EXPERT SYSTEMS

One of the largest area of applications of artificial intelligence is in expert sytems, or knowledge based systems as they are often known. This type of system seeks to exploit the specialised skills or information held by of a group of people on specific areas. It can be thought of as a computerised consulting service. It can also be called an information guidance sytem. Such systems are used for prospecting medical diagnosis or as educational aids. They are also used in engineering and manufacture in the control of robots where they inter-relate with vision systems. The initial attempts to apply artificial intelligence to generalised problems made limited progress as we have seen but it was soon realised that more significant progress could be made if the field of interest was restricted.

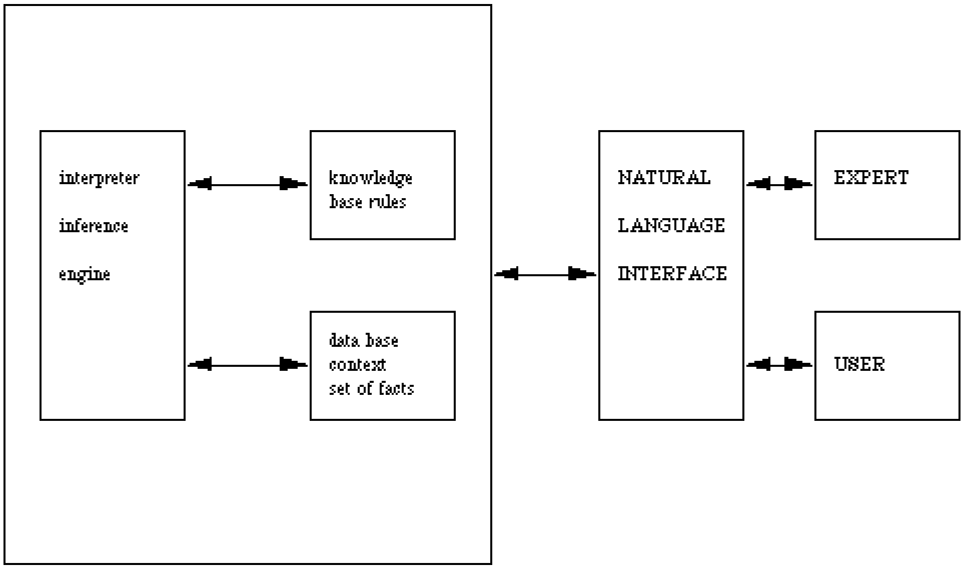
STRUCTURE

The internal structure of an expert system can be considered to consist of three parts:

the knowledge base ; the database; the rule interpreter.

This is analagous to the production system where we have

the set of productions; the set of facts held as working memory and a rule interpreter.



The knowledge base holds the set of rules of inference that are used in reasoning. Most of these systems use IF-THEN rules to represent knowledge. Typically systems can have from a few hundred to a few thousand rules.

The database gives the context of the problem domain and is generally considered to be a set of useful facts. These are the facts that satisfy the condition part of the condition action rules as the IF THEN rules can be thought of.

The rule interpreter is often known as an inference engine and controls the knowledge base using the set of facts to produce even more facts. Communication with the sytem is ideally provided by a natural language interface. This enables a user to interact indelendently of the expert with the intelligent system.

OPERATION OF THE SYSTEM

Again there are three modes to this:

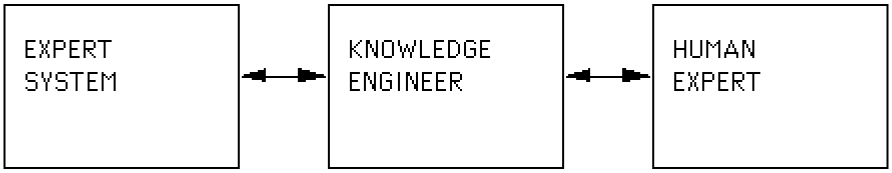
the knowledge acquisition mode;

the consultation mode;

and the explanation mode.

We shall consider each in turn.

KNOWLEDGE ACQUISITION



The system must liaise with people in order to gain knowledge and the people must be specialised in the appropriate area of activity. For example medical doctors, geologists or chemists. The knowledge engineer acts as an intermediary between the specialist and the expert system. Typical of the information that must be gleaned is

vocabulary or jargon, general concepts and facts, problems that commonly arise, the solutions to the problems that occur and skills for solving particular problems. This process of picking the brain of an expert is a specialised form of data capture and makes use of interview techniques. The knowledge engineer is also responsible for the self consistency of the data loaded. Thus a number of specific tests have to be performed to ensure that the conclusions reached are sensible.

CONSULTATION

The system is in this mode when a user is interacting with it. The user interacts by entering data in English and the sytem responds by using its inference engine to carry out the reasoning involved in deriving an answer to the questions posed by the user. The role of the inference engine is to search for facts that match the condition part of the productions that match the action part of the question. There is scope for the use of heuristics in this role. The following example is hypothetical but illustrates the action of typical systems.

System: What is the patient's name?

User: George Washington

System: What is the patient's age?

User: 252 years

System: Is the patient male or female?

User: Male

System: What are the symptoms?

User: Headache dizziness upset stomach high fever

System: When did these symptoms first appear?

User: December 13,1799

System: I conclude that the patient has a viral infection , my recommendation is that he should take two aspirin, drink plenty of fluid, get lots of rest and call me in the morning.

George Washington died the next day of a viral infection no antibiotics

EXPLANATION

This mode allows the system to explain its conclusions and its reasoning process. This ability comes from the AND/OR trees created during the production system reasoning process. As a result most expert systems can answer the following why and how questions

why was a given fact used?

why was a given fact not used?

how was a given conclusion reached?

how was it that another conclusion was not reached?

Let us illustrate the action of an expert system by considering the behaviour of an actual one MYCIN a medical expert system.

Mycin has been informed that a patient has contracted meningitis while in hospital. Doctors wish to interrogate Mycin to check this diagnosis.

Mycin then asks the doctors this question

Mycin: Is Meningitis a hospital acquired illness?

Doctor: why?

Mycin: [2.0] this will aid in determining the orgamisms that might be causing the infection

It has already established

[2.1] the infection that requires therapy is meningitis

[2.2] organisms were not seen on the stain of this pending csf culture

[2.3] the type of infection is bacterial

[2.4] neurosurgery is not one of the relevant items of patient 100

[2.5] the age of patient 100 is greater than 25

therefore if [2.6] the infection was acquired while the patient was in hospital then there is evidence that the organisms that might be causing the infection are E coli(0.75) staphylococcus-coag-pos(0.3) pseudomonas-aeruginosa(0.3)

klebsiella-pneumoniae(0.5)

If the doctors wanted to know how something was concluded they might ask

Doctor: How 2.3

The system then reponds with:

Mycin: The following rules concluded about the likelihood that the type of the infection is bacterial:

[2.7] RULE148 (0.19)

[2.8] RULE500 (0.51)

[2.9] RULE501 (0.95)

[3.0] RULE502 (0.97)

[3.1] RULE526 (0.98)

[3.2] RULE504 (0.97)

[3.3] RULE524 (0.98)

The system takes the doctors through the production system AND/OR tree to explain its conclusions. The numbers in brackets are called certainty factors because little can be absolutely certain. The certainty factors range from -1 to +1 where -1 means absolutely not and +1 absolutely so and 0 undecided either way.

A BLACKBOARD SYSTEM

HEARSAY uses the concept of a blackboard sytem where each partition or subsystem is treated as a specialist. Each specialist is a set of production rules in its own right. It is illustrated below.

Hearsay is a speech recognition system that is about 90% accurate. The specialisms for Hearsay are also illustrated below.**TEIRESIAS**

Human experts communicate using TEIRESIAS with the performance program MYCIN to discover how well MYCIN is doing in repsonding to the requests for help that it receives. Through using TEIRESIAS the expert can coach MYCIN into producing better responses so that an element of learning is introduced.

TEIRESIAS recognises that some of the knowledge required for the knowledge base is inexact or rough and it succeds in offering advice for its correction. It also provides method for extracting the best performance out of an expert system such as MYCIN.

TRANSFER OF EXPERTISE

It is well known that an expert knows more about a topic than is appreciated and cetainly more than the expert is able to offer at short notice. Thus questions such as

'tell me all you know about staph infections' will yield only a fraction of an expert's knowledge.

TEIRESIAS's approach is to make use of the expert's ability to correct incorrect or partially correct responses to common questions and then to focus the expert's reply as follows

'what do you know that the program doesn't know that makes your expert diagnosis different in this case?'

Interactive transfer of expertise between an expert and an expert system begins when the expert identifies an error in the performance of the program and invokes TEIRESIAS to help track down and correct the error.The expert can spot errors when the program produces responses that the expert would not have made or lines of reasoning appear inappropriate.

Typical errors might be the wrong identification of a bacterium, or when MYCIN asks a question that in the expert's view would not assist in the identification of the bacterium. Both kinds of error indicate a bug in the knowledge base of MYCIN and transfer of knowledge occurs when TEIRESIAS is called in to correct the deficit.

TEIRESIAS fixes bugs by

1 stopping the performance program when the human expert identifies an error;

2 working backwards through the steps in the performance program that led to the error until the bug is found;

3 helping the expert fix the bug by modifying the knowledge base.

The expert can find faulty reasoning by using how and why commands to ask TEIRESIAS to back up through previous steps explaining why the steps were taken. This process can be used even if there is no apparent bug in the program as it often helps the user understand the mechanisms of the performance program. As these program s are often involved in comples doagnoses then this facility is very useful.

TEIRESIAS's techniques for generating explanations are based upon two assumptions

The two assumptions used are

1 that a recapitulation of program actions can be an effective explanation as long as the correct level of detail is given

2 that there is some shared framework for viewing the program's actions that will make them understandable to the user.

These assumptions are valid for MYCIN but there are expert systems in which this is not the case.

The first assumption simplifies the task because it requires only the ability to record and play back the history of events. But what level of detail is needed. If a program reasons symbolically repeating the arguments may be beneficial but if there are numerical calculations then it is quite different.

The repetition must be

detailed enough to enable each operation cited to be understandable

high enough in conceptual level to enable each operation cited to be meaningful

complete enough to allow the operations cited to be sufficient to account for all behaviour.

Knowledge acquisition Rule models and schemas

When the expert has identified a deficit in the knowledge base of the performance program, TEIRESIAS questions the expert in order to correct the deficit. This process relies heavily on meta level knowledge about the performance program, encoded in rule models and schemas. In other words, TEIRESIAS uses these data structures to represent knowledge about what the performance program knows. The meta-knowledge about objects in the domain includes structural and organisational information and is specified in data structure schemas. Inquiries in simple English about the values of the schemas components makes the knowledge acquisition process appear to the expert as a natural high-level query about the new concept. Knowledge acquisition about new objects takes place as a process of instantiating the schema.

TEIRESIAS's rule models are empirical generalisations of subsets of rules, indicating commonalities among the rules in that subset. For example, in MYCIN there is a rule model for the subset of rules that conclude affirmatively about organism category, indicating that most such rules mention the concepts of culture site and infection type in their premise. Another rule model notes that mention site and infection type in the premise also tend to mention the portal of entry of the organism. An example of TEIRESIAS's use of rule models in its knowledge acquisition dialogue is given in the sample protocol below.

IF (1) the infection is a pelvic-abscess, and

(2) there are rules that mention in their premise Enterovacteriaceae, and

(3) there are rules that mention in their premise gram positive rods,

THEN There is suggestive evidence (.4) that the rules dealing with Enterobacteriaceae should be evoked before those dealing with gram positive rods.

An Example: THEIRESIAS in the Context of MYCIN

We now illustrate THEIRESIAS's operation in affiliation with the MYCIN system (see previous lecture), paying particular attention to the explanation and knowledge-acquistion facilities of TEIRESIAS's. MYCIN is intended to provide physician with advice about the diagnosis and drug therapy for bacterial infections. The user interacts with TEIRESIAS, which in turn communicates with the MYCIN system, although the user is unaware of more than one program being involved. The system asks questions about the patient, the infection, the cultures grown from specimens from the patient, and any organisms growing in the culture.

MYCIN's knowledge base is composed of rules that specify a situation involving a sick patient and the conclusions that can be drawn in that situation.

For example to conclude whether a patient suffers from a bacterium in the Enterobacteriaceae category MYCIN invokes rule 95.

IF The site of the culture is blood, and

the gram stain is positive, and

the portal of entry is gastrointestinal tract, and

[A-the abdomen is the locus of infection, or

B-the pelvis is the locus of infection]

THEN There is strongly suggestive evidence that Enterobacteriaceae is the class of organisms

for which therapy should cover

The procedure of working backward from the conclusion the situation that would make the conclusion true is called backward chaining and is MYCIN's fundamental reasoning mechanism. Backward chaining can go several levels deep, meaning that the causes in a situation that must be true (for the conclusion to be true) might themselves be conclusions of other rules. For example to identify Enterobacteriaceae a condition is that the infection be in the abdomen or pelvis and this is itself a conclusion of another rule.

See the protocol showing MYCIN's basic style of interaction (the physician's responses are prefixed by ** which refers to PATIENT 281 ALBERT HALL on the handout)

This then concludes

My therapy recommendations will be based on the following probable infection(s) ad potential causative organism(s):

INFECTION-1 is PRMARY-BACTEREMIA

<Item-1> E.COLI [ORGANISM-1]

<Item-2> PSEUDOMOS-AERUGINOSA [ORGANISM-1]

Summarising : Mycin constructs a goal tree of questions that must be resolved in the course of a consultation to conclude the identity of a bacterium. When it cannot deduce anything from what it knows then it asks the respondent. Each node in the tree has subnodes that result from the application of a rule. see figure B1 on handout

Explanation: HOW and WHY. Suppose that during a consultation with MYCIN the clinician wants an explanation of the reasoning behind some question that MYCIN has asked. The explanation given by TEIRESIAS in response to the physician's question, WHY, involves backing up the goal tree to the previous node and presenting the rule used to derive the subnode from its ancestor.

Consider the dialogue in the handout.

(Again, the physician's responses are prefixed by **.)

In contrast to WHY, which spells out "higher" nodes in the goal tree, the HOW command asks TEIRESIAS to explain the nodes lower down the tree that have been or will be expanded. For example, in the following exchange, the expert asks how MYCIN will determine whether the abdomen or the pelvis is the locus of infection.

** HOW [3.4]

Note: (3.4) is the prefix of the statement of interest in the exchange above.

[I.e., how were you determining that the Abdomen is the locus of infection, or that the Pelvis is the locus of infection?]

[4.0] At that point Rule 021 was being used.

If the doctor now wants to see rule 21, he asks TEIRESIAS to explain HOW it was used, referring to the prefix [4.0] of the statement in which it was mentioned.

** HOW [4.0]

[i.e. how was Rule 21 used?]

[5.0] it has already been established that

[5.1] the culture is recent

therefore if

[5.2] there is therapeutically significant disease associated with

the occurrence of this organism.

THEN

It is definite (1.0) that the site of the culture is the locus of infection in the patient

The explanation facilities of TEIRESIAS involve climbing up and down MYCIN's goal tree, explaining why questions are asked and how they are answered. If MYCIN makes a misdiagnosis, these same facilities are used to back up through the rules that achieved the diagnosis until the expert has found the faulty one or determined that one is missing. TEIRESIAS is then used to expedite the alteration or construction of rules by the expert. (See Article IX.C8 for a discussion of how a similar explanation facility is used in the tutorial system EXCHECK.)

Knowledge acquisition

TEIRESIAS can help an expert formulate rules in English because it has expectations about what rules "look like." These expectations are stored internally as rule models that are formed automatically by examining all the rules in the knowledge base. TEIRESIAS knows, for example, that there is a class of rules that have as their conclusions the probable identity of a bacterium and that these rules typically have as conditions (a) the site of the culture, (b) the stain of the culture, and (c) the probable portal of entry of the bacteria. These rule models help TEIRESIAS translate English versions of new rules into a machine representation acceptable to MYCIN.

To illustrate TEIRESIAS's debugging and correction facilities, we will continue with the consultation shown earlier. Suppose that after MYCIN presents the expert with its diagnosis, THEIRESIAS steps in and checks whether the expert agrees (the physician's response is indicated by **):

see the appropriate section of the handout

SUMMARY

TEIRESIAS and EXPERT SYSTEMS

TEIRESIAS aids a human expert in monitoring the performance of a knowledge-based system. When the human expert spots an error in the program's performance, in either the program's conclusions or its line of reasoning, TEIRESIAS assists in finding the source of the error in the database by explaining the program's conclusions - retracing the reasoning steps until the faulty (or missing) rule is identified. At this point, TEIRESIAS assists in knowledge acquisition, modifying faulty rules or adding new rules to the database. Meta-level knowledge about the kinds of rules and concepts in the database is applied to build expectations in TEIRESIAS's model-based understanding process. Meta-level knowledge is also used to encode problem-solving strategies, in particular, to order the invocation of rules so that those that are most likely to be useful (given the current knowledge of the program) are tried first.