The nature of expertise and its elicitation for business expert systems: a commentary

A. D'AGAPEYEFF
Expertech Ltd

Summary

The knowledge engineering practices of one experienced company are outlined, commenting on differences in emphasis in some of the knowledge elicitation and AI research literatures.

1 Introduction

The elicitation of human expertise that is required for building Expert Systems is widely regarded by AI researchers as being extremely onerous (Hoffman, 1987). Although this view may often be correct it is not universal. An alternative hypothesis is that elicitation will be relatively simple, or rather difficult, or virtually impossible, depending on the expert and the expertise. This alternative offers more hope for business. This commentary presents a portrait of one approach to knowledge elicitation which has had some success in business. It relies on a different kind of expertise, a different style of elicitation, and has different goals to those commonly set for expert system development projects.

There are difficulties in presenting this portrait to a scientific audience. First, it is derived from a particular interpretation of Feigenbaum's early thesis (Feigenbaum, 1971) which is either so obvious, or so heretical, that it does not appear to have been hinted at in the formal AI literature. Second, it has no established theoretical basis, being based principally on *empirical* knowledge, frequently expressed in terms of rules of thumb. It is a feature of such rules that they lack theoretical justification aside from an expectation of probable value. Nevertheless the systems that we shall discuss are largely composed of such rules expressed more or less as they were articulated by a human expert. Third, the evidence for the success of these applications is mainly, though not entirely, anecdotal. They do not appear in the literature because, in Britain at least, such papers are rarely prepared by commercial enterprises who do not wish to alert their competitors. Moreover they are unlikely to be accepted by journals which do not publish applications, or which require evaluation criteria which are so stringent that commercial developers cannot justify the effort required to meet them.

My justification for presenting this commentary is that there is a form of elicitation which can be made sufficiently routine that it has considerable business value. Many research efforts to build expert systems are increasingly emphasizing formal models, deep knowledge, integrity constraints and the like. While our approach may not meet certain standards of rigour, nor amount to "Artifical Intelligence", we claim it works and should not be neglected through an overpreoccupation with difficulties.

2 Programming from know-how

Much human knowledge is only expressed dynamically but, I assert there is a form in which it is held ready for articulation. In common parlance this is called know-how. It reflects a unique human ability to share skills via verbal communication, thereby allowing communities to accumulate the experience of the individuals who discovered it (and who may of course be dead). During the greater part of our history, verbal communication has been the sole means of technical exchange other than physical demonstration. Indeed it is possible that the tradition of verbal transmission of know-how has existed as long as *homo sapiens sapiens* (Foley, 1987) which suggests a tradition of over 50,000 years. Since the invention of printing of course, this has changed but the implicit assumption that such exchanges have largely been replaced is not correct, even in highly "technical"domains. As we all know, printed versions of the methods by which humans like to pretend they solve problems or reach decisions can amount to mere *post hoc* rationalizations. The know-how people really use and the expertise it embodies, may never have appeared in print.

Among the reasons for this are that know-how is not naturally elicited out of the context of its use. The main focus of this commentary is that the effective elicitation of know-how requires sustained prompting within the problem solving setting. The informal methodology described below has emerged from a few hundred applications built by ourselves or our customers. While our method is still evolving and requires refinement, application building is now routine in our user community.

Programming from know-how was pioneered by Unilever Research Department and by Chemists in ICI from about 1981/2 onwards. Their aims were, among others, to supplement research reports and corporate manuals (which are as experience shows seldom thoroughly studied) with digestible chunks of practical expertise. They noticed that elicitation for such applications was not a major problem when the expert liked the software tools offered and believed in the project. They therefore began to develop an empirical, practical routine of refining applications by prompting the expert as mistakes in the program emerged. Their efforts were unpublished (though they were generously explained to the author during an Alvey Survey (d'Agapeyeff, 1984)), but both companies have subsequently developed a broad range of expert systems that embody many kinds of human expertise.

In the autumn of 1984, our company was launched with the aim of building on this work by developing and supplying tools for building know-how applications on personal computers. The need to prompt actively for the recall of know-how was treated as paramount from the outset. As part of this objective it was critical that the software should enable the application to be intelligible to everyone concerned e.g. the sponsoring manager, the expert, the knowledge engineer and the users.

While intelligibility, explanation etc. have been emphasized from the beginning of expert systems, commercial product development has in fact increasingly concentrated on programming facilities. The efforts to develop more and more sophisticated programming tools, systematic knowledge engineering methodologies etc. have, it seems to me, been at the price of the human expert being effectively locked out of details of the knowledge engineering process. While sustaining the intelligibility of a knowledge base can be very hard work we believe it is critical to the success of the knowledge engineering effort.

The commentary reflects our experience since 1984, much of it gleaned from customers. Although it is a salutory thought that the knowledge we have elicited is inferior in quality and quantity to that held in the head of a single expert, my colleagues have collectively, probably seen more know-how written down than any other group in history. My purposes are to remind the reader of Feigenbaum's emphasis on empirical know-how, to argue for a renewed enthusiasm for this undervalued resource, and to emphasize the importance of maintaining intimate involvement of human experts throughout all phases of application development.

2.a *Some applications*

Our preferred description of know-how (Feigenbaum, 1971) is that it is the empirical expertise acquired from the experience of doing a job. This expertise determines the overwhelming majority of decisions taken by veterans in that job. In an early and *ad hoc* experiment at Expertech, 18 general practitioners were induced to spend a weekend recording their know-how on the diagnosis of four routine but not trivial medical conditions. At the end of the weekend they concluded that, once focused on explicitly programmed rules, they could agree on draft knowledge bases. The knowledge bases were far from complete but the experiment seemed to show that given the right motivation, elicitation need not be a serious problem and was felt to be instructive by the experts involved. This has since proved to be the normal position in the business settings we work in.

To give a flavour of the kinds of applications which are the basis of our experience here are a few examples:

1 A Finnish Bank, OKO, has a branch-wide application providing decision support for loans to farmers. This required about 3,000 rules (in 25 chained knowledge bases) (Kontio, 1988), and has been followed by two further branch wide applications by different Finnish Banks.

2 The Friends' Provident insurance company have announced an application to assist their tied agents to comply with the UK Financial Services Act. It is on 400 sites and is expected to rise to over 1,000 in 1989.

3 Ernst and Whinney, the international chartered acccountants, have an application for their auditors involved with IBM System 36 computers (it is a kind of dynamic check list on security and related topics within Ernst and Whinney's standard procedures) which is being distributed to their 200 offices worldwide.

4 British Petroleum have a burner (fault diagnosis) application, which links to graphical displays picturing different flames, that has been distributed to their 30 worldwide refineries.

2.b Some general observations on know-how

We are not professional researchers, but our experience constructing some 2,000 knowledge bases in a single family of software is a potent asset. These knowledge bases are not all complete or externally available but enough are (Expertech, 1988) to attempt to analyze them, together with the experience of eliciting their contents, to try to identify some lessons (though admittedly with the full benefit of hindsight). Some notable properties of know-how which are suggested by an analysis of current applications are:

1 Most rules of thumb are flawed to some extent, but conclusions drawn from a sequence of such rules seem to be better than an accumulation of those flaws would suggest. This presumably results from possession of a hugely redundant library of know-how, including control rules and rules for checking intermediate results (which are seldom articulated) as well as the obvious ones for producing solutions.

2 Experts have contradictory rules of thumb which may be unnoticed until the contradictions surface within a single context. In such circumstances the contradictions are resolved by use of either special case or meta-rules.

3 Experts are rarely conscious of the combinatorial potential of their problem solving. As has been observed in the literature this is probably because they rely on match (directly linking actions to frequently encountered situations) rather than search in a problem space. If search is employed it is narrowly focused and not exhaustive.

4 Although people have many well-rehearsed rules including some they can quote from childhood, they seem willing to modify many of them given fresh evidence. The building and use of know-

how applications can be very rewarding to those engaged in it; the clarification of one's own understanding can be an important motivation.

2.c Working with experts

It is something of a mystery to us how the notion originated of experts being inarticulate, argumentative and quickly bored. These conditions are always possible but they are very infrequent. An authoritative expert, confident of his know-how, comfortable in his surrounding and adequately prompted will exhaust any knowledge engineer. The know-how does not, of course, emerge in a tidy sequence nor, typically, in a neat structure, but viewed as a raw material it is produced in volume. (Achieving sufficient completeness for it to meet a specific purpose, and refining and polishing an application for target users, are of course separate and substantial, undertakings.)

One major factor in this fluency is motivation. People are fascinated by know-how (d'Agapeyeff, 1986), not least by their own. They will not have seen their expertise recorded before and been able to explore it for themselves. This can be a very exciting adventure. Some observers declare flatly "you will never get experts to part with their expertise". This has never been our experience. The one qualification we would make is that the expert must be given the opportunity to become a willing and intrigued volunteer. If the expert has been conscripted by managerial edict then the project is, very probably, doomed.

Motivation is also important because it is absolutely essential that the know-how reflects what the expert really does, and not what he may believe will make him look impressive or rational. Users who have nothing like the same expertise, will be quick to spot and very disparaging of pretence or propaganda. At the same time they will readily tolerate the odd genuine mistake in an otherwise valuable application. A pretence of plausible expertise can be a temptation for a newly engaged expert but it soon disappears. People are quick to acknowledge that the convention of writing up their decisions, as if they stemmed from a neat progression of logical steps, is a courtesy unconnected with reality. (We have found, for example, that civil servants with a lifetime devoted to discretion, have been amazed at the frankness of their colleagues in expressing the realities of their decision making on sensitive topics.)

Frankness is also vital to help determine the boundaries of a contributor's know-how, and to establish how the expert would tackle a particular situation within his skill but outside his direct experience. One conventional account is that the expert will have recourse to deep knowledge, using a model of the underlying causal or other qualitative structure, relevant to that class of situation. This does not sit very well with know-how where the most profound understanding of a domain is reflected in quite general but nonetheless empirical, knowledge of rules of thumb and relationships. (In a certain sense the AI "weak methods" for problem solving, such as divide and conquer, hill climbing, means-ends analysis are very general examples of such empirical rules, though most expert knowledge tends to be more practical and application-specific.) If no such rule is available then it seems that experts use relationships that have just been found to work in the past; these can be thought of as a "model" but they are empirical rather than causal. For example, in fault diagnosis the relationships link elements in a baffling set of symptoms to possible faults, to areas for further investigation and/or to actions which will simplify the problem (e.g. by excluding certain types of potential fault).

Many researchers, traditional data-processing staff, and others have serious doubts about whether empirical know-how is a reliable commodity, worthy of sustained attention. "But surely" they worry "this 'know-how' is nothing more than prejudice?" The sceptics are unlikely to be impressed by the honest reply "how true, but it is a prejudice that is derived from practical experience". "But would it not be better to employ logic, or a carefully designed algorithm?" Here the reply may be less clear cut. If an algorithm is available and appropriate then obviously, it should be employed. Those who work with know-how have no doubt that it evolved to serve very rapid decisions, embodying compromises, for noisy, even contradictory situations which are often incompletely described by the available data—but business life is full of that kind of problem.

Interestingly, in our experience it is not managers who are the sceptics in our customers' companies. This is, perhaps, the primary justification of know-how applications. Logic and mathematics are not applicable to business situations where the data are insufficient, but these situations exist and even pragmatic computer solutions appeal to management.

2.d Notes to elicitors

2.d.1 Initial applications

A company's initial set of applications should be selected with the greatest care. They have to serve a real need because nobody will take any notice of toy solutions. The problems should not be too big or too complex since most, if not all, of those involved with the implementation will not have had prior, pertinent experience. Indeed the primary benefit of these applications will be the lessons that can be learned from them, right across the company.

2.d.2 Human decision-rules are subtle, prejudiced, unrecorded—and important

The know-how actually employed in much human decision making had not been recorded prior to the emergence of expert systems. There are solid reasons for this. The detail of know-how is not recalled by an effort of will, by a simple interrogation or by pecking at a typewriter. Instead it requires deliberate and sustained prompting, preferably in context, to elicit enough of the know-how to support the chosen task. The hardest point is getting started, but once an application runs, it itself supplies an effective prompt to further refinement.

Humans have an arsenal of short cuts and usually an inexhaustable supply of checks and filters on their solutions. These are often subtle and not routinely volunteered, but are nonetheless real and important. Their identification can sometimes be prompted by involving the expert in thorough testing of prototypes with many combinations of data (it is remarkable how a new combination of circumstances that causes an error in inference can provoke the expert to voice, without apparent thought, a rule involving factors that are entirely new to the knowledge engineer).

Precision is not central to human know-how, often for good reasons. An early prototype application by a financial institution, dealing with the law on small estates, reduces the required formalities (e.g. probate) where the value of the estate is less than 5,000 pounds. A traditional program would typically follow that law exactly. But that is not what an expert would do. Since the bank has safeguards up to 5,000 it might as well treat the estate in the same way when it amounts to, say, 5,200 (or possibly more).

Know-how may not obey theory. In medicine, general practitioners are under strong pressure (by the queue in the waiting room) to make a quick diagnosis. They seem to come rapidly to a conclusion and then conscientiously (for the most part) check that it fits the symptoms. This may mean the first diagnosis is produced from surprisingly few rules and then requires a large number of checking rules. This is not an approved strategy from a theoretical perspective, but again it is real and important to understand in the context of a medical service with many built-in safeguards; I suspect that most experts act in the same way but initially they may be reluctant to admit it.

In a very different application dealing with physical abuse of children, a rule intended as part of a system for alerting school teachers was also revealing. The rule more or less as uttered was:

> if suspicious condition is repeated
>   and the child's account is implausible
>   and the bruising has a pattern
>   and the site of the bruising is inside-of-limbs
> then social services should be alerted

One might have imagined that a child would always wish to be protected and, in a secure setting, report unhappy circumstances. Not so; apparently the slogan to remember is "a child will lie to defend her abuser when, say, that is the only daddy she has got". The "obvious" may be as misleading in human judgement as received theory is in technical judgements.

The general point of such examples is that the knowledge and assumptions which lie behind elicited rules of thumb are often subtle and prejudiced, and will invariably be unrecorded. But that knowledge and those assumptions should be reflected in the application—and in the elicitor's whole approach to the application.

2.d.3 Know-how applications should be practical, relevant and rewarding

Solemnity is boring and a handicap to communicating with people. Among the more boring documents, in the majority of companies are their safety manuals. (These often achieve nothing except to prevent companies being prosecuted for not having them.) But the law does not require that they be readable, nor read, nor that they be effective. But safety is about know-how, and communicating that know-how to others. A know-how application has to relate to *practical contexts* and address *specific people*. Expertise is then required to determine what a sensible person would do in those contexts. Systems of this kind are only a supplement to, and not a replacement of, existing manuals. Many are pretty slim affairs (few companies bother much about safety) but ICI has serious safety applications on corrosion and thermal insulation.

Another widespread boring manual is the pension scheme or regulations manual, with the result that very few employees have any understanding of their pension scheme. Some companies have endeavoured to increase this understanding by providing know-how on the meaning of the rules within a particular perspective (e.g. old stagers really want to know what they are going to get and when). Following recent changes in the UK law, Minet, and no doubt other pension consultants, have built a know-how application to offer suggestions tailored to the input of an individual's exact circumstances.

Since 1985, clarification of regulations by know-how interpretations of the way they should be applied in specific contexts has become a major activity. Robson Rhodes have a product clarifying the UK law on dismissal from employment and on maternity. A Building Societies Expert Systems Club has been set up to tackle the new rules on compliance under the British Financial Services Act. Various companies are tackling their regulations on the packaging of toxic materials, on sales and brokerage commissions, on export paperwork and many other tasks.

2.d.4 Know-how can be refined and pooled

The know-how of an expert can be improved by his own reflections on using the software. However, this seems to depend on many influences such as the degree of involvement, and the skill of the knowledge engineer. The appropriateness of the software also has a role but hitherto this has been a matter of chance. Only recently has software begun to be designed with this aim in view (e.g. browsing facilities for the expert and studies of retrieval by "soft" matching).

The more clear-cut gains are achieved by peer groups of experts. Again this is a recent development. Among the most fruitful applications of this type has been assistance to help desk centres in DP Departments (handling calls from users about their terminal, word processor or whatever, being faulty). It has been widely presumed (but notice on almost zero experience) that experts would differ, and therefore, multiple simultaneous contributors of know-how was a complexity best avoided. Then it was discovered that peer reviews of a draft application could be popular and very productive. The discussion, when focused on individual rules and sequences within a well-understood context, benefitted the participants and refined the programmed expertise. Gradually knowledge engineers have become more confident and frequently encourage this joint involvement (although the first drafting may continue to be done with a single expert).

3 The expert

Attracting and recruiting the commitment of the right experts is the essence of obtaining high quality know-how. Above all it must constantly be remembered that experts are people with their individual wants, needs, and styles of work. An experienced knowledge engineer must therefore have a range of tools and strategies which he will tailor for a given project, but mostly for a particular expert.

Occasionally the know-how may be too intangible for it to be articulated in a way that can be programmed, so planners must be flexible enough to revise the project. It may be possible to vary the scope of the application depending on the type and quality of the available know-how. In business there is usually room for flexibility, and willingness to exercise it during project management is crucial.

3.a Attracting the right expert

It has been said that the right expert is authoritative about the practicalities of the task, motivated by the project, articulate about his expertise and available to assist the implementation. The practicalities can be a cause of difficulty. Management frequently suggest the name of an (undoubted) expert from head office or from the research department. This person may be an excellent reviewer of an application, once written, but does he know the wrinkles employed by practitioners? Richard Perkin of the UK National Coal Board observed that during fault diagnosis those who have been long familiar with conditions down a mine go to extreme lengths to prevent dirt getting into control units because if a unit has to be opened then the entire equipment (e.g. a coal cutter) may have to be taken out of the mine for cleaning. This problem may be less thoroughly appreciated at head office.

Having identified a suitable expert he must then be intrigued by the enterprise. Notwithstanding the debate about induction techniques in knowledge acquisition, British Petroleum often find induction of value in this early stage. They ask for sample cases, induce a few rules, discuss them with the expert, and make changes within a few minutes. Others show off a more developed knowledge base and make changes to it in line with the expert's suggestions. In both cases the intention is to lure the expert into becoming a contributor before he has a chance to worry about the difficulties. He may then be interested in the process of recording his know-how and in its results; and on this basis a bargain can be struck.

3.b The bargain

The concept of a bargain between the expert and the knowledge engineer is beneficial. A bargain is not inevitable but in the majority of cases can be established and maintained. The knowledge engineer wants to obtain a proportion of the expert's expertise. In return, a rewarding new experience is being offered. Emphasis must be placed on the importance of the expert's know-how and he must be given the opportunity of becoming a full partner in the enterprise. The bargain is seldom formal but it must be recognized and explicitly pursued; the relationship between the parties must be on the basis of mutual respect and trust.

A sound bargain is not always achieved. Sometimes the relationship does not work but may be repaired by having a different knowledge engineer. Sometimes the expert does not want to proceed and a substitute must be found. A fully committed, if second class, expert is a possible substitute; further expertise may be obtained subsequently from other sources whereas an over-extended knowledge engineer may be fatal to a project, because the organization has no criteria for judging and improving the knowledge engineering.

A bargain may be threatened if the expert becomes disappointed over the limited amount of expertise that can be practically programmed. He may think, for example, that he could explain his use of, and basis for, analogies in his problem solving—until he comes to try. On the other hand he may discuss all examples as special cases making no comment about the generic rules that he has at his disposal. Although the target users may never have heard of these, the expert may regard them as blindingly obvious and not worth mentioning. Dogged pursuit of the blindingly obvious can turn an expert off but can be vital to success and must be carried out with tact.

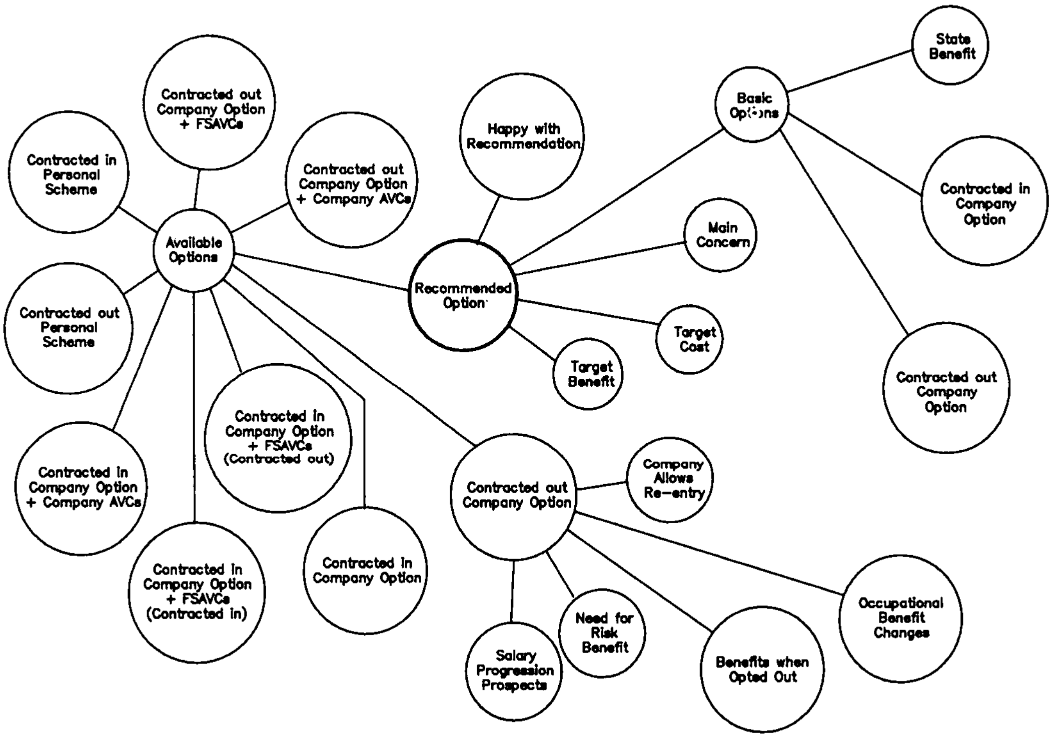

Figure 1 A spider diagram.

4 Strategies of knowledge acquisition

The knowledge acquisition strategies we use will be quite familiar. Perhaps our experience is just more encouraging than much of that reported in the research literature, although we do have a more restricted interpretation of expert systems. At any rate the main method of knowledge gathering is carried out by prompting the expert's recall, which we return to below, but other strategies, viz induction, deduction from interviews, and conventional programming are important aids.

4.a Interviews

We find that unprompted interviews with an expert, which do not incorporate any feedback, are a dubious means of eliciting rules of thumb. However a start has to be made somewhere and many experts prefer to begin by talking informally about their expertise. This may lead to the first outline of a knowledge base. Yet there is more to elicitation than the gathering of rules. The interviews have to establish many things, including:

- an initial description of the nature and structure of the problem
- the approach to be adopted
- the usage options to be provided (e.g. for new, regular or occasional users)
- whether any tutorial option capabilities are needed, and so on.

To document these discussions we have developed what we call "spider diagrams"; a network of nodes and links showing various important relationships at different levels of detail. For example at the top there may be a summary of the advice to be provided by the application. The nodes will be the generating sources (e.g. knowledge bases) and the links the outputs produced. Other diagrams may show the relationships between component knowledge bases (an application may have many chained together) or the conditional inference structure within a particular knowledge base. An example is shown in Fig. 1.

Spider diagrams continue to be used throughout knowledge engineering. They can become untidy as they are drawn, modified, modified again and redrawn. We have not found a suitable charting package which will allow sufficient flexibility for repeated alterations, but there are of course packages which can reproduce poor hand drawn diagrams so that they are clear enough for management to assess and for documentation of the program.

4.b The backbone: prompted elicitation

Broadly the major part of our knowledge engineering routine involves the following activities:

- preparation
- groundwork
- prompted prototyping
- establishing generalized and checking rules
- private review and extensions
- peer review

4.b.1 Preparation

The preparation for elicitation starts with the knowledge engineer briefing himself about the selected task domain. (The more the knowledge engineer knows about the domain the better, but the real objective of this briefing may be more mundane: it prevents him appearing to be a fool through having to constantly interrupt to ask about fundamentals.)

In contrast it is also necessary to brief the expert about know-how, about the software, and the typical stages of an implementation. Experts vary in what they want to know, with those at the extremes wanting to appreciate everything involved in the knowledge engineering and those who only see themselves as a source of expertise.

4.b.2 The groundwork

Our work typically involves providing a number of services for the customer, within a single package. We therefore need to establish the necessary knowledge bases that will be needed within the final system, and the uses to which they will be put. The next step therefore, is one or more interrogations with the expert, about the general shape of the application and the main classes of transaction. (Here the knowledge engineer takes care not to act like a lawyer facing a hostile witness; this easily happens.) A draft outline of possible knowledge bases and a tentative listing of required transaction types are sought. Although I describe it as groundwork these activities can recur throughout a project. An experience which lingers in memory is to hear a chance remark about "blue widgets" during the last stages of a project. This was the first time the knowledge engineer had ever heard of the existence of such widgets. The question "is that the complete list of transactions?" is a regular refrain during elicitation.

Early on in the project, rules etc. are elicited well away from the machine; a blank screen is off-putting for a novice. Discussion about the application may even have the knowledge engineer acting the part of an inference engine! Alternatively a draft knowledge base may be obtained for a junior expert from the same department. In fact some DP departments have composed the initial knowledge bases from their own resources entirely, which sounds dangerous but can be successful.

During the past two years we have also accumulated experience with a conventional, rule generating induction package (Xi rule, Attar Ltd). However, the package produces its output in the syntax of Xi Plus, a rule-based expert system shell, as well as a decision tree, and the output can therefore be executed directly as part of other Xi Plus applications. This had lead to a practice somewhat different from that envisaged by Michie in his early advocacy of induction techniques (Michie, 1982) because applications are rarely built entirely by induction. The induction package is treated simply as another tool available to knowledge engineers.

We have come to recognize a number of important uses for the induction package. First it may be used to build a pilot application as part of the recruitment of an expert (see above). Second it is an option when many attributes are simultaneously relevant to a decision (the rules are big) and there is difficulty in drafting the necessary rules, or in checking those already drafted. Third, induced rules can be treated as an input to the expert during routine knowledge engineering, both as an additional prompt and as a check on his/her own offerings (with larger applications this can be very important).

4.b.3 Prompted prototyping

Nearly everyone finds it easier to be a critic of existing rules than a creator of new rules. Thus there are advantages for a novice contributor of know-how starting by reading and correcting a knowledge engineer's rules. The new prototypes are then input for testing against new data. This can be an arduous time for the knowledge engineer. He must listen to the expert, strive to phrase the rules so that they both understand them, be aware that he is actually demonstrating how the software can be exploited (and hope the expert will eventually do some of this for herself) and add to the effect of the program in prompting the expert's memory recall. The program is run, makes mistakes, which prompt for correction, the rules are revised and (immediately) re-entered. The program is re-run and the cycle is repeated.

We believe this sort of prompted cycle really does work because it is now a daily routine for many project teams. The rules are often special purpose, i.e. relevant only to very particular contexts, but they pour out and the causes of mistakes can be readily identified. This will get a little more difficult when the number of rules in a given knowledge base grows more numerous. Some experts persist in concentrating on a given subtask until they are satisfied it is complete while others dart about articulating rules on different topics as they are recalled. During the intervals away from the expert, the knowledge engineer will draft clarifications and refinements, and do some volume testing on his own.

4.b.4 Robustness: generalizations and rules for checking

During the early period, at least until the expert is comfortable with the elicitation process, most knowledge engineers will simply follow the expert's lead. The sterner parts of the elicitation are postponed. Special purpose rules are however seldom sufficient and it is important to aim for generalization where possible. By identifying a more general form for a set of special purpose rules the knowledge base is made more compact, but more important the application becomes more robust and flexible from the user's perspective (because the general rules cover classes of cases, individual instances of which may not actually occur in the examples being used). Identifying the generalizations is not, however, always easy and sooner or later the knowledge engineer will have to take on a more positive role in directing the elicitation.

There is a spectrum of generalizations. There are those which seem to be held in an explicit form; they require little or no prompting. We refer to these as "generic" rules; a good generic tends to stay intact throughout the project while other rules are repeatedly changed and merged. To take a graphic (if unfortunately sexist) medical example there is the common adage that the typical gall stones patient is "fat, female, fertile and forty". While a great oversimplification, the adage does have some truth and did prove to be a productive focus for extending part of the knowledge base.

To be fair there is some controversy among knowledge engineers about the value of such generic rules. Some find them helpful for explanations but little benefit for inferencing. For example the principle behind double-entry book-keeping namely:

> if a transaction has a value
> then credit that value to the transferrer
>   and debit the recipient

may convey insight to humans but cannot easily be employed in an inference precisely because it is very general; there is insufficient contextual detail.

In contrast there are generalizations which are implicit in an expert's reasoning and which the knowledge engineer may have to actively draw into a rule form. Elicitation deliberately aimed at generalization proceeds from a narrow focus outwards. Repeatedly applying a rule set to new cases will occasionally prompt the expert to recognize generalized forms which will replace many specialized rules. In prompting for generalized rules and inferencing patterns the knowledge engineer may rely on a combination of his own questions and on feedback from running the program with varying data. Repetitive prompting seems to be the key for generalized rules.

Most applications require rules for checking intermediate results, and/or controlling inference dynamically. In some expert systems such rules seem to be little more than artifacts of knowledge engineering. In our view, however, these should as far as possible also be regarded as objects of elicitation. People seem to have extraordinary abilities to check their work, though relatively few checking rules seem to be part of the tradition of spoken expertise, which makes elicitation harder. The experienced knowledge engineer can, however, prompt at the right moment as to the reasonableness of intermediate results either to propose, or elicit, a rule for detecting a possible error. (Similar remarks can be made about control rules.)

4.b.5 Private and peer reviews

The moment will come when the expert will need to stand back and review the draft application, possibly after quite a short interval after the direct elicitation of rules. It will be necessary to reconsider assumptions about the structure of the application, and the generality of the compiled knowledge, from the standpoints of usage, omissions (whether because topics are outside the direct experience of the expert or simple errors) and so forth.

In some cases the expert can do this on her own; in others the knowledge engineer may bring to light necessary changes in assumptions, but these will be exceptional, and peer reviews are highly recommended. To carry out peer reviews successfully however, a climate of opinion must be cultivated where this is seen as a normal procedure and not a criticism of the expert. When peer reviews are most successful many of the participants say they have also benefitted from a greater grasp of their joint know-how, and may indeed have extended their combined expertise.

5 Conclusion

Empirical know-how is a real and potent resource from which to build an expert system. The prime aim of our approach to expert systems is to communicate know-how which has been checked and refined between people. In general, people have a substantial, if latent, ability to articulate their know-how in a form that will assist others providing their memory recall is appropriately prompted. They can be successfully prompted without great difficulty for narrowly focused tasks and where the subtasks can be readily identified.

The heart of our informal methodology is one in which the expert is intimately involved in the development process. In consequence we believe that if the software does not make the knowledge representation intelligible, or makes it difficult for the contributing expert to express knowledge directly, more or less as he would naturally (or with a little help), then he cannot ensure it adequately reflects his expertise in appropriate contexts. He will therefore be unable to check the accidental, or deliberate, alterations made by the knowledge engineer.

During routine knowledge engineering we place great emphasis on:

- intelligibility of rules and the inference process
- eclectic use of knowledge engineering tools and methods
- creating and maintaining a close relationship between expert and knowledge engineer
- continuous refinement by repeatedly exercising and testing the application working directly with the expert (prompting in context)
- a pilot trial—before going live it is important to check with representative users that they find the program relevant, interesting and rewarding.

Some techniques of elicitation which are being discussed in the research literature do not seem to reflect these emphases, suggesting as they do that one can, as it were, simply analyze a set of cases or a set of interviews in order to produce a body of knowledge which can then be encoded (frequently in an unintelligible computerese). In our opinion elicitation is in essence a journey of discovery, not a design process. Perhaps some of the reported difficulties of knowledge elicitation reflect failures to act on such unremarkable, but easily forgotten, observations.

References

d'Agapeyeff, A and Hawkins, C J B (1984). "A short survey of expert systems in UK business". Supplement to Alvey News Issue No. 4.

d'Agapeyeff, A (1986). "The sharing and computer exercising of human know-how" *British Journal of Radiology* **59** (707).

Expertech Ltd (1988). *101 Uses of Xi Plus*. Slough.

Feigenbaum, E A, Buchanan, B G and Lederburg, J (1971). "On generality and problem solving: a case study using the Dendral program" *Machine Intelligence 6*, Edinburgh University.

Foley, R (19). "Hominid species and stone tool assemblages: how are they related?" *Antiquity* **61** (233).

Kontio, J (1988). "Matias, OKO Bank's loan analysis expert system" Proceedings of the BANKAI Conference organized by SWIFT, London, March 1988.

Michie, D (1982). *Introductory Readings in Expert Systems*. Gordon and Breach, London.