R/D 5133-CC-01-F

(2)

KNOWLEDGE ACQUISITION FOR EXPERT SYSTEMS IN CONSTRUCTION

Final report

Contract DAJA 45-85-C-0033

December 1988

by E G Trimble
C N Cooper

KNOWLEDGE ACQUISITION FOR EXPERT SYSTEMS IN CONSTRUCTION

E G Trimble and C N Cooper

FINAL REPORT

Contract DAJA 45-85-C-0033

December 1988

Department of Civil Engineering
Loughborough University of Technology
Ashby Road
Loughborough
Leicestershire LE11 3TU
U.K.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

ADA203458

REPORT DOCUMENTATION PAGE

Form Approved OMB No 0704-0188 Exp Date Jun 30, 1986

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S) R&D 5133-CC-01
6a. NAME OF PERFORMING ORGANIZATION Loughborough University of Technology 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION USARDSG-UK
6c. ADDRESS (City, State, and ZIP Code) Department of Civil Engineering, Loughborough, Leicestershire, LE11 3TU	7b. ADDRESS (City, State, and ZIP Code) Box 65 FPO NY 09510-1500
8a. NAME OF FUNDING/SPONSORING ORGANIZATION USARDSG-UK ARO-E 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER DAJA45-85-C-0033
8c. ADDRESS (City, State, and ZIP Code) Box 65 FPO NY 09510-1500	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. 61102A PROJECT NO. 1L161102BH57 TASK NO 03 WORK UNIT ACCESSION NO

11. TITLE (Include Security Classification)

(U) Knowledge Acquisition for Expert Systems in Construction

12. PERSONAL AUTHOR(S)

Professor Geoffrey Trimble and Mr Chris Cooper

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final	FROM 85/8/1 TO 88/9/30	December 1988	112

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	
12	05		expert systems, knowledge-based systems, construction, domain selection, knowledge acquisition, knowledge representation, expert system shells

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The authors have investigated knowledge acquisition both by developing eight systems themselves and by involvement in a survey of 70 systems developed by others. They conclude that the key to successful expert system development lies in the organizational environment in which the application is undertaken. Hence development should be client led and problem orientated with clearly defined end users. Simple applications are more likely to be successful.

Several knowledge elicitation techniques will be required for any given application. The authors have investigated unstructured and structured interviews, case histories, rule induction, observation, paper models (intermediate representations), iterative prototyping, use of published material and the problems of eliciting uncertain knowledge.

continued on next page....

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL Dr. John M. Zavada	22b. TELEPHONE (Include Area Code) 01-409 4423 22c. OFFICE SYMBOL AMXSN-UK-RI

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

Unclassified

19 (continued)

A general approach to knowledge elicitation is suggested.

Systems developed by the authors and described in the report include BREDAMP (building dampness diagnosis), CRANES (tower crane selection), BID/NO BID (bidding decision), MATSEL (boiler tube steel selection), RRES (aeroengine fault diagnosis) and GECES (alternator out-of-balance correction). (KR)

The shells Savoir, ESE, Leonardo and Goldworks have been used and some knowledge representation issues are discussed.

CONTENTS

KNOWLEDGE ACQUISITION FOR EXPERT SYSTEMS IN CONSTRUCTION

E G Trimble and C N Cooper

1 PREAMBLE

The contract was dated July 1985. Work commenced immediately. The work statement was as follows.

Objective

To develop a methodology for acquiring information suitable for the knowledge-base of an expert system to be used in support of construction management.

Scope

The construction experts of G Wimpey Construction Company have agreed to work with the University Researchers to develop a knowledge-base for construction management. The primary concern of this research is to investigate the approaches to eliciting knowledge from non-computer oriented experts. It is anticipated that knowledge bases from case studies will be produced along with a method for developing additional databases.

In addition to Wimpey Construction UK Ltd the following organizations have been involved:

IDC Consultants Ltd (design and construct contractors)

Foster Wheeler Power Products Ltd (involved in the design and construction of large industrial boilers).

We are also able to report on our experiences of developing expert systems under contracts in which the following have been involved:

Building Research Establishment
Stewart Hughes Ltd
GEC Research
Rolls Royce plc

In 1985 and 1986 the methodology adopted was of a dual nature namely

1. To develop expert systems for various domains and to monitor our own experience in doing so.
2. To explore the knowledge acquisition process more generally and to monitor the experience of other people undertaking similar work.

At the end of 1986 we received separate funding from the (British) Science and Engineering Research Council (SERC) to undertake a study with somewhat similar objectives to that being supported by CERL. To simplify administration and with the agreement of the US Army Research Development and Standardization Group, UK, we divided the work from 1 January 1987 so that the method 1 study continued with support from CERL while method 2 proceeded with support from SERC. A summary report entitled "Knowledge acquisition for expert systems" was submitted to SERC in April 1988. They have agreed that we may provide copies on request.

Other documents prepared in connection with this work include the following:

"Experience with Knowledge Acquisition for Expert Systems in Construction", G Trimble and C N Cooper, Proc. 1st European Workshop on Knowledge Acquisition for Knowledge-based Systems, Reading, UK Sept. 1987.

"Cranes - a rule-based assistant with graphics for construction planning engineers", C N Cooper, Proc. of 3rd International Conference on Civil and Structural Engineering Computing, London, Sept 1987.

""MATSEL - use of a commercial shell in a domain involving code of practice knowledge" C N Cooper, M C Allen and A J Jones. (not yet published).

"Knowledge Elicitation: Some of the Practical Issues", G Trimble 1988 (see below).

A copy of the paper "Experience with Knowledge Acquisition for Expert Systems in Construction" is attached as Appendix 6 of this report. The papers describing the CRANES and MATSEL systems are summarised in Appendices 2 and 4 respectively. Copies of these papers are available on request. The last paper forms a chapter in a book to be published shortly on knowledge elicitation (Diaper 1989). The text of this chapter has been used in developing the main body of this report.

Apart from Professor E G Trimble, the principal investigator, the following scientific personnel have been closely involved in the execution of this research:

Dr R J Allwood

Mr A E Bryman

Mr C N Cooper

Dr J Cullen

Mr Cooper is registered for the degree of PhD and expects to submit his thesis in 1989.

2 INTRODUCTION

In recent years our team in the department of Civil Engineering at Loughborough has been studying the practical problems of building expert systems with particular reference to the knowledge acquisition process. This team has built or is building eight systems covering a wide spectrum of domains. The principal investigator has also been involved in a survey of 70 "real world" expert systems again with the objective of exploring the knowledge acquisition process. This involvement has revealed that there are two types of situation in which a system may be built:

1. An identifiable client requires a system and has instructed a contractor or employee to produce it. The client has a reasonably good idea of why he wants the system and how it will be used when complete.
2. There is no identifiable client. An enthusiast, (or a group of enthusiasts) initiates a system with the objectives of exploring the subject of expert systems and producing a demonstration system.

It is considered that the problems of knowledge acquisition can only be explored realistically in situations of type 1. In type 2 the enthusiasts usually have domain expertise themselves so there is little need to elicit expert knowledge from the domain expert via a non-domain-expert (e.g. a knowledge engineer) to prepare some form of intermediate representation.

This background has prompted the team to seek systems that are working in the real world or at least are being developed with the intention of real use. This analysis has led to the identification of the kind of situation that will promote the development of useful systems and the factors that inhibit their development. In some ways this parallels a study (Arditi 1973) to explore the factors that make for success in the application of network analysis i.e. computer based project scheduling and control. Section 3 explores this parallel and offers tentative advice as to the factors that are likely to promote the development of successful real world systems. The other sections are as follows:

- Situations that affect the elicitation process
- Knowledge Elicitation Techniques
- Knowledge Elicitation - General Advice
- Knowledge Representation
- Conclusions

The systems the team has built are summarized below:

CONPLANT	-	selection of materials handling equipment for multi-storey construction sites (Wijesundera and Harris 1985)
BREDAMP	-	diagnosis of the cause of dampness in buildings (Allwood et al 1988)
CRANES	-	selection of tower cranes for multi-storey construction sites (Cooper 1987)
BID/NO BID	-	decision support for a design and construct contractor on whether to bid for new contracts
NETWORK	-	diagnosis of faults in a national computer network
MATSEL	-	material selection for boiler pressure parts

The first five systems utilise the SAVOIR commercial shell program and run on an IBM PC XT. The last named uses the IBM shell ESE and runs on an IBM mainframe computer.

CONPLANT is an "enthusiast's" system which drew largely on the expertise within the authors' department and on expertise from Taylor Woodrow plc and Tarmac plc. All other systems were developed either within a type 1 situation (i.e. with an identifiable client) or as near to this situation as it was possible to engineer. Where companies did not specifically commission a system the industrial collaborators were encouraged to act as though they were clients and in particular to reject any part of the system which they considered unrealistic:

- o BREDAMP was specifically commissioned by the Building Research Establishment.
- o CRANES was developed using expertise from Wimpey Construction UK Ltd.
- o BID/NO BID is based on input from IDC Consultants Ltd.
- o NETWORK was commissioned by a hotel and brewing chain.
- o MATSEL was undertaken in association with Foster Wheeler Power Products Ltd.

Currently under development are two further systems funded by the British Government's Alvey Directorate. These are in association with Stewart Hughes Ltd, Rolls Royce plc and GEC Research. Both concern the interpretation of sophisticated analyses of vibrations in rotating equipment. One assists in the diagnosis of faults in aero engines; the other in the balancing of the rotors of very large alternators.

The principal investigator was also a member of two of the Community Clubs (PLANIT and ARIES) established by the Alvey Directorate. These comprise groups of industrial organizations with a common interest in exploring particular applications of expert systems. PLANIT explored the use of expert systems as an interactive aid to project managers; ARIES was formed by the insurance industry and produced one prototype system to assess premiums for fire insurance and another as an aid to investment managers in selecting equities.

3 FACTORS THAT MAKE FOR SUCCESS

In the early 1970's Arditi studied the factors that promote success in the application of network analysis techniques (NAT) such as PERT, CPM etc in construction. At the time of the study the techniques were already widely known and quite widely applied (Harries & Trimble 1974). It was therefore feasible to assemble data about real world applications of NAT and to examine them using statistical techniques. This is in contrast to the present (1988) position regarding the application of expert systems. However, the expert system study suggests that there may be some parallels that are worth examining. These are:

1. The early study demonstrated that success was strongly correlated with intitial and continuing support from senior management. Where there is a clearly identifiable client it can usually be assumed that management support has been obtained. Successful expert systems have been developed in this type of situation. However, where systems have been developed by enthusiasts there has been no need to define a client role. Such systems are seldom adopted in the real world.

2. The motivation of those involved, particularly the more senior people, was shown by the study to be a key determinant of success. This motivation was most often generated by the commitment of managers, for example the commitment that is made by spending money on the development of special purpose software. Parallels are more difficult to find in this case but it seems likely that, where a client can be clearly identified, the person in the client role sees himself or herself as having commitment.

3. The early studies showed that simpler applications of NAT were more likely to succeed than complex ones. For example the integration of cost control with NAT reduced the success rate and even regular updating of networks had a negative influence. Although the evidence is limited it does seem that a very substantial majority of successful expert systems deal with relatively simple domains.

4. It was also found that success with NAT was greater when need for the techniques was identified within the organization (i.e. the contractor in this case) than when their use was imposed upon it by contractual requirements. A loose parallel with this has been seen in the recent very successful work of Stone & Webster inc. of Boston, Massachusetts. This company approaches

expert system applications simply as real world problems that may respond to some form of analytical treatment. They make no prior commitment to the use of expert system technology but look only at the problem. If the solution is wholly or partially an expert system they are pleased; if it is a solution that cannot in any way be described as an expert system they are equally pleased! Their criterion is solely whether the solution does its job effectively in a real world application.

It is too early to assemble good statistical data regarding real world applications of expert systems. However the experience we have gained through the development of our own systems and a survey of other peoples' systems, together with the previous experience of NAT systems reviewed above, does seem to provide useful pointers. We conclude that if the objective is to provide systems for the real world the following advice should be considered:

1. Development should be "client led" with the objective of solving a real world problem, as opposed to a research exercise conceived by enthusiasts. This does not necessarily mean that client and developer should be from different organisations - they could both work in the same company.
2. Senior management should be involved in defining the problem and during development of the system, and should know how the eventual system will actually work.
3. It should be clear what the system is to do, who is going to use it and why it is needed. Preliminary discussions should be held with managers, experts and intended users in order to establish these criteria. The value of such discussions was demonstrated during selection of the MATSEL domain (Appendix 4).
4. Ensure that the system is intended to embody existing human expertise, and has not been conceived with the idea of solving a hitherto intractable problem. Unfortunately the "hype" associated with expert systems has led some individuals to the latter view.
5. Ensure that enthusiastic experts are available. Clearly the experts are more likely to be enthusiastic if discussions are held with them when the domain is selected. Also they will be positively motivated if the system will relieve them of mundane routine work (e.g. BREDAMP - Appendix 1) or helps them to work more effectively (e.g. MATSEL). Also ensure that the experts are

using their domain expertise in their work on a regular basis. We found that the GEC domain experts had not used the required expertise for several years.

6. Correct identification of the individuals who will use the completed expert system is crucial to system development. It is vital that the developers understand the real needs of the users, and try to ensure that they are enthusiastic about the proposed system. There is no point in developing a system if the users are going to refuse to use it. Once again, holding discussions with users at an early stage will help in this respect. Also the nature of the system is important. We note that a piece of conventional software was purchased by GEC but rejected by users because it imposed solutions upon them and entirely removed the decision making role they had previously enjoyed. In many cases expert systems can support decision-making on the part of users rather than eliminating this, and we have particularly tried to achieve this in the CRANES and MATSEL systems. The explanation facilities of an expert system may also help to make the reasoning process more accessible to the user.

7. Keep the system as simple as the domain conditions permit, and avoid excessively complex domains. Some authors have suggested that domain complexity can be estimated by assessing the time it takes a human expert to carry out the task which the system is intended to perform. Tasks which take a few hours at most have been suggested as suitable (Nii 1983, Bobrow et al 1986). We certainly found we could only address a very small proportion of the available knowledge for CRANES in a domain in which an expert would take 3 days to reach a solution. Several of our domains have proved complex even though it might only take an expert, say, 30 minutes to perform the relevant task. The assessment of the time an expert takes is itself difficult. For example when designing boiler tubing an expert may waste considerable time assembling the codes of practice and design manuals he requires, and then finding relevant clauses within these, before starting the actual design.

8. Avoid the danger of techniques looking for applications. At all stages of development look for the most effective solution to the problem, irrespective of whether or not this involves an expert system.

4 SITUATIONS THAT AFFECT THE ELICITATION PROCESS

It is clear that the nature of the situation within which the knowledge is elicited will have a major influence on the knowledge elicitation methods to be selected. Some of the important factors will now be introduced:

1. Is the expert naturally introspective? Some experts will have a clear view of what they believe to be their decision making process - this is particularly true of those who are involved in training or lecturing and therefore are used to expressing their knowledge in a structured form. Other experts may never have been required to express their knowledge in this way, and although highly competent at a task they may find it very difficult to explain what they do in a coherent manner. The expert for BREDAMP was an outstanding example of the first category. On the other hand our expert in the NETWORK domain found it impossible to answer a question such as "what are the symptoms of a modem fault" even though he was eminently capable of recognising such faults in practice.

An expert who has problems accessing his knowledge may be helped by appropriate cues or examples. One method of obtaining knowledge is therefore to watch the expert at work and then ask why he did what he did - here the work situation prompts the expert to recall knowledge. Another method is to generate artificial examples as cues and to ask the expert what he would do in these circumstances. Use of this method to obtain the Bayes factors for BREDAMP is discussed in section 5. A third method is to ask the expert to describe examples which he has worked on in the past.

2. Is the knowledge intuitive? When he is learning his skill an expert may make decisions based on reasoned steps, but as the task becomes more familiar he may be able to make an immediate decision based on how closely a situation resembles something he has seen in the past. Because the decision is now intuitive he no longer uses the step-by-step decision-making process that a knowledge engineer would wish to incorporate in an expert system. For example a medical practitioner may diagnose measles because the patient looks the way people with measles look. We encountered this type of problem while doing the knowledge elicitation for CRANES where the experts could recognise structural steel members that were difficult to position in the building frame, but were unable to quantify the criteria for defining a "difficult" member.

3. The "depth" of knowledge i.e. does it represent deep fundamental knowledge such as that relating to molecular structure or shallow "heuristic" knowledge which is known to hold true but which cannot be explained in terms of fundamental principles. We encountered heuristic knowledge in a number of the domains we investigated.

4. The expert may be involved in making judgements rather than identifying "correct" solutions. For example the BREDAMP expert might decide that dampness resulted from either of two causes, one of which perhaps being more likely than the other. Here the knowledge involves uncertainty, as opposed to being deterministic (where solutions are either true or false). The building dampness and bidding (Appendix 3) domains involved mainly uncertain knowledge, whereas the others involved a mixture of uncertain and deterministic types. Uncertain knowledge presents special elicitation difficulties as discussed in Section 5.11.

5. The attitude of the domain expert or experts is crucial to the knowledge elicitation process and the importance of finding enthusiastic experts and fostering that enthusiasm should not be underestimated. Resistance may occur if a domain expert fears that, by giving up his knowledge, he will weaken his position in his organisation. Unless some incentive can be engineered such an expert is unlikely to provide the basis for a useful system. Organizational resistance may also arise and has been observed in the Community Clubs established in Britain by the Alvey Directorate. For example one club member may provide an expert but then realize that commercially valuable knowledge could be transmitted via the system to a competitor. Another source of difficulty is the sceptical expert who believes that his knowledge is too complex to be modelled by a computer system.

Once the decision has been taken to develop an expert system in a given domain these difficulties can only be tackled in a situation-specific manner. Resistance can be countered by demonstrating that the expert system will have overall benefits to the experts or organisations concerned. For example a system that reduces the workload of a hard-pressed expert by giving users access to his less complex knowledge should be welcomed. A sceptical expert should become more cooperative when he sees a credible prototype system in operation.

We were fortunate to secure cooperative experts for all the systems we developed. The BREDAMP and MATSEL experts were particularly helpful and this must be in part due to the benefits they themselves would gain from these systems. When some of the CRANES experts became bored during the early stages of development, knowledge elicitation became very difficult.

6. The availability of domain experts may be a problem since the skills sought by the knowledge engineer are likely to be in constant use in the service of the expert's employers. Some knowledge elicitation techniques produce results in a shorter time than others.

7. In addition to asking experts how they work, it may also be of value to study the details of problems the experts have already solved. The experts may recount these themselves, or there may be sufficient documentation to allow the knowledge engineer to review this material in his own time - this will help reduce the time required of the expert.

8. Some or all of the required knowledge may be documented. The MATSEL system was largely based on code of practice knowledge, and in other domains text books or manuals may be available. At the very least these will help the knowledge engineer understand the background to the subjects without taking up valuable time on the part of the experts.

9. The expert may use machinary or test equipment in the course of his work. In this case the knowledge engineer will need to see this in use in order to understand the domain knowledge. Part of the GEC project involved construction of a working scaled model of an alternator which enabled us to see the balancing process performed on a working machine.

5 KNOWLEDGE ELICITATION TECHNIQUES

The knowledge elicitation techniques which we have used will now be described. Initially we started with the view that it should be possible to isolate situations (or combinations of factors) that would point to the selection of a single method in given circumstances. Our experience has not borne out this view and, for example, in the MATSEL domain six methods were used at different stages. Thus our advice is that the knowledge engineer should acquire a feel for the alternative methods and should use them flexibly as the position unfolds. This approach has been advocated by other researchers (eg Cordingley - see Diaper 1989).

5.1.1 Unstructured interviews with individuals

The knowledge engineer starts with a blank sheet of paper and asks the domain expert to say what he knows. The emphasis is on getting the expert to explain his knowledge in the way he finds easiest, and the knowledge engineer takes a passive role and interrupts as little as possible. The great merit of this approach is that by taking a passive role the knowledge engineer avoids prejudicing the responses of the domain expert. Thus less obvious points may emerge that can be very important. The method however is time-consuming and requires patience on the part of both expert and knowledge engineer. It is also unlikely to be successful when the knowledge is highly intuitive, and an expert who is not introspective will find it difficult to know how to start or how to explain what he does in a coherent manner. We have found that unstructured interview sessions are generally required at the start of knowledge elicitation to enable the knowledge engineer to get a feel for the domain and to develop a rapport with the domain expert. Also if other experts become involved later in the development process it is advisable to start knowledge elicitation in an unstructured manner in each case.

5.1.2 Unstructured interviews with several experts

Where a number of domain experts are available it may be valuable to involve them all in early knowledge elicitation sessions. The approach is the same as that involved with individuals and again it is to be hoped that by adopting a passive role the knowledge engineer will avoid prejudicing responses. However when a number of experts are involved simultaneously they should respond to each others' comments thus provoking the rapid elicitation of a large quantity of knowledge. Problems may occur if the more senior or extrovert experts dominate the discussion, or if the experts become embroiled in arguments about different approaches to problem solving, but later elicitation sessions can be used to resolve these difficulties.

We found an unstructured session involving six experts gave us a valuable introduction to the CRANES domain, but that subsequent discussions rapidly became more detailed so that it was not possible to involve such a large number of experts simultaneously.

5.2 Structured Interviews

Once the knowledge engineer has formed an overall view of the structure and content of the domain knowledge he will wish to take a more active role in interview sessions so that he can develop knowledge statements suitable for coding. Knowledge acquisition is likely to proceed more rapidly once this structured approach starts. However the knowledge engineer must avoid forcing the structured approach too early since there is a danger that the system structure will become something the knowledge engineer has invented rather than that which the expert uses. The accessibility of the experts is an important issue here. If access is very limited then the structured interview approach may have to be started early irrespective of the risk of prejudicing responses.

The knowledge engineer may gain his understanding of the domain from prior knowledge or from experience with building a similar system with different experts, rather than from unstructured interviews. Reading in the subject area may also help, but in all these cases there is some risk of prejudicing responses.

Structure may be imposed on the interviews in different ways which usually involve sub-dividing the domain. When developing BREDAMP the interviews were structured around the fifteen goals of the system (causes of dampness) with two knowledge acquisition sessions carried out for each goal. The MATSEL system had only one goal (the cost of tubing) and it was more natural to structure interviews using sub-goals such as pressure thickness, welding, costing etc. We note that as knowledge elicitation develops interviews generally progress from being broadly focussed to narrowly focussed. During a broadly focussed interview a knowledge engineer might pose quite general questions such as "What are the parts of a typical boiler system?" whereas during a narrowly focussed interview. "Are there any welding problems associated with carbon steel?" would be more typical.

Other authors report more formal methods of structuring interviews with origins in experimental psychology and we will briefly mention repertory grids and card sorting. A typical repertory grid procedure could start by collecting about ten goals. These would then be presented to the expert in threes and he would be asked to say in what way two of the three are alike and different from a third. All possible combinations of three would be presented, with the objective of discovering precisely how all the goals could be distinguished from one another.

Card sorting involves a set of cards with one object name (typically a goal) written on each. The expert is asked to name an attribute and sort the cards into piles according to different values of that attribute. He is then asked to name a different attribute and re-sort the cards, and so on.

Both these techniques only seem applicable to a classification or diagnosis domain. As such they might have been applied to the building dampness domain which has fifteen goals although they seem ill-suited to the uncertain knowledge in that domain. They seem to have little application in our other domains, and we feel they are likely to be slow and laborious to use. However our most serious criticism is that they would appear obscure to the expert - more like games than a serious attempt to elicit knowledge. Hence we would be reluctant to try these techniques other than in non-critical circumstances since we fear they would alienate the domain expert. This view was shared by others engaged in practical knowledge acquisition who were present at a workshop we attended in 1987. Some early results from experimental work intended to establish the effectiveness of card sorting relative to other techniques was presented at that workshop (Burton and Shadbolt 1987).

A number of variants of structured interviews have been described, of which we favour direct interviewing. Whichever approach is adopted it is most unlikely that knowledge retrieval will be complete because of the incomplete nature of human recall. This problem will be compounded if the knowledge is intuitive so that it is difficult for experts to describe how they make decisions irrespective of the form of questioning adopted. Other forms of knowledge acquisition can help in these circumstances.

5.3 Case histories

An expert often cannot access rules and relationships directly. However his recall can be assisted by providing cues and examples to assist him. One method is to ask him to describe problems he has worked on in the past and also the work he is currently doing. He can then be asked why particular decisions were made. We have found this technique a valuable addition to more direct questioning in several domains, most notably those of CRANES and MATSEL. In particular the structure of MATSEL evolved from a single set of design calculations produced by one of the experts. We advise the knowledge engineer to assess carefully the nature of the cases presented however. One of the MATSEL experts tended to describe the extreme cases which he found most interesting, and these tended to overshadow the more basic problems that the expert system users would be tackling on a regular basis.

We used case histories to gain an overall view of the work of experts, and to reveal fragments of knowledge which experts had overlooked during direct interviews. In some circumstances it may be appropriate to induce rules from these cases.

5.4 Rule induction

There are several computer programs that will induce rules from sets of cases. Of these Expert-Ease is probably the simplest and best known. This program is suited to domains where the knowledge consists of hierarchical knowledge trees with each outcome arrived at by single, as opposed to multiple, paths. It can model uncertainty and we were able to test the rule induction approach on part of the BREDAMP domain knowledge. In order to do this the knowledge engineer and domain expert worked together at a computer. The expert defined some cases which the knowledge engineer entered into Expert-Ease and used to induce rules. The expert then reviewed the rules to identify errors or omissions and tried to add more cases to eliminate these. This process was repeated.

We found some disconcerting problems. One of these was that the natural sequence of questioning which is inherent in a domain is not respected. For example a pair of questions might be:

- Is the pipe a drain?
- Have you performed a drain-test?

If the sequence of these questions is reversed, as it may be when automatic rule-induction is applied, then the confidence of the user will quickly evaporate. Another problem is that, like regression analysis, rule-induction works on cases irrespective of any causal connections. Bramer has made similar, more detailed, criticisms of automated rule induction with respect to the popular ID3 algorithm (Bramer 1987).

Rule induction therefore only appears to be satisfactory for very simple well-defined applications. Our findings suggest that for applications involving only quite modest levels of complexity the rules prepared by induction are unsatisfactory for direct incorporation in the expert system. Despite these shortcomings, attempts to apply rule induction to limited modules of a total application can encourage the expert to consider factors that are not revealed by other methods. This must improve the validity of the knowledge base even if the induced rules are themselves discarded.

It is not, of course, imperative to use the rule-induction program. Manual examination of sets of cases will often indicate relationships that can be coded on an ad hoc basis and Bramer has suggested using only a "semi-automated" form of rule induction. The expert will frequently find it easier to recall his knowledge through the recounting of case histories than through other forms of interview.

It is not imperative to use rule induction on cases acquired from an expert by interview techniques. This method is frequently applied to data sets maintained for record purposes.

5.5 Observation

Another knowledge elicitation technique which may be of particular value when the knowledge is intuitive is observation. Here the expert is asked to explain what he is doing while actually carrying out his job. Typically his comments would be tape recorded for later transcription and analysis. If diagrams or equipment were used a video of what happened could also be required. One form of observation is protocol analysis where the subject only describes his thoughts or impressions but does not attempt to explain them - that is left to the knowledge engineer. Alternatively the expert can be asked why he is taking particular decisions - although if things happen rapidly there may not be time for him to do this.

The authors have very limited experience of the observational method. However it can clearly be a beneficial approach if it provides the initial evidence the knowledge engineer requires in structuring the problem. Where machinery or test equipment is involved it may be essential to understanding the experts' role. In the GEC domain a one metre long scaled working model of an alternator proved very useful in this respect and avoided the need for a knowledge engineer to witness experts working with large scale equipment on site (which would have been very difficult to arrange and extremely expensive). Observation also has an important role as a check that the expert performs his tasks in the way he claims to use during interview sessions, i.e. during interview sessions he may be telling the knowledge engineer how he thinks the job ought to be done rather than how it is done in practice. We undertook a brief observation exercise during the development of CRANES for this reason.

The use of observational methods, and task analysis in particular, has been extensively discussed by other authors (Diaper 1989).

5.6 Paper Models

Our experience is that understanding and organising the knowledge collected in the course of knowledge elicitation can be even more of a problem than the actual elicitation itself. This was particularly true in the CRANES domain where a very large amount of knowledge was collected and it was difficult to keep track of cross-references between discussions held at different times. Also it is valuable to develop the structure of the knowledge-base on paper before starting computer coding. For these reasons we have experimented with paper models (also known as intermediate representations) in the BREDAMP and MATSEL domains, and also in the NETWORK domain on which we are not able to report in detail.

We have found that it can be valuable to discuss the knowledge presented in the paper model with the domain expert, since this can stimulate feedback and hence release of additional knowledge. We have also found that a paper model allows separation of the tasks of knowledge engineer and computer coder. Hence the knowledge engineer prepares a paper model which is a formal statement of the knowledge the system should contain, and the computer coder works from that document. The knowledge engineer can then concentrate on developing a rapport with the experts, and on elicitation of knowledge in the form in which the experts perceive it, without prejudicing that knowledge by imposition of the available computer knowledge representations.

The BREDAMP paper model was of a graphical tree form and was prepared using a CAD draughting package. Part of this paper model is shown in Appendix 1 Figure 1. Meaningful variable names were used and nodes were defined as boolean or uncertainty relationships as appropriate. These diagrams were found valuable for discussions with the BREDAMP expert who had no difficulty understanding the representation used and was on several occasions able to identify missing lines of reasoning. However the same form was also used in the NETWORK fault diagnosis domain and in this case the diagnostic technician who acted as expert was unable to follow the diagrams. This may be because the BREDAMP expert was highly introspective whereas the NETWORK expert was quite the reverse.

The MATSEL paper model was textual rather than graphical, and had a format similiar to that used successfully by Logica in their work for the ARIES Alvey Community Club. A page from this paper model is shown in Appendix 4 Figure 2. The paper model acted as a specification for the system, and included statements of the objectives of the system and warnings and advice which should be made available to the user as well as the knowledge to be implemented. Knowledge was generally stated in the form of production rules with an english-like syntax which was intended to be comprehensible to the domain experts.

In addition to recording the rules themselves, the paper model was also used to document the source for each rule. If a rule was derived from a code of practice then the name of the code and the relevant clause was recorded. Other rules were recorded by the section of the design manual from which they were taken, or by the name of the source expert and the date on which he was interviewed. A key feature was the recording of the status of all items in the paper model by a letter code in the left hand margin. These codes enabled the knowledge engineer to keep a formal record of rules which were finalised, those which had recently been amended and those which were expected to require further discussion with the domain experts etc.

The paper model was revised at frequent intervals and could be presented to the domain experts for review. After a full review of the preliminary version the domain experts could confine their attention solely to items which had been changed (which were marked with appropriate codes). The principal expert reviewed the paper model on two occasions, the first time about a month after commencement of work and the second after three months when the system was nearing completion. Other experts also reviewed the model at various times. The experts seemed to nave no difficulty in following the production rule representation and made useful comments.

The use of paper models has been discussed by other authors (Diaper 1989).

5.7 Iterative prototyping

The difficulties which experts frequently have in recalling their knowledge and the benefits of providing them with appropriate cues or prompts have already been discussed. One particularly useful form of prompt is to show an expert an existing computer system which purports to model his knowledge. Experts who have difficulty recalling knowledge or explaining what they do will frequently find it much easier to say that a computer system has come to an incorrect conclusion. Discussion of why this result is wrong can then lead to an improved system. A typical approach involves development of a rudimentary solution which is repeatedly demonstrated to the expert, improved on the basis of the comments made, and demonstrated again. This process is referred to as iterative prototyping.

A number of approaches are possible, and these can be summarised as follows:

1. A prototype system is available before knowledge acquisition starts. This could have been developed in a closely similar domain using different experts, or it could have been based on the prior knowledge of the knowledge engineer. Knowledge elicitation starts with a demonstration of this prototype to the expert and follows a process of repeated revisions and demonstrations of the prototype in response to modifications suggested by the expert.

2. There is no prior system. The first knowledge elicitation session involves the expert and the knowledge engineer sitting at a computer terminal with access to an expert system shell. The expert is asked to provide a rule relating to his domain and this is entered at the terminal as an embryo knowledge-base. The knowledge engineer demonstrates that a very limited consultation can already be run, and prompts the expert for further rules. Iterative improvement of the system continues as before. We will refer to this approach as "instant prototyping".

3. The knowledge engineer uses interview or other techniques for the early knowledge elicitation He may experiment with different representations for the knowledge but he does not start coding a substantial prototype system until he feels confident that an appropriate representation has been identified. This prototype is then shown to the expert and evolves by a process of iterative prototyping.

4. It is conceivable that prototyping could be ignored altogether. The knowledge engineer would continue interview or other techniques until he felt he had collected all the domain knowledge required. He would then develop a system embodying that knowledge and deliver the completed version to the expert.

The use of a pre-existing system as the focus for knowledge elicitation could be very productive since it would almost certainly provoke criticisms and suggestions from the expert. However it could have the effect of prejudicing both the expert and the knowledge engineer into accepting a form of system which did not correspond to the way in which the expert approached his domain in practice. It could thus divert the expert from some of the subtle, more intuitive knowledge that might be of crucial importance in the operation of the new system.

When we started work on CRANES the CONPLANT system was available for the same domain. The idea of developing CRANES actually arose when two of the building planning experts from Wimpey Construction UK Ltd were shown a demonstration of CONPLANT (which they had no previous involvement with). These experts said that their approach to the selection of materials handling equipment was very different from that of CONPLANT which they criticised for lack of graphics and a failure to cater for the irregular 3-D nature of real-world buildings. CONPLANT therefore served as a valuable stimulus for these experts to become involved in the development of CRANES, but it was not used as a knowledge elicitation tool. We feel that our decision to start a completely new knowledge-base for CRANES was the correct one since the system which was developed is radically different from CONPLANT. On this basis we therefore consider that concentration of early knowledge elicitation sessions on an existing system offers a very real danger prejudicing responses. Also if the knowledge engineer has developed the first prototype on the basis of his own knowledge then there is a serious likelihood that his view of the domain knowledge will be prejudiced.

We have no experience of instant prototyping techniques. However we doubt whether meaningful prototype systems could have been produced by these means during the first knowledge elicitation session for CRANES, MATSEL or the GEC system. In these cases considerable knowledge elicitation effort was required before it became clear what the structure of the system should be. We consider that the pressure the knowledge engineer must be under to add to the embryo system is liable to prejudice responses when the instant prototyping approach is adopted. We also consider there is a danger of the expert seeing the embryo prototype as trivial and therefore becoming prejudiced against the elicitation process. However the technique has some strong advocates, most notably within British Petroleum plc who use rule induction to generate an "instant" prototype (Guilfoyle 1986).

Our approach has been to not demonstrate a prototype system to the expert until knowledge elicitation is fairly advanced. In the CRANES domain we found this demonstration important because it revealed that one of our experts had seriously misunderstood the type of system we were developing, expecting a much higher level of automated data acquisition from site layout drawings than was possible. An early demonstration of a prototype will therefore help to resolve misconceptions about the nature of the system on the part of both expert and knowledge engineer.

More importantly, we have found that demonstration of a suitable prototype system can be a major factor in promoting the enthusiasm of the experts. This was particularly the case with CRANES and the GEC system. Here we observed that early demonstrations of simple elements of the final system were only of limited interest to the experts. However when a substantial prototype was shown which could give useful advice there was a very noticeable increase in expert enthusiasm.

In conclusion our experience of prototyping has therefore been that it always provokes useful and constructive criticisms from experts. We consider that demonstration of trivial prototypes very early is not of substantial benefit, and pressure to produce such systems at an early stage could lead to prejudicing of the knowledge representation used. However we would also warn that if the demonstration is left to a very late stage, as in the fourth development approach suggested, then the experts may feel an anti-climax if the system does not meet their expectations. We therefore suggest that a prototype should be demonstrated to the expert as soon as it can give a significant and plausible piece of advice to a user.

One final comment is that the nature of a system affects the usefulness of demonstration sessions. The BREDAMP expert could quickly recognise when the system had reached an incorrect conclusion for a given set of symptoms. However because substantial computation was required, it was less easy for the MATSEL experts to recognise when that system advised on, say, an incorrect tube thickness. Hence it was necessary to run the latter system for prepared examples, and to check the domain knowledge more fully in paper model form.

The prototyping approach is dealt with extensively by other authors (Diaper 1989).

5.8 Published material

There is a lot of interest in the use of expert systems to guide users in the interpretation of regulations and codes of practice. In this situation it may appear that there should be no problem of human interaction as the views of the human experts should be fully recorded in the published text. However attempts to "computerise" regulations were made before the recent surge of interest in expert systems. These attempts often revealed inconsistency and vagueness which made computerisation difficult (Diaper 1989). Perhaps this should be anticipated since differences between the views of the members of the original committee that drafted the regulations eventually have to be resolved by compromise.

The MATSEL system contained a high proportion of code of practice knowledge and here we found it very important that knowledge elicitation was carried out with experienced design engineers. These engineers could tell us which parts of the codes of practice were important, and also had additional knowledge which was not incorporated in the codes of practice but which was important for good practical design.

In domains where codes of practice or regulations are not involved, it can still be valuable for the knowledge engineer to do some basic reading in advance of knowledge elicitation in order to gain an elementary knowledge of the subject area. However this material should not be allowed to prejudice the knowledge subsequently elicited from the expert.

5.9 Coding by the expert

When the domain expert is also a reasonably competent user of computers it may be possible for him to produce his own expert system without the use of a knowledge engineer as intermediary. This approach is only possible where the expert exhibits a high degree of self-knowledge and is likely to be unsuccessful where the knowledge is largely intuitive. There are problems when people attempt to access their own knowledge, particularly for highly practiced tasks which may be represented procedurally and in consequence be difficult to verbalise (Diaper 1989). The expert may or may not be inclined to use an expert system shell to assist him in his efforts.

5.10 Knowledge acquisition systems

During our work on this project we have become aware of developments in the use of computers to assist in the process of knowledge acquisition. Notable amongst these developments is the AQUINAS system developed by Boeing at Seattle (Boose and Bradshaw 1987). It is hoped that future funding will enable us to study the operational implications of this approach.

5.11 The special problem of uncertainty

The BREDAMP system generated some useful insights into the problems of uncertain knowledge. The domain expert was exceptionally cogent and well motivated. However, it was necessary to attach probabilities to the goals e.g. to conclude that there was a 90% likelihood that the cause of the dampness was rising damp. For this the dependencies between variables are calculated using Bayes theorem for which affirmative and negative factors must be established. While the domain expert could be expected to describe the behaviour of dampness, it was impractical to obtain from him estimates of the BAYES factors. This must be expected when uncertainty is incorporated into an expert system.

To overcome this problem, the key factors relating to each cause (or goal) were first obtained from the domain expert. These were then compiled into tables in the following form and the domain expert was asked to suggest values to replace the question marks:

Factor	Suggested values	
Evidence	A stain	A stain
Height of stain	9 inches	15 inches
Age of building	8 years	9 years
Component wetter inside than out	Yes	Don't know
Positive salts test	Don't know	Yes
Probability of rising damp	?	?

To derive BAYES factors from these data it was sufficient to use an ad hoc approach i.e. a combination of simultaneous equations and trial and error. A better approach would have been some form of regression analysis although it can be difficult to elicit a sufficient number of cases to make this approach possible.

This case illustrates a further point namely confidence limits. At present BREDAMP offers only a set of probabilities for each of the defined causes of dampness. For example:

Rising damp	90%
Rain penetration	27%
Others	less than 5%

The rising damp figure may in fact mean that the probability is in the range 89-91% or it may mean that it is in the range 80-100%. A user would react differently if he had these ranges available.

This extension of the information provided by a system has been mentioned in several contexts, but no actual implementation has so far been identified by the authors.

6. KNOWLEDGE ELICITATION - GENERAL ADVICE

The knowledge engineer must become familiar with the alternative knowledge elicitation methods described in Section 5, and should be prepared to use them flexibly as elicitation develops. There is a good case for using a wide range of techniques since the different approaches are likely to reveal different aspects of an expert's knowledge. The approach which we have evolved in the course of this research can be typified by the following progression:

(i) Unstructured interviews

(ii) Case histories and broadly focussed interviews

(iii) Narrowly focussed interviews, discussion of paper model and demonstration of first prototype

(iv) Discussion of paper model and iterative prototyping

We attach considerable importance to the first demonstration of the prototype system to the domain experts. We believe this to be important as a means of revealing significant misconceptions on the part of either knowledge engineer or experts so that an early demonstration has advantages. However if the knowledge engineer becomes prematurely over-enthusiastic and demonstrates a piece of trivial code the confidence of the experts may be lost. We therefore suggest that the prototype should not be demonstrated until it can give a piece of recognisable and plausible advice to the user. We have found that a substantial prototype of this type can generate valuable enthusiasm on the part of the experts.

A number of practical issues relating to the organization of knowledge elicitation will now be discussed:

6.1 The knowledge elicitation team

It is most important that a good rapport is established between the knowledge engineer and the domain expert. We consider that this will be achieved more easily if the knowledge engineer has some training in the basic principles of the domain. For example our background is in engineering and we believe this has helped us to develop good relationships with the civil and mechanical engineers who have acted as domain experts. In contrast to this we note that during the Alvey work a human sciences researcher was involved in the knowledge elicitation. Because of this background he had to ask the domain experts to explain fundamental concepts such as "phase" and "hertz". Clearly such a lack of fundamental understanding can be damaging to rapport with a busy expert. However it should be possible to ease this problem by appropriate background reading prior to meeting the experts for the first time.

There are also dangers in using knowledge engineers with an extensive prior knowledge of the domain. Such individuals must be careful to avoid distorting the expert's knowledge by introducing their own ideas.

There appear to be advantages in separating the task of knowledge elicitation from that of encoding the information for the computer. This allows the knowledge engineer to concentrate on the knowledge as perceived by the expert and on establishing a good human relationship with the expert. In particular he is less likely to distort the expert's knowledge by imposing an available computer representation on it at a premature stage. We successfully separated the tasks of knowledge engineer and computer coder during the development of BREDAMP, NETWORK and MATSEL, using the paper model as a means of communication between the knowledge engineer and computer coder.

6.2 Organisation of knowledge elicitation sessions

A good relationship is maintained during knowledge elicitation sessions if a small number of individuals are involved - say one or two knowledge engineers and one or two experts. During the development of MATSEL one session involved three knowledge engineers and one expert. The expert found this unacceptable and complained of "an inquisition".

At the start of the development of CRANES two unstructured sessions were held at which six experts were present. This was successful in these early stages but when discussions became more detailed it was not possible for such a large number of experts to be involved and some individuals became bored.

The knowledge engineer must decide whether he wishes to tape record the elicitation sessions. We recorded the interviews for MATSEL and found that transcription of the tapes was excessively time-consuming - for example a two hour tape took twelve hours to transcribe. Generally we have found it satisfactory to keep notes during knowledge elicitation sessions and to transcribe these after the meetings. It is possible that tape recordings could be used to back up the hand written notes, rather than as a basis for full transcription, but we have not found it necessary to do this.

The attention of both knowledge engineers and experts will wane if knowledge acquisition sessions are too long. However very short interviews may not be productive because the participants do not have time to talk in depth about the issues under discussion. We consider that knowledge acquisition sessions should last about 90 minutes, and that sessions longer than two hours should be avoided.

6.3 Conflict between experts

In discussions of knowledge acquisition from a number of experts the issue of conflict between experts is often raised. With the exception of BREDAMP, all the domains which we have investigated have involved more than one expert. Our experience is that although conflicts of opinion do occur, these do not generally cause serious problems. Our experts usually approached problems in a similar manner and differed only in their detailed knowledge. This is probably because each domain involved experts from a single organisation who regularly worked together. Had we used experts from a number of organisations to develop a single system then it is probable that much more fundamental disagreements would have resulted. We have some evidence for this from the development of the CONPLANT and CRANES systems. CONPLANT drew largely on expertise from the construction companies Taylor Woodrow plc and Tarmac plc. Experts from a third construction company, Wimpey Construction UK Ltd. were shown CONPLANT and commented that their approach to the selection of materials handling plant was fundamentally different. These experts then assisted in the development of CRANES.

The first point to make regarding conflicts between experts is that good documentation of the knowledge elicitation process is necessary to enable the knowledge engineer to detect and understand these conflicts in the first place. This necessitates detailed records of each knowledge elicitation session and a systematic approach to collating the information collected into some form of intermediate representation.

Once conflicts have been recognised then we have identified three posssible approaches to resolving these:

(i) Hold joint interviews with the experts who have conflicting views and see if they can reach a consensus.

(ii) Produce a system which will present alternative views to the user and allow him to choose the approach he prefers.

(iii) Adopt the views of the expert who is judged to be the most knowledgeable.

We have used all three methods at different times and selection of the correct approach is clearly very situation-dependent. Solution (iii) seems to raise difficulties since it may put on the knowledge engineer the onus of judging the relative abilities of different experts. However during the development of MATSEL we found this approach particularly appropriate. MATSEL was developed in conjunction with an engineering company in which senior managers and directors were expected to be highly experienced in the design process. When conflicts arose the experts therefore turned to more senior colleagues for advice. Hence in this particular case the experts tended to adopt the third approach listed of their own accord.

7. KNOWLEDGE REPRESENTATION

7.1 Introduction

All the expert systems developed in the course of our work have been built in their final form using commercially available shell software. In two cases some early work was done on the development of shell software in house. The first of these was an interpreter written in Basic for the BID/NO BID knowledge-base. This proved successful as an initial prototype system, and was used to assist knowledge elicitation. The knowledge was then re-coded in the form used by the SAVOIR expert system shell. This re-coding took a little over two weeks and produced a system with a much improved interface and user facilities.

Stewart Hughes Ltd attempted to develop their own shell in lisp at the start of the Alvey project. After one year of effort the shell had many deficiencies and was still limited to a forward chaining inference mechanism. The work was abandoned and the commercial shell Goldworks adopted (it is unlikely that any shell of the sophistication of Goldworks could have been obtained for a PC at the time the project started).

In view of these experiences we consider that most developers should use a commercially available shell rather than attempting to develop their own software. Commercially available shells are evolving rapidly and offer increasingly sophisticated facilities. A developer is unlikely to improve on these facilities unless he has considerable experience in the development of expert systems and in the use of a range of shells. He will also need considerable resources which will have to be justified on commercial grounds. Should all these conditions be met, then development of a shell in house will produce a product tailored more closely to the needs of the developer than would be possible with a commercial system.

Most organisations embarking on an expert system development are therefore likely to adopt a commercially available shell. The shell must be selected with care since it must offer forms of knowledge representation which correspond to the knowledge to be encoded. Where this criterion is met a shell can facilitate very rapid coding of a knowledge-base. For example the coding of BREDAMP and NETWORK in SAVOIR required only nine weeks and four weeks respectively. In contrast the same shell proved inadequate to represent an important aspect of the CRANES knowledge, and coding problems took up a considerable proportion of the eleven month development period allotted to this system.

The following sections summarise some important aspects of knowledge representation which we have encountered in the course of this work.

7.2 Production rules

Production rules have the general form "if premise then conclusion". For example:

IF pressure-part is a 'tube'
AND method-of-manufacture is 'seamless class 1'
THEN tolerance-on-o-d = 0.005 * outside-diameter

Production rules are the most commonly available form of knowledge representation, and were available in all the shells we used. The syntax used varied considerably between the shells.

This form of knowledge representation was used to varying degrees in all the systems we developed. It was the principal form used in NETWORK and proved ideal in this domain. Production rules are likely to be suitable for simple fault diagnosis or selection tasks where uncertain reasoning is not required.

7.3 Bayesian Rules

The SAVOIR shell offered probability statements with the following form:

PROBABILITY rd-dpc-faulty

rd-salt-test-positive	LS 100	LN 0.01
moisture-test-positive	LS 4	LN 0.2
straight-stain	LS 0.1	LN 1.4

PRIOR 0.5

This statement indicates that at the start of any dampness investigation it is known that half of all dampness faults are due to a faulty damp proof course (i.e. prior probability 0.5). The numbers after LS are weighting factors applied to the PRIOR probability of rd-dpc-faulty if each piece of evidence is confirmed true. The numbers after LN are used if the evidence is not true.

Knowledge representation in BREDAMP used bayesian rules extensively in conjunction with some production rules. The system embodied a fault diagnosis task for which uncertain reasoning was required.

Other types of representation are available for uncertain knowledge, most notably certainty factors.

7.4 Focus Control Blocks

The MATSEL system relied heavily on production rules. The system was required to calculate the minimum boiler tube wall thickness required using up to twelve different steels. Most of the rules were the same whichever steel was considered, but in order to simultaneously calculate the tube thickness for a number of steels a simple production rule system would require many of the rules to be stated twelve times, once for each possible steel. The shell ESE used for MATSEL incorporated Focus Control Blocks. All rules were owned by one or more Focus Control Block which offered facilities for defining control text, hierarchical structuring of the rule-base and multiple instantiation of a rule set within a Focus Control Block. In MATSEL the latter facility is used to create a new copy of the design rules each time a new steel is considered, so that one rule set can be applied to all twelve materials in a single consultation. Hence this more powerful knowledge representation facility permits a much more sensible embodiment of the domain knowledge than would have been necessary if a simple production rule shell had been used.

ESE is the only shell in which we have encountered Focus Control Blocks. A much more widely used form of representation is Frames. Frames could have been used for the development of MATSEL as an alternative to the multiple instantiation of rule-sets.

7.5 Frames

A frame describes a class of objects in terms of the attributes which distinguish the members of that class. For example in the GEC system we found it necessary to define frames for the component parts of an alternator - bearings, spans (lengths of shaft between bearings), oalance planes and transducers. A transducer has attributes such as position, type and reading, and the transducer frame contains slots corresponding to these attributes as follows:

Name:

IsA: Transducer

Position:

Type:

Amplitude-of-Reading:

Phase-of-Reading:

Each individual transducer is represented as an instance of this frame, with values entered in the slots as appropriate. A single rule can then be applied to the entire class of transducers. For example:

for all Transducer

if Type: of Transducer is 'displacement'
and Amplitude-of-Reading: of Transducer > Max-allowed-displacement
then list-of-unacceptable-readings includes Name: of transducer

In some applications the programmer will define all the instances of a class and their attributes in the knowledge-base. For example we have recently had involvement with a system to assist in the selection of protective paint treatments. A typical class would be that of primer paints and all the primer paints would be defined by the programmer as instances of the class primer, with details of the attributes of each paint recorded.

For the GEC system the programmer defines only the transducer class frame structure, and instances of this frame are dynamically created at run-time when the user defines the number of transducers on the particular machine he is looking at. We found that the Leonardo shell provided the frame representation facilities we needed for the paint system, but that it did not offer the dynamic instantiation capability which we required for the GEC system. This latter system is now being coded using the Goldworks shell which offers the degree of complexity required. (We note that the Leonardo shell has been developed further since we considered it for the GEC system, and that it now also offers a dynamic instantiation facility).

These experiences illustrate the importance of selecting a shell appropriate to the domain knowledge to be modelled. We consider one of the skills of an experienced knowledge-base builder lies in recognising the structure of the domain knowledge and matching this to an appropriate representation. Once a particular representation has been selected care is needed in choosing the most appropriate shell with the required facilities. For example two suppliers may quite justifiably state that their products offer a frame representation. However one product may permit the dynamic instantiation of frames whereas the other may not.

7.6 Readability of the Knowledge-Base

We consider that it is important to construct knowledge-bases that are easy to understand. If this can be achieved then the domain experts, who will frequently have no computing experience, will more easily be able to read the knowledge-base. The knowledge-base itself can then be used as a focus for knowledge elicitation sessions, and it may be possible for the expert to maintain the knowledge-base himself after the work of the knowledge engineer is completed. This last objective is particularly challenging, since we have found that using currently available shells a high degree of understanding of the shell is usually required to fully appreciate the working of a finished system.

We regard easy-to-understand knowledge-bases as an important future objective for expert system developers. However we can report some recent success with a system to select protective paint treatments. This system was developed using the Leonardo shell and we found that the domain expert was able to understand and check listings of the knowledge-base.

The following examples illustrate differences between the knowledge-base syntax of some shells we have used in this research. The first two state the same knowledge using the syntax of the SAVOIR and Leonardo shells respectively.

(i) CONDITION modem-power-off
NOT lights-showing-on-modem
AND NOT modem-plugged-in

```
(ii)  if lights-showing-on-modem is 'no'
      and modem-plugged-in is 'no'
      then fault is 'modem-power-off'
```

A second example shows knowledge similar to that used in the example of Section 7.5 represented using the syntax of Goldworks and Leonardo respectively:

```
(i)   IF (INSTANCE ?TRANS IS TRANSDUCER
            WITH TYPE DISPLACEMENT
            WITH AMPLITUDE-OF-READING ?AMP)
      AND
            (MAX-ALLOWED-DISPLACEMENT ?MAX)
      AND
            (> ?AMP ?MAX)
      THEN
            (INSTANCE ?TRANS IS TRANSDUCER WITH VALUE UNACCEPTABLE)

(ii)  for all Transducer
            if Type: of Transducer is 'displacement'
            and Amplitude-of-Reading: of Transducer
                  > Max-allowed-displacement
            then Value: of Transducer is 'unacceptable'
```

The intention of these examples is to illustrate the considerable variation in readability between the knowledge-bases of different shells. This cannot be the only criterion for shell selection - for example the knowledge representation in Goldworks is very flexible even though the rule syntax is obscure to the layman.

7.7 Interfaces to External code

We have frequently found it necessary to incorporate calls to external procedural code into the expert systems we have developed. These have been used to perform the following operations:

NETWORK	-	accessing a simple data-base
MATSEL	-	reading data from file
CRANES	-	graphics display and accessing data files
Rolls Royce	-	graphics display and image processing
GEC system	-	data analysis, graphics and communications

Clearly such external interface capabilities will be essential to many engineering applications of expert systems. The four shells we have used have all had external interface capabilities, but we have noticed that the quality of facilities provided is much better in the more recently released shells. SAVOIR and ESE could only interface to functions which returned a single value whereas there is no restriction of this type with Leonardo and Goldworks. As well as permitting external calls Leonardo also provides an internal language for the development of simple procedures, and Goldworks allows Lisp functions to be called in the same manner. We have successfully interfaced Goldworks, which is a Lisp system, to software written in Fortran and C using facilities developed by Stewart Hughes Ltd.

8. CONCLUSIONS

Knowledge acquisition has on numerous occasions been described as a "bottleneck" in expert systems development. We have assembled considerable experience of expert systems development both from developing eight systems ourselves and from a survey of 70 "real world" systems developed by other people. This experience shows that knowledge acquisition may be difficult and complex. However our findings indicate that the key to successful development of real world expert systems lies not at the knowledge acquisition stage, but rather in the organisational circumstances in which the application is selected and undertaken. We have concluded that if the objective is to provide systems for the real world then the advice which follows should be considered. These items are discussed at greater length in Section 3 of this report:

1. The development should be "client led" with the objective of solving a real world problem, as opposed to a research exercise conceived by enthusiasts.

2. Senior management should be involved in defining the problem and during development of the system, and should know how the eventual system will actually work.

3. It must be clear what the system is to do, who is going to use it and why it is needed. Preliminary discussions should be held with managers, experts and intended users in order to establish these criteria.

4. Ensure that the system will embody existing human expertise, and has not been conceived with the idea of solving a hitherto intractable problem.

5. Ensure that enthusiastic experts are available. Also ensure that the experts are using their domain expertise in their work on a regular basis.

6. It is vital that the developers understand the real needs of the users, and try to ensure they are enthusiastic about the proposed system. There is no point in developing a system if the users are going to refuse to use it.

7. Simple applications are more likely to be successful than complex ones. Keep the system as simple as the domain *conditions permit.*

8. Avoid the danger of techniques looking for applications. At all stages of development look for the most effective solution to the problem, irrespective of whether or not this involves an expert system.

We started with the view that it should be possible to isolate situations (or combinations of factors) that would point to the selection of a single method of knowledge elicitation in given circumstances. Experience has not borne out this view and we consider that a number of methods will be used at different stages in the development of an expert system. The knowledge engineer should acquire a feel for the alternative methods and should use them flexibly as development proceeds. The following comments summarise our experience of different knowledge elicitation methods, as presented in Section 5:

1. Unstructured interviews may be time-consuming but avoid prejudicing the responses of the domain expert.

2. Unstructured interview sessions with simultaneous involvement of a number of experts may yield a large quantity of knowledge at the start of elicitation.

3. Knowledge elicitation is likely to proceed more quickly once structured interviews start but there is a risk of prejudicing responses.

4. We have found elicitation of case histories to be a valuable technique and this provides cues which assist experts to recall their knowledge.

5. Rules prepared by induction are unlikely to be satisfactory for direct incorporation into an expert system. However the technique may encourage an expert to consider factors that are not revealed by other methods.

6. Observation methods may be of particular value where the knowledge is intuitive. Also they can be used to check that the expert performs his tasks in the way he claims to use during interview sessions (i.e. during interviews he may be telling the knowledge engineer how he thinks the job ought to be done rather than how it is done in practice).

7. Understanding and organising the knowledge collected in the course of knowledge elicitation can be even more of a problem than the actual elicitation itself. A paper model assists in this task. We have used paper models as knowledge tools which act as a focus for discussions with domain experts. Paper models also facilitate separation of the tasks of knowledge engineer and computer coder.

8. The demonstration of prototype systems is valuable as a means of revealing misconceptions about the nature of the system under development and, more importantly, as an effective stimulus to the enthusiasm of the domain experts. We believe that the demonstration of a trivial prototype could be counter-productive, and therefore recommend that the prototype is not shown to the experts until it can give a piece of significant and plausible advice to the user. Thereafter iterative prototyping provides a valuable knowledge elicitation technique.

9. Knowledge may be obtained from published material. During the development of the MATSEL system using code of practice knowledge, human experts were found invaluable to explain how that knowledge was used in practice, and to provide additional practical knowledge not embodied in the codes.

10. Where a bayesian representation is used to represent uncertain knowledge, elicitation of the uncertainty parameters is best done by examples.

The following points summarise advice presented in Section 6 which may be of use to those planning knowledge elicitation work:

1. The approach to knowledge elicitation which we have evolved in the course of this work can be typified by the following progression:

   - o unstructured interviews
   - o case histories and broadly focussed interviews
   - o narrowly focussed interviews, discussion of paper model and demonstration of first prototype
   - o discussion of paper model and iterative prototyping

2. We consider that a better rapport may be developed with the experts if the knowledge engineer is familiar with some of the basic principles of the expert's domain.

3. It may be beneficial to separate the tasks of knowledge engineer and computer coder since this allows the knowledge engineer to concentrate on developing a rapport with the domain expert.

4. Knowledge *elicitation sessions should* ideally involve one or two knowledge engineers and one or two experts. More experts may be involved in brainstorming sessions at the start of knowledge elicitation.

5. Transcription of tape recordings of knowledge elicitation sessions is very time-consuming. We have found it satisfactory to rely on hand written notes which are transcribed after each session.

6. Knowledge elicitation sessions should last about 90 minutes.

7. We have not found conflicts between experts a major problem. Where these have occurred we have used one or more of the following approaches:

- o Hold joint interviews with the experts who have conflicting views and see if they can reach a consensus.
- o Produce a system which will present both conflicting views to the user and allow him to choose the approach he prefers.
- o Adopt the views of the expert who is judged to be the most knowledgeable. During the development of MATSEL our experts adopted this approach themselves by turning to more senior colleagues for advice when conflicts arose.

Skill and experience is required when selecting the best computer representation for the domain knowledge. We offer the following advice which is summarised from Section 7:

1. Expert system developers should not consider writing their own shell software unless they have considerable previous experience in the development of expert systems using a range of commercially available shells.
2. Commercially available shells offer a variety of knowledge representation forms. It is very important that the shell selected is capable of representing the domain knowledge - coding may be very slow or impossible if this is not the case.
3. The clarity of knowledge statements varies considerably between shells. English-like statements should be used wherever possible so that the knowledge-base can be used in discussions with the domain expert. In some cases he may then be able to maintain the knowledge-base after the involvement of the knowledge engineers has ceased.
4. Most of the systems we have developed have involved interfaces to procedural code for tasks such as reading from and writing to files, creating graphic displays and complex computation. Good external interfaces are clearly very important to engineering applications of expert systems.

9. REFERENCES

Allwood, R J., Stewart, D.J., Hinde, C., and Negus, B., "Evaluation of expert system shell programs for construction industry applications", Department of Civil Engineering, University of Technology, Loughborough, 1985.

Arditi, D., "An investigation into the behavioural and technical factors affecting success in the use of network analysis in the construction industry of Great Britain", PhD thesis, Loughborough University of Technology, 1973.

Bobrow, D.G., Mittal S., and Stefik M.J., "Expert systems: perils and promise", Communications of the ACM, Vol 29, No. 9, Sept. 1986, pp880-894.

Boose, J. and Bradshaw, J.M., "AQUINAS: A knowledge acquisition workbench for building knowledge-based systems", Proc. of first European Workshop on Knowledge Acquisition for Knowledge-Based Systems, Reading, Sept. 1987, session A6.

Bramer, M.A., "Automatic induction of rules from examples: a critical analysis of the ID3 family of rule induction systems", Proc of the first European Workshop on Knowledge Acquisition for Knowledge-Based Systems, Reading University, UK, Sept. 1987, session F2.

British Standards Institution, "BS1113: 1985 British Standard Specification for Design and Manufacture of Water-tube Steam Generating Plant (including superheaters, reheaters and steel tube economizers)".

British Standards Institution, "BS3059: 1978 Specification for Steel Boiler and Superheater Tubes".

British Standards Institution, "BS3600: 1976 Specification for Dimensions and Masses per unit length of Welded and Seamless Steel Pipes and Tubes for Pressure Purposes".

Burton, M. and Shadbolt, N., "A formal evaluation of knowledge elicitation techniques for expert systems: domain 1", Proc. of SERC workshop on knowledge acquisition for engineering applications, SERC Computing Facilities Committee, June 1987.

Diaper, D. et al, "Knowledge elicitation: principles, techniques and applications", Ellis Horwood, to be published Feb 1989.

Guilfoyle, C., "Ten minutes to lay the foundations", Expert Systems User, August 1986, pp16-19.

Harries, T. W. and Trimble, E.G., "Factors which affect success in network applications", Fourth Internet Congress, Paris 1974.

Nii, H. P., Quoted in Feigenbaum, E. A. and McCorduck, P., "The fifth generation", 1983, p 81.

Welbank, M., "A review of knowledge acquisition techniques for expert systems", British Telecom Research Laboratories, 1983.

Wijesundera, D.A. and Harris, F.C., "The integration of an expert system into the construction planning process", Proc. of 2nd Int. Conf. on Civil and Structural Engineering Computing, 1985 2, pp399-405.

APPENDIX 1 - BREDAMP

The development of the BREDAMP system for diagnosing the cause of dampness in buildings is described in detail in a paper "Building Dampness: Diagnosing the Causes" by R J Allwood, M R Shaw, J L Smith, D J Stewart, and E G Trimble, which was published in Building Research and Practice volume 16 no. 1 1988. The following is a summary of the issues which are relevant to our research for CERL. Copies of the paper are available on request.

THE BREDAMP SYSTEM

The Building Research Establishment is an organisation funded by the UK government to carry out research and development in the building and construction fields. Advice is available from the BRE Advisory Service (BRAS), and one of the many subjects on which advice may be obtained is in diagnosing the cause of dampness problems occurring in a building. The objective of the BREDAMP system described here is to diagnose the cause of building dampness using knowledge elicited from the principal BRAS dampness expert. Professor Trimble was commissioned by BRE to develop this system. More recently he was commissioned to undertake an evaluation of the system with potential user organisations in order to assess the possibilities and implications of commercial exploitation of the system.

DOMAIN SELECTION

The number of dampness experts within BRAS is small, and these individuals deal with numerous routine enquiries which considerably restrict the time they have available to spend on the more complex problems which are presented. A primary reason for developing BREDAMP was to obtain an expert system which would enable non-specialists to give first level advice to routine enquiries, thereby freeing the specialists to tackle the interesting and demanding problems more befitting their skills.

It was also hoped that such an expert system could be used as a training aid within BRAS. The system would enable newcomers to the advisory service to gain experience of the questions which have to be asked for any particular problem without having to formulate those questions in the light of site experience. A facility for asking the system to justify its' reasoning would provide a foundation for asking the questions and hence some understanding of why they were asked, rather than doing it by rote.

BREDAMP might therefore be expected to act as an archive for dampness diagnosis knowledge, thereby providing a means of capturing and storing some limited, but very valuable, expertise of the current BRAS staff. This would help to alleviate the loss of human expertise which occurs when staff leave or retire from BRAS.

The diagnosis of dampness was also representative of many defect diagnosis problems, and lessons learnt in the development of an expert system in this subject area could be readily applied elsewhere. Furthermore diagnosis has been the most popular application of expert systems, and is well suited for the goal-directed reasoning mechanisms provided by most low-cost micro-computer based expert system shells. A micro-computer based system might have commercial potential, and sale to, say, consulting architects and surveyors would be in line with the BRE desire to sponsor technology transfer to industry.

Finally the principal BRE dampness expert was particularly enthusiastic about the project, so that cooperation and commitment of time on the part of the domain expert was not a problem.

KNOWLEDGE ACQUISITION

The structure of the system

It was recognised at the outset that there were fourteen possible causes of dampness and that these could be selected by the application of appropriate rules. The goals of the system were therefore these fourteen pre-defined diagnoses. One further goal of condensation on pipes emerged during the knowledge acquisition.

It was also recognized that the diagnosis would have to yield probabilities for each goal since the symptoms of dampness often lead to the cause lying between two or three possibilities.

The methodology adopted

During the development of BREDAMP the tasks of knowledge engineer and computer coder were carried out by separate individuals. This was successful, and has potential advantages since it enables the knowledge engineer to concentrate on developing a rapport with the expert. He is also able to record the knowledge as the expert perceives it without being influenced by the forms of knowledge representation which may be available for coding the system.

The domain expert was extremely cogent about the subject of dampness in buildings. There was no difficulty in getting him to provide the necessary information, and he was keen to see his expertise represented in a form that would enable it to be used directly by others.

The principal investigator acted as knowledge engineer for the BREDAMP system. For each of the fifteen dampness causes at least two interviews were held, although for convenience some of the simpler causes were taken in groups. The first interview of each pair was unstructured and sought to ascertain, for the chosen cause, the factors that confirm or deny the possibility of that cause being the source of trouble in a building. For example it was found that for rising damp:

- occurrence is limited to ground floors and basements
- the evidence is a stain (in a band) at the bottom of a wall
- the dampness grows gradually at no more than 32mm per year
- the inside of the affected component is wetter than the surface

The second interview was more structured. In advance the knowledge engineer drew up a table of representative conditions such as that shown below:

Case No.	1	2
Stain present	Yes	Yes
Height of stain	9 inches	12 inches
Age of building	8 years	8 years
Inside wetter than outside	Don't know	Yes
Probability of rising damp	?	?

The knowledge engineer first checked that he had properly interpreted the expert's information about relevant factors and then asked for an assessment of the probability of dampness being due to this particular cause for each case.

By considering many combinations of the symptoms, a sufficiently large set of these artificial case studies was generated to cover a full range of high or low probabilities for the cause under examination. This enabled the appropriate factors to be calculated for the uncertainty knowledge statements. For more complex causes it was natural to break down the overall problem into sub-goals. For example, rain penetration is clearly two sub-problems of penetration through a roof or wall. Penetration through a roof could be further sub-divided according to the type of roof covering.

Once the relationships for a particular goal (or cause) had been obtained they were coded and a self-contained sub-system was developed. Each sub-system was demonstrated to the expert, who was invited to comment on the suitability of the questioning and the appropriateness of the assessed probability. This enabled each cause to be "fine-tuned". Once all sub-models were available they were merged to form the total system.

Paper model

The process of eliciting knowledge and coding it into a reliable system poses many difficulties. The expert is unlikely to express his knowledge in a form which resembles the coding, the knowledge engineer may not interpret and record the expert's ideas correctly, the coding may be at fault and the expert may overlook some facets of his own knowledge.

In order to assist the knowledge acquisition process a detailed graphical paper model or knowledge diagram was created using a CAD drafting package. Part of this is shown in Figure 1. Features of the diagram are that the goal is on the left, the variable names are made meaningful and the logical relationships between variables are indicated by AND, OR, BAYES, MAX or MIN, the last three indicating uncertainty operations. The truth of each goal is ascertained by chaining backwards from the questions on the right-hand side, but where a rectangular box appears the conditions in that box must be true before the backward chaining search goes down any branch.

The value of these diagrams was twofold. Firstly, they enabled the coding to be discussed with the expert in a detailed manner, providing a valuable check on the whole process of converting his knowledge to coding statements. Secondly, the ability of the eye to scan rapidly across the diagram provided an opportunity for the expert to see the scope of the knowledge he had expressed. This led on several occasions to the identification of missing lines of reasoning.

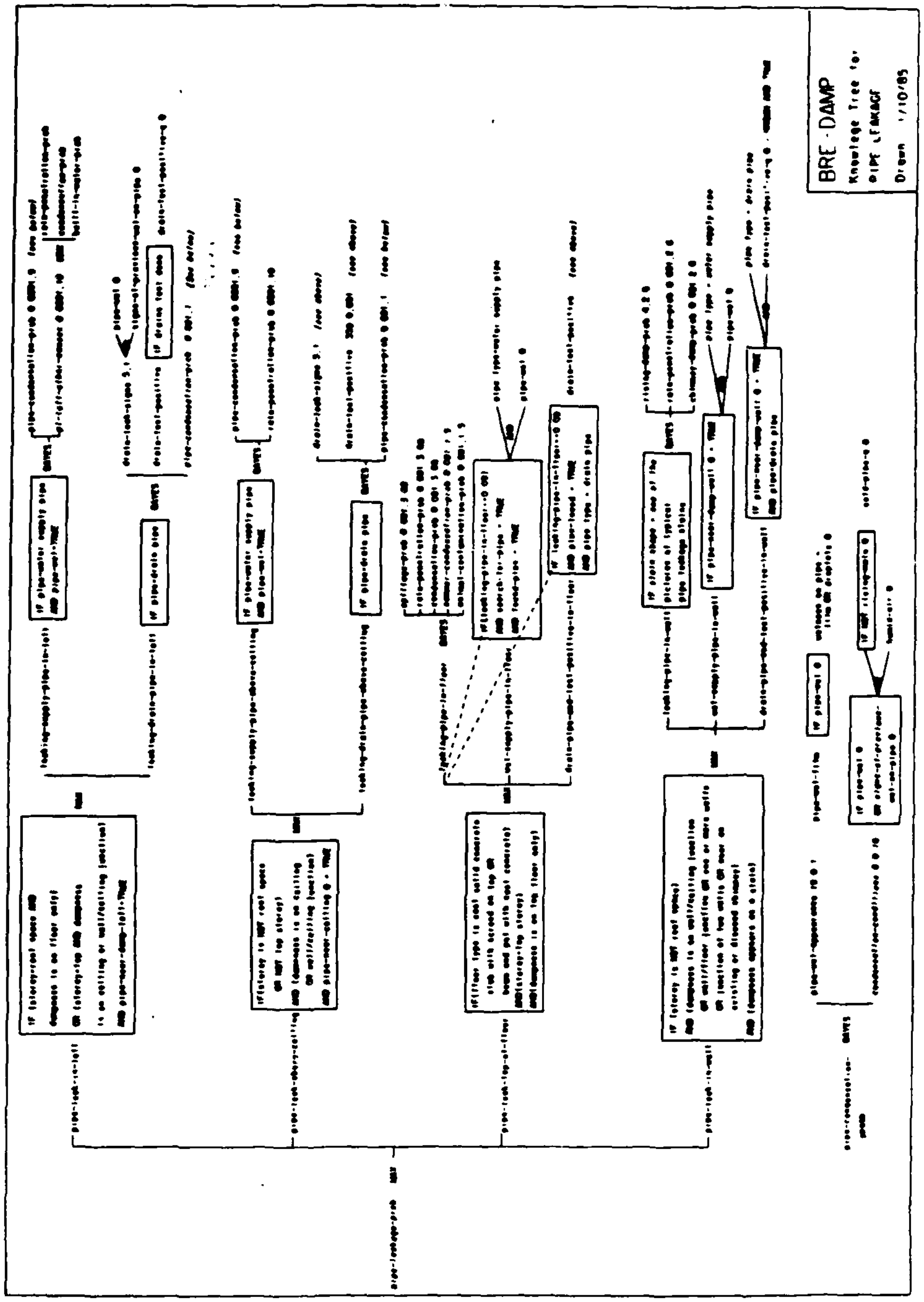

Figure 1 - A page from the BREDAMP paper model

Rule induction

Subsequent to the development of BREDAMP some of the domain knowledge has been used to test the technique of rule induction as a method of knowledge acquisition. In order to do this the domain expert sat at a computer with a knowledge engineer who was familiar with the Expert-Ease rule induction package. The domain expert was asked to provide details of examples where the dampness resulted from a given cause. The knowledge engineer entered these examples and ran Expert-Ease. The domain expert was then asked to examine the rules induced. If these were not satisfactory he was asked to define one or more additional examples. This process was then repeated.

The results from this exercise indicated that the induced rules were seldom satisfactory. For example when rules were induced relating to dampness caused by leaking pipes the resulting consultation should have contained the following questions:

(i) Is the pipe a drain? (yes)

(ii) Have you performed a drain test?

In fact these questions were asked in the reverse order, giving the appearance of an illogical consultation.

On the other hand the process of using the rule induction program forced the expert to define variables and cases that had not emerged in the normal interview process. We therefore consider that, although the rules induced may be of little direct use, rule induction may have the potential to play an important role in prompting the release of knowledge. This might prove most useful in "second stage" elicitation. Thus initial domain parameters might be established through conventional means (via interviews, observation etc) and Expert-Ease used to fine-tune these initial parameters and to identify gaps in the knowledge base. Expert-Ease would be suited to this role in domains where the knowledge consisted of hierarchical knowledge trees with each outcome arrived at by single rather than multiple paths.

KNOWLEDGE REPRESENTATION

Shell Selection

The Savoir expert system shell developed by ISI Ltd was selected for the coding of BREDAMP after a careful survey of available software. The reasons for this choice were as follows:

- (i) BRE had built a simple prototype of BREDAMP using MicroExpert, an earlier ISI shell, and this experience would be useful when using Savoir.
- (ii) Savoir offered reasoning for uncertain conditions.
- (iii) External Pascal routines could be linked to Savoir - this allowed BRE scope to develop special user display software.
- (iv) Savoir provided good facilities for control of the inference process, and for generating report and display output.
- (v) To facilitate possible distribution of the system, the system could run on an IBM PC XT computer.

The following important requirements were recognised before coding commenced and also influenced the choice of shell:

- (i) the system should always work through a short set of preliminary questions before investigating likely causes of dampness individually.
- (ii) all possible causes should be examined.

Both requirements were easily implemented using Savoir. The second represented a cautious approach to the use of expert systems - we have coded BREDAMP so that it will drop the investigation of a cause when the related probability falls to a very low value, but we have not stopped the investigations of remaining causes when the probability of one is found to be very high. The system will continue checking the remaining causes.

A decision related to this was that the order in which causes of dampness would be investigated was to be derived from the statistics of the BRE

Advisory Service. Examination of several hundred site investigation reports yielded an order of likelihood for the goals and a prior probability for each goal.

To complete the overall plan it was agreed that the system should provide the user with help at all stages and should display (and optionally print) a report on each cause. Savoir provides help facilities such as explanations and amplifications of questions, but extra facilities were added to allow re-starts, exits and a display of the current status of all goals.

Coding

The system was written and, wherever possible, tested in separate modules corresponding to the initial preliminary questions and the investigation of each goal. One strength of expert systems is their ability to automatically find the cross-connections when a particular symptom occurs in several knowledge statements relating to separate goals. Hence it was sometimes necessary to test some goals together, but the modularity was still maintained in the source form of the knowledge.

Savoir provides knowledge statements and question statements. Both can have preconditions which, if satisfied, allow the knowledge to be processed or the question asked and, if not satisfied, provide alternatives for the result. Results are stored in variables just as in a programming language. There are four types of variable, of which the most important are designated CONDITION and PROBABILITY. The former can have the value TRUE/FALSE/UNKNOWN, the second stores two values of the probability. These are the maxium and minimum possible values attainable by a variable according to the range of possible answers to as yet unasked questions. As the system works through its questions, these values are updated according to the user's answers, converging when all related questions have been answered.

The two forms of knowledge statement allow rules or uncertain expressions between variables - typically between question variables and sub-goals, or between sub-goal variables and the final goals. The uncertain form uses the Bayes theorem to allow for the effect of evidence of various symptoms to be weighted together. The figure below shows one example of a sub-goal of rising damp due to a faulty damp proof course:

PROBABILITY rd_dpc_faulty					
rd_salt_test_positive	LS	100	LN	0.01
moisture_test_positive	LS	4	LN	0.2
straight_stain	LS	0.1	LN	1.4
PRIOR 0.5					

The numbers after LS are weighting factors applied to the PRIOR probability of this sub-goal if each piece of evidence is confirmed as true. The numbers after LN are used if the evidence is not true.

The weighting factors were derived from the artificial case studies generated during the knowledge acquisition process. These factors ensure that the expert system gives the same final probability to the overall cause i.e. rising damp in this case, as the expert did for all the combinations of evidence considered in the case studies.

We consider that the probabilities calculated for the individual causes of dampness by the system are unsatisfactory in that they provide no measure of confidence in the results. For example a 90% probability that dampness results from a given cause could mean that the probability lies in either of the ranges 80-100% or 89-91%. The user might respond differently to these sets of information if they were presented.

DEVELOPMENT EFFORT

The knowledge acquisition and overall planning of BREDAMP took 30 man days of effort including the time of the expert. The coding and testing took 45 man days. The knowledge-base contains approximately 4000 lines of text but much of this is for display purposes. The knowledge is represented by 143 questions and 171 rules expressed as either CONDITION or PROBABILITY statements.

CONCLUSIONS

- We regard the development of BREDAMP as having been very successful. We attribute this to the "client driven" nature of the system, to careful consideration prior to development of what the domain should be and what the system should do, and to the involvement of an exceptionally articulate and enthusiastic domain expert.

- The concept of an expert system as a "line of defence" for human experts seems to have great potential.

- The goals of the system were identified at an early stage and were used to structure the knowledge acquisition process.

- A graphical paper model was of value to the knowledge acquisition process.

- Uncertain knowledge was successfully elicited by asking the expert to assess the probability of a given form of dampness being present in each of a set of artificial cases.

- Uncertain rules were checked and refined by iterative prototyping.

- Testing of rule induction in this domain has indicated that it may have a role to play in prompting the release of knowledge but that the actual rules induced should be treated with caution.

- The Savoir shell proved appropriate for representing knowledge in this diagnosis domain. This shell offers production rules and uncertainty statements, and has good facilities for controlling the order in which goals are investigated.

APPENDIX 2 - CRANES

Details of the CRANES system are contained in a paper "Cranes - a rule-based assistant with graphics for construction planning engineers" by C N Cooper which was presented at the 3rd International Conference on Civil and Structural Engineering Computing held in London in September 1987. This appendix mainly comprises a shortened version of the paper. Copies of the paper are available on request.

THE CRANES SYSTEM

Selection of the cranes and other materials handling equipment to be used on a construction site is generally carried out by highly experienced engineers in the planning department of a contracting organisation. In 1985 this domain was investigated by development of the CONPLANT system in the Department of Civil Engineering at Loughborough University (Wijesundera and Harris 1985). CONPLANT advised on the selection of items of plant for handling materials during the construction of a concrete framed building. The authors of CONPLANT reported that this early system was limited by a lack of production information appertaining to plant and labour on construction sites and they also recognised that a graphics facility was required.

CONPLANT was subsequently demonstrated to planning engineers from Wimpey Construction UK Ltd who clearly had a different approach to crane selection from the domain experts interviewed during the development of CONPLANT. We therefore invited Wimpey Construction UK Ltd to participate with us in the further development of the CONPLANT work and to act as though they were our client. We consider this "pseudo-client" role to have been important as it ensured that only realistic knowledge was assembled. The first stricture that Wimpey's imposed in this capacity was that the system should be capable of treating buildings as irregular solids and that the user should be able to see the physical relationships between the structure to be built and the cranes to be used. This was a major factor as it required us to develop a graphics module.

The ultimate objective of the work with Wimpey Construction UK Ltd was to develop a system which could be used as a training aid for junior engineers in a construction planning department.

THE DEVELOPMENT OF CRANES

Since Wimpey Construction UK Ltd had no involvement with the development of CONPLANT and appeared to use a different approach to that modelled by the earlier system, the decision was taken to construct a completely new knowledge-base for CRANES which would be specific to Wimpey expertise.

Knowledge acquisition sessions were very productive and it soon became clear that structuring and representation of the knowledge would be a major problem. The sub-domain of selection of tower cranes for multi-storey construction sites was therefore chosen in order to provide a more limited domain on which to concentrate initial development.

The Savoir knowledge-based system shell was chosen for representation of the crane selection knowledge. Adoption of this shell was the result of a comparative evaluation of expert system shells for construction industry applications previously carried out in the Department of Civil Engineering at Loughborough (Allwood et al 1985). Specific features which led to the adoption of Savoir included availability on an IBM Personal Computer and the ability to interface with procedures written in a procedural programming language (Pascal).

The experts interviewed adopted two main criteria for deciding on the number of cranes required as follows:

(i) all parts of the building and material off-loading areas must be overswung by one or more cranes and all loads must be capable of being lifted by one or more of the cranes installed.

(ii) the combination of cranes selected must be capable of lifting materials at a rate to match the planned speed of construction.

These requirements are processed in two modules of the CRANES system which are referred to as the Graphics Module and the Hook-Time Module respectively. At the start of a consultation a third module, known as the Preliminary Module, is available. This is primarily intended for a novice user and will advise which type of crane (i.e. tower, crawler or mobile crane) is likely to be most appropriate for his application. The user is also given advice as to how the total contract period should be allocated between construction of foundations, construction of the superstructure and installation of finishings and fittings.

THE GRAPHICS MODULE

The graphics module of CRANES is mainly a procedural program written in Prospero Pascal with the Prospect graphics toolkit and the GSX graphics operating system extension used to create graphic displays. The Pascal code is compiled with the Savoir expert system shell so that data can be directly transferred between the procedural graphics routines and the Savoir rule-base. Examples of expert systems which interface with graphics facilities are still relatively rare.

In order to use CRANES on a new project a user is required to digitise the building plan and site boundaries. The building layout is represented as a series of polygons each with a defined constant height. In addition the site is divided into zones on the basis of the maximum crane load in each zone. The area covered by load zones includes material off-loading and storage areas as well as the building plan.

During a consultation with CRANES the user is asked to specify crane centres on a display of the building plan by moving a cursor to appropriate points, and to define the crane jib lengths required. The system uses the digitised load zone information to determine the minimum lifting requirements of each crane as it is defined. A search is then made of a data-base of available tower cranes to determine whether the size and lifting requirements the user has requested can be satisfied. If these requirements cannot be satisfied then the user must adjust the location or jib length of the crane he has selected. He is prompted to continue adding cranes to the layout plan until all the load zones are covered.

When program control returns to the rule-base from the graphics module details of the cranes selected are transferred. On the basis of these data CRANES uses rules to calculate mast heights for the cranes selected using appropriate criteria to avoid jib-jib or jib-mast collision situations where cranes overlap. If the configuration is such that crane collisions cannot be avoided then the user is advised of the problem and returned to the graphics display so that he can amend his choice accordingly.

THE HOOK TIME MODULE

The combination of cranes selected for a project must not only be physically capable of lifting all the loads required for construction but must also be capable of lifting materials at a rate sufficient to match the speed of construction. This criterion is checked by calculating the "hook time" required for the structure i.e. the total number of hours of crane time required to perform all the materials handling tasks during construction. From a knowledge of the hook time requirement and the available construction period the minimum number of cranes required can be calculated. This value is generally not an integer.

The CRANES system makes extensive use of the Savoir amplification option, which enables a user to request help in answering system questions. Much of the expertise embodied in the hook time module is made available to the user in this way. Via this option the user can request advice on typical values for unit lifting rates of materials, crane efficiency, length of working week etc and also advice on factors which could lead to a different value being adopted. The expert knowledge of appropriate unit hook time allowances for different materials which is embodied in these hook time texts is extensive and represents one of the main strengths of the system. Considerable knowledge acquisition effort was expended in order to assemble the information presented.

For large in-situ concrete elements of the structure, notably slabs, placing can either be by crane and concrete skip or by pumping. The goals of the hook time module are therefore two crane hook requirement values, one calculated on the basis that all materials are moved by crane and the second calculated on the assumption that wherever possible concrete will be pumped.

OPERATION OF CRANES

A central objective of the system is to check that the crane requirements determined from the graphics module (using site coverage criteria) are compatible with those determined from the hook time module (using crane productivity criteria). For example the user might have selected two cranes to cover the site whereas only one would be sufficient to move the necessary materials within the construction period. The basic assumption behind the method of working of the domain experts, at least in the early stages of crane selection, is that the number of tower cranes installed should be kept to a minimum. The user is therefore advised of ways by which he can both keep the number of cranes required to a minimum and maximise the use of the machines installed. A transcript of this final stage of a consultation is shown in full in the paper attached. Each time the user changes his crane selection in response to system advice that advice is reassessed. Hence the system helps the user to progressively improve his crane selection.

CRANES has been developed as a helpful assistant to the user and therefore relies on the user to generate solutions (i.e. crane layouts) which the system criticises. The user is always free to ignore the advice given by the system if he knows of important criteria that CRANES does not address. This form of system seems to be a more realistic objective than one which automatically generates "correct" solutions. The design of CRANES recognises that there is more than one acceptable solution, and that decisions required in order to generate a solution may be subjective and influenced by factors which the system has not been coded to take into account.

KNOWLEDGE ACQUISITION

Knowledge acquisition was carried out using interview techniques and iterative prototyping. In general two knowledge engineers were present for each session. Hand-written notes were taken and these were later transcribed in detail.

The domain experts were all experienced engineers from the Building Planning Department of Wimpey Construction UK Ltd. In the early stages of knowledge elicitation up to six experts were successfully interviewed simultaneously in unstructured sessions. However as interviews became more structured it was not possible to involve such a large number of participants. Hence later sessions were conducted with only the two most senior experts present.

In all some 20 hours of knowledge acquisition was carried out over eight sessions. The authors believe it would have been better to have used a larger number of sessions of between one and two hours duration. When sessions were longer than two hours concentration on both sides started to wane.

The knowledge acquisition techniques adopted can be classified as follows:

(i) Unstructured interviews
(iii) Structured interviews
(iii) Case-histories
(iv) Self-reporting

(v) Iterative prototyping

Two unstructured interview sessions were carried out at the start of knowledge elicitation to allow the knowledge engineers to get an overview of the domain. Subsequent sessions were mainly structured interviews with prototyping of the first element of the system taking place at the fourth interview. Elicitation of case histories was found to be very valuable as this technique allowed the knowledge engineers to see the approach taken in practice and also revealed some knowledge not elicited by the other techniques. A brief self-reporting session was also carried out. This is good practice in any domain since it serves as a check that the domain experts are describing how they approach a task in practice rather than how they think it ought to be approached.

Demonstration of prototype systems was valuable in that criticism of the prototype always revealed new knowledge. The prototype also served to correct misconceptions about the nature of the system being developed, particularly on the part of the domain experts. Early versions of the prototype merely incorporated small elements of the system - such as a graphic display or part of the hook time calculation. The experts became noticeably more enthusiastic when they saw a later prototype which provided the user with advice in the form of suggested methods for improving his crane selection. In the light of this experience the authors recommend that a prototype system should be developed and demonstrated at a fairly early stage. However for the best response the first demonstration should not take place until the system can give a piece of recognisable and plausible advice.

SUITABILITY OF REPRESENTATION

Up to the present time the Department of Civil Engineering at Loughborough has used commercial shell products for the development of expert systems since we believe that these are more appropriate to the construction industry than the development of inference engines in-house. Savoir enabled us to develop the graphical routines and data-base handling facility for CRANES in procedural Pascal code. It is essential for shells which are to be used in real-world applications to permit direct links to existing or specially developed procedural routines.

Savoir also provided an attractive user interface (a colour window display format) and a variety of user options. The amplification option was essential to this particular system, and an explanation facility is an important feature of all knowledge-based systems. The explanation text in Savoir has to be written by the programmer, whereas the authors believe that an automated explanation facility which displays the text of rules within the system should be used. Development of explanation text by the programmer is time-consuming, and there is a risk of explanations becoming out-of-step with the actual logic of the system due to coding errors or a failure to amend the explanation text when rules are changed. Admittedly some rules can appear obscure if displayed directly, and an ideal system would provide an automated display of the rule in use plus a facility for adding additional explanatory comments where necessary.

The iterative nature of the crane selection procedure involves repeated looping through the graphics and hook time modules. Loops can be created in Savoir using demons, with "made" variables as loop counters if required (a demon is a command which is executed immediately the trigger condition which qualifies it is satisfied). Care is required to ensure that the firing of the demons controlling the loop produces the desired effect. If a number of operations are to be carried out by a single demon (i.e. displaying text, clearing previous results, calling a Pascal routine, incrementing a loop counter) then the hierarchy of operations used by Savoir may lead to an outcome not anticipated by the programmer.

Provision of an accumulator for the total hook time required the use of a nested loop. This feature was particularly difficult to provide due to the vagaries of demon control.

The most serious problems of representation were encountered when rules were required to determine the mast heights of a number of overlapping tower cranes. The experts suggested a hierarchy of criteria for deciding the relative heights of overlapping cranes which can be summarised as follows:

(i) Cranes with luffing jibs must always be the tallest to avoid iib-jib collision when the luffing jib is raised.

(ii) If two crane masts are at a separation which is less than the jib length of one crane, then the crane with the longer jib must be tallest to avoid jib-mast collision. (Note - if both jibs were longer than the distance of separation of the masts then the user would have been instructed to amend his crane selection at a previous stage).

(iii) The crane with the greater load carrying capacity should be the taller.

(iv) The crane with the longer jib should be the taller.

General rules were required which could be applied irrespective of the number of cranes selected by the user. Savoir did not provide a means of representing such rules, and required different rule sets to be written for cases involving one, two, three cranes etc. Only the cases of one and two cranes have been coded. Even the case of two cranes is difficult because the hierarchical nature of the above criteria is tortuous to represent in Savoir.

Savoir only offers a production rule representation of knowledge (and also uncertainty statements which were rarely used in this system). Our more recent work with other shells suggests that a shell incorporating frames and pattern matching is required for adequate representation of the overlapping cranes problem.

CRANES currently comprises 90 rules and 100 demons, and incorporates 1000 lines of Pascal in the graphics module. Response time is generally good on the IBM PC XT on which the system is mounted but slows at a couple of key points in the consultation.

SPEED OF DEVELOPMENT

Knowledge acquisition sessions produced a wealth of information about crane selection and the system at present only addresses a small proportion of this. Difficulty was experienced in structuring this information and in deciding "where to start". Once coding had commenced delays were experienced at times due to the difficulties of knowledge representation and demon control described in the previous section.

Some of the features which slowed development have previously been recognised by other authors. It has been suggested that problems which take a few hours at most to solve represent good candidates for knowledge-based systems (Nii 1983, Bobrow et al 1986). A building planning engineer might spend about three days selecting appropriate cranes for a multi-storey building contract and this time span is reflected in the wealth and complexity of the knowledge encountered in this domain.

Some other features have made this domain more difficult than others that the authors have attempted:

(i) use of spatial reasoning
(ii) use of iterative processes

CONCLUSIONS

o At the start of knowledge elicitation, unstructured interview sessions involving up to six experts were successful. However once we were past this initial stage it was not possible to keep such a large number of participants involved.

o Elicitation of case-histories was found to be a valuable technique in this domain.

o Demonstration of a prototype system served to foster enthusiasm on the part of the domain experts and also served to correct some serious misconceptions about the nature of the system. Furthermore it stimulated the release of more knowledge. Hence early prototyping is valuable, but it should not take place until the system can give a piece of recognisable and plausible advice.

o A self-reporting session serves as a check that the domain experts are describing how they approach a task in practice, rather than how they think it ought to be approached.

o The interest of participants starts to wane if knowledge elicitation interview sessions are more than 2 hours in duration.

o We were only able to represent a small part of the knowledge in what proved to be a very large domain. The complexity of the domain knowledge is evidenced by the time taken by a planning engineer to select cranes for a site - typically about three days.

- o CRANES relies on the user to create solutions which are then criticised by the system. The system has been written as a helpful assistant rather than one which automatically generates correct solutions.

- o The amplification option of the Savoir expert system shell was valuable since much expert knowledge was embodied as advisory texts to help a user respond to system questions.

- o The production rule knowledge representation of Savoir was not sufficiently powerful for this domain and a frames representation was required to permit successful reasoning about problems such as that of overlapping crane jibs.

APPENDIX 3 - BID/NO BID

INTRODUCTION

At an early stage in our consideration of expert systems we had recognised the potential value of a system that would help a contractor to decide whether to bid against each new enquiry. We saw this as particularly important for contractors who provide a design/construct service as the cost of making their decisions is high and the cost of wrong decisions still higher.

We worked with IDC Consultants Ltd in developing a system of this type. The initial knowledge acquisition sessions were conducted by the principal investigator and the principal expert was the deputy chairman of IDC Consultants Ltd. We initially assumed that a decision would be made quite quickly on each enquiry but it became clear that this model is incorrect. The pattern is that one of IDC's "negotiators" establishes contact with a potential client and works with him in evolving the design brief. Thus there is often a quite protracted period of negotiation, often occuring over several months, that precedes the signing of a contract. IDC monitor the negotiations as they proceed and decide from time to time whether to continue with them or not. The system has to be capable of dealing with assessments of likelihood and goodness which will change as the negotiations proceed.

It was originally envisaged that the system when complete would be used by senior management as an aid to decision-making. As the work proceeded it became clear that, if the system is to be used, it is more likely to be in the hands of less senior negotiators. In this position it would help to ensure that all relevant factors are kept under consideration during the negotiation process.

THE BID/NO BID SYSTEM

A user who consults BID/NO BID is asked a series of questions which are grouped under the headings 'preliminary factors', 'client factors', 'project factors', 'bid factors' and 'resource factors'. In the preliminary section the system investigates five goals which can lead to a recommendation that the user should either definitely continue negotiation or definitely cease negotiation. For example if it is known that more than eight companies are pursuing the work then the system will categorically advise against continuing. On the other hand a prospect of working for a high prestige client on an average or large size project should definitely be pursued.

If none of the five preliminary goals is applicable then the system enters a scoring phase which builds up an overall score for consultation on the basis of a weighted mean score from four groups of questions. The four groups are as follows:

- (i) client factors - client characteristics such as financial soundness
- (ii) project factors - details of the proposed project
- (iii) bid factors - characteristics of the bidding situation such as the number of competitors involved
- (iv) resource factors - the capacity of IDC to supply the resources required for the project

Within the system each factor is allocated a weight between 1 and 5 determined from discussion with the experts. At the end of a consultation BID/NO BID displays an overall consultation score on a scale between +5 and -5, with advice based on that score. For example a final score between -1.5 and -3.5 would yield the advice "it is probably not worth bidding". The term "bidding" here implies "continuing negotiation".

KNOWLEDGE ACQUISITION

The principal expert in this case was the deputy chairman of IDC Consultants Ltd, with some additional comments made by the sales director and an assistant. Comments made by members of the sales team when evaluating various prototypes were used to enhance the system.

The knowledge base was initially developed using two main techniques: interviews and documentation analysis. Initial knowledge elicitation sessions were carried out in the form of "unstructured" interviews. A subsequent session was carried out using a *structured interview approach*. The data obtained from the initial interviews was analysed and classified into categories. In particular, contradictions in the data were identified and resolved. This "resolution of contradiction" approach proved to be a particularly fruitful means of expanding the knowledge-base. The knowledge-base was also expanded by incorporating a number of factors drawn from a previous project on bid strategies carried out by IDC for training purposes.

Once the prototype system had been developed a copy was given to IDC and one of the experts involved in the elicitation sessions was asked to monitor the use of the system and record responses to it. The system was also demonstrated to a number of individuals in IDC including the sales director, and four of the company salesmen were asked to run the system for jobs which they were currently negotiating. All these prototyping exercises produced useful comments and criticisms which resulted in improvements to the knowledge-base. Some comments related to the quality of the user interaction and led to the original interpretative system being entirely rewritten using an expert system shell (this is discussed below).

An extensive review of the rewritten system was carried out with six IDC salesmen (the target users of the system). These individuals were again asked to consult the system with regard to current or recent projects, and in the course of the evaluation ten consultations were carried out. The results of this evaluation were very encouraging. The advice given by the system and the scores for individual projects were in general similar to user expectations. However several examples occurred where users and BID/NO BID arrived at a similar score for different reasons and there is clearly scope for further refinement of the knowledge base. A number of detailed comments were made about the wording of system questions and this is clearly an area which needs careful attention during system development.

In general the users considered that the system would provide an unbiased assessment of a project and that it would be valuable to have this available to compare with their own more subjective views. Also the system forced them to think about all aspects of a bid opportunity, some of which they would tend to overlook. The young inexperienced negotiators were particularly enthusiastic about BID/NO BID and one was very keen to have a copy on his own computer so that he could have free access to it. The more senior negotiator was more critical but saw a role for the system as a training tool. He was sympathetic to the idea of BID/NO BID and had contributed to it at an early stage. However he made some perceptive comments about its use in practice. In particular he considered the assumption behind the system was that a negotiator would obtain the details of a project and then decide whether to bid or not. In practice project opportunities tended to evolve over a number of months with the project details emerging very slowly. The salesman would have to interact with the client throughout this period and could not wait until most of the information was available before running BID/NO BID and hence deciding whether to proceed. Furthermore the salesman should at all times encourage the client to appoint IDC without recourse to competitive bidding. These comments are valuable because they reinforce our findings in other domains that correct identification of the role and users of a system prior to development is of extreme importance. Even so the evaluation sessions were generally very successful.

KNOWLEDGE REPRESENTATION

Initially the prototype BID/NO BID system was developed as a custom-built forward chaining interpreter implemented on an IBM PC XT computer in Basic. The knowledge-base was kept separate from the interpretation mechanism so that the domain knowledge could be very easily amended. In the early stages of the work it appeared that this simple system would be quite adequate as a means of representing the domain knowledge, and it had the additional advantage that IDC could use it without the need to purchase a licence for a commercially produced expert system shell.

This initial system operated with a knowledge-base containing some 120 "checklist" items and an interpreter which calculated the bid "scores" from the values assigned by the user to each checklist item. The checklist items were weighted according to the significance ascribed to them by IDC's experts. The system also contained a built-in editor which could be used to add items, delete items, change the weights of items and amend wordings. The user would see the text of each item and be asked to respond in one of three ways: -5 to +5; yes or no; 0 to 100.

The interpreter was demonstrated to IDC personnel on a number of occasions and proved valuable as a focus for the knowledge acquisition process. However it was criticised for asking questions which previous responses had rendered irrelevant. Also users found some of the questions ambiguous and there appeared to be a need for clarifying text to be available on request - such a facility is provided as the "amplification" facility in the Savoir shell. We concluded from these comments that there was a need to code the system within a more sophisticated shell.

A new prototype system was developed using the Savoir shell and this showed the following improvements:

- unnecessary questions were eliminated
- an explanation facility (of the reason for asking the current question) was made available
- a simple why facility (i.e. why has this advice been given) was made available
- amplication text could be easily inserted
- in addition to an overall score for the bidding opportunity, specific advice was given such as "you should definitely bid" or "you should seriously consider bidding" etc.
- advice was made dependent on the range within which the final bid score lay. Once sufficient data had become available to limit the range of all possible outcomes to one of the critical ranges the system could reach a conclusion. Hence the system was now often able to advise of an outcome without needing to ask all the available questions. (If this occurred the user was given the options of ending the consultation immediately or going through the remaining relevant questions).
- during the main section of the consultation the user could at any time review how the range of possible final scores had been reduced by the data that he had provided up to that point.
- the system recorded a checklist of the items which the user had indicated were not yet known. For example, if he replied that the form of contract had not yet been agreed then the system would remind him at the end of the consultation that lack of this information had reduced the accuracy of the final score.

The Savoir shell was used to reimplement BID/NO BID because of our familiarity with it gained when writing other systems and because it would run on an IBM XT. All the features described above used standard Savoir facilities and we were able to rewrite BID/NO BID in a little over two weeks. The final system included some 40 rules and could ask up to 60 questions of the user in a full length consultation.

CONCLUSIONS

o The final version of BID/NO BID was evaluated on nine real world project examples and was in general well received by the sales team.

o Even during the final stages of development there was still a debate about how BID/NO BID would be used in a real bidding situation. This underlines the importance of fully defining exactly who will use a system and what its role should be when the domain is selected. This may often be difficult as both the client and the knowledge engineer have to go through a learning process. The client has to learn about the potentials of expert systems; the knowledge engineer has to learn about the domain and its environment.

o Initially knowledge acquisition was successfully carried out by interview techniques with some additional information obtained from documents.

o Iterative prototyping proved valuable, and was successfully carried out using the simple interpretative system even though this was eventually re-written.

o The early system was rewritten using the Savoir expert system shell in only two weeks. This demonstrates that once a knowledge engineer is familiar with a shell, and provided that shell offers a suitable knowledge representation (in this case production rules), then a shell provides a means of very rapidly producing a system with a good user interface and user facilities.

APPENDIX 4 - MATSEL

The MATSEL system is described in a paper "MATSEL - use of a commercial shell in a domain involving code of practice knowledge" by C N Cooper, M C Allen and A J Jones which is at present unpublished, and much of this appendix is taken from that paper. Copies of the paper are available on request. We have permission from our collaborators, Foster Wheeler Power Products Ltd, to report to CERL on system development issues.

INTRODUCTION

Foster Wheeler Power Products Ltd are a major engineering company specialising in the design and construction of large industrial boilers. The company wished to study the potentials of expert systems. This was their primary objective and the choice of domain was not critical provided that it related to practical procedures that the company deals with in their normal business.

During the domain selection process the most important criterion used was that there should be a real need for the system and that the end users could be clearly defined. A second key criterion was that suitable domain experts could be identified who were sympathetic to the development of such a system. The first step was therefore to talk to as many people as possible, both experts and users, about the possible domains available in order to assess whether a real need for a system existed.

The first domain proposed related to the commissioning of boilers on site, and this was rejected both because the precise mode of use of the system was hard to define, and because the application was excessively complex. It has been suggested that tasks which take a human expert a few hours at most to perform are suitable for expert system applications using current technology (Nii 1983, Bobrow et al 1986). The lengthy and highly complex boiler commissioning process far exceeded this limit.

The domain finally chosen was the design of tubing for the heat exchangers of large industrial boilers. A tube design involving comparison of a number of materials could be accomplished in, say, 30 minutes once the required codes and manuals had been assembled and the relevant sections located. An advantage of the material selection domain, which only came to light during development, was the ease with which the domain could be sub-divided. The initial intention was to address material selection for tubes to

British, American and German codes of practice and perhaps to consider pressure parts other than tubes as well. However it quickly became clear that because of the extent of the knowledge involved it would be necessary to restrict this initial development to tubes designed to British codes of practice.

If correctly performed the boiler tube design process should lead to selection of the grade of steel which can satisfy the design requirements in the most cost-effective manner. In the boiler industry this process is known as 'material selection' and the system developed has therefore been given the name MATSEL. MATSEL was written using the IBM shell ESE which can be used on the IBM mainframe computers operated by Foster Wheeler.

THE DESIGN OF BOILER TUBING

A typical industrial boiler contains a number of banks of tubing which act as heat exchangers. Different heat exchangers perform different functions and are located in different parts of the furnace so that their design conditions vary.

Each bank of tubing must be designed according to the operational requirements of the boiler. The designer must take into account the internal pressure the tube is subject to, working temperature, design life and possible erosion and corrosion during that life. Thinning and work hardening will occur when tubes are bent, and tubes with welded fins or other attachments need special consideration. Various methods of manufacture are available, and a variety of steels can be used. The key decision lies in choosing the steel which will satisfy the design requirements at least cost.

Many industrial boiler contracts are let on a fixed price design and construct basis. The tender period is generally about three weeks, and within this time-scale the contractor must carry out preliminary design work in order to arrive at a realistic tender price. Lack of precision at this preliminary design stage can lead either to an overpriced bid and failure to win the contract, or to an injudiciously low bid which erodes profit margins. Furthermore some overseas clients may resist design changes between the preliminary and final designs, even when these involve little more than a change in the grade of steel used. Hence a need can be demonstrated for the preliminary design to be carried out with greater precision for the same time spent, and without significant increases in design costs.

Increased use of computers presents the most obvious solution to this problem. However difficulties could occur, in particular as a result of loss of design expertise on the part of design engineers when their work is automated. Also the design process is subject to constant change resulting from the updating of codes of practice, and the imposition of client-specific variations. This leads to a need for constant modification of software.

A possible solution to these difficulties seems to be the use of expert design systems rather than conventional software. The explanations the system can give the user help to keep the design procedure accessible to the engineers, and the design knowledge can be concentrated in a knowledge-base which is structured as rules documented by their rule-names.

CODE OF PRACTICE KNOWLEDGE

Most tube design criteria are addressed by codes of practice, and much of the knowledge contained in MATSEL is defined in British Standards BS1113, BS3059 and BS3600. However in some cases a designer will adopt company practice which is additional to the code of practice requirements. An example of this might be restrictions on the parts of the boiler in which tubes with welded seams are used. Additionally the code may instruct the designer to consider certain criteria but give him no advice on how this should be done - an example is the allowance for external erosion and corrosion of tubing.

MATSEL therefore contains a high proportion of code of practice knowledge but combines this with the expertise that a skilled designer uses in the practical design of boiler tubing. Published work on the use of codes of practice as knowledge bases is summarised in the paper on which this Appendix is based.

A CONSULTATION WITH MATSEL

As in any interactive expert system, a consultation with MATSEL commences with a number of questions to the user relating to the function of the tubing to be designed, and design criteria such as internal pressure, fluid temperature and required working life. At any stage the user can use the "Why?" query facility to ask why the system has asked a particular question. The rule currently in use and the chain of inference will then be displayed.

Some use is made of the "What?" facility of ESE whereby a user can ask to see text which will help him answer the current question. However some of this advice is required to be context sensitive, and furthermore there is a risk that users might never bother to access text which is normally hidden from view in this way. Accordingly use is made of the ESE "Exhibit" facility by which means advice on answering a question tailored to previous user answers can be displayed alongside the question itself. In other cases the user is specifically asked whether he wants to view screens of advisory text, for example on corrosion problems.

In the course of a consultation MATSEL is required to calculate the tube wall thickness required for a number of materials under the design criteria specified by the user. On completion of the calculations for each steel a screen is displayed which summarises the tube diameter, thickness and method of manufacture selected. The user is able to ask "How?" the system arrived at the tube thickness stated, and will then see the final rule used in the calculation and its' rulename. Any of the parameters referenced in the premise clause of the rule can be queried, and the rules used to derive these values obtained. In this way a user can step back through the logic used to derive any result, and hence check on the validity of the reasoning followed. Actual values are displayed at each stage. The use in ESE of the actual text of rules to explain the consultation is a valuable feature since it ensures that explanations are always consistent with the reasoning process used by the system.

When the user has seen calculations performed for all the steels he is interested in he can ask for a summary screen to be displayed. On this he will see a list of all the steels examined, together with the method of manufacture and nominal thickness of each, and a relative cost for each. Actual costs of tubing are very volatile and vary considerably between suppliers. However the ratios of cost between different grades are more stable and therefore a database of relative costs should require less maintenance then one of actual costs.

Normally the user would select the steel which has been shown to be the cheapest. However a display of all results is given so that in unusual circumstances he is able to select another, perhaps slightly more expensive material.

PAPER MODEL

The knowledge acquisition for MATSEL was carried out by interview techniques, discussion of a paper model and iterative prototyping. Since an intermediate representation or paper model appears to be less widely in use than the other techniques this will be described in some detail.

It has been the experience of the authors that the organisation and structuring of the knowledge acquired from domain experts may be even more of a problem than the initial acquisition process. The objective of the paper model described here is to help in the organisation of the knowledge prior to coding the system.

A page from the paper model created for the MATSEL system is presented in Figure 2. The format is based on that used very successfully by Logica in their work for the ARIES Alvey Community Club. The paper model in effect acted as a specification for MATSEL and included a statement of the objectives of the system, the rules the system would contain and warnings and advice which should be made available to the user. Knowledge is generally stated in the form of production rules with an english-like syntax which are intended to be comprehensible to the domain experts. The adoption of an english-like syntax also helps to keep the intermediate representation to some extent independent of the knowledge representation in the expert system produced, *although some form of production rules* were presupposed when constructing the paper model described here.

In addition to recording the rules themselves, the paper model also documents the source for each rule. If the rule is derived from a code of practice then the name of that code and the clause on which the rule is based is noted. Other rules are recorded by the section of the design manual from which they were taken, or by the name of the source expert and the date on which he was interviewed.

A key feature is that the status of all items in the paper model is recorded by a letter code in the left hand margin. An ampersand facilitates searching for specific codes on a word processor. These codes allow the knowledge engineer to keep a formal record of rules which are finalised and rules which are still tentative and require further discussion with the domain expert. The full list of letter codes used in the MATSEL paper model is as follows:

&f - finalised i.e. this line or rule is unlikely to change.

&t - temporary rule intended only to define the limits of the current domain.

&d - line or rule to be deleted when the system is next revised.

&x - line or rule which is expected to require refinement.

&z - line or rule which has recently been amended.

The paper model was revised at frequent intervals and could be presented to the domain experts for review. After a full review of the preliminary version, the domain experts could confine their attention solely to changes made since the last issue and tentative items, and could therefore scan the paper model looking only at those items marked with &x or &z. Had the paper model become larger a simple piece of software could have been developed which on request would have restricted printed versions to these recently amended or tentative items.

The development of MATSEL was carried out by two individuals who acted as knowledge engineer and programmer respectively. Both were present during knowledge acquisition sessions but the knowledge engineer undertook the transcription of interviews, research of codes of practice and creation of the paper model. The programmer then coded the system from the paper model. An intermediate representation therefore assisted in separating the knowledge elicitation and programming tasks, and provided an acceptable specification for the programmer to work from.

```
                MATERIAL SELECTION PAPER MODEL        12.06.87

Design conditions

&f   The user should be asked for the 'calculation  pressure'  in
&f   N/mm2 and for the maximum temperature in Celsius of the fluid
&f   which will be carried by the pressure part (denoted  'fluid
&f   temperature' in the rules which follow)
&f
&f
&f   BS1113 : 2.2.4
&f
&f   The user will also be asked for the 'design lifetime' of the
&f   element  (in hours) if the material is in the creep range or
&f   if  lifetime  is  of  relevance  to  erosion  or  corrosion
&f   allowances. Lifetime could also be of relevance to  fatigue
&f   considerations.
&f   The user should be offered a menu of 100000,  150000, 200000
&f   and 250000 hours.
&f
&f
&f   REMIND USER : BS1113 : 1.3.6 {Note: 1.3.6b has been omitted}
&f
&f        The 'calculation pressure' should be gauge pressure  in
&f        N/mm2  and should take account of  all  the  following
&f        under the most severe conditions of operation:
&f             -the highest set pressure of any safety valve
&f             -pressure drops due to line losses
&f             -hydraulic head
&f
&f
&f   RULE : BS1113 : 2.2.3.3a
&f
&f        IF the pressure part is 'furnace tube' or 'convection
&f        bank'
&f        AND the pressure part is subject to 'radiant heat' from
&f        the combustion chamber
&f        THEN 'design temperature' = 'fluid temperature' + 50
&f
&f
&f   RULE : BS1113 : 2.2.3.3b
&f
&f        IF the pressure part is 'furnace tube' or 'convection
&f        bank'
&f        AND the pressure part is not subject to 'radiant  heat'
&f        from the combustion chamber
&f        THEN 'design temperature' = 'fluid temperature' + 25
&f
&f
&f   RULE : BS1113 : 2.2.3.4a
&f
&f        IF the pressure part is 'superheater' or 'reheater'
&f        AND the pressure part is subject to 'radiant heat' from
&f        the combustion chamber
&f        THEN 'design temperature' = 'fluid temperature' + 50
```

Figure 2 - A page from the MATSEL paper model

KNOWLEDGE ACQUISITION

General

The following knowledge acquisition techniques were used:

(i) Unstructured interviews
(ii) Structured interviews
(iii) Case-histories
(iv) Paper model
(v) Iterative prototyping

During development of the system twenty-four knowledge acquisition sessions were undertaken with eleven domain experts. The total duration of knowledge acquisition sessions was twenty-nine hours and in all but one case the domain experts were interviewed individually. Some of the experts spent additional time assembling documents and case history material for discussion during interview sessions.

Two domain experts provided much of the core knowledge for the system and a further three individuals were consulted on specialist issues such as corrosion, procurement and quality assurance. Of the remainder three provided further background knowledge and three were involved only in assessing MATSEL during the final stages of development.

The bulk of knowledge acquisition was by interview techniques. Typically the knowledge engineer used an unstructured technique when starting to work with a new expert or on a new topic. Interviews then progressed to broadly focussed and narrowly focussed structured questioning. However the distinctions between these forms of questioning are rather subjective and it was always necessary for the interviewer to vary his approach according to the responses he obtained from the expert being interviewed.

The elicitation of case histories was a particularly valuable technique in this domain and the structure of MATSEL evolved from a set of calculations presented by one of the two key experts.

A key expert was asked to review the paper model on two occasions, the first approximately one month after the commencement of work and the second after three months when the system was almost complete. He seemed to have no difficulty in following the production rule representation and made useful comments. There seems to be scope for greater involvement of domain experts in the development of a paper model of this type once they have understood the basics of rule-base construction.

The prototype system was demonstrated for the first time about two months after commencement of knowledge acquisition when it was capable of calculating material thicknesses. The comments resulting from this demonstration were useful and worthwhile but less new knowledge was elicited through iterative prototyping than the authors have experienced with some other systems. This is because errors in the operation of MATSEL, manifested as incorrectly calculated material thicknesses, were not as immediately recognisable as, say, an incorrect diagnosis from a fault-finding system would be. Even so demonstrations of the prototype were still important as a means of promoting enthusiasm on the part of the experts and checking for gross errors. However perhaps more effort should have been expended in reviewing the minutiae of the paper model in which the rules embodied in the prototype were made explicit.

The nature of the expertise used to build MATSEL was somewhat different to that in other domains which the authors have worked in. Much of the knowledge elicited related to sources of information the experts would use in order to select materials. For example an expert would know where to look in BS1113 for allowable stress data, and where to look in the Foster Wheeler Engineering Design Manual to obtain criteria for the bending and welding of tubes. In many expert system domains the experts carry nearly all the domain knowledge in their heads. Because of the nature of the MATSEL expertise it was necessary for the knowledge engineer to understand and extract knowledge from documents and codes of practice as directed by the experts. This process could be time-consuming, particularly in the case of British Standards.

Domain experts were in general interviewed by two people whose primary roles were knowledge engineer and programmer respectively. Interview sessions involving one expert and two knowledge engineers seemed successful. On one occasion a third person also sat in as an interviewer and the expert who was confronted by three interrogators not unreasonably complained of "an inquisition".

Most of the knowledge acquisition sessions were taped. The time required to transcribe the tapes was a major problem - for example a two hour tape took nearly twelve hours to transcribe. If time is a problem the best solution seems to be the use of hand-written notes with a tape back-up which can be used to check specific points but is not transcribed in full.

Conflict between Experts

In discussions of knowledge acquisition from a number of experts the issue of conflict between experts is often raised. In the domain of material selection much knowledge is embodied in codes of practice and design manuals in an unambiguous form so that the scope for disagreements between experts is not as great as in many other domains. Even so there was some disagreement between the MATSEL experts over issues of good engineering practice on which no clear guidelines were available in written documents.

The first point to make is that good documentation of the knowledge elicitation process is necessary to enable the knowledge engineer to detect and understand these conflicts in the first place. This necessitates detailed records of each knowledge elicitation session and a systematic approach to collating the information collected into some form of intermediate representation.

Once conflicts have been recognised then there are three possible approaches to resolving these:

(i) Hold joint interviews with the experts who have conflicting views and see if they can reach a consensus.

(ii) Produce a system which will present alternative views to the user and allow him to choose the approach he prefers.

(iii) Adopt the views of the expert who is judged to be the most knowledgeable.

All three methods were used at different times during the development of MATSEL and the correct approach is clearly very situation-dependent. Solution (iii) seems to raise more difficulties since it may put on the knowledge engineer the onus of judging the relative abilities of different experts. However in the engineering company environment senior managers and directors are expected to be highly experienced in the design process and when conflicts arose the experts tended to turn naturally to more senior colleagues for advice. Hence the experts themselves tended to adopt the third approach listed.

A very high degree of cooperation was given by the domain experts for MATSEL and information was freely provided. This seemed to result from the perception of this system as one which would enable material selection to be carried out more rapidly and with more individual materials considered. Hence it would be a tool that the experts could use to perform a part of their work more quickly and with greater precision.

KNOWLEDGE REPRESENTATION

Production Rules

MATSEL was coded using the IBM Expert System Environment (ESE), a shell developed and marketed by IBM which is only available for use on IBM mainframe computers. The use of ESE fitted well with Foster Wheeler policy which is to offer computing services via terminals linked to centrally located mainframe computers.

Knowledge representation in ESE takes the form of production rules such as:

```
RULE:      BS3059_PT2_11_2_1_A

IF         pressure_part is 'tube'
AND        method_of_manufacture is 'seamless class 1'
THEN       tolerance_on_o_d = 0.005 * outside_diameter
AND        tolerance_on_i_d = 0.005 * inside_diameter
```

ESE rules are given unique names by the programmer and these names offered a valuable facility for documenting the contents of the knowledge-base. Hence the rule name records that the rule shown above is taken from British Standard BS3059 clause 11.2.1. Should that clause change then the rule or rules which require alteration are immediately recogniseable.

Focus Control Blocks

The ESE knowledge-base is structured so that all rules and parameters are owned by one (or sometimes more than one) focus control block (FCB).

This is in itself a useful feature because it provides a means of structuring the rules in the knowledge-base. The FCB's offer the following facilities:

(i) Control text can be defined in each FCB
(ii) FCB's can be defined in a hierarchy
(iii) Multiple instances of an FCB can be created

Control text constitutes the definition of whether forward or backward chaining is to be used and which conflict resolution strategy is to be implemented. Different inference and conflict resolution strategies can be defined in each FCB.

MATSEL uses both forward and backward chaining. At the start of a consultation a number of basic questions are asked and forward chaining is then used to determine which individual codes of practice are relevant and which materials could be considered for the application. The costs per unit length of tubes in each of these materials are then defined as goals and MATSEL backward chains from each goal in turn.

In the course of either backward or forward chaining a number of rules may simultaneously become relevant to the reasoning process. The inference engine employs a user-defined conflict resolution strategy to decide in what order these rules should be fired. The objective is to try and find a solution by the shortest possible route - which should keep the number of questions asked of the user to a minimum. The conflict resolution strategy is defined for each FCB and includes options such as "least unknown premises first" and "most true premises first" etc.

For MATSEL to be a practical tool it was important that the costs of using several different materials should be calculated in a single consultation. Many of the rules are common to all the materials considered and it would be pointless to code each rule separately for each material, as would be necessary with a shell restricted to production rules alone. ESE allows copies of FCB's to be generated at run-time so that a set of rules can be used repeatedly and all the results preserved. In MATSEL new rule sets are automatically generated whenever a new material is considered by the system. This replication process is known as instantiation.

External Interface

In an expert system it is frequently necessary to carry out complex calculations or data-base searches by calling routines written in a procedural language. ESE is capable of interfacing with routines written in Pascal and Fortran and this allowed a simple Pascal routine to be written to access a file containing tables of allowable stresses taken from a code of practice. ESE allows any number of parameter values to be passed to an external routine as data items but only allows a single result to be returned to the knowledge-base. This is extremely irritating and can only be circumvented by using the external routine to create a temporary work file from which values are accessed one at a time by a second external function as required.

User Options

ESE offers twelve user options via programmable function (PF) keys. Three of the most important of these are:

What -	displays text to expand the current question
How -	i.e. How did the system get this result?
Why -	i.e. Why is the system asking me this question?

DEVELOPMENT EFFORT

MATSEL contains 145 rules and includes an external routine for referencing tables of allowable stresses. At the start of the project the knowledge engineer had experience of expert system development using a different shell, and the programmer had no previous expert systems experience. The shell used is very sophisticated and allows complex control of the inferencing so that an extended learning process was necessary on the part of the programmer.

The total development time for the system was 770 man-hours. This is made up as follows:

Management Time	50 hours
Experts' Time (interviews/reviews of prototype)	40 hours
Knowledge Elicitation (interacting with experts)	60 hours
Transcribing selected tapes of interviews	60 hours
Research/Rule synthesis/Paper model	157 hours
Shell familiarisation (including training course)	124 hours
Development of MATSEL code	279 hours

Development of a further application using the same shell and personnel could be accomplished without the shell familiarisation stage, and the coding of the system would be more rapid. It is therefore reasonable to assume that an overall development input of about 500 man-hours would be required.

CONCLUSIONS

o The success achieved in developing MATSEL is attributed to careful selection of the domain with particular attention paid to the selection of an application of modest complexity for which users could be clearly identified. It was also important that a number of enthusiastic experts were readily available.

o The use of an intermediate representation (paper model) was valuable for a number of reasons:

(i) It helped to organise and document the large amount of knowledge collected.

(ii) Discussion of the paper model stimulated further knowledge acquisition.

(iii) It facilitated separation of the tasks of knowledge engineer and computer programmer.

o Demonstrations of the prototype system were important as a means of maintaining the interest of the experts.

o A single expert confronted by three knowledge engineers during an interview not unreasonably complained of an "inquisition".

o Tape recording and full transcription of interviews was very time-consuming. For example a two hour tape took nearly twelve hours to transcribe.

o Conflicting views between different experts were not a major problem. Where they did occur three approaches were considered as follows. The last of these was often appropriate:

(i) Hold joint interviews with the experts who have conflicting views and see if they can reach a consensus.

(ii) Produce a system which *will present both conflicting* views to the user and allow him to choose the approach he prefers.

(iii) Adopt the views of the expert who is judged to be the most knowledgeable.

o There was a need to apply the same rules to a number of materials and a simple production rule representation was not appropriate. The ESE facility for multiple instantiation of rule sets contained within Focus Control Blocks was used to achieve this. A frame representation could also have been used.

o The completed system offers a number of advantages, some of which could also have been achieved using a procedural language, and some of which are specific to the expert system solution adopted.

o The benefits which could have been achieved using either approach include the following:

(i) Code of practice knowledge is combined with the experience of senior design engineers.

(ii) The system is much quicker to access than codes of practice.

(iii) The system should ensure the correct application of codes of practice.

(iv) The design engineer is able to compare the use of several materials and make simple cost comparisons. When working manually only one material could be investigated in the time available.

o The implementation of MATSEL as an expert system offers the following additional benefits:

(i) The user can interrogate the system for a justification of design results and for intermediate results.

(ii) The automated explanation facilities also aid debugging.

(iii) Rule names are used to document the source of each rule, and in particular the source clause for code of practice rules. This facilitates rapid updating of the rule-base.

(iv) The rule-base is easier for a non-programmer to understand than the equivalent procedural code would be.

APPENDIX 5 - ALVEY CONTRACT

INTRODUCTION

In July 1987 we started work on a research contract funded by the Alvey Directorate, a funding body for information technology research formed by the British Government. The project we are involved with involves research into the applicability of expert systems to engineering applications as exemplified by vibration analysis. The principal consortium members are as follows:

Stewart Hughes Ltd - a company specialising in systems for measurement and analysis of vibration.

Rolls Royce plc - manufacturers of aero-engines.

GEC - a major engineering company with wide ranging interests including the design and manufacture of power station alternators.

Loughborough University - Department of Civil Engineering.

Loughborough University - Human Science and Advanced Technology Research Centre (HUSAT).

The objective is to examine two possible expert system applications which relate to the particular interests of Rolls Royce and GEC respectively. Roles within the project consortium are as follows: Rolls Royce and GEC are providing knowledge in their respective domains and have also developed Fortran algorithms where appropriate. Stewart Hughes Ltd are supplying an expert system shell with interfaces to the Fortran modules. We are eliciting knowledge and coding this into the shell, and HUSAT have a limited role advising on human-computer interface issues. We did not become involved in the project until 13 months after the start and were therefore not involved in the domain selection stage.

We are able to report to CERL on the knowledge acquisition and development aspects of these systems.

THE ROLLS ROYCE SYSTEM

Rolls Royce are interested in vibrations in gas turbine engines. Tests are carried out by attaching transducers to the engine which is then accelerated from zero to full speed. The recorded vibrations are examined at discrete speed intervals using a fourier transform analysis so that the results from a given transducer can be displayed on a diagram which is known as a "Z-MOD". The Z-MOD has axes of running speed and frequency, with amplitudes of vibration shown by the use of colour. A typical plot appears as a complex of straight and curved lines. A skilled vibration engineer is able to deduce the phenomena which give rise to these lines - which include shaft vibration, blade vibration, electromagnetic interference, flutter, bearing faults etc. The objective of the project is an expert system which can perform this interpretation task. Rolls Royce are developing Fortran routines which can detect the lines on the Z-MOD and describe these in terms of line of best fit, amplitude, width, cross-sectional shape etc. The expert system will be able to call these Fortran routines, and will then use rules to deduce the phenomenon which has given rise to each line.

An expert system could in the long term have two potential uses. The first would be to carry out preliminary checks on Z-MOD diagrams. Because engine tests are very expensive large numbers of transducers are used in order to maximise data collection. Hence a single engine test may result in the production of 300 Z-MOD diagrams. It would be helpful to have an automated system which could categorise these diagrams as either (i) normal or (ii) unusual and requiring detailed attention from a human expert.

The second use would be as a vibration workstation which a vibration engineer could interact with as a means of interpreting and understanding Z-MOD diagrams. For example a user would be able to place a cursor over a particular phenomenon shown on a screen display of a Z-MOD diagram, and would then receive an explanation of that phenomenon with details of the reasoning process used by the system.

THE GEC SYSTEM

GEC wish to explore the use of an expert system in the balancing of large alternator rotors. The only source of vibration considered in this project is that due to an out-of-balance shaft (in contrast to the Rolls Royce system in which all types of vibration phenomena are considered). The objective of the GEC work is a system which will help a user to correct the out-of-balance in a shaft so that vibrations are brought within predefined acceptable levels.

Alternator shafts are balanced by changing the numbers and locations of small balance weights. Many constraints are imposed on the balancing procedure by the construction of the alternator - the shaft is generally only accessible at bearings and balance weights can only be positioned in a limited number of places along the shaft known as balance planes. Furthermore the number of transducers available for taking measurements on the rotor is generally very limited.

Balancing is to some extent a trial and error process. A number of test runs of the alternator are carried out with a different combination of balance weights used each time. An algorithm is used to process the combined test results and deduce an optimum disposition of balancing weights.

A single test involves running an alternator up to full speed, maintaining full speed for several hours so that "heat soak" can occur, and then reducing speed to zero. As a result the balancing process can take several days, and if the alternator concerned is a 660 MW set this "outage" can represent a major cost to the operator in terms of lost power generation. The objective of the expert system is to assist a vibration engineer in balancing the alternator with the aim of minimising the number of test runs required before balance can be achieved. The system will be interfaced to Fortran routines developed by GEC for complex numerical work, but is expected to contain rule sets to advise on where transducers should be located, the modes of vibration seen, and the best configurations of balance weights for tests and for final balance etc.

An interesting feature of the GEC work is that the company had previously purchased a computer package for the balancing of rotors. This was a conventional program which instructed an operator on what to do but gave him no insight into how decisions were being made. The balancing engineers reacted strongly against this "black box" solution and refused to use it. A major aim of the current work is to produce a system with justification rather than simply imposing actions on the user.

KNOWLEDGE ACQUISITION - ROLLS ROYCE

Two experts are available at Rolls-Royce and an engineering programmer who has developed their Fortran line recognition software also attends knowledge acquisition sessions. The numbers of participants from Loughborough and Rolls Royce present during knowledge acquisition sessions are generally the same - detailed issues may be discussed on a one to one basis, whereas broader discussions can be usefully conducted with up to three from each side present. Knowledge acquisition has been carried out by interviewing and we have found it valuable to write up formal notes of the knowledge acquisition sessions and to ask the experts to comment on these. These notes have been successfully produced from hand-written notes taken during the meetings - tape recording of meetings has not been found necessary. Since the objective of the system has been interpretation of the Z-MOD diagram it has been important to see and discuss such diagrams during knowledge acquisition sessions.

There has been considerable discussion as to how the system should work and the approach that is evolving will require a number of algorithmic techniques to be used in order to extract all the line information from the Z-MOD diagram. In addition to the notes recording knowledge elicited it has therefore also been necessary to produce and discuss structure diagrams for the overall system.

As in all the other domains we have studied we have produced prototype systems and have found these very important in maintaining the interest and enthusiasm of the experts.

To a certain extent we have found the less senior expert to be the more profitable for knowledge acquisition. This is partly because we have found him to be in more day to day contact with Z-MOD interpretation and partly because he has experience of conventional computer software and can better visualise the structure and requirements of the final system. Rapport with both experts has been good.

Progress on the expert system itself has been relatively slow since we have put considerable effort into amassing the knowledge and pattern recognition techniques required for the system. Some of the issues may be summarised as follows:

(i) As in many image recognition problems, we are finding that although linear features on the Z-MOD are easy for the human eye to identify, it is very difficult to develop reliable algorithms to do the same thing.

(ii) We are investigating whether high level rules have a role to play in this line recognition process. Such rules can more easily be "tuned" than procedural code and also enable the decision-making process to be reproduced as an explanation. Whereas some of the principles these rules might embody are available from humans, the testing of those principles and formulation into useable rules requires time-consuming study of existing Z-MODS. We are therefore developing new knowledge in the course of the work.

(iii) The second stage at which rules are used consists of deducing the phenomenon which gave rise to each line on the Z-MOD given the set of lines found on the diagram along with parameters such as shape, slope, width, peakiness etc. Here we find that the principles expressed by the experts sometimes do not hold true, and that effort is required to quantify some of the knowledge which is only expressed in qualitative terms. Hence we are again developing and refining knowledge as well as acquiring it from others.

KNOWLEDGE ACQUISITION - GEC

Two experts have contributed to the GEC system, and a graduate engineer has also been involved. We found it much more difficult to establish a rapport with the GEC experts than we did with those from Rolls Royce. This was because when we joined the consortium the GEC experts were understandably disappointed that over a year into the project they had still seen little that was relevant to their problem. In fact they had not seen a completed expert system demonstrated and we therefore demonstrated the CRANES system and the Leonardo shell in an attempt to improve relationships.

Knowledge acquisition was by unstructured and structured interviews, generally involving two experts and two or three knowledge engineers. Asking the experts to comment on notes produced after meetings produced useful feedback. Many of the early meetings involved the refinement of a structure diagram for the required system. This was found to embody a large procedural element and was complicated by the need to offer a variety of graphics facilities and to interface with existing Fortran programs. Production of this structure diagram was essential to gaining an understanding of the rather complex balancing process, but it also helped to improve relationships because the experts had evidence that some progress was being made.

A demonstration of an early prototype written using the Leonardo shell also contributed to a slightly better relationship, but a breakthrough occurred when a much more substantial Goldworks demonstrator was shown. This is somewhat similar to our experiences in developing CRANES which suggested that demonstration of an advanced prototype system can be a major factor in promoting enthusiasm on the part of the experts.

The GEC system will remain a source of difficulty for a number of reasons. Firstly a large proportion of the knowledge is procedural since much of the balancing operation is a step by step process. As a result of this the overall structure diagram is a flow chart and could be implemented in a procedural language. An expert system approach seems to be appropriate to about five key decisions, the most important of which is deciding the disposition of balance weights for a test run.

The experts have a number of existing Fortran programs that do not interact so that balancing involves running a program, recording key results and manually entering these into another program etc. They see a role for the expert system in controlling these programs and effecting all the necessary data transfers. In fact this task would be well suited to a procedural system written in, say, Fortran.

In the light of problems with the previous "black box" package the experts also place great emphasis on a variety of graphical displays which will keep the user fully aware of what is happening and give him sufficient information to make his own decisions and if necessary over-ride the system. Again these graphics facilities could be provided by a procedural system.

Much of the system could therefore be implemented in procedural code, and the need for an expert system seems to hinge on substantial knowledge existing in the five decision areas previously referred to. However we have had difficulty identifying what this knowledge is. These problems are exemplified by our lack of progress in eliciting knowledge in the key area of suggesting optimum positions for weights during balancing tests. We have found that:

(i) The experts have not themselves balanced a rotor for several years.

(ii) The experts have strongly resisted proposals that we should meet those who are balancing alternators on a regular basis (this is partly because those who regularly do balancing are in a different group company and partly because they are said to be difficult to approach). This rules out observation as a knowledge acquisition method.

(iii) There are only very limited records of how machines have been balanced on site in the past so that case histories are not available.

(iv) A scaled model rotor has been designed and built for testing the expert system and data collection software but this has only three bearings, in contrast to full size machines which may have up to eighteen. Work with this can only give limited insights into the knowledge required to balance a full size machine.

We have described some of the problems in this domain in detail since they are relevant to the general issue of successful expert system development. In particular:

(i) The objective should always be to solve a problem rather than to use an expert system per se - the GEC system is required to have extensive graphical and data handling capabilities which could easily be implemented in a procedural system.

(ii) Domain experts should be solving problems in the chosen domain on a regular basis.

A final issue relating to the GEC work is that of the knowledge engineers. Our knowledge elicitation has been carried out by a civil engineer and a mechanical engineer, both of whom have had a basic training in the principles of vibration. A computer science graduate from HUSAT has also been involved and we note that he was at a considerable disadvantage with the experts due to a lack of understanding of basic jargon such as, for example, "phase" and "hertz". We see an advantage in the knowledge engineer having at least a basic training in the expert's field.

KNOWLEDGE REPRESENTATION

The original intention was that Stewart Hughes Ltd would develop their own shell in Lisp. However after one year of work this attempt was abandoned and the Lisp shell Goldworks from Gold Hill Computers Inc. was purchased. We experienced some delay in delivery of the Personal Computer with 8 Mbyte RAM required to run Goldworks and in the interim period developed two demonstration systems using the Leonardo shell which runs on a PC machine with 640k RAM.

The GEC system relates to alternators which can be described in terms of bearings, spans (lengths of shaft between bearings), balance planes and transducers etc. The numbers of each type of element will vary between machines, and each class of element has a number of attributes. For example a transducer has attributes which include position, type, calibration, amplitude of reading and phase of reading. Furthermore it is necessary to write single rules which can be applied to the whole class of transducers or the whole class of bearings etc. This problem requires a frame representation. The following illustrates the form of a transducer frame:

Name:
IsA: Transducer
Position:
Type:
Calibration:
Amplitude_of_Reading:
Phase_of_Reading:

At the time when the first demonstration systems were produced the Leonardo shell offered facilities whereby a programmer could define a class frame and any number of instances of that frame, inserting values for each slot of each instance as appropriate. This facility was a considerable advance on simple production rule systems but for our applications it was necessary to create an appropriate number of instances and corresponding slot values at run-time (so that each time the system was run the instances related to the machine considered). Such "dynamic instantiation" facilities are provided by Goldworks. (Recently dynamic instantiation and greater slot accessibility has become available in Leonardo but we have not yet been able to test these facilities).

The Goldworks shell is very sophisticated and has proved suitable for representing the balancing and Z-MOD interpretation knowledge. (In the latter case each line identified on the Z-MOD diagram is represented by an instance of the "line" frame, with slots for line equation coefficients, peakiness, bandwidth etc). However we have found the syntax of the pattern matching rules (and therefore system explanations) obscure and difficult to explain to domain experts. We have also found that as the number of rules and frames increase the system response becomes extremely slow.

It is easy to call Lisp functions from Goldworks rules and Stewart Hughes Ltd have developed software to interface Lisp to Fortran subroutines so that incorporation of algorithmic software written in Fortran has been carried out successfully. For the GEC work Stewart Hughes have also developed communications software which allows the PC-based expert system to access and run software on the PDP-11 machine used for data collection.

CONCLUSIONS

o Care must be taken in selecting a domain for the application of an expert system. In particular:

(i) The domain experts should be engaged in solving problems in the chosen domain on a regular basis.

(ii) The objective should be the development of a useful system rather than the use of expert system technology per se.

o Previous GEC experience highlights the dangers of a "black box" system - the operators demanded a system which would help them make decisions rather than one which imposed decisions on them. Increased graphic displays of data were seen as one means of increasing user confidence in the new system.

o Discussion documents including knowledge acquisition notes and flow charts were found useful during knowledge elicitation sessions.

o Demonstrations of prototype systems were crucial to fostering enthusiasm on the part of the experts. Advanced prototypes were the most effective in this respect.

o There appeared to be advantages in using knowledge engineers who had a similar basic training to the domain experts.

- We note the difficulties experienced by Stewart Hughes Ltd in developing a shell in-house. This would only seem to be appropriate where the developers have considerable experience of expert systems work.

- Frames and the dynamic creation of instances were essential to an adequate knowledge representation in these domains.

- The syntax of the Goldworks pattern matching rules and explanations was intimidating to the domain experts.

- The Goldworks shell was sophisticated but slow and cumbersome.

APPENDIX 6 - PAPER

"Experience with Knowledge Acquisition for Expert Systems in Construction"
G Trimble and C N Cooper.
Proceedings of the 1st European Workshop on Knowledge Acquisition for Knowledge-Based Systems, Reading, UK, Sept. 1987.

EXPERIENCE OF KNOWLEDGE ACQUISITION FOR EXPERT SYSTEMS IN CONSTRUCTION

E G Trimble, Professor of Construction Management, Department of Civil Engineering, Loughborough University of Technology.

C N Cooper, Research Assistant, Department of Civil Engineering, Loughborough University of Technology.

INTRODUCTION

The authors are currently engaged in research into methods of knowledge acquisition for expert systems. Although their original focus was on construction industry applications the findings so far suggest that the choice of industry does not of itself affect the knowledge acquisition method selected. Rather it is the situation that determines the appropriate method. By situation we mean such factors as:

- Available time of the domain expert
- Whether he holds his expertise in explicit or intuitive form
- Whether he is motivated to help the process or hinder it.

The authors are associated with two research projects into knowledge acquisition methods. The first involves the development of a number of expert systems in conjunction with industrial clients as summarized below:

CONPLANT - selection of materials handling equipment for multi-storey construction sites.

BREDAMP - diagnosis of the cause of dampness in buildings.

CRANES - selection of tower cranes for multi-storey construction sites (incorporating a graphic interface).

BIDDER - decision of a design and construct contractor on whether to bid for a project.

NETWORK - diagnosis of faults in a national computer network.

MATSEL - material selection for boiler pressure parts.

The first five systems utilise the SAVOIR commercial shell program and run on an IBM PC XT. The last named uses the IBM shell ESE and runs on an IBM mainframe computer. Different client organizations and experts have been involved in each case and this has enabled the authors to gain a broad spectrum of experience of knowledge acquisition.

The second area of research is being undertaken by Mr Alan Bryman and Dr Joe Cullen of the Department of Social Sciences at Loughborough University in association with Professor Trimble. The aim of this work is to look at the experiences of industry in developing in-house expert systems. The work involves questionnaire surveys and interviews with staff from a broad range of companies.

Some of the author's experience of knowledge acquisition will now be described. Much of this material was recently presented at a conference in Washington D.C. but some adjustments have been made to reflect recent findings.

SITUATIONS THAT AFFECT THE ACQUISITION PROCESS

It is clear that the nature of the situation within which the knowledge is acquired will have a major influence on the knowledge acquisition methods to be selected. The categories so far identified are:

1. The knowledge is held in a largely intuitive undefined format.
2. As category I but some closely similar domains have been examined previously.
3. Cases can be defined that reflect a body of decision-making within the domain.
4. There is published material about the domain.
5. The domain expert has sufficient knowledge about expert systems to enable him to define the knowledge (or at least to play a significant role in its definition).

Superimposed on this list of categories are other dimensions such as:

- o The "depth" of knowledge to be represented, i.e. does it represent fundamental knowledge such as that relating to molecular structure or "heuristic" knowledge which includes a substantial amount of personal opinion.
- o The attitude of individual experts to the system.
- o The extent to which a consensus among experts can be found.

The foregoing categories are now elaborated.

Intuitive knowledge

Some knowledge engineers favour a method which requires the development of a prototype system based very often on the prior knowledge of the knowledge engineer. The prototype is demonstrated to the domain expert who suggests modifications and amplification. The changes are made and the revised system demonstrated again. The iterations of this process continue until the domain expert is satisfied that the model is acceptable. If a good initial model is produced this method can be very productive. However it can have the effect of prejudicing the responses of the expert and thus diverting him from some of the subtle, more intuitive knowledge, that might be of crucial importance in the operation of the system.

An alternative is to start with a blank sheet of paper and ask the domain expert to tell you what he knows. A fairly extensive set of knowledge is then assembled before the initial system is coded. This approach is fundamentally better but its success is critically dependent on the time that the domain expert can devote to the process.

Intuitive knowledge with precedents

Where systems have been produced for very similar domains it may be safe to introduce a short-cut in the form of structured interviews based on the content of the previous systems. The danger of prejudicing responses must always be borne in mind.

<u>Defined cases</u>

There are several computer programs that will induce rules from sets of cases. Of these EXPERT-EASE is probably the simplest and best known. At first sight this approach has much to recommend it. However, extensive trials of the early programs have revealed some disconcerting problems. One of these is that the natural sequence of questioning that is inherent in a domain is not respected. For example a pair of questions might be:

o Is the pipe a drain?

o Have you performed a drain-test?

If the sequence of these questions is reversed as it may be in rule-induction the confidence of the user will quickly evaporate. Another problem is that, like regression analysis, rule-induction works on cases irrespective of any causal connections.

It is not, of course, imperative to use the rule-induction program. Manual examination of sets of cases will often indicate relationships that can be coded on an ad hoc basis. The expert will frequently find it easier to recall his knowledge through the recounting of case histories than through other forms of interview.

<u>Published material</u>

There is a lot of interest in the use of expert systems to guide users in the interpretation of regulations and codes of practice. Clearly, in this situation, there should be no problem of human interaction as the views of the human experts should be fully recorded in the published text. As an aside it should be noted that attempts to "computerize" regulations were made before the recent surge of interest in expert systems. These attempts often revealed inconsistency and vagueness which made full "computerization" difficult. Some investigators have suggested that this should be anticipated since differences between the views of the members of the drafting committee eventually have to be resolved by compromise.

Coding by the expert

When the domain expert is also a reasonably competent user of computers it may be possible for him to produce his own expert system without the use of an intermediary. This approach is only possible where the expert exhibits a high degree of self-knowledge and is likely to be unsuccessful where the knowledge is largely intuitive. The expert may or may not be inclined to use an expert system shell to assist him (and constrain him) in his efforts.

SELECTING THE METHOD

The previous section identifies the following methods of knowledge acquisition:

o Unstructured interviews

o Structured interviews

o Case histories

o Prototype system evolved iteratively

o Rule induction

To this list should be added:

o Observational

In the observational method the knowledge engineer observes the domain expert as he performs tasks which require him to draw on his expertise.

o Paper models

We have recently focussed our attention on the use of paper models after seeing the successful use of this technique by the ARIES Alvey Community Club. The knowledge engineer creates a document detailing the rules elicited and develops this as knowledge acquisition progresses. The document records the status of each rule - i.e. finalized, tentative, needs review etc and also records the source from which each rule was obtained. A natural language format is adopted so that the paper model can in many cases be reviewed by the domain experts. We have found that this review process can itself stimulate further knowledge acquisition.

o o o

Our studies started with the view that we should be able to isolate situations (or combinations of factors) that would point to the selection of a single method. Our experience has not borne out this view and for example in one of our applications five different methods were used at different stages. Thus our current advice is to acquire a feel for the alternative methods and to use them flexibly as the position unfolds. The following comments augment those in the preceding section:

Unstructured interviewing has the great merit of not prejudicing the responses of the domain expert. Thus less obvious points emerge that can be very important. The method however is time-consuming and requires patience on the part of the expert.

Structured or focussed interviewing achieves results quickly and is appropriate when the knowledge engineer is fairly confident of his understanding of the domain. This understanding may result from prior knowledge or from the results of earlier acquisition methods. Interviews can be broadly focussed or narrowly focussed. During a broadly focussed interview a knowledge engineer might pose questions such as "describe the parts of a typical boiler system" whereas during a narrowly focussed interview "are there any welding problems associated with carbon steel?" would be more typical.

Prototyping has much to recommend it particularly as each interaction can provide cues to prompt the expert in his thoughts about his intuitive knowledge. As with structured interviewing there is a danger that less obvious points may get overlooked.

The authors have very limited experience of the observational method. However it must be a beneficial approach in providing at least the initial evidence that the knowledge engineer will require in structuring the problem. Furthermore observation can be used to check that the expert performs his tasks in the way he claims to use during interview sessions, i.e. during interview sessions he may be telling the knowledge engineer how he thinks the job ought to be done rather than how it is done in practice.

Rule induction appears to be satisfactory for simple well-defined applications. However for applications even of quite modest levels of complexity we have found that rules prepared by induction are unsatisfactory for direct incorporation in the system. Despite these shortcomings we have found that attempts to apply rule induction to limited modules of a total application can force the expert into considering factors that are not revealed by other methods. This must improve the validity of the knowledge base even if the induced rules are themselves discarded.

SOME HUMAN FACTORS

Knowledge acquisition for expert systems is a human process and several of the human aspects have already been mentioned. The purpose of this section is to itemize the human problems that arise so that readers can be aware of them. This is not to suggest that we can yet offer solutions; the process is likely to remain largely ad hoc for some time. Before proceeding it is worth reminding ourselves that the process of knowledge acquisition is the transfer and transformation of expertise from some source (usually human) to a computer program.

Resistance

The domain expert may fear that, by giving up his knowledge, he will weaken his position within his organization. Unless some incentive can be engineered such an expert is unlikely to provide the basis for a useful system. Organizational resistance may also arise and has been observed in the Community Clubs established in Britain by the Alvey Directorate. For example one club member may provide an expert but then realize that commercially valuable skills could be transmitted via the system to a competitor. It should be noted, on the other hand, that positive motivation may be encountered when an expert is bored with providing personal advice in one subject and would welcome the chance to have this process automated.

<u>Accessibility and prejudicing responses</u>

An expert may have the necessary knowledge and motivation but may have other duties that prevent his spending an adequate amount of time with the knowledge engineer. We have already mentioned the dangers of prejudicing responses by over-structuring interviews and by offering detailed prototypes. However, the methods that prejudice responses are usually quicker so some compromise will often be necessary.

<u>Cues and examples</u>

Experts are often better at doing things than explaining what they are doing and why. So one method of obtaining knowledge is to watch the expert at work and then ask why he did what he did. The problem is that the expert often cannot recall from his subconscious the rules and relationships that have become intuitive. A method that also deals with this problem is to generate artificial examples as cues and to ask the expert what he would do in these circumstances. Our experience in obtaining the Bayes factors for BREDAMP is an illustration of this method (see below).

<u>Rapport and roles</u>

Clearly the knowledge acquisition process will proceed more smoothly and effectively when rapport is established between the knowledge engineer and the domain expert. As a corollary to this, it is usually better to separate the tasks of knowledge acquisition from those of coding the information for the computer. This enables the knowledge engineer to concentrate on the knowledge as perceived by the expert and on establishing a good human relationship with him.

Much of our work has been undertaken in construction industry domains and we believe that our engineering background has enabled us to develop a rapport with the domain experts more easily than would have been the case had we been from an unrelated discipline. However a knowledge engineer in this situation must be careful to avoid distorting the expert's knowledge by introducing his own ideas.

SOME PRACTICAL NUANCES

Development environment

So far the great variety of domains that have been attempted and the approaches adopted have made generalized analysis virtually impossible. However, one conclusion is inescapable, namely that there are at least two distinct kinds of situation namely:

- A client has identified his own need for an expert system and engaged an employee or contractor to deliver a system to the client's requirements.
- An enthusiast, typically an academic, has defined an interesing application and has persuaded host organizations to provide relevant knowledge.

The nature of the relationship between the client (or host organisation) and the knowledge engineer is of crucial importance. Where a client has defined his own needs it is likely that the experts will be readily available and that they will be uninhibited by company policy and commercial confidentiality. They may still be inhibited by their own motivations. At one extreme they may be eager to impart the knowledge in order that the expert system will eventually relieve them of routine tasks which have degenerated into boring chores. At the other extreme they may fear that revealing their knowledge will undermine their security and generate a situation where they become redundant.

Our experience is that where an application is undertaken by an enthusiast the objectives and in particular the intended uses to which the system will be put are often not clearly defined. The experts may also be less readily available and inhibited by commercial considerations. As a result such systems are much less likely to succeed than those which are client inspired. However this type of situation may in some cases have its compensations when the enthusiast himself has substantial domain knowledge.

A possible approach to development

We have found that the approach to knowledge acquisition must be flexible and be specific to the domain under consideration. Our own general approach can be characterised as the following progression:

(i) one or two unstructured sessions

(ii) case histories and broadly focussed interviews

(iii) narrowly focussed interviews

(iv) prototyping and narrowly focussed interviews

In our work the demonstration of a prototype system to the experts has provided a valuable stimulus to further knowledge elicitation. We have found that the point at which the expert first sees this prototype should be selected with care. On the one hand the knowledge engineer may become prematurely over-enthusiastic about the system and as a result demonstrate a piece of code which is trivial, thus losing the confidence of the expert. On the other hand we have encountered an expert who had serious misconceptions about the capabilities of the system being developed, and such misconceptions must be dispelled at an early stage. In the light of experiences such as these our advice is to demonstrate the prototype as soon as it is capable of giving a piece of recognisable and plausible advice.

Following our most recent work we also recommend that a paper model should be developed as a means of documenting the knowledge acquisition process and as a focus for discussion with the domain experts.

Uncertainty

The BREDAMP system generated some useful insights into the problems of uncertain knowledge. This system was commissioned by the Building Research Establishment; its purpose is to diagnose the cause of dampness in buildings. The domain expert was exceptionally cogent and well motivated. However it was necessary to attach probabilities to the goals e.g. to conclude that there was a 90% likelihood that the cause of the dampness was rising damp. For this the dependencies between variables are calculated using Bayes theorem for which affirmative and negative factors must be established. While we could expect the domain expert to describe the behaviour of dampness phenomena it was impractical to obtain from him estimates of the BAYES factors. To overcome this problem we first obtained from him the key factors relating to each cause (or goal). We then compiled tables in the following form and asked the domain expert to suggest values to replace the question marks:

Factor	Suggested values	
Evidence	A stain	A stain
Height of stain	9 inches	I5 inches
Age of building	8 years	9 years
Component wetter inside than out	Yes	Don't know
Positive salts test	Don't know	Yes
Probability of rising damp	?	?

To derive BAYES factors from these data it was sufficient to use an ad hoc approach i.e. a combination of simultaneous equations and trial and error. A better approach would have been some form of regression analysis although it can often be difficult to elicit a sufficient number of cases to make this approach possible.

This case illustrates a further point namely confidence limits. At present BREDAMP offers only a set of probabilities for each of the defined causes of dampness. For example:

Rising damp	90%
Rain penetration	27%
Others	less than 5%

The rising damp figure may in fact mean that the probability is in the range 89-91% or it may mean that it is in the range 80 - 100%. A user would react differently if he had these ranges available. With a narrow range he is likely to conclude that he has gone as far as the system will allow and he may then decide to take remedial measures to cure the problem on the assumption that the cause is in fact rising damp. If the wider range (80-100) is shown he will probably undertake additional, quite cheap, tests to improve the reliability of his diagnosis. This extension of the information provided by a system has been mentioned in several contexts, but no actual implementation has so far been identified by the authors.

CONCLUSIONS

The following are offered as reminders of the key points in this paper:

- o System development should be client led.
- o Knowledge acquisition methods depend much more on situation than domain.
- o Flexibility of approach is essential to the knowledge acquisition process. Factors which will determine this approach include:
  - the form in which the knowledge is available
  - the depth of knowledge (i.e. *fundamental* or *heuristic*)
  - the degree of consensus among experts
  - the attitudes of individual experts to the system.

- It is unlikely that a single method of knowledge acquisition can be adopted for development of an application - rather a number of methods will be required.
- Cues and examples can help an expert to recall intuitively held knowledge.
- Using a computer program to induce rules from cases may provide some enlightenment but is unlikely to provide working rules except perhaps for simple systems.
- Even with a very responsive expert ascertaining Bayes factors is best done by examples.
- Our experience has been that the prototype system should be demonstrated to the experts as soon as it starts to give plausible and recognisable advice. This dispels misconceptions at an early stage and in general provokes further knowledge elicitation and promotes enthusiasm on the part of the expert.
- We have recently had good experience using a paper model as an intermediate representation for the knowledge elicited. This helps the knowledge engineer cope with the mass of information gathered, and with the right experts can itself be used as a knowledge elicitation tool.

FURTHER READING

Welbank, M. A review of knowledge acquisition techniques for expert systems. British Telecom research laboratories, 1983.

Gotts, N.M. Knowledge acquisition for medical systems - a review. Artificial Intelligence in Medicine Group, University of Sussex, 1984.

Buchanan, B.G. Some approaches to knowledge acquisition. Dept. of Computer Science, Stanford University, 1985.

Allwood, R.J., Stewart, D.J., Trimble, E.G., Shaw, M.R. and Smith, J. BREDAMP - an expert system to diagnose the causes of dampness in buildings. Dept. of Civil Engineering, Loughborough University of Technology and Building Research Establishment. To be published shortly.

Cooper, C.N. CRANES - a rule-based assistant with graphics for construction planning engineers. To be presented at 3rd Int. Conf. on Civil and Structural Engineering Computing, London, september 1987.

Wijesundera, D.A. and Harris, F.C. The integration of an expert system into the construction planning process. Proc. of 2nd Int. Conf. on Civil and Structural Engineering Computing 1985, Vol 2, pp399-405. (Describes CONPLANT).