13

Expert systems

SHIRLEY GREGOR

This chapter explains the meaning of basic terms and concepts related to expert systems, describes the architecture and functions of an expert system, and the conditions under which the building or purchase of an expert system is feasible. The benefits and disadvantages attached to the use of expert systems will be analysed. Also included are a description of the tools and methods used for building expert systems plus the criteria used to evaluate and compare these tools. Finally the reader will gain an understanding of the difficult process of knowledge acquisition and an awareness of the legal implications of the use of expert systems.

Expert systems were one of the first areas to be commercially fruitful within the field known as **artificial intelligence** (AI). These systems are now operational in many areas, including medicine and health care. Expert systems are related to **decision support systems**, which are discussed in a separate chapter in this book. Both expert systems and decision support systems assist humans carrying out difficult tasks, but decision support systems traditionally provide assistance with complex calculations and modelling, while expert systems provide assistance with qualitative knowledge and reasoning, assisting the human memory with complex rules or regulations, and providing expert strategies for attacking a problem. There are also differences with respect to the amount of responsibility the human user takes for problem-solving methods and the final decision. With a decision support system the user is required to contribute more in choosing problem-solving methods. With an expert system the choice of methods and even the final decision may be completely, or almost completely, automated.

Artificial intelligence is a broad disciplinary field which encompasses work in robotics, computer vision and natural language processing, as well as expert systems. Definitions of artificial intelligence (Charniak & McDermott 1987) vary from 'the study of mental facilities through the use of computational models' to 'the science of making machines do things that would require intelligence if done by men' (accredited to Minsky). The beginning of the modern field of artificial intelligence is often traced to the so-called Dartmouth conference in 1956, organized by John McCarthy and Marvin Minsky. McCarthy at Stanford University, and Minsky at the Massachusetts Institute of Technology, have continued as leaders in the field of artificial intelligence, together with Allen Newell and Herbert Simon of Carnegie-Mellon University.

Expert system development began in the late 1960s. Earlier work in artificial intelligence had focussed on developing systems for general-purpose problem-solving. An example was the GPS (General Problem Solver), developed by Newell, Simon and J.C. Shaw. Though work on such systems has had enormous influence, the systems themselves were not suitable for practical application. It was not until work began on building systems for more limited areas of knowledge, where the use of specific rules and heuristics (rules-of-thumb) enabled more efficient processing, that systems could be applied successfully to practical problems. It is these narrower systems that are known as expert systems.

Perhaps the earliest expert system was DENDRAL, a system originally intended to do chemical analysis of the soil on Mars. Eventually, the program (and the computer it ran on) were too large to travel on the NASA vehicle to Mars, but the value of encoding the expertise of a human expert in a computer program was demonstrated. DENDRAL has outperformed human experts in carrying out mass-spectogram analysis of chemical structures, and has discovered errors in published literature (Parsaye & Chignell 1988). Another early and influential system was MYCIN, also developed at Stanford University (Buchanan & Shortliffe 1984). MYCIN diagnoses bacterial infections and prescribes treatment.

An expert system can be defined (Firebaugh 1988) as possessing the following characteristics:

- performance is at a level generally recognized as equivalent to that of a human expert or specialist in a particular field,
- knowledge is highly domain specific, that is, the system has a narrow range of knowledge relevant to one problem area,
- the system can explain its reasoning, being able to justify its advice, analysis, and conclusions (though not all systems can do this)
- systems can be built where some knowledge is uncertain, probabilistic or fuzzy, and give a range of alternative solutions with associated likelihoods.

The term ***knowledge-based system*** (KBS) is now used almost synonymously with the term expert system. The former term, however, has a more general meaning, to reflect the ideas that these systems can include knowledge drawn from sources other than human experts, and that systems may be useful although they are not, strictly speaking, equivalent in performance to a human expert.

Systems with some similarities to expert systems are **neural net** systems. These systems also focus on qualitative rather than quantitative reasoning, but were originally inspired by attempts to model the reasoning processes of the human brain. In the human brain, activity consists of impulses passed from one neurone to another in a complex network. Neural networks employ a *black box* approach, where the system is given sets of inputs and outputs, and *learns* the relationships between the inputs and outputs in a number of training sessions,. These learned relationships are then used to evaluate future cases. The system is a black box because it is usually not obvious to the outside observer what the learned relationships are. Thus, it is not usual for these systems to be able to explain how they reach conclusions. Neural nets are particularly good for identifying patterns. For example, a hospital system developed in South Carolina predicts length of stay and type of hospital discharge from a given diagnosis (Kestelyn 1991). It is claimed that this system has saved millions of dollars by allowing the hospital to administer resources more effectively.

Architecture and functioning of expert systems

A knowledge-based system includes a ***knowledge base***, an ***inference engine*** (which may include routines for providing explanations), and a ***user interface.*** Figure 13.1 shows the main components of a knowledge-based system.

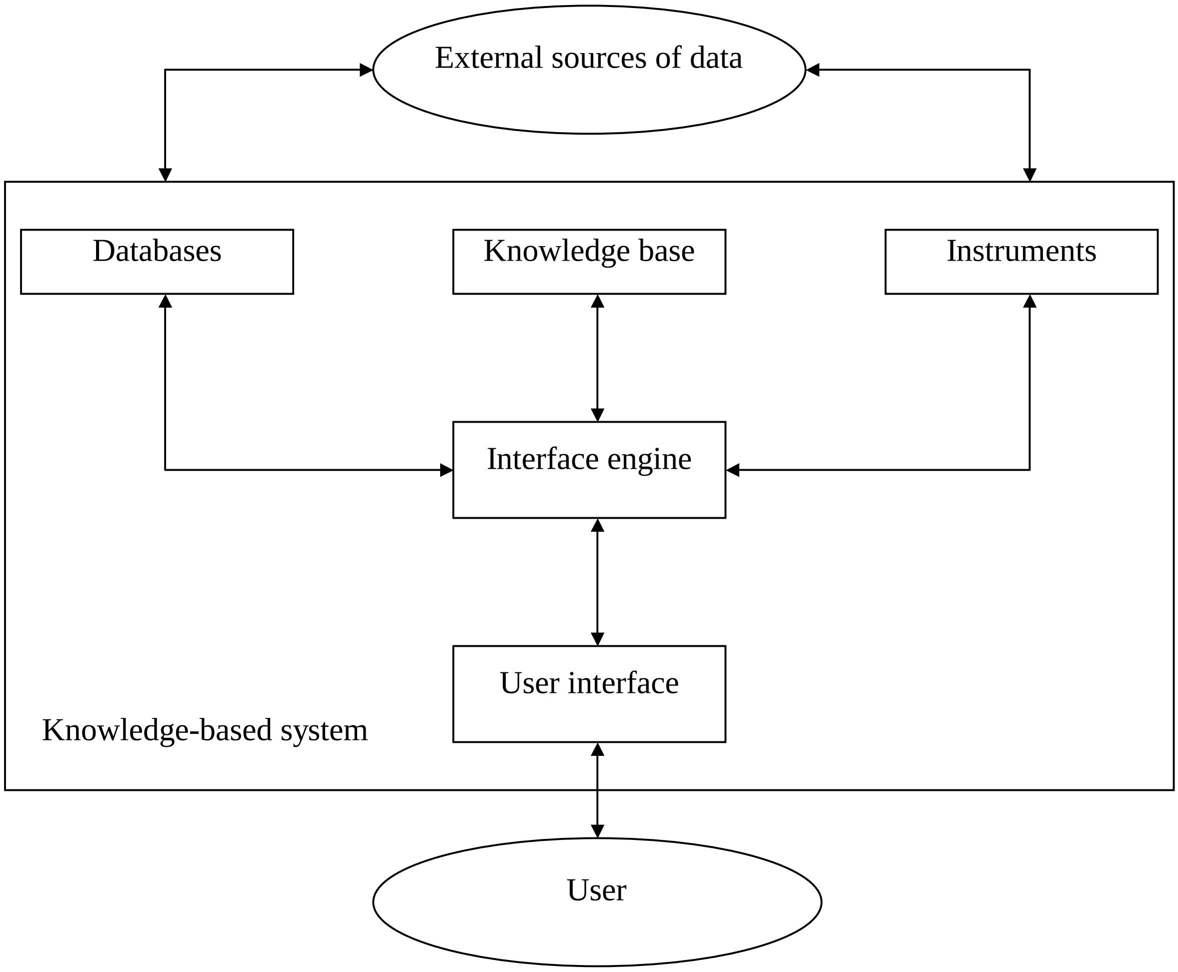

Figure 13.1 Components of an expert, or knowledge-based, system

The inference engine is the main driving component for the expert system and controls the operation and accessing of the other components. The inference engine acquires necessary information from the expert-system user through the user interface, controls the way the knowledge in the knowledge base is accessed and communicates results to the user, again through the user interface. The inference engine may have access to internal or external databases, and instruments such as sensors in process control, or devices for monitoring patients. It is possible to use the same inference engine with different knowledge bases, and in effect, this is what is done when we use an ***expert-system shell,*** a software tool that can be used to build an expert system for a particular problem area.

The user interface can be thought of as simulating the interaction which would occur when there is a consultation between a human expert and a seeker after advice. This simulation is virtual rather than substantive, as though some systems provide forms of natural language processing, most systems use interface methods similar to those in other programs: menu, question-and-answer, form fill-in and graphical interfaces. These methods appear to work well in practice. It is not clear that users wish to have interactions with computers that are imitations of human-to-human interactions.

The knowledge in the knowledge base may include implicit (tacit) or explicit knowledge from an expert, formal knowledge such as that found in texts or legislation, heuristic knowledge, procedural (problem-solving) knowledge and declarative (factual) knowledge. There are many different ways of representing this knowledge including logic, production rules, semantic nets, and frames. These methods are covered in greater detail in a separate chapter in this book. Many systems are rule-based and are relatively simple, easy to understand, and efficient in diagnostic situations. An example of a simplified rule from MYCIN is:

Rule543

If 1 the infection which requires therapy is meningitis,
2 only circumstantial evidence is available for this case,
3 the type of the infection is bacterial,
4 the patient is receiving corticosteroids,

then there is evidence that the organisms which might be causing the infection are e.coli (.4), klebsiella-pneumoniae (.2), or pseudomonas-aeruginosa (.1) (Clancey 1983).'

When an expert system is consulted about a particular problem it gathers facts specifically to do with the problem addressed. Some systems also allow for ***machine-learning,*** where new knowledge updates the existing knowledge in the knowledge base. For example, a new set of facts may lie outside of the existing knowledge, so a new rule is added to cover them. Knowledge representation systems also often allow for uncertainty, or less than perfect confidence in aspects of the knowledge held, as shown in the example from MYCIN above, where the numbers in brackets represent certainty factors.

Blackboard systems are a newer development in which the knowledge base is divided into independent knowledge sources, and each independent source communicates via a blackboard, which is organized in a hierarchical structure representing the problem to be solved.

The emphasis on the separation of knowledge from processing is the main distinguishing feature of expert systems when compared with more conventional programs. More traditional programs may have knowledge incorporated into their procedures, but it may not be explicitly recognised that this is so. For example, in a billing system, the rules for charging patients may be buried in the depths of a large program amongst the procedures for printing accounts and updating files. Recognizing that the knowledge base is a separate component is important as it allows us to think more clearly about problem areas which are specifically to do with the knowledge base: acquiring the knowledge in the first place, finding an appropriate representation method, and keeping the knowledge up to date.

When is an expert system feasible?

A survey of reports of expert system use in Australia (Parakala & Gregor 1992) identified over 60 systems in use in government, health and industrial organizations. The largest number of systems were in engineering, science and industry, in business and agriculture. Progression from research to successful commercial systems has been slow. Analysis suggests that successful deployment involves considerable effort, integration of knowledge-based systems with mainstream technology, emphasis on ease-of-use, systems which assist experts rather than replacing them, and the use of special purpose shells for developing the systems. Many successful systems, especially in business, have an automated, intelligent checklist approach.

In general, expert systems can be developed where:

- The task requires symbolic reasoning more than numeric calculations,
- Heuristic search is required more than algorithmic procedures,
- Domain-specific knowledge is more dominant than common sense,
- The task has well-defined solutions that can be specified in advance,
- The inference logic is predetermined,
- The task is of manageable size, but complex enough to benefit from an expert system.

Advantages of expert systems include:

- Increased output and productivity,
- Increased quality,
- Capture of scarce expertise,
- Flexibility in services offered,
- Reliability,
- Shorter response time,
- Integration of several experts' opinions,
- Educational benefits (training of novices may occur as a side-effect of expert-system use),
- Enhancement of problem solving,
- Use in remote locations where experts are unavailable.

Disadvantages of experts systems include:

- Updating the knowledge base may be laborious and error prone,
- The systems have no common-sense,
- Brittleness, performance does not degrade gracefully at the limits of the expert system's knowledge, and a novice user may not be aware of the limitations of the system,
- Legal responsibility for the advice offered is a "grey" area.

Examples of expert systems in use

The first expert system to be used daily in Australia was a medical system (Catlett 1990). Garvan-ES1 was introduced in 1984 at the Garvan Institute of Medical Research at St Vincent's Hospital in Sydney to assist with the interpretation of lab tests on patients' thyroid levels. The program was written in the C language, and is said to produce reports that are 99% correct, with a higher quality product at a lower cost.

Miller (1994) gives a full review and bibliography for medical diagnosis decision support systems from 1954 to 1993. Miller states that medical diagnosis systems, many simple, and some complex, are now ubiquitous and such systems have become an established component of medical technology. In particular, he notes many highly successful systems are specialized and focussed. For example, commercial systems for EEG analysis are now in widespread use. Systems for cytologic recognition and classification have also found successful application in devices such as automated differential blood count analysers.

Further examples of expert and decision support systems used in nursing can be found in Ozbolt, Vandewal, Hannah (1990).

Developing expert systems

Guidelines for the development of expert systems exist, though a variety of approaches is available. Prerau (1990) gives useful guidance. One point of reasonable general agreement is that prototyping and some form of evolutionary or iterative design is needed. A **prototype** is a rough, small-scale system which is developed quickly. The prototype helps the users to see what they really want and express what needs to be done to achieve this. Using a prototype must be an iterative process, as a number of cycles through the adaptation of the prototype and retesting on users is necessary.

Weitzel and Kerschberg (1989) argue that knowledge-engineering can be done by average people (ordinary systems analysts) rather than high-priced **knowledge engineers**, who are specialized staff with knowledge of the technical side of expert system development. These authors feel that new software and new development methods are needed. Shells and programming environments facilitate prototyping by allowing knowledge engineers to focus on the problem instead of coding. Definition of the problem and feasibility assessments are more difficult and are done tentatively at first, becoming firmer in subsequent passes. Identifying the conceptual structure that underlies the expert's thinking is important and differs from the basic systems analysis tasks in transaction processing systems. Objects about which data are to be gathered for a transaction system may be peripheral details in a knowledge-based system. What is more important is how the experts make decisions; for example, what concepts require value judgements from the expert. Multiple points of view during conceptual design are important, to allow different knowledge representations. It is important to have validation carried out by people outside the development process so that assumptions peculiar to individuals are uncovered.

It is still important to use methods from conventional system development that are considered good practice, when developing and testing expert systems. These methods include the use of structured programming techniques when appropriate, the reduction of the complexity of programs by the use of relatively self-contained modules, documentation of coding and use of meaningful data names, an emphasis on program readability and maintainability rather than efficiency, and the use of configuration management and control procedures. The validation of knowledge and the assessment of expert system performance can be extremely difficult. It should also be recognized that validation of knowledge should be an ongoing process, a system cannot be specified as correct at just one point in time.

Knowledge acquisition

Knowledge acquisition refers to the extraction and formulation of knowledge from various sources, especially from human experts. Knowledge acquisition is recognized as one of the particularly difficult and time-consuming aspects of expert system development, which leads to the concept of a knowledge engineering bottle-neck.

Approaches to knowledge acquisition include:

- Interviewing, where the knowledge engineer obtains knowledge from the human expert through a series of interviews and encodes it in the expert system. These interviews may be structured or unstructured and questionnaires may be used.
- Learning by interaction, where experts directly interact with a computer program that helps to capture their knowledge. A technique can be used called repertory-grid analysis, where relationships between actions and results can be developed and tested.
- Verbal protocol analysis, a process tracing method, where individuals are asked to think aloud while they solve a problem (Ericsson & Simon 1980).
- Learning by induction or machine learning, where a computer program distils knowledge by examining data and examples.

Interviewing is very common. Learning by induction is not uncommon, and can be done with even moderately priced shells, for example, VP-Expert. In Australia, Ross Quinlan, of the University of Sydney, has made significant contributions in the area of machine learning, having developed “Interactive Dichotomizer 3” (ID3), which uses classification to learn the essential features of a set of examples. ID3 has provided the basic inference algorithm for a number of commercial systems.

Also, a considerable number of systems rely heavily on the systemized knowledge codified in manuals, regulations and books rather than on knowledge extracted directly from experts. A survey of reported systems in the legal domain (Gregor & Watts) showed that systems which were legislation based had reached more advanced stages of development than other types.

Keeping knowledge in systems current is a major undertaking. A novel approach is reported by Edwards et al (1993), who, in work following from Garvan-ES1, have introduced the notion of ripple down rules. Newer rules are patched on to the old rules at an appropriate point. The authors report that this approach allows the knowledge base to be easily maintained and updated by the pathologists using the system.

There are a number of problems in knowledge acquisition, including:

- Insufficient availability of the expert
- Lack of enthusiasm by the expert
- Communication difficulties between the expert and the knowledge engineer
- Lack of domain knowledge by the knowledge engineer
- Inarticulateness of the expert
- Implicit or tacit knowledge held by the expert, which is not accessible to awareness
- Knowledge is only collected from the first available or most convenient source. For example, the knowledge engineer might rely on books or manuals when these contain

parts that are not used by experts in practice, or have exceptions that the expert has discovered and learned to work around.

- Irrelevant knowledge is collected from the expert. One way of identifying which knowledge is really relevant is by comparing the knowledge of people with different levels of expertise on the task. Critical knowledge will be indicated when it is known by experts but not by those with less expertise.
- The range and flexibility of the expert's knowledge is not sufficiently explored. A working system can be expected to encounter many different situations in the field. Thus it is essential that its knowledge base contain as many exceptions and unusual cases as possible.
- Difficulties in observing behaviour. Experimenter effects, where people's behaviour changes when they know they are being observed, are well-known.

Johnson's (1983) article on the development of expertise is interesting for its insights into the problem of implicit knowledge. Johnson observed that a colleague's teaching of medical diagnosis differed from how he did the diagnosis himself. He did not teach what he seemed to do. When asked about this, the colleague said: 'Oh, I know that, but you see I don't know how I do diagnosis, and yet I need things to teach students. I create what I think of as plausible means for doing tasks, and hope students will be able to convert them into effective ones.' This anecdote illustrates both the problem of reconstructed methods of reasoning, and implicit or tacit knowledge. Implicit knowledge is knowledge acquired as expertise develops,which is no longer available to the expert's own awareness. The protocol method of knowledge acquisition is one means of uncovering implicit knowledge. For a more detailed discussion of knowledge acquisition see Turban (1992).

Tools used in development

When building expert systems, the choice of tools to assist in the task includes:

- Conventional programming languages, for example LISP, Prolog, C,
- Special-purpose programming languages, for example, OPS5
- Expert system shells, which are special-purpose environments with most of the mechanisms needed in an expert system already present. Just the knowledge base needs to be added.

Brody (1990) gives a good discussion of the relative merits of various expert system shells, as well as a checklist of features which can be used when comparing different products. Stylianou, Madey and Smith (1992) give a comparison of the perceived importance of selection criteria for expert system shells.

Following are some of the important question to ask before purchasing a shell.

- How much does it cost?
- Can run-time versions of the program be produced, to execute separately from the shell? This is important if you wish to have the program used by many users. Otherwise, each user who uses the program must pay for the licence to run the shell.
- Is the final program as efficient as needed: does it operate quickly enough? In some cases, especially where a separate run-time version of the program cannot be obtained,

the program plus the shell will need a large amount of disk space and will be slow once the system reaches any appreciable size.

- What does the user interface of the final system look like? Is it easy to use? Can a graphical user interface, like Windows, be produced?
- Can explanations of the system's reasoning be produced?
- Is the shell easy to use and well-documented?
- Is local support available in case of difficulties?
- Is the system compatible with other languages? For example, it may be necessary to link to routines written in some other language such as C.
- Is the system compatible with external files and databases? For example, it may be necessary to access data held in Lotus 1-2-3 or dBase format.
- Is there a subroutine library to carry out commonly required statistical routines, date manipulations and other calculations?
- What forms of knowledge representation are available? Opinions include rules, objects, and frames. Many systems now offer more than one form of knowledge representation, and are known as hybrid systems.
- What methods are used to control reasoning? Options include forward and backward chaining.
- Is it possible for the system to learn by induction from examples or from updated knowledge bases?

Legal issues

Legal issues surrounding experts system development and use are still somewhat of a 'grey' area. There have been insufficient cases to establish precedents, though Mykytyn and Mykytyn (1991) discuss eight legal cases in the USA pertaining to expert systems.

Organizations, however, should be aware of potential liabilities that necessitate careful planning in the testing, evaluation and documentation of systems. Potential liabilities arising from the use of expert systems include the following (Mykytyn, Mykytyn & Slinkman, 1990):

- The erroneous, incomplete, or conflicting opinions of experts could lead to the development of a faulty system. In this case, the experts may be held liable. Developers should chose experts carefully and carry out thorough reliability testing.
- The knowledge engineer might be held liable for errors or incompleteness in the knowledge base if these were thought to result from incompetence, bias or negligence. The knowledge engineer should document the development process carefully to guard against such charges.
- An organization selling an expert system might be held liable for defective advice, especially if the system is regarded as a service, rather than a product, which could be covered under a warranty.
-

Consideration can also be given as to whether there are legal consequences from the non-use of an expert system. For example, in the absence of a doctor, a nurse may choose not to use an available expert system which could, potentially, give critical advice.

The provision of explanations by a system may be relevant to the question of liability. If a system has the capability of explaining how it reaches conclusions, the users do have some opportunity to verify for themselves the accuracy of the conclusions and to use their own judgement in qualifying the system's advice. It is interesting to note, however, that users often express little interest in explanations. Botsman and Smith (1992) report on the development of a drug advisory system at Mackay Base Hospital where there is a narrow margin between therapeutic and toxic levels of a drug and, in the worse case, inadequate prescription could result in death. The authors comment that 'There was very little curiosity - even after some gentle prodding - about the mechanics (How and Why) of the system' (p. 60).

Conclusions

Expert systems are already being used to advantage in many fields. It seems probable that use will continue and increase. One prediction (Turban 1992) is that expert systems, as well as other applications of artificial intelligence, will tend increasingly to be embedded in traditional software. A recommendation to readers is that they obtain one of the lower priced, or public-domain, expert system shells and experiment with building a simple system themselves. This experience will help with the appreciation of the material covered in this chapter, and could lead to some useful systems!

Review questions

- Which component of an expert system is primarily responsible for reasoning?
- Distinguish between knowledge acquisition and knowledge representation.
- List some limitations of expert systems.
- Describe some problems for which an expert system might be a solution. Also describe some features of other types of problems which are likely to make them unsuitable for an expert system solution.
- Your organization has asked you to assess the implications of moving towards the use of expert systems. List the factors you would consider in making recommendations to the board of directors.
- Why might evaluation of the worth of an operational expert system be difficult?
- List and discuss some difficulties associated with the development and use of expert systems.

References

Botsman K, Smith J D 1992 A drug dosage advisory system for use by physicians and others: experience with the implementation of an exemplary system. In: Adams A, Sterling L (eds) Proceedings of the 5th Australian joint conference on artificial intelligence AI'92, Hobart, pp. 57-63

Brody A 1989 The experts. Infoworld. 11(25): 59-75

Buchanan B G, Shortliffe E H (eds) 1984 Rule-based expert systems the MYCIN experiments of the Stanford heuristic programming project. Addison-Wesley, Reading

Catlett J 1990 Expert systems the risks and rewards. Hub Information Technology 2(7): 20-26

Charniak E M, McDermott D 1987 Introduction to artificial intelligence. Addison-Wesley, Reading

Clancey W J 1983 The epistemology of a rule-based expert system - a framework for explanation. Artificial Intelligence (20) 215-251

Edwards G, Compton P, Malor R, Srinivasan A, Lazarus L 1993 Piers: a pathologist-maintained expert system for the interpretation of chemical pathology reports. Pathology (25): 27-34

Ericsson K A, Simon H A 1980 Verbal reports as data. Psychological Review (87) 215-251

Firebaugh M W 1988 Artificial intelligence a knowledge-based approach. PWS-Kent, Boston

Gregor S D, Watts J 1991 The use of intelligent systems in taxation practice. In: Proceedings of the annual conference of the accounting association of Australia and New Zealand, Brisbane

Jackson P 1990 Introduction to expert systems. Addison-Wesley, Wokingham

Johnson P E 1983 What kind of expert should a system be? Journal of Medicine and Psychology (8): 77-97

Kestelyn J 1991 Application watch. AI Expert (Feb): 71-72

Lucas P, Van der Gaag, L 1991 Principles of expert systems. Addison-Wesley, Wokingham

Miller, R A 1994 Medical diagnostic decision support systems - past,present,and future. J Am Med Informatics Assoc. (1): 8-27

Mykytyn K, Mykytyn P. P. (jr), Slinkman C W 1990 Expert systems: a question of liability. MIS Quarterly (March): 27-38

Mykytyn P P (jr), Mykytyn K 1991 Legal perspectives on expert systems. AI Expert (Dec): 41-45

Olson D L, Courtney J F 1992 Decision support models and expert systems. Macmillan, New York

Ozbolt J G, Vandewal D, Hannah K J (eds) 1990 Decision support systems in nursing. Mosby, St.Louis

Parakala S R K, Gregor S D 1992 Critical factors in the successful implementation of knowledge-based system technology in Australia: a recent survey. In: Proceedings of the Australian Computer Society Queensland branch conference, Bond University

Parsaye K, Chignell M 1988 Expert systems for experts. John Wiley, New York

Prerau D S 1990 Developing and managing expert systems proven techniques for business and industry. Addison-Wesley, Reading

Stylaniou A C, Madey G R, Smith R D 1992 Selection criteria for expert system shells: a socio-technical framework, Communications of the ACM, 35(10): 30-48

Turban E 1992 Expert systems and applied artificial intelligence. Macmillan, New York

Weitzel J R, Kerschberg L 1989 Developing knowledge-based systems: reorganizing

the system development life cycle. Communications of the ACM (32) 4: 482:488

Zahedi F 1993 Intelligent systems for business expert systems with neural networks. Wadsworth, Belmont