEN the battery might be flat OR the battery connections poor.

It would also be possible to use a *confidence factor* (probability factor) and say:

THEN there is strong evidence (0.9) that the battery is flat.

The factor can range from 0.0 (certainly wrong) to 1.0 (certainly right). This kind of reasoning is often called 'inexact reasoning' or 'reasoning with uncertainty'.

The above example has illustrated that human reasoning often consists of IF-THEN rules. It was therefore obvious to encode the expert's knowledge in such a form. The rules have the following general form:

IF <antecedent 1>
<antecedent 2>
<antecedent m>
THEN <consequent 1> [with certainty C1]
<consequent 2> [with certainty C2]
<consequent n> with certainty Cn

The brackets denote that the certainty factors C1...Cn are optional.

The IF part of a production rule is also called the

condition part
left-hand side (LHS)
patterns

The THEN part of a production rule is also called the:

action part
right-hand side (RHS)
actions

We say that a production rule fires if the condition part is satisfied. This means that the designated actions will be performed.

2.2 Production System

A production system consists of three parts:

1. rule base
2. data base
3. rule interpreter

In Figure 2.1 the interaction of these components is shown in diagram form. The data base contains symbols (e.g. facts, assertions, answers). The contents of the data base represent the state of the production system. The interpreter scans the condition parts of each rule until one is found that can be fired. The firing changes the state of the data base by adding or replacing symbols. Then the next cycle starts, and the interpreter tries to find another rule that can be fired. The execution ends if no rules are applicable.

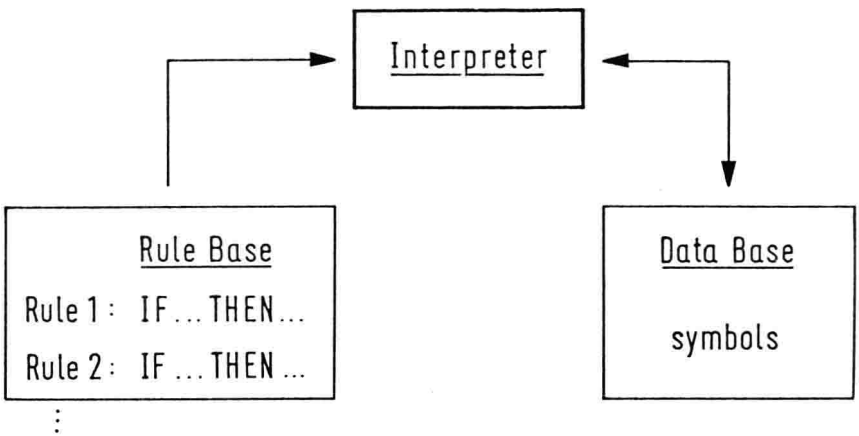

Fig. 2.1 The basic components of a production system.

Unfortunately the terminology is not uniform. Here's a selection of terms commonly used:

The rule base is also called:	knowledge base
The data base can be called:	global data base
	working memory
	short term memory (STM)
	context list (CL)
The interpreter is called:	inference engine
	inference system

We will keep to the terms 'rule base', 'data base' and 'interpreter' throughout this book.

The cycling of the interpreter is also called:

select-execute loop
recognize-act loop
situation-action loop

All these terms describe the same thing with different words. The production system reacts to changes in its data base by applying (or firing) some rules and thereby modifying the contents of the data base.

2.3 A Sample Production System

The best way to understand how a production system works is to take a really small set of production rules and reproduce the interpreter's cycling with pencil and paper. That's exactly what we are going to do now.

Let's consider a system for identifying drinks. There are five of these: beer, wine, grape juice, mineral water and lemonade. The rule base consists of the following set of rules:

R1: IF for children THEN non-alcoholic
R2: IF for drivers THEN non-alcoholic
R3: IF non-alcoholic THEN thirst-quenching
R4: IF ideal when hot THEN thirst-quenching
R5: IF thirst-quenching AND for adults only THEN beer
R6: IF made from grapes AND for adults only THEN wine
R7: IF made from grapes AND taste of fruit AND non-alcoholic THEN grape-juice