In: Research and Development in Expert Systems VII, pp147-159,
Eds: T. R. Addis and R. M. Muir, UK: Cambridge Univ Press (1990)
http://www.cs.utexas.edu/users/pclark/papers

REPRESENTING KNOWLEDGE AS ARGUMENTS: APPLYING EXPERT SYSTEM TECHNOLOGY TO JUDGEMENTAL PROBLEM-SOLVING

Peter Clark (pete@turing.ac.uk)
Turing Institute
36 N. Hanover St.
Glasgow G1 2AD.

Abstract
In many domains, it is not possible to easily gather a definitive body of expertise for problem-solving. A particularly well-known problem is that, when experts disagree, it is not easy or even possible to identify the 'right' answer, a characteristic particularly true of problems involving human judgement. In such domains, the process of argumentation between experts plays a crucial role in pooling knowledge, locating inconsistencies and focusing attention on areas for further examination.

In this paper we recognise this process of argumentation – reasoning about *why* inferences are valid – as important in problem-solving, and present techniques for using AI technology to assist experts in this process. In the implementation we present, models of different opinions are represented separately rather than combined. The user and system interact to solve a given problem, the system arguing its case on the basis of consistency with previous decisions and the user supplying extra knowledge which the system is unaware of. Dialog focuses on the meta-level justifications for inferences which are made, not normally represented in the usual expert system methodology. By exploiting the computer's power of memory and search a powerful decision support tool can be constructed. We illustrate this with a full-scale system, named Optimist, for assisting geologists in oil exploration.

1 Introduction

Many expert systems are constructed on the principle of assembling a body of expertise, representing it within the system and then using it to solve the task at hand. Domains where a well-defined body of knowledge exists are particularly suitable for this methodology as reflected by applications which have been successful [Andrews, 1989], and criteria of having a well-defined domain and agreement among experts have become established guidelines for selecting applications (eg. [Buchanan, 1985, Prerau, 1989]).

However, it is also recognised that not all domains are as 'neat' as to allow a definitive body of expertise to be easily gathered and then straightforwardly applied. A particularly well-known problem, which is the topic of this paper, is that in many domains there can be considerable variation of opinion among experts due to their differing knowledge and judgement. Experts sometimes disagree with each other and may change their opinion over time and with experience, and an expert's personal knowledge of previously solved problems can strongly influence his or her solutions to new problems.

In this type of domain, human problem-solving additionally involves retrieving similar previous cases and discussion among experts. The process of **argumentation** between experts serves to pool knowledge together and enforce consistency, and plays a crucial role in many problem-solving tasks. This feature is particularly prominent in tasks involving risk assessment [Alexander and Evans, 1988].

It is also clear that the value of such discussions in solving problems is dependent on the availability of experts, and their knowledge of and ability to recall relevant previous cases. Assistance in this process would thus be valuable, and as Reboh points out [Reboh, 1983] expert system research offers precisely a technology in which this expert reasoning can be captured and, by combining this with the computer's power for faithful memory and extensive search, provide a useful decision-support tool.

In this paper, we highlight the problems in constructing such a decision-support tool and present the techniques we have developed for dealing with them. These techniques are based on the modelling of differing expert opinions within a single system, and then using these records to involve the computer in the argumentation process. This work is framed in the context of risk assessment in oil exploration, and a full-scale implementation named Optimist is described.

2 Problem Description

2.1 Risk Assessment in Oil Exploration

We briefly describe the salient characteristics of expert reasoning in our illustrative domain to highlight the problems which our techniques address.

In the search for hydrocarbons such as oil and gas, the sites of potential prospects need to be appraised to assess the probability that hydrocarbons are present, and if so in what quantities. This task of **prospect appraisal** is formidable. Typically it takes a small team of highly trained experts several weeks of work, involving the sifting of a large amount of data to produce a final estimate of the probability of there being hydrocarbons present.

Some of the characteristics of reasoning in this domain are as follows:

1. Discussion or 'argumentation' among experts plays a central role in problem-solving, involving the retrieval of similar previous cases and forming hypotheses about unobservable features of the prospect. This process serves to pool knowledge together and enforce consistency, and is essential in risk assessment.
2. Experts frequently disagree; it is often impossible to identify the 'correct' answer, as the accuracy of probability judgements can only be assessed over sets of cases rather than in any single case.
3. Retrieval of previously drilled, similar cases is essential to making accurate risk assessments. The cases form a focus for discussion.

As a reflection of the differences in expert opinion in this domain, Table 1 presents some of the prospects appraised by different experts. As can be seen, there is often substantial disagreement.

Table 1: An illustration of differing expert opinions, showing assessments of the likelihood of finding oil at three real-world prospects by different experts (each figure is the conclusion of a detailed appraisal process).

	Opinion 1	Opinion 2
Prospect 1	72%	40%
Prospect 2	9%	17%
Prospect 3	1%	7%

2.2 Other Domains

The characteristics of disagreement among experts, the need for discussion and the important role of similar cases also hold for a large number of other real-world domains, particularly those where judgemental reasoning is involved. Some examples in the AI literature are business decision-making [Premkumar, 1989], financial risk analysis [Alexander and Evans, 1988], financial marketing [Castner et al., 1986] and legal reasoning [Rissland et al., 1984]. A particularly interesting case was the development of the Prospector system [Duda et al., 1981], where its encoded models were originally developed with and tested by a single (same) expert. Towards the end of the project, it became apparent that experts did often disagree, and Gaschnig optimistically raised the hope that discussion about Prospectors' models would smooth out the disagreements [Gaschnig, 1982]. Finally, in 1983 Reboh argues in favour of modelling (rather than seeking to eradicate) conflicting expertise and speculates on some of the mechanisms for doing this [Reboh, 1983]. The techniques which we present are thus also relevant to other domains such as these.

2.3 Applying Expert System Technology

The above problem characteristics make it particularly difficult to apply the normal expert system methodology of identifying and encoding the 'right' expertise for solving the problem. Instead we have had to develop alternative methods for assisting in problem-solving in this domain. In particular, we have made three atypical design decisions:

1. Conflicts between different opinions are not resolved during knowledge-base development, but instead the **different opinions are represented** separately within the system. Often the different opinions and methods of different experts constitute substantial domain-specific knowledge, and the location of conflict an important method of focusing attention during problem-solving [Klein and Lu, 1989].
2. The system's role is not the usual one of autonomous, explainable problem-solving but instead is to provide **decision support** by identifying, retrieving and presenting relevant information to the expert by 'arguing' with him or her. For knowledge-based systems to assist human *experts* rather novices, a decision support role is often more appropriate than one of autonomously repeating what the experts are already good at [Mittal and Dym, 1985].

3. **Specific, previous cases** constitute an important source of knowledge for problem-solving as well as that in a general knowledge-base. The importance of case records is also reflected in a number of other expert systems, where cases act as 'anchor points' for guiding decisions and knowledge-base evolution (eg. Garvan-ES1 [Compton et al., 1989], Prospector [Duda et al., 1981]).

The basic principle underlying these design decisions is that, in some domains, expert reasoning should be viewed as the processing of **arguments** rather than the application of a well-defined body of knowledge. We now briefly elaborate on these design decisions.

3 From Rules To Arguments

3.1 Meta-Level Knowledge for Justifying Arguments

We can define an argument simply as an inference (or chain of inferences) whose validity is open to question. Reasoning with arguments includes the use of **meta-level** knowledge about the justification for inferences being used, and thus this meta-level knowledge must be represented. This is in sharp contrast to the normal expert system methodology where the justification for a rule's validity is not represented. While in a normal system the proof of a rule's conditions can be queried, the justification for the presence of the rule itself is not represented. For example, consider a hypothetical rule

```
IF hightemp THEN meningitis cf 0.2.
```

The justification for expecting a weak correlation (`0.2`) between `hightemp` and `meningitis` is inaccessible to the user; this statistical and/or theoretical evidence is not represented in a normal expert system approach as the user is not expected to question inferences. However, in real-world problem-solving, reasoning *about* the justifications for inferences plays in important role.

3.2 The Dynamics of Argumentation

In the system we present, argumentation is incorporated in a framework of **interactive problem-solving**. The interaction involves an exchange of information between the system and user, discussing *why* a particular risk is valid. There are several strategies for this, but the over-riding principle (for both parties) is the **principle of consistency**: the same decisions should be made in the 'same' circumstances. There are several sources where consistency information be sought, including similar previous cases, statistical evidence and other experts' opinions.

4 Tools And Techniques

4.1 Tool Requirements

There are three major requirements of an implementation which must be met to allow the above functionality, which we now list.

1. **Ability to compare**. In order to construct justifications for risk, it must be possible to make easy comparisons between details of different user's opinions.
2. **Ability to modify**. As experts opinions may change, they must be able to easily modify the system's model of their opinions without the intervention of a programmer every time.
3. **Ability to argue**. In order to justify conclusions, there must be some internal representation of *why* something is believed.

We now describe the representational methods used in Optimist and how they fulful these requirements.

4.2 Representing Different Opinions

In order to represent different opinions it is necessary to keep the different opinions separate in some way. A simple (if time-consuming) method is to engineer a separate knowledge-base for each user, rules and 'facts' tagged with the believer's name. This approach has been adopted in the several systems, including Negoplan [Matwin et al., 1989] and LeClair's multi-expert knowledge system [Alexander and Evans, 1988].

For our particular requirements, however, there are two disadvantages with this approach. Firstly, given the known bottleneck of formalising expert knowledge, the time involved to separately engineer (and maintain) knowledge-bases for each user is likely to be impractically long. Secondly, the problem of making meaningful comparisons across knowledge-bases with completely different structures is formidable.

To overcome these problems we restrict this general approach to a more tractable version which we describe as a '**skeleton + certainty sets**' representation. First, a *single* rulebase for risk appraisal is developed, and then the *body* of the rules are separated from the *certainty measures* attached to them. As a result, a set of certainty values for the rulebase can be loaded, modified and stored under a different name. Thus multiple 'rule' sets (ie. sets of certainty values) can evolve. As there is a single common rulebase to which each set of certainty values attaches, the task of making comparisons and isolating points of disagreement is made possible. We show this schematically in Figure 1.

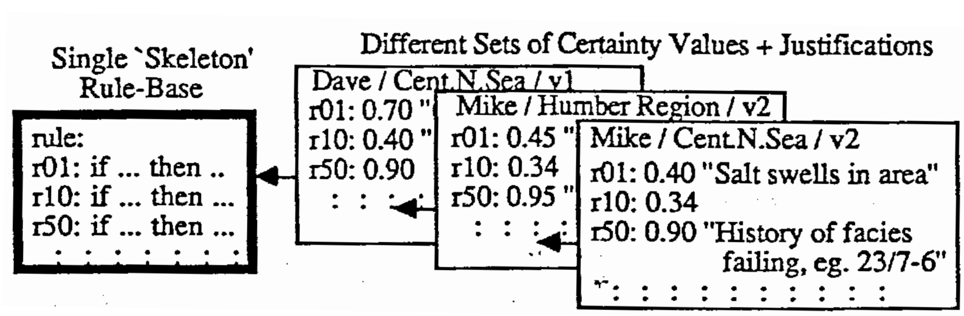
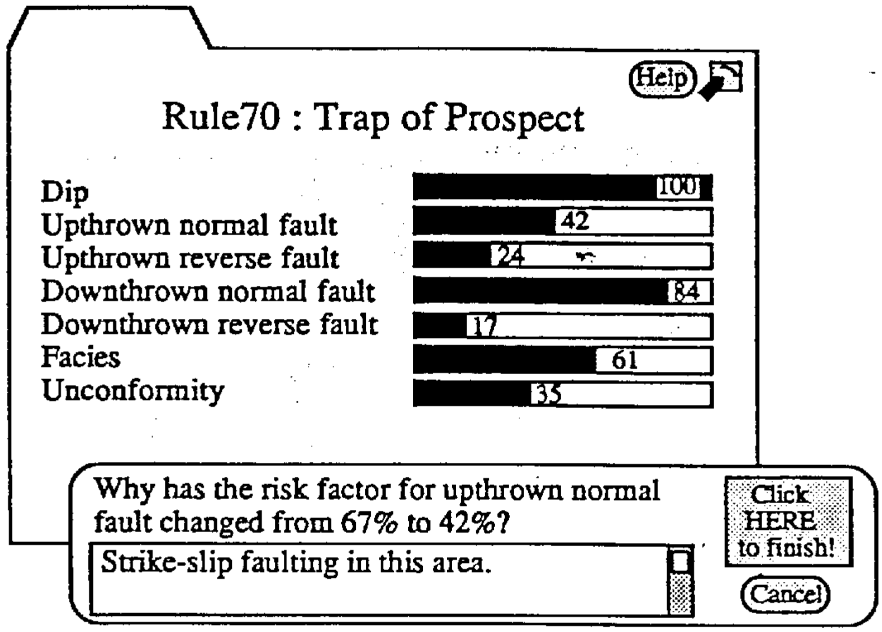


Figure 1: The separation of a single rule-base from sets of certainty values

Figure 2: The use of sliders for displaying and changing certainty values. The sliders are attached to a rule which, given a particular type of trap (ie. dip, facies, etc.) is present in the prospect, concludes a contributing probability of oil being present (ignoring all other factors).



4.3 Human-Computer Interaction Issues

A second requirement of our representation (Section 4.1) is that the users should be easily able to modify the system's model of their opinions. Given our adoption of a 'skeleton + certainty sets' framework, we have a simple method by which the users can alter the system's conclusions: simply by changing the certainty values, along with a textual justification for the changes they make. This is implemented by allowing the user to drag graphical sliders, as shown in Figure 2.

This method allows users to easily maintain the system's record of their risk judgements themselves, thus reducing the need for intervention from a 'knowledge engineer'. Engineering of the knowledge-base thus requires enumerating all the possible sources of risk, while the details and degree of risk are tuned by the users. More structural changes to the knowledge-base still requires intervention from a programmer, but such interventions are infrequent and users are able to perform the day-to-day maintenance of the system's model of their opinions themselves.

4.4 Representation Details

4.4.1 Skeleton + Certainty Sets

While this described method is independent of our particular rule format and certainty calculus, we briefly overview the details here.

The calculation of the probability of finding oil is divided up into six loosely independent risking sub-problems relating to the probability of finding an oil-retaining rock structure and its connection to an oil-producing source. The overall probability is thus the product of the probabilities for each of the six conditions.

For each of the six conditions, there are a number of possible factors which may cause the condition to 'fail' (ie. be absent). Each of these is represented by a different rule with an attached 'probability of success' or **risk factor**. For example a very simple rule is:

```
rule03 ::
       if trap(DirnX) of new_prospect =  downthrown_reverse_fault
       then oil_present cf 65%.
```

The risk factor here is 65% (note that 'certainty factor' and 'risk factor' are used synonymously here). We adopt a simple certainty calculus in which these risk factors are again considered loosely independent. The probability for a condition is thus the product of the applicable risk factors.

It should be noted that these rules are high abstractions of the much more complex reasoning of experts in assessing risk. They should be thus be regarded as typical or **normative** risk values, rather than as absolute. The conclusions of the system thus represent a normative risk judgement.

Other rules are used to derive the conditions of the risk factor rules. These rules have no certainty measures attached (ie. are logical).

4.4.2 Representing Justifications

Risk factors are then stored separately from the rules, parameterised by rule number and attribute value, along with an attached textual justification as shown below.

Certainty Set: Mike, Humber Region, version1

rule	attribute	risk factor	date	justification
rule03	dip	90%	16.10.89	"Dip histories good here"
rule03	d.r.fault	65%	16.10.89	"Evidence of calcification in well 23/2 worrying"
:	:	:	:	:

Other tests besides equality on attribute values can be used to select a risk factor from this set.

In addition to using rules, some risk factors are derived instead by applying special-purpose statistical **procedures** based on available well data. This is described in detail in [Clark, 1988].

Because geological properties vary over regions, users can set up a number of 'rule sets' (ie. certainty sets) each for a different geographical region.

4.4.3 Storing Case Histories

In addition to storing personalised 'rule sets', we also record the *application* of those rules in solving particular problems. These history records are used extensively by the system during appraisals of other prospects. The stored history is akin to a proof tree, the tree depicting the formal structure of the argument while the textual justifications within the tree representing domain-specific knowledge. Smolensky et al. have described languages making this 'divide' between formal structure and 'informal' (ie. textual) content as **semi-formal languages**, and illustrate the power of this technique in their system Euclid [Smolensky et al., 1987].

5 Example

5.1 Initial Risk Calculation

At startup, the user gives their name and can enter/view a prospect using a map and graphical drawing tools, and providing values for the prospects' attributes.

To perform a risk evaluation, the user selects their 'rule set' (ie. certainty set) which they have configured for that region, and then selects one of the six top-level risk components to work through. If statistical procedures are to be used to calculate risk (Section 4.4.2), plotting tools are called up. Otherwise, rules are applied to the prospect's description and a summary of those which fired given. A numerical **risk factor** denotes the *probability* of finding oil given the prospect has a certain feature, ignoring all risks due to other features. Thus a *high* risk factor implies a *low* risk. Risk factors range from 0% (zero probability of oil, ie. infinite risk) to 100% (oil certain, ie. no risk). For example:

Risk for trap effectiveness:

1. Faulting is more recent than the seal	risk factor:	60%	(1 in 1.7)
2. Trap in dirn 1 is an unconformity	risk factor:	35%	(1 in 2.9)
3. Trap in dirn 2 is upthrown norm. fault	risk factor:	87%	(1 in 1.1)
4. Trap in dirn 3 is a dip	risk factor:	100%	(1 in 1.0)
5. Trap in dirn 4 is a dip	risk factor:	100%	(1 in 1.0)

From this I'd conclude a likelihood of 1 in 5.5...

This likelihood ($1/\prod(r_i/100)$) represents the normative value according the rule-base which Mike loaded.

5.2 Argumentation Options

At this point, various options (in addition to normal explanation facilities) are available to discuss this risk value. Below we present loose textual equivalents of the graphically-presented information (using a HyperCard-like interface) which the user can inspect. The user's actions are underlined. Some of these functions are incorporated in the current system while others are still under development. We consider below the user querying/disagreeing with the normative risk of 87% caused by the faulting in the oil trapping structure.

<u>Why faulting risk?</u>
The initial fault risk factor as set at 70% (12-Oct-89).
You raised it to 80% (14-Dec-89) because: good fault seals proven in wells 33/46g and 35/39k nearby.
You raised it again to 87% (3-Feb-90) because: well 55/6y, involving upthrown normal faulting, struck oil.

<u>Other appraisals?</u>
Searching...

Dave's already appraised this prospect (12-Dec-89), he gave a risk factor of 85% for the downthrown normal fault. He re-appraised it on 3-Jan-90 but didn't change his opinion here.
Hey! Steve appraised this prospect on 3-Feb-90 and gave the faulting risk factor as 25% because: seismic sections s34313-7 indicate the faulting here is abnormally also strike-slip, a considerable risk to the prospect.

Try other rulesets?
Searching...
Hey! According to the general region rules, Steve and Dave would have pretty much agreed, but Andrew considers the fault much more risky!
According to my stored rule sets, the normative risk factors should be:
85% (Tim) because: good region for faults (eg 3/3r, 34/3w both sealed)
85% (Dave) because: similar to Humber region risks.
60% (Andrew) because: p65/32 was dry; I reckon faulting was to blame.

Here the user might call up Andrew's appraisal of p65/32 for inspection.

Search cases?
Searching...
You've previously appraised downthrown normal faults in nearby prospects 24/37y, 32/11w and 33/34d. You risked them as 80%, 85% and 50%.
Do you want to see the record for 33/34d?

The risk factor assigned for prospect 33/34d is in greatest conflict with the one assigned here. The user is asked if he or she wishes to view this appraisal record.

Statistics?
Please sketch an area on the map to take statistics over. (User does so)
Searching...
According to my records, in this region there are 74 prospects,
13 with downthrown normal faults, 8 which struck oil.
Of those 8, 2 had the faulting as the major source of risk to the prospect.
Of those 2, 2 struck oil and 0 did not.

At this point the user might choose to inspect the appraisal records of the 13 prospects with downthrown normal faults.

5.3 Arguing Back

The user may, after this consideration, disagree with the risk factor and change it. This change is effected by using graphical sliders (as shown previously in Figure 2).

Disagree.
What do you want to change the risk value to? 65%
Please justify this change from the normative value:
Seismics show strike-slip faulting. Andrew's appraisal of p65/32 and its subsequent failure to strike oil suggests this factor is problematic.

This change and justification is stored in the user's appraisal of this prospect. If the change was for more general, ie. regional, reasons then the user would request that his or her ruleset for the geographical region *also* be updated.

Now the risk appraisal is redone with this changed risk value, and the dialog continues. In this way the user is able to 'argue back' with the system, providing more information which differentiates the current case from other cases, thus removing it from the common context which the system has suggested. This information is stored and hence is available to experts when performing other appraisals.

6 Evaluation

Following an initial prototype in 1988, a full-scale system named **Optimist** has been developed, installed and is now in use. We report from our experience of its use by geologists for real-world risk appraisal problems.

We first note that evaluation of the system is difficult for two reasons. Firstly, the system's role is not stand-alone problem-solving but to assist the user in producing a (machine-readable) argument for his/her risk by encouraging consistency. As the risk factors attached to rules are maintained by the users themselves, it is meaningless to compare them with the users' judgements (they are the same thing). Secondly, assessments of the practical utility of applications depend not only on the underlying theory but also on many aspects ancillary to AI, a point made also by other authors eg. [Gaschnig, 1982].

Following its installation, Optimist has been regularly used so far over a period of about five months and most importantly the users remain highly enthusiastic about its value to the appraisal process. Most fundamental to the system's operation has been finding a pragmatic solution to the 'knowledge maintenance' problem, by which users can maintain the system's models of their risk judgements without regular intervention of a knowledge engineer. The fact the system is now in use, away from its development environment, best testifies to the practical usefulness of the 'skeleton + certainty sets' method for representing and modifying different models of reasoning. During the system's development and following installation the system has been used to varying degrees by 8 different exploration experts, all of whom were able to dispute and correct the system's reasoning to their satisfaction.

One important question concerns the adequacy of experts sharing a single 'skeleton' rule-base and differing only in the attached risk factors. In practice, experts can disable rules which they consider irrelevant by setting the risk factor to 100% (no risk) and thus it is only important rules cover all *potential* sources of risk. It also should be noted that rules represent abstract summaries of more complex reasoning (Section 4.4.1); experts thus may still disagree about details of assessing a feature's risk, reflected in their textual justifications, without requiring two different rules to be encoded. During Optimist's use so far it has only been necessary once to extend the rulebase itself, reflecting the relatively high independence of the system from a trained programmer to maintain models of different opinions.

Logs of the system's usage support the claim that there can be substantial disagreement between experts. Table 1 earlier showed the breakdown of risk values for three of the real-world prospects where more than one geologist had made an appraisal. As can be seen, substantial disagreement can exist. This illustrates the difficulty of attempting to form a single, agreed-on body of knowledge for risk appraisal, as required by a more traditional ES methodology.

Finally, a significant observation is that experts will often 'work back' from a partially pre-conceived risk value. Thus, the expert does not only work through the six risk components in turn but may also return to a particular component to alter the risk value, reflecting the use of the system to *construct* rather than *derive* a structured argument for a conclusion. This observation agrees with the 'interpret-then-justify' model of reasoning, reported in other domains [Compton et al., 1989] and with major implications for knowledge engineering [Compton and Jansen, 1989]. It also reflects a degree of arbitrariness in the numerical measures and partly explains why violations of the independence assumption do not cause serious problems in Optimist. However, the numbers are not meaningless: In particular the *relative* risks are important to geologists, reflecting statements 'A is more risky than B'. The experts can make these qualitative judgements easily, and thus the numeric risk framework can be viewed as an approximately fitting model of these judgements.

7 Conclusion

In this paper we have presented a system for supporting experts in risk assessment, based on modelling different expert opinions and involving the user in argumentation about the validity of different risk judgements. Our main conclusion is that the process of argumentation can be made accessible to both skilled and 'computer-naive' users using the techniques we have described, and be used to provide a valuable decision support tool in domains where discussion and pooling of experience plays an important role. This work is significant as it enables expert system technology to be applied in domains where a single, agreed-on body of knowledge cannot be easily assembled.

From our observations of the system's usage following installation, three directions for future work are indicated. Firstly, it has become clear that the most useful argumentation options are those which deal with specific appraisal records rather than the more general regional risk values, suggesting that risk values always need to be presented along with specific appraisals to make them meaningful. Secondly, from users' comments, there is still potential for further developing knowledge-based methods for analysing the appraisal records, and involving the computer more in the location of inconsistencies. Finally, improvements to the user-interface and the graphical presentation of risk comparisons are possible.

Acknowledgements

This research and application was funded by Enterprise Oil plc, and I particularly thank our exploration experts Mike Whyatt and Dave Rhodes of Enterprise Oil for their involvement and continuing enthusiasm. Thanks also to Tim Niblett for valuable discussions on earlier versions of this paper.

References

Alexander, S. M. and Evans, G. W. (1988). The integration of multiple experts: A review of methodologies. In Turban, E. and Watkins, P. R., editors, *Applied Expert Systems*, pages 47–53. North-Holland, Amsterdam.

Andrews, B. (1989). *Successful Expert Systems*. Financial Times Business Info., London.

Buchanan, B. G. (1985). Expert systems. In *Knowledge-Based Systems and Their Applications: Texas Instruments AI Satellite Symposium*. Texas Instruments. (Seminar Notes).

Buchanan, B. G., Barstow, D., Bechtel, R., Bennet, J., Clancey, W., Kulikowski, C., Mitchell, T., and Waterman, D. A. (1983). Constructing an expert system. In Hayes-Roth, F., Waterman, D. A., and Lenat, D. B., editors, *Building Expert Systems*, pages 127–167. Addison-Wesley, Ma.

Castner, J., Apteé, C., Griesmer, J., Hong, S. J., Karnaugh, M., Mays, E., and Tozawa, Y. (1986). A knowledge-based consultant for financial marketing. *AI Magazine*, 7(5):71–79.

Clark, P. (1988). Exemplar-based reasoning in geological prospect appraisal. TIRM 034, Turing Institute, Glasgow, UK.

Compton, P., Horn, K., Quinlan, J. R., Lazarus, L., and Ho, K. (1989). Maintaining an expert system. In Quinlan, J. R., editor, *Applications of Expert Systems*, volume 2, pages 366–384. Addison-Wesley, Sydney.

Compton, P. and Jansen, R. (1989). A philosophical basis for knowledge acquisition. Tech. Report TR-FD-89-01, CSIRO IT Division, Sydney.

Duda, R., Gaschnig, J., and Hart, P. (1981). Model design in the Prospector consultant system for mineral exploration. In Webber, B. L. and Nilsson, N. J., editors, *Readings in Artificial Intelligence*, pages 334–348. Tioga, Ca.

Gaschnig, J. (1982). Application of the Prospector system to geological exploration problems. In Hayes, J. E., Michie, D., and Pao, Y.-H., editors, *Machine Intelligence 10*, pages 301–323. Horwood, Chichester.

Klein, M. and Lu, S. C.-Y. (1989). Conflict resolution in co-operative design. *AI in Engineering*, 4(4):168–180.

Matwin, S., Szpakowicz, S., Koperczak, Z., Kersten, G. E., and Michalowski, W. (1989). Negoplan: An expert system shell for negotiation support. *IEEE Expert*, 4(4):50–62.

Mittal, S. and Dym, C. L. (1985). Knowledge acquisition from multiple experts. *AI Magazine*, 6(2):32–36.

Premkumar, G. (1989). A cognitive study of the decision-making process in a business context: Implications for design of expert systems. *International Journal of Man-Machine Studies*, 31:557–572.

Prerau, D. S. (1989). Choosing an expert system domain. In Guida, G. and Tasso, C., editors, *Topics in Expert System Design*, pages 27–43. North-Holland, Amsterdam.

Reboh, R. (1983). Extracting useful advice from conflicting expertise. In *IJCAI-83*, volume 1, pages 145–150.

Rissland, E. L., Valcarce, E. M., and Ashley, K. D. (1984). Explaining and arguing with examples. In *AAAI-84*, pages 288–294.

Smolensky, P., Bell, B., Fox, B., King, R., and Lewis, C. (1987). Constraint-based hypertext for argumentation. In Smith, J. B. and Halasz, F., editors, *Hypertext '87 Proceedings*, pages 215–245, NY. Assoc. Computing Machinery.