A SHORT ACCOUNT OF KNOWLEDGE ENGINEERING

John Fox
Imperial Cancer Research Fund Laboratories
London

SUMMARY

Although expert systems and knowledge engineering are attracting great attention there is still confusion about what they are. Most expert systems deal with problems which are familiar to other disciplines and their distinctive character is not always recognised. This tutorial describes the origins of expert systems in Artificial Intelligence, and the technical concept of "knowledge". The central feature is the ideal of explicitly representing knowledge as facts, rules and other symbolic structures, rather than the traditional representation as abstract numbers or algorithms. These developments yield new solutions to old problems, and ways of solving problems whcih have previously been thought to be the preserve of the human intellect.

INTRODUCTION

Expert systems are fashionable and the subject is increasingly served by national and international conferences, commercial seminars and tutorials. However, few instructors have personal experience of building expert systems, and consequently there is still widespread misunderstanding of the subject. Furthermore designers take different views on what is important, often under the influence of practical compromises. Consequently software designers within industry and the professions, even researchers in fields touched by expert systems, sometimes find it hard to see the basic concepts or their distinct contribution. The tutorial provides a framework for understanding expert systems, some terminology and illustrates differences from traditional information processing. The framework is an idealised but reasonably complete perspective on the field.

Artificial Intelligence as Science and Engineering

Expert systems are derived from Artificial Intelligence (AI). AI appeared in its modern form about 1965; it was seen as a scientific subject not an engineering discipline, concerned with understanding intelligence in man and animals as much as in machines. AI studied:

> The principles of intelligence using information processing concepts as its theoretical framework and the computer as its principal tool.

The scope of AI was wide. It studied perception (understanding the mechanisms of vision and hearing), language (the spoken and written word), decision making, problem solving and learning. Progress was quite rapid though rather controversial given its subject matter. Few people expected the next major development however, the appearance of expert systems.

By 1975 a number of AI systems had been developed that appeared to solve difficult, practical problems in medicine, science and engineering. It was not that intelligence was now understood, but that symbolic information processing techniques had been developed which were able to solve specialist problems at a level which seemed comparable to human specialists. "Knowledge engineering" had appeared:

> The design of computer systems which can cope with tasks which would require intelligence if carried out by people, and which exploit, in part, human understanding of the tasks.

Knowledge Engineering and Its predecessors

AI has always been controversial. One of the reasons for this is that there are few subjects which AI has tackled which other disciplines have not previously attacked. Table 1 summarises these topics and their earlier counterparts.

Artificial Intelligence	'Conventional' information processing
problem-solving/inference	decision analysis
language understanding	machine translation
speech understanding	word recognition
speech production	word generation
visual image interpretation	pattern recognition
object manipulation	robotics
learning	adaptive control theory

Table 1

AI systems are often larger, more elaborate, and their authors more flamboyant, than their predecessors. It is not surprising that many people find the claims that AI is "new" hard to accept. However the key technical ideas of the AI approach are very different from the earlier, algorithmic, often highly mathematical approaches. The concept which cuts across much of AI, and which is central to expert systems, is the representation and use of knowledge. The next section explains what AI workers mean by knowledge.

What is new?

One important function of an expert system is that it can make decisions or help people to make decisions. However, expert systems were not the first computer systems for aiding decision making. For more than twenty years statisticians and others have developed mathematical techniques for medical diagnosis, educational assessment, personnel selection, resource planning etc. This was often done with great success. Figure 1 summarises the main components of a typical statistical system, taking medical diagnosis as a good example of the kind of decision the techniques have been used for.

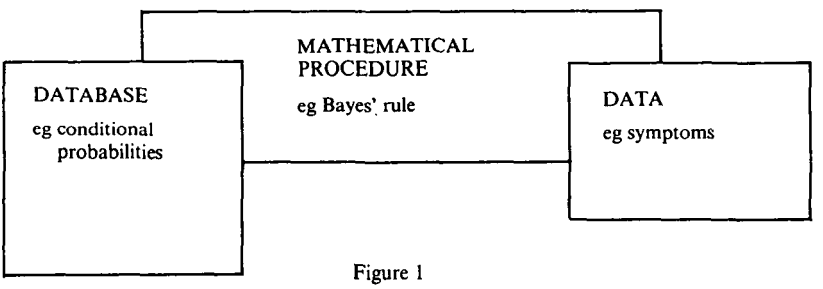


Figure 1

Suppose a patient presents with particular symptoms. The aim is to analyse these symptoms and decide upon a probable diagnosis. To do this we use a database containing a set of numerical parameters, typically conditional probabilities. they represent the likelihood of particular symptoms occurring with particular diseases. Finally some mathematical procedure, such as 'Bayes' rule' uses this database to calculate the probability of each possible diagnosis for the patient given his or her particular symptoms. Many mathematical methods have been developed for dealing with many kinds of decision, but they all fall within this general scheme.

Figure 2 is an analogous, schematic diagram of an expert system. Superficially the skeleton is similar, but this is deceptive. The first characteristic of an expert system is that the expert systems emphasise qualitative, logical reasoning not quantitative calculations. Logical inference uses logical data, consequently the database of an expert system does not contain numbers exclusively, and may not contain numbers at all. We call this material the knowledge base.

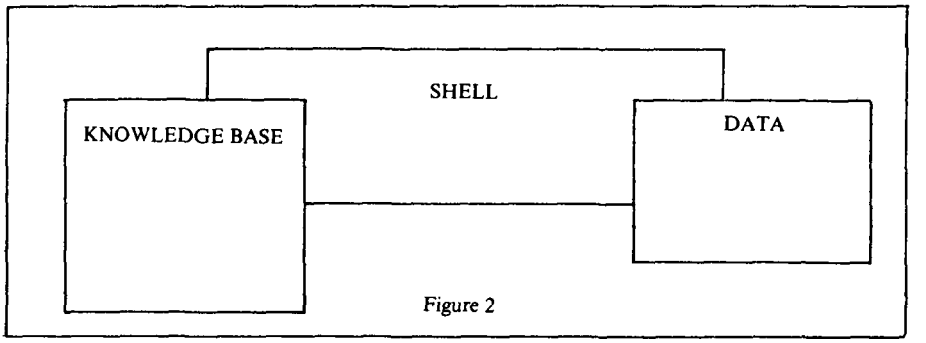


Figure 2

A key feature of a knowledge base is that its contents are not abstract symbols like conditional probabilities or other numbers. Probabilities record the magnitude of the relationship between a symptom and a disease, but not the meaning of that relationship. An important component of a knowledge base is often a body of facts which represent the way in which concepts are related to each other. Diseases cause symptoms; some symptoms are side-effects of treatments; some diseases are specific types of more general categories, and so on.

In expert systems the aim is to represent meaning explicitly by recording these relationships in a way that reflects people's concepualisation of the relationships, while in a form that the computer can exploit, rather than reducing them to abstract quantities.

FROM SYMBOLS TO SKILLS

Figure 3 is a schematic representation of many of the things to be found in the knowledge base of an expert system. The most primitive element, shared with all other computer programs, is the symbol. By progressively adding more elements the ability of these symbols to represent concepts and ideas becomes richer and more useful. At the top we have a simple kind of expert system or "skill system". Let me explain the steps involved.

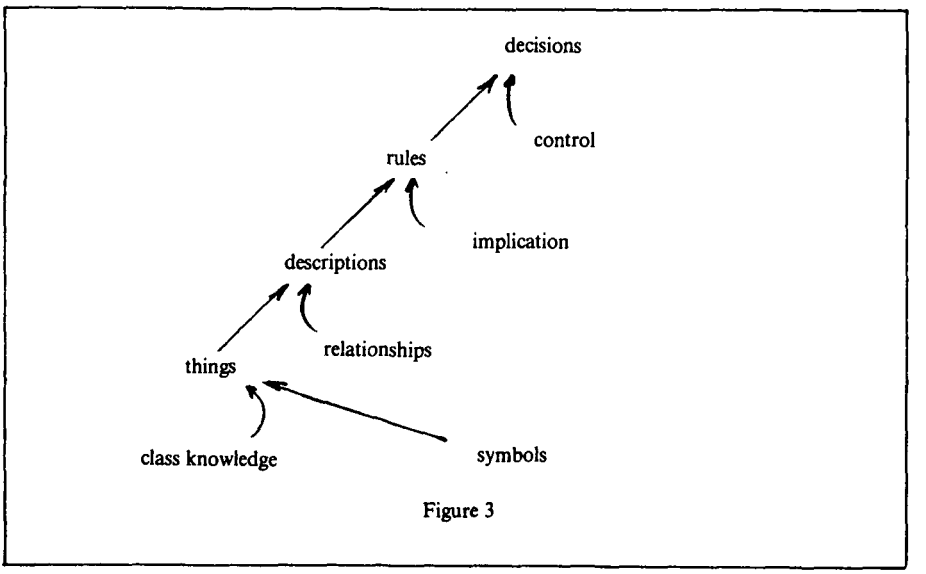


Figure 3

Computer programs manipulate symbols. Usually, however, these symbols are simply character strings like "Mr. Jones", "measles", "age" and "55" which have meaning for a human being looking at them, but no explicit meaning for the computer. The first step in attaching meaning to these symbols is to say what kind of things they are. Here are some examples. They represent some basic facts about leukaemia. (Most of my examples are taken from the field of leukaemia diagnosis and treatment). These facts can be stored in the knowledge base in exactly the form given here:

eg leukaemia is-a-kind-of cancer
cancer is-a-kind-of disease
chemotherapy is-a-kind-of treatment

The next step is to add more meaning by relating things to other things. A simple language of relationships lets us describe an infinite variety of facts and data:

'John Smith' suffers-from leukaemia
leukaemia is-an-abnormality-of blood
anaemia is-a-side-effect-of chemotherapy

Descriptions can be more complex than this but for clarity I shall limit my examples to relational triples.

Descriptions alone are not sufficient for reasoning, making decisions or solving problems. We also need to be able to infer new facts or new data from existing ones. This is often done by means of rules. The new ingredient that is incorporated in rules is if. . . then. . . implication:

if: spleen is enlarged, and
lymph-nodes are asymmetrically-enlarged
then: Hodgkin's-disease is possible

Given some data about a patient rules like this are capable of making inferences, like inferences about possible diagnoses or treatments. However we can't just put a large number of rules in a pot and stir. When we have many rules we often need to control the reasoning process. For example some rules should take precedence over others. Consider these two rules:

if: anaemia is present
then: iron-supplement is needed (1)

if: anaemia is present, and
spleen is enlarged
then: investigation is needed (2)

Rule 2 is more specialised than rule 1. Therefore if the available data satisfy rule 2, then rule 1 will also be satisfied. But rule 2 refers to a situation where a significant disease may be present; an iron supplement might be quite inappropriate. A skillful decision-maker does not apply rules blindly, but in a controlled way and an expert system must be similarly controlled. The inference engine of a knowledge based decision system usually provides this control.

There are two main ways in which inference engines control the use of knowledge in solving problems. In the "data driven" method it examines the available data and applies all the rules which are satisfied to derive new data. The conclusions which are made contribute to the fund of data and may in turn allow other rules to operate. The cycle may be repeated many times until eventually a definite diagnosis or a firm recommendation is reached.

The other method, the "goal driven" approach, starts at the opposite end of the logical process. It identifies the conclusions it needs to establish, its goals, and examines the rules to see which ones contribute to these goals. Then by looking at the "if. . . " part of the rules it locates any missing data which are required to make the conclusions. The missing data are then filled in by treating them as new "sub-goals", or by asking the user. As in the data driven method this cycle may be repeated many times. It may require many generations of sub-goals before the system obtains all the necessary data to achieve its main goals.

If the facts and rules are sufficiently comprehensive they contain all the information that is logically required to make any decision; all the inference engine does is make sure that relevant facts and rules are retrieved at the right time. In principle data driven or goal driven techniques are sufficient for all the logical steps. Although an expert system shell may provide tools to make the job easier, the power of the systems comes from the quality and comprehensiveness of the knowledge base not from sophisticated features of the shell.

OBJECTS AND CONCEPTS

It is popular to represent knowledge as rules and facts but knowledge can be expressed in other ways. For example an important area of development is the "object oriented" style of representing concepts as structured collections of interrelated facts and rules. Figure 4 illustrates the typical hierarchical organisation of an object oriented knowledge structure.

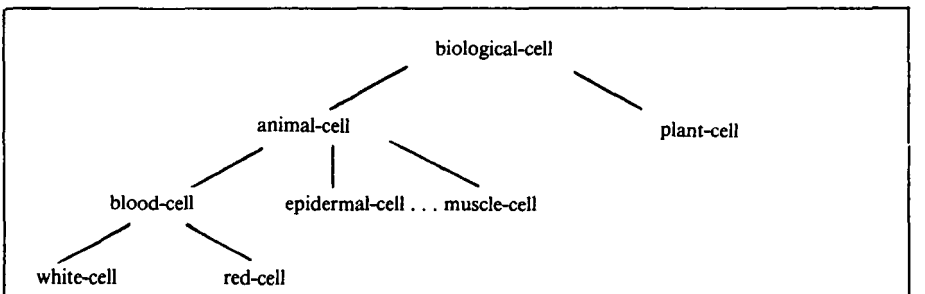


Figure 4

In an object-oriented representation objects are often organised into "inheritance hierarchies". Figure 4 shows an inheritance hierarchy for different types of biological cell. The highest object, or "frame", in the hierarchy contains the generic descriptions of biological cells:

biological-cell
- is-a-kind-of living-thing
- has-component cytoplasm
- has-component cell-membrane
- has-component nucleus
- has-size microscopic
- consumes energy . . . etc

The more specific types of cell in the hierarchy inherit these descriptions automatically. In addition they have their own characteristic properties which can be represented by relationships recorded in their own frames. Plant and animal cells are usually distinguished by the materials they store for their energy supplies, and more specific types inherit these properties.

animal-cell
- stores glycogen

plant-cell
- stores starch

Object oriented representations are proving to be convenient and powerful ways of organising knowledge. They are not limited to classification hierarchies but can represent other aspects of knowledge. 'Part-of' hierarchies can describe the objects which are components of other objects, and so on. Similiarly, as we shall see later, the objects may be more abstract, as when they represent laboratory techniques or treatment methods.

Broadly speaking, object oriented representations are most suitable for representing facts and data - declarative parts of knowledge. The procedural parts - the parts that control the processes of symbol manipulation - are usually dealt with separately. The most convenient way of representing procedural knowledge is, so far, the rule. Like descriptions rules can be stored within object frames. They are used when inferences about an object or set of objects are required, but they are not usually thought of as part of the inheritance structure.

THE CRUCIAL DIFFERENCE

The details of expert systems vary enormously from design to design, but the idea of knowledge is always central. As we rise through the hierarchy in figure 1, from elementary symbols to systems capable of making decisions, we are explicitly adding knowledge. This is the crucial difference between the traditional approach to software and the AI or expert system approach.

FUTURE DEVELOPMENTS

The development of a simple expert system for a new application can be routine for an experienced designer working with someone who is knowledgeable about the application. The necessary concepts and tools are sufficiently well understood that standard packages for developing them are available. But some kinds of problem-solving, such as planning, design and learning, do not fit easily into this framework, and more advanced techniques are being developed. An important element of these developments is the use of knowledge about how to use other knowledge, or meta-knowledge. Signal interpretation and the representation of uncertainty are other areas where progress is being made. The most remarkable area of long-term development may be in expert systems that can learn for themselves about how to solve problems. This is the least understood area of Artificial Intelligence, but it could lead to computer programs that are capable of discovering or inventing new ideas.

FROM SKILL TO EXPERTISE

Many people have sucessfully built simpler expert systems for many applications. In the United Kingdom, for example, these applications include:

medical diagnosis (of various kinds)
advice on social security benefits
assessment of entitlement to British Citizenship
corrosion analysis
packaging advice
fault diagnosis
computer performance prediction

Following d'Agapeyeff I have called these systems 'skill systems' because, although they do valuable work, they are inflexible compared with human experts. Human expertise is probably characterised by the ability to cope with tasks of many different kinds, it surely means the ability to be flexible and to cope with problems that have not been encountered before; the current generation of knowledge based systems is limited to relatively routine work.

Although the most famous expert systems are often much larger and more elaborate than these examples they are usually also simple in the sense that they do not match the flexibility of human experts. Research is now aimed at improving their flexibility.

Logic Programming

Logic based techniques have much more power than is exploited by the empirical rules of thumb in the examples. One important technique, which is being explored in logic programming allows the use of general purpose rules which can be used in new situations which have not been explicitly catered for by the designer. To illustrate this consider the "special case" rule:

radiotherapy is-a-treatment-for leukaemia, and
chronic-lymphocytic-leukaemia is-a-kind-of leukaemia implies that
radiotherapy is-a-treatment-for chronic-lymphocytic-leukaemia

The rule can be rewritten in a general form that can be applied automatically to more situations as facts are added to the knowledge base (capitalised words are generic concepts, represented by 'variables' in an artifical intelligence language), the other terms are relations, as before:

if: Treatment is-a-treatment-for Disease-Type, and
Particular-Disease is-a-kind-of Disease-Type
then: Treatment is-a-treatment-for Particular-Disease

We can even define rules which are almost completely abstract, and consequently become independent of particular applications in much the same way that mathematical methods are relatively independent of the details of the application. Rules are readable representations of logical theorems:

if: X Relation Y, and
Y' is-a-kind-of Y, and
Relation is-a-kind-of inheritable-relation
then: X Relation Y'

At the moment we have very little understanding of how to design such abstract rules, or how to control their use. However it is possible that we are on the threshold of a logical discipline of 'symbolics'to be compared with the disciplines of numerical mathematics.

Meta-knowledge

Perhaps the most significant development in AI which may substantially improve the flexibility of knowledge based systems is the introduction of the concept of meta-knowledge or "knowledge about knowledge". Systems with meta-knowledge can stand back, as it were, and critically consider the different techniques available to them in solving problems.

An early realisation within AI was that the traditional distinction between programs and data was restrictive and unnecessary. They are both simply symbolic structures and consequently programs can be data for other programs. Rules are programs and they can be used to apply, adapt or create other rules. Meta-rules can be analysed in much the same way that decision rules have been analysed.

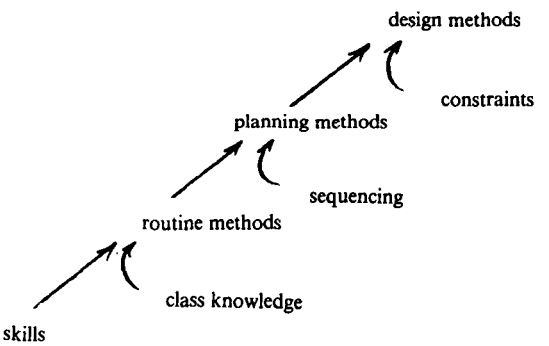


Figure 5

The first step is to regard all the concepts introduced so far, facts, rules, decisions, control mechanisms, as 'things' just like diseases, symptoms and so on. Consequently they can also appear in descriptions:

leukaemia-diagnosis is-a-kind-of diagnosis
diagnosis is-a-kind-of skill
data-driven is-a-kind-of control

and rules:

if: leukaemia-diagnosis is needed
then: clinical-diagnosis is needed, and
blood-analysis is needed

We can also have abstract methods for choosing specific methods, putting variables in the rules ('Method' is a variable, as before) and drawing upon descriptions in the knowledge base:

if: decision-method is needed, and
Method is-a-kind-of decision-method
then: Method is-a-candidate-for decision-method

We can also specify different kinds of method, such as methods for planning and designing, which incorporate routine elements. Plans are sequences of actions and routines which achieve some goal but in which the particular actions and routines are not specified - such as the planning of a drug treatment.

if: drug-treatment-plan is needed
then: drug-type is needed, and-then
method-of-administration is needed, and-then
supportive-care-plan is needed

This is a planning rule because it specifies a sequence of goals; the approprate method for administering a drug cannot be decided until the decision about which drug to use has been taken. Each component decision might be carried out by a skill system specialised in that particular decision. Planning rules may, however, also refer to lower level plans (like 'supportive-care-plan') which more detailed planning rules elaborate. Once again general rules that select among the alternative specific plans can also be designed.

Designs are like plans, but with the added requirement that acceptable solutions to the problem are constrained in some way. For example we may wish to design a drug treatment, while limiting the side-effects on the patient (or,a non-medical example, to design an electronic circuit while keeping production costs to a minimum).

if: drug-type is needed, and
age-group is elderly
then: low-potency is-a-constraint-for drug-type

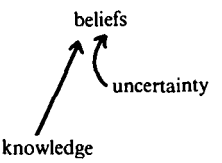
The examples of planning and design rules are special-case rules for the leukaemia domain, but they could be generalised to cover other domains.

if: G is needed, and
G is-a-kind-of G', and
Constraint is-a-constraint-for G'
then: Constraint is-a-constraint-for G

Progress has been made on general techniques for planning, but abstract techniques for design are not as advanced.

REASONING WITH UNCERTAINTY

Knowledge bases consist, broadly speaking,of factual knowledge, logical knowledge for drawing inferences, and strategic knowledge. In their simplest form all of these types of knowledge are deterministic. But the world is not deterministic; the same situation does not always produce the same result so deterministic reasoning is insufficient. Knowledge is not often certain:



A number of techniques have been used in expert systems for reasoning about beliefs. Curiously, as we shall see, this has taken a different approach to most of knowledge engineering - the emphasis has been on numbers, not knowledge, though as our understanding improves this exclusive approach will probably change.

We know of three places where uncertainty can manifest itself. 'Data' can be imprecise, 'facts' can be unreliable, and 'rules' can be rough and ready.

The earliest expert system which took a systematic approach to reasoning with uncertainty was Shortliffe's MYCIN. The essential idea was that rules should have parameters attached to them which represent the reliability, or certainty factor, associated with the rule. In this MYCIN rule which makes a conclusion about the nature of an infecting organism the conclusions 'facultative' and 'anaerobic' are qualified with numerical certainty factors:

RULE 027
If: 1) The site of the culture is blood, and
2) The organism was able to grow aerobically, and
3) The organism was able to grow anaerobically
Then: There is evidence that the aerobicity of the organism is facultative (.8) or anaerobic (.2)

All hypotheses have a certainty factor. As rules are applied they invoke a specially designed machinery to update the current degree of belief in the hypothesis using the certainty factor recorded with the rule. (It is

also possible for a user of the MYCIN system to record the reliabiliy of data items as these are entered, though in our experience this facility is rarely used.)

The CASNET system, which modelled casual processes in the eye disease glaucoma, dealt with uncertainty in a different way by associating numerical probabilities with the facts in its knowledge base. For example:

elevated-intra-ocular-pressure causes retinal-damage (.7)

CASNET also incorportated a technique for revising its belief in hypotheses (eg the hypothesis 'retinal-damage') as data accumulate.

The Prospector system combined rule based techniques and fact based techniques. In this system the premises of each rule refer to facts about geology, and data about the specific problems. When a rule has a number of uncertain premsies then some overall measure of the reliability of the pattern is needed. The designers chose a fuzzy logic technique. As with MYCIN, Prospector also treats rules as unreliable; it has a mechanism for updating the strength of a conclusion based on the reliabilty of the rules. This is separate from the fuzzy mechanism for assessing the overall reliability of the facts.

These expert systems all used variants of probability theory to represent uncertainty. They attach numerical 'coefficients of belief' to the knowledge elements. These coefficients are a simple, but abstract, form of meta-knowledge. We would like to make the knowledge behind our uncertainties explict - so we can reason about it - and there has been some progress here. Cohen and Grinberg suggest a "theory of endorsements (which) provides a richer representation of the factors that affect uncertainty and supports multiple strategies for dealing with (it)". Elsewhere I have proposed a 'linguistic' approach to reasoning about uncertainty, which would allow the expert system to reason about many of the kinds of uncertainty that we distinguish in our everyday language.

FROM SIGNALS TO SYMBOLS

Expert systems normally accept data in a discrete form and infrequently (eg a patient's symptoms typed at a terminal). However, an important class of data for many applications is signal data where the primary source of the data is some sort of analogue device and the sampling rate is high (eg greater than 1 kHz). The non-numerical bias of expert systems, and the relative slowness of symbolic reasoning by comparison with conventional algorithms, currently makes them unsuitable for real-time data processing applications. Medical instrumentation and process control are areas where the data-rates are currently far too high for knowledge based processing of the signals.

However knowledge based techniques can be used for interpreting the data once it has been encoded in an appropriate form. Although conventional signal processing techniques can contribute a lot to analysing signals (eg by increasing signal to noise ratios with filtering techniques) they cannot bring knowledge to bear on the interpretation of those signals.

We can begin to exploit knowledge when the outside world has been represented symbolically. How should we represent signals symbolically? Figure 6 shows the process in an idealised way.

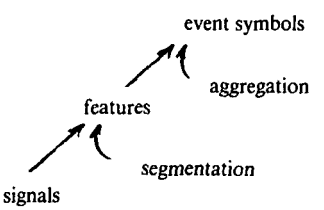


Figure 6

Consider the case of a simple signal, from a voltmeter say, which has been passed to the computer via an analogue-to-digital converter.

The signal is represented in the computer's memory as a one-dimensional vector of numbers.

...123 123 123 125 144 126 121 123 125 127 129 131 ...

This vector can be turned into a symbolic representation in two stages, segmentation and then aggregation. Segmentation exploits discontinuities in the signal (eg abrupt discontinuities like pulses) to parcel up the signal into segments. Aggregation techniques can then be applied to link together the segments which are in some way similar (eg a periodic series of pulses). The signal is now represented as a train of elementary events, which can be encoded symbolically and entered into descriptions like other symbols:

segment-44 is-a pulse
segment-44 is-a-part-of segment-60
segment-60 is-a regular-beat

The same principles can be applied with more complex signals, eg images, which have at least two dimensions, to describe shapes, textures and the surfaces of objects.

Some military work has shown that this approach may become practical. In sonar interpretation knowledge of classes of ship, their believed speeds, last known position, heading and so on, can be used to disambiguate noisy or intermittent data.

Much of the work has been a compromise, integrating conventional signal processing equipment with an expert system "back end". This sort of coupling severely limits the possible performance of the whole system because the output of the conventional software is designed to be passed to human beings via displays, not expert systems. Furthermore opportunities for feedback from the interpretation level to alter the settings of low level data capture processes are very limited. However, as more computing power and better methods of integrating different processing techniques become available it is likely that signal interpretaion will enter the standard repertoire of expert systems.

FROM EXPERT SYSTEMS TO INNOVATION SYSTEMS

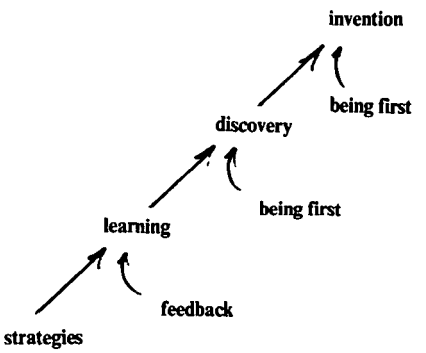


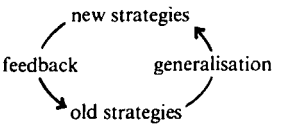
Figure 7

Learning is one of the hardest problems that has been addressed by AI. It is hard enough to develop systems with great skill or expertise, but to learn these capabilities as well is far more demanding. One way of viewing learning is to treat it as another area where meta-knowledge can be used.

Suppose we have general purpose techniques for making decisions, planning and designing. We can apply the now familiar trick of representing them as 'things' by saying what kind of thing they are:

diagnosis is-a-kind-of strategy
hierarchical-planning is-a-kind-of strategy
failsafe-design is-a-kind-of strategy

Theoretically we can now apply rules which generalise or adapt these techniques to make them applicable to new situations. If the results of these changes are monitored then the success or failure of the new versions of the techniques can be assessed (by the computer system). Unsuccessful generalisations can be fine-tuned, or abandoned. Successful versions can be stored for future use. Learning has been achieved.



If computer syustems can learn new knowledge then they may be capable of solving problems in new ways; conceivably. They could even be 'creative'. Traditionally we have regarded creativity as a mysterious characteristic of human beings which is beyond the capabilities of machines. However some AI researchers have suggested that, while creativity is difficult to understand, it may be the product of knowledge based processes like the others we have discussed.

> "Perhaps the new thoughts originating in creative thinking are not wholly novel in that they have their seeds in representations already in the mind? Perhaps they are not totally inexplicable in that we can manipulate familiar representations so as to generate others which are somehow fresh or original?"
>
> M Boden

Doug Lenat of Stanford University has taken this possibility particularly seriously. He has developed a number of AI systems to investigate the posibility that programs can originate new knowledge. His approach develops the learning techniques outlined above, but his programs learn in new areas, which human beings have not fully explored. He has applied these ideas to discovery in mathematics (AM - the Automated Mathematican) and invention of solid state devices (Eurisko) and elsewhere. Lenat claims that AM has proved theorems in number theory that were previously unknown, and that Eurisko invented a new solid-state device which has since been patented.

There is a strong theory of innovation inherent in Lenat's work. It says that discovery and invention result from controlled generalisation of strategies. This produces new ways to make decisions, plans and designs. When the generalised strategies are applied to new situations they yield new results. Taking Lenat's results at face value this kind of 'routine innovation' is clearly of huge significance for science, technology and industry. The crucial thing in discovery and innovation is 'being first', not necessarily being creative in some obscure way. Whether AM and Eurisko are creative in any more interesting sense is a subject of fierce debate.

CONCLUSIONS

The current generation of skill systems is useful but limited; the next generation is likely to be much more competent and flexible in decision making, planning and design tasks, and will include expert systems for interpreting signals and other kinds of analogue data. The key concept which distinguishes these systems from their mathematical and algorithmic predecessors is the explicit use of knowledge. Knowledge based techniques are also beginning to yield systems with far greater abilities than traditional computer programs. Just how far they can go remains to be seen.