1-1-1990

The development of an expert system for the diagnosis of diseases in fibre and dairy goats

Tim S. Roberts
Edith Cowan University

USE OF THESIS

A Submission

for the Award of

Master of Applied Science

(Computer Studies)

The Development of an Expert System for the Diagnosis of Diseases in Fibre and Dairy Goats

Tim S Roberts B.Sc.(Hons),

PostGradDip.AppSci.(Comp Stud) (Distinction)

School of Arts and Applied Sciences

Western Australian College

of Advanced Education

March 1990

ADDENDUM

The GOATS Expert System was entered in the Royal Agricultural Society's annual Farm Software Competition for 1990.

It was awarded first prize in its section, on a unanimous decision of the judges. The system also won the special Fujitsu sponsor's award.

And what is good, Phaedrus, and what

is not good -

need we ask anyone to tell us these things ?

Robert M Pirsig

Zen and The Art of

Motorcycle Maintenance, 1974

ABSTRACT

This thesis details the development of an expert system for the diagnosis of diseases in fibre and dairy goats.

Divided into five sections, five appendices, and a bibliography, this thesis centres on the methods used to build the expert system; the decisions taken at the outset of, and during the course of, development; some of the problems encountered, and the solutions to those problems. A detailed appraisal is made of the development process and suggestions are made for future developments over similar domains (for example, the diagnosis of diseases in animals other than goats).

Much emphasis is placed on three topics in particular: the selection of the expert system tool(s) to be used (and the rejection of numerous others); the methodology employed for this selection process; and the methodology used for the process of development.

Other topics which are routinely found in texts on expert systems (for example, knowledge elicitation techniques, explanatory facilities, expert system evaluation etc) are dealt with only briefly. However, for the reader interested in further information on these topics, references are made in the

text to appropriate sources.

An elementary knowledge of expert system technology is assumed; for the novice, a brief glossary of some of the terms used throughout this thesis is provided in the appendices.

I would like to acknowledge the contributions of the domain expert, Dr Sandra Baxendell BVSc(Hons) PhD MACVSc, for her vital input to the current work, and of my supervisor at W.A.C.A.E., Dr Jim Millar BSc(Hons) PhD MACS.

I certify that this thesis does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any institution of higher education and that, to the best of my knowledge and belief, it does not contain any material previously published or written by another person except where due reference is made in the text.

Tim S Roberts **January 1990**

CONTENTS

1 Introduction

Expert systems are computer programs that mimic the behaviour of human experts in some specified domain of knowledge (Liebowitz, 1988). As a body of knowledge, expert systems is a branch of artificial intelligence, which is itself a part of the much wider field known as computer science.

Expert systems research is still very much in its infancy; the situation is not unlike that concerned with conventional software development in the 'sixties. Successful systems are still relatively few in number, and tend to be based on very small and specialised domains.

The expert system normally considered the fore-runner of all others is DENDRAL (Feigenbaum, 1971), a program for interpreting mass-spectrogram results. But the most well-known, and certainly most influential, expert system remains MYCIN (Shortliffe, 1976), a system developed in the mid-seventies to aid in the diagnosis of infections of the blood. Although there have been a number of extensions, revisions and abstractions of MYCIN since 1976, the system is still primarily used for the training of human diagnosticians, rather than for actual diagnosis (Jackson, 1986).

Research is currently being undertaken on a wide-front. Amongst the areas of expert systems development receiving most attention are the processes of knowledge elicitation and acquisition; machine learning; reasoning under uncertainty; and the processes of explanation and justification. Readers interested in any of these four topics are referred to (Boose, 1988), (Lenat, 1988), (Lemmer & Kanal, 1988), and (Jackson, 1986) respectively for descriptions of the state-of-the-art.

In addition, enhanced versions of artificial intelligence languages, such as Lisp and Prolog, continue to be developed and marketed to an increasingly enthusiastic audience of AI researchers; and new and more sophisticated expert system shells are appearing that claim to make the development of expert systems a more efficient and less time-consuming process (for example, advertisement for PC-Easy and PC-Plus in AI Expert, January 1987, p58).

This thesis concerns the building of an expert system that is intended to exhibit expertise over a fairly broad domain - that is, the diagnosis of diseases in goats (a detailed discussion of the objectives is provided in section 2). No such expert system is currently available in Australia, and research has failed to discover any similar product elsewhere in the world.

The goat is a member of the family Bovidae (Artiodactyla Ungulates) -

cloven-hooved mammals which include cattle and sheep. Products from goats include flesh and milk, skins, hair, and wool.

The goat population is spread across all the continents of the world, with the exception of Antarctica. The greatest concentrations are to be found in the Mediterranean area, parts of India, China, the West Indies, and West Africa (Jeffery, 1970).

In Australia, goats are on the fringe of commercial activities. Indeed, the Australian Year Book does not mention them at all, either in terms of their existence as livestock or in terms of their contribution to the economy (Australian Bureau of Statistics, 1988). Nevertheless, there are in excess of half a million domesticated animals in Australia, primarily kept for milk, cashmere, or mohair. Production of milk is in excess of one million litres per annum, and fleece production is currently close to 600,000 kilograms per annum (Australian Bureau of Statistics, 1989). Detailed figures of goats and goat products, broken down by state and type, are given in Appendix 3.

The author of the development discussed in this thesis is a computer scientist, with only minimal knowledge of the diseases of goats (gleaned from a 14-week course undertaken at Bentley TAFE in early 1988, on Goat Husbandry). Successful development would therefore depend, as is usually the case with such developments, on successful cooperation between the author and one or more domain experts.

This thesis details the objectives of the development, the problems that were encountered, and the results that were obtained. However, by far the greater part of the work is devoted to a discussion of the methods and methodologies used. Finally a detailed appraisal is given, together with suggestions for developers of expert systems over similar domains. Appendices and a comprehensive bibliography are provided at the end.

2 Objectives

The ultimate objective of the work (of which the research leading to this submission forms a major part) can be stated as follows:-

> **1. The successful development of an expert system which is usable, useful, and affordable to veterinarians throughout Australia in the part of their work that involves the diagnosis of diseases in fibre and dairy goats.**

The secondary objective of this work, which this submission aims to fulfil, is:-

> **2. The development of strategies and techniques which will enable expert systems based on similar domains (for example, diseases of animals other than goats) to be developed more efficiently than would otherwise be the case.**

These objectives will be dealt with in turn.

The first objective differs from that originally proposed in only two respects: 'Australia' has replaced 'Western Australia', and 'fibre and dairy goats' has replaced 'dairy goats'. The reasons for these two changes are outlined in section 3.1.1.

There are many implications arising from objective 1.

To be usable,

* the system should run on hardware that is commonly available to veterinarians;
* a minimal amount of computer knowledge should be expected on the part of the user;
* the language and terms used by the system should be those that are generally understood by veterinarians; and
* consultation times should not be excessive.

To be useful,

* the final diagnoses should be reliable;
* the system should demonstrate a level of expertise at least equal to that of the veterinarian; and
* the expert system should compare favourably with other available aids, such as text books.

To be affordable,

* run-time copies of the system should be able to be made and distributed easily and cheaply.

All of these factors have ramifications which are discussed in the relevant sections.

The second objective pinpoints the fact that the research into this development has an importance outside of the development itself; that is, it is reasonable to expect that lessons learned from this work should make future developments over similar domains less difficult and less time-consuming - and therefore, from a commercial point of view, more cost-efficient.

Although section 5.2 is specifically devoted to addressing the lessons that other developers can learn from this work, it is felt that maximium benefit would be gained from a reading of the whole thesis, in order that the reasons for the decisions taken in the current development may be fully appreciated.

3 Methods

This section details the methods and methodologies that were used for the development of the expert system.

Particular attention is focussed on the decisions made at the outset of the project, such as the delineation of the domain boundaries, the selection of the domain expert(s), and the criteria used for selection of the tool(s) to be used in constructing the system.

Further sub-sections discuss knowledge acquisition, knowledge representation, inferencing, prototyping, testing and validation, and documentation, as they relate to the current development.

As with any development of this nature, changes to, and refinements of, the methods used were an inevitable result of particular difficulties encountered along the way. As one of the objectives of this research was precisely to pinpoint these difficulties and outline solutions, these are described in a later section.

3.1 Initial Choices

If development of the expert system was to prove successful, much would depend on choices to be made at the outset of the project. These included the selection of:

1. The domain boundaries.
2. The domain expert(s).
3. The development methodology.
4. The tool(s) to be used.

The choice of the domain expert(s) and of the development methodology to be used occurred at the very earliest stages of the project. The domain boundaries were refined throughout the first few months.

The choice of the tool(s) to be used seemed likely to be one of the most significant decisions in the project, as it would have a bearing on most other planning decisions, and would affect in a major way all aspects of development. It is therefore dealt with at length in section 3.2, and forms an important component of this thesis.

3.1.1 Selection of the domain boundaries

Given the general objective of developing an expert system for diagnosing diseases in goats, a number of questions relating to the boundaries of the domain soon became apparent. These were:-

1. Which categories of goats should be considered ?
2. What assumptions (if any) should be made about the geographical location ?
3. Which diseases should the system consider ?
4. To which class of users should the system be targeted ?

Each of these will be discussed in turn.

Which categories of goats should be considered ?

Goats are commonly classified into one or other of three categories - dairy (or milk) goats, fibre (or hair) goats, and feral (or wild) goats. Each category is further divided into breeds (for example, breeds of dairy goat include Saanens, Toggenburgs, British Alpines, and Anglo-Nubians). Individual animals may be cross-breeds not only within categories but also between categories.

Initially, it was intended to develop a system specifically for dairy goats; however, very early in the development it was realised that this was unnecessarily restrictive, and it was decided to build a system which would encompass all categories. (However, since the system is to be used primarily for husbandry, it is anticipated that the major use of the system will be to cater for dairy and fibre, rather than feral, goats.)

What assumptions (if any) should be made about the geographical location ?

Any expert system is inherently dependent on the integrity of the knowledge provided by the domain expert; and thus, if the expert's knowledge is parochial to any degree, this may be expected to be reflected in the expert system itself.

This implies that limits (in this case geographical) should be placed on the use of the expert system to ensure that it does not fall into disrepute (that such limits exist is not unreasonable - a general practitioner in metropolitan Perth would not be expected to be equally at home in rural India, for example).

It was therefore decided, following selection of the domain expert, that the system should be restricted to Australia and New Zealand.

It should be stressed here that these restrictions are of major importance, in that they are not easily changed. For an enhancement to be made at a later date to cater for, say, the United States, would require not only many additions, but also a major rewriting of the existing system. Diseases already catered for may be more or less prevalent in a different area, and may show different symptoms or require different treatments. Tests may be carried out in different ways, and results recorded using different measurements.

There are no recognised techniques for changing an existing expert system on such a broad scale. It should therefore be appreciated that the objective of the current work is limited to some extent by the geographical region under consideration.

Which diseases should the system consider ?

It was decided at a very early stage that the system should cover as broad a range of diseases as possible, in order that the system could be of maximum benefit as an aid to diagnosis. At best, this meant that in excess of 200 distinguishable diseases should be included in the system. This number would be sufficient to include all diseases recorded in Australasia, as well as some that would normally be considered exotic.

To which class of users should the system be targetted ?

This question was particularly relevant when deciding upon the type of English to be used when asking questions, and the quantity and quality of help to be provided to the user.

It was decided very early in development that one of the objectives of the system was, in part, to be useful as an advisory system to qualified veterinarians. It was therefore felt that sophisticated terms (such as Opithotonous, Nystagmus, etc) not only could be used, but indeed should be used, in order to positively discourage use of the system by others.

However, to aid development, evaluation, etc., it was decided to incorporate into the system a number of help facilities, some of which specifically explain the terminology used. These may be easily removed in any future production version.

3.1.2 Selection of domain expert(s)

For any development of an expert system to be successful, (Keller, 1987, p13) states that "there must be at least one person who is provably better at the task than non-experts, and who is willing and able to be active on the project long enough to develop the knowledge base to maturity."

The first question to be answered on the selection of the domain expert(s) for the diagnosis of diseases in goats was not who, but how many.

The second question was how to ensure the expert(s) true expertise in the field.

The third question was how to ensure the expert(s) cooperation and availability.

These questions will be dealt with in turn.

In a situation where knowledge acquisition is likely to be a major bottleneck, as is the case with most expert systems development, the advantages of having more than one expert include the following:-

* the temporary unavailability or absence of a single expert should be less crucial to the successful completion of the project
* the experts themselves may feel more willing to cooperate in a joint venture rather than one in which their's is the sole responsibility
* the knowledge acquired may be more 'reliable' because of the synthesis of different opinions

Many articles have been written about the process of knowledge acquisition with multiple experts, for example (Shaw, 1988).

However, there are many dangers inherent in using more than one source of expertise. (Keller, 1987, p30) recommends starting with as few experts as possible "...otherwise the amount of uncertainty about the decision process may become overwhelming".

(Shaw, 1988) makes the valid points that experts may not agree on their terminology in talking about a topic, and may not even agree amongst themselves about the topic itself. For the current project, any such disagreement could place the development as a whole in jeopardy.

(Bowerman, 1988, p262) says that the use of too many experts tends to create a knowledge base that is broad and shallow, and also that "employing a single expert tends to remove expertise conflicts in the early stages of development ... starting with one expert at least gives a coherent view...".

One other fact that had to be considered was the scarcity of available experts in the area. For this and the reasons mentioned above, it was decided to proceed with a single domain expert.

The second question proved the easiest to resolve. Dr Sandra Ann Baxendell BVSc(Hons) PhD MACVSc had recently moved to Western Australia, to run goats on a farm outside Gidgegannup. She had graduated from the University of Queensland in 1975 with a final year essay concerning pregnancy toxaemia in goats, had been awarded a university medal in 1976, and had obtained a PhD in 1986.

She established the Marawah Goat Stud back in 1971, and has carried out much goat research for the Queensland Department of Primary Industries, which she joined in 1980.

She has written extensively about many areas of goat husbandry. In 1988, she wrote "The Diagnosis of the Diseases of Goats" for the Vade Mecum Series for Domestic Animals (Baxendell, 1988), which enjoys a very high reputation amongst veterinarians throughout Australia.

The close proximity of such a prominent expert in the field, together with her distinguished research record and practical experience, made Dr Baxendell an outstanding choice for domain expert.

The third and final question to be resolved was the proposed expert's cooperation and availability.

(Bowerman, 1988, p262) states that key considerations are "...the willingness and availability of the expert to work on the project, the clarity of the expert in verbalizing problem-solving techniques, and the personality matches between the expert and the knowledge engineer".

Initial contact with Dr Baxendell was made in December 1988; without reservation, she expressed interest in becoming involved with the project, and later confirmed her willingness and availability to cooperate with the development.

A subsequent face-to-face meeting at a farm in Gidgegannup was successful both on a personal and professional basis, and a good rapport was established.

Thereafter it was decided that development would proceed with Dr Sandra Baxendell as the domain expert.

3.1.3 Selection of development methodology

The development approach that was decided upon was not innovative. It was summarised effectively by (Bielawski & Lewand, 1988) as comprising four stages:-

1. Problem and resource identification.
2. Development tool selection.
3. Prototyping and system building.
4. Testing, validation, and maintenance.

Prototyping has long been established as the preferred methodology for developing expert systems, and is now rapidly gaining supporters as the preferred method for conventional software development as well - for arguments in favour of this see, for example, (Gilb, 1988).

The traditional life-cycle model, with its emphasis on the pre-implementation stages of analysis and design, is inappropriate for a system based on heuristics and rules, and where a lengthy period of cooperation with a domain expert is necessary. The rationale for conventional life-cycle design is to improve the correctness of programs; however, most software failures result from faulty specification rather than faulty coding, and "...prototypes help to make the full implications of the specification and design explicit as early as possible" (Black, 1986, p67).

The prototyping approach is discussed in more detail in section 3.6.

For such an approach to be successful, it must be possible to modify the existing system quickly and efficiently. While advocating the prototyping approach, (Black, 1986, p67) goes so far as to say "it is feasible to take .... (the prototyping) approach only if appropriate high-level tools exist". The next section deals with this point.

3.2 Selection of tool(s)

One of the first decisions to be made, and certainly one of the most important, was whether to build the expert system from scratch, or whether to make use of one or more of the many expert system development tools now available.

Most commercially available tools minimally consist of some sort of inferencing mechanism, together with an I/O interface (of varying sophistication) intended to shield the ultimate user from the internals of the system. Such tools are commonly called shells. Many of these shells contain other features in addition, such as explanatory facilities, or automated knowledge acquisition modules.

Conventional software development, high-level languages such as COBOL or PL/1 are often used in preference to machine-code, not because they offer additional features (they do not), but because they make development and maintenance very much more efficient and economical (at the expense of a certain degree of flexibility).

The same is true of the use of shells in the development of expert systems: such shells are primarily used to reduce the amount of time needed for effective development. Given infinite time and resources, every system would benefit from being written from scratch, using either a conventional language such as the two mentioned above, or an artificial intelligence language such as Lisp or Prolog. In the extreme case, of course, there is no reason why machine code could not again be used.

However, practical considerations must prevail in all but purely hypothetical situations, if any worthwhile end-product is desired. As one of the objectives of the current work was indeed to make progress towards such an end-product, it was decided that an expert system shell should be used, provided only that a suitable shell could be found.

The selection of an appropriate shell formed a major part of the development process. Sections 3.2.2 through 3.2.5 detail how this selection was made.

3.2.1 Conventional and AI languages

The decision to use an expert system shell (provided only that a suitable one could be found) meant that any discussion as to the merits of conventional versus AI languages became irrelevant. Such discussion is therefore beyond the scope of this thesis.

There are many books available for the reader interested in learning Lisp or Prolog, and these are not detailed here or in the bibliography. However, those who are interested in a comparison between these two languages, or between them and more conventional languages, may like to have their attention drawn to the following sources:-

(Liebowitz, 1988) for a very brief treatment of the history and major features of Lisp and Prolog.

(Arnold, 198[illegible]) for a good overview of Lisp.

(Keller, 1988) for a good treatment of the use of conventional and AI languages in building an expert system, focussing on the facilities provided by Prolog.

(Charniak, 1985) for an excellent description of the facilities needed in different branches of AI, with a fairly in-depth treatment of Lisp.

Starting from the above sources, the interested reader should have no difficulty in finding other material on this topic.

3.2.2 Expert system shells

Thirty-nine shells were considered for use in the expert system development. As is usually the case in any such project, it was not possible to review each of these products first-hand. Instead, extensive use was made of reviews in the available literature.

The shells considered were the following:-

1. Advisor ((Bowerman, 1988))
2. AGE ((Hayes-Roth, 1983))
3. Aion ((Bowerman, 1988))
4. ART ((Bowerman, 1988), (Gilmore, 1985), (Harmon, 1988))
5. Duck ((Gilmore, 1985))
6. EMYCIN ((Hayes-Roth, 1983))
7. Envisage ((Bowerman, 1988))
8. ESE ((Bowerman, 1988), (Harmon, 1988))
9. ESP Advisor ((Bowerman, 1988))
10. ESP Frame-Engine ((Bowerman, 1988))
11. EXPERT ((Hayes-Roth, 1983))
12. Expert-Ease ((Bowerman, 1988), (Harmon, 1988))
13. Expert-Edge ((Bowerman, 1988))
14. EXSYS ((Bowerman, 1988), (Harmon, 1988))

15. ExTran 7 ((Harmon, 1988))

16. 1st-Class ((Bielawski, 1988), (Harmon, 1988))

17. Goldworks ((Bowerman, 1988))

18. GURU ((BOwerman, 1988), (Harmon, 1988))

19. IKE ((Bowerman, 1988))

20. Insight 2 ((Bowerman, 1988), (Harmon, 1988))

21. KAS ((Hayes-Roth, 1983))

22. KEE ((Bowerman, 1988), (Gilmore, 1985), (Harmon, 1988))

23. KES ((Bowerman, 1988), (Gilmore, 1985), (Harmon, 1988))

24. Knowledge Craft ((Bowerman, 1988), (Gilmore, 1985))

25. KWB ((Bowerman, 1988))

26. M1 ((Bowerman, 1988), (Gilmore, 1985), (Harmon, 1988))

27. MicroExpert ((Cox, 1984))

28. Nexpert ((Harmon, 1988))

29. PC-Easy ((Bielawski, 1988), (Bowerman, 1988))

30. PC-Plus ((Bowerman, 1988), (Harmon, 1988))

31. Picon ((Bowerman, 1988))

32. ROSIE ((Hayes-Roth, 1983))

33. Rulemaster ((Bowerman, 1988), (Gilmore, 1985))

34. S1 ((Bowerman, 1988), (Gilmore, 1985))

35. SAGE ((Bowerman, 1988), (Keen, 1984))

36. Super Expert ((Bielawski, 1988), (Harmon, 1988))

37. TIMM ((Bowerman, 1988), (Harmon, 1988))

38. TIMM-PC ((Bowerman, 1988))

39. VP Expert ((Bielawski, 1988), (Harmon, 1988))

Several of these tools have changed their names over time, or are known by pseudonyms. Expert-Ease was previously known as Expert One; Goldworks was known as Acorn; ESE as ESCE/ESDE; EXSYS as XSys.

EXPERT refers to the system developed at Rutgers University in New Jersey, and should not be confused with a multitude of other, lesser tools of the same name.

KWB is also known by its full name, Knowledge Work Bench; the 'PC' in both PC-Easy and PC-Plus stands for Personal Consultant; and KAS stands for Knowledge Acquisition System.

This list of tools should not be considered to be comprehensive. In particular, it includes only those tools which could be described as shells; that is, tools which have some form of inferencing mechanism and explanatory interface built in. It specifically does not include the numerous languages that are now available, such as the many dialects of Lisp and Prolog. However, a couple of tools (AGE and ROSIE) that seem to be on the border between languages and shells have been included.

Many of these shells exhibit common features or backgrounds; for example, EXSYS, 1st-Class, PC-Plus, and S1 are all based on EMYCIN. Several of the shells on the list are currently in the process of enhancement. Also, new expert system shells are appearing all the time, and older ones are becoming unavailable. The reader is cautioned, therefore, that this list, while currently up-to-date, could quickly become misleading unless used in conjunction with current literature.

3.2.3 Selection methodology

Some authors, for example (Gilmore, 1985) and (Hayes-Roth, 1983), have conducted reviews of available expert system tools, apparently without any attempt to develop a methodology for review or evaluation in a practical sense (or, at least, if such methodology has been developed, it is not made explicit).

Other authors define in varying amounts of detail their preferred method for evaluating tools.

(Harmon, 1988) suggests the following features and characteristics should be considered:

* Knowledge representation, inference, and control
* Developer interface
* User interface
* System interface
* Training and support
* Cost

A similar list is given by (Bielawski, 1988):

* Fit of the tool to the system
* Effectiveness of the developer interface
* Effectiveness and friendliness of the user interface
* Integration capability with existing software
* Run-time licensing for delivered systems

It can be seen that, despite the differences in terminology, there is a close correspondence between the above lists. (Waterman, 1985) takes a slightly different approach:

* Development constraints
* Support facilities
* Reliability
* Maintainability

* Problem features
* Application features

A more comprehensive description of strategies for selecting an appropriate expert system tool is that given by (Bowerman, 1988). In an extensive chapter, the following twelve criteria are used:

* Knowledge representation structures and inference mechanisms
* Fact uncertainty and unreliability of rules
* Logic and mathematics
* Effects of rule-sequencing and use of meta-rules
* User interfaces for knowledge-based display
* Graphic, database, and programming interfaces
* "HOW" and "WHY" explanations and "HELP" facilities
* Knowledge acquisition, maintenance, and learning
* Machine requirements, RAM, disks, MIPS, ALU, and peripherals
* Rule capacity, volatility, and response time
* Pricing and availability
* Vendor stability and growth

This list is not too different in substance from the lists given earlier (but is clearly more detailed).

It should be noted that the objective of the current work was not to review all of the tools mentioned in section 3.2.2, but to select the one most suited to the task in hand - as is usually the case with any expert system development. That is, the process is fundamentally not one of review, but of selection.

It was decided that the following criteria would provide a sound basis for selection. The list is largely based on those given above (though a certain reordering has taken place), while adding at the top of the list one aspect which should not be taken for granted:-

* Tool availability
* Machine requirements
* Rule capacity
* Cost
* Form of knowledge representation
* Inferencing mechanisms
* User interface
* Explanatory facilities
* Developer interface
* Ease of maintenance

* Vendor support facilities

The first three criteria are placed at the top of the list because any one of these may rule a tool out of consideration. The third, rule capacity, should not be taken as an assumption of the use of a rule-based system, but rather that certain size requirements will inevitably constrain the choice of tool.

Cost is the fourth criterion. Tools passing the first three criteria may still have to be rejected because of the cost of the run-time license. The cost of the development package, while still relevant, is less crucial, as it is likely to have less impact on the ultimate price to the user.

The remaining seven criteria should be regarded as a totality, rather than as separate requirements. A system with an excellent user interface may be preferred to one with an excellent developer interface, or vice versa, depending on the merits of each and the particular situation. Nevertheless, an attempt has been made to list the criteria in a rough 'order of importance'.

The actual selection process - that is, the application of the above criteria to the tools mentioned in section 3.2.2, with regard to the development of an expert system for the diagnosis of diseases in goats - is detailed in the following sections.

3.2.4 The selection process

The objective of the selection process was to select the single most appropriate tool for the development of an expert system for the diagnosis of diseases in fibre and dairy goats. Only if no single tool proved suitable would other options be considered - for example, the use of multiple tools.

The criteria to be used were those listed in the previous section.

* **Tool availability:** of the 39 tools under consideration, only two - ROSIE and MicroExpert - could not be firmly established to be available. Given this element of doubt, both were ruled out of consideration.

* **Machine requirements:** the first objective states, in part, that the system must be usable by veterinarians. It was considered unlikely that any veterinarian would purchase a new machine specifically to run this expert system; it was preferable that the system should be compatible with pre-existing hardware and software. Thus, a pilot survey was carried out to establish the preferred environment for the expert system. Partial results of the survey are given in Appendix 4.

Although the pilot survey was small (40), it was apparent that the most common equipment in use was of an IBM-XT type, running MS-DOS (or equivalent).

It was therefore decided that the tool selected should be capable of developing a system that would run in this environment. A distinction is necessary here between the development environment (as yet, unestablished) and the run-time environment; it was the latter which was the important consideration at this stage.

ART, Duck, EMYCIN, Envisage, ESE, ExTran 7, Goldworks, IKE, KAS, KEE, Knowledge Craft, KWB, Nexpert, Picon, Rulemaster, S1 and TIMM are not available in PC-XT / MS-DOS versions, and were eliminated at this point (Bowerman, 1988), (Harmon, 1988), (Liebowitz, 1988), (Waterman, 1985).

* **Rule capacity:** some estimate had to made about the likely size of the system, taking into account possible future growth. At this initial stage it was recognised that any estimate could only be very approximate.

It was decided that a minimum guaranteed capacity of 1000 rules (or equivalent) should be required. Additionally, systems should allow for at least 250 possible diagnoses.

Advisor provides for only 255 rules; Expert-Ease for 300 examples; PC-Easy for approximately 400 rules; TIMM-PC for '500+' rules (Bowerman, 1988). 1st-Class allows for only 32 'results', though combined with a sister product called Fusion this figure increases to 128. This was still inadequate. Super Expert, an induction-type tool, allows for only 8 results (Bielawski, 1988). All six tools mentioned were eliminated from consideration.

* **Cost:** anecdotal evidence returned with the pilot survey indicated a strong reluctance on the part of veterinarians to consider any expert system which would cost more than A$75. This could only be an indication, as the sample size was small, and attitudes might change if the system acquired a good reputation. However, as one of the primary objectives was to develop a usable and affordable system, it was decided that the system tool selected should not push any possible purchase price above this figure.

Ignoring the cost of the development license, run-time licenses normally come in two varieties: a one-time cost that pays for any number of copies, or a charge for every run-time copy.

None of the tools from vendors that employ the former type of license were ruled out at this point, as a large number of sales could minimise the effects of even a very expensive one-time license; however, where vendors offered only the latter type of license, a maximum cost of US$60 was used as a cut-off point.

Aion ($750), GURU ($400), Insight 2 ($95), M1 ($500) and PC-Plus ($95) were discarded (Bowerman, 1988).

At this point only 9 of the original 39 tools remained under consideration. These were AGE, ESP Advisor, ESP Frame-Engine, EXPERT, Expert-Edge, EXSYS, KES, SAGE, and VP Expert.

Each of these tools was examined in some detail, using as a focus the seven criteria listed: form(s) of knowledge representation, inferencing mechanisms, user interface, explanatory facilities, developer interface, ease of maintenance, and vendor support facilities.

A brief overview of each of the nine tools follows, together with a reference for further information.

* **AGE** only just made the final list, as many would consider it closer to a language than a true shell. Developed at Stanford, it is at its most fundamental a collection of INTERLISP functions; it is designed for use by experienced Lisp programmers.

The fact that AGE lacks useful facilities for I/O, data base interaction, and explanation, combined with a probable difficulty with convenient support facilities, suggested that AGE would not be an appropriate tool.

For further information, see (Hayes-Roth, 1983) and (Waterman, 1985).

* **ESP Advisor** and **ESP Frame-Engine** are both written in Prolog, and hence use backward-chaining as their primary inferencing method. Knowledge representation is in the form of IF-THEN type rules. The user interfaces are in the form of window-based text, and both include "HOW" and "WHY" type explanatory facilities.

A closer look at the inferencing mechanism of these two shells reveals that neither support reasoning under uncertainty - both could be described as non-probabilistic systems, or "decision-tree managers". As the nature of most diagnosis-type systems is precisely to deal with uncertainty (and the present case is no exception), both of these tools were discarded.

Information on both tools can be found in (Bowerman, 1988).

* **EXPERT**, like AGE, lies somewhere on the border between languages and shells. It assumes a rule-based representation, and includes a forward-chaining control mechanism, uncertainty handling, and efficient and

transportable code. User facilities are sophisticated and include explanation, knowledge acquisition, and consistency checking.

EXPERT has been widely used for medical applications, particularly in the fields of opthalmology and rheumatology.

EXPERT is considered in some detail in (Hayes-Roth, 1983) and (Waterman, 1985).

* **Expert-Edge** uses IF-THEN rules with backward, and limited forward, chaining. It can handle uncertainty, with Bayesian statistics being used to handle probabilities, and rules can include calculations and equations. Written in C, the user and developer interfaces make use of menus, windows, and colour. Basic explanatory facilities are included. There are also direct interfaces to DBase III and Lotus 1-2-3.

Expert-Edge is described by (Bowerman, 1988).

* **EXSYS** uses IF-THEN-ELSE rules with forward and backward chaining, and can handle reasoning under uncertainty. A full set of arithmetic operations is provided, as are explanatory facilities and context-sensitive help. Up to 5000 rules can be used in a PC environment, and developer and user interfaces are friendly.

English-like rules are input and updated internally, without the need for an external editor. They may include variables and equations, and external programs may be called at any point, with parameters being passed in the form of ASCII files.

Additional programs are available which expand the facilities of EXSYS, such as the ability to use frames (FRAME) or look-up tables (TABLET).

More details can be found in (Bowerman, 1988) and (Harmon, 1988).

* **KES** uses a structured IF-THEN rule language with backward chaining, and also employs frame-like structures and demons. Inferencing mechanisms include abductive reasoning, and the use of Bayesian statistics.

Uncertainty and rule unreliabilty are handled, and a large number of numerical functions are included. Written in C, there are interfaces to any C-callable language. Explanatory facilities include "WHY" and "JUSTIFY".

The knowledge base is created with any external ASCII text editor and then read in to the system. All variables used in the knowledge base must be declared, which has the advantage of protecting against mis-spellings (but can be frustrating if trying to build a quick prototype).

A hard disk and 640K RAM are required.

Further information is in (Bowerman, 1988) and (Gilmore, 1985).

* **SAGE** uses IF-THEN rules with backward chaining; it uses fuzzy logic and Bayesian inference, handles both uncertainty and rule unreliability, and includes all basic arithmetic functions. Written in Pascal, it has interfaces to Pascal, Fortran, and ADA. Some explanatory facilities are also included.

See (Keen, 1984) and (Bowerman, 1988).

* **VP Expert** is a rule-based system with backward chaining, and an inductive front end. A full implementation of the confidence factor schema is provided. Written in C, there are direct interfaces into DBase III and Lotus 1-2-3. Menus and colour are used to provide reasonably friendly user and developer interfaces, though the former in particular has been criticised (Bielawski, 1988) for the number of keys that have to be pressed in different situations. Trace facilities are provided in the form of limited "HOW" and "WHY" commands.

Rules are typed in directly, bearing in mind some fairly rigid syntactical constraints; there are some 67 key words, so entering rules is not unlike writing a program.

There are more details in (Bielawski, 1988) and (Harmon, 1988).

3.2.5 Final selection

Three of the nine tools remaining at the end of section 3.2.3 (AGE, ESP Advisor, and ESP Frame-Engine) were discarded for reasons outlined in the previous section. Further information was sought from suppliers of all of the remaining six (EXPERT, Expert-Edge, EXSYS, KES, SAGE, and VP Expert) before the final selection took place.

The combination of the selection methodology and the information available (from suppliers and the quoted sources) provided sufficient differences to be identified between the tools to enable a clear order of preference to be established.

EXPERT was placed last on the list, primarily because of its poor developer interface, and also because of doubts about its availability and its possible lack of vendor support.

KES was placed fifth - a major disadvantage here was its dependence on a hard disk. Cost was also a factor, as at $4000 it was considerably more expensive than others on the list. The use of an external text editor was also considered a slight disadvantage.

VP Expert was placed fourth, because of strong doubts about the development environment, and a more general concern about its lack of features. There was also considerable doubt about its capacity to handle the number of rules that might be required.

It seemed probable that all three remaining shells (Expert-Edge, EXSYS, and SAGE) could be used to develop a successful system. However, clear differences enabled the final ranking to be made with some confidence.

SAGE was placed third because of its (relatively) unfriendly user and developer interfaces, and its total reliance on backward chaining. Support facilities also appeared likely to be less comprehensive than would be the case with either of the two remaining tools.

Expert-Edge was a clear second - the forms of knowledge representation and the inferencing mechanisms appeared adequate and the user and developer interfaces were good. However, Expert-Edge was more expensive than EXSYS ($1495 as against $995) and appeared to offer less flexibility.

EXSYS was the tool selected for development because of its ability to perform both forward and backward chaining (and combinations thereof), its friendly user and developer interfaces, and its ability to handle variables, equations, and external program calls, all of which might be required. The internal editor was also considered an advantage, as was the range of optional add-on programs.

Vendor support appeared to be readily available (although located in California, the company offered assistance by 'phone, mail, or FAX). The company was well-established, with a large turnover, and the product has been successful in the market-place for a considerable period.

The final order of prefernce was therefore

1. EXSYS
2. Expert-Edge
3. SAGE
4. VP Expert
5. KES
6. EXPERT

with EXSYS being the tool selected for development.

3.3 Knowledge Acquisition

The word "bottleneck" in relation to knowledge acquisition is used in perhaps 70% of the sources on expert systems listed in the bibliography - for example (Waterman, 1985). This is because knowledge acquisition can be a long, meticulous, and often tedious, process.

Before discussing briefly current techniques for knowledge acquisition, the term "knowledge elicitation" should be mentioned. Many authors treat this as synonymous with knowledge acquisition, although some texts draw a distinction. Keravnou & Johnson (1986, p39), for example, consider elicitation takes place first, and that "...the primary purpose ... is to explore the expert's domain and gather data, out of which one gains an understanding of the domain knowledge structure and the dynamics of the operative strategies (i.e. how strategies interact and interplay during problem solving activities)." On the other hand, the primary purpose of knowledge acquisition is "...to instantiate (the) representation structures to ensure that the resulting computer model has sufficient grasp of the domain's factual and reasoning content to exhibit competent behaviour".

In the current work the term knowledge acquisition will be used to cover both activities, as there seems little point, except for semantic reasons, to differentiate between them.

3.3.1 Selection of technique(s)

(Hoffman, 1987) presents five methods that can be used for knowledge acquisition; while not directly relevant here, the article is interesting for two reasons.

The first is the distinction drawn between structured and unstructured interviews. In order to add structure to an otherwise unstructured interview, says Hoffman, the knowledge engineer "...initially makes a first pass ... by analysing the available texts ... or by conducting an unstructured interview. The expert then goes over the first pass ... making comments ... recording this process is not necessary because the knowledge engineer can write changes and notes on a copy of the printout of the first pass ... the result is a second pass..." (Hoffman, 1987, p56).

Secondly, Hoffman gives some interesting statistics about the development of some of the more well-known systems. MYCIN, apparently, took "many years"; INTERNIST "...took 10 years with the help of a full-time specialist in internal medicine". R1 "...took two man-years to develop by a team of about a dozen researchers and is still being refined".

PUFF, however, "...was reported to have been developed in less than 10 weeks. The likely reason for this brevity was that most of the rules were easily gleaned from archived data ... and only one week was spent interviewing the experts" (Hoffman, 1987, p62).

Given the above, it was decided that the knowledge acquisition process should be based on a few initial semi-structured interviews, after which most of the knowledge for the initial prototypes would be taken from an existing publication written by the domain expert, entitled "The Diagnosis of Diseases in Goats" (Baxendell, 1988).

While it was possible that some knowledge would be extracted incorrectly, or some terms misunderstood, this method, together with the use of a pre-existing shell, seemed likely to enable initial prototypes to be written very quickly; errors could be picked up before they became serious through the normal prototyping process, described in section 3.6.

Thus, it was decided that the knowledge acquisition process would consist of:

* one or two semi-structured interviews (to discuss methods, timescales, basic terms etc.)
* extensive use of a pre-existing text

and, once the initial prototype had been developed,

* many prototype feedback cycles, involving both structured dialogue and further references to the text.

3.3.2 Further reading

As it was clear from the outset that the knowledge acquisition process was likely to be a major bottleneck, much pre-reading was carried out on different techniques. A full discussion of these is beyond the scope of this thesis, but the interested reader is referred to the following sources:-

(Boose, 1988) gives an excellent and comprehensive introduction to the whole area of knowledge acquisition.

(Waterman, 1985) gives a general discussion of different techniques.

(Prerau, 1987) discusses in detail some 30 techniques that arose out of the COMPASS project.

(Hoffman, 1987) discusses knowledge extraction from the point of view of experimental psychology.

(Gammack & Young, 1985) gives a breakdown of different kinds of knowledge, and a brief description of protocol analysis, task analysis, multidimensional scaling, personal construct theory, and concept sorting.

(Cookson et al, 1985) discusses the use of a computer-based system to elicit knowledge.

(Weilings & Breuker, 1984) is an excellent work detailing the different levels of mapping between verbal data and knowledge structures.

(Boose, 1984) gives a very concise description of personal construct theory and the transfer of human expertise.

(Hart, 1986) has a thorough overview of the whole topic of knowledge acquisition, together with excellent chapters on fact-finding by interviews, fuzziness in reasoning, and the use of the repertory grid in personal construct theory.

3.4 Knowledge Representation

EXSYS uses IF-THEN-ELSE type rules as its primary means of knowledge representation; frame-type representation is also possible by the addition of an extra package called, appropriately, FRAME (EXSYS, 1985).

For the current system, this additional capability was thought to be neither necessary nor desirable (the former because small frame-type structures may be accomodated within IF-THEN-ELSE type rule structures, and the latter because it was thought preferable to use as few representations as possible, in order to simplify the finished system). However, the use of FRAME was not ruled out, because the development process itself might indicate its usefulness.

The knowledge base in EXSYS is constructed using CHOICES, QUALIFIERS, and RULES. The choices correspond to the possible final diagnoses, that is, the diseases which the system considers (for example, BACTERIAL PNEUMONIA).

The qualifiers correspond to the questions which may be asked of the user. For example, There is a history of chronic arthritis in the herd

1) yes

2) no

3) unknown

Such qualifiers are asked of the user only if, firstly, the answer cannot be discovered from already known facts, and, secondly, any possible answer will aid the diagnosis in this particular case.

The rules, which are of an IF-THEN-ELSE type, correspond to the expertise that the system uses to arrive at a diagnosis. For example,

IF tests or analyses so far carried out
include NOT soil analysis
THEN soil analysis indicates cobalt levels
are unknown
and soil analysis indicates copper levels
are unknown
and soil analysis indicates molybdenum levels
are unknown
and soil analysis indicates sulphur levels
are unknown

This rule prevents the user from being asked about the levels of certain soil components if a soil analysis has not been carried out. Note the English-like (but NOT English) form of the rule.

Choices can also occur in the IF, THEN, or ELSE parts of rules, in which case they are used together with "certainty factors". For example,

IF the antigenic pillus K99 (or K88 + 987 antigen) is not present
THEN [53] Colibacillosis - Probability=0/10

This rule effectively states that Colibacillosis may be ruled out if a certain pillus is not found.

For more details on the knowledge representation system used by EXSYS, see (EXSYS, 1985).

3.5 Inferencing

At the heart of any expert system is the inference engine - that part of the system that applies a logical reasoning mechanism to the knowledge contained in the knowledge base.

A discussion of the logical reasoning mechanisms required in a diagnosis-type system is to be found in section 3.5.1.

Meta-level reasoning is "reasoning about reasoning" - for example, strategic decision-making about the order in which the rules contained in the knowledge base will be applied. The facilities which EXSYS provides for meta-level reasoning are discussed in section 3.5.2.

Reasoning under uncertainty describes the process of making inferences which are uncertain; how EXSYS builds this uncertainty in to the system, and the rules it uses for propagating probabilities are both discussed in section 3.5.3.

3.5.1 Logical reasoning mechanisms

In an excellent comparison of diagnostic paradigms, (Poole, 1988) points out that there are three predominant approaches to the problem of diagnosis (for the purposes of the current work some of the terminology has been changed and some of the descriptions simplified):-

* minimising, in a manner consistent with all knowledge and observations, assumptions of abnormal components
* abductive diagnosis, that is finding a set of causes which can imply the observations
* rule-based diagnosis, where there are a set of symptom-cause rules, that is determining what abnormalities can be predicted based on the evidence

The distinction between these three is of some importance, and each of the cases will be looked at briefly in relation to the present development, which is concerned with the diagnosis of diseases, that is, the discovery of the causes of certain observations, or symptoms.

In the first case, the sort of knowledge required is of the form symptom => cause. Thus if C_1, C_2, ... , C_n are possible causes to explain symptom S_i, we have as a fact or a default

$$S_i \Rightarrow C_1 \text{ or } C_2 \text{ or } \ldots \text{ or } C_n$$

That is, if we have a symptom S, we presume there must be a cause - it is inconsistent that we do not have any of C_1 through C_n being true.

In the second case, the sort of knowledge required is that from some cause we can prove the observations (symptoms). This sort of knowledge is of the form cause => symptoms. The base faults become the possible hypotheses. That is,

$$C_j \Rightarrow S_1 \,\&\, S_2 \,\&\, \ldots \,\&\, S_m$$

This corresponds closely to "real-world" knowledge; that is, a disease is known to cause certain symptoms.

In the third case, we have to explain a fault (symptom). The sort of knowledge required is of the form observation (symptom) => abnormality. Thus, if S_i represents an observed symptom, then we regard this as evidence for C_j. That is,

$$S_i \Rightarrow C_j$$

This case, while similar in nature to the first, is found to be unsound, and may lose the structure of the problem.

Both of the first two cases are found to be sound. The first is found to be closer to a real (human) diagnosis session, while the second is more modular. This modularity greatly increases the ease with which new knowledge may be incorporated into the system.

Despite this, it was decided to base the current development on the first, because of its correspondence with the knowledge representation structures used by EXSYS, and its similarity to the diagnosis procedure used in a typical veterinary consultation.

For further details and examples of the three paradigms, the reader is referred to the original source (Poole, 1988).

3.5.2 Meta-level reasoning

EXSYS allows for both backward and forward chaining, and variants thereof. In total, four different basic options are allowed (EXSYS, 1985).

One commonly quoted rule-of-thumb for deciding between forward and backward chaining runs as follows: "...if a problem involves significantly more conclusions than facts, forward-chaining is preferred; if the problem involves significantly more facts than conclusions, backward-chaining is probably more appropriate" (Bielawski, 1988, p35).

This would indicate an initial preference for forward chaining in the current circumstances (where the number of possible diagnoses would exceed 200). Further, backward chaining has the disadvantage that questions are frequently asked in what appears to the user to be an illogical order; as Cox (1984, p125) points out, in describing the building of MicroExpert,

"...the human users can find it rather disconcerting if the system jumps about from subject to subject in what appears to be a random manner ... it is often necessary to be able to control this".

Nevertheless, backward chaining has been employed successfully in many diagnosis-type systems, such as MYCIN (Buchanan & Shortliffe, 1985). The main advantage here is that backward chaining allows for only those qualifiers which are relevant to a particular choice to be asked in a normal consultation - questions which cannot lead to a valid diagnosis are bypassed.

For example, if disease X has been ruled out, then no further questions which relate to disease X alone are asked. In a forward chaining system this will often not be the case; and thus, the number of irrelevant questions asked is increased.

Some of the detrimental effects of backward chaining may be minimised by using additional features provided by EXSYS to guide the questioning process. Given this, and as one of the aims of the system was to provide reasonably short and concise consultation sessions, it was decided that the system should use backward chaining.

3.5.3 Reasoning under uncertainty

In most diagnosis-type systems, uncertainty plays a central role; it was considered essential, therefore, that the system should cope with uncertainty in a meaningful way - meaningful both internally (so that two or more "probabilities" could be combined) and externally (so that the "probabilities" made sense to the user).

EXSYS provides five different systems for allocating and combining what the manual calls "probabilities". More importantly, EXSYS allows for variables, formulae, and external program calls to be included in rules, thus allowing for the possibility of the developer including his/her own modifications to the basic system.

Despite this ability, EXSYS insists on one of the five basic systems being selected at the outset; this is a particularly important decision, as it cannot be altered at a later stage.

The first system offered is based on a simple 0 (false) or 1 (true) value, and is designed to cater for those systems which do not require uncertainty. It was therefore considered unsuitable for the present development.

The second system is based on a 0 to 10 scale, where 0 and 10 (representing absolutely false and absolutely true respectively) lock the final value - once a disease is given a value of 0, for example, it is not considered again. All other values (1 through 9) represent degrees of confidence ranging from very probably false to very probably true. These values are averaged over all of the rules that contain the choice to arrive at a final value for that choice (so, 5, 6, and 1 would be averaged to give 4; 10, 3, and 2 would still give 10).

The third system is based on a -100 (almost certainly false) to +100 (almost certainly true) scale. There are no "locking" values as previously, and all values are averaged to arrive at a final value.

The fourth system is based on a 0 to 100 scale, and values are combined as if they are dependent probabilities.

The fifth system is similar to the fourth, but here values are combined as if they are independent probabilities.

These last two systems, while initially appearing quite attractive (they are the only two to use actual probability theory), were both rejected; since in any diagnosis-type system, values obtained are neither wholly dependent nor wholly independent.

For example: in the fourth system, suppose we have two rules, both of which indicate a degree of confidence in disease X of 90/100. Combining these results in a figure of 90% * 90% equals 81/100, a lower figure than if we had the evidence of just one rule.

The fifth system results in this case in a more "acceptable" figure of 1 - (1-90%) * (1-90%) equals 99%. However, now suppose that the first rule indicated a likelihood of 90%, while the second indicated that this solution was quite unlikely (say 20%). Then combining these two results in a final value for the disease of 92% !

If EXSYS was to be used, therefore, the choice rested between the second (0 to 10) and third (-100 to +100) systems. The final choice was based on two factors.

Firstly, and most importantly, it was thought desirable that the system should have the ability to clearly rule out as many diseases as possible at an early stage. The capability of doing this by the simple allocation of a value of zero was thought to be beneficial.

Secondly, it was thought that an 11 point scale was quite sufficient to cater for the human expertise in this area. The added precision of a -100 to +100 scale was thought to be not only unnecessary, but in fact positively intimidatory to the domain expert, who could not reasonably be expected to justify a value of, say, +56 rather than +57.

Thus, it was decided that the system based on a 0 to 10 scale, where 0 and 10 are locking values and all other values are simply averaged, should be used.

Reasoning under uncertainty is currently a topic of considerable research interest, and the reader interested in the theoretical foundations of the topic (only the very elementary facilities provided by EXSYS have been discussed here) has a wealth of sources from which to draw. The following list is, therefore, only a small sample:

(Weiss & Kulikowski, 1984) give a very good introduction to Bayes theory, hypothesis testing, decision theory, and approximiate statistical methods.

(Buchanan & Shortliffe, 1984) is excellent for the uncertainty reasoning used in MYCIN, and also for the Dempster-Shafer theory of evidence.

(Forsyth, 1984) gives a good overview of fuzzy reasoning systems.

(Naylor, 1984) covers Bayes theory and the inferencing engine in general in a fairly elementary manner.

(Kowalik & Kitzmiller, 1988) is an excellent collection of research and development articles centred around the topic of numerical computing in expert systems.

(Lemmer & Kanal, 1988) is a collection of expanded versions of papers originally presented at the Workshop on Uncertainty in Artificial Intelligence which was held on August 8-10 in 1986 at the University of Pennsylvania in Philadelphia. A wide range of sub-topics is considered, with the book divided into four sections: analysis, tools, theory, and applications.

3.6 Prototyping

In standard system development, it is normal for all elicitation (analysis) to take place before the system is coded and implemented. In building expert systems, this is rarely the case, and many expert system developers use a prototyping approach almost as a matter of course (Black,1986).

Advantages of the prototyping approach are mentioned in many sources. (Bielawski, 1988) gives the following four advantages:-

1. A prototype enables the developer to judge whether the system is feasible.
2. A prototype enables the developer to test the suitability of the development tool that has been selected.
3. A prototype will suggest the amount of time required to build the whole system.
4. A prototype often makes a convincing argument for gaining support (for the project).

Another point should also be mentioned. That is, a prototype greatly enhances the effectiveness and efficiency of any interaction between the knowledge engineer and the domain expert. Progress can be seen and appreciated from the very earliest stages of the project right through to its final implementation.

(Black, 1986) summarises the feedback cycle in the prototype refinement approach as follows:-

1. Select new set of test cases.
2. Elicit new rules or rule refinements from the expert to account for new problem cases.
3. Encode the new rules (into the system).
4. Test the new knowledge base against the current problem set with the expert as critic.
5. Repair the rules until they work for the current problem set.

At the end of each of these cycles, the knowledge base should be in a consistent state where it has been "black box tested" against a set of problem cases. It can never be said to be complete. New test cases may not be handled correctly without modifying or adding to the knowledge base.

In order for the system to be useful, the proportion of cases handled correctly must be very much higher than those handled in error. It is thus necessary to continually monitor the performance of the system to ensure that it meets acceptable limits.

(Buchanan & Shortliffe, 1984) describe the process of building an expert system prototype as cyclic interaction between the expert, the knowledge engineer, and the system itself, in a manner similar to, but varying from, that given above:-

1. The expert tells the knowledge engineer what rules to add or modify.
2. The knowledge engineer makes changes to the knowledge base.
3. The knowledge engineer runs one or more old test cases for consistency checking.
4. If there are any problems with the old cases, the knowledge engineer discusses them with the expert, and then goes back to step 1.
5. The expert runs the modified system on new cases until new problems are discovered.
6. If no problems are found in a substantial number of cases, then the expert stops changing the system. Otherwise, the expert goes back to step 1.

As (Boose, 1986, p15) points out in reviewing the above, "...in practice, the procedure may be slightly different". In particular, in the current development, it was not intended that the domain expert "would tell the knowledge engineer what rules to add or modify" (step 1). This would require a more detailed understanding of the structure of the knowledge base by the expert than was likely to exist. Also, it was possible that the "simple" addition of a new rule would have severe repercussions for the inferencing process. Thus all changes would be made by the knowledge engineer (developer) in an effort to ensure the integrity of the system as far as possible.

As the prototyping approach is well established for building expert systems, nothing more need be said here, except that it was expected that the availability of written material on the domain of expertise would greatly speed up the prototyping process, particularly in relation to those parts of the cycle where new problem cases were being sought, and new rules entered.

3.7 Testing and Validation

One advantage of a prototyping approach is that testing occurs throughout the whole period of development, rather than being concentrated near the end, when most of the work has been completed. This enables major faults to be rectified as soon as they arise, and the results of such faults are therefore less likely to have potentially disastrous consequences.

The testing and validation process for this particular development consisted of three phases:-

1. Continuous testing by the domain expert throughout the prototyping process.
2. Testing by the domain expert and selected others at the completion of the prototyping phase; that is, when the knowledge base and inferencing mechanisms are thought to demonstrate reliability and integrity, though not necessarily completeness. This phase includes difficult and extreme cases.
3. Testing and validation by selected release; the system, complete with manuals, help facilities, etc is tested and validated "in the field" by its application to real cases.

At the time of writing, phase 3 has not been undertaken. The system will not be considered suitable for general release until phase 3 has been completed, and satisfactory performance has been indicated by those using the system. It is expected that such a process would last for a period of approximately 12 months.

The criteria used in phases 1 and 2, and which will be used in phase 3, are those given by (Bielawski, 1988):-

1. Accuracy
2. Completeness
3. Reliability and consistency
4. Effective reasoning
5. User-friendliness
6. Run-time efficiency

The reader interested in testing and validation (and also evaluation) is referred to the following sources:-

(Liebowitz, 1988) gives an alternative list of criteria to those given above.

(Hayes-Roth, 1983) provides an excellent and detailed chapter on expert system evaluation.

(Stevens, 1984) gives an excellent argument on the difficulties of judging expert systems; to take one sentence, "...on the one hand we have the opinions of experts and on the other we have the conclusions of a computer system which we shall judge by using the opinions of experts..." (Stevens, 1984, p45).

(Liebowitz, 1985) details a fairly comprehensive survey of different approaches to expert system evaluation.

3.8 Documentation

The user interface provided by EXSYS is such that, during prototyping, the domain expert was able to run the system without any external documentation. The system itself provided all of the help facilities that were required during running.

Development of the system was made feasible by the extensive listings and cross-references provided by EXSYS. Printed listings were required after each major update, and much use was made of the cross-references between the choices, qualifiers, and rules which formed the knowledge-base.

Although EXSYS provides such listings on demand, they quickly became so lengthy it was considered necessary to write programs to condense these (by such means as removing all blank lines). The listings given in the appendices have all been condensed in this way.

4 Results

As with any development of this kind, theoretical models employed at the outset of a project can provide a sound foundation upon which to build; there are always unforeseen difficulties and problems encountered along the way, however, which if not handled correctly, can lead to an inefficient system, or, at worst, to the abandonment of the whole project.

Some of these problems, and their solutions as implemented in the current system, are detailed in section 4.1.

The current state of the expert system, as at the time of writing (January 1990), is outlined in section 4.2.

Results so far achieved are discussed in section 4.3.

Finally there is a brief discussion of the work still to be done in section 4.4

4.1 Problems and solutions

This section deals with the difficulties that were experienced during development, and the efforts that were made either to overcome them, or to minimise their harmful effects.

No attempt has been made to categorise these difficulties under topic headings, as many cut across several areas. Instead they are dealt with chronologically as they arose; it is to be hoped that this will not only give a flavour of the actual development process, but will also prove beneficial for other developers who may be following similar lines.

Problem: the number of diseases to be covered was very large. How should the prototyping process commence ?

Solution: clearly some decision had to be made about which diseases to include in early prototypes. It was not feasible to build an initial system based on in excess of 200 diseases.

One possibility was to select a certain class of diseases (for example, those of the eye) upon which to base initial work. This was rejected, as it was possible that an unfortunate choice might prove atypical. It is important in this type of prototypical development that eary work is as typical as possible of that which is to follow.

Rather than choose a group of diseases at random, it was decided to ask the domain expert to give all diseases a "likelihood ranking" from 0 to 9, where 0 was assigned to those diseases never yet known in Australia, and 9 to those which are most prevalent. Work was then commenced using those diseases which were considered most common.

The two advantages of this approach, which proved very successful, were firstly that initial prototypes could be based on a small number of diseases, thus minimising the amount of work involved before real results were obtained; and secondly, that should the system eventually prove unable to handle the whole gamut of diseases, it could still be a useful product, as it would only exclude those diseases considered rare or exotic.

Problem: when should a particular disease be no longer considered by the system ?

Solution: this is a very basic, and major, problem. To illustrate: a patient visiting a doctor because of a sprained ankle may have, in addition, a serious heart problem. Should the doctor check for this ? Or should the doctor treat the perceived symptom(s), in this case a painful ankle, and send the patient on his way ?

Clearly any goat may have any disease, or combination of diseases, at any time. However, if the system checked out every possible disease on every occasion, consultation times would be excessive. Further, the veterinarian would normally be looking only for the cause of some specific problem or problems.

Therefore it was decided to discard all diseases which did not show at least one symptom. That is, if the only reported symptoms are convulsions and excitability, then only those diseases which could cause one or both of these symptoms are considered by the expert system.

Problem: what mechanism can be used to discard diseases ?

Solution: EXSYS provides no easy way to rule out choices. The obvious way, of giving a disease a "probability" of 0/10, will not work simply because a later question about symptoms might bring that disease back in to consideration.

The only feasible solution to this problem was to write an external program to take all symptoms, and return to EXSYS, via an external file, a list of all those diseases which could now safely be assigned a zero probability.

This short program is currently some 30 lines in length, and is written in Pascal. It also reports to the user the number of diseases being considered by the expert system. Its operation is automatic, and is transparent to the user of the system.

Problem: which diseases should be reported by the system ?

Solution: the problem here is fairly subtle; that is, it is not usually possible to completely rule out any cause of a particular symptom. It is, however, possible to say that a particular cause is very unlikely.

Thus, initial prototypes tended to list the most probable cause(s) at the top of the list, followed by a large number of diseases to which it had assigned a "probability" value of, say, 1/10.

Here it is important to remember that the system was never intended to produce one, correct, diagnosis, but to act as an aid to veterinarians. It was therefore decided to report all possible diseases (that is, all diseases with the exception of those assigned a zero probability), but to stress to the user that those diseases listed with a 2/10 or 1/10 probability were very unlikely.

Problem: how to combine factors pointing to a disease ?

Solution: this problem goes to the heart of reasoning under uncertainty. In the EXSYS scheme chosen, certainty factors (CFs) - what EXSYS calls "probabilities" - are simply averaged. Suppose we have two independent factors - factor A yields a CF of 7 for disease X, while factor B yields a CF of 9 for the same disease. Averaging, we get a final CF of 8 - that is, less than if we had factor B alone !

One possible solution to this is to have a multitude of external programs to compute more reasonable CFs. However, the number of programs (or at least sub-programs) required would be immense, as some rules are more independent than others. The degree of independence would be a major factor in computing a sensible CF.

The actual solution used is far from ideal in a theoretical sense, but is adequate in practice. In many cases, the simple averaging proved adequate. In others, particularly those where there were several factors combining, new rules were introduced for different combinations of factors. For example, for two conditions A and B, up to four new rules of the form IF

A & B, IF A & NOT B, IF NOT A & B, and IF NOT A & NOT B were introduced. The difficulties here are clear: with three factors eight rules are required; with four, sixteen; and so on. In practice it was found that only certain combinations were needed, however, and others could be left to rely on simple averaging.

While, perhaps surprisingly, the above solution seems adequate in the current development, and does not expand the size of the rule base unduly (because such combined rules are required only rarely), it cannot be guaranteed that this approach will be feasible in other developments.

However, it should be pointed out that there can be no simple solution to this problem if the system is rule-based. There can be no "correct" formula for deducing (A & B) given both A and B, as this will be different in all cases concerned with real-world problems.

This is an area where further research is required.

Problem: how to prevent unintended side-effects during maintenance of the rule base ?

Solution: during prototyping, the rule base was constantly being updated.

When a diagnosis proved incorrect, one or more rules were added, updated, or removed. This proved troublesome in that the effect of such a change would often adversely (and inadvertantly) affect other possible diagnoses.

It was decided at an early stage, therefore, that in most cases separate rules were required for each disease. Thus, a rule such as IF A THEN X & Y would become two rules, IF A THEN X and IF A THEN Y.

Although this does not prevent inadvertant effects, it has the effect of reducing them to manageable proportions. Also, the increased size of the rule base is offset by the added simplicity introduced.

Problem: how to cope with the unexpected unavailability of the expert ?

Solution: at very short notice, the domain expert was required for personal reasons to leave Western Australia at an early stage in development.

In many similar projects, this could have had disastrous consequences, such as the abandonment of the whole project. However, an earlier decision to base much of the knowledge in the expert system on a pre-written text ensured that development could proceed at almost the same pace. Contact was kept with the domain expert via letter, telephone, and FAX, and these proved more than adequate.

In the event, the domain expert returned to Western Australia (equally unexpectedly) after a gap of some four months, and normal progress resumed.

Problem: how to mitigate the effects of backward chaining ?

Solution: Unlimited backward chaining can cause questions to be asked in what appears to the user to be a nonsensical order; this is dangerous in that it causes the user to lack confidence in the system. For example, the system may first try to prove conjunctivitis (a disease of the eye) followed by pneumonia (a disease of the lung) when it is painfully obvious to a human expert that the goat has a broken leg.

The solution here was to force EXSYS to ask certain basic questions (such as the type and age of the goats, and the symptoms observed) before any real diagnosis was begun. This is made possible in EXSYS by the construction of a short configuration file which is checked before each session.

Problem: how to prevent the user entering absurdities ?

Solution: for example, it is quite possible for the user to say (either accidentally or intentionally) that the goat in question is a buck, and it has just had an abortion.

This type of absurdity is checked by the system, and, if such is found, an appropriate help screen is sent automatically to the user, and the consultation session stopped. These help screens are stored individually as external ASCII files.

Problem: the user is given poor explanatory facilities.

Solution: in fact, the explanatory facilities provided by EXSYS are at least as good as, if not better than, those provided by many other expert system shells.

The user may ask for explanations in many situations, and EXSYS aids this by allowing the developer to enter notes and references along with each rule. However, the problem arises in those rules where the developer has used variables, formulae, or, worst of all, external program calls.

Notes can be provided for the user to explain these, but the user is still presented with what may appear to be an unintelligible rule. This is unfortunate, and is a problem that seems insoluble to the developer.

It is to be hoped that the EXSYS expert system shell will be improved in future versions to allow certain rules to be hidden from the user, should the developer of the system so desire.

4.2 The current system

Early in January 1990 the development of the system had progressed to prototype version 7.0.

This version contained information on 221 diseases; had 474 rules; and included 281 questions that might be asked.

The system consists of 21 files, which fit on to a single 360K floppy disk. Of these, 8 are files used by EXSYS itself; 4 are files that contain the domain knowledge and meta-knowledge; and the remainder are mainly help files containing information on the domain (for example, one file contains a list of potentially poisonous plants) that are displayed to the user as and when necessary.

A standard consultation lasts several minutes. The more symptoms that are entered, the longer will be the consultation. Users are warned at the outset to enter only those symptoms that they are sure of, in order to minimise the length of the consultation.

The final diagnosis consists of all those diseases that have not been eliminated, together with their "probability". The final diagnosis may be viewed on the screen, or reported to a disk file or printer, as required.

4.3 Test results

The most recent set of tests was performed by students at the Muresk Institute of Agriculture, on version 6 of the prototype system. These students have little or no computer knowledge, and possess no knowledge as to the internals of the expert system.

Twenty-six tests were carried out. Consultation sessions varied in length according to the symptoms reported, with the minimum being 28 questions asked (to identify 'Pink Eye'), and the maximum 77 (to identify Enterotoxaemia). In the latter case, five different symptoms were reported to the system.

The lengths of time taken for these consultations was not recorded, but it seems likely that in practice times will vary between three and ten minutes. The majority of questions asked are of a simple "yes or no" type, and little time for thought or contemplation is required. Delays caused by the system itself are minimal - the longest delay is of a few seconds immediately before the final diagnosis is listed.

All trials consisted of students assuming a hypothetical situation (as to location, conditions, feed etc) in which the goat(s) in question were suffering from one particular disease. In several of these cases, the disease was one which would be considered either rare or exotic to Australia.

Of the 26 cases, the system diagnosed the disease in question as the number one probability on 9 occasions; and within the top three probabilities in 16 out of the 26 trials.

(On several occasions the domain expert was of the opinion that, given the information available, the diagnosis provided by the system was "more likely" than the disease intended by the students.)

On 4 occasions the expert system failed to list the disease expected. Of these, one was caused by incorrect information in the knowledge base; one occurred because the disease had been omitted completely from the system; and two were caused by incorrect responses entered by the users.

The first two cases may be considered "bugs", and have been corrected in the current version. The latter two, caused by student errors, were largely due to the theoretical nature of the testing - students were responding to questions based on a hypothetical, rather than real, situation.

However, it is of course probable that incorrect answers will be input inadvertantly in real-life situations. The only protection against this is the "Change and Rerun" option provided by EXSYS, which enables the user to change any of the responses without necessarily repeating the entire session.

4.4 Further development and maintenance

Of the testing and validation phases listed in section 3.7, phase 1 has been completed; phase 2 (testing by the domain expert and selected others) is currently progressing; and phase 3 (testing and validation by selected release) has not yet commenced.

Much work still remains to be completed before the expert system is considered for commercial release. Amongst the steps remaining are:

* further refinement of the system to ensure a greater reliability of the final diagnosis
* further testing to be conducted by the domain expert and selected others
* testing and validation by selected release
* protection of the system against unauthorised copying
* writing of user and technical documentation
* packaging, advertising, and marketing of the system

Further, discussions need to take place, and decisions reached, as to the following points:

* the ownership of the expert system
* the division of royalties resulting from commercial sales
* the legal responsibilities in the event of incorrect or inaccurate diagnoses
* the responsibilities for future maintenance

It is envisaged that to develop the system to a stage where it could be marketed commercially would involve perhaps another twelve months, using the resources currently involved. In addition, work would need to proceed on all of the points mentioned above.

The system is now at a stage where further development should not encounter any theoretical difficulties. All diseases likely to be included are already in the system; the number of rules and questions are unlikely to expand by a factor exceeding 20%. Even a small expansion may cause the system to become sufficiently large to require the use of more than one 360K diskette, but this is not seen as a major problem.

Maintenance would best be handled by the current developer, though could, if necessary, be handed over to a third party. As with many expert systems, maintenance would be non-trivial, and all changes would require rigorous checking in consultation with the domain expert.

5 Conclusions

This section looks with the benefit of hindsight at the methods and methodologies employed, and the results achieved so far, and attempts an analysis of these with respect to the original objectives of the development.

Section 5.1 concentrates on the first objective, that of the successful development of a practical system, of which the research leading to this thesis forms a major part.

Section 5.2 looks outside of the current development, at lessons that can be learned which might assist future developments in similar areas. This section therefore relates to the second objective.

5.1 Expert system appraisal

The first objective stated in part that the system should be usable, useful, and affordable to veterinarians throughout Australia. There is little doubt that the current system, although still in prototype form, meets all of these criteria, with the possible exception of affordability, which is yet to be firmly established.

The development, though not without problems, has been successful for many reasons. Principal amongst these has been the time and care taken at the outset of the project.

A clear delineation of the domain boundaries ensured that the project did not expand beyond original intentions. Many conventional developments suffer from problems caused by increased expectations on the part of the users as development proceeds - this was not the case here.

The selection of a single domain expert proved highly beneficial. Multiple experts, even if in total agreement, would have created additonal overheads in terms of both time and resources.

The use of prototyping as the development methodology enabled the system to prove its potential at an early stage. The domain expert was able to use and appreciate the system when the rule base was still very small. Potential problems were sighted and cured before they caused any difficulties.

The decision to use an expert system shell cut development time dramatically. The facilities built in to the system would have taken many man-months to replicate from scratch.

The methodology behind the choice of an appropriate tool proved useful to the extent that little time was wasted in examining many tools that, in any event, could not be used. Without such a methodology there would always be temptations to use a tool that was familiar, or readily available, without the necessary exercise of looking in detail at the tool's suitability for the project in hand.

It is difficult to compare the system built using EXSYS with the system that might have been built if another shell had been selected. The strengths that EXSYS gave to the development of the system include:

* a friendly user interface, which enabled the system to be used with minimal prior instruction
* a friendly developer interface, which enabled rapid prototyping
* the ability to link to external programs, and display external files
* the use of an in-built editor, which enabled changes to the knowledge base to be made easily and quickly
* the inclusion of a change-and-rerun facility, which enabled multiple tests to be conducted efficiently
* the facilities offered for controlling the inferencing mechanism and the order of questions

* the ability to print out the knowledge base on request

The weaknesses displayed by EXSYS included the following:

* the poor handling of reasoning under uncertainty (and the misleading use of the word "probability")
* the lack of a sophisticated explanatory facility
* the inability of the developer to control which rules may be displayed to the user
* the inability of the user to correct input errors until the end of each run
* a difficult-to-use and poorly-structured manual

The use of a pre-existing text proved essential; without this, development would have been slowed enormously. The problems encountered because of incorrect assumptions or mis-understanding of terms proved minimal, and the prototyping process ensured that all such problems were picked up early and corrected.

The IF-THEN-ELSE rule-base structure, upon which EXSYS is based, proved perfectly suitable. In fact, the ELSE structure presented certain difficulties, and 98% of rules in the system are based on a simple IF-THEN structure.

Results of tests conducted on the latest prototype (version 7) are not available at the time of writing, but results on the previous version are highly encouraging (see section 4.2).

There is no doubt that the system is easy to understand and to use. Further, the system will run successfully on any IBM-XT or compatible machine which is using DOS 2.0 or higher. A monochrome screen is acceptable, though as use is made of colour a CGA screen is desirable. A hard disk is not required.

In terms of the first objective, therefore, the development to date has been successful.

However, further work remains, as outlined in section 4.4, and it is not expected that the system will be released commercially in the immediate future.

5.2 Lessons for other development projects

It is to be hoped that the current development will enable other expert systems over similar domains to be developed more efficiently (see objective 2, p5).

Many texts provide useful advice to those about to embark upon the development of an expert system. Amongst many that may be particularly recommended are (Keller, 1987), (Bowerman & Glover, 1988), (Bielawski & Lewand, 1988), and (Weiss & Kulikowski, 1984).

A review of the current project indicates the following:-

1. Establish the boundaries of the domain as early as possible. Changes to the boundaries should be made only if it is clear that substantial advantages would be obtained. Be aware that such changes may have dramatic consequences for the content and form of the knowledge base.

2. Decide early on the number of domain experts. If availability is a problem, it may be wise to choose two or more experts so that development is not held up; however, one expert is easier to work with than several, and overheads are lower.

3. Ensure that the domain expert is fully aware of the time commitment that will be involved. It should be clearly established that a certain number of hours per week will need to be set aside, for a certain length of time;

usually, at least twelve months.

4. The use of one or more pre-existing texts can speed up development time enormously. If the domain expert is the author, so much the better; in any case the text(s) used should be recommended by the domain expert.

5. At the beginning of the project, allow sufficient time to understand, and preferably become proficient in, most of the terminology used. If this is not done, there will be delays (at best) or incorrect knowledge entered into the system (at worst).

6. If time or resources are limited, use an expert system tool to aid development unless there are sound reasons for doing otherwise. Writing a system from scratch will mean that the development is a far larger undertaking.

7. Use a methodology such as the one used in the current development to select an appropriate expert system tool. Rule out as many tools as possible before looking in detail at particular tools.

8. Decisions should be made in advance about the relative importance of different system tool attributes to the current development. Avoid the temptation to use a tool merely because it is familiar, or has been used

before.

9. Do not underestimate the importance of the tool being able to interface with external programs. This ability can be most important in cases where the tool does not include some function or facility that is found to be necessary. Do not expect to find a tool that will meet all of your requirements.

10. Be aware that in most cases the cost of the run-time licence will be more important than the cost of the development licence.

11. Use prototyping. Do not attempt to cover the whole domain, but choose a representative subset on which to base early prototypes. Use these early prototypes to test all facets of the system. Solutions should be found to theoretical problems before the knowledge base is expanded further.

12. Decisions should be made as to the importance of reasoning under uncertainty within the system. Systems which deal only with certain knowledge present far fewer difficulties during development. When dealing with uncertainty, decisions must be taken about the degree of precision which is appropriate; whether rules are wholly dependent, partly dependent, or wholly independent (the middle case is the difficult one); and

about upper cut-off points (above which some choice is considered certain) and lower cut-off points (below which some choice is no longer considered).

13. The developer should document problems and solutions as they arise. It is important also to keep records as to the state of the knowledge base at various stages. This helps normal development, and aids in debugging when unexpected problems arise.

14. A system of testing and trialing should be planned in advance. Tests should be carried out independently of the developer, if possible, to ensure the integrity of the results.

15. EXSYS is an expert system shell that has many features to recommend it in the development of diagnosis-type systems. Other shells may be better.

APPENDICES

A.1 Glossary

Many of the definitions in this glossary are taken from (Waterman, 1986).

Backward chaining: an inference method where the expert system starts with one possible conclusion and tries to establish the facts needed to prove this conclusion. If this proves impossible, the system tries another possible conclusion, and so on.

Domain expert: a person who is proficient at problem-solving in a particular domain, and who acts as the source of the expertise which is built in to the expert system.

Explanatory facility: that part of an expert system that explains how conclusions were reached, and justifies the steps used to reach them.

Forward chaining: an inference method where the IF-portion of rules are matched against known facts to establish new facts; the expert system works forward from existing knowledge towards possible conclusions.

Frame: a method of knowledge representation that associates features with nodes representing concepts or objects. These features are described in terms of attributes (called slots) and their values. These nodes are organised in a hierarchical network.

Fuzzy logic: an approach to reasoning in which the rules of inference are approximate, rather than exact, in order to better manipulate information that is incomplete, imprecise, or unreliable.

Inference engine: that part of an expert system that contains the general problem-solving knowledge used to process the domain knowledge.

Knowledge acquisition: the process of extracting, structuring, and organising knowledge from some source (so that, for example, it may be processed by a computer program).

Knowledge base: the portion of an expert system that contains the domain knowledge.

Knowledge engineer: the designer and builder of an expert system, often a computer scientist.

Knowledge engineering: the process of building an expert system.

Knowledge representation: the process of structuring knowledge so that it may be processed efficiently.

Meta-level knowledge: the knowledge in an expert system about how the system is to operate or reason. More generally, knowledge about knowledge.

Rule-based system: an expert system where the knowledge is organised in the form of a set of rules, such as IF premise THEN conclusion.

A.2 Selected tool suppliers

EXPERT	Computer Science Dept., Rutgers University, Busch Campus, New Brunswick, New Jersey 08903, U.S.A.
EXSYS	California Intelligence, 912 Powell Street #8, San Francisco, California 94108, U.S.A.
Expert Edge	Jeffrey Perrone & Associates, 3685 17th Street, San Francisco, California 94114, U.S.A.
KES	Software Architecture and Engineering, 1600 Wilson Boulevard, Suite 500, Arlington, Virginia 22209, U.S.A.
SAGE	Systems Designers Software Inc., 444 Washington Street, Woburn, Massachusetts 01801, U.S.A.
VP-Expert	Paperback Software Inc., 2830 Ninth Street, Berkeley, California 94710, U.S.A.

A.3 Goats and goat products in Australia

F. OTHER LIVESTOCK
TABLE 38. DEER, GOATS, HORSES AND OTHER LIVESTOCK
AT 31 MARCH 1986 TO 1988
('000)

	Australia			*1988*							
	1986	*1987*	*1988*	*NSW*	*Vic*	*Qld*	*SA*	*WA*	*Tas*	*NT*	*ACT*
Deer	18.1	28.2	33.4	7.8	8.4	5.9	2.6	2.4	6.2	—	—
Goats(a)-											
Fibre	n.a.	475.6	589.0	327.2	63.4	43.1	47.7	87.9	19.2	0.3	0.2
Milk	n.a.	4.6	5.8	1.5	0.7	1.6	0.3	1.0	0.6	0.1	—
Other	n.a.	68.1	85.1	69.8	2.5	5.0	2.1	4.7	0.7	0.4	—
Total	366.0	548.3	680.0	398.4	66.6	49.6	50.1	93.5	20.5	0.8	0.4
Horses(b)-											
Stud	99.8	90.2	88.6	36.3	20.1	16.4	6.0	6.7	2.2	0.9	0.1
Other	246.5	222.3	246.0	61.3	20.6	120.9	9.4	17.1	2.8	13.6	0.4
Total	346.3	312.5	334.6	97.6	40.6	137.3	15.4	23.8	4.9	14.4	0.5
Other livestock(c)	15.9	17.2	17.4	0.4	0.2	0.5	0.6	0.1	—	15.6	—

(a) Goats for fibre and milk production are not available prior to 1987. (b) Comprises horses on establishments with agricultural activity.
(c) Includes camels, donkeys, mules and domesticated buffaloes.

TABLE 39. GOAT NUMBERS, FLEECE AND MILK PRODUCTION (a)
AT 31 MARCH 1988

Type	*Unit*	*Australia*	*NSW*	*Vic*	*Qld*	*SA*	*WA*	*Tas*	*NT*	*ACT*
Mohair-										
Does	number	185,871	91,899	24,768	18,312	21,556	24,422	4,884	—	30
Total Goats	number	318,906	168,334	38,672	30,959	34,522	36,904	9,315	200	30
Fleece production	kg	530,851	263,225	37,762	45,558	72,674	65,868	15,694	—	70
Cashmere(b)-										
Does	number	195,454	117,589	17,187	6,097	10,064	37,026	7,291	100	100
Total Goats	number	258,289	151,699	23,332	10,526	12,689	49,940	9,862	120	121
Fleece production	kg	61,383	28,469	7,685	1,819	3,041	17,712	2,656	—	—
Cashgora-										
Does	number	9,978	6,494	1,392	964	252	838	18	—	20
Total Goats	number	11,791	7,175	1,446	1,615	468	1,037	19	—	31
Fleece production	kg	4,741	2,377	533	752	150	914	15	—	—
Milking-										
Does	number	4,927	1,364	618	1,153	280	913	511	88	—
Total Goats	number	5,792	1,516	661	1,564	336	988	603	124	—
Milk production	litres	1,144,106	286,619	122,070	223,164	108,826	285,427	113,000	5,000	—
Other-										
Does	number	47,674	36,171	1,934	3,479	921	4,462	401	305	1
Total Goats	number	85,072	69,773	2,486	4,954	2,124	4,666	704	365	1
Total Goats-										
Does	**number**	**443,904**	**253,517**	**45,899**	**30,005**	**33,073**	**67,661**	**13,105**	**493**	**151**
Total Goats	**number**	**680,003**	**398,420**	**66,597**	**49,618**	**50,139**	**93,535**	**20,503**	**809**	**183**

(a) Milk for human consumption. (b)Total fleece weight including guard hair.

Tables taken from (Australian Bureau of Statistics, 1989)

A.4 Results from pilot veterinary survey

A pilot survey consisting of twelve questions was sent out to 40 veterinary clinics located in Western Australia; of these, 20 were located in the Perth metropolitan area, and 20 in country areas. No claim is made here about any statistical significance attributable to such a small sample.

Thirty-one of the forty surveys (77%) were completed and returned.

Only those questions with relevance to the development of the expert system are reported here.

In answer to Q3, "Does the clinic use a computer ?", 9 replied YES, 22 NO.

Those replying NO to Q3 were asked "Is the clinic likely to purchase or hire a computer during the next eighteen months ?", 4 replied YES, 15 NO, 2 wrote in "Maybe", and 1 did not answer.

Those replying YES to Q3 were asked "What type of computer(s) does the clinic use ?". The responses were IBM (or compatible) 7, Apple (or

MacIntosh) 0, Commodore (or Amiga) 1, Other 1.

Ten of the forty surveys were sent out with a final question which read "If such an expert system retailed for around $450, would you be..." to which 0 replied "Very Interested", 1 replied "Possibly Interested", and 7 replied "Not Interested". (Only 8 replies were received). Two of the respondents pointed out that the system should cost no more than a good book on the same topic.

The remaining thirty of the forty surveys suggested a price of $45 (rather than $450). Of the 23 replies, 8 replied that they would be "Very Interested", 3 "Possibly Interested", and 12 "Not Interested".

Other questions, of no direct relevance here, involved the number of veterinarians at the clinic; the frequency of consultations concerned with goats; the level of expertise with regard to goats; and the uses to which any existing computers were put.

A.5 Extracts from the knowledge base

QUALIFIERS:

1 This is the first of three screens listing clinical signs. Please list ALL signs that you have observed in the goat(s) in question. Clinical signs are

Abortion

Abscesses / Body swellings

Anaemia

Ataxia, incoordination and staggering

Blindness

Circling

Coma

Conjunctivitis

Convulsions

Coughing

Death lingering

Dyspnoea

None of the above

Used in rule(s): 114 182 184 253 392 420
425 431 432 433 434 435
436 437 438 439 440 441
442

2 This is the second of the three screens listing clinical signs. Please reply by listing all signs you have observed. The clinical signs are

Excitability

Facial Paralysis

Failure to Cycle

Head Pressing

Hyperaesthesia

Illthrift

Lameness

Low Temperature

Milk Composition

Milk Taint

Nasal discharge

Nystagmus

None of the above

Used in rule(s): 112 113 184 348 363 443
444 445 446 447 448 449
450 451 452 453 454

3 This is the third and final screen listing clinical signs. The signs that have been observed include

Opisthotonus

Paralysis

Photosensitization

Pyrexia

Scouring in Adults or Kids over 4 weeks

Scouring in Kids under 4 weeks

Short oestrus cycles, or failure to conceive

Skin disease

Stargazing

Sudden Death

Tremors

Udder Swelling

None of the above

Used in rule(s):	83	110	111	133	181	253
	345	348	362	369	409	455
	456	457	458	459	460	461
	462	463	464	465	466	

4 One or more deaths has occurred

yes

no

Used in rule(s): (409) 410

5 A post-mortem has been carried out

yes

no

Used in rule(s): 119 120 121 154 260 292

(410) 415

6 The goat(s) affected are

bucks

does

Used in rule(s): 33 113 114 242 243 244

269 331 401 427

7 The goats affected are

dairy goats

fibre goats

Used in rule(s): 59 71 84 95 157 215

216 364 370 419 426 427

8 Other goats in the herd show external abscesses or scar tissue

yes

no

Used in rule(s): 165 166 167

9 There are signs of wasting, chronic coughing and dyspnoea

yes

no

Used in rule(s): 168 169 170

10 There are signs of diarrhoea, illthrift, weight loss, and anaemia

yes

no

Used in rule(s): (411)

11 Recent feed has included

bracken or rock fern

a high concentrate diet

none of the above

Used in rule(s): 178 217

12 The age(s) of the goats or kids in question is/are

1 week or less

1 week --> 4 weeks

1 month --> 6 months

Over 6 months

Used in rule(s): 52 53 59 135 138 152

159 161 185 194 199 208

212 213 215 216 222 224

238 258 259 260 261 262

269 277 288 289 290 304

306 317 325 331 364 376

384 396 427

14 Tests or analyses so far carried out include

Urine analysis

Faecal analysis

Blood samples

CSF analysis

Rapid Mastitis test

Somatic Cell count

Milk samples

Bacterial culture

Soil analysis

None of the above

Used in rule(s):	1	3	5	7	29	30
	37	78	79	84	91	92
	93	94	153	163	164	173
	174	179	(412)	413	421	422
	423	424	428	429		

15 Have any test or analyses been carried out so far ?

yes

no

Used in rule(s): 412

17 Blood samples indicate

low glucose levels

low transketalose levels

low copper levels (below 500 ug/l)

hypomagnesaemia and/or hypocalcaemia

low cholinesterase activity

low beta-mannosidase levels

high white cell count and neutrophilia

presence of trypanasomes

none of the above (or unknown)

Used in rule(s): 13 77 84 137 188 259

266 316 333 395 (413)

18 There is an increased respiratory rate, perhaps with nasal discharge and abnormal lung sounds

yes

no

Used in rule(s): 350

31 Has any treatment been given so far ?

yes

no

Used in rule(s): 414

32 Goats have been treated with

Antibiotics

Vitamin B12

Copper supplement

Selenium supplement

Levamisole

Benzimidazole(s)

Tylosin

Calcium disodium versenate

Calcium borogluconate

Atropine

Magnesium / Calcium salts

Methylene blue

Thiamine

Antisera

Sodium Nitrate

Intravenous glucose

Cough suppressant

Cobalt bullets

Thyroxin

Dieuretics

None of the above

Used in rule(s): 14 15 24 25 54 61

62	87	107	127	128	132
134	175	176	197	219	220
256	258	263	264	279	280
305	329	334	335	340	377
378	(414)				

33 The area where the goats are kept could best be described as

Coastal calcareous sands

Tropical pastures

Sandy and flood plain soils

Acid soils in high rainfall areas

"Red Gum Country"

None of the above (or unknown)

Used in rule(s): 19 27 379

36 Treatment has been

effective

ineffective

unknown

Used in rule(s): 14 15 24 25 54 61

62 87 107 127 128 134

175 176 197 219 220 256

258 263 264 279 280 305

329 334 335 340 377 378

37 The skin disease is

pruritic

non-pruritic

Used in rule(s): 199 209 228 229 230 232

301 308 309 357 358 359

375 388 389 390 403 404

405 408

39 Post-mortem has indicated

High blood lactate levels

Decreased rumen pH

Large amount of grain or scraps in rumen

An empty abomasum

Utilisation of fat reserves

Increased rumen pH

High blood ammonia levels

Heavy firm grey lungs

Brain is fluorescent under UV light

A yellow tinge to the cerebral cortex

Shrunken liver with thick bile ducts

Scar tissue replacement in the gut

Rumen contents includes bits of leaf with bitter almond smell

Red frothy blood discharge from mouth and nose

Very low pH in rumen and/or urine

Enlarged thyroid

Erosive stomatitis and abomasal congestion

Abomastitis, enteritis and septicaemia

None of the above

Used in rule(s): 20 28 160 172 206 257
260 271 278 292 347 371
399 (415)

41 There is straining on urination, frequent urination, and/or staining of the coat below the vulva

yes

no

Used in rule(s): 55 56

77 There is ruminal stasis and constipation, and intense jaundice

yes

no

Used in rule(s): 295

78 There is a history of recent access to

poisonous lantana varieties

azaleas, oleanders or yews

sugar or manna gums

the bark, fruit or leaves of avocado plants

Ellangowan poison bush or boobialla

pure stands of signal grass

caltrops or yellow vine

variegated thistle, mintweed, or capeweed

rapidly growing young sorghum or sudan grass type crops soda bush, sour sob, rhubarb, spinach, dock or pigweed

zamia palms

other dangerous plant varieties

None of the above

Used in rule(s): 83 98 99 100 101 146

204 235 294 326 330 341

407 (416)

92 The goat has within the last hour been injected with procaine penicillin, procaine local anaesthetic, or clostridial vaccines

yes

no

Used in rule(s): 131

93 There is a fine red ring at the base of the milk container

yes

no

Used in rule(s): 35

94 There is evidence of a vulvo-vaginitis

yes

no

Used in rule(s): 149 151

95 The coughing is chronic and there is an abnormal pulse

yes

no

Used in rule(s): 162

96 Type of anaemia indicated is

Blood loss

Haemolytic

Aplastic

Hepatic Disease

Protein Loss

Aplastic with hyper gamma globulinaemia

Other

Unknown

Used in rule(s): 30

100Swellings are at base of wattle and contain clear or thin yellow fluid

yes

no

Used in rule(s): 210 (417)

101There are scabs in the external ear

yes

no

Used in rule(s): 214

139The goats are showing signs of

loss of condition

a rough coat

loss of appetite

diarrhoea

itching and rubbing

cold extremeties

depression

none of the above

Used in rule(s): 6 8 9 11 22 27

49 85 191 269 372 383

411

142The goats are

Angoras

Cashmeres

Neither of the above

Used in rule(s): 45 60 189 273 277 384

388 (419)

143There is blindness and exaggerated leg movements while walking

yes

no

Used in rule(s): 300

159The abscesses or swellings are mainly

At the site of a previous injection

On the head, neck, shoulders or backline

Under the jaw

In the cheek or jaw area

Around the lips or cheeks

At the base of the wattles

In the ventral neck

Either side of the trachea

On the abdomen

At the back or middle of the udder

On the sides of the udder

In the umbilical region

Under the tail or around the vulva

On the feet

At the base of the horns or scurs

None of the above

Used in rule(s): 63 207 215 216 240 241

270 276 282 307 370 388

389 390 391 396 417 (420)

160Urine is positive for glucose

yes

no

unknown

Used in rule(s): 179

161Faecal analysis indicates unusually high thiaminase activity

yes

no

unknown

Used in rule(s): 174

162Blood samples indicate very low erythrocyte transketalose activity

yes

no

unknown

Used in rule(s): 173

163Soil analysis indicates cobalt levels are

below 0.8 ppm

above 0.8 ppm

unknown

Used in rule(s): 26 (421)

164Blood vitamin B12 levels are

below 0.2 ug/mL

above 0.2 ug/mL

unknown

Used in rule(s): 23 (413)

170Soil analysis indicates sulphur levels are

less than 2 g/kg

more than 2 g/kg

unknown

Used in rule(s): 16 (421)

171Post-morten has indicated liver copper levels

below 50 ppm

50 - 400 ppm

above 400 ppm

unknown

Used in rule(s): 12 200 (415)

172There is some evidence of a masculinization and softening of the ligaments around the tail base

yes

no

Used in rule(s): 57

173Kids may have been fed a poor quality milk replacer or skim milk powder

yes

no

Used in rule(s): 212 213

198The goats are undernourished or overfat, or carrying a large number of foetuses

yes

no

Used in rule(s): 75 76 80

199Urine analysis indicates increased levels of ketone bodies

yes

no

unknown

Used in rule(s): 78 79

200The doe(s) are showing signs of masculine behaviour and smell, and possibly early development of a beard

yes

no

Used in rule(s): 355

201The goat(s) have recently been milked for the first time

yes

no

Used in rule(s): 360 361

239The Angoras are well-fleeced but of small body size

yes

no

Used in rule(s): 45

242There are dogs or dingoes on or around the property

yes

no

Used in rule(s): 252

243Feeders may have been contaminated with soil, or goats have been foraging on pine needles

yes

no

Used in rule(s): 302

244There is a history of access to poorly fermented or alkaline silage, sewage, weathered hay or soil contaminated feed or water

yes

no

Used in rule(s): 103

245CSF analysis results

show increased white cell count (usually 0-4 per cu mm)

show increased protein levels (usually 0-39 mg/dl)

show large numbers of erythrocytes

show none of the above

are unknown

Used in rule(s): 104 343 (428)

246Urine analysis results

show strong glucosuria

show moderate ketonuria

show neither of the above

are unknown

Used in rule(s): 105 (429)

247There is a history of colibacillosis or joint infection

yes

no

Used in rule(s): 317

278Abortion(s) occurred

in early gestation

mid-term

in late gestation

at full term

Used in rule(s): 406 430

Subject:

The diagnosis of the diseases of fibre and dairy goats.

Author:

Tim S Roberts

Starting text:

Welcome to the GOATS expert system ! The knowledge base in this system is largely based on the work of Dr S A Baxendell BVSc, PhD, MACVSc.

Date: 06.12.89

Version: 06F Status: PROTOTYPE

Rules: 468

Diseases: 221

When asked for the clinical signs, respond with ONLY THOSE SYMPTOMS WHICH YOU ARE SURE ABOUT.

When asked a question which requires a Yes or No answer, and you are unsure, then choose NO.

A normal consultation will last a few minutes.

Ending text:

Diagnosis follows - the diseases are listed in descending order of likelihood, i.e. the most probable disease is at the top. Diseases with likelihood factors of 2 or less are possible but unlikely.

Please note that this diagnosis should be regarded AS A GUIDE ONLY. Its aim is to help the veterinarian, not replace him or her !

Diseases in the diagnosis marked with a * are normally considered exotic to Australasia.

Uses all applicable rules in data derivations.

RULES:

RULE NUMBER: 1

IF:

Tests or analyses so far carried out include NOT Faecal analysis

and Goats may have been kept in overcrowded and/or hot wet conditions yes

THEN:

[109] Haemonchosis - Probability=6/10

and [225] Trichostrongylosis - Probability=7/10

RULE NUMBER: 2

IF:

Goats have had access to pasture yes

THEN:

[109] Haemonchosis - Probability=8/10

and [225] Trichostrongylosis - Probability=8/10

ELSE:

[109] Haemonchosis - Probability=1/10

and [225] Trichostrongylosis - Probability=1/10

RULE NUMBER: 3

IF:

Tests or analyses so far carried out include Faecal analysis

and Faecal egg counts are significant for None of the above

THEN:

[109] Haemonchosis - Probability=0/10

and [225] Trichostrongylosis - Probability=0/10

and [50] Chronic Fascioliasis - Probability=0/10

RULE NUMBER: 5

IF:

Tests or analyses so far carried out include Faecal analysis

and Faecal egg counts are significant for Haemonchus (>500 epg)

THEN:

[109] Haemonchosis - Probability=9/10

and [98] Gastro-Intestinal Parasitism - Probability=9/10

RULE NUMBER: 6

IF:

Faecal analysis results are unknown

and The goats are showing signs of loss of condition or a rough coat

THEN:

[98] Gastro-Intestinal Parasitism - Probability=7/10

RULE NUMBER: 7

IF:

Tests or analyses so far carried out include Faecal analysis and Faecal egg counts are significant for Trichostrongylus (>500 epg)

THEN:

[225] Trichostrongylosis - Probability=9/10

and [98] Gastro-Intestinal Parasitism - Probability=9/10

RULE NUMBER: 8

IF:

The goats are showing signs of loss of condition

and The goats are showing signs of a rough coat

THEN:

[88] Fascioliasis - Probability=5/10

RULE NUMBER: 9

IF:

The goats are showing signs of itching and rubbing

THEN:

[87] External Parasites - Probability=5/10

....

RULE NUMBER: 33

IF:

The goat(s) affected are bucks

THEN:

[7] Acute Metritis - Probability=0/10

and [8] After acute illness and fever in the doe - Probability=0/10

and [25] Blood in the milk from a leaking capillary - Probability=0/10

and [67] Cystic Ovaries - Probability=0/10

and [68] Cystitis - Probability=3/10

and [76] Dystocia - Probability=0/10

and [95] Freemartins - Probability=0/10

and [97] Gangrenous Mastitis - Probability=0/10

and [117] Hydrometria - Probability=0/10

and [146] Maiden Milkers - Probability=0/10

and [184] Pregnancy Toxaemia - Probability=0/10

and [167] Nutritional Stress - Probability=0/10

and [215] Stress / Dog Worry - Probability=0/10

and [235] Vulval Melanoma / Squamous Cell Carcinomas - Probability=0/10

and [147] Mastitis - Probability=0/10

and [165] Normal Pregnancy from unrecorded service - Probability=0/10

and [186] Postaglandin Induced Short Cycles - Probability=0/10 and [178] Physiological Short Cycles - Probability=0/10

and [218] Subclinical Uterine Infection - Probability=0/10

and [228] Udder Oedema - Probability=0/10

and [153] Milk Fever - Probability=0/10

and [197] Retention of kids at kidding - Probability=0/10

and [208] Sheep x Goat Mating - Probability=0/10

and [44] Chlamydial Abortion - Probability=0/10

and [193] Q Fever Abortion - Probability=0/10

and [107] Habitual Abortion of Angoras - Probability=0/10

and [238] Witch's Milk - Probability=0/10

RULE NUMBER: 53

IF:

The age(s) of the goats or kids in question is/are Over 6 months

and There are signs of kidding - large abdomen, relaxed ligaments at base of tail, a large swollen vulva and milk in the udder yes

and There is a yellow vaginal discharge yes

THEN:

[76] Dystocia - Probability=8/10

RULE NUMBER: 65

IF:

There is a history of owner intervention at previous kidding or

clinical metritis no

and Lochia was abnormal or persisted for an abnormally long period yes

THEN:

[218] Subclinical Uterine Infection - Probability=5/10

RULE NUMBER: 66

IF:

There is a history of owner intervention at previous kidding or

clinical metritis yes

and Lochia was abnormal or persisted for an abnormally long period no

THEN:

[218] Subclinical Uterine Infection - Probability=5/10

RULE NUMBER: 67

IF:

There is a history of owner intervention at previous kidding or

clinical metritis yes

and Lochia was abnormal or persisted for an abnormally long period yes

THEN:

[218] Subclinical Uterine Infection - Probability=7/10

RULE NUMBER: 87

IF:

Goats have been treated with Calcium borogluconate

and Treatment has been effective

THEN:

[153] Milk Fever - Probability=9/10

RULE NUMBER: 88

IF:

The goats are high milk-producers, close to kidding yes

THEN:

[153] Milk Fever - Probability=5/10

RULE NUMBER: 107

IF:

Goats have been treated with Antibiotics

and Treatment has been effective

THEN:

[17] Bacterial Pneumonia - Probability=9/10

and [40] Caseous Lymphadenitis (C.L.A.) Lung Abscess - Probability=1/10

and [41] Caseous Lymphadenitis Internal Abscesses - Probability=1/10

and [142] Listeriosis - Probability=8/10

and [175] Pasteurellosis - Probability=6/10

RULE NUMBER: 111

IF:

This is the third and final screen listing clinical signs. The signs that have been observed include Skin disease

THEN:

display(goats.ski)

and * dummy disease * - Probability=1/10

RULE NUMBER: 112

IF:

This is the second of the three screens listing clinical signs. Please reply by listing all signs you have observed. The clinical signs are Milk Taint

THEN:

display(goats.mil)

and * dummy disease * - Probability=1/10

RULE NUMBER: 117

IF:

There is a possibility that the goats have had access to poisonous grasses or shrubs More information, please !

THEN:

DISPLAY(goats.pla)

and CLEAR(Q 28)

and CLEAR(R 117)

ELSE:

The plant information is known yes

RULE NUMBER: 118

IF:

Post-mortem has found presence of larvae and evidence of pressure atrophy of the brain

THEN:

[2] Aberrant migration of Oestrus Ovis - Probability=9/10

ELSE:

[2] Aberrant migration of Oestrus Ovis - Probability=1/10

RULE NUMBER: 121

IF:

A post-mortem has been carried out yes

and There is a dark oedamatous bowel with a distinct demarcation line with the normal gut yes

THEN:

[3] Abomasal-Intestinal Torsion - Probability=8/10

RULE NUMBER: 138

IF:

The age(s) of the goats or kids in question is/are 1 week or less

and The goats are Anglo Nubians

and The goats have been unable to stand from birth yes

THEN:

[19] Beta-Mannosidosis - Probability=8/10

RULE NUMBER: 153

IF:

Tests or analyses so far carried out include CSF analysis

and Protein value > 40 mg/dl and/or white cell count > 5 per cu mm yes

THEN:

[33] Caprine Retrovirus (CAE) - Encephalitic - Probability=10/10

RULE NUMBER: 154

IF:

A post-mortem has been carried out yes

and Post-mortem has found brownish areas on cross-section of the brain or spinal cord

THEN:

[33] Caprine Retrovirus (CAE) - Encephalitic - Probability=10/10

RULE NUMBER: 163

IF:

Tests or analyses so far carried out include Bacterial culture and Corynebactgerium pseudotuberculosis has been identified no

THEN:

[39] Caseous Lymphadenitis (C.L.A.) - Probability=1/10

RULE NUMBER: 164

IF:

Tests or analyses so far carried out include Bacterial culture

and Corynebactgerium pseudotuberculosis has been identified yes

THEN:

[39] Caseous Lymphadenitis (C.L.A.) - Probability=10/10

RULE NUMBER: 165

IF:

Lymph nodes are swollen with abscesses three centimetres or more in diameter yes

and Other goats in the herd show external abscesses or scar tissue no

THEN:

[39] Caseous Lymphadenitis (C.L.A.) - Probability=6/10

RULE NUMBER: 166

IF:

Lymph nodes are swollen with abscesses three centimetres or more in diameter yes

and Other goats in the herd show external abscesses or scar tissue yes

THEN:

[39] Caseous Lymphadenitis (C.L.A.) - Probability=9/10

RULE NUMBER: 167

IF:

Lymph nodes are swollen with abscesses three centimetres or more in diameter no

and Other goats in the herd show external abscesses or scar tissue yes

THEN:

[39] Caseous Lymphadenitis (C.L.A.) - Probability=0/10

and [40] Caseous Lymphadenitis (C.L.A.) Lung Abscess - Probability=8/10

and [41] Caseous Lymphadenitis Internal Abscesses - Probability=8/10

RULE NUMBER: 168

IF:

There are signs of wasting, chronic coughing and dyspnoea yes

and The goats are adult dairy bucks yes

THEN:

[40] Caseous Lymphadenitis (C.L.A.) Lung Abscess - Probability=9/10

and [41] Caseous Lymphadenitis Internal Abscesses - Probability=9/10

RULE NUMBER: 333

IF:

Blood samples indicate low cholinesterase activity

THEN:

[168] Organophosphate Poisoning - Probability=9/10

RULE NUMBER: 334

IF:

Goats have been treated with Atropine

and Treatment has been ineffective

THEN:

[168] Organophosphate Poisoning - Probability=1/10

RULE NUMBER: 335

IF:

Goats have been treated with Atropine

and Treatment has been effective

THEN:

[168] Organophosphate Poisoning - Probability=9/10

RULE NUMBER: 336

IF:

There is salivation, dyspnoea, muscle stiffness, ataxia and contraction of the pupils yes

THEN:

[168] Organophosphate Poisoning - Probability=9/10

ELSE:

[168] Organophosphate Poisoning - Probability=1/10

RULE NUMBER: 337

IF:

There is a history of very recent contact with organophosphates eg drenching or dipping etc yes

THEN:

[168] Organophosphate Poisoning - Probability=7/10

ELSE:

[168] Organophosphate Poisoning - Probability=0/10

RULE NUMBER: 338

IF:

Head is tilted and there is circling in one direction, nystagmus the other way yes

THEN:

[169] Otitis Media / Interna - Probability=5/10

RULE NUMBER: 359

IF:

The skin disease is pruritic

and The skin disease is on the head, ears, or base of tail and backline yes

THEN:

[191] Psoroptic Mange - Probability=5/10

RULE NUMBER: 360

IF:

There are tiny pustules on the udder, under tail and on inner thigh yes

and The goat(s) have recently been milked for the first time no

THEN:

[192] Pustular Dermatitis - Probability=5/10

RULE NUMBER: 361

IF:

There are tiny pustules on the udder, under tail and on inner thigh yes

and The goat(s) have recently been milked for the first time yes

THEN:

[192] Pustular Dermatitis - Probability=7/10

RULE NUMBER: 375

IF:

The skin disease is pruritic

and The skin disease is on the head, ears, or base of tail and backline yes

and There is extreme pruritis with skin thickening and alopecia, and peripheral lymph nodes are increased in size yes

THEN:

[205] Sarcoptic Mange - Probability=7/10

RULE NUMBER: 376

IF:

The age(s) of the goats or kids in question is/are Over 6 months

and When rubbed on abdomen or flank, goat draws back its lips and shows its teeth yes

THEN:

[206] Scrapie * - Probability=3/10

RULE NUMBER: 377

IF:

Goats have been treated with Selenium supplement

and Treatment has been ineffective

THEN:

[207] Selenium Deficiency - Probability=1/10

RULE NUMBER: 409

IF:

This is the third and final screen listing clinical signs. The signs that have been observed include Sudden Death

THEN:

One or more deaths has occurred yes

RULE NUMBER: 410

IF:

One or more deaths has occurred no

THEN:

A post-mortem has been carried out no

RULE NUMBER: 411

IF:

The goats are showing signs of NOT diarrhoea

THEN:

There is or was yellow, green pasty diarrhoea, rapidly becoming watery, low volume, with mucous, shreds of bowel mucosa and/or blood no

and There was severe gastro-enteritis with a blue-green diarrhoea no

and There is profuse watery-yellow to creamy-white diarrhoea no

and There are signs of diarrhoea, illthrift, weight loss, and anaemia no

and Diarrhoea is foul-smelling and kids are pot-bellied no

and There is diarrhoea which is dark and foetid no

RULE NUMBER: 413

IF:

Tests or analyses so far carried out include NOT Blood samples

THEN:

AGID (or ELISA) test in herd is unknown

and Blood vitamin B12 levels are unknown

and Blood samples indicate none of the above (or unknown)

and Gel Diffusion Precipitin Test (GDPT) result is unknown

and Blood lead levels are unknown

and C.F. test results are unknown

RULE NUMBER: 414

IF:

Has any treatment been given so far ? no

THEN:

Goats have been treated with None of the above

RULE NUMBER: 416

IF:

The plant information is known yes

and There is a possibility that the goats have had access to poisonous grasses or shrubs no

THEN:

There is a history of recent access to None of the above

RULE NUMBER: 417

IF:

The abscesses or swellings are mainly NOT At the base of the wattles

THEN:

Swellings are at base of wattle and contain clear or thin yellow fluid

no

RULE NUMBER: 418

IF:

Volume of milk is markedly reduced no

THEN:

There has been a severe reduction in milk volume, and the milk is very

thick no

and Milk is very watery and very reduced in volume no

RULE NUMBER: 421

IF:

Tests or analyses so far carried out include NOT Soil analysis

THEN:

Soil analysis indicates cobalt levels are unknown

and Soil analysis indicates copper levels are unknown

and Soil analysis indicates molybdenum levels are unknown

and Soil analysis indicates sulphur levels are unknown

RULE NUMBER: 423

IF:

Tests or analyses so far carried out include NOT Bacterial culture

THEN:

The antigenic pillus K99 (or K88 + 987 antigen) is unknown

RULE NUMBER: 424

IF:

Tests or analyses so far carried out include NOT Milk samples

THEN:

Milk sample shows bacterial growth unknown

RULE NUMBER: 425

IF:

This is the first of three screens listing clinical signs. Please list ALL signs that you have observed in the goat(s) in question. Clinical signs are Anaemia

THEN:

The goats are showing signs of anaemia (eg pale mucous membranes and a rapid pulse) yes

----

RULE NUMBER: 427

IF:

The goat(s) affected are bucks

and The goats affected are dairy goats

and The age(s) of the goats or kids in question is/are Over 6 months

THEN:

The goats are adult dairy bucks yes

ELSE:

The goats are adult dairy bucks no

------------------ ------------------

RULE NUMBER: 428

IF:

Tests or analyses so far carried out include NOT CSF analysis

THEN:

CSF analysis results are unknown

RULE NUMBER: 429

IF:

Tests or analyses so far carried out include NOT Urine analysis

THEN:

Urine analysis results are unknown

RULE NUMBER: 430

IF:

Abortion(s) occurred in early gestation or mid-term

THEN:

Abortions occurred in very late gestation with retained foetal membranes no

and Abortion occurred within 2 months of gestation yes

RULE NUMBER: 431

IF:

This is the first of three screens listing clinical signs. Please list ALL signs that you have observed in the goat(s) in question. Clinical signs are Abortion

THEN:

[S01] IS GIVEN THE VALUE "S01:215;167;008;222;044;193;141;030;107;129;208;239"

ELSE:

[S01] IS GIVEN THE VALUE "S00"

REFERENCE:

pp1,2

RULE NUMBER: 432

IF:

This is the first of three screens listing clinical signs. Please list ALL signs that you have observed in the goat(s) in question. Clinical signs are Abscesses / Body swellings

THEN:

[S02] IS GIVEN THE VALUE "S02:039;233;125;121;109;129;078;105;202;069;072;220;145;108;113;185;036;229;149;199;103;236"

ELSE:

[S02] IS GIVEN THE VALUE "S00"

REFERENCE:

pp3-7

RULE NUMBER: 433

IF:

This is the first of three screens listing clinical signs. Please list ALL signs that you have observed in the goat(s) in question. Clinical signs are Anaemia

THEN:

[S03] IS GIVEN THE VALUE "S03:109;225;055;053;063;088;082;087;036;223;187;048;157;016"

ELSE:

[S03] IS GIVEN THE VALUE "S00"

REFERENCE:

pp7,8

RULE NUMBER: 434

IF:

This is the first of three screens listing clinical signs. Please list ALL signs that you have observed in the goat(s) in question. Clinical signs are Ataxia, incoordination and staggering

THEN:

[S04] IS GIVEN THE VALUE "S04:043;063;153;033;142;138;116;240;002;213;204;231;042;168;170;119;056;172;194;206;111;016"

ELSE:

[S04] IS GIVEN THE VALUE "S00"

REFERENCE:

pp9-14

RULE NUMBER: 435

IF:

This is the first of three screens listing clinical signs. Please list ALL signs that you have observed in the goat(s) in question. Clinical signs are Blindness

THEN:

[S05] IS GIVEN THE VALUE "S05:043;184;033;138;042;204;002;213;056;172"

ELSE:

[S05] IS GIVEN THE VALUE "S00"

REFERENCE:

pp14-17

RULE NUMBER: 464

IF:

This is the third and final screen listing clinical signs. The signs that have been observed include Sudden Death

THEN:

[S34] IS GIVEN THE VALUE "S34:109;079;055;157;057;212;149;097;022;021;006;066;164;012;013;173;174;024;046;003;135;064"

and [S34A] IS GIVEN THE VALUE "S00a014;203;111;016"

ELSE:

[S34] IS GIVEN THE VALUE "S00"

and [S34A] IS GIVEN THE VALUE "S00"

RULE NUMBER: 465

IF:

This is the third and final screen listing clinical signs. The signs that have been observed include Tremors

THEN:

[S35] IS GIVEN THE VALUE "S35:043;033;063;019;119;164;116;170;231"

ELSE:

[S35] IS GIVEN THE VALUE "S00"

BIBLIOGRAPHY

This bibliography includes all sources that were used in the development of the expert system; not all have been referred to explicitly in the text.

Addis T R (1985), Designing Knowledge-Based Systems, Kogan Page Ltd., London.

Aikens J (1983), Problems of Designing a Medical Expert System, in Artificial Intelligence #20, pp163-210.

Alty J C & Coombs M J (1984), Expert Systems: Concepts and Examples, N.C.C. Publishers, Manchester.

Australian Bureau of Statistics (1988), Year Book Australia 1988, Commonwealth of Australia, Canberra.

Australian Bureau of Statistics (1989), Livestock and Livestock Products Australia 1988, Commonwealth of Australia, Canberra.

Baxendell S A (1988), The Diagnosis of the Diseases of Goats, in the T G Hungerford Vade Mecum Series for Domestic Animals, University of Sydney Post-Graduate Foundation in Veterinary Science, Sydney.

Bettenay R A (1984), Common Ailments of Goats, Farmnote (No 14/84), W.A. Department of Agriculture, Perth.

Bielawski L & Lewand R (1988), Expert Systems Development, QED Information Sciences, Massachusetts.

Black W J (1986), Intelligent Knowledge Based Systems, Van Nostrand Reinhold, Wokingham, England.

Bode A M and Brooks H (1976), Australian Angora Goat Husbandry, A M Bode, Torrens Creek, Queensland.

Boose J H (1984), Personal Construct Theory and the Transfer of Human Expertise, in Advances in Artificial Intelligence, ed Tim O'Shea, Elsevier Science Publishers B.V., Amsterdam.

________ (1986), Expertise Transfer for Expert System Design, Elsevier Science Publishers B.V., Amsterdam.

________ (1988), Knowledge Acquisition Techniques and Tools: Current Research Strategies and Approaches, in Proceedings of the International Conference on Fifth Generation Computer Systems 1988, Institute for New Generation Computer Technology (ICOT), Tokyo, Japan.

Bowerman R G & Glover D E (1988), Putting Expert Systems into Practice, Van Nostrand Reinhold, New York.

Bramer M A (1982), A Survey and Critical Review of Expert Systems Research, in Introductory Readings in Expert Systems, ed D Michie, Gordon and Breach, London.

_________ (1986), Expert Systems in Britain: Progress and Prospects, in Research and Development in Expert Systems III, ed M A Bramer, Cambridge University Press, Cambridge.

Buchanan B G & Shortliffe E H eds. (1984), Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project, Addison-Wesley, Massachusetts.

Bundy A (1978), Will It Reach The Top ? Prediction in the Mechanics World, in Artificial Intelligence #10, pp129-146.

Clancey W J (1983), The Epistemology of a Rule-Based Expert System: A Framework for Explanation, in Artificial Intelligence #20, pp215-251.

Colomb R M (1987), Expert Systems: Early Applications in Australia, CSIRO Technical Report P-FC-87-04.

_________ (1987), Future Development of Computers for Farm Decision Support, CSIRO Technical Report PP-FB-87-06.

Conley W (1987), Intelligent Simulation Models, A Demographic Simulator with Heuristic Reasoning, in Coupling Symbolic and Numerical Computing In Expert Systems II, ed Kowalik J & Kitzmiller C, Elsevier Science Publishers, Amsterdam.

Cookson M J, Holman J G, & Thompson D G (1985), Knowledge Acquisition for Medical Expert Systems, in Research and Development in Expert Systems, ed M A Bramer, Cambridge University Press, Cambridge.

Coop I E ed. (1981), Sheep and Goat Production, Elsevier Scientific Publishers, Amsterdam.

Cox P R (1984), How We Built Micro Expert, in Expert Systems: Principles and Case Studies, ed R Forsyth, Chapman & Hall Ltd., London.

Devendra C & McElroy G (1982), Goat and Sheep Production in the Tropics, Longman Group, London.

Ellis T M (1985), Big Knee Virus of Goats, Farmnote (No 60/85), W.A. Department of Agriculture, Perth.

EXSYS (1985), Expert System Development Package manual (1985), California Intelligence, California.

Feigenbaum E (1971), On Generality and Problem Solving, in Machine Intelligence #6.

Ferrand P (1984), SESAM: An Explanatory Medical Aid System, in Advances in Artificial Intelligence,ed Tim O'Shea, Elsevier Science Publishers B.V., Amsterdam.

Forsyth R ed. (1984), Expert Systems: Principles and Case Studies, Chapman & Hall Ltd., London.

Freedman R S & Shooman A M (1985), MERLIN: A Tool for Automated Inferencing with Uncertainty, in Artificial Intelligence Applications: The Engineering of Knowledge-Based Systems, ed C R Weisbin, IEEE Computer Society Press, Washington D.C.

Gall C ed. (1981), Goat Production, Academic Press, London.

Gammack J G & Young R M (1985), Psychological Techniques for Eliciting Expert Knowledge, in Research and Development in Expert Systems, ed M A Bramer, Cambridge University Press, Cambridge.

Gilb T (1988), Principles of Software Engineering Management, Addison-Wesley, Wokingham, England.

Gilmore J F & Pulaski K (1985), A Survey of Expert System Tools, in Artificial Intelligence Applications: The Engineering of Knowledge-Based Systems, ed C R Weisbin, IEEE Computer Society Press, Washington D.C.

Goodman I & Nguyen H T (1985), Uncertainty Models for Knowledge-Based Systems, Elsevier Science Publishers B.V., Amsterdam.

Harmon P, Maus R, & Morrissey W (1988), Expert Systems: Tools and Applications, John Wiley & Sons, New York.

Hart A (1986), Knowledge Acquisition for Expert Systems, McGraw-Hill, New York.

Hasling D W et al (1984), Strategic Explanations for a Diagnostic Consultation System, in Developments in Expert Systems, ed M J Coombs, Academic Press, London.

Hayes-Roth F, Waterman D A, & Lenat D B eds. (1983), Building Expert Systems, Addison-Wesley Publishing Company, Massachusetts.

Hayward S A (1984), Is A Decision Tree An Expert System ?, in Research and Development in Expert Systems, ed M A Bramer, Cambridge University Press, Cambridge.

Hoffman R R (1987), The Problem of Extracting the Knowledge of Experts from the Perspective of Experimental Psychology, in AI Magazine, Summer 1987.

Howe P A (1984), Diseases of Goats, University of Sydney Post-Graduate Foundation in Veterinary Science, Sydney.

Jackson P (1986), Introduction to Expert Systems, Addison-Wesley, Wokingham, England.

Jansen B (1987), Applying Software Engineering Concepts to Rule Based Expert Systems, CSIRO Technical Report TR-FD-87-02.

Jeffery H E (1970), Goats, Cassell & Co. Ltd., London.

Jones C ed. (1986), Programming Productivity: Issues for the Eighties, IEEE Computer Society Press, Washington D.C.

Karlsson L J E (1984), Pregnancy Toxaemia and Ketosis in Goats, Farmnote (No 10/84), W.A. Department of Agriculture, Perth.

Keen M J R & Williams G (1984), Expert System Shells Come Of Age, in Research and Development in Expert Systems, ed M A Bramer, Cambridge University Press, Cambridge.

Keller R (1987), Expert System Technology: Development and Application, Yourdon Press, Eaglewood Cliffs, New Jersey.

Keravnou E T & Johnson L (1986), Competent Expert Systems, Kogan Page Ltd., New York.

Kidd A ed. (1987), Knowledge Acquisition for Expert Systems: A Practical Handbook, Plenum Press Ltd., New York.

King D (1984), Current Practices in Software Development, Yourdon Press, New York.

Kowalik J S & Kitzmiller C T eds. (1988), Coupling Symbolic and Numerical Computing in Expert Systems II, Elsevier Science Publishers B.V., Amsterdam.

Lemmer J F & Kanal L N eds. (1988), Uncertainty in Artificial Intelligence 2, Elsevier Science Publishers B.V., Amsterdam.

Lenat D B (1988), When Will Machines Learn ?, in Proceedings of the International Conference on Fifth Generation Computer Systems 1988, Institute for New Generation Computer Technology (ICOT), Tokyo, Japan.

Liebowitz J (1985), Evaluation of Expert Systems: An Approach and Case Study, in Artificial Intelligence Applications: The Engineering of Knowledge-Based Systems, ed C R Weisbin, IEEE Computer Society Press, Washington D.C.

__________ (1988), Introduction to Expert Systems, Mitchell Publishing Inc., California.

MacKenzie D (1970), Goat Husbandry, Faber and Faber, London.

Minsky M (1975), A Framework For Representing Knowledge, in The Psychology of Computer Vision, ed P H Winston, McGraw-Hill, New York.

National Goatbreeders Conference (1979), Australian Goatbreeding (Dairy and Angora): Husbandry Science and Veterinary Practice - Proceedings of the 2nd National Goatbreeders Conference, Perth 1979, Perth.

Naylor C (1984), How To Build An Inferencing Engine, in Expert Systems: Principles and Case Studies, ed R Forsyth, Chapman and Hall Ltd., London.

Negoita C V (1985), Expert Systems and Fuzzy Systems, Benjamin Cummings Publishing Co., California.

Poole D (1988), Representing Knowledge for Logic-based Diagnosis, in Proceedings of the International Conference on Fifth Generation Computer Systems 1988, Institute for New Generation Computer Technology (ICOT), Tokyo, Japan.

Prerau D S (1987), Knowledge Acquisition in the Development of a Large Expert System, in AI Magazine, Summer 1987.

Salmon J (1981), The Goatkeeper's Guide, David & Charles, Newton Abbott, England.

Shaw M L G (1988), Validation in a Knowledge Acquisition System with Multiple Experts, in Proceedings of the International Conference on Fifth Generation Computer Systems 1988, Institute for New Generation Computer Technology (ICOT), Tokyo, Japan.

Shortliffe E (1976), Computer Based Medical Consultations: MYCIN, American Elsevier, New York.

Sloman A (1985), Why We Need Many Knowledge Representation Formalisms, in Research and Development in Expert Systems, ed M A Bramer, Cambridge University Press, Cambridge.

Steels L (1986), Second Generation Expert Systems, in Research and Development in Expert Systems III, ed M A Bramer, Cambridge University Press, Cambridge.

Stevens A (1984), How Shall We Judge An Expert System ?, in Expert Systems: Principles and Case Studies, ed R Forsyth, Chapman & Hall Ltd., London.

Taki H (1988), Knowledge Acquisition by Observation, in Proceedings of the International Conference on Fifth Generation Computer Systems 1988, Institute for New Generation Computer Technology (ICOT), Tokyo, Japan.

Waterman D A (1985), A Guide To Expert Systems, Addison-Wesley, Massachusetts.

Weilings B J & Breuker J A (1984), Interpretation of Verbal Data for Knowledge Acquisition, in Advances in Artificial Intelligence, ed Tim O'Shea, Elsevier Science Publishers B.V., Amsterdam.

Weiss S M & Kulikowski C A (1984), A Practical Guide to Designing Expert Systems, Rowman & Allowheld, Totowa, New Jersey.

White A P (1984), Inference Deficiencies in Rule-Based Expert Systems, in Research and Development in Expert Systems, ed M A Bramer, Cambridge University Press, Cambridge.

Winter C L & Girse R D (1985), Evolution and Modification of Probability in Knowledge Bases, in Artificial Intelligence Applications: The Engineering of Knowledge-Based Systems, ed C R Weisbin, IEEE Computer Society Press, Washington D.C.

Wolfgram D, Deer T, & Galbraith C (1982), Expert Systems for the Technical Professional, John Wiley, New York.

Yerex D (1986), The Farming of Goats: Fibre and Meat Production in New Zealand, Ampersand Publishing Associates, Carterton, New Zealand.

* * * * *   * * * * *