THE COMMERCIAL APPLICATION OF EXPERT SYSTEMS TECHNOLOGY

Tim Johnson
Ovum Limited
14 Penn Road, London N7
This is the executive summary from a detailed report of the same name, published by Ovum.
©Ovum Ltd

A NEW SOFTWARE TECHNOLOGY

The relentless growth of the computer industry over more than 30 years has been driven on by a series of major innovations. High-level languages; solid-state logic; compatible machine ranges; disk storage; time-sharing; data communications; virtual machine architectures; the use of LSI and solid-state memory; text processing; personal computers; sucessful packaged software; each advance in its turn has opened up new markets and set off a new spurt of expansion. A vital conclusion of this study is that expert systems promise to be another such major innovation.

Expert systems have emerged in the last few years as the leading practical application of the techniques developed in artificial intelligence (AI) research. They have been much spoken of in the press and on television, usually with references to a few spectacular pioneering systems, especially those developed by Ed Feigenbaum's group, and others linked with Stanford University. Prospector, which discovered a $100m molybdenum deposit overlooked by human geologists; Dendral, the parent of them all, which is used by hundreds of chemists to determine molecular structures; and Mycin, which can assist physicians with the diagnosis of infections, are among the most frequently quoted.

However, this report is more concerned with a new generation of expert systems, now being put to use on day-to-day problems because they provide the most cost-effective means of doing the job. These systems are less spoken about in the media because they are in less glamorous applications than medical diagnosis or mineral prospecting. They are carrying out such tasks as configuring computer systems, planning the repair of telephone cables, and helping people to use complex software. But it is in this more technical area that the best immediate prospects for expert systems lie.

Another significant difference in the approach adopted here is in the definition of an expert system. Many academics working in the area emphasise the objective of emulating the performance of a human expert at a high level. On this basis, building an expert system must by definition be a difficult and expensive project.

In this report an alternative, technology-based definition is preferred; an expert system is a set of computer programs which emulates human expertise by applying the techniques of logical inference to a knowledge base. A knowledge base is itself an organised body of information in computer memory. It is distinguished from a database or a computer file because it holds information about how to carry out a task,such as the rules which a human expert learns by experience, rather than just statements or facts. The importance of the knowledge base idea in the practical application of the results of artificial intelligence research is recognised in the name "Intelligent Knowledge-Based Systems" (IKBS) which is the official designation in Britain of the area of research which includes expert systems.

The practical implication of this definition is that an expert system can be of almost any size or difficulty, as long as it makes us of inference from a knowledge base. "Small" or "modest" expert systems are as valid a way of applying the new technology as the major systems which receive most of the publicity, and they are likely to produce quicker real benefits.

The reasons why expert systems have so much to offer as an alternative to conventional software architectures are set out in detail in this report, but they can be summed up under four headings.

1. Expert systems allow the computerisation of tasks which were previously unprogrammable. A leading example is the system used by Digital Equipment Corporation (DEC) to configure their VAX minicomputer installations, originally known as R1 and now usually called XCON. attempts by DEC to write programs for the configuration task failed and the problem was only solved by the development of XCON, beginning in 1979. Now the system runs every day in both the USA and Europe and provides configurations for over 97% of VAX sales.

2. Expert systems are easier for users who are not programmers to understand. Because a knowledge base is a fairly direct representation of human knowledge, non-specialists can understand what it contains, and monitor its correctness and progress. At the same time, the inference-engine/knowledge-base architecture allows the system to run even before the knowledge base is complete, and to provide some explanation of the reasoning behind a given conclusion. These features can be invaluable in the

development phase of a system.

3. Expert systems can allow a spectacular increase in programming productivity. This is as much a function of the very powerful programming environments that have been developed for AI work in general, as it is of specifically using expert systems architectures. The productivity gains come from a variety of advantages, mostly connected with the flexibility with which partially-completed programs can be run, and decisions left open instead of having to be fully specified in advance. Productivity claims approach the incredible, ranging from four to 10 or even 50 times that achieved by conventional methods; this is one area where the advantages of expert systems technology need to be proved in routine use.

4. Expert systems can provide a genuine extension of human capabilities. Even though this report maintains that the most valuable early applications will be for tasks which are mostly fairly easy and routine in human terms, expert systems have already shown the capability of exceeding human performance in certain circumstances. A system developed at the University of Illinois for diagnosing disease in soya-bean plants can now produce more reliable diagnoses than the leading expert who set it up. More dramatically, in the foreseeable future expert systems will be able to take decisions in fast-changing environments, ranging from battles to foreign-exchange trading, more effectively than humans ever could. Where such systems will eventually lead is impossible to predict; their capabilities could be virtually boundless.

Certainly, the implementation of expert systems will have its difficulties and drawbacks. The number of major systems completed, let alone those which are well-documented successes, is still too few to demonstrate their benefits conclusively. The process of knowledge engineering, discovering the information which is relevant to a task and devising the means of structuring and representing it in the computer, is labour intensive and relies on skills and experience which are likely to remain in short supply for years to come. The systems themselves ideally require expensive, high-performance hardware for their development and running. In many potential applications systems face likely resistance from established human experts whose interests are threatened.

Taken together, these factors will ensure that the progress of expert systems is less rapid than some of the wilder forecasts have suggested. But the advantages of the new technology will be too great to be ignored, and the report identifies many areas where the expert systems model will be applied in the next few years, both to computerise new applications and improve the performance of older ones.

The expert system architecture

The concept of expert systems is closely bound up with the work now being done to develop a new generation of computer technology. Following the announcement of the Japanese "Fifth-generation" research programme in October 1981, with the mission of developing computer systems to meet the needs of the 1990s, almost all the leading industrial countries have responded by setting up comparable programmes. In the USA there is the DARPA (Defence Advanced Research Projects Agency) Strategic Computing project; the UK has the Alvey Programme; France and Germany also have national programmes, while Esprit covers all the countries of the Common Market. Even the USSR and its satellites have a joint computing five-year plan focussing on similar topics.

The aims of all these programmes are quite broad. They are concerned with the development of a range of "enabling technologies," such as software engineering or VLSI (Very Large-Scale Integration), as well as with the applications of artificial intelligence and the other requirements of advanced data-processing systems. But expert systems developments, both of applications and the tools needed to implement them, are a significant part of each programme, and their relative importance is tending to grow as the most promising directions of research emerge more clearly.

The technologies needed to implement an expert system are well established, and their main features - with some of the controversies that surround them - are outlined in this section. There are two main components -the software comprising the expert system itself, and the environment in which it is developed. The application software must include:

- a knowledge base
- an inference engine
- a user interface.

Interfaces to other data sources, and independent programs, may also be required.

The environment in which the system is developed will generally consist of;

- an expert system shell and development tools
- implemented in one of the favoured artificial intelligence languages
- running on a special-purpose workstation or other dedicated machine.

Once developed, the system may run on a different machine, more appropriate to routine use - the delivery vehicle.

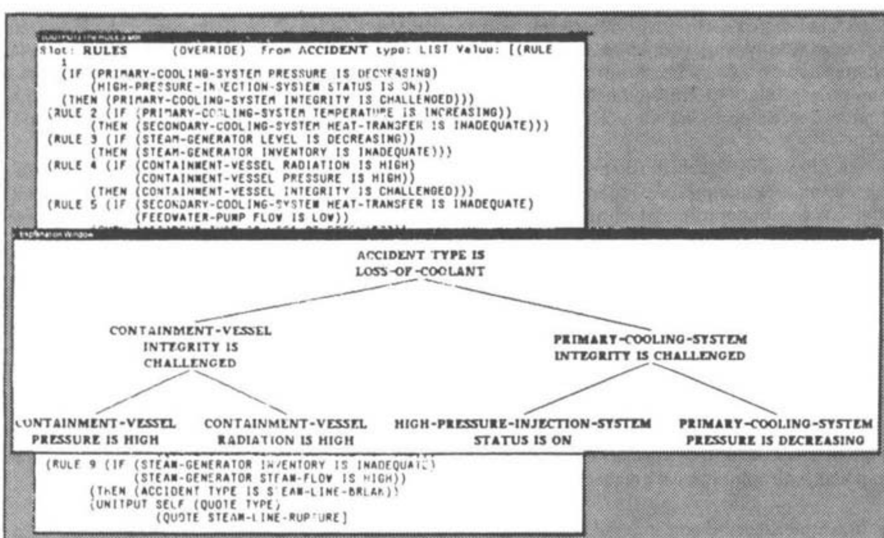
The design and content of the knowledge base is crucial to the performance of the whole expert system. The primary tasks of the knowledge engineer are to elicit the "knowledge" - meaning the empirical mixture of heuristic rules, facts and models of the real world - which an expert uses to solve problems in his or her profession, and design the means of structuring and representing it in the knowledge base.

In particular the knowledge engineer will choose a language - probably a special-purpose one - in which to represent the rules the expert uses to make decisions. To take an example from "Reactors" a small system developed by IntelliCorp to demonstrate its Kee expert-system shell, the statement:

> "If (1) the pressure in the primary coolant system is decreasing, and (2) the high-pressure injection system is on, then the integrity of the primary coolant system is challenged."
>
> is represented in the Kee language as:
> (If: (Primary-coolant-system pressure decreasing)
> (High-pressure-injection-system on))
> (Then: (Primary-coolant-system integrity challenged))

A snapshot from this knowledge base is shown in Fig 1.

Fig 1 Display from a demonstration expert system



Source: IntelliCorp.

Depending on the facilities offered by the expert system shell, the knowledge base may include algorithms, small programs or subroutines, which would be called in appropriate circumstances, and it will also include straighforward facts or statements. Thus a knowledge base is essentially different from a conventional database. A familiar tool which can be seen as a form of knowledge base is a computer spreadsheet, where each cell may hold a fact, a procedure or, in the more sophisticated versions, a rule.

An important feature of expert systems technology is the separation between the knowledge base, which contains all the information and procedures which are specific to the application, and the inference engine which, at least ideally, is a general-purpose program, free of application-specific features. While knowledge bases are virtually unlimited in their size and complexity, the heart of a simple inference engine may be no more than a few lines of code.

In the simple case where the knowledge base is built up only from IF-THEN rules (known as "productions"), the task of the inference engine is to search for rules where the "IF" side is satisfied, and then to "fire" the rule so that the statement on the "THEN" side is used to augment the data available about the situation. In this way the system builds up towards a result, perhaps from deciding that the primary coolant system is threatened to concluding that the power station should be shut down.

In real systems the situation is usually much more complicated. The search through the knowledge base has to be guided and controlled, otherwise it might never reach a conclusion. The knowledge base has to be structured to provide control and the inference engine must be tailored to it. Two main alternative search strategies are used - backward chaining and forward chaining - and each is appropriate to different problems. In practice, most real-world applications need a combination of both.

Backward chaining involves working back from a conclusion or goal to see if the conditions which would make it true are satisfied. It is appropriate where the possible conclusions can be specified in advance - for example in medical diagnosis or diagnosing faults in equipment.

Forward chaining builds up from the available facts about a situation to deduce conclusions. It is appropriate where the possible conclusions cannot be pre-specified, for example in designing a computer configuration, where an endless variety of end results are feasible.

Builders of expert systems no longer have to program their own knowledge representation structures, search strategies and inference engines from scratch. To a large extent these are available ready-made in proprietary expert system shells. Already these offer a wide range of alternatives, from the most specific shells where all the structures and other facilities are pre-defined, to kits of parts and tools which the system designer uses as the main building blocks of a new system.

The first generation of shells were mostly created as a by-product of pioneering expert systems applications, and tend to be more specific, with a single inference mechanism. This is backward-chaining in shells derived from diagnostic systems, such as Stanford University's Emycin, or SPL International's Sage, and forward-chaining in a few cases such as Rutgers University's Expert. But the new generation, only now appearing as commercial products - Kee from IntelliGenetics, Loops from Xerox, Art from Inference Corporation - offer a choice of mechanisms and an increasing range of knowledge representation facilities.

The new generation of shells aim to provide a powerful development environment for expert systems, as well as a ready made framework. An important aspect of this is the quality of the user interface provided. Instead of the screen menus or forms which are now familiar in thousands of personal computer packages, the new shells offer "wide bandwidth" interfaces, where the user is presented with a high-resolution bit-mapped screen with "windows" which allow the simultaneous display of, and interaction with, different parts of the software. This is combined with a "mouse" control which can be used to identify and manipulate different parts of the display. The user has a two-dimensional plan view of the system instead of the essentially one-dimensional linear view presented by a scrolling screen interface. The result should be to multiply the knowledge engineer's productivity in building a system by a significant factor.

The wide-bandwidth interface demands a special display and a powerful processor to drive it, but the demands of artificial intelligence programming have created a need for special-purpose workstations for other reasons as well. In particular, the languages favoured for AI can be grossly inefficient when run on computers with conventional architecture.

The best established AI language is Lisp, invented at the Massachusetts Institute of Technology (MIT) by John McCarthy in the 1950s. Lisp is more convenient for AI work than conventional data-oriented languages; one reason is that it allows the direct representation of symbolic concepts and the relationships between them in the form of data structures called lists - in fact lists are the only data structures in Lisp.

Another convenience of Lisp is that it does not require the data types of each variable, and the allocation of memory to each data type, to be specified at the beginning of a program; instead data types are determined at run time, and memory is allocated flexibly according to requirements.

The penalty is that a heavy overhead of extra instructions must be carried to deal with data typing at run time, and the demands on main memory are very large. The response has been to develop a line of workstations which are microcoded to handle data typing in hardware rather than by software, and which have the ample memory needed to run Lisp efficiently. In the USA, two manufacturers - Symbolics and Lisp Machine Inc - are now marketing descendants of MIT's original Lisp Machine design, and Xerox has developed an independent line of equipment based on the work of its Palo Alto Research Centre. In Europe, a joint company formed by Racal and Norsk Data will begin deliveries of a machine in late 1984. These workstations are being sold for AI and symbolic computing applications of all kinds, but expert systems developments promise to be an increasingly important component of their markets.

The line of thinking which has led to these powerful, expensive, sophisticated special-purpose Lisp workstations is only a part of the story. Most significant expert systems running today were actually developed on "stock" hardware - conventional computers designed primarily for conventional applications. By far the most widely used in the AI and expert systems community have been the DEC system 10s and 20s, and increasingly the VAX range. But a wide variety of other machines have also been used on occasion, from Apples to IBM mainframes. And in those cases - relatively few to date - where a system has been developed on one machine for eventual operation on another, the chosen delivery vehicle has virtually always been stock hardware.

Many stock hardware developments have also used stock languages - Pascal, C, Fortran, even Basic in some cases. Since any valid computer language can be used, in the ultimate, to program any computable problem this has been largely a matter of convenience and availability for the users concerned. In some cases the need to use a stock language has led to the development of specialised sub-routines to emulate the facilities available in Lisp, so that users of these systems may have no compelling reason to change.

The other alternative to Lisp is to use another of the languages specially suited to AI and symbolic computing, of which Prolog - developed mainly in France and the UK during the 1970s - is the most prominent. Prolog originated as an attempt to design a language which would allow the programmer to specify the objectives of a task in terms of symbolic logic; a Prolog program is predominantly "declarative," in that it is concerned with stating what has to be done, in the form of rules and facts, while a conventional program is more "procedural", concerned with how the task should be done. A major advantage of Prolog is that the expert systems concept of an inference engine working against a knowledge base, and seeking to satisfy assigned goals by firing rules, maps very directly onto the language; in a sense any Prolog program can be seen as a sort of expert system.

Prolog got a major boost when the Japanese announced that they intended to use it as the core language for their Fifth-generation project. But it has suffered from being relatively new, and lacking in implementations which can compare in efficiency and facilities with the Lisp environments on special-purpose workstations. The Fifth-generation project's "PSI" (personal sequential inference) Prolog workstation may provide a solution; more immediately implementations which combine Prolog and Lisp facilities may make the controversy between Prolog and Lisp redundant. Routes to this solution include the Poplog environment now getting increasing attention in the UK; LMI's new workstation which can be microcoded for Prolog as well as Lisp; and advanced expert systems environments such as Loops.

The fervent controversy between the advocates of Lisp and Prolog is only one of many over the most appropriate technology for expert systems and AI work in general. Perhaps the most fundamental division is that between those who see expert systems as essentially big projects, involving many man-years of work by highly skilled specialists with the most sophisticated equipment, and the advocates of small systems, who point to the benefits that can be achieved in a few months by self-taught knowledge engineers working with stock hardware. Both sides are right to say that their approach can produce valuable results. The danger is that the heat of controversy will blind participants to the value of their opponents's work.

HOW WILL EXPERT SYSTEMS BE USED?

Systems today

Reviewing the current status of expert systems, three main features become apparent.

1. There have been a large number of experimental projects and demonstrations. Over 180 from the USA and the UK alone are listed in the main report, developed by 95 different organisations.

2. Far fewer projects have achieved routine real application - perhaps no more than 20.

3. Almost all the projects which are in regular use are "small" expert systems, generally taking six man-months or less to develop.

Medical diagnosis is a major theme among the projects listed. Mycin, for the diagnosis of blood infections, has already been mentioned. Other famous diagnostic systems include Internist/Caduceus, for internal medicine, Casnet for eye disorders, Puff for lung disorders, and Teiresias for acquiring medical knowledge. But despite the amount of work done on medical systems, few of them are in routine operational use. Reasons for the emphasis on medical applications include the suitability of the knowledge for representation in expert systems, and the ready availability of research funds in the area, while publicity for the results is not inhibited by the requirements of commercial and military security.

Further much-publicised applications are in oil exploration, such as Schlumberger's Dipmeter Advisor - assisting in the analysis of oil-well logs - and Teknowledge's Drilling Advisor, intended to help drilling teams diagnose and clear problems more quickly, and developed for the French oil company, Elf Acquitaine. But typically these systems have been large and ambitious, and have proved more difficult to put into operation in the field than their sponsors originally hoped.

So most of the expert systems which have shown the first real commercial return are in less glamorous areas. They could be called "technician" instead of "expert"systems because they supplement or substitute for the brainpower and experience of skilled technical staff rather than highly trained professionals. Many of them are concerned with complex modern equipment; designing its configuration, diagnosing its faults, assisting in its use. The biggest single user of this kind of system is the computer industry itself.

Without doubt, the leading operational expert system is XCON, which as mentioned above is responsible for configuring almost all DEC's orders for VAX computers. XCON is by far the biggest expert system known to be in regular operation, with over 2000 rules by early 1984. It has achieved clear and specific benefits for its users, allowing the same team of engineers to handle a greatly increased flow of orders, and reducing the proportion of configuration errors in delivered systems from 35% to 2%. And it has provided the basis of DEC's plans to extend the use of expert systems to many aspects of its business.

XCON has succeeded because of its suitability for expert systems technology, and because of a favourable environment. The knowledge domain with which it is concerned, the configuration of specified minicomputers, is complex but clearly defined. John McDermott, of Carnegie Mellon University, who was responsible for the original idea and its implementation, was able to structure the task to allow forward-chaining within sub-tasks processed in pre-defined order. It has taken four years for the system to gain experience and achieve its current (1984) high level of performance, but it was able to provide some useful support to the configuration team from early in its development.

The environment has been favourable because the technicians responsible for VAX configurations were under pressure to keep up with sales as orders grew rapidly. There has been no question of reducing their numbers; instead they find themselves in an area - expert systems applications - of increasing attention and activity in DEC. And the company as a whole is better placed than most to provide the financial and technical resources needed to carry the system through to sucess.

DEC is demonstrating its belief in the expert systems approach in the best way possible, by investing heavily in a series of further systems which should eventually provide "intelligent assistants" for every stage in the process of selling, building and supplying computers. Some of these systems, such as XSEL, which gives DEC's sales people interactive assistance in proposing suitable configurations, site plans and delivery dates, are already in the field test stage. But many of these systems will be working in more complex environments than XCON, and raising more difficult issues, so it will not be surprising if they prove more difficult to implement successfully.

Compared with XCON, all the other identified "technician systems" in actual operation are relatively small. A variety of companies have found that fault diagnosis systems can be implemented with modest equipment in months rather than years. US General Electric, British Telecom, Ferranti and SPL International have all built systems to run on micro or minicomputers, for the diagnosis of diesel locomotive repair needs, faults in PABX power supplies, minicomputer crash dumps, and electronic equipment failures respectively.

Another area where technician systems are being implemented can be broadly described as the front-ending of complex equipment. These systems assist a user to set up or operate the equipment, or to analyse the results it generates. The main classes of equipment involved are computer software and scientific instruments. Cognitive Systems and Rutgers University have developed systems which interface to complex software for analysing geological data. Unilever and IntelliCorp have developed systems which assist in setting up and operating scientific instruments. The use of expert systems to assist in analysing instrument data goes right back to their beginnings with Dendral.

Again, most of the working front-end systems, like the fault diagnosis ones, are relatively small. And similar small systems can be found in operation in a number of other areas. But major expert systems which have got beyond the field test stage are few and far between as of mid-1984. XCON may be the only firmly established example of the new generation of systems. Dendral, Macsyma (a set of programs used for solving problems in symbolic algebra), and a number of other programs for scientific research are widely used. But they belong to the earlier generation in which the expert systems architecture was not fully developed.

The lack of major systems which have achieved operational status could be used to dismiss the claims of expert systems to represent a major innovation in computer technology. But in fact it can be ascribed to the elapsed time and man years needed to bring a major system to fruition, combined with the shortatge of people who have the appropriate skills and experience.

Estimates suggest that the number of major commercial expert systems completed and put to work in the USA in 1982 was only two, and in 1983 three, while in 1984 it will be only five. These figures are based on the number of knowledge engineers available in previous years, the time - span needed to complete a major expert system (typically three years), and allowances for the proportion which fail, and for competition for skilled people for research, the development of software tools and work on small projects.

Taking these factors into account, it ceases to be surprising that the number of known operational major systems remains small even in the USA, while they have yet to appear in the UK. With the rapid build-up in the numbers of knowledge engineers, from an estimated 200 in the USA at the beginning of 1982, to 400 in 1983 and 700 in 1984 the number of major projects reaching fruition should increase sharply. The more successful major projects which have been initiated in the past few years, particularly for the oil industry and financial services, and probably for defence, should begin to deliver results during 1984-86. Meanwhile the computer industry should be increasing its dependence on expert systems. If these developments do not take place, then it will be time to say that the hope placed in expert systems is proving unrealistic.

Future applications

Besides continuing development in computing, electronics and the information technology industries in general, some of the richest future application areas for expert systems will be in financial services, oil exploration and the military. These are the industries where the conditions for applications are most favourable, and also where a large proportion of development efforts are now focussed. But there will be successful applications in other industries as well, and some of these could prove to be of major importance in practice. Readers of this report may be able to identify form their own knowledge other applications with the characteristics which suggest they will be favourable for expert systems.

The first test of an expert systems application must be whether it is technically feasible. Given this basic requirement, then the characteristics which can point to sucess are:

1. A favourable environment
2. A big potential benefit
3. Relatively low technical difficulty.

The factors which define each of these characteristics are discussed in the full report. The environment is identified as being the most important, because it can be seen as a major determinant of whether systems once developed are actually used. Favourable environments are provided by applications in wealthy industries with the experience and resources to adopt advanced new computer technology. Applications have a better chance of sucess if they do not change the work of important established interest groups or have wide organisational implications.

The benefit which can be foreseen from using a major expert system must be large, or it will not be worth developing in the current state of the technology. Given the uncertainty of the development costs, and the risks of total failure, a project which has to be justified by detailed cost analysis is probably not worthwhile. In the words once used to describe the cost of yachting, "if you have to ask the price, you can't afford it." Again, this favours applications in large and wealthy industries, where there are opportunities for better-informed decisions to save or earn large sums of money. It also favours the development of small systems, which can be justified as an experiment or a training exercise and dropped without embarrassment if the results have insufficient practical value.

The technical difficulty of the task is placed last in the list of characteristics because the capabilities of expert systems technology have already outrun the requirements of many applications. Factors which determine the technical difficulty include:

- the nature of the task. Is it, for example, a matter of simple classification, which can rely on a low-cost backward-chaining inference mechanism, or a creative process which will require a variety of knowledge representations and inference mechanisms, and a carefully designed control structure?

- the nature of the knowledge domain. Is it clearly-bounded and uniform or wide-ranging and heterogeneous? Is it already codified or must it be laboriously assembled by talking to the experts?

- the mode of operation. Batch, interactive, real-time? Real time expert systems offer major benefits, in financial trading or process control, for example, but present day hardware and operating systems are not up to the task of processing thousands of logical inferences per second, which will be required in many applications.

Taken together these characteristics point to the industries already mentioned, with the possible addition of other highly concentrated sectors such as automobiles, chemicals, mass-market consumer goods and the major retailers. In these industries, the major systems applications will eventually be at the points where the flow of money and resources, and the corresponding opportunities for increased efficiency, are greatest. They will be involved with the management and scheduling of complex production processes, asset management for a bank, or the design of new products.

But because these major systems will be concerned with managing the most vital functions of an organisation, they will be correspondingly difficult and slow to implement. There will be an easier route to applicaton in bypassing expertise, replacing the expert who is too scarce or too expensive, or simply not suffficiently reliable in performing a largely standardised task. Systems for fault diagnosis or front-ends to complex machinery will grow in number and greatly extend their range. Typically they will be small systems and relatively easy to implement. Other small systems, packageable in a micro for interactive use, will join them for such functions as configuring personal computers or advising on personal investment portfolios.

The small systems may well have more practical impact and market significance than the big ones, at least in the short run. Estimates suggest that the resources available will allow the implementaion of more than 1000 small commercial systems in the USA in the years up to 1988, with barely 200 major ones. If some of the small systems are well-chosen and packaged for mass markets they could become widely used while the promoters of major systems are still struggling to prove their value and get them accepted.

Increasing programming productivity

Projections of the ever-rising share of software in total costs are a regular feature of computer industry prognostications. In practice the pressures of economics, and the ingenuity of the manufacturers, have combined to ensure that software and hardware costs have remained roughly in balance for some time -although the channels through which software is provided have changed radically. The technologies which have been developed in the AI communtiy, and which are being applied to implement expert systems, could provide a further opportunity for the manufacturers to keep their share of the customer's dollar by providing expensive but highly productive equipment.

As mentioned above, programs in Lisp, Prolog or other AI languages require generous allocations of main memory and at least an order of magnitude increase in the number of instructions processed. Hardware to develop and run them is correspondingly expensive, but if the claims made for the productivity of the specialist programming environments are justified, then the cost would be recovered many times over.

At one extreme, a figure of 10 program instructions per day designed, wirtten, debugged and documented, is still often quoted as a norm for software development - corresponding to little more than 2000 a year. At the other, expert Lisp programmers are claimed to be able to produce the equivalent of 100,000 lines of code a year. Divide the difference between the two by factor of, say, 20, and the comparison still suggests that programming productivity could be increased by as much as 150%. On this basis, the cost of a $100,000 specialist workstation could be recovered within a year from increased output.

The reasons why these techniques should allow such a dramatic increase in productivity are discussed in more detail in the full report. At the generalised level, one major factor is that they allow a much more flexible, exploratory approach so that programs can be developed by a process of continuing experiment rather than having to be fully defined in advance. By extending the individual programmer's span of control they achieve further gains, because the problems of coordinating and interfacing separately developed program modules are reduced.

At a deeper level, the languages used in AI allow programmers to work in more generalised, abstract concepts, and expert systems methodology provides for emphasis on defining what has to be done rather than specifying how it is to be done in rigorous detail. They provide for interactive editing and debugging of programs as problems arise in the course of a run, not only at a separate editing session. And the flexibility with which they allocate memory and data types relieves the programmer of many complex decisions.

For the moment, only a small number of people in the computer industry have real experience of these benefits and are in a position to advocate the wider use of AI and expert systems techniques. But it hardly needs emphasising that if the claims being made are justified they should be taken very seriously. Users who first succeeded in putting high-productivity techniques to work would have a very significant advantage over their competition, not only in software costs, but in the range of programs which they would practically be able to produce.

The advantage applies mainly to the more complex and less well-defined software projects, with a significant symbolic content, rather than purely numerical or textual processing. Other promising applications are for tasks where computerisation was previously impossible because the specification changed faster than the system could be implemented. Systems using these techniques are already being developed in such areas as computer-aided design, image and signal processing, natural language recognition and complex interfaces, as well as for expert systems development. If the claims made for the technology are at least partly justified, then the rate of progress in these areas should be significantly speeded up.

THE COMMERCIAL OPPORTUNITIES

If expert systems are indeed a distinct new technology, with valuable applications, what new commercial opportunities will they create? The main ones can be identified by considering a model of the industry in terms of the financial flows which will drive it over the next few years (Figure 2). This reveals four different types of opportunity.

- the benefits to be gained from using expert systems
- the investment possibilities they offer
- the markets for expert systems products and services
- the demand for knowledge engineering and applications skills.

All these opportunities have their source in the benefits from using expert systems. Unless users obtain these benefits, the flow of money through the industry will soon dry up. But the factor which will limit the rate of flow for some years to come is the availability of the skills to engineer new systems and, perhaps even more important, to put them successfully to work. This factor has been used as the basis of estimates of the growth of the industry in the next few years, as discussed in the main report.

It is impossible to quantify the full scale of user benefits from expert systems. In a few cases, substantial direct savings can be identified, but in general the really significant returns will come where expert systems allow users to raise their operations to new levels of quality, service or performance. To take an example of the first kind, the "Drilling Advisor" system, developed by Teknowledge for Elf Acquitaine, could save $250,000 by preventing the loss of a single day of driling time on an oil well. On the other hand, although some manpower savings could be attributed to DEC's XCON system, its chief value is that it allows DEC to continue giving its customers a very high degree of flexibility in how they configure their systems. Without XCON, DEC's policy would probably have become insupportable, given the complexity and sales volume of the VAX range.

One can only quantify gains of this kind by indirect methods, for example by estimating the loss of market share a company would have suffered without the extra quality a new technology can provide. To a large extent the growth of the expert systems industry will depend on end-users having the vision to see these opportunities for enhancing their operations.

For the meantime, until end-user sales build up, the infant industry is heavily dependent on long-term investments from other sources - primarily research funding and venture capital. These are estimated to be running at about $60m and $20m respectively in 1984. (All the figures quoted in this chapter are rounded off and refer to the USA plus the UK at constant 1984 prices. The UK is roughly 10% of the total; fuller details are given in the full report).

Research and venture capital investments are not expected to grow very sharply; figures of $90m and $40m are estimated for the late 1980s. There is good reason for restraint; given the shortage of human skills an excessive expenditure on research and new start-ups could slow down the pace of real applications rather than speeding them up. This in itself suggests that the best opportunities for venture capital will be in operations which have assembled a strong base of knowledge engineering skills, and have an orientation to real applications. Companies which are focussed on research rather than applications are unlikely to be rewarding places to put venture capital. This type of work should be funded on the same basis as other pure research programmes rather than in hope of a highly uncertain long-term commercial payoff.

The venture capital is mainly being channeled into systems houses - meaning in this context companies which get most of their revenue from writing and selling computer software and systems. Figure 2 shows three main sources of expert systems revenue for the systems houses; custom contracts, packages, and development tools. By definition, products in the last category will be bought by organisations which are also developing expert systems, and so the cash flow generated is internal to the industry. An additional market will be created for the hardware on which to develop expert systems, and the delivery vehicles on which to run them, and this in turn will be split between stock and special-purpose hardware.

Estimates presented in the full report for each of these markets are summarised below.

	1984 ($m)	1990 ($m)
Software		
Custom contracts	19	460
Package sales	2	110
Development tools	4	66
Development hardware		
Special-purpose	18	51
Stock	11	76
Delivery vehicles		
Special-purpose	2	696
Stock	16	863

The provision of stock hardware for expert systems is naturally the business of the established hardware suppliers, with DEC far away in the lead but others showing increasing interest. But almost all the other market sectors listed are already proving active areas for new start-up companies, mostly backed with venture capital.

The listing of expert systems projects, and the companies responsible for them, in an appendix to of the report includes 18 systems houses specifically founded to serve the AI or expert systems markets - companies such as Apex, Inference Corp. or Teknowledge in the USA and Expert Systems Ltd or Helix Software in the UK. For the moment, a large part of their work is paid for by research funds, or custom contracts with a hidden research or venture capital element. Package products are still a rarity.

In the development hardware sector, Symbolics and Lisp Machine have already been mentioned, and are both heavily dependent on venture captial. Big manufacturers - Xerox, Racal and, it is expected, Texas Instruments - are also venturing into the business. But at the time of writing there is no-one yet selling special-purpose delivery vehicles. This is a market which has yet to be created.

The market estimates provided do not point to a bonanaza for these companies, or the others which will surely join them. Although the projected growth rate is quite substantial, it is much less inflated than some of the forecasts which have been made for the short-term growth of the artificial intelligence business. The reason is only partly that the figures refer just to the expert systems sector; it is also because this report takes a fairly conservative view of the market prospects. Real growth will be limited by the availability of knowledge engineers and the rate at which succesful applications can be implemented. Forecasts which ignore these constraints must be unrealistic.

The implication is that hard work and good management should bring a fair return and rapid growth for companies in the sectors identified above, but the big payoffs, and the opportunities for a few expert systems or AI-based companies to join the industry's leaders, will not come until the 1990s. None of the new AI start-ups we see today will emulate Apple's feat of coming out of a garage to become the eleventh-biggest US computer company in six years.

The one area which could see sensational growth is expert-system packages and the machines on which to run them. One or two packages, such as ExpertEast, invented by Donald Michie, show hints of becoming a universal tool to rival word-processing or spreadsheets. ExpertEase is a program which first induces rules from given examples and can then run them as a miniature expert system. Future versions could become the decision-making equivalent of a pocket-calculator for managers and system designers. And they would also help to get round the knowledge engineering bottleneck.

This leads to the last of the four areas of opportunity – the provision of knowledge engineering and applications skills. As mentioned above, the supply of knowledge engineers has been used as an indicator of the expected growth of the market, but it would have been equally valid – although harder to quantify – to base projections on the rate at which real applications experience can be gained. The difficult process of teaching and encouraging people to use novel techniques, and of adapting the techniques to serve users efficiently, in the light of experience, threatens to be the narrowest bottleneck of all. Even the few examples available to date have shown that bedding down a large expert system in a routine application is a long and difficult process; some serious research on the difficulties that arise could be very helpful to future applications.

These shortages and bottlenecks create an opportunity for individuals who have the right experience, or some way of acquiring it. They also offer some significant commercial opportunities for training, staff recruitment and information. These services can help to speed the growth of the expert systems industry by making it less dependent on experienced knowledge engineers, and spreading the benefits of experience more rapidly. The perceived need for practical information about the applications of expert systems has been one of the main motivations for producing the report, and Ovum hopes to make further contributions in this direction in future.

Fig.2 Financial flows in the expert systems industry