A Model of Argumentation and Its Application in a Cooperative Expert System

Peter Clark

Ph.D.

Turing Institute
Department of Computer Science
University of Strathclyde
Glasgow

1991
Abstract

This thesis is about a particular type of problem-solving, based on the discussion which occurs between experts when addressing difficult problems. We term this activity argumentation. Argumentation is characterised by its interactive or social nature, and is based on the construction, criticism, justification and modification of arguments. In this thesis we present a model of argumentation, and evaluate an implementation’s utility for decision support where the system is placed in the role of a colleague to its users.

Keywords: argumentation, cooperative systems, expert systems, Toulmin, case-based reasoning, knowledge acquisition, problem-solving.
Acknowledgements

I am greatly indebted to Tim Niblett for the many useful discussions concerning this thesis work, from which many of the ideas presented here have arisen. I am also particularly grateful to our oil exploration experts Mike Whyatt, Nigel Capon and Dave Rhodes of Enterprise Oil plc for their enthusiasm, ideas about and involvement in the development of Optimist. Thanks also to Claude Sammut, who contributed some of the early ideas for tackling the geological problem. I also thank Mike Whyatt and David Aha for their comments on an earlier draft of this thesis, and Robin Boswell for his generous assistance with presentation details.

To maintain confidentiality of Enterprise Oil plc’s geological data in the evaluation, all geologically significant names have been changed. This does not affect the description or evaluation of the underlying AI techniques.
# Contents

1 **Introduction** ................................................................. 1  
   1.1 Introduction and Goals .............................................. 1  
   1.2 Thesis Overview ...................................................... 1  
   1.3 Models of Problem-Solving ......................................... 2  
   1.4 The Autonomous Expert System Model ............................ 3  
      1.4.1 Introduction .................................................. 3  
      1.4.2 The Development of Expert Systems ......................... 3  
      1.4.3 The Scope of Application .................................... 4  
   1.5 Cooperative Models .................................................. 4  
      1.5.1 Distributed Models of Cooperation ......................... 5  
      1.5.2 Eristic Models of Cooperation .............................. 5  
   1.6 Eristic Problem-Solving ............................................. 6  
      1.6.1 Motivation and Functionality ............................... 6  
      1.6.2 Design for a System ........................................ 7  
      1.6.3 Related AI Techniques ...................................... 8  
   1.7 Application Domains ................................................ 10  
      1.7.1 Risk Assessment in Geology .................................. 10  
      1.7.2 Business Decision-Making .................................... 12  
      1.7.3 Financial Marketing ......................................... 12  
      1.7.4 Legal Reasoning ............................................. 13  
      1.7.5 Architectural Design ....................................... 13  
      1.7.6 Archaeology .................................................. 13  
   1.8 Summary ............................................................... 13  

2 **General Review** .......................................................... 14  
   2.1 Introduction ........................................................... 14  
   2.2 Expert Systems in the Oil Industry ............................... 14  
      2.2.1 Introduction .................................................. 14  
      2.2.2 Overview of Expert Systems in the Oil Industry ........... 15  
      2.2.3 Review ........................................................ 17  
      2.2.4 Prospector ..................................................... 17  
      2.2.5 Dimpeter Advisor ............................................ 21  
      2.2.6 Explorationist ............................................... 22  
      2.2.7 SPII-2 ....................................................... 23  
      2.2.8 Geologix and Foenix ....................................... 23  
      2.2.9 Conclusion ................................................... 24  
   2.3 Decision Support Systems .......................................... 24  
      2.3.1 Introduction .................................................. 24  
      2.3.2 Examples ..................................................... 24  
      2.3.3 Discussion ................................................... 25  
      2.3.4 Group Decision Support Systems ............................. 26
2.4 Hypertext Approaches to Argumentation ........................................... 27
   2.4.1 Euclid ................................................................. 27
   2.4.2 gIBIS and rIBIS ..................................................... 28
   2.4.3 NoteCards ............................................................ 29
   2.4.4 Summary ............................................................... 29
2.5 Evaluation Issues ................................................................. 30
   2.5.1 Introduction ........................................................... 30
   2.5.2 System Accuracy ...................................................... 30
   2.5.3 User Effectiveness .................................................... 31
   2.5.4 User Satisfaction ...................................................... 32
   2.5.5 System Usage ........................................................ 32
   2.5.6 Information Analysis ................................................. 33
   2.5.7 Conclusions ........................................................... 34
2.6 Summary ............................................................................. 34

3 A Model of Argumentation ......................................................... 35
   3.1 Introduction .................................................................... 35
      3.1.1 Goals ................................................................. 35
      3.1.2 Overview of the Chapter ......................................... 35
   3.2 Basic Definitions ........................................................... 35
   3.3 Models of Argument ........................................................ 36
      3.3.1 Requirements ....................................................... 36
      3.3.2 Expert System Approaches ..................................... 36
      3.3.3 Formal Logic ........................................................ 37
      3.3.4 Social Argumentation Models .................................. 37
   3.4 A Model of Argument Structure .............................................. 37
      3.4.1 Motivation ............................................................ 37
      3.4.2 Toulmin’s Model ..................................................... 38
      3.4.3 Notation ............................................................... 40
      3.4.4 The Role of Models of Opinions ................................ 40
      3.4.5 Constraining Models of Opinions .............................. 40
      3.4.6 Language Issues .................................................... 41
   3.5 Formalising Argument Content .............................................. 41
      3.5.1 Introduction .......................................................... 41
      3.5.2 Grounds, Warrants and Claims ................................. 42
      3.5.3 Backings .............................................................. 42
   3.6 Consistency .................................................................... 43
   3.7 Summary of the Argument Model ............................................. 43
   3.8 An Example of our Argument Model ......................................... 44
      3.8.1 A Common Set of Skeleton Warrants ............................ 44
      3.8.2 A Model of Opinion ................................................ 44
      3.8.3 Problem Description ............................................... 44
      3.8.4 Arguments ............................................................ 44
   3.9 Practical Argumentation using the Model .................................... 46
      3.9.1 Modelling and Comparing Different Opinions ............... 46
      3.9.2 Open-ended Domain Backings ................................... 46
      3.9.3 Justifications and Case-Based Reasoning ..................... 47
      3.9.4 Conflict Resolution ............................................... 47
      3.9.5 Knowledge Acquisition and Maintenance ..................... 47
   3.10 Related Work .................................................................. 48
      3.10.1 Modelling and Comparing Different Opinions ............... 48

vi
3.10.2 Using Uninterpreted Text for Backings .................................................. 48
3.10.3 Justifications and Case-Based Reasoning ............................................... 48
3.10.4 Conflict Resolution ............................................................................... 49
3.10.5 Knowledge Maintenance ......................................................................... 50
3.10.6 The Theory of Determinations .................................................................. 50
3.11 Summary ...................................................................................................... 51

4 The Model Applied to Oil Prospect Assessment .............................................. 52
4.1 Introduction .................................................................................................... 52
4.2 The Problem Domain .................................................................................... 52
  4.2.1 The General Context of Oil Exploration .................................................. 52
  4.2.2 Finding Oil in a Prospect .......................................................................... 53
4.3 Applications of the Model ............................................................................ 55
4.4 Overview of the Implementation ................................................................... 55
  4.4.1 Description ............................................................................................... 55
  4.4.2 Chapter Overview ..................................................................................... 55
4.5 Representing Data in Optimist ...................................................................... 57
4.6 Representing Qualifiers and Strengths .......................................................... 57
  4.6.1 Requirements ......................................................................................... 57
  4.6.2 Optimist's Combination Rules ................................................................. 58
  4.6.3 Discussion ............................................................................................... 58
  4.6.4 Generalising the Calculus ....................................................................... 59
  4.6.5 Logical Warrants .................................................................................... 59
4.7 Representing Arguments .................................................................................. 59
  4.7.1 Description ............................................................................................... 59
  4.7.2 Important Features .................................................................................. 60
  4.7.3 Constraints .............................................................................................. 62
  4.7.4 Special Warrants ..................................................................................... 62
  4.7.5 Storing Arguments .................................................................................. 63
4.8 Representing Models ...................................................................................... 63
  4.8.1 Representing Warrants ............................................................................ 63
  4.8.2 Creation and Organisation of Users’ Models ............................................. 66
  4.8.3 Constructing Arguments from Models ..................................................... 67
4.9 The Dynamics of Argumentation ................................................................... 67
  4.9.1 Goals of the Interaction .......................................................................... 67
  4.9.2 Starting the System ................................................................................ 67
  4.9.3 Argumentation ........................................................................................ 70
  4.9.4 Responding to Correction ...................................................................... 72
  4.9.5 Storage of the Argument ....................................................................... 72
4.10 Implementation Information ......................................................................... 72
  4.10.1 System and Default Model Construction ............................................... 72
  4.10.2 Issues of Interface Design ...................................................................... 73
  4.10.3 Implementation Statistics ...................................................................... 74
4.11 Summary ........................................................................................................ 74
4.12 Example of The System In Operation ........................................................... 77
  4.12.1 Startup .................................................................................................... 77
  4.12.2 Problem Description .............................................................................. 77
  4.12.3 Set Up Appraisal .................................................................................... 78
  4.12.4 Selecting a Model .................................................................................. 78
  4.12.5 Starting the Appraisal .......................................................................... 78
  4.12.6 Dispute .................................................................................................. 81
5 Evaluation

5.1 Introduction .................................................................................. 88
5.2 The Evaluation Context ................................................................. 88
  5.2.1 Assessing Performance Improvement ........................................ 88
  5.2.2 Influences in Risk Appraisal .................................................... 89
  5.2.3 Pragmatic Constraints ............................................................ 89
5.3 Evaluation Criteria ........................................................................ 89
5.4 System Usage .............................................................................. 91
  5.4.1 Context and Overview ............................................................ 91
  5.4.2 Data Collection ..................................................................... 92
  5.4.3 Discussion ............................................................................ 93
5.5 Stability and Maintenance of Models .............................................. 95
  5.5.1 Introduction ........................................................................ 95
  5.5.2 Metrics ............................................................................... 95
  5.5.3 Results ............................................................................. 95
  5.5.4 Conclusions and Discussion .................................................. 98
5.6 Measures of Variability ................................................................. 99
  5.6.1 Overview ........................................................................... 99
  5.6.2 Variability of Manual and Optimist Appraisals ......................... 100
  5.6.3 Comparison of Appraisals with and without Optimist .......... 103
  5.6.4 Summary ........................................................................ 105
5.7 Acquisition of Extra Knowledge .................................................. 105
  5.7.1 Introduction .................................................................... 105
  5.7.2 Categorisation of Backings .................................................... 106
  5.7.3 Justification for Strength Differences ..................................... 106
  5.7.4 Summary ........................................................................ 106
5.8 Differences of Opinion ................................................................. 108
5.9 Summary and Conclusion ............................................................... 108

6 Generality and Extensions .................................................................. 111
6.1 Introduction ................................................................................... 111
  6.1.1 Overview of the Chapter ....................................................... 111
  6.1.2 A General Argumentation ‘Shell’ ........................................... 111
  6.1.3 Basic Functions Supported by the Model ............................... 111
6.2 Alternative Applications of the Model ........................................... 113
  6.2.1 Introduction .................................................................... 113
  6.2.2 General Requirements ....................................................... 113
  6.2.3 Greater Variations in Argument Structure ......................... 114
  6.2.4 Chaining of Warrants ........................................................ 116
  6.2.5 Qualified Grounds ............................................................... 117
  6.2.6 Alternative Models of Strength Combination ....................... 117
  6.2.7 Summary ........................................................................ 118
6.3 Additional Requirements for Application .................................... 118
  6.3.1 The Task is Knowledge-Based .............................................. 118
  6.3.2 The Task is of Establishing a Claim ..................................... 118
  6.3.3 Disagreements are Localised ................................................ 119
6.4 The Skeleton Warrants: Strengths, Constraints and Extensibility 119
6.4.1 Introduction ............................... 119
6.4.2 The Flexibility of the Model for Expressing General Changes .... 120
6.4.3 Changing the Set of Skeleton Warrants Themselves .......... 120
6.4.4 Summary .................................. 122
6.5 Application in Another Domain: The ML Advisor ................ 122
6.5.1 Introduction ................................ 122
6.5.2 The Problem Task and an Argumentation Approach .......... 122
6.5.3 A Sketch of an Implementation .......................... 123
6.5.4 Hypothetical Interaction ............................ 124
6.5.5 Summary .................................. 128
6.6 Application in Expert System Domains .......................... 128
6.7 Future Work ................................... 129
6.7.1 Representing Semantics of Text Backings .................... 130
6.7.2 Changing the Warrant Set ................................ 130
6.7.3 Development of Models of Opinion .......................... 131
6.7.4 Unconstrained Use of Strengths and Qualifiers ............... 132
6.7.5 More Extensive Postmortem Analysis ........................ 132
6.7.6 Personalised Problem Descriptions .......................... 132
6.7.7 Recording the Dynamics of the Argument .................... 133
6.8 Summary ..................................... 133

7 Conclusion ..................................... 134
7.1 Summary of the Thesis ................................ 134
7.2 Contributions of the Work ................................ 134
  7.2.1 Expert System Technology ............................... 134
  7.2.2 Case-Based Reasoning .................................. 135
  7.2.3 Knowledge Acquisition .................................. 135
7.3 Closing Remarks .................................... 136

Glossary Index ................................. 146

Index ............................................. 147

A Data Structures for Storing Arguments ......................... 148

Appendices

B A Simple Profit-and-Loss Model ............................... 150

C Summary of Appraisals .................................... 152

D Backings Provided by Users .................................. 156
  D.1 Backings Used in Appraisal Arguments ...................... 156
  D.2 Backings In Users’ Models ................................ 160

E A Domain-Independent Shell .................................. 162
  E.1 Pseudo-Code ........................................ 162
  E.2 Prolog Code ......................................... 165
    E.2.1 The Argumentation Shell ............................ 165
    E.2.2 A Knowledge Base for Prospect Appraisal .............. 171
    E.2.3 A Knowledge Base for a Software Advisor .............. 173
E.3 Trace of Usage ...................................................... 175
   E.3.1 Prospect Appraisal ........................................... 175
   E.3.2 Advice about Using ID3 .................................... 179

F Details of the ML Advisor Application 182
   F.1 Problem Description Language ................................. 182
   F.2 The Skeleton Warrant Set .................................... 183
   F.3 A Model of Opinion ........................................... 185
   F.4 Advice Cases ................................................... 186
# List of Tables

2.1 Final reported status of oil exploration systems from the general literature. . . . 17  
2.2 A characterisation of application tasks in the overviewed systems. . . . . . . . 18  
3.1 A set of skeleton warrants. ................................. 45  
4.1 Summary of Terminology. ................................ 56  
4.2 Syntax of skeleton warrants. .............................. 64  
4.3 Some example skeleton warrants. ....................... 65  
4.4 Strengths and backings in a model of opinion. ....... 65  
4.5 Revised numbering of warrants to group subsets together. .................. 66  
4.6 The Argument Construction Algorithm. ................ 68  
4.7 The algorithm for generating special warrants in Optimist. ................ 70  
4.8 Number of lines of code in different modules in Optimist. ............... 75  
5.1 Modification of Warrant Strengths for Different Users’ Models. ............ 97  
5.2 The accuracy of the default and geologist B’s tuned model after 16 appraisals. . 98  
5.3 Variability of Manual and Optimist Appraisals. .......................... 102  
5.4 Comparison of appraisals of 8 prospects with and without Optimist. .... 104  
5.5 Distribution of backings among categories. .......................... 106  
5.6 A sample of the backings used in appraisals in Optimist. .................. 107  
5.7 Analysis of differences between appraisals by different users. ............ 109  
C.1 The strengths of warrants (1-17) occurring in appraisal arguments in Optimist. 153  
C.2 The strengths of warrants (18 onwards) occurring in appraisal arguments in Optimist. 154  
C.3 Additional information about the appraisals. ................................ 155
List of Figures

2.1 Interpolation with uncertain evidence in Prospector. .................. 19
3.1 An example of a medical argument, showing Toulmin’s structure. ........ 38
3.2 Models constructed from a single set of skeleton warrants. ............. 41
3.3 Translation of language for argumentation. ............................ 41
3.4 A model of opinion. ................................................. 44
3.5 An example argument for the claim ‘trap is ineffective’. ............... 46
4.1 An example of the formation of an oil field. ............................ 54
4.2 Architecture of Optimist. ........................................... 57
4.3 The structure of an argument in Optimist. .............................. 61
4.4 Separating the subarguments. ........................................ 61
4.5 Normal and constrained use of strengths and qualifiers in subarguments. 62
4.6 Flow of Control in Optimist. ....................................... 69
4.7 Starting up the system. ................................................ 77
4.8 Data entry in Optimist. ................................................ 78
4.9 Summary of stored appraisal arguments for prospect 3b/a-2b. ............ 79
4.10 Selection of a model of opinion. ..................................... 79
4.11 The six risk components, prior to constructing an appraisal argument. 80
4.12 Summary of warrants in the argument for the (sub-)claim ‘trap is present’. 80
4.13 Attachment of the subargument into the overall argument for oil presence. 81
4.14 Summary of warrants in the argument for (sub-)claim ‘trap is effective’. 82
4.15 The strengths of seven warrants in the selected model of opinion. ......... 82
4.16 Modification of a warrant’s strength and backing by the user. .......... 83
4.17 Selection of how far to propagate changes to the warrant. ............... 83
4.18 The modified argument for the claim ‘trap is effective’. ............... 84
4.19 An abbreviated list of precedents supporting the current argument. .... 84
4.20 An abbreviated list of precedents conflicting with the current argument. 85
4.21 Defining the similarity metric for searching for precedents. ........... 85
4.22 Comparison of qualifications (‘TotRisk’) to the main claim ‘oil is present’ in selected arguments. ........................................ 85
4.23 Generation of a special warrant for the claim that adequate gross is present. 86
4.24 Summary of the complete appraisal argument. .......................... 87
5.1 Cumulative usage of Optimist. ....................................... 93
5.2 Cumulative usage of Optimist (Geologist B). ............................ 94
5.3 Cumulative usage of Optimist (Geologist C). ............................ 94
5.4 Reduction of error in a user’s model of opinion. .......................... 96
6.1 The top level procedures of a simple argumentation shell. ............... 112
6.2 The Base Argument for ID/CN’s use in Circuit Fault Diagnosis. ......... 125
6.3 Producing Diagnostic Advice using an Expert System. ................... 129
A.1 An example argument ..................................................... 148

B.1 How the difference in estimated and actual oil probability affects profit. ........... 151
Chapter 1

Introduction

1.1 Introduction and Goals

This thesis is about a particular type of problem-solving, based on the discussion which occurs between cooperating experts when addressing difficult problems. This form of problem-solving is characterised by its interactive or social nature and by the existence of differences of opinions among experts, and is based around the construction, criticism, justification, and modification of arguments. We describe this activity as argumentation (argumentation: “debate, discussion, disputation”, from the Oxford English dictionary [Syk76]), and the type of problem-solving as eristic problem-solving (eristic: “the art of disputation” [Syk76]).

This work was originally motivated by the difficulties we encountered in applying normal expert system techniques to the task of geological appraisal of hydrocarbon prospects (i.e. potential sites of oil and/or gas). These included the following:

- Petroleum geology is a complex, potentially unbounded domain in which all the necessary knowledge cannot be represented in the computer.
- Geologists often disagree, making it difficult to identify the ‘correct’ knowledge for problem-solving.
- Geological expertise changes with time, posing problems for maintaining a system’s knowledge base.

In addition, the desired role of the system differed from the normal expert system role of explainable problem-solving: geologists desired not so much a system to provide solutions but one which would help them construct and improve their own assessments of risk.

Partly as a result of these characteristics, geologists frequently meet to discuss particular geological problems they are working on. Argumentation among geological experts is an integral part of risk assessment, serving several purposes which contribute to better decision-making including: alerting experts to extra relevant information, exploring alternative opinions, checking consistency, identifying focal points for further analysis, and updating each expert’s knowledge.

The goals of this work, then, were to develop a practical computational model of this process and to evaluate an implementation’s utility when placed in the role of a colleague to the experts. In this way, the computer acts as a partner rather than an oracle, and can usefully contribute to discussions by exploiting its special capacities for memory and computation.

1.2 Thesis Overview

This thesis is set out as follows. First, we establish the place of eristic problem-solving within the broader context of AI research. We identify it as a particular type of cooperation between
the computer and user, and highlight its purpose, benefits, and the functions which are required to support it. We overview AI techniques related to our goal, and identify a number of domains whose characteristics suggest an eristic approach could be usefully applied. In Chapter 2 we give a general review of three additional areas related to argumentation and our application domain, covering expert systems in the oil industry, decision-support systems, and hypertext approaches to argumentation. We also review issues in evaluation of these systems to provide the context for evaluating this thesis work.

In Chapter 3, we present a model of argumentation, developed as part of this work, based on earlier ideas by Toulmin. The two objectives of the model are that:

1. It supports the argumentation functions we have identified, so that an implementation can usefully participate in eristic problem-solving alongside experts.

2. It is practical, i.e. can be implemented so that non-computer-skilled users can use it away from the laboratory environment.

We argue for its ability to meet these objectives, and in Chapter 4 we present one way in which this model can be applied. The application is called Optimist, and assists with the task of geological risk assessment. In Chapter 5 we empirically evaluate the implementation using data from its use by geologists over the past twenty one months. Following this, we discuss the generality of our argumentation model and its potential for application in other ways and in other domains.

A glossary index is provided on page 146 for the reader’s benefit. A brief index is also given on page 147.

1.3 Models of Problem-Solving

To establish the place of eristic problem-solving within AI, we present two contexts in which it can be viewed. First, as an alternative model of problem-solving. Second, as a development of expert system technology, building on the now better understanding of its scope and limitations.

Placing argumentation in this first context, AI can be loosely described as the study of computational models of problem-solving. A number of approaches to problem-solving can be identified:

The Autonomous Expert Approach: Many AI systems are based on the metaphor of a single, autonomous expert solving problems by virtue of his or her extensive and possibly informal knowledge about the domain. Problem-solving is seen as the application of a number of heuristics or ‘rules of thumb’ to reason from known facts to a conclusion. We could loosely describe this as a heuristc model of problem-solving. Well-known examples include Mycin (for diagnosis and treatment of infectious diseases) [Sho76], Prospector (for evaluating prospective mineral sites) [DG81], and Internist (for diagnosis in internal medicine) [MPM82].

The Rule-Book Approach: Some problems require the application of an already existing body of rules to reach a conclusion. These rules may be legal statutes, company procedures, physical constraints etc., and have already been determined by some authority. We describe this as a bureaucratic model of problem-solving, and examples include R1 and XCon (for configuring computers to meet system requirements; XCon also included heuristic knowledge) [McD82], the system by Sergot et al. (for encoding the British nationality act) [SSK+86], and the system by Hammond (for reasoning with the DHSS regulations) [Ham86].
The Domain Simulation Approach: Some AI systems solve problems by simulating the application domain's behaviour rather than the reasoning of an expert. This approach models problem-solving as a formal analysis of the domain. Examples include many planning systems (eg. Strips [FHN81], Nonlin [Tat77]), and qualitative modelling techniques (eg. [BML89, Pea88, SM89]). Many problem-solving techniques in engineering adopt this approach (eg. finite element analysis).

Cooperative Approaches: Cooperative models are based on a view of problem-solving as essentially a collaborative process, in which a number of experts plus the computer work together to solve a problem. Cooperative models can be seen as adopting a wider view of problem-solving, in which the complete problem-solving system is not just the computer but also the users and environment in which it is embedded.

We distinguish two complementary styles of cooperation:

Distributed problem-solving: in which different experts work on different sub-problems, the results combining into an overall problem solution.

Eristic problem-solving: in which experts work on the same overall problem, interacting to produce an overall solution. This type of communal or social problem-solving is based on the interaction of different, possibly conflicting chains of reasoning.

This thesis is about the latter style of cooperative problem-solving: each expert works on the same overall problem, cooperation occurring by the experts discussing and exchanging opinions so that the solution they arrive at together overcomes the limits of any individual’s knowledge. An analogy would be a committee meeting to discuss solutions to a particular problem.

1.4 The Autonomous Expert System Model

1.4.1 Introduction

A second context in which eristic problem-solving can be viewed is that of the ongoing development of expert system technology. It can be seen as adopting a broader view of those expert systems intended for use by other experts, in which the overall problem-solving system is the collaborating group of experts, the system acting as a colleague within that group. In this role, the system not only has to solve problems but must also be able to interact, argue with, and respond to other experts who may have alternative opinions.

The value of adopting a broader view of the role of expert systems beyond that of the autonomous problem-solver (the ‘replacement expert’) is now well recognised, and stems from the greater understanding of the scope and suitability of expert systems gained from experience of application builders. We now briefly summarise this experience. Our points are that the autonomous expert system’s role is clear but bounded, and that a number of important issues must be addressed to extend its scope.

1.4.2 The Development of Expert Systems

Expert systems are largely based on a model of the autonomous expert solving a problem by a combination of extensive knowledge and deductive inference. During the 1970s and 1980s, substantial work on expert systems was conducted, founded on this model of problem-solving. The early years were characterised by a number of successful and influential laboratory achievements, demonstrating for the first time the feasibility of capturing aspects of expert reasoning within a computer. Examples include Mycin [Sho76] and Prospector [DGHH81].

Following these achievements, the 1980s saw a rapid growth in commissioned expert systems. However, while a number of remarkable successes were achieved, surveys also reveal high failure
rates from this period. Johnson’s 1984 survey of over 180 commercially-targeted systems reveals that less than 20 achieved commercial success [Joh84], and a later survey in 1987 showed the failure rate remained high [DH88]. Earlier claims of the general applicability of heuristic expert systems came under strong criticism (eg. [Whi88]).

1.4.3 The Scope of Application

Containment of Knowledge

Further experience led to a greater understanding of the scope and appropriateness of systems based on the ‘autonomous expert’ model.

Most significantly, this model demands that all the knowledge required to solve the problem can be adequately represented. We describe this constraint as requiring containment of knowledge, that we can identify, bound, and represent all that we need to know to adequately solve the required class of problems in the domain.

Numerous application builders have established guidelines for identifying domains in which containment is likely to be possible. Buchanan [Buc85], for example, cites the criteria that:

- the domain should be bounded (the relevant concepts and relationships are enumerable in a practical way),
- there should be general agreement about the required knowledge, and
- the problem should be relatively easy to solve.

Other application-builders have established remarkably similar criteria. Following his survey of commercially targeted systems, Johnson cites that successful applications were generally those for simple but non-trivial tasks (taking about ten minutes for an expert to perform), where the expertise was well understood and where the application system was small (taking less than 6 man-months to build) [Joh84]. A more recent report for the Alvey directorate in 1988 similarly concludes that applications should be small, simple and well understood, and experts should be in agreement about the solutions to the problem. A survey of commercial ES successes in 24 companies [And89] also reaches similar conclusions. Principles such as these have become established guidelines for many in deciding where ES technology is most likely to be successfully applied (eg. [CCT85, Lie89, Pre89b]).

The Role of The System

A related characteristic of the ‘autonomous expert’ model is its oracle-like behaviour, i.e., its ability to describe chains of reasoning but not to input and assimilate new knowledge. The requirement for the containment of knowledge is partly a consequence of this characteristic. Systems based on this model are thus best suited to tasks where knowledge can be adequately captured and where its explanations are not expected to be challenged by users.

1.5 Cooperative Models

In practice, almost all ‘autonomous’ expert systems can also be viewed as cooperating with people within the broader context of their application environments. This fact, and the better understanding of the scope of expert systems, has resulted in further direct study of issues of cooperation. Cooperation is beneficial as it allows the computer to operate in ‘ill-structured’ domains, domains where the expertise required for problem-solving cannot be completely represented, as the user can validate the system’s performance and supply missing knowledge.

We earlier distinguished two complementary styles of cooperation (Section 1.3), namely distributed problem-solving and heuristic problem-solving. As analogies, consider the cooperation
1.5. COOPERATIVE MODELS

between builders constructing a house (some working on the roof, others the walls, others the wiring etc.) and the cooperation between a committee of experts.

This distinction is important because the issues and benefits of each style differ considerably. We now summarise the major issues for and benefits of each, and their relation to other relevant AI research areas.

1.5.1 Distributed Models of Cooperation

Benefits and Issues

While ‘distributed problem-solving’ is often used to refer to cooperation within the computer (e.g. parallel processing), we use the word more generally to refer to a type of cooperation between the computer and its user(s), the problem-solving burden being distributed between them.

Distributed cooperation offers one solution to the problems of applying computers in ill-structured domains, based on specialising the role of the computer to constrained, computationally feasible sub-problems which exploit the computer’s relative strengths (calculation, memory, search). This is beneficial as the user’s problem-solving skills are augmented by computer support.

Important issues for distributed problem-solving (both with the user and within the computer) include:

- Burden-sharing (how to divide work on the problem),
- Identifying new support functions which the computer can perform (for human-computer cooperation),
- Control of distributed work on a problem, and
- interaction (computer-computer and human-computer).

Areas of Research

One form of distributed problem-solving is when the computer acts as a tool for the user, performing specialised sub-tasks which may be too time-consuming or difficult for the user to solve alone. Work on decision-support systems (DSSs) falls into this category of cooperation, characterised by the use of analytical decision models to provide important information to users [KM90]. Many expert systems also act in this role, performing specialised, knowledge-based tasks for the users such as offering advice (e.g. Rex [And89]), search (e.g. Dendral [BF81]) and report compilation (e.g. Foenix [And89]). These systems often support extra functions in addition to explainable problem-solving, e.g. sensitivity analysis, or answering ‘what if’ questions. Recently the combination of AI and DSS techniques has been studied, providing DSSs with additional capabilities for knowledge-based reasoning. These systems are sometimes referred to as knowledge-based decision support systems (KB-DSSs) [KM90].

An alternative form of distributed problem-solving is one where the user assists the computer, the user being asked to solve parts of the problem beyond the scope or ability of the machine. Examples include systems by Mackworth for remote sensing [Mac89], and by Niwa for risk management [Niw89, Niw86].

1.5.2 Eristic Models of Cooperation

Benefits and Issues

Distributed cooperation avoids some of the problems of ill-structured domains by specialising the computer’s role to solving feasible sub-problems. In contrast, eristic cooperation addresses
these problems directly by involving the computer in the process of argumentation. This offers an alternative way of applying computers in these domains, the computer's assistance focussed on the interaction of different, possibly conflicting lines of reasoning. This is beneficial as the computer again has particular strengths to offer in this process (memory and search).

Important issues for eristic problem-solving include:

- Identifying and handling conflict,
- Representing multiple lines of reasoning,
- Justification and exchange of information, and
- Adaptation.

We examine these issues in more detail in Section 1.6.

**Areas of Research**

We briefly summarise relevant areas of research here. In Section 1.6 we describe the strengths and limitations of these approaches.

Many AI systems which are targeted for use by experts can be viewed as eristic: the expert is not expected simply to adopt the system's conclusion, but at least to check that the system's conclusion is correct. Thus two lines of reasoning — the computer's and the expert's — are interacting here during problem-solving.

In some domains, supervised problem-solving can offer time-saving benefits by the computer quickly solving problems under supervision from an expert. For example, the systems Garvan-ES1 [CHQ89] and by Srinivasan et al. [SCM91] both assisted medical experts by automatically generating diagnostic reports following their automatic (supervised) diagnoses of patients.

Alternatively, the computer can be used to provide the user with information to help in his or her problem-solving. While one simple method is simply to present a 'second opinion' to the user, there are also alternatives which enable information to be provided in a more focussed way. One approach is for the computer to critique the expert's solution rather than generate its own (sometimes referred to as a critiquing approach), eg. Oncocin [LS83]. Another approach is to use ‘multi-expert’ knowledge-bases where different expert opinions are stored separately, and the user can explore those opinions of most interest to him or her eg. by LeClair [LeC85], and as speculated on by Reboh [Reb83].

Eristic cooperation involves not only problem-solving, but also justification and adaptation in the light of new information. These two processes are central to argumentation, and involve a focussed, two-way exchange of information. This allows the user to benefit from the computer's resource of knowledge, and the computer to improve that resource for the future. For justification, there are many ways of supporting decisions besides the normal expert system explanation techniques. Alternatives include using statistical evidence and the use of precedents (the area of case-based reasoning). For adaptation, work on learning apprentice systems has examined the task of the computer learning directly from the user during problem-solving.

We now examine these areas in more detail.

### 1.6 Eristic Problem-Solving

#### 1.6.1 Motivation and Functionality

Argumentation involves a number of processes compounded together, including reasoning, explanation, justification, interaction and adaptation. All of these are useful in a system, and a system which could support them all would clearly be desirable. Indeed, a general solution to
these issues would solve a number of the major problems in AI. The issue here is thus not to argue that such processes are generally useful, but to identify the functions they must support for the particular task of assisting experts in complex domains (alternative tasks include knowledge acquisition, psychological modelling, machine learning etc.). The identification of these functions acts as a guide to how the processes can be simplified to a manageable level, a goal which we wish to achieve with our model of argumentation.

We identify the following benefits which we would like a system's functions to support:

- To identify the strengths and weaknesses in an argument.
- To alert the arguer to relevant information he or she may be unaware of.
- Using that information, to improve the consistency of decisions.
- To help experts to articulate arguments.
- To offer alternative or overlooked interpretations of data.
- To explore other opinions.
- To rapidly identify areas of disagreement, and locate the causes and reasons for that disagreement.
- To communicate knowledge, thus updating the knowledge of the receiver.

There is considerable motivation for involving the computer in this process by virtue of its capacity for memory and calculation:

1. Facts and records of reasoning in the computer are persistent, ie. do not fade or become forgotten with time.

2. The computer's capacity for routine search allows arguments to be thoroughly checked and explored.

In these ways, the computer can valuably assist experts by extending their resources of knowledge and reasoning.

1.6.2 Design for a System

There are several ways an argumentation system can be constructed. Here we describe and justify the general approach taken in this research, and identify the functions which a system must support to provide the benefits listed in the previous section. This provides a framework for surveying related work and a set of requirements which a suitable argumentation model should meet.

The system's knowledge used to construct an initial argument can be seen as a particular model of an opinion about the domain, which may then be argued about with the user. One possible design for a system would be to encode a particular model (eg. based on a particular expert) which the system would then use and keep up-to-date through argumentation. However, in this thesis we adopt a different approach in which the system maintains several models representing the viewpoints of different users. During argumentation, a single model of opinion is selected (usually that of the current user) for the system to adopt and argue from.

This design decision – that the system works with a model of the user’s opinion rather than develop its own opinion – is novel and important. In particular, if the user disagrees with the system's conclusion (using the model of his or her opinion) it implies either

1. the user is being inconsistent, or
2. the system’s model is incorrect and needs updating, or

3. while the model is generally correct, there is some idiosyncrasy of the current problem which the user knows about but the system is unaware of (it is not currently represented in the system). This acts as a cue for the user to provide the missing information.

Thus this design decision provides several advantages:

1. Dialogue is focussed on the user’s opinion, i.e. exactly that which the user requires assistance with.

2. The system quickly acquires a representation of the user’s opinion about a particular problem, on the grounds that correcting the system’s ‘first guess’ is easier for the user than entering his or her entire argument from scratch.

3. Given this representation, the system can subject it to analysis for supporting and conflicting evidence, ‘arguing’ with the user using the results.

4. It provides a natural means of knowledge maintenance: the user’s corrections of the system (even after dialogue) are in fact corrections to its representation of his or her opinion.

5. Different opinions can be adopted by the system for comparison.

Thus a system based on this particular design is required to support the following functions:

- To construct arguments.
- To argue for (and against) a particular line of reasoning by exploiting the availability of data and knowledge within the system.
- To compare and contrast different arguments, identifying where disagreements exist.
- To represent and maintain models of different experts’ opinions about the domain.
- To maintain a comprehensive record of all arguments created by users.

### 1.6.3 Related AI Techniques

We now describe related areas of work which meet or partially meet these requirements.

**Critiquing Systems**

One function of argumentation is to identify the strengths and weaknesses of an argument. Expert systems have limited ability to analyse users’ reasoning as their dialogue is focussed solely on their (the systems’) own reasoning (they are ‘egocentric’). To overcome this, an argumentation system first needs to be aware of the user’s opinion.

One step in this direction is to adopt a ‘critiquing approach’. Instead of offering a decision or piece of advice, critiquing systems accept the user’s decision or plan and subject it to critical evaluation. One example of a critiquing system is Oncocin [LS83] for cancer therapy. Originally the system produced advice in the normal expert system fashion, but it became a source of annoyance to the users to have to override the system every time it produced wrong or inadequate recommendations. To overcome this, the system was modified to criticise the therapy plan of the user, locating differences between the user’s and system’s plan. While this was a small change to the system’s design, it improved its acceptability. Another example is Attending, for anaesthesiology management [Mil84].

Several points about critiquing systems should be made. First, these systems overcome the egocentricity criticism of a normal expert system, as the computer is focussed on the user’s
solution and not solely its own. They mark a step towards eristic problem-solving as dialogue is now guided by the interaction of two lines of reasoning (the user’s and system’s). Second, however, the system still has only limited knowledge of the user’s opinion as he or she enters only his or her problem solution and not the reasoning underlying it. This limits the analysis the system can perform about the user’s opinion. Finally, it assumes the user can express his or her solution to the computer in a form it can understand. While this may be acceptable for a problem solution, it may prove difficult if the user was also required to express the reasoning behind that solution to the computer directly. We hope that our alternative approach will ease these latter difficulties, in which the system first generates a solution using a representation of the user’s knowledge, and second the user challenges and/or corrects the system.

Multi-Expert Systems

A requirement of our argumentation model is that different opinions are represented separately, and that they can be compared and contrasted with each other.

Several authors have suggested modelling different expert opinions by keeping several separated knowledge-bases. In LeClair’s system [LeC85], different expert opinions were stored by simply tagging rules with their owner’s name, and the user could then explore the consequences of applying different experts’ knowledge to the known facts. Similarly, towards the end of the Prospector system’s development, Reboh speculated on encoding separate models of expertise for each expert and then using sensitivity analysis on these models to identify the most significant points of disagreement. From this analysis, suggestions and advice to the user can be generated. We discuss this in more detail in the context of oil exploration systems in Section 2.2.4.

While keeping multiple, separate knowledge-bases is a straightforward idea, two problems still remain. First, it is a potentially time-consuming task to construct several knowledge bases if each must be engineered separately through interviews with different experts. In Prospector, for example, it took over a year to encode only a limited portion of a single expert’s knowledge [Gas80]. Second, even given several knowledge-bases have been constructed, substantial difficulties remain for comparing and contrasting their reasoning. A means of overcoming these difficulties is a necessary part of a practical argumentation model.

‘Learning Apprentice’ Systems

Another requirement we make is that models of users’ opinions can be easily maintained by the users themselves. This seeks to overcome some of the problems caused by the static nature of knowledge within expert systems (‘the fossilisation problem’ [Stu87]). Some research in knowledge acquisition and machine learning has addressed this. In particular, ‘learning apprentice’ systems learn through experience gained during normal problem-solving and are thus of direct relevance.

An early ambitious example of a learning apprentice is the rule-based system Teiresias [Dav77]. If the user disagrees with the system, the focus of the disagreement (a rule or missing rule) is located by debugging techniques and the problem corrected by the user adding or deleting rules or conditions in a rule’s premise or conclusion. However, Teiresias had a number of limitations preventing its use on real-world problems, the major one being the translation of the users corrections, expressed in natural language, into Teiresias’s rule language.

Two more recent systems for run-time knowledge acquisition are the classification systems Protos [BPM89, BPW90] and Srinivasan et al’s system [SCM+91]. Both allow the user to correct the system, the user providing/selecting corrections expressed directly in a constrained, artificial language. In Srinivasan et al’s system, a methodology called ‘ripple-down rules’ is used, whereby the user selects appropriate refinements to a knowledge-base from a list of possible refinements, the list being generated by the system comparing correctly classified and misclassified examples.
Other learning apprentice systems have sought to improve efficiency rather than alter knowledge by learning at run-time (eg. LEAP [MMS90]).

Again, a number of points of relevance should be made. First, the general learning problem is extremely difficult to solve. The systems above achieved some practical success (in itself a major achievement), but were dependent on an adequate, bounded vocabulary being provided for the users to use, constraining their scope of application. Second, their objectives (knowledge acquisition) differ from ours (supporting experts). As a result, issues such as comparing and justifying opinions are not addressed. While the ripple-down method, for example, could be adapted to refine separate knowledge-bases for each user, comparisons across the resulting differing knowledge bases would still pose a problem as discussed in the previous section.

**Justification and Case-Based Reasoning**

Finally, argumentation involves justifying as well as presenting (‘explaining’, in earlier expert system literature) a line of reasoning. This involves locating and presenting supporting evidence for some knowledge. There are many different types of support, for example statistics, analogy, precedent, etc. Wick and Slagle identify as many as seventeen types [WS89]. The use of cases (specific instances of solved problems) as evidence plays a role in many of these, and work on *case-based reasoning* has highlighted the value of cases in this role [RVA84].

We wish to draw on these techniques to provide justification facilities. This requires that not only general knowledge is stored, but also databases of cases are maintained and exploited as sources of evidence supporting or contradicting that knowledge.

### 1.7 Application Domains

Before presenting a model of argumentation which could meet our requirements, we turn our attention to possible application domains in order to illustrate the practical as well as theoretical relevance of an argumentation approach.

We have stated that eristic problem-solving can be seen as an extension of, rather than a replacement for, a conventional expert system methodology. The essential characteristics which may make a domain intractable to the standard approach are:

- Lack of agreement between experts, and discussion between experts forming an important component in decision-making
- Inability to represent all the necessary knowledge
- Knowledge changing over time and with experience

This research was motivated by the particular task of risk assessment in hydrocarbon exploration, a domain which has these characteristics. We now briefly describe this and some other domains which have similar characteristics, and which could be suitable for the application of an eristic approach.

#### 1.7.1 Risk Assessment in Geology

**The Risk Assessment Task**

An important geological problem is that of risk assessment in hydrocarbon exploration. Geologists refer to this as *prospect appraisal*. In the context of this thesis we use the terms prospect appraisal and risk assessment as synonymous estimating the probability of finding hydrocarbons at a particular location. However, it should be noted that the overall exploration risk depends on other factors also, such as estimating the expected volume of hydrocarbons, the recovery rate (ie. the proportion of hydrocarbons extractable) and the hydrocarbon quality.
The task of prospect appraisal is formidable. It takes several months to gather, process and interpret the appropriate seismic and well log data, and involves numerous discussions among experts as to what the appropriate likelihood of hydrocarbons should be. The reasoning is complex, involving activities such as seismic pattern recognition, spatial reasoning and mental visualisation of 3D rock structures, and their likely behaviours over time. The petroleum geologist David Hawkins cites a “conservative list” of the disciplines involved as including geophysics, seismology, mineralogy, log analysis, sedimentology, stratigraphy, paleontology, geochemistry, and statistics [Haw83]. Additionally, substantial experience of previously appraised prospects in the same geographical region is essential.

During appraisal, experts often meet to discuss interpretations and opinions. These interactions are motivated because experts have limited experience (there is more relevant data available than any one expert can completely internalise). In addition, they “don’t know what they don’t know”, i.e. they do not always know what other relevant information might be available. Thus by presenting an opinion to a fellow geologist they are also testing the scope of their own knowledge, seeking to ensure they have considered all relevant information and interpretations. This distinguishes eristic interaction between experts from the other type of interaction which occurs, when the expert simply asks another for a specific piece of information (a kind of ‘database lookup’).

Argumentation also serves another identifiable purpose, namely to help a geologist articulate his or her reasoning. The geologist can be viewed as reaching an opinion partly through methods difficult to articulate, eg. visual pattern recognition, mental geometrical matching of prospect structures etc. In our interviews, geologists often mentioned their ‘gut feeling’ about the prospect, referring to conclusions they had reached in ways difficult to articulate. Thus the expression of an argument can be viewed as an attempt to translate this reasoning into a communicable form, and argumentation with other geologists can indicate weaknesses in this translation process.

Thus several points can be made about this domain. Firstly, it is complex, involving processes beyond AI’s current state-of-the-art to adequately represent. Secondly, it is controversial and disagreement is common. Finally, discussion and argumentation among experts play an integral part of the risk appraisal process, fulfilling several important functions described above.

**AI Approaches**

We make some brief comments about the way existing geological expert systems have tackled this domain. In Section 2.2 we provide a more detailed review of these and other systems.

First, prospect risk appraisal is only one task of many in the energy industry, and the majority of successful geological expert systems have addressed other, better-defined tasks – in particular data analysis and presentation tasks (eg. Foenix [And89] and Biomarker [WG91] for geochemical analysis and report generation, Geologix [And89] for helping geologists link up rock strata, and by MacCallum et al. [MBS88] for inspection scheduling).

A smaller number of systems have been applied to this particular task of geological risk assessment, but have addressed different issues to those we have identified in this thesis. In particular, the knowledge representation issue has been an important focus, examining how risk assessment knowledge can be represented and combined. The systems SPII-2 [LMCP87] and Explorationist studied the use of fuzzy logic and a modified Bayesian technique respectively for encoding and propagating risk estimates, the work intended primarily as studies of knowledge representation techniques rather than focusing on their integration into a geological exploration environment.

The well-known Prospector system [DGH81] also pioneered important Bayesian techniques applied to several tasks, including risk assessment, in mineral exploration. Prospector successfully demonstrated that models whose conclusions closely matched those of an expert could
be constructed, but encountered some of the now familiar hurdles when integrating it into an exploration environment. “It is rather common for geologists to disagree among each other” Gaschnig reports [Gas82], and at that time optimistically hoped that discussion of Prospector’s models among geologists would serve to “smooth out” such disagreements. Later, the authors speculated on a number of techniques in which Prospector could be better integrated in the eristic environment of mineral exploration, including enhancing its use for distributed cooperation (eg. case retrieval functions [Dud80], sensitivity analysis [Gas82]) and eristic cooperation (eg. representing multiple opinions [Reb83]). However, by that time resources had expired, and Prospector was reported in 1984 to have been “largely left on the shelf” since 1980 [Joh84].

1.7.2 Business Decision-Making

The business domain shares many characteristics with the risk assessment domain. An analysis by Premkumar [Pre89a] cites, among other characteristics, that

- the problems are not well-structured
- the knowledge domain is not well-defined
- incomplete and probabilistic information is used
- the acquisition of knowledge is evolutionary and continuously modified and refined with experience to suit new circumstances
- expertise is distributed among multiple experts requiring integration

The majority of successful systems applied to this domain have acted in the role of assistants, ie. adopting the distributed approach to cooperation with experts. Decision support systems have been reported to have been particularly successful, allowing analytic decision models to provide extra information to assist decision-makers eg. Finsim, providing financial simulation and analysis to support decisions [KM90], by Nishikawa et al., providing analytic tools for inventory planning and control [NNSN86], and by Malmberg et al. [MBK87] for maintaining stocks of spare parts.

1.7.3 Financial Marketing

Kastner at al. [CAG+86] report work in applying expert systems to the domain of financial marketing. Again, this domain has characteristics making a normal expert system approach difficult. They write

“This domain differs from the great majority of previous expert system domains in that there are no well-defined answers (in the traditional sense); the goal here is to obtain satisfactory arguments to support the conclusions made” p71

Again, the authors identify the importance of the computer not as problem-solver but as an extended information resource. They write

“A typical financial marketing problem frequently has no one solution. There might be no definitive answer to a problem. The issue is not merely a question of computing financial optimality. The importance lies not only in the answer you provide but also in the explanation and justification that you use to back the answer. For this purpose, it is important to generate a convincing financial argument that strengthens the selling of the answer.” p71

Again, the authors adopt a model of distributed cooperation in which their system, FAME, provides specialised expertise for guidance and planning.
1.7.4 Legal Reasoning
Legal reasoning and legal expert systems have received considerable attention in the AI literature [Ris89]. Again, there is not always a universally agreed on right answer and expert opinion will differ. This disagreement and argumentation among legal experts is recognised as an integral part of law, with institutions (eg. courts) established to formally handle these.

Study of law in AI has focussed not so much on finding the best legal answer but on the legal argumentation process itself. Here, argumentation and the search for pointers to previously unconsidered, relevant evidence is recognised as part of the problem-solving process, with the use of legal precedents playing an important role. Example systems include Hypo [RVA84, AR87] which seeks to generate useful hypothetical examples for consideration in a legal case, and Taxman which searches for appropriate ‘deformations’ of a current case to help match it with an old case [MS82].

Legal arguments are complex, and the majority of AI work in this area has focussed on the specialised task of case representation and retrieval rather than full argumentation. Although this domain is an obvious candidate for an eristic approach, substantial issues of knowledge representation would need to be addressed further. This domain would seem a good advanced test for an eristic system.

1.7.5 Architectural Design
In their study of the architectural design process, Klein and Lu highlight the cooperative nature of the task [KL89]. Again, there is no one identifiable right way of designing architecture, and they argue that the process of discussion and conflict resolution among experts constitutes an important part of the problem-solving process.

1.7.6 Archaeology
Archaeology is a domain which involves not only routine analysis but also much interpretation of available data. Stutt [Stu87] gives a detailed description of reasoning in this domain, highlighting its semi-formal nature and the limitations of the autonomous expert system in this domain. He gives a detailed exposition of the eristic or argumentation approach as a model of problem-solving which can overcome these limitations.

1.8 Summary
Thus we have identified argumentation as a particular type of collaborative problem-solving, fulfilling a number of important roles for reasoning in ill-structured domains. We now give a general review of several related areas of work, namely expert systems in hydrocarbon exploration (our chosen application domain), decision-support systems, and hypertext systems for argumentation, to identify how they address issues of argumentation and reasoning in ill-structured domains. We describe their contribution to our goal and their relationship to this thesis work, and discuss issues of evaluating such systems. The argumentation approach can be seen as a natural progression of work in these three areas; we present our model of argumentation in Chapter 3.
Chapter 2

General Review

2.1 Introduction

In the previous chapter we presented argumentation as a particular style of cooperative problem-solving. We introduced several AI technologies related to argumentation – critiquing systems, multi-expert systems, learning apprentice systems, and case-based reasoning – and examined their strengths and weaknesses for use in argumentation. We also introduced a number of application domains where argumentation and conflicting opinion have been identified as significant factors in expert problem-solving, namely risk assessment in geology, business decision making, financial marketing, legal reasoning, architectural design, and archaeology.

In this Chapter, we provide a general review of three areas. First, as the original motivation for this work arose from experience in the oil exploration domain, we review software and AI methods applied in the oil industry and their relevance to our proposed approach. Second, as we are interested in argumentation as a means of supporting decision-making, we review the relation of work in decision-support systems (DSSs) to our goal. Third, we survey hypertext approaches to argumentation, a technology allowing structured, open-ended documentation of arguments. We discuss their strengths and weaknesses for decision-support.

Finally, we discuss the difficult issue of evaluating such systems. This provides the context for our evaluation in Chapter 5.

2.2 Expert Systems in the Oil Industry

2.2.1 Introduction

Our goal is to develop a practical model of argumentation, and evaluate its application taking oil exploration as a target domain. The purpose of this section is to explore our goal from the viewpoint of this application. There has been substantial development of software and expert techniques in the oil industry. Here we review their scope, relevance, and contributions to our objectives.

First, an important point should be made. The oil industry is wide and diverse, spanning an enormous range of disciplines (e.g. geology, engineering, mathematics, process control, microbiology). Applied systems all deal with specific sub-areas within this field, and thus the oil industry should be thought of as a context for application rather than an application domain in its own right.

Because of the diversity of applications which the oil industry encompasses, we first provide a brief but comprehensive summary of expert systems and related software for this industry, and then identify and review in more detail those of particular relevance to our application area.
2.2.2 Overview of Expert Systems in the Oil Industry

Introduction

This overview is divided into three parts, summarising published systems by the oil companies BP and Shell, and then from the general literature. The two company surveys provide a useful picture of practical expert system application within them, complementing the overview of other commercial and academic research. Following this we characterise the type of application they are targeted at, the degree of success they achieved, and their relation to the application motivating this thesis.

BP International

BP International's Information Systems Services have developed a number of expert systems, ranging from assisting in oil platform design to small one-off programs for specific problems. We summarise those described in the literature below.

GasOil was developed by BP for the design of vessels for separating gas, oil, and water. The application was inductively generated from 3500 examples [SMZ86].

The Multifo system assists production and petroleum engineers to select the pressure drop correlations of the output from subsea wells [And89].

The jacket weight estimation system, by BP, calculates the structural weights of offshore oil and gas platform jackets, using a large number of rules and equations [And89].

The burner fault diagnosis system assists in the identification of problems on process combustion equipment, ensuring combustion to specification [And89].

Machtex is a system integrating a database package and vibration analysis facilities to provide a complete off-line machinery monitoring application [And89].

Dyce ECS provides screen-based advice to non-specialists who might be called on to cope with an event needing emergency services co-ordination [And89].

BioMarker was designed to help geochemists identify the depositional environment of a rock from its biological 'signature'. The system is akin to a hypertext encyclopedia of different biomarkers and their identification [WG91].

Shell Research

Similarly, we summarise documented systems developed by Shell.

Foenix was designed to assist in the interpretation of and reporting on hydrocarbon samples, following geochemical analysis in Shell's laboratories. The system interprets the analysis, held on a conventional database, and produces reports using desk-top publishing techniques [And89].

Geologix helps geologists link up rock strata in different wells, facilitating oil field development [And89].

Dentix was designed as an intelligent front end for algorithmic software assessing collision damage to offshore oil platforms [And89].

The seismic parameter selection system was intended to assist in the choice of parameters for a particular seismic processing step, developed using rule induction techniques [And89].
The solids control system designs an optimum process for treating drilling mud [And89].

Ressix was an intelligent interface to gas and oil reservoir simulators, aiming to help petroleum engineers perform simulations [She87].

General Literature

Denvad identifies the depositional environment of a given rock formation, using well logs, their lithological interpretation, and dipmeter measurements [Yan86].

Dipmeter, by Schlumberger, determines the inclination of rocks from well log data interpretation [SB83].

Elas is a production rule expert system which acts as an intelligent front end to conventional well-log interpretation software [AW87].

Explorationist was a prototype system built to assist petroleum geologists assess the amount of oil and gas in unexplored regions. It represented geological knowledge in a network structure developed from Prospector’s Bayesian networks [RFTT85].

Gridding advisor, a small rule-based system, helps users select one of six gridding algorithms by matching characteristics of the data with capabilities of the gridding algorithm [Mas87].

Litho was designed to provide lithological interpretation of well log data, determining the main lithological type, palaeo-environment, and plausible lithofacies for each zone in the log [BD83].

Mendel aims to provide the most consistent description of oil reservoirs and fluid content using well log data [vH86].

Prospector was a rule-based system to aid geologists in evaluating the favourability of an exploration site for occurrences of mineral deposits. This task is very similar to our application of interest, namely oil prospect evaluation [DGH81].

SDIES is a rule-based system ambitiously aimed at seismic data interpretation [YL86].

SPII-2 is a backward chaining inference engine employing a fuzzy logic representation of uncertainty. It was applied to part of the task of prospect risk appraisal, the same application as this thesis [LMCP87].

The Practical Success of these Systems

It should be noted that, despite the research effort expended, only a minority of these systems have been independently reported to have achieved commercial success. Shell’s systems were selected (by Shell) as examples of 25 systems they claim to be in actual effective use (as of 1989), with the exceptions of Ressix and the seismic parameter system, both which were eventually abandoned [And89] p32. BP’s systems were claimed to be among 60 which were installed by 1989, but no information about their usage was provided. Of these, GasOil was independently reported to be in use in four company sites in 1987 [Her87], and Biomarker was reported as not yet being in regular use as of early 1991 [Bra90]. Of the other systems, a thorough literature search could find no evidence of any of them achieving regular usage (see Table 2.1), apart from Dipmeter of which it was reported that there were several versions that the field had used [Bra90]. The extent of this usage is not known. At a symposium in London on Expert Systems in Geological Interpretation [Vid91] in February 1991, only 2 of the 10 presented systems were in use, one of which was Optimist. The other was a data management system handling data collected in gold exploration, producing summary reports on areas for the users as requested.
2.2. EXPERT SYSTEMS IN THE OIL INDUSTRY

<table>
<thead>
<tr>
<th>System</th>
<th>Final status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prospector</td>
<td>Installed but “largely left on the shelf” [Joh84]p216</td>
</tr>
<tr>
<td>Explorationist</td>
<td>Prototype built 1985, but no further references could be found.</td>
</tr>
<tr>
<td>SPII-2</td>
<td>Reported to be tested (no further refs).</td>
</tr>
<tr>
<td>Litho</td>
<td>Reported to be validated (no further refs).</td>
</tr>
<tr>
<td>Denvad</td>
<td>Designed as a prototype (no further refs).</td>
</tr>
<tr>
<td>Mendel</td>
<td>Prototyped developed (no further refs).</td>
</tr>
<tr>
<td>Dipmeter</td>
<td>Currently offered as a service by Schlumberger [Bra90]</td>
</tr>
<tr>
<td>Elas</td>
<td>Prototype only; not developed into commercial system.</td>
</tr>
<tr>
<td>SDIES</td>
<td>Research project only; no evaluation or further references.</td>
</tr>
</tbody>
</table>

Table 2.1: Final reported status of oil exploration systems from the general literature.

Discussion

A characterisation of the type of task the overviewed systems are performing is important for this discussion and presented in Table 2.2. In several cases the systems’ authors use the term ‘expert system’ rather loosely, including systems for algorithmic data analysis and documentation.

We make several points. First, as just described, the success of expert systems and related software in the oil industry is clear but qualified. Part of the difficulty in achieving success reflects the general complexities of developing real-world applications. However, part of the difficulty can also be attributed to the tasks targeted for the systems. The most successful, as described by BP and Shell, have mainly involved well-defined data analysis tasks, and reflects an awareness of the scope of the applicability of the normal expert system approach (Section 1.4.3). This contrasts with the focus of our thesis on tasks in which the lack of analytical procedures may result in different, possibly conflicting lines of reasoning being followed.

2.2.3 Review

We now select the more relevant of these systems for more detailed examination. The two selection criteria we use are first, systems dealing with interpretative rather than analysis tasks (ie. where well-defined solution algorithms do not exist), and second, systems performing risk assessment. We wish to learn what techniques and lessons can be learned from these systems.

2.2.4 Prospector

Prospector is relevant to this thesis as it was targeted at a similar problem to that motivating this work, mineral (rather than oil) prospect evaluation. Its contributions to this work concern three aspects: the suitability of the ‘autonomous expert’ model of problem solving, its knowledge representation, and the evaluation issues which it presented. In addition, pointers from Prospector’s authors’ to possible future developments are relevant.

Description

Prospector is essentially a rule-based system to aid geologists in evaluating the favourability of an exploration site for occurrences of mineral deposits. The earlier versions of Prospector
<table>
<thead>
<tr>
<th>System</th>
<th>Task</th>
<th>System</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>BP</td>
<td>Design</td>
<td>Literature</td>
<td>Interpretation</td>
</tr>
<tr>
<td>GasOil</td>
<td>Data analysis</td>
<td>Denvad</td>
<td>Interpretation</td>
</tr>
<tr>
<td>Multiflow</td>
<td>Data analysis</td>
<td>Dipmeter</td>
<td>Front-end</td>
</tr>
<tr>
<td>Jacket weight</td>
<td>Front-end</td>
<td>Elas</td>
<td>Interpretation</td>
</tr>
<tr>
<td>Burner fault</td>
<td>HyperText</td>
<td>Explorationist</td>
<td>Front-end</td>
</tr>
<tr>
<td>Machtex</td>
<td>HyperText</td>
<td>Gridding advisor</td>
<td>Front-end</td>
</tr>
<tr>
<td>Dyce ECS</td>
<td></td>
<td>Litho</td>
<td>Interpretation</td>
</tr>
<tr>
<td>Biomarker</td>
<td></td>
<td>Mendel</td>
<td>Interpretation</td>
</tr>
<tr>
<td>Shell</td>
<td>Interpretation</td>
<td>Prospector</td>
<td>Interpretation</td>
</tr>
<tr>
<td>Foenix</td>
<td></td>
<td>SDIES</td>
<td>Interpretation</td>
</tr>
<tr>
<td>Geologix</td>
<td></td>
<td>SPI-2</td>
<td>Interpretation</td>
</tr>
<tr>
<td>Dentix</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Seismic param</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Solid control</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resisix</td>
<td>Front-end</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Data analysis refers to data processing by algorithmic means, i.e., where well-defined procedures for manipulating the data exist. Data interpretation refers to problems where no well-defined algorithm and/or target solution exist. Front-end refers to applications providing an intelligent interface to existing data processing packages.

Table 2.2: A characterisation of application tasks in the reviewed systems.

...took as input a description of a prospective mineral site, and determined how well this site data matched the various models of mineral deposition encoded in it [Dud80]. Later these models could also work in conjunction with site-specific, map-based information to compute a map of favourability of any chosen area for mineralisation. The system was targeted for use by field geologists, to act in the role of a consultant.

The knowledge base of Prospector contained a collection of models of different classes of ore deposits. Each ore deposit model was encoded as a modified Bayesian inference network, in which nodes represent field evidence or important geological hypotheses and arcs represent the relationship between the nodes. An example of field evidence might be “The host has thick, high angle cross beds” while a geological hypothesis might be “The prospect is in an aeolian sequence”. The top level assertion in each network is the assertion that the available evidence best matches that particular model of ore deposit, and its associated probability gives the strength of that match.

The network can be seen as constructed from rules of the form $E \rightarrow H$, where $E$ and $H$ are nodes in the tree and the arc represents the rule itself. Each node was assigned a prior ‘probability’ of occurrence (the interpretation of the assigned numbers as probabilities is difficult, hence we caution the use of the term). Arcs were labelled with two measures, $LS$ and $LN$, representing the two likelihood ratios:

$$LS = \frac{p(E|H)}{p(E)}$$

$$LN = \frac{p(E|H)}{p(E|\neg H)}$$

$LS$ is known as the sufficiency factor of $E$ for $H$ and $LN$ is the necessity factor of $E$ for $H$. These factors were couched in terms of odds and are related to $E$ and $H$ by the following
Figure 2.1: Interpolation with uncertain evidence in Prospector.

formulae:

\[
\begin{align*}
o(H|E) & = LS \cdot o(H) \\
o(H|\overline{E}) & = LN \cdot o(H)
\end{align*}
\]

where \( o(X) \) is the prior odds of event \( X \) and \( o(X|Y) \) is the conditional odds of \( X \) given event \( Y \). Given that \( E \) has posterior probability between 0 and 1, the posterior probability of \( H \) is computed by a piecewise linear interpolation in probabilities as shown in Figure 2.1. Evidence was combined either by assuming independence or by using logical combinations AND, OR and NOT as appropriate; the reader is referred to [DGH81] for further details.

A heuristic control strategy was employed to determine which piece of evidence to next request of the user, based on a scoring method of model to pursue and an analysis of which question’s resolution would have the greatest effect on the goal.

Prospector's later models were developed with the assistance of the tool KAS [Reb81]. By the time of the final report in 1980 there were twelve models encoded.

Despite its extensive development and evaluation, Prospector did not achieve regular commercial use and was reported in 1984 to have been “largely left on the shelf” since 1980 [Joh84], although a programme to extend it, underway in 1984, was reported by Walker [Wal87]. Its famous discovery of a large molybdenum deposit in 1980 was an example of it being predictively tested rather than in routine use [CHDH82]. Despite this, Prospector made significant and influential contributions to AI research, especially in the areas of knowledge representation and Bayesian inference. Its success was thus primarily scientific rather than economic, stimulating substantial new research in expert systems (eg. the systems Hydro [RRG82], Conphyde [BA82], and Explorationist, described in Section 2.2.6).

### Discussion

Prospector has several points of relevance to this thesis work.

First, Prospector’s knowledge representation offers a possible scheme for encoding geological
knowledge in an argumentation system. Its adequacy was partially demonstrated by the authors' when they showed that Prospector's advice closely matched that of the modelled expert. However, a number of limitations can also be identified:

- Engineering the models was a time-consuming task, for example taking a year to develop just three uranium exploration models despite the assistance of extensive knowledge acquisition tools [Reb81]. In the final reports the authors view the small number of encoded models (twelve) as one of the main limitations of the system [Dud80].

- Geologists found the networks, and hence the explanations given by the system, difficult to understand. “Few geologists can even evaluate Prospectors models, let alone contribute to their development” Duda reports. Geologists found it difficult to specify network parameters directly, and later the knowledge acquisition system KAS [Reb81] was used to help in this task.

Second, much of the reported experience of the developers illustrates the limitations of the ‘autonomous expert’ model of problem-solving in this domain. Prospector was unable to take into account disagreements among experts, the full significance of which was not reported until towards the end of the project. (The early evaluation of Prospector compared how well its conclusions using a single geological model matched those of the expert who designed the model [DGH81]). Following field trials in 1980, Gaschnig reported that “it is rather common for geologists to disagree among each other” [Gas82] and he optimistically hoped that discussion of Prospector’s models among geologists would serve to smooth out such disagreements. It was also found that experts sometimes disagreed on the answers to the initial input questions. Sensitivity analyses showed that in some cases these differences could have large effects on Prospector's conclusions [Gas82]. These factors raised significant problems for validating the system to the satisfaction of its potential users.

Finally, related to the ‘autonomous expert’ model, Prospector’s role was essentially one of explainable problem-solving. However, as other authors have also noted (eg. [MD85]), this is not the only or most appropriate role for a knowledge-based system. Although Duda speculated on a variety of additional functions which it might perform (eg. search and retrieval of similar cases, sensitivity analysis), these were not incorporated in the system and again limited the support which it could offer the user in decision-making.

These problems were recognised and discussed at length by the designers in various articles [Dud80, Gas82]. The authors speculated on a number of ways Prospector could be further developed to alleviate them, but at this stage development of Prospector was coming to an end. This thesis work is closely related to those suggestions, including:

**Pointers to Previous Cases** The identification and use of previously explored geological sites plays an important role in geological assessment. Duda speculates on the use of Prospector for similar case retrieval:

“Prospector could compare the replies made by the user during the consultation session with those for the known deposits on record, and determine which of these the user's case matches most closely. ... For example, Prospector might report

‘the answers you have given indicate that your prospect is very similar to the Yerington deposit, except that in your prospect the intrusive system is not as favorable as the one at Yerington, for the following reasons:...’” [Dud80]

**Representation of Different Opinions** In 1983, Reboh speculated on encoding separate models of expertise for each expert and then using sensitivity analysis on these models to identify the most significant points of disagreement. From this analysis, dialog and suggestions
to the user as to how to proceed would occur. He suggested a hypothetical dialogue of the form:

“If you were to consult with expert A, he would suggest...
If you were to consult with expert B, he would suggest...
However,
IF you believe that observation $E_1$ has a strong positive effect on intermediate hypothesis $H_1$ THEN I suggest you follow A’s advice.
IF, on the other hand, you don’t believe THAT, and you believe strongly that observation $E_2$ (which, by the way, has not been considered by A) has a negative effect on intermediate hypothesis $H_2$, THEN you should follow B’s advice.
OTHERWISE supply me with the following additional information:...” [Reb83]p149

However, although these suggestions were made, many of the issues they raise were left open. Duda’s case retrieval suggestion was based on exploiting the database of known facts about cases, but did not extend to include the recording of and matching by experts’ conclusions about those cases. As described earlier, (Section 1.7.1) these conclusions play an important role in geological debate. Issues of assessing case similarity and integrating case retrieval into Prospector’s reasoning were not addressed. Reboh’s speculation on representing different opinions is highly relevant to this thesis; however, he did not explore issues of constructing, maintaining, and comparing multiple knowledge bases. The developers difficulty of constructing even a single model for a single expert suggests that a multi-expert system within the Prospector framework would be impractical to construct and maintain without additional facilities being added. This thesis work can be seen as starting from the point where Prospector’s construction came to an end, developing some of these suggestions in a model of argumentation and addressing these issues directly.

2.2.5 Dipmeter Advisor

While Dipmeter Advisor was targeted at a different geological task (dipmeter log interpretation), the complex and interpretative nature of the reasoning involved makes it of relevance to this thesis.

Description

Dipmeter Advisor [SB83] was developed at Schlumberger-Doll Research for inferring subsurface geological structure from dipmeter logs and general geological knowledge. The system was built using a blackboard architecture, employing sets of production rules reflecting the expertise of human advisors. The system was developed over several years (1981-87) at a reputed cost of $20m by 1984 alone [And89].

The task of dipmeter interpretation was divided into eleven successive phases. After each phase of analysis, the user was engaged in an interactive dialogue with the system. The user could examine, delete, or modify conclusions reached by the system or even add his or her own conclusions. The user could also revert to earlier phases of the analysis to refer to conclusions or to rerun the computation.

The production rules in the knowledge base were thus partitioned into several distinct sets according to the phase of analysis. The sets of rules were evoked in a forward chaining, data directed fashion. Conflicts were resolved by rule order. Multiple conclusions could be drawn for the same zone (region of constant lithology) in the log, the user left to resolve which conclusion is most appropriate.

It was reported to have been tested in the field in 1985 [Lan85] and 1986 [Fra86], but details were not given. The earlier report by Langley did not discuss how the system was validated. The later report did not discuss this issue except to say that it proved difficult to specify appropriate
criteria for validation of the system as human experts were observed to frequently disagree among themselves in their evaluations of dipmeter logs. In October 1990 Dipmeter was reported to have been deployed on Xerox Lisp machines that the field used (it is not known if this refers to the field trials or subsequent use), and is currently offered as a service by Schlumberger in its Field Log Interpretation Centres [Bra90]. There are plans to incorporate it in Pleiades, a Geophysical and Petrophysical Interpretation Workstation.

Discussion

The main point of relevance to this thesis is the way in which the authors sought to accommodate users’ disagreements with the system. Problems were first reported during the initial evaluation in 1983, when the designers observed that “there are often disagreement[s] even among experts”, and questioned with hindsight the wisdom of using a single expert during the systems’ development [SB83]. Additionally, they describe their experts as “moving targets”, referring to the changing opinion of individual experts. This also complicated the evaluation of Dipmeter, and re-enforces the point that expert systems in geological interpretation must be so designed that their knowledge bases can be easily maintained.

Dipmeter’s primary means of addressing these problems was to keep the user ‘in the loop’. As described, the user could add, delete, or modify Dipmeter’s conclusions at any point during the interpretation, and sometimes Dipmeter would call on the user to resolve conflicts. Substantial data manipulation and graphical facilities were also provided for the user. This marks a shift from the initial conception of a more autonomous system [DAC+81] to one which was mixed-initiative and interactive.

To summarise, Dipmeter accommodated user disagreements through the use of interactive problem-solving. It thus illustrates a first step towards an eristic system, and indicates one way in which expert systems can be applied to ill-structured tasks. However, Dipmeter made no attempt to record or exploit disagreements as a resource of information in its own right, and instead support for the user was primarily centred on the data manipulation facilities it provided. Our work here can be seen as development of the interactive approach but focussed instead on the information which can be acquired from that interaction.

2.2.6 Explorationist

Explorationist was a prototype expert system to assist petroleum geologists in assessing the amount of oil and gas in unexplored regions [RF85]. No evidence could be found of it progressing beyond the prototype stage. The one report on the system in the literature only describes the knowledge representation scheme used. Issues of its incorporation into the full geological system (eg. comprehensibility, accuracy, ease of knowledge acquisition, multiple opinions) are not mentioned, and thus this work has a limited contribution to this thesis. The main item of relevance is the knowledge representation scheme used.

Explorationist represented knowledge using a semantic network, designed to reason with both numeric and symbolic information. This contrasts with Prospector’s Bayesian networks, where reasoning about numeric quantities was not incorporated. Like Prospector, conditional probabilities are used to quantify relations between nodes in the network. A key feature of the representation is the lack of assumptions regarding the dependencies between variables, obtained at a cost of additional probability estimates being required of the expert.

As a knowledge representation scheme, its structure and use of conditional probabilities are similar to Prospector, and thus likely to suffer the same difficulties in terms of comprehensibility and knowledge acquisition; this is likely to make it inappropriate for an argumentation system where it is essential the user has an intuitive feel for the reasoning which the system proposes.
2.2.7 SPII-2

SPII-2 was a simple backward chaining inference engine, employing a fuzzy logic representation of uncertainty [LMCP87]. It was applied to part of the task of prospect risk appraisal, the application of interest in this thesis. As with Explorationist, however, the single report on it describes only the knowledge representation scheme used. No evaluation or issues of its practical use were given, and thus again only its knowledge representation for risk appraisal is of relevance to this thesis.

SPII-2 dealt with the problem of representing vague concepts and uncertainty of inference through the use of fuzzy set theory. Both of the former are characterised by possibility distributions derived from the theoretical work of Zadeh [Zad88]. Approximate numerical values are represented by trapezoidal distributions which restrict the more or less possible values that the ill-defined real quantity can take. Distributions representing the addition or multiplication of these values can be computed provided the assumption is made that the facts do not interact.

While SPII-2 allowed computation from a fuzzy probabilistic model, the problem of how such a model is constructed in the first place and maintained was not addressed; for an argumentation system, this issue is particularly important. It does, however, illustrate an alternative inference scheme for risk appraisal in which experts can specify a range of answers to questions with an associated probability distribution. We discuss its suitability for argumentation later (Section 6.2.6).

2.2.8 Geologix and Foenix

Finally, some discussion is merited of Shell’s systems Geologix and Foenix. These were the only two systems described by BP and Shell for geological interpretation tasks (as opposed to data analysis etc., see Table 2.2), and unlike the previously discussed systems, it was claimed they had achieved active use. Their relevance is in the way they successfully applied expert system technology. Full reports on these systems are unavailable; however, comments from Shell’s then Netherlands research manager Van de Kraats are informative [And89].

Most importantly, both Geologix and Foenix were designed as assistants to the experts. As with Dipmeter Advisor, their strengths were their support of data analysis and documentation functions. The knowledge-based processing they performed were reported as secondary to these data analysis functions: Van de Kraats comments that “Geologix became a success when we introduced interactive manipulation of logs. The addition of the knowledge base was considered a bonus”. Similarly Foenix’s main strength was the high quality reports it generated. “Even if Foenix did not have a knowledge base it would probably still be a success,” Van de Kraats reports, “because the system makes it so easy to work with all the data and produce the report”. With both Geologix and Foenix, the expert could simply over-ride the system’s recommendation if he or she disagreed.

They key point here is again to illustrate that expert systems have been applied in ill-defined geological tasks, but also to show their success has been through interactive methods focusing on data analysis and presentation. In the two systems by Shell, the computer performed the routine tasks, freeing the expert up for more complex decisions. BP’s 1989 Information Systems manager reports a similar philosophy behind their work: “experts agree 80% of the time, on average. This is the aspect of their work which an expert system may be built to handle”. For Foenix and Geologix, this pragmatic approach appears to have allowed expert system technology to become incorporated in interpretative tasks, but at the price of not providing direct computer involvement in aspects where there is subjectivity and disagreement among experts. In contrast, it is precisely these aspects which we are seeking to support in this thesis work through a practical model of argumentation.
2.2.9 Conclusion

To summarise the main points of the review:

- While there has been much development of AI and related software in the oil industry, the success of these systems is limited.
- The majority of successful systems performed well-defined, analytical tasks in suitable niches in the oil exploration domain.
- Successful systems supporting users in ill-defined tasks concentrated on the well-defined data analysis and presentation aspects of the problem. Disagreements were tolerated by keeping the user ‘in the loop’, but not exploited as a resource of information in its own right.

The most relevant surveyed system is Prospector; this thesis can be viewed as taking up some of the authors’ suggestions for further developing an eristic approach for geological exploration.

2.3 Decision Support Systems

2.3.1 Introduction

While many of the oil exploration systems described fall into the category of decision support systems (DSSs), we now examine this as a subject area in its own right. We examine the way in which systems from the DSS community have approached the goal of aiding decisions, and its relationship to this thesis work.

Although DSSs have been around for several years, there is no universally accepted definition of these systems. Broadly, the term refers to computer programs which could support a manager in making ill-structured decisions. (eg. Keen and Scott-Morton offer the definition “a coherent system of computer-based technology...used by managers as an aid to their decision-making in semi-structured decision tasks” [Bid89]). More strictly, DSSs are often characterised as achieving this objective through the use of analytical decision models and access to databases [KM90].

Several research areas are closely related to DSSs including data processing, management information systems (MISs), operational research, and expert systems. Development and expansion of all these areas has resulted in the distinction between them becoming increasingly blurred. Expert systems, for example, often act in a decision support capacity (eg. Garvan-ES1 [CHQ+89]), while recent developments in the DSS community have expanded to include knowledge-based analysis techniques.

2.3.2 Examples

According to these definitions, several of the oil industry systems already described in Section 2.2 fall into this category. We provide some additional brief examples below, and follow this by a discussion of their relation to the argumentation approach proposed in Chapter 1. These examples have been selected as reasonably typical DSSs, the purpose of their description being to help characterise this research area further:

**Population Maintenance** Malmborg et al. describe a DSS for evaluating the maintenance requirements of ‘recoverable populations’ (ie. sets of repairable items) [MBK87]. The user supplies input parameters of failure rates, parts costs, delivery times etc., and the system runs a numeric maintenance model to produce information about maintenance costs and stock levels. Users can also perform 'what if' analyses by changing the initial parameters. This is particularly valuable if data is not available (eg. part failure rates), allowing them to estimate values and perform sensitivity analyses on those estimates.
2.3. DECISION SUPPORT SYSTEMS

Inventory Planning  The DSS of Nishikawa et al. supports inventory planning and control for managing interior building materials [NNSN86]. The system is based around an optimisation model for meeting the users' requirements (eg. target profits, minimally acceptable stock levels). In a similar way to the DSS of Maiborg et al (just described), users can perform 'what if' analyses to examine the effects of different decisions. This provides a valuable tool for allowing users to explore different ways of implementing overall policies. The system is in operation and is reported to have won popularity among its users.

Financial Analysis and Simulation  FinSim is a DSS system providing the user with a range of financial analysis functions, including an analysis of the financial history of a company and a simulation of the consequences of its main financial decisions [KM90]. It supports the user in four main ways:

- Data collation, analysis and presentation.
- Application of a numeric forecasting model to make predictions.
- A rule-based component providing financial diagnostic advice.
- Report generation.

2.3.3 Discussion

The bulk of work in DSSs, particularly the earlier work, approached the goal of decision support by performing analytic tasks that were too time consuming or difficult for the user to perform him or herself, the computer acting as a tool for the user. We described this in Section 1.5.1 as a form of distributed problem-solving.

There are several advantages of this approach to supporting decisions. First, the principle of complementarity – let the computer do what it is particularly good at, freeing users to do what they are good at – is a sound one. The computer has much to offer in terms of data analysis, speed and presentation. Second, it allows the wealth of research in modelling, planning, and optimisation to be applied to practical decision-making. Third, it eases the social problem of accountability for decisions, as the computer is not making decisions itself but supporting the human decision-maker.

Despite these, there are also limitations to this approach to decision support. First, although the decision problem itself may be ill-structured, the computer is still confined to parts of the problem which can be well-defined. Second, the user is receiving a particular type of assistance with his or her judgements, based on analytic modelling and data analysis (eg. viewing the consequences on a simulation model). Other types of assistance, though, are not supported; in particular more subjective sources of information such as experts' knowledge and opinions are not exploited.

More recent work in the DSS community has sought to include knowledge-based techniques in DSSs (eg. [KM90]), allowing DSSs to offer 'expert' as well as analytical advice to users. This is typically effected by encoding advice or ill-structured analysis methods in a rule-base (eg. part of FinSim, described earlier). This extends the types of support which can be offered, but also introduces many of the problems of knowledge-based reasoning which we identified in Chapter 1. For example, the user may disagree with the advice, or want more justification for it, or want to correct it for future users. Systems including static knowledge-bases for support are limited in their ability to respond to these problems.

This thesis work is concerned with these issues of knowledge-based support. In particular, we wish to view users' opinions and chains of reasoning as an extra resource of support information in its own right, complementary to the database information typically manipulated by earlier DSSs. Representing, maintaining, and exploiting users' knowledge as a resource raises substantial new problems which we have discussed at length in Chapter 1, and which we seek to address in this thesis.
To summarise, this thesis work can be seen as part of the ongoing progression of work in DSSs, developing the knowledge-based aspects of decision-support further. In particular we wish to exploit the interaction of the system’s and user’s reasoning to allow focussed support for the user and the automatic maintenance of knowledge within the system, thus extending the support which knowledge-based DSSs can offer in decision making.

2.3.4 Group Decision Support Systems

Introduction

A more recent area of decision support research is in group decision support systems (GDSS). These systems aim to improve the process of group decision making by removing common communication barriers, providing techniques for structuring decision analysis, and systematically directing the pattern, timing, or content of discussion [VNGD88].

Although included within the scope of DSSs, GDSSs differ substantially in character to DSSs. Most importantly their main role is to facilitating group interaction, aspects of analytic decision modelling and database access being either minor or absent.

Types of Group DSSs

Here we briefly list types of GDSSs with examples. We subsequently discuss their relevance to computer argumentation.

Decision Room A decision room GDSS provides computer support for normal group meetings in a conference room. Each participant has a terminal, and can contribute input to a large central screen visible to all (eg. [GL88]).

Local Area Network Systems Some GDSSs allow users to participate in group problem-solving while in different rooms through terminals connected together in a local area network (LAN). Examples are gIBIS and rIBIS. We describe these in more detail in the context of hypertext for argumentation (Section 2.4.2).

Remote Communication Systems It has been argued that multiple user telecommunication systems can even be regarded as GDSSs (eg. electronic mail, teleconferencing) [Bid89].

Discussion

GDSSs share some commonalities with our proposed argumentation approach, but also some important differences exist. We make several points of comparison.

First, while GDSSs are designed to support group decision making – an activity often involving argumentation – they are primarily intended as facilitators rather than participants in this process. They do not play an active role in argumentation itself. Their role is analogous to that of a non-participatory chairperson, namely to help and structure information flow between participants. This partly contrasts with our approach of computer argumentation in which a participatory component is added.

Second, although in both styles of GDSS opinions must be represented in some way, the requirements of that representation differ. In particular, some formal structure of argument content is required for computer participation. This contrasts with the coarser argument structures suitable for non-participatory computer GDSS. (We summarise some of these shortly in our review of hypertext argumentation systems.)

Finally we note that many GDSSs can also be used for problem-solving by individuals, providing a tool for single users to structure and organise ideas (Rein and Ellis describe the GDSS rIBIS being used in this way [RE91]). Computer argumentation can assist in this style of usage, automatically providing the feedback that other group members would normally provide.
2.4 Hypertext Approaches to Argumentation

We have reviewed oil industry expert systems, and the relation of decision support systems to our goal of argumentation. We now review work combining argumentation and hypertext. Although such work is scarce, it is also relevant to consider for two reasons. First, argumentation is a task primarily focussed on information organisation and retrieval, a task which is also a basic goal of hypertext systems. Second, as emphasised in Chapter 1, argumentation is an 'open-ended' process: the manipulation of uninterpreted text as an open-ended representation of knowledge by hypertext systems is thus of interest. Although uninterpreted text may appear inappropriate as a representational tool, research in hypertext suggests that its organisation can provide sufficient semantic information to allow its use in argumentation. We review these approaches here, surveying three hypertext systems applied to argumentation.

2.4.1 Euclid

Introduction

A fundamental philosophy underlying hypertext systems is that information can be usefully manipulated by virtue of knowledge of its structure alone. While components of information in a hypertext system are generally represented in a form semantically inaccessible to the system (typically as free text), these components are also organised together in a formal way, representing the structure of that information. Smolensky et al. use the term ‘semi-formal language’ to describe languages used to express this informal content/formal structure, and have partially designed their own semi-formal language, ARL, explicitly for representing and manipulating arguments in their system Euclid.

Description

Euclid is a hypertext system for “supporting users in reasoned argumentation” [SBF+87]. It is essentially a tool for documenting arguments, and can be seen as a specialisation of more general hypertext systems for relating ideas together on the screen. The user constructs an argument by typing in its component parts and indicating their relationship using the mouse. Resulting arguments can be stored, viewed or printed.

Smolensky et al. apply the formal structure/informal content distinction to arguments as follows:

- argument content refers to knowledge of the subject domain.
- argument structure refers to knowledge of argumentation per se.

For example “lower interest rates lead to bull markets” embodies argument content, while statements of argument structure might include

Claim C1 justifies claim C2
Claim C is the main point of argument A
Claim C1 made by author A1 contradicts claim C2 made by author A2
Their claim is that, by embedding an argument's content within a formal representation of its structure, Euclid will offer advantages for information retrieval, browsing of arguments, and comprehensibility of arguments.

The authors adopt a coarse structure of arguments, only partially specified in their publication and illustrated using Searle's well-known 'Chinese room' argument. The bulk of the argument network which the user constructs with Euclid depicts how primitive arguments combine (eg. the 'internalised Chinese room argument' refutes 'the person-neuron analogy argument') using connectives including supports, refutes and contradicts. The structure of primitive arguments, though, is not broken down further apart from identifying the main-claim as a distinct element within it.

The system was only partially implemented at publication in 1987, and was not evaluated by the authors. No evidence of its completion could be found.

Discussion

Despite its incomplete implementation, Euclid has several points of relevance to this thesis. Most important is the clearly articulated philosophy underlying the work, that argument structure can be formalised and combined with a textual representation for components within that structure, providing useful organisation of knowledge.

We make two other points of relevance. First, the actual structure of arguments proposed was limited. As other authors have shown (and we describe later), it is possible to break arguments further down into constituent parts, but this was not done in Euclid. This limited the structuring and manipulation of arguments which the system could offer. Second, as argument content was represented completely informally, the system was unable to perform any active reasoning. As a consequence, several of the useful argumentation operations we identified in Chapter 1 could not be supported eg. argument comparison, search and retrieval for evidence, and checking the user's arguments. In order to support these functions, some formalisation of argument content is necessary but this option was not explored in Euclid.

2.4.2 gIBIS and rIBIS

Introduction

gIBIS (graphical Issue-Based Information System) [CB87, BC88] and rIBIS (real-time IBIS) [RE91] are hypertext systems designed to support design deliberations among cooperating experts. As mentioned earlier, these systems are hypertext systems for group decision support, and thus span these two categories in this review.

We describe gIBIS below. rIBIS is an alternative implementation by different authors, which follows the same structure for deliberations as gIBIS.

Description

Using gIBIS, users can document their deliberations about specific issues in a structured way, constructing a network on the screen of issues, positions, and arguments in the deliberation. gIBIS can operate as a multi-user system, its graphical representation of the deliberations being worked on simultaneously by cooperating team members over a local area network (LAN) of computers.

The IBIS method is based on the principle that the design process is fundamentally a conversation among 'stakeholders'. gIBIS aims to facilitate, structure, and record this conversation or 'deliberation'. Users construct a network representing the deliberation from three basic types of node: issues, positions and arguments. An issue is an identified problem for the design. A position is a child node of an issue, and proposes a particular solution to that issue. An
argument is a child node of a position, and either supports or objects-to a position. We sketch this hierarchy:

Argument supports Position responds-to Issue

There may be one or more positions responding to an issue, and one or more arguments supporting or objecting to a position. There are six other connectives users can employ: specialise and generalise (to organise issues into a hierarchy), questions or suggests (to relate issues), and replaces and other.

gIBIS thus records the positions (on issues) of participants and their relationships. It is claimed to have achieved ‘wider and more prolonged usage’ than other hypertext systems in the authors’ environment. rIBIS is described as being used in 16 design sessions (the majority of which concerned improving the design of rIBIS itself) [RE91]. Users reactions to rIBIS were described as mixed, ranging from frustrating and unproductive to satisfying and productive.

Discussion

Although gIBIS is a system for supporting group deliberation – an activity of argumentation – the computer does not play a participatory role in this process. As described in Section 2.3.4, this style of GDSS makes differing requirements for argument representation compared with those if computer participation was also required. In gIBIS, the structure of the deliberations are represented at a coarse level of granularity – for example, a single user’s argument is a primitive of the system with no further structure being required. This makes the system simple to use, but the lack of structure within arguments prohibits any analysis of the argument by the system. As a result, analysis of arguments (eg. identifying conflicts, locating other relevant information) is left to the users themselves.

Thus gIBIS illustrates an approach to argumentation where the computer is not an active participant. Its relevance is two-fold. First, it indicates that the computer can assist in group problem-solving through manipulation of arguments, suggested by its qualified application. Second, it provides a useful point of reference for this thesis, showing how a coarse representation of argument structure can be used in systems with a passive role. However, because of this, gIBIS was also unable to exploit the information stored in the arguments, limiting the assistance which it could provide.

2.4.3 NoteCards

NoteCards is also a hypertext ‘idea processing’ system for developing and relating ideas together on the screen. Unlike Euclid, it is intended as a general-purpose tool. Marshall has examined its specific application to the representation and organisation of arguments [Mar87]. Her representation of argument structure is of relevance to this thesis.

The central construct in the NoteCards system is a semantic network consisting of electronic notecards connected by typed links. Each notecard contains information as text, graphics etc. Each link designates a specific relationship between two notecards; the relationship may be user or system defined. Tools for constructing and browsing networks are provided.

Unlike the previous systems, Marshall also represented the internal structure of arguments. This was done using Toulmin’s model of argument structure, which we describe later in Chapter 3. Although Marshall’s interest was primarily in whether and how NoteCards could represent arguments, and thus the research was not pursued further, the fact that she found Toulmin’s model a suitable representational framework is the main point of relevance to this thesis.

2.4.4 Summary

We summarise the main points of conclusion:
A small number of hypertext systems have been applied to the task of argumentation.

For these systems, the context has been one of argument documentation, the system constraining the structuring of arguments but not participating in reasoning. giBIS shows how this can be used in the context of group decision-support.

This has placed different requirements on argument representation. In particular, a detailed structure of the content of primitive arguments have generally not been needed or used.

In NoteCards where more detail was provided, Toulmin’s model of arguments was used. This suggests the possible use of Toulmin’s model for argumentation where the computer participates.

Although these approaches can assist user(s) with structuring and recording opinions, the lack of a formal representation of argument content prohibits the computer constructing and checking arguments itself. Additionally, the lack of structure in the recorded arguments limits the system’s ability to compare arguments and retrieve information. A goal of this thesis work is to provide this additional functionality.

2.5 Evaluation Issues

2.5.1 Introduction

Before turning to the central part of the thesis, we finally review criteria and methods for evaluating decision-assisting systems. Our aim is to highlight the issues involved, providing a context for the evaluation of this thesis work in Chapter 5.

Evaluation of decision-aiding systems is particularly difficult. Measures of success are hard to define, and often the entire perceived benefit is intangible (eg. increased user effectiveness, a better decision-making process). It is also an issue which is often neglected, particularly in AI research (for example, of the 22 geological systems summarised in Section 2.2.2, only publications on Prospector included details of evaluation). For these reasons, it is valuable to provide some discussion of this issue here. Our main concern is to identify possible classes of evaluation criteria which can be used, and examine their strengths and weaknesses as indicators of success. Following this, we make some points of summary providing a guide for the evaluation of our particular application in Chapter 5.

2.5.2 System Accuracy

System accuracy – how often the system makes or recommends the correct decisions – is one criterion which has been used to assess decision-aiding systems. This is particularly common for decision-aiding expert systems: if these systems are meant to accurately model human reasoning, then their decisions or recommendations should at least be acceptably close to human performance.

This criterion provides one measure of the adequacy of the system’s encoded knowledge. In some domains, accuracy can be relatively easily measured (eg. sometimes in medical diagnosis, where a test set of patients with known disease status is available). However, there are also several problems with this measure:

1. Most seriously, it equates accuracy of advice with quality of support for the user. Other important aspects of decision support (eg. quality of explanations, usefulness of presented information, comprehensibility, impact on the user) are not taken into account. In particular, if the system is to support experts who can already solve the problem themselves,
2.5. EVALUATION ISSUES

strong emphasis on the system’s own solutions is less appropriate (Mittal and Dym describe encountering this problem when installing the MDX system [MD85]). Many expert systems have been shown to be accurate but failed to show evidence of usefully aiding experts. As Sutton notes, the computer’s accuracy may matter less than its credibility [Sut89].

2. This metric can only be applied to systems which offer decision advice directly (as opposed to performing data analysis subtasks, for example).

3. In many domains the correctness of the system’s (or user’s) answer is difficult or impossible to assess. We discuss this further in the context of user effectiveness below.

2.5.3 User Effectiveness

‘Improved user effectiveness’ has often been suggested as an overall goal for decision-aiding systems [O’K89]. However, ‘effectiveness’ itself cannot be measured directly; instead a more specific definition of the term is needed, suitable for the application domain of interest. Possible indirect measures of effectiveness include users’ speed, users’ accuracy, and company profitability.

In some applications, suitable indirect measures of effectiveness can be applied; for example, if the decision-aiding system is to help users perform a task faster, then this can often be tested easily. However, in many domains there are several difficulties which can be encountered:

1. Indirect measures of effectiveness may not be readily available. For example, in many domains where the system is to help users’ make better decisions, measurement of decision improvement may be difficult or impossible.

User accuracy is one such measure of decision improvement which can often be difficult to success (the problems are similar to assessing system accuracy). This is particularly true for tasks of probability estimation, especially where data is collected slowly. Risk assessment in oil exploration exemplifies this, where data points (exploration wells) are acquired infrequently at a cost of approximately five million pounds (our statistics in this respect are 1 of 6 drilled prospects finding oil (Section 5.2.1). It is difficult to draw significant conclusions from such low sample data.

2. It is often impossible to isolate the contribution to effectiveness caused by the decision-aiding system alone. A user’s effectiveness is also influenced by his or her experience, other personnel, quality of dynamics within the company, other software tools etc. Thus measured changes in effectiveness may not necessarily reflect on the decision-aiding system itself.

For some systems, their use is necessarily accompanied by a change in working practices. Again, care is needed as improved user performance may be attributable to those changes rather than use of the system itself. In his study of applied medical expert systems, for example, Sutton argues that a main cause of improved diagnostic accuracy was simply that users had been forced to use more structured data collection methods [Sut89], and that the diagnostic systems were themselves responsible for only a small part of the improvement.

3. Often metrics of effectiveness are long-term (eg. company profit), which may be impractical to measure or simply not available within the time-frame of the system’s use.

For some applications a formal cost-benefit analysis may be possible to quantify financial gain from an application [TF90]. These methods serve to translate measures of effectiveness into financial terms, useful for a company’s financial planning, but leave open the question of how the changed users’ performance can be assessed in the first place. We thus point to these
methods as mainly useful for post-evaluation analysis, but note that evaluation of the basic system’s performance must still be made.  

One way of measuring effectiveness is through the use of control group experiments. Here, performance with and without the system is assessed and compared, the control group providing some normative measure of performance against which to make the comparison. This helps to isolate the contribution which the decision-aiding system has made. However, there are problems also with this experimental method. Most severely, practical issues may prohibit or limit its application:

1. It may be infeasible to achieve sufficiently large groups to eliminate natural performance variation between users [BWB87].

2. It may be infeasible to perform a sufficiently long experiment to adequately measure a system’s impact (again, this depends on the application task). In Sharda et als’ review of twelve laboratory studies of DSS effectiveness, the majority of systems were evaluated after only one or two sessions with the system [SBM88].

3. Users’ behaviour in the laboratory setting may be unreflective of normal behaviour within real-world decision-making contexts. Thus laboratory trials are only indirect indicators of the system’s utility within those contexts.

### 2.5.4 User Satisfaction

A third criterion for measuring success of a decision-aiding system is user satisfaction. User satisfaction has a number of advantages as a metric: perceived benefit is often an indicator of actual benefit, and is also an indicator of the system’s suitability for integration into a decision-making context [Lan86]. A common method for evaluating user satisfaction is through the use of interviews or questionnaires, eg. employing attitude scales.

Despite this, there are also several problems associated with this criterion:

1. User satisfaction is closely related to the users’ expectations. High expectations may result in poor user satisfaction, regardless of the system’s value.

2. User satisfaction is strongly influenced by implementation and hardware details (eg. user interface, documentation, speed, robustness); the influence caused by the system’s underlying design principles are almost impossible to isolate.

3. Users may not be aware of all the impacts of the system; for example, users may make more consistent decisions using a DSS but not be directly aware of this change.

4. There may be social effects on users’ opinions, particularly (as with many systems) where users have been involved in the system’s development. Examples include: personal desire to see the system succeed (or fail) and personal relationships (eg. with other users with vested interests).

### 2.5.5 System Usage

Fourth, (non-mandatory) system usage can be used as a metric of a system’s utility in aiding decision-making. This criterion has several advantages. Most importantly, it shows that users perceive a benefit in using the system, a strong indicator that the assistance the system provides is of value. Additionally, it indirectly measures acceptance of the system by users, argued for by Lane as essential for achieving practical decision-support [Lan86]. However, again we raise some cautionary points:
1. High usage may not necessarily imply success if that usage is mandatory (e.g. manual methods removed), or is to use peripheral aspects of the system only (e.g. report generation).

2. Usage does not necessarily imply performance improvement. However, for many DSSs, usage implies provision of information to the user, and the system's usage implies the user considers that information worth obtaining. Additional metrics are needed to assess the quality and effects of that information.

3. Limited usage may not necessarily indicate failure as the system may only be expected to be used rarely (e.g. crisis management) or the payoff when it is used may be high.

4. Proper assessment of usage is a long-term issue; some early usage may be expected anyway eg. users initially trying the system. For new systems, long-term usage data may simply not be available.

5. As with user satisfaction, considerable social and other influences determine usage. Even if a system is not used at all, it does not necessarily imply failure of its underlying model. Other factors such as a poor system interface, perceived threat to jobs, the overhead of learning new working practices, and general mistrust of computers may be responsible, leading to the demise of a system which may be successful in a more favourable environment.

2.5.6 Information Analysis

So far we have discussed evaluation of a system's overall impact in supporting decisions. However, for many applications there are also specific claims which can be examined by direct analysis of data within the system. For example, we may wish to know how sensitive a system's recommendations are to its input data, and thus perform a sensitivity analysis (this was done for Prospector). Often a main claim of system effectiveness can be additionally supported by showing important sub-claims hold; conversely, if subgoals thought to be necessary for aiding decisions are not met, this also valuably contributes to the evaluation. In the context of evaluating decision-aiding systems, analysis of the value of information which the system provides has sometimes be used for evaluation [O’K89]. Again, however, indicators of this are often indirect and can be difficult to apply.

Encoding Subjective Probabilities

One class of analysis concerns the evaluation of encoded subjective probability estimates. We single this out for further discussion as our application task is of oil probability estimation, hence these forms of analysis are directly relevant.

In a review of subjective probability encoding [WB83], Wallsten and Budescu identify two aspects of probability estimation which can be assessed: reliability (absence of random error) and validity (its accuracy as a representation of opinion). In their detailed review [WB83], they identify four types of analysis which can be performed: reliability (measures of random variance by comparing probability estimates at different times), internal consistency (observation of whether expressed probabilities obey the laws of probability), calibration (correlation of estimates with independently obtainable measures), and inter-response correlation (comparison of estimates expressed in different ways). The applicability of these different measures is dependent on the domain, but shows that analysis of probability encoding can be performed as an indicator of a decision-assisting system's effectiveness. In our particular application domain, although these measures are difficult to apply, we are able to construct several comparative tests of the variability of users' encoded subjective probabilities. We elaborate further on these issues in our particular evaluation in Chapter 5.
2.5.7 Conclusions

We thus draw several conclusions concerning evaluation of decision-aiding systems. First, evaluation of these systems is particularly difficult as the benefits of such systems are often intangible. Second, appropriate criteria to use (in terms of both what defines success and what is feasible to measure) are both domain- and user-dependent, and thus must be chosen to suit the application context(s) of the system. Third, as criteria for measuring benefit are generally indirect, the use of several criteria is most appropriate for proper evaluation of these systems. Finally, we highlight sustained usability as one of the strongest indicators of having achieved success (although interpretation of non-usage is more difficult as discussed). However, we also emphasise that additional criteria must be used to confirm the nature of that success.

This review provides the context for evaluation of this thesis work in Chapter 5. As well as critiquing several classes of evaluation criteria, it highlights the need to design an evaluation which can operate within the constraints of the domain and application environment, to carefully clarify what constitutes success, and to use several evaluation methods if possible. These provide guidelines which we follow during our evaluation.

2.6 Summary

Several important areas have been reviewed: expert systems in the oil industry, decision-support systems, hypertext approaches to argumentation, and issues involved in evaluating these systems. The argumentation approach proposed in Chapter 1 develops dimensions of all of these areas, extending the use of knowledge-based techniques for representing and manipulating expertise in a domain. We now turn to the argumentation approach itself, first presenting a model to support user-computer argumentation, and subsequently describing and evaluating its application.
Chapter 3

A Model of Argumentation

3.1 Introduction

3.1.1 Goals

Our goal is to develop knowledge-based systems which can usefully participate in communal problem-solving alongside experts. To achieve this goal, we have developed a model of argumentation based on Toulmin’s work. This model consists of a specification of structures for representing arguments and general domain knowledge, constraints on their relationship, and a description of how they can be used for cooperative problem-solving. We present and justify this model here, and in Chapter 4 describe an implementation in a cooperative system for hydrocarbon exploration. The implementation illustrates one way in which the components of the model can be instantiated. In Chapter 5 we evaluate the extent to which the model has been successful in supporting practical argumentation.

3.1.2 Overview of the Chapter

Following some basic definitions, we first describe why Toulmin’s structure of arguments has been selected as a suitable framework for our model. We then describe how we have developed it for our own purposes. Following this, we discuss how the content of elements within an argument can be represented, and how the notion of consistency can be applied. Following this, we describe how our model of arguments can be used for practical argumentation, achieving several of our objectives described in Chapter 1. Finally, we briefly overview the relation of additional work from the literature to our model.

3.2 Basic Definitions

Some basic definitions which we use, partly following Toulmin [TRJ79], are as follows:

An argument, in the sense of a train of reasoning, is the sequence of interlinked claims and reasons that, between them, establish the content and force of the position for which a particular speaker is arguing.

Inference is the process by which arguments are constructed.

Reasoning is the activity of presenting the reasons in support of a claim, so as to show how those reasons succeed in giving strength to a claim.

Argumentation refers to the whole activity of making claims, challenging them, backing them up by producing reasons, criticising those reasons, rebutting those criticisms, and so on.
The field of discourse refers to the domain of argumentation eg. law, medicine, aesthetics.

A party refers to a participant in argumentation.

3.3 Models of Argument

3.3.1 Requirements

A model of argumentation suitable for our purposes should meet the following requirements:

- expression of arguments.
- accommodation of conflicting knowledge.
- support for comparison operations between arguments.
- support for justification of steps within an argument.
- allow for practical human-computer disputation.

We examine possible frameworks for argument representation and their suitability for meeting these requirements, and describe why Toulmin’s work has been selected as most appropriate.

3.3.2 Expert System Approaches

Much work in expert system research has concerned the issue of knowledge representation. However, the focus of this work is substantially different to our concerns in this thesis. In particular, expert system work has concentrated on issues of modelling expert reasoning and explanation. Argumentation, though, places additional requirements on the representation scheme as presented above, several of which are poorly supported by existing expert system approaches. In particular, the representations used generally:

- are not designed to accommodate differences of opinion internally. Instead, conflicts are resolved when the knowledge base is constructed.
- support only limited disputation with the user; typically the system can only ‘explain’ its reasoning, but not justify the validity of knowledge within its knowledge base.
- are not designed to respond to disagreement by the users. To do this would require knowledge represented in a way suitable for users to express their disagreements.

Our point is not that these representation schemes are inherently poor, but that they have been designed for a different purpose which limits their suitability for argumentation.

Substantial recent work in AI has sought to expand the capabilities of expert systems. The majority of this work has concerned knowledge representation for more effective problem-solving (eg. qualitative reasoning, causal modelling). However, as reviewed in Section 1.6.3, some work has addressed issues of argumentation including critiquing the user’s opinion, representing different opinions, responding to corrections, and improved explanation/justification facilities. We do not repeat this review here, but make the point that this work has addressed these requirements individually. None provides a representation which meets all our requirements above, and thus we still seek a suitable framework in which these contributions can be integrated.
3.3.3 Formal Logic

There has also been substantial work on arguments in logic, arising from philosophy (eg. [Rus67]) and in AI (eg. [GN87]). Here, the focus has been on systems for establishing the formal validity of a line of reasoning. However, while this work has been valuable for formal analysis of inference, our concern is primarily with argumentation as a social phenomenon. In the social context, reasoning can be inconsistent, imprecise, and ambiguous; the concept of a formally valid argument or belief rarely has a role [Tou58]. This makes formal logic an inappropriate model to meet our requirements.

3.3.4 Social Argumentation Models

There have been very few models of social argumentation proposed in the literature. We reviewed several applied in the context of hypertext systems in Section 2.4. For the systems Euclid, gIBIS and rIBIS, arguments were represented at a coarse level of detail, allowing the relation between arguments to be represented but providing no internal structure for them. Aside from these approaches, Toulmin’s model [Tou58] is the only other which has been proposed that also provides a detailed structure of the elements within primitive arguments. We detail this shortly. Because we wish the computer to participate in argumentation, such structure is necessary to allow inference to take place, focal points of conflict to be located, and fine points of comparison to be made. The structure afforded by Toulmin’s model allows these requirements to be met. In addition, Toulmin’s proposed structure supports justification of individual argument steps by the introduction of ‘backings’ as elements within the argument. Finally the model is simple, lending support for its comprehensibility – essential for human-computer disputation.

As a result, we adopt Toulmin’s model as an appropriate framework within which to develop our own model; it is the only one which is both designed to support argumentation and also provides sufficient argument structure to meet our requirements.

We also make the important point that the syntactic structure of Toulmin’s model is similar to that of rule-based systems (we elaborate on the equivalences shortly), although some components of his model have no equivalent. Thus we could have alternatively presented this work using the terminology of rule-based systems (suitably extended for Toulmin’s elements with no rule-based equivalent). However, because we wish to emphasise that the thrust of this work concerns argumentation rather than rule-based inference, we adopt the vocabulary which Toulmin has already provided.

Finally, we note that Toulmin’s work only partially specifies a computational model. Our contribution in this thesis is to develop Toulmin’s general framework further, providing a computational model supporting the argumentation functions identified in Chapter 1.

3.4 A Model of Argument Structure

3.4.1 Motivation

We wish to form a general model of arguments. However, some features of arguments depend on the particular field in which they are being used. Legal arguments, for example, may involve the use of fixed statutes, interpretation of terminology, and citing precedents; scientific arguments often involve mathematical techniques and statistical analyses of empirical evidence; and so on. We now describe Toulmin’s structure of arguments, and the elements within it which he identified as independent of the field of discourse (‘field-invariant’ p15 [Tou58]). Following this, we show how we have developed it to construct our own model of arguments. We discuss the representation of elements in this structure and how it can be employed for practical argumentation.
3.4.2 Toulmin’s Model

Elements of an Argument

Toulmin identifies six elements of an argument:

Claim: The conclusion reached.

Grounds: The facts upon which the argument is based. The grounds are the foundation of the argument.

Warrant: A statement of a general relationship between the grounds and the claim.

Backing: The knowledge that supports the warrant. The backing is sometimes referred to as the justification for a rule in the AI literature (eg. [WS89]), and describes why the warrant itself should be believed. (This is distinct from a typical expert system-like explanation of how the grounds of a warrant were satisfied).

Rebuttal: Exceptions that invalidate the claim. The rebuttal can attack any part of an argument (except itself).

Qualifier: The strength or weakness, conditions, and/or limitations with which a claim is advanced. Qualifiers are modal, applying to the entire argument.

The six elements together can be read as:

Grounds, so qualified claim, unless rebuttal, since warrant, on account of backing.

Each element of an argument can itself be the claim of another argument, and thus arguments can chain together. A medical example of how these elements can combine to construct an argument is shown in Figure 3.1.

Relationship with Rule-Based Systems

The components of a rule-based system can be interpreted using this structure:
Note that the backing is not normally represented in rule-based systems. However, backings play an important role in argumentation as participants may challenge the validity of the warrant ("why should I believe this rule is valid?"). Response to such a challenge is not normally possible by most rule-based systems, and thus the representation and use of backings is an important distinguishing feature of the model with which we are working.

Rebuttals are also normally omitted in rule-based systems (but see [Win86] for an example of their use in machine learning). Instead, statements of rebuttal can be included in the conditions of rules (e.g., a rebuttal R of the rule if A then B can be accounted for by re-expressing the rule if A and not R then B).

Expression of Arguments

There may be several ways of expressing an argument within this structure. For example, the choice of which facts are stated in the warrant and which contribute to the backing depend on how a party decides to express the argument. As illustration, an argument could be phrased as follows:

the agreement to take cooking in turns being presented alternatively as part of the warrant or backing.
3.4.3 Notation

Toulmin's Structure

We introduce a notation for Toulmin's argument structure, following suggestions by Niblett [Nib91]. Warrants express a relationship between grounds and a claim, which we denote:

\[ g_1 \land \ldots \land g_n \rightarrow_s C \]

where \( g_i \) are the grounds, \( C \) is the claim and \( s \) is the strength of the warrant. The warrant's strength represents the degree of influence which the grounds have on the claim. (A precise definition of this is introduced in our implementation). The qualifier \( q \) of an argument's claim is related to the strength \( s \) of the warrant and the qualifiers of the grounds. The distinction between \( s \) and \( q \) is important to note, and is similar to the distinction between a rule's and a data item's certainty in a rule-based system.

In a similar way to rule-based systems, it is more convenient to express warrants as general statements, using universally quantified variables, instances of which appear within an argument.

Our Model of Argument Structure

We develop Toulmin's model in four ways. First, we explicitly separate a warrant into two components, namely its strength (introduced in the previous section), and its skeleton. We define a skeleton warrant to be a warrant with the strength omitted. The skeleton warrant can be regarded as a statement of potential relevance, that \( g_i \) are potentially relevant to \( C \). Such a statement of relevance is uncontroversial, as an expert can in a concrete argument assign a value of \( s \) which means that the \( g_i \) make no contribution to our belief in \( C \).

Second, we introduce the concept of a model. A model is a set of skeleton warrants \( W \), an assignment of a strength \( s_w \) to each \( w \in W \), and a backing \( b_w \) supporting the warrant \( (w + s_w) \). We sometimes refer to this as a 'model of opinion'. We will introduce constraints on these models, described later in Section 3.4.5.

Third, we specify a structure for combining Toulmin's structures into an overall argument. An argument is represented as a tree, where each node represents a claim or conjunct of claims, also serving as the grounds supporting a claim in its patent node. Leaf nodes are claims whose supporting warrants and grounds are not represented, and which derive their authority from a database or the user. Each arc represents a warrant \( w \), connecting grounds \( (\land g_i) \) of the warrant to its claim \( C \). Arcs may also have an associated backing \( b_w \) for the warrant they represent. We define the structure of the tree as this tree with the strengths, qualifiers and backings removed.

Fourth, we do not include rebuttals in our model of arguments. This simplifies the argument structure without serious loss of expressive power, as statements of rebuttal can be included within other argument elements (either as extra grounds of warrants or in the warrants' backings).

3.4.4 The Role of Models of Opinions

A model of opinion can be used to represent a particular party's beliefs about the relationship between grounds and claims, with the backings representing his or her justification for those beliefs.

An argumentation system must keep track of the different opinions of different parties with which it interacts. To do this, each party's opinion is represented as a different model.

3.4.5 Constraining Models of Opinions

Argumentation often involves comparing different models of opinion, locating similarities and differences. To make meaningful comparisons between parts of different models, those parts must
have equivalent roles in some way. To make such comparisons mechanically, that equivalence must be identifiable from the syntactic structure of the parts. If each party were to use their own language and terminology for warrants, then such comparisons would be difficult.

To reduce this problem, we add the constraint to our argumentation model that all models of opinion share the same set of skeleton warrants. In the terminology of rule-based systems, users must share a single, common set of rules (skeleton warrants) although they can customise the attached certainty factors (warrant strengths). This is illustrated schematically in Figure 3.2.

This requirement is important, as it permits inter-model comparisons to be easily made. The adequacy of requiring users to share the same set of skeleton warrants is discussed further in Chapter 6.

3.4.6 Language Issues

Requiring parties to express opinions in a common language (defined by the skeleton warrant set) does not imply that all parties naturally use a common language when reasoning about a problem on their own. It does however require users to be able to translate their arguments to and from this language if they are to interact with a system using a single language for arguments. We illustrate this schematically as shown in Figure 3.3.

There are additional research issues which arise here of how to translate vocabulary, including words whose meanings are not precisely defined (‘open textured’ [Ris85]). As we require this translation to occur outside our argumentation framework, we do not pursue these issues, but note they can be significant in argumentation (eg. in the field of law). These issues are discussed further in [RVA84, vG83].

3.5 Formalising Argument Content

3.5.1 Introduction

Section 3.4 described the structure of arguments, ie. the domain-independent elements of an argument and their relationship. This section examines the representation of the content of
the argument elements. A formal representation is required for the computer to be able to mechanically construct and manipulate arguments.

3.5.2 Grounds, Warrants and Claims

In automated reasoning, mechanised inference is performed by manipulating the syntactic structure of represented knowledge. Many formalisms exist for representing knowledge and performing sound mechanical inference in this way. We can adopt one of these to represent grounds, warrants, claims and qualifiers. This is important as it allows the computer to mechanically construct arguments and thus play an active role in argumentation.

3.5.3 Backings

Backings and Warrants

Toulmin’s structure of arguments extends the structure of knowledge in an expert system to include backings. Backings represent support for warrants (‘justifications for rules’). Including backings in a representation of knowledge presents difficulties because backings do not always logically imply the warrants. In this sense, Toulmin’s arguments are partly *semantic* since the soundness of an argument cannot be judged purely on the basis of its syntax. Toulmin summarises this:

> “Once we bring into the open the backing on which (in the last resort) the soundness of our arguments depends, the suggestion that validity is to be explained in terms of ‘formal properties’, in any geometrical sense, loses its plausibility.” p120 [Tou58]

Whereas grounds and warrants can be viewed as a particular model of belief with which we can perform mechanised inference, backings can be viewed as representing evidence of *why* we should believe that model is correct in the first place.

In some special cases, the backing might logically imply the warrant. Toulmin refers to these cases as *analytic arguments*. Otherwise, Toulmin refers them as *substantial* arguments.

Types of Backings

Although backings do not always logically imply warrants, evidence that they provide support for warrants can often be found. These types of evidence include:

**Statistical:** Sometimes warrants are backed by statistical evidence of their validity. Given a set of examples, warrants could be automatically generated from the data, backed as being from statistical evidence in that data set. This is in fact the task that inductive rule learning systems perform [CN89, Qui83].

**Precedent:** Warrants can obtain some support even from a single example of their use, especially when that example is recognised as representative in some way of a class of examples.

Many warrants could be generated from a precedent example, and thus extra domain knowledge is needed to identify those where the example offers more support than simply statistics over a sample size of one. Some work in ‘explanation-based generalisation’ in machine learning can be regarded as performing this task [MKKC86].

Alternatively, *given* an existing warrant, the location of precedents can play an important role in increasing support for it (eg. in law, where precedents strongly determine the interpretation of terminology).
3.6 Consistency

A key principle for argumentation is that of consistency, namely that similar decisions should be made in similar situations. We define inconsistency within our model as follows:

- Two warrants are inconsistent if they share the same skeleton but have different strengths.
- Two arguments are inconsistent if there exist two warrants, one in each argument, which are inconsistent.
- Two models are inconsistent if there exist two warrants, one in each model, which are inconsistent.

These definitions of inconsistency are used when arguing with the user. They constitute argumentation knowledge ('how to argue') rather than domain knowledge. We use the term 'conflict' and 'inconsistency' synonymously.

3.7 Summary of the Argument Model

Our model of arguments can be summarised as follows:

1. The structure of an argument is expressed using Toulmin’s elements (grounds, claims, etc).
2. A model of opinion consists of a set of warrants and their backings.
3. Within a particular domain, different models of opinions must use the same (skeleton) warrants; only the warrants’ strengths can be customised.
4. Arguments are combined into a tree structure (Section 3.4.3).
5. The relationship between grounds, warrants, claims and qualifiers can be expressed formally. Thus sound, mechanised inference is possible.
6. Backings expressing domain knowledge are represented as uninterpreted strings of text from the user.
<table>
<thead>
<tr>
<th>SWarrant</th>
<th>Strength</th>
<th>Backing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>v. weak</td>
<td>“Dips almost always effective here, due to thick seal presence”</td>
</tr>
<tr>
<td>2</td>
<td>medium</td>
<td>“Stratigraphic seals, while risky, often seal up with calcite precipitate in this area”</td>
</tr>
<tr>
<td>3</td>
<td>strong</td>
<td>“From theory, a late fault cutting the oil reservoir may cause it to ‘leak’ along the fault. This has been shown to be a particularly serious problem in this area (eg. see 23rd round drill results)”</td>
</tr>
<tr>
<td>4</td>
<td>certain</td>
<td>(definitional)</td>
</tr>
<tr>
<td>5</td>
<td>certain</td>
<td>(definitional)</td>
</tr>
</tbody>
</table>

Figure 3.4: A model of opinion.

3.8 An Example of our Argument Model

This section presents a small, simplified example, adapted from the implementation described in Chapter 4. The domain is hydrocarbon exploration (we describe this as ‘oil exploration’ for conciseness). The claim of interest is whether a particular oil trap (a dome-shaped layer of impermeable rock) is effective at containing oil in porous rock layers below. A detailed description of a full implementation is presented in Chapter 4.

3.8.1 A Common Set of Skeleton Warrants

Different users share the same set of skeleton warrants, such as shown in Table 3.1.

3.8.2 A Model of Opinion

A model consists of the skeleton warrants \( W \) and an assignment of a strength and backing to each warrant \( w \in W \). A verbal expression of strength is used here, although the implementation uses a numerical representation. One geologist’s model might be represented as shown in Figure 3.4.

3.8.3 Problem Description

A description of the problem to work on (the focus of the argument) is required. Here, we are interested in the effectiveness of a prospect’s trap. This prospect is described using attribute-value pairs:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>( e/2-1b )</td>
</tr>
<tr>
<td>Trap in direction 1</td>
<td>dip</td>
</tr>
<tr>
<td>Trap in direction 2</td>
<td>major fault</td>
</tr>
<tr>
<td>Prospect seal</td>
<td>Madeline</td>
</tr>
<tr>
<td>Most recent faulting</td>
<td>cretaceous</td>
</tr>
</tbody>
</table>

3.8.4 Arguments

A model is used to construct an argument for a qualified claim. The argument produced by applying the model of opinion to this prospect is sketched in Figure 3.5. Note that warrant 2 (Table 3.1) does not occur in the argument as its grounds are false. This figure shows the strengths of warrants 1 and 3 (‘very weak’ and ‘strong’ implication that trap is ineffective
skelwarrant 1 ::
if trap in direction N of prospect is dip
then trap is ineffective.

skelwarrant 1 :: If the oil trap is a ‘dip’ structure, then the effectiveness of the trap
might be affected.
Note that the warrant expresses potential relevance, rather than logical implication,
as the strength of the warrant is missing. Given the grounds, the claim might be
qualified by ‘probably’, ‘it is unlikely that’ etc.

skelwarrant 2 ::
if trap in direction N of prospect is stratigraphic
then trap is ineffective.

skelwarrant 3 ::
if trap of prospect is faulted
and ‘most recent faulting’ of prospect is Rock1
and ‘seal unit’ of prospect is Rock2
and Rock1 is_younger_than Rock2
then trap is ineffective.

skelwarrant 3 :: faulting after the oil reservoir has formed may rupture the reservoir
and cause the oil to leak away.

skelwarrant 4 ::
if trap in direction N of prospect is F
and F isa fault
then trap of prospect is faulted.

skelwarrant 5 ::
if age of Rock1 is AgeRock1
and age of Rock2 is AgeRock2
and AgeRock1 < AgeRock2
then Rock1 is_younger_than Rock2.

skelwarrants 4 and 5 :: these are definitional warrants, and thus would have maxi-
mum strength, equivalent to logical implication.

Table 3.1: A set of skeleton warrants.
respectively) combining into an overall qualifier ‘likely’. We do not give general combination rules here, as this example is illustrative only. In the implementation, though, combination rules for strengths and qualifiers are specified (a numeric representation is used).

### 3.9 Practical Argumentation using the Model

In Section 1.8 we summarised several functions which we wish our argumentation model to support. Here, we highlight the main characteristics of our model and discuss their ability to support these functions. We also describe other benefits which they provide.

#### 3.9.1 Modelling and Comparing Different Opinions

Different opinions are represented by different models. Although we keep models separate, the models all are constructed with the same set of skeleton warrants. This allows comparisons between models and arguments to be made easily by virtue of this commonality. This would be difficult if each model used its own unique warrants and terminology for expressing opinions.

#### 3.9.2 Open-ended Domain Backings

In our model, backings from the user are represented as uninterpreted text (Section 3.5.3). This provides an ‘open-ended’ knowledge representation; there are no bounds placed on the knowledge that can be expressed. This is essential in domains where a clear bound cannot be distinguished around the knowledge relevant to the problem at hand.

Although the semantics of text backings are not accessible to the system, the backings can still be manipulated and replayed to the user in a useful fashion by virtue of knowing their role within an argument. For example, given a backing $B$, the system knows that it justifies some warrant $W$, that it explains why $W$ was preferred to some conflicting warrant $W'$, etc. The
effectiveness of this technique for manipulating informal text representations has already been
demonstrated by several hypertext systems which are summarised in Section 3.10.2.

3.9.3 Justifications and Case-Based Reasoning

An argumentation system must be able to generate justifications for its decisions in order to
respond to challenges from the user. There are several types of justification that can used. (As
mentioned in Section 1.6.3, Wick and Slagle identify seventeen types, eg. statistical evidence,
analogy, definition.) However, the most important of these supported by our model of arguments
is the use of precedents. This involves the location of previous cases where a judgement similar
to the current one was made, and then presenting that case to the user. Precedents provide
support for a decision by virtue of the principle of consistency, namely that a rational agent will
make similar decisions in similar situations.

The storage and recall of precedents is a form of case-based reasoning. Our model supports
this activity in several ways:

1. A case contains not only a problem description (the grounds) but also a representation of
reasoning about that description (the argument). The inclusion of the argument in a case
constitutes an explicit representation of the relationships between grounds and claims.
This is significant: it allows us to distinguish between relevant and incidental features
relating to a decision, and hence to know which case features to search for when looking
for precedents. We are thus using knowledge-based rather than statistical techniques (eg.
[Aha90]) to identify feature relevance in precedents.

2. Warrants provide a natural way of expressing the components in a chain of reasoning
and how they combine to conclude a claim. Because our model constrains all arguments
to be constructed from the same set of skeleton warrants, we can directly identify and
compare equivalent components of reasoning in the current and precedent cases. This is
very important; if the user challenges a warrant in the current case, we are able to identify
its equivalent in precedents by locating the same skeleton warrant within it. This would
not be possible if our argumentation model did not impose this common skeleton warrant
set constraint.

3. The backings for warrants play a key role in case-based justification. Having retrieved
a precedent, it is the backing which provides the justification about why the precedent
decision was made. An important role of the argument structure is thus to organise
backings.

3.9.4 Conflict Resolution

Although our model supports the location of conflicts between different arguments, we have not
described any method for the computer to resolve those conflicts itself. Conflict resolution is a
complex task, requiring substantial domain knowledge as we describe shortly (Section 3.10.4).
As a result, we prefer to let the users themselves ultimately arbitrate among conflicts which
may exist. This does not detract from our objective of decision support for users; the model’s
importance is in its support for location rather than resolution of conflict.

3.9.5 Knowledge Acquisition and Maintenance

For a system to respond to corrections from the user, the user must be able to express those
corrections. Ideally, we would like users to express their corrections both easily and formally.
However, there is a trade-off between the cost (in terms of difficulty) to the user in expressing
corrections in a formal way and the benefits provided by a formal expression that the system
can then manipulate.
In our model of argumentation, the user changes a model by altering the warrant strengths. This is easily effected if strengths are represented numerically. The user also justifies this change by providing a backing represented as uninterpreted text. This corresponds to a particular point in this trade-off where the formal expression of a correction is tightly constrained (users cannot modify skeleton warrants themselves, only warrant strengths). This helps our model meet the requirement that it is practical, but at a cost in flexibility. We discuss its adequacy in Chapters 5 and 6. Because of these constraints, we term this knowledge maintenance rather than knowledge acquisition.

3.10 Related Work

We now relate our work to other approaches in the literature, some of which were introduced in Section 1.6.3.

3.10.1 Modelling and Comparing Different Opinions

Only a few researchers have explored the theme of representing different opinions as separate knowledge-bases within the same system. This is easily effected by simply labelling represented knowledge with the owner's name (e.g. Negoplan [MSK+89], LeClair's system [LeC85] and Young's system [You87]). However, the more difficult task is to compare knowledge from separately encoded knowledge bases, made difficult if differences in language and rule structure exist. As described earlier, our argumentation model overcomes this problem by constraining users to use the same skeleton warrant set.

3.10.2 Using Uninterpreted Text for Backings

Our model uses backings represented as uninterpreted text. This removes any constraints on the types of backings which can be expressed, but also prohibits computer understanding of their semantics, constraining the extent to which an argumentative dialogue can be sustained. We do, however, know their role within the argument (i.e. which warrants they justify), represented by their position in the argument structure.

In Section 2.4 we reviewed several hypertext approaches to argumentation. The key hypothesis underlying these systems is that arguments can be usefully manipulated by virtue of knowledge of their argument structure alone. While we additionally include a formal representation of argument content (the warrants), we follow this hypothesis by using backings represented as uninterpreted text.

As described in the review, these systems differ from our approach primarily as they do not have any active reasoning component. In contrast, our model includes machinery for reasoning. Additionally, although a backing's strength of support for a warrant cannot be computed from uninterpreted text, we can find this strength by simply asking the user to make this interpretation. This provides the key bridge between informal backings and their formal summary, thus allowing their use in automated reasoning.

3.10.3 Justifications and Case-Based Reasoning

Our model supports the use of previous precedent cases for justifying (or challenging) conclusions. Cases include not just data (the grounds) and a claim, but also a representation of the reasoning about how the claim was concluded from the grounds (the argument).

Aha describes case-based reasoning systems as comprising (i) a similarity assessment component, (ii) a performance component and (iii) a learning component [AKA91]. For similarity assessment, there are several case-based reasoning systems which also use domain knowledge to identify important features (e.g. Hypo [RVA84], Protos [BPM89] and by Cain et al., [CPS91]).
Our model is thus an example within this group. Second, our model retrieves cases in order to warn users of possible inconsistencies in their reasoning. A similar approach was explicitly used in Ladies [DBG91], and is also the implicit goal of many case-based systems solely performing case retrieval (e.g. [Kol88]). Our argumentation model, however, is distinguished by presenting not only the previous, relevant judgement but also the justification why that judgement was made (the backing). The backing is, in fact, the target of the case retrieval – the argument structure can be seen as a structure with which to organise backings. The inclusion of this justification knowledge is novel for case-based reasoning.

Finally, our model assumes cases will be stored in a simple database; this contrasts with case-based approaches which store cases in a more complex, organised structure (e.g. Judge [Bai86], Unimem [Leb86] and Cyrus [Kol83]). We thus assume standard database mechanisms will be adequate for search and retrieval.

3.10.4 Conflict Resolution

Ideally, an argumentation system should be able to resolve conflicts which it locates in its knowledge. We now summarise approaches to this task and the difficulties encountered with them. Most importantly, these approaches suggest that conflict resolution requires substantial domain-specific knowledge, and that general, domain-independent strategies are of limited use. For this reason, our model relies on the user to ultimately adjudicate among conflicts between arguments that the system may locate.

Consensus Approaches

One class of approach to conflict resolution is to combine conflicting evidence or conclusions together using analytic methods. For example, Bayesian techniques combine evidence relating to a conclusion together [Che88], and game-theoretic methods select the best course of action from alternatives given quantified reliabilities, costs and payoffs [Ros85, vM44]. This approach can be taken in domains where probabilistic or decision-theoretic models can be precisely specified. However, in many domains this is not possible, and thus these methods are of limited utility with our model.

Adversarial Approaches

In contrast, we define adversarial approaches as those where a single opinion or solution dominates (rather than taking some form of average of them). Truth maintenance techniques are examples of this approach. Given a set of (possibly conflicting) statements, a truth maintenance system searches for a consistent subset of them [Doy79, deK84] – equivalent to retracting some of the conflicting statements. These systems, though, used little or no knowledge about which statements should be blamed in the case of a conflict. Truth maintenance systems offer useful machinery for exploring the consequences of different assumptions, which could be used in an implementation of our argumentation model. However, they would require more knowledge to resolve conflicts in a less arbitrary way.

Some authors have looked for domain-independent principles for deciding which statement should dominate given a conflict (e.g. [Lou89, Sim89]). The only major forthcoming principle was that of specificity: if \( A_1 \Rightarrow B \), and \( A_2 \Rightarrow \neg B \) and \( A_2 \) is a specialisation of \( A_1 \), then the second rule should dominate. This principle is based on a probabilistic interpretation of the implication relationship. It is unable to resolve conflicts when a specificity relationship does not hold, and thus would not be generally useful for resolving conflicts within our argumentation model.
Knowledge-Based Approaches

Konolige, in an analysis of conflict resolution, concludes that:

“general domain-independent principles [for adjudicating among conflict] will be very weak, and that information from the semantics of the domains will be the most important way of deciding among competing arguments” [Kon88 p381]

Konolige thus recognises that domain knowledge (eg. that contained in backings) is needed to resolve conflicts. In his system ‘ArgH’ [Kon88], he reflects this by labelling each inference rule as describing either an event or persistence of facts. Conflicts are resolved by event rules dominating persistence rules. This resolution technique is thus a first attempt at formulating general conflict resolution principles using domain-specific information. However, such principles are difficult to formulate, and even this simple principle is sometimes violated as he himself illustrates.

Cohen has similarly strongly advocated the use of domain knowledge to reason about uncertainty and conflict [Coh85]. He uses the term endorsements to describe elements of meta-knowledge supporting facts and rules within a system, similar to Toulin’s backings. Despite his persuasive philosophy, however, Cohen had substantial difficulties in developing a computational model of this theory. Essential functions such as the ranking of endorsements were not fully implemented and left as a “very difficult” open question. In his implementation, simple labels (one of model-based, causal or correlational) were used to represent endorsements, resulting in a conflict resolution strategy of limited discriminatory power.

Thus, although domain-specific knowledge is recognised as being important for conflict resolution, progress on formally representing and reasoning with it has been limited. As a result, we use an approach primarily designed to identify rather than resolve conflict.

3.10.5 Knowledge Maintenance

The main feature of the knowledge maintenance technique supported by our model is its simplicity: users express their knowledge as uninterpreted text and modify the corresponding warrant strength accordingly. This simplicity distinguishes it from other knowledge acquisition systems, described earlier in Section 1.6.3. With Protos [BPM89], for example, the user expresses his or her opinions in a formal, Lisp-like syntax, reported to require either programming skills or assistance for its use [Sha91]. This, of course does not detract from Protos’s value as a knowledge acquisition tool; however it does make its knowledge acquisition method inappropriate to meet our goal of usability by unsupervised, non-computer-skilled users.

Srinivasan et al’s ripple-down rules methodology [SCM+91] is an alternative whereby users select corrections to a rule from a list of possible corrections. However, this requires an adequate space of possible rules to be identified, thus constraining the corrections the users can express. This technique has been successful in the medical domain of chemical pathology interpretation, where an adequate rule space could be formulated. However, in general, this constraint may be more difficult to meet.

Finally, we point out that our model’s constraints on knowledge maintenance are not imposed solely to support its use by non computer skilled users, but also to support the other operations of argument retrieval and comparison as described in Sections 3.9.1 and 3.9.3. Thus there is an additional complication if we wished to apply these other knowledge acquisition techniques, as we wish to preserve the model’s support for these other functions also.

3.10.6 The Theory of Determinations

A skeleton warrant expresses an unspecified relationship between its grounds and claim; in other words, the grounds are relevant to the determination of the claim. This is closely related
to Russell’s representation of determinations [DR87] and work on representing relevance in AI [SG87]. A determination is a predicate schema which makes a logical statement of relevance between predicates. For example, the statement “a person’s nationality determines whether or not he or she needs a visa” could be represented:

\[
nationality(Person, Nat) \geq I_1 \text{ needs\_visa(Person)}
\]  

(3.1)

where \( \geq \) is the determination relation, \( Person \) and \( Nat \) are variables and \( I_1 \) is a polar variable, denoting that the truth of its following expression \( (\text{needs\_visa(Person)}) \) is not being specified. Equation 3.1 is thus analogous to a skeleton warrant and the polar variable \( I_1 \) analogous to the warrant’s unspecified strength. Unlike warrant strengths, however, polar variables are not directly instantiated by the user; instead, an example of the determination relationship reduces it to a logical implication. For example, given \( \text{nationality(marianne, swiss)} \) and \( \neg \text{needs\_visa(marianne)} \), Equation 3.1 then implies that:

\[
nationality(Person, swiss) \Rightarrow \neg \text{needs\_visa(Person)}
\]  

(3.2)

The formal machinery for this and its use is described in [DR87, Rus88a, Rus88b].

We make several points of comparison. First, determinations and skeleton warrants are both representations of relevance. Second, determinations represent a logical type of uncertainty (“I don’t know whether this relationship holds or not”) contrasting with a skeleton warrant’s quantified degree of uncertainty (“I don’t know how strong this relationship is”). Third, a determination’s determinant (left-hand side) expresses all that is needed to find the truth of its resultant (right-hand side). It thus assumes containment of knowledge, as discussed in Section 1.4.3. This contrasts with a skeleton warrant, which represents only one possible influence on a claim. Because determinations assume containment and do not represent degrees of uncertainty, their use for argumentation is limited. The possibilities for and limitations of determinations for argumentation are discussed further by Clark [Cla88b, Cla88c].

### 3.11 Summary

We have presented a model of argumentation, specifying structures for representing knowledge, how they relate and how they can be used for cooperative problem-solving. We now describe and evaluate one way in which this model can be applied.
Chapter 4

The Model Applied to Oil Prospect Assessment

4.1 Introduction

An important goal of this research is to develop an implementation that is both practical and useful. By practical, we mean that the system is usable by non-computer-skilled experts outside the laboratory environment. By useful, we mean that it can be applied to real-world problems, and provide sufficient benefits to users to merit the time required to run the system. This goal places considerable constraint on the implementation’s design. As a result, details of the implementation should not be viewed as arbitrary choices simply to illustrate the argumentation model but as equally important to the overall goal of constructing a practical argumentation system.

The model of arguments presented in Chapter 3 was chosen to address some of these practicality constraints. We now describe an application, named Optimist, for assisting geologists in hydrocarbon exploration in the offshore subsurface environment. We show how the model has been instantiated and applied to this particularly difficult task. An evaluation of the application’s success in achieving our goals is presented in Chapter 5.

Although hydrocarbon exploration involves searching for both oil and gas, this chapter’s title and contents are presented in terms of oil exploration only for conciseness.

4.2 The Problem Domain

4.2.1 The General Context of Oil Exploration

Hydrocarbon exploration is an important task, involving many people and much resources. The financial costs and potential rewards are high. In the North Sea, for example, wells cost around four to eight million pounds to drill and only about one in four will strike oil.

The complete process from exploration to oil recovery can be considered in two phases. First, potential locations of oil (‘prospects’) are identified, and second a decision is made about whether it is cost-effective to drill an exploratory well. If a decision to drill is taken and sufficient oil is found, further wells may be drilled to recover the oil on a commercial basis.

Estimating the cost-effectiveness of sinking a well in a prospect involves geological and financial considerations. Four key questions that must be answered are:

1. What is the probability of finding oil?
2. If there is oil, how much will there be?
3. How much will a well cost to drill?
4. How much can the oil be sold for?

These questions are difficult to answer. In particular, the geological question of the probability of finding oil and the financial question of predicting the oil's future commercial value are fraught with problems. The continuous changes in the oil price over the past two decades typifies the sort of unpredictabilities inherent in this crucial cost-benefit calculation.

The task considered here addresses only the first of these questions, namely what is the probability of finding oil in this prospect? Answering this question is called **prospect risk appraisal**, or simply `prospect appraisal`. This is the problem which our cooperative expert system is to assist with.

### 4.2.2 Finding Oil in a Prospect

Before describing Optimist's design and application, some background to the geological problem should be given. We briefly describe the process by which oil prospects form, and how geologists assess the risk associated with this process.

#### The Geological Process

Several conditions must hold for a prospect to contain oil. First, there must be a geological structure in which oil can be **trapped**. In subsea conditions oil, being lighter than water, rises, and thus this trapping structure is a dome-shaped impermeable layer of rock lying above porous rock (e.g. sandstone). The porous rock is called the oil **reservoir**. It is here that the oil accumulates.

In addition, some **source** of oil is required to produce the oil in the first place. Oil is formed from ancient organic matter that has been trapped, heated and compressed over millions of years below ground. After sufficient heating and compression the source is said to be **mature** and produces oil, which will seep or `migrate` away from the source along available paths of porous rock and faults. To fill the prospect, a path must exist by which the oil can migrate from the source to the reservoir, i.e. there must be **communication** between the reservoir and source.

A schematic, simplified diagram of the way an oil reservoir forms is shown in Figure 4.1. Here, the oil has formed in the source and migrated upwards along a porous bed of rock, eventually reaching a point where it is trapped by a geological fault and a sloping bed of impermeable rock.

#### Assessing the Probability of Oil

The various requirements for the formation of an oil field can be summarised as six loosely independent conditions. Thus the probability of finding oil can be considered as the product of the probabilities of each one of these conditions being true. Each condition involves a different method of assessment:

1. **Trap presence** is assessed mainly from detailed examination of seismic sections over the area of interest, requiring identification of a geological structure which can potentially seal oil in. As seismic sections display the time taken for sound to bounce back from rock layers, they provide only an indirect measurement of the rock depths and the time-to-depth conversion can contribute a source of risk.

2. **Trap effectiveness** is assessed using knowledge of geological structures in the area. In particular, traps formed by faults or discontinuities in rocks can often 'leak', the risk being conditional on the type of fault or discontinuities involved. Thus the quality of the trap's seal (its impermeable boundary) is a fundamental consideration. Knowledge of the success of previously drilled, similar prospects is vital for this assessment.
3. **Reservoir presence** is often inferred from the known presence of reservoir rock in nearby drilled wells, or sometimes is even visible on the seismic. For oil, there must be at least some reservoir present (described as adequate ‘gross thickness’ in the implementation). Additionally, the composition of the reservoir rock must contain a sufficient component (‘net thickness’) of sufficiently porous rock (‘porosity’) for accumulating hydrocarbons, again inferred from data from nearby wells.

4. **Reservoir communication** with the source is difficult to assess, requiring consideration of the migration pathway between the oil source and reservoir. Again, nearby wells produce the main evidence. Even if a nearby well is dry, it may contain evidence of oil having reached it but then leaking away (‘oil shows’). This improves the likelihood of finding oil at the current prospect.

5. **Source presence** can be inferred from the known geology of the region and discoveries of oil in existing wells.

6. **Source maturity** can be inferred from geophysical studies of the time, depth and temperatures of the oil source since its formation, or can again be inferred from known discoveries of oil. In some cases source presence and maturity can be assessed directly if an existing well penetrates the hypothesised source.

These six different conditions, and the three sub-conditions for reservoir presence (adequate gross/net/porosity), can be considered loosely independent. As a result, the appraisal task can be broken down into six sub-problems, the probability of finding oil being the product of the probability of each condition occurring. We sometimes refer to these six conditions as risk components.
4.3 Applications of the Model

Our model of argumentation can be applied in a number of ways, one of which (namely Optimist) we describe and evaluate in the next two Chapters. Optimist applies the model in a straightforward way: for example, user-supplied grounds are not qualified and warrants with modifiable strengths do not chain. The application can thus be seen as a specialisation of the general model we have presented.

We emphasise that the following describes only one way in which the model can be applied. For example, other alternative applications may allow warrants with modifiable strengths to chain, or grounds to be qualified. There are thus additional issues concerning the generality of the model which are raised. To fully discuss these issues, it is useful to first illustrate how the model can be applied and then discuss alternatives; this allows the specific application to act as a reference point against which alternative applications can be discussed. Because of this, we defer discussion of these issues to Chapter 6. We note here that Optimist is only one instance of how the model can be applied.

4.4 Overview of the Implementation

4.4.1 Description

In Optimist, each appraisal of a prospect is represented as an argument whose claim is that oil is present and whose qualification is the probability that the claim is true. An argument is constructed from a model of opinion, comprising a set of skeleton warrants and their associated strengths and backings. Argumentation between Optimist and the user is based on this argument. During the dialogue, the user may query some of the reasoning and perhaps alter the argument on the basis of extra knowledge that the system does not have. Optimist will argue its case on the basis of consistency with previous decisions which have been made, alerting the user to relevant information which he or she may not know. When the user is satisfied with the argument and its conclusion, it is stored along with any changes made to the model.

4.4.2 Chapter Overview

The implementation is described in two parts, presenting:

1. The data structures used to represent warrants, arguments and models of opinion. These are described in Sections 4.6, 4.7 and 4.8.

2. The procedures describing how arguments are constructed and how argumentation occurs. If the reader desires a quick preview of this, he or she is referred to the flow chart in Figure 4.6 on page 69.

Following this, an extended example illustrates Optimist in use.

To provide the context for describing the data representations in Optimist, it is useful to summarise the three databases which the system uses:

Data, for storing descriptions of the prospects and other geological objects.

Models, representing general, personalised domain knowledge. Each model is a set of warrants.

Arguments, summarising how conclusions were reached using the data, a model and additional information from the user.

This is illustrated in a sketch of the system's architecture, shown in Figure 4.2.

To assist the user, a summary of the terminology introduced in Chapter 3 is given in Table 4.1.
<table>
<thead>
<tr>
<th>Data:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Grounds $g_i$</td>
<td>Known data about a prospect or other geological object.</td>
</tr>
<tr>
<td>Qualifier $q$</td>
<td>Expresses the degree to which a ground or claim is believed.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Models:</td>
<td></td>
</tr>
<tr>
<td>Model</td>
<td>A set of warrants, with personalised strengths and backings.</td>
</tr>
<tr>
<td>Warrant $w_i$</td>
<td>A general rule of inference expressing the relationship between grounds and a claim. It has the following structure:</td>
</tr>
<tr>
<td></td>
<td>if <strong>grounds</strong> then <strong>claim</strong> with <strong>strength</strong> $s$, because <strong>backing</strong></td>
</tr>
<tr>
<td></td>
<td>skeleton warrant specific to model of opinion used</td>
</tr>
<tr>
<td>Strength $s$</td>
<td>Expresses how strongly a warrant’s grounds imply its claim.</td>
</tr>
<tr>
<td>Backing</td>
<td>A justification for why that strength is believed.</td>
</tr>
<tr>
<td>Rebuttal</td>
<td>Rebuttals are not used in our model of arguments (see Section 3.4.3).</td>
</tr>
<tr>
<td></td>
<td>Every model shares the same skeleton warrants. The strengths and backings are customised by the user.</td>
</tr>
<tr>
<td>Arguments:</td>
<td></td>
</tr>
<tr>
<td>Claims $C$</td>
<td>Data, similar to grounds, inferred by argumentation rather than provided by the user.</td>
</tr>
<tr>
<td>Qualifier $q$</td>
<td>Expresses the degree to which a ground or claim is believed.</td>
</tr>
<tr>
<td>Argument</td>
<td>An argument represents how conclusions were reached using the data, a model, and additional information from the user.</td>
</tr>
<tr>
<td>Tree</td>
<td>An argument is represented as a tree, each node representing a (qualified) claim or conjuncts of claims, and each arc representing a warrant with an attached backing.</td>
</tr>
<tr>
<td>Structure</td>
<td>The structure of a tree is that tree with qualifiers, strengths and backings removed.</td>
</tr>
</tbody>
</table>

Table 4.1: Summary of Terminology.
4.5 Representing Data in Optimist

A prospect is described in Optimist as a set of attribute-value pairs. In addition, a number of other geological objects (wells, faults, and regions) required for the appraisal are described in the same way. The grounds of arguments refer to this data.

Although our model allows grounds to be qualified, we make the simplifying assumption in Optimist that grounds are always unqualified (i.e. assumed certain). Instead, uncertainty is expressed in the ground itself. For example, “it’s unlikely that (qualifier) there is a trap (grounds)” would be rephrased “{certainly (implicit qualifier)} the trap is unlikely (grounds)”. This simplifies the calculus of strengths and qualifiers.

4.6 Representing Qualifiers and Strengths

Before describing how arguments are represented, we first present the calculus of strengths and qualifiers used in Optimist. For clarity, we re-iterate the distinction between qualifiers and strengths. Qualifiers are associated with grounds and claims, describing degrees of belief that those grounds or claims are true. Strengths are properties of warrants, referring to the relationships between their grounds and claims.

4.6.1 Requirements

An argument concludes a qualified claim from some grounds using warrants. In order to compute the appropriate qualifier for a claim, some calculus is needed to combine the qualifiers of grounds and the strengths of those warrants. There are two requirements which the calculus must meet:
1. It is local, i.e., the combination rules are functions of the warrants’ strengths and the grounds’ qualifiers only. This requirement follows from our definition of warrants (Section 3.4.3).

2. It is easy to understand. For meaningful dialogue to occur between users and the system, users must be able to understand the system’s behaviour and be able to express their disagreements.

### 4.6.2 Optimist’s Combination Rules

To meet these requirements, we adopt a simple calculus as follows:

- Qualifiers are equated with (subjective) probability estimates that their corresponding claims are true.

- Warrant strengths are equated with the amount that the claim’s prior qualifier (i.e., probability) should be multiplied by or modified given that the warrant’s grounds are true. This makes an implicit independence assumption, which broadly holds for prospect appraisal. We discuss this further in Section 4.6.3.

- The initial qualifier for claims is 1.0 (the ‘initial presumption’, using Toulmin’s terminology), and warrant strengths are always less than or equal to 1.0. Thus warrant strengths reduce this initial qualifier if the reasoning is not conclusive.

Thus, if grounds are unqualified, the claim’s qualifier $q$ is simply the product of the strengths $s_i$ of those warrants concluding a claim and whose grounds are true:

$$q = \prod s_i$$  \hspace{1cm} (4.1)

This calculus is suitable as it is easy to understand, and reflected the way geologists reasoned about the probability of finding oil at a prospect. To estimate the strength $s_i$ of a warrant relating some grounds $g_i$ to a claim ‘oil is present’, geologists thought in terms of a ‘perfect prospect’ as follows:

“Consider a (nearly) ‘perfect prospect’ where the only risk (that the prospect is dry) is posed by the grounds of a particular warrant. (All other properties of the prospect are ideal, posing no risk). What, then, is the probability of oil at that prospect?”

This probability is then assigned as the warrant’s strength. Warrants strengths thus range from 0 (the grounds strongly indicate the claim is false) to 1 (the grounds have no effect on the claim). Note that we are using warrant strengths to denote ‘negative evidence’ here, as a strength less than 1 can only decrease the qualification of the claim (Equation 4.1). (Warrants are defined only as expressing a relationship between grounds and claims; there is no reason why this relationship cannot be a negative one). Also note that a warrant has no effect on the claim (i.e., no strength) when $s = 1.0$.

### 4.6.3 Discussion

We make three points. First, the particular calculus we use is not central to our argumentation approach – any calculus which meets our two requirements (Section 4.6.1) will suffice. Second, the simplicity requirement is essential for the argumentation model to be practically applied, and is the primary motivation for choosing the simple combination method described. Alternative calculi, e.g., Prospector’s Bayesian calculus, can present substantial problems for users to understand. Duda, for example, reports:
“Because the [Bayesian] techniques...are unfamiliar to most geologists, few geologists can even evaluate Prospector's models, let alone contribute to their development” [Dud80] p26

Third, the role of strengths and qualifications must be carefully understood. Their primary roles are to organise backings and to permit users to express their opinions adequately. Provided these roles are fulfilled, we do not necessarily require absolute precision in quantifying belief. Thus minor variations in quantified belief due to the imprecision of users’ judgements and minor violations of the independence assumption will be tolerated.

4.6.4 Generalising the Calculus

Optimist further simplifies the use of qualifiers and strengths by constraining grounds to be unqualified (Section 4.5). We point out, however, that a generalisation of this calculus can be envisaged in which grounds can be qualified. The qualification \( q \) of a conjunct of grounds is simply the product \( \prod q_{gi} \) of each ground’s qualification \( q_{gi} \), assuming independence of each ground. Qualified grounds would have reduced effect on the claim’s qualification. Given a warrant with strength \( s \), and whose grounds \( g \) had qualification \( q_g \), the claim’s prior qualifier \( q_o \) would be modified after applying the warrant to become

\[
q = q_o(1 - q_g(1 - s))
\]

Equation 4.1 is thus a special case of this where \( q_g = 1.0 \).

As an example, consider the claim ‘oil is present’ to initially have a qualifier \( q_o = 1.0 \). A warrant with strength \( s = 0.5 \) would reduce \( q \) to be 0.5 if its grounds were certain (\( q_g = 1.0 \), or to be 0.75 if its grounds were qualified by a probability of 0.5 (\( q_g = 0.5 \), or leave \( q \) unchanged at 1.0 if its grounds were false (\( q_g = 0.0 \). Thus if the warrant definitely applies (\( q_g = 1.0 \)) then the qualifier is reduced the most, and reduced less if the warrant’s grounds are less certain.

4.6.5 Logical Warrants

We have defined warrant strengths as modifying a belief in a claim. However, it is useful to also define a different type of warrant in which the claim is completely determined by the grounds with the following properties:

- Given the grounds are true, the claim will definitely be true.
- Given the grounds are false and no other warrant concludes the claim, the claim is assumed false.

These warrants are equivalent to logical implication under the closed world assumption (CWA). We term these logical warrants. Warrants which express definitions, for example, are logical warrants. If the grounds \( g \) are qualified, then the claim's qualifier \( q \) is simply equal to the grounds qualifier, i.e. \( q = q_g \). Again, conjuncts of grounds are combined by multiplication i.e. \( q_g = \prod q_{gi} \), assuming independence of each ground.

4.7 Representing Arguments

4.7.1 Description

We now turn to the representation of arguments in Optimist. In the subsequent section we describe how models encode the warrants used for constructing these arguments and the algorithm for constructing arguments from them.
Optimist's arguments share similarities in structure because they are constructed from a common set of skeleton warrants. These structural similarities are domain-specific, determined by the skeleton warrant set.

For Optimist, a particular form of argument structure was designed reflecting the general risk assessment procedure described in Section 4.2.2. It should be noted that the structuring of the appraisal into six risk components is a company-specific practice; other companies may use different structures, and hence the skeleton warrant set may be different for an application in a different oil company.

This is as follows:

1. The claim \( C \) (the root node) is that the prospect contains oil. Its qualifier \( q \) is the probability that this claim is true. Judging the value of this qualifier is the overall task of prospect appraisal.

2. A single logical warrant connects the six necessary conditions for oil (described in Section 4.2.2) to the root node. This warrant occurs in every argument in Optimist. These conditions are: trap presence, trap effectiveness, reservoir presence, reservoir communication, source presence and source maturity.

3. For one of these six, reservoir presence, a single logical warrant similarly connects its three necessary (sub-)conditions to the node. These three conditions are presence of gross, net and porosity.

4. For four of the six (trap presence/effectiveness and source presence/maturity), evidence affecting the (sub-)claim (eg. trap is present) is connected to it via warrants whose strengths are modifiable by the user. These strengths represent how the user believes the evidence affects the claim.

The number of warrants that are applied in the argument is prospect-dependent, depending on the particular features of the prospect.

5. Further down the tree still, the grounds of these warrants might be concluded by further warrants. These warrants are constrained to be logical warrants. The grounds of these logical warrants might be the claims of other logical warrants, whose grounds might be claims of other logical warrants, and so on.

6. The grounds at leaf nodes in the tree are represented as certain, ie. have qualification \( q = 1.0 \).

7. For reservoir communication and gross/net/porosity presence, single 'special' warrants connect grounds describing the prospect's location and depth with the claim.

These single warrants are handled in a special way in Optimist, since their strengths are determined statistically using nearby well information rather than by retrieving a value of \( s \) from the user's model. The statistical sample and method used to determine \( s \) represents the backing to the warrant. This is described in Section 4.7.4.

A sketch of the argument tree is shown in Figure 4.3.

### 4.7.2 Important Features

The main features of the argument tree can thus be summarised:

1. At the top of the tree, two logical warrants break the appraisal problem into eight sub-problems.

2. For each subproblem, warrants with user-tunable strengths connect with the claim.
Figure 4.3: The structure of an argument in Optimist.

Figure 4.4: Separating the subarguments.
In the right-hand diagram, 0.28 is the product of the warrant strengths \((0.8 \times 0.7 \times 0.5)\). Note that warrants reduce the initial qualifier of \(q = 1\). This is because warrant strengths are less than 1, denoting evidence against the claim (Section 4.6.2).

Figure 4.5: Normal and constrained use of strengths and qualifiers in subarguments.

3. Logical warrants may in turn be used to prove the grounds of these warrants.

4. Leaf nodes are expressed as certain.

A sketch of this separation is shown in Figure 4.4.

4.7.3 Constraints

Constrained Use of Strengths and Qualifiers

For each subproblem, only warrants connecting directly with the subproblem’s claim can have their strength altered (since all warrants lower in the argument tree are logical warrants). Figure 4.5 illustrates an argument constructed under this constraint.

The motivation for this is to aid comprehensibility and ease of use of the system: The strength/weakness of a subargument contributing to the claim is localised in a single warrant rather than dispersed over all the warrants in the subtree. Thus the warrant provides an immediate focus for dispute. Additionally, the comprehensibility of a numeric representation for strengths and qualifiers is assisted by limiting the number of times they are combined within the argument.

Grounds expressed as Certain

For similar reasons, as described in Section 4.5, grounds are assumed unqualified. Figure 4.5 also illustrates an argument constructed under this constraint.

4.7.4 Special Warrants

Optimist additionally includes the use of ‘special warrants’. Unlike other warrants in a model, their strengths are not directly represented but instead estimated at run-time from statistics
over ‘relevant’ wells, weighted by their degree of relevance. For some statistics the user may be required to assist in deriving a warrant strength from them.

The relevance of each well is itself concluded from another argument, whose claim is that the well is relevant and whose qualifier represents the degree of relevance. The warrants concerning well relevance are stored in the model in a similar way to warrants concerning prospect risk.

The backing of a special warrant is thus a statistical justification, which can be further examined by the user (“why this special warrant strength? well then why this statistic? well then why is this well relevant?”). Special warrants can thus be seen as using a more sophisticated form of backing. The algorithm for generating special warrants is summarised in Table 4.7.

### 4.7.5 Storing Arguments

For speed and memory efficiency, the whole argument tree is not stored in Optimist. Instead, only the warrants with user-tunable strength and special warrants are stored. The qualifications of claims above these warrants are also stored for speed, although strictly this is not necessary as they can be derived from the warrants lower in the tree.

From this stored information, the complete argument tree can be reconstructed except for details of how logical warrants were instantiated to satisfy the grounds of user-tunable warrants.

An example of the data structures used is given in Appendix A.

### 4.8 Representing Models

#### 4.8.1 Representing Warrants

A model consists of a set of skeleton warrants, and an assignment of a strength and backing to each skeleton warrant.

**Skeleton Warrants**

A skeleton warrant is represented in a rule-like fashion using the syntax shown in Table 4.2.

Logical warrants are a special case of normal warrants, in which the warrants’ implication behaves like logical implication. In skeleton form, they are indistinguishable from normal warrants. To identify them, they are tagged with an identifier.

Some of the skeleton warrants in Optimist are shown in Table 4.3.

**Warrant Strengths and Backings**

The assignment of strengths and backings to the skeleton warrants, according to a particular model of belief, is stored as a set of 4-tuples of the form:

\[
\text{WarrantStrength} = \langle \text{WarrantNo}, \text{Strength}, \text{TimeStamp}, \text{Backing} \rangle
\]

where:

- **WarrantNo** is the number of the skeleton warrant,
- **Strength** is the warrant’s strength, an integer between 0 and 100 (representing strength as a percentage),
- **TimeStamp** records when **Strength** was assigned. (This allows historical records of changes to the warrant’s strength to be maintained), and
- **Back**ing is an uninterpreted string of text entered by the user, justifying why the warrant was considered of strength **Strength**. In a model, a backing is a general (rather than prospect-specific) justification.
\[
\begin{align*}
\text{<skeleton-warrant>} & ::= \text{if <grounds> then <claim>} \\
\text{<grounds>} & ::= \text{<grounds> and <grounds>} \\
& \quad | \quad \text{<grounds> or <grounds>} \\
& \quad | \quad \text{not <grounds>} \\
& \quad | \quad \text{<ground>} \\
\text{<ground>} & ::= \text{<att> of <object> <reln> <val>} \\
& \quad | \quad \text{<object> <reln> <att>} \\
& \quad | \quad \text{<prolog-call>} \\
\text{<object>} & ::= \text{prospect : well | prospect(<var>) | well(<var>) | fault(<var>) | lith(<var>) | region(<var>)} \\
\text{<reln>} & ::= \text{is | isnt | has | hasnt | ==} \\
\text{<att>} & ::= \text{<feature> at <horizon>} \\
& \quad | \quad \text{<feature>} \\
\text{<claim>} & ::= \text{<ground>} \\
\end{align*}
\]

Vocabulary definitions:
\[
\begin{align*}
\text{<prolog-call>} & ::= \text{An arbitrary Prolog predicate, evaluated by the Prolog interpreter when its truth value is required} \\
\text{<feature>} & ::= \text{depth, net, gross, porosity, maturity, etc.} \\
\text{<horizon>} & ::= \text{top forties, top fulmar, base cretaceous etc.} \\
\text{<value>} & ::= \text{<integer> | <real> | <string>} \\
\text{<var>} & ::= \text{is a variable} \\
\end{align*}
\]

Table 4.2: Syntax of skeleton warrants.
logical-warrant 0 ::
if trap is present
and trap is effective
and reservoir is present
and reservoir is 'in communication'
and source is present
and source is mature
then oil is present.

skelwarrant 1 ::
if 'reservoir mapping' of prospect is direct
then trap is effective.

skelwarrant 2 ::
if 'reservoir mapping' of prospect is 'indirect, constant'
then trap is effective.

skelwarrant 3 ::
if 'reservoir mapping' of prospect is 'indirect, anomalous'
then trap is effective.

skelwarrant 4 ::
if trap of prospect is faulted
and 'most recent faulting' of prospect is Rock1
and 'seal unit' of prospect is Rock2
and Rock1 is_younger_than Rock2
then trap is effective.

logical-warrant 5 ::
if trap_in_dirn( ) of prospect is Fault
and is_a_fault(Fault)
then trap of prospect is faulted.

logical-warrant 6 ::
if stage_age(Rock1, AgeRock1)
and stage_age(Rock2, AgeRock2)
and AgeRock1 < AgeRock2
then Rock1 more_recent_stage_than Rock2.

Table 4.3: Some example skeleton warrants.

<table>
<thead>
<tr>
<th>Warrant</th>
<th>Strength</th>
<th>TimeStamp</th>
<th>Backing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>23-Jan-91 10:31am</td>
<td>‘Direct mappings perfect by definition’</td>
</tr>
<tr>
<td>2</td>
<td>80</td>
<td>20-Jan-91 5:40pm</td>
<td>‘Constant corr. proven reliable in 14th round’</td>
</tr>
<tr>
<td>3</td>
<td>40</td>
<td>20-Jan-91 5:43pm</td>
<td>‘Anomalies serious here (see p:43/21-1)’</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>02-Feb-91 12:43pm</td>
<td>‘Late throws no problem as calcite conc. high’</td>
</tr>
</tbody>
</table>

Table 4.4: Strengths and backings in a model of opinion.
<table>
<thead>
<tr>
<th>Warrant</th>
<th>Strength</th>
<th>Time</th>
<th>Backing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-'direct'</td>
<td>100</td>
<td>23...</td>
<td>'Direct mappings perfect by definition'</td>
</tr>
<tr>
<td>1-'indirect, constant'</td>
<td>80</td>
<td>20...</td>
<td>'Constant corr. proven reliable in 14th round'</td>
</tr>
<tr>
<td>1-'indirect, anomalous'</td>
<td>40</td>
<td>20...</td>
<td>'Anomalies serious here (see p43/21-1)'</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>02,...</td>
<td>'Late throws no problem as calcite conc. high'</td>
</tr>
</tbody>
</table>

Table 4.5: Revised numbering of warrants to group subsets together.

For example, the assignment of strengths and backings according to a particular model might appear as in Table 4.4. Each model also has a header summarising it of the form:

\[
\text{ModelHeader} = \{\text{ModelNo, ParentModelNo, Owner, Name, LastUpdated}\}
\]

where:

- \text{ModelNo} is the code number of the model
- \text{ParentModelNo} is the parent model from which \text{Model} was originally copied
- \text{Owner} is the name of the person who owns the model
- \text{Name} is the name of the model
- \text{LastUpdated} is a time stamp of the last update to the model

For example:

<table>
<thead>
<tr>
<th>Model</th>
<th>Parent</th>
<th>Owner</th>
<th>ModelName</th>
<th>LastUpdated</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>'Peter Clark'</td>
<td>'Central N. Sea Model'</td>
<td>02-Feb-91 12:43pm</td>
</tr>
</tbody>
</table>

Grouping Warrants

Sometimes, a warrant will include a test for a particular value of an attribute, and then other warrants will test on all other values of that attribute. An example is in Table 4.3, where warrants 1, 2 and 3 test on each of the (exhaustive, mutually exclusive) values of \text{reservoir mapping}.

To reflect this grouping, we can make a simple syntactic transformation whereby the attribute value is looked up, rather than tested on, during application of the warrant. To keep the different warrant strengths for different attribute values distinct, the looked-up value must be included in the warrant’s ‘number’. For example, the first three warrants in Table 4.3 would be replaced by a single warrant:

\[
\text{skelwarrant 1-Value ::}
\]

\[
\text{if 'reservoir mapping' of prospect is Value}
\]

\[
\text{then trap is effective.}
\]

and the assignment of strengths would become as in Table 4.5.

This makes the set of skeleton warrants more concise, and allows Optimist to display other warrant strengths from the group for comparison when the user queries a particular warrant’s strength.

4.8.2 Creation and Organisation of Users’ Models

The degree of risk associated with a particular feature of a prospect may be (geographical) area-dependent, or even prospect dependent. To represent this, users may have several models, each one appropriate for a specific areas. These models are arranged hierarchically (a hierarchy for each user), each inheriting warrant strengths from its parent if none is specified. The root or
'default' model is the initial model set up and tuned by the geologists during system development (Section 4.10.1). A new model is created by copying one of the existing models (usually the default model) as a starting point.

4.8.3 Constructing Arguments from Models

An argument is constructed from a model, a claim and a case description. The algorithm for constructing an argument is described in Table 4.6, and implements argument construction subject to the constraints described in Section 4.7.3. The algorithm for generating special warrants in shown in Table 4.7.

4.9 The Dynamics of Argumentation

We have described the three main sources of data used by Optimist: facts, arguments and models. We now turn to the procedural side, and how these structures are used for argumentation.

The overall flow of control within Optimist is illustrated in Figure 4.6. Pseudo-code and Prolog code implementing this are given in Appendix E. We now describe and discuss the main features of the interaction. An extended example of the system in operation is given in Section 4.12.

4.9.1 Goals of the Interaction

Although the word argumentation has adversarial overtones, the interaction is essentially cooperative. The user is not trying to 'win' against the system; rather he or she is trying to construct as good an argument as possible for a conclusion. When 'arguing' against the user, the system is not trying to be stubborn but to alert the user to information which he or she may be unaware of. The user's motivation for using the system is to obtain this information, and thus the user is positively interested in what contradicting and supporting knowledge Optimist can find.

The goal for Optimist is to help construct and store a model of the user's argument for the probability of oil at a prospect. In particular, Optimist argues for consistency with previous judgements which have been made, represented in its database of arguments. The user's role is to supervise the argument construction process, arguing with conclusions he or she disagrees with, and in doing so supplying any extra information (as backings) which the system is unaware of. The user's roles are thus as a critic and a source of new knowledge in the form of backings.

Thus a symbiotic relationship exists between the Optimist and the user: Optimist extends the user's experience by exploiting its persistent and faithful memory of previous chains of reasoning used in appraisals; the user provides extra knowledge to overcome inadequacies and approximations in the geological knowledge captured in Optimist's models.

4.9.2 Starting the System

Initially the user logs onto the system, providing his or her name. This is important, as arguments constructed during the session are tagged with the user's name and date.

The user selects the prospect he or she wishes to appraise, and provides or edits the data describing the prospect (box 2 in the flow chart in Figure 4.6). Following this, if the user desires to conduct a new appraisal, an initial argument is constructed by Optimist. This is done using a model of the user's opinion which the user has selected (box 5) and the argument forms the focus of subsequent interaction.
This algorithm constructs an argument for a claim, following Optimist’s constraints described in Section 4.7.3. Pseudo-code and Prolog code for a simple shell employing this algorithm are given in Appendix E.

**Data structures used:**
- Argument = {Qualifier, Claim, Warrants}
- Model = {Number, Owner, Name, Warrants}
- Warrants = {Warrant_1, ..., Warrant_n}
- Warrant = {WarrantNo, Grounds, Claim, Backing, Strength}
  (Logical warrants are tagged, and have no Strength)
- Grounds = {Ground_1, ..., Ground_n}
- Claim, Ground = Att_i=Val_j
- Case = {CaseName, Data}
- Data = {Att_1=Val_i, ..., Att_n=Val_j}

```plaintext
procedure construct_argument(Claim, Model, Case) returning Argument:
  let ApplicableWarrants := {}
  forall Warrant in Model where Claim is-in Warrant
    if    forall Ground in Grounds, establish(Ground, Model, Case)
    then  add Warrant to ApplicableWarrants
  let Qualifier := combine_strengths_of(ApplicableWarrants)
  let Argument := {Qualifier, Claim, ApplicableWarrants}.

procedure establish(Ground, Model, Case) returning true/false:
  if       Ground is the negation of some other ground Ground2
  then    if not establish(Ground2) then return true else return false
  elseif  Ground is-in Data in Case then return true
  else    let SubClaim := Ground
            if there is a LogicalWarrant in Model where SubClaim
               is-in LogicalWarrant, and, forall SubGround in
               LogicalWarrant, establish(SubGround, Model, Case)
            then return true
            else return false.

procedure combine_strengths_of(Warrants) returning Qualifier:
  forall Warrant_i in Warrants, lookup Strength_i in Warrant_i
  let Qualifier = [ Strength_i ].
```

Table 4.6: The Argument Construction Algorithm.
The boxes describe actions from Optimist’s point of view.

Figure 4.6: Flow of Control in Optimist.
procedure generate_special_warrant(Claim, Model, Case) returning SpecWarrant:
  (this procedure uses the global database of wells and other geological objects)
  (Claim is of the form ‘Parameter is adequate’, eg. ‘porosity is adequate’)
  let WellsToUse := {} 
  forall Well in the global database of wells 
      let RelevanceArgument :=
          construct_argument(Well is relevant, Model, Case)
      let Relevance := Qualifier in RelevanceArgument
      add { RelevanceArgument, Relevance, Well } to WellsToUse
  Ask user for statistical Method to use (eg. plot, histogram)
  Using Method, interpolate from known values of Parameter in WellsToUse, 
  each weighted by Relevance, to predict Value of Parameter for Case 
  and Confidence that the Value is adequate for finding oil.
  lookup Position and Depth in Case
  let Grounds := { Position, Depth } 
  let Strength := Confidence
  let Backing := { WellsToUse, Method, Value } 
  let SpecWarrant := { Grounds, Claim, Backing, Strength }.

Table 4.7: The algorithm for generating special warrants in Optimist.

4.9.3 Argumentation

Having constructed an initial argument for oil probability, the system and user then interact 
to discuss this argument (the main argumentation loop shown in Figure 4.6). The system can 
either argue in defence of a warrant in the argument (responding to a challenge by the user) or 
can check the argument by attempting to argue against warrants within it. The same procedure 
is used for both these operations, with a parameter controlling whether to argue for or against 
warrants.

Optimist attacks (or defends) a warrant by locating evidence which conflicts with (or supports) 
its strength. Evidence which is consistent with the warrant is considered to support it, and 
inconsistent evidence attacks it. There are several sources of evidence which Optimist can use 
(eg. statistics, other models of opinions). However, the most important is the use of previous, 
precedent arguments. For brevity we present the use of precedents for challenging a warrant only.

Arguing with Precedents: Case-Based Reasoning

To attack a warrant, Optimist searches for previous appraisal arguments (‘cases’) where the 
same (skeleton) warrant was applied, but with a significantly different strength. These cases 
suggest a possible inconsistency may exist in the current (or previous) reasoning, and the user is 
alerted. The backing of the warrant in the precedent case explains why the different strength was 
considered appropriate, and can be inspected by the user. It is the backing which substantiates 
the challenge to the current argument under construction.

The use of precedent arguments is a form of case-based reasoning, where the case consists 
ot only of a problem description but also a ‘snap-shot’ of a geologist’s reasoning about that 
problem. Using Aha’s framework for case-based reasoning [AKA91], the details of the three 
main components are as follows:

Similarity Assessment: A relevant, previous case must be one in which same skeleton warrant
whose strength is under dispute was applied. However, its relevance — the degree to which the perceived risk in the previous case should also apply in the current case — also depends on a number of other properties. In the prospect appraisal problem the most important of these are:

- the similarity in the geological structure of the old and new prospects,
- the similarity in their reservoir stratigraphies, and
- their physical separation distance.

The similarity function is thus a function of three relationships between the old and new cases.

Two structures are considered ‘similar’ if they both occur below the same, user-selected node in a taxonomy of structures. Stratigraphies are treated likewise, using a hierarchy of stratigraphies. For distance, the user provides a threshold distance below which the cases are considered similar and above which they are not.

This similarity function is unusual for case-based reasoning for two reasons. First, it is boolean valued rather than continuous. Second, it is interactive, the user selecting how constrained a definition of similarity should be used. These two features are chosen to provide an interactive, easy-to-use case retrieval mechanism for argumentation. Typically the user will experiment to obtain a suitably constrained number of precedents. This is analogous to the experimental use of key-word combinations when searching an on-line library database in order to suitably constrain the number of documents retrieved.

**Performance Component** Having retrieved previous, relevant arguments which conflict with the current argument, Optimist presents them to the user. Following this, the user may ask to see the backings explaining why the contradicting judgements were made, view the particular arguments, comment on them or modify his or her current argument in the light of this evidence. Thus the performance component is one of assisting expert judgements in the context of argument construction.

Again, this performance component is unusual for a case-based reasoning system. Rather than performing the typical task of classification, cases are used to provide evidence for/against claims made in the current working argument. This illustrates a practical use of case-based reasoning integrated into a wider, knowledge-based task (argument construction).

**Learning Component** The system’s knowledge is expanded as new cases and arguments are stored. The argument database does not use any complex indexing or memory organisation techniques beyond those implicit in the (Oracle) database software itself.

A second use of case-based reasoning is in the construction of special warrants (Section 4.7.4 and Table 4.7) where ‘relevant’ wells are gathered. Each well’s relevance (a kind of similarity) is assessed by constructing a relevance argument, using warrants for the claim that well is relevant. This is described in more detail in [Cla88a]. The performance component makes predictions using statistics over the relevant wells, and learning occurs simply by adding new wells to the database.

**Postmortem Analysis**

Because Optimist maintains time-stamped arguments for appraisals, it can perform an after-the-event (‘post mortem’) analysis of appraisals after the outcome of drilling is known. While a dry well may not necessarily indicate an appraisal was wrong, it can lend useful support to defend (or attack) arguments in future. Postmortem analysis can be viewed as a modified form of
case-based argumentation, in which a warrant is challenged by cases where the same judgment was made but the predicted outcome did not occur.

**Argument Analysis Functions**

Optimist includes several other argument analysis tools for use by the user. These are:

**Argument Comparisons:** Given a complete argument, it is often useful to compare it with other stored arguments, locating points of difference. This is particularly useful when different geologists arrive at different conclusions about the same prospect, or when a geologist's conclusion has changed over time as new information has become available.

**Running Alternative Models:** The user can also examine the application of (a model of) someone else's opinion to see if a different conclusion would have been reached.

**Prospect Retrieval:** Optimist also allows the user to search for prospects with similar descriptions to the current prospect (even if there are no appraisal arguments associated with them)

We do not claim to have provided a full set of features in support of arguments. Rather, we have aimed to satisfy users in their daily use of the system, providing a sufficiently broad set of facilities to meet this objective.

### 4.9.4 Responding to Correction

We have described how Optimist supplies information to the user, presented in the context of argument construction. The user can also supply information back to Optimist by over-riding the system's current argument. To do this, the user indicates which warrant he or she disagrees with, and provides the revised strength and backing for the warrant (box 8 in Figure 4.6). This change can be made for the current argument only, or can also be made to the general model of opinion used to construct the argument. From here, the user can ask the system to check the argument or challenge other warrants, thus iterating around the argumentation loop in Figure 4.6.

### 4.9.5 Storage of the Argument

When the user finally accepts the current argument, it is stored, indexed by the user's name and date. The stored argument represents a snapshot of the user's reasoning at that time, about that prospect, and includes the judgments and justifications for those judgments. In this way, the system's repertoire of arguments is always expanding as new appraisals are conducted with it. Section 4.12 gives an illustrated example of the system in use.

### 4.10 Implementation Information

#### 4.10.1 System and Default Model Construction

The system was initially prototyped between November 1987 and June 1988, followed by a full-scale implementation over the next year. During that period, close interaction with two company geologists was maintained. These are geologists 'A' and 'B' in the evaluation in Chapter 5. Meetings were held approximately once every eight weeks, offering valuable feedback on the design of the system and its interface.

As stated in Section 4.8.2, the initial 'default' model of opinion (from which users' own personalised models are established) was constructed during system development. Early on, the warrant strengths were assigned values by geologist A. During meetings in the year of system
development, approximately five real-world prospects were appraised using the partially developed system. These appraisals were made with a single model of opinion, thus enabling it to be ‘tuned’. Both geologists A and B contributed to modifying the warrant strengths, with conflicts between them either being resolved by compromise or simply by decree of the user controlling the system. Thus the default model is a rather mixed and partial representation of both these geologists opinions during that year; however, as its role is as a starting point for users to construct their own models (rather than for providing advice) its precise tuning is not essential. Both geologists A and B considered it satisfactory for this role at system installation. At installation in July 1989 the default model was updated to a copy of this tuned model. The appraisals made during development time were not included in the installed system.

4.10.2 Issues of Interface Design

We briefly describe some of the user-interface design issues which arose during Optimist’s construction. Although independent of the general argumentation model, these considerations contributed to the specific application’s usability, hence we mention them here. Although the screen design is perhaps not perfect, it was quickly accepted for use in the company environment. The lessons which were learned during its design are thus valuable to recount:

1. As the interface is mainly event-driven, users found it important to know whether the system was waiting for them to perform an action or whether they should wait for the system. A traffic light icon was used to illustrate this (red: system busy, green: system waiting, amber: system waiting for response to a specific question).

2. A pop-up question box was used to ask users for specific items of information when needed, thus focusing their attention on what their next action should be.

3. The interface consists of a number of ‘cards’ (screen layouts), only one of which is visible at one time. Although it would have been possible to let users arbitrarily jump from card to card, it was found to be better to force users to step forward and backward through cards in a fixed sequence. Although slightly more time consuming for the users, it helped them quickly acquire a picture of where they were in the system.

4. A particularly interesting issue arose concerning post-installation changes to the interface: it was confusing for the users when the cosmetic features of the interface were changed, even if a change improved the consistency of the interface (e.g., always exit a card with the same shaped button). The two points here are that first, it is important to make sure the initial interface is as good as possible, and second, that there is a strong case to be made for making minimal changes to interfaces once installed – even if this means inconsistencies remain. We express caution that consistency must be balanced against cost of relearning within an established system.

5. The use of colour, graphics, and the mouse were important in achieving an acceptable, usable interface.

6. Icons were only of moderate use; often it was necessary to annotate them with text to clarify their meaning. Their main role was to improve the visual impact of the interface.

7. It was found sometimes useful for the system to interrupt the user’s request to offer additional functions (e.g., if the user requests to save a finished appraisal, the system interrupts to ask whether the user would first like a brief analysis of the appraisal made). This helps users explore the system’s capabilities, but also must be used sparingly in order not to become a source of frustration for the user.
4.10.3 Implementation Statistics

Optimist is implemented in approximately 15,000 lines of Quintus Prolog, running on a Sun workstation. It uses HyperNeWS [Rud90] for its user interface, and the Oracle database for mass data storage.

A summary of the percentage of code in different modules of Optimist is shown in Table 4.8. We summarise the most interesting features:

1. Although the core of the system can be implemented in about 1000 lines of code (listed in Appendix E), the full-scale commercial system is approximately 15 times larger, reflecting the substantial software engineering issues which must be addressed for practically applying AI.

2. As the system is highly interactive, it is difficult to classify code as 'user interface' or 'underlying system'. However, we can say that control of the user interface accounts for a substantial portion of the system; we place this figure at approximately 40% - 50%. Furthermore, the low-level graphical commands controlling the appearance and position of items in the interface are coded in a further 6500 lines of PostScript code and are not included in Table 4.8.

Other authors have reported similar high figures for user-interface components, eg. 42% for Dipmeter [SB83] and 33%-50% for Pride [BMS88].

3. The argument construction algorithm is small and simple, accounting directly for about 3% of the code (it indirectly uses other code, eg. interfaces to models and data).

The set of skeleton warrants comprises 31 warrants relating to the claim that oil is present, 23 relating to the claim that a well is relevant to the current prospect (used for generating special warrants), and a further 10 logical warrants.

4.11 Summary

As the implementation has been described in some length, it is worth summarising the main features and the benefits they produce:

- Using the argumentation approach, the system constructs structured records of experts' reasoning (arguments) in solving specific problems, as well as maintaining general models of different experts' opinions.

- These records are a powerful resource in solving problems. An expert's skill is in part determined by his experience in previous problems. However, no single expert knows every appraisal, of every prospect, in every location, which has ever been conducted. Optimist can extend the expert's resource of experience by storing and selectively bringing appraisal arguments to the user's attention through argumentation.

- Defining warrant strength as a 'probability modifier' (Section 4.6) provides a simple, easy-to-understand calculus for warrant strengths. We examine this further in the evaluation in Chapter 5.

- Logical warrants are used to express definitions and logical relations.

- The grounds for arguments are expressed as certain (any uncertainty being represented within the grounds themselves, Section 4.5), and only the top-level warrants have user-tunable strengths. This again makes the strength calculus simpler and hence easier to understand.
### Component (and comments) | Lines of code
---|---
**Basic system utilities:**
database interface | 2000 (13%)  
general supporting utilities | 2000 (13%)  
**Startup modules:**
startup | 500 (3%)  
data entry | 500 (3%)  
display/select arguments | 500 (3%)  
selection of, comparison of and interface to models | 1000 (7%)  
**Argumentation:**
HCI (general user interface and flow of control) | 1500 (10%)  
inference engine (for initial argument construction) | 500 (3%)  
special warrants (machinery for generating them) | 500 (3%)  
argumentation (system’s argumentation functions) | 1000 (7%)  
warrant modification | 500 (3%)  
**Other facilities:**
plotting (used in generating special warrants) | 1500 (10%)  
map display (presentation of geological information) | 1500 (10%)  
report generator (hard copy appraisal summaries) | 1000 (7%)  
dumb terminal interface | 500 (3%)  
Total lines of code | 15000

Table 4.8: Number of lines of code in different modules in Optimist.
• The degree to which it is possible to generalise a risk judgement varies from global judgements to prospect-specific judgements. Users can maintain models of their opinion in different areas, arranged hierarchically, reflecting these degrees of generalisation. Special warrants are used where the warrant strength is almost always prospect-specific.

• Optimist constructs the initial argument, and the dialogue focuses on this argument. This reduces the work for the user, and provides a starting point for the appraisal that is consistent with the user’s model.
4.12 Example of The System In Operation.

We now illustrate the process by which the system and user ‘argue’ to interactively construct an argument for a probability of oil. The flow of control loosely follows the flow chart in Figure 4.6 unless otherwise stated. Interaction with the system is through a graphical interface, snapshots of which we show below. The expert can click on buttons or select a line of text by double clicking on it.

It should be noted that the target users often use the word risk as synonymous with probability. This flexible use of the word ‘risk’ has also been employed in the user-interface to keep the system’s messages in a language familiar to users. Warrant strengths are described to the user as ‘risk values’ for the same reason. Finally, probabilities are often presented to users as odds, again for their ease of understanding. The reader should bear this in mind when reading the word risk below.

4.12.1 Startup

Initially the user logs on to the system, giving his or her name and the region in which he or she wishes to work. The username is important, as the work a user does is tagged with his or her name. The user will select and zoom on a map of the region containing the prospect to appraise, as shown in Figure 4.7.

The positions of prospects and wells are shown as highlighted objects superimposed on the map image. These have been previously outlined on the map by the user using the mouse. New objects can be added if the user so desires.

4.12.2 Problem Description

The user then selects the prospect he or she wishes to appraise, either clicking on an existing prospect on the map or adding a new prospect to start a new appraisal.

Following this, the user inputs or edits an attribute-value description of the prospect, as illustrated in Figure 4.8. (This corresponds to box 1 in the flow chart in Figure 4.6).
4.12.3 Set Up Appraisal

To start a specific appraisal of a prospect, the user clicks on the Total Risk arrow. Following this, Optimist displays a summary of the appraisal arguments which have already been constructed for this prospect, shown in Figure 4.9. This corresponds to box 2 in Figure 4.6. Each line is a summary of the complete argument for the oil likelihood shown, stored by the system indexed by user and date.

At this stage, the user can load one of these arguments to view or edit it, or start a new appraisal. Here we consider the user starting a new appraisal from scratch.

4.12.4 Selecting a Model

The user must select a model to use before he can start an appraisal, illustrated in Figure 4.10 (corresponding to box 3 in Figure 4.6). The model is termed a ‘rule set’ for ease of understanding by the users, but is in fact a set of warrant strengths and backings which attach to the common set of skeleton warrants for constructing the appraisal argument. The user can either select his own model for the given region, or someone else’s for comparison (Figure 4.10).

Models are organised hierarchically. At the top level is the global, default model. Below this there is a global model for each user of the system, and below this models which the users have tuned for particular geographical areas in which they are interested. Facilities are also provided for copying, comparing and moving models in this hierarchy. Here the user clicks on the model to use, and Optimist returns to the summary card (Figure 4.9).

4.12.5 Starting the Appraisal

Clicking on Go! starts the argumentation part of the system, generating an argument for an appraisal. The overall appraisal is broken down into the six subproblems described at the start of this Chapter. Before the argument is constructed the claims for these six components have unknown qualifiers, as shown in Figure 4.11.
4.12. EXAMPLE OF THE SYSTEM IN OPERATION.

Figure 4.9: Summary of stored appraisal arguments for prospect 3b/a-2b.

Figure 4.10: Selection of a model of opinion.
Figure 4.11: The six risk components, prior to constructing an appraisal argument.

Figure 4.12: Summary of warrants in the argument for the (sub-)claim ‘trap is present’.
In Optimist, the initial argument is constructed in stages and on demand from the user, working through the claim for each of the six components in turn. (The flow chart in Figure 4.6 is thus a simplification of this). First, the user clicks on the component which he wishes to work on. In this example the user clicks on *trap presence*, causing Optimist to move to the next page of display.

Optimist now constructs the initial argument for the (sub)claim that the trap is present, applying the warrants in the chosen model. A summary of the argument is displayed (Figure 4.12). This corresponds to box 4 in Figure 4.6. Here, only two warrants had their grounds satisfied. The claim ‘trap is present’ is assigned a qualifier of 0.85 (ie. 1 in 1.2).

The user is queried as to whether he agrees or not (in contrast with a more conventional expert system). Here the user agrees.

Clicking on the up arrow causes the system to return to the summary of the overall argument (Figure 4.13). The qualifier of the claim ‘trap is present’ is displayed. From here, the user can select another component to construct an argument for.

### 4.12.6 Dispute

Similarly warrants are applied for trap effectiveness, the resulting argument shown in Figure 4.14. Here though, the user disagrees with the qualifier of 0.16 (ie. 1 in 6.1). The particular focus of the disagreement is the judgement that the risk contributed by a facies boundary is 1 in 3.0 (warrant number 2 in Figure 4.14). Although the user considers this value to be generally reasonable, he has extra information to suggest that the facies boundary should be even more a source of concern at this particular prospect. In the language of arguments, this warrant should have a different strength than it currently has.

The user indicates this warrant which he disagrees with, and in response, the system loads the ‘rule editor’ (box 5 in Figure 4.6). (The flow chart depicts the system arguing back at this point. In fact, in Optimist a more general strategy is adopted whereby it can argue back at any point on demand for the user. We illustrate this later in Section 4.12.7).

Having selected a warrant, graphical ‘sliders’ display the strengths of this and six other warrants relating prospect structure to trap effectiveness, shown in Figure 4.15. (The grouping of these seven warrants is represented using the technique described in Section 4.8.1).
Figure 4.14: Summary of warrants in the argument for (sub-)claim ‘trap is effective’.

Figure 4.15: The strengths of seven warrants in the selected model of opinion.
The user lowers the strength of the warrant for facies boundaries. In response, the system asks the user to justify the change, providing a backing for it (Figure 4.16).

Here, although the general warrant was reasonable, the user knows two dry prospects nearby (p32 and p76) had facies boundaries and cites this evidence to support his concern about this type of prospect structure for the current prospect. This change and backing is recorded.

The user can make this change at various levels of generality: either to this particular argument only leaving the selected ‘N. West Basin’ model unchanged, or to the ‘N. West Basin’ model, or to his own global model (Figure 4.17). Here the user decides to change the warrant in the current argument only, leaving the warrant in Optimist’s models of his general opinion unchanged.

The argument for trap effectiveness is then modified and re-displayed, shown in Figure 4.18. Note that the backing for the change which the user supplied has now become part of the revised description of the argument (the second warrant in Figure 4.18).

4.12.7 Argumentation Options.

In addition, various argumentation options are available to the user, applied on demand from the user. The Argue for button causes Optimist to search for previous appraisal arguments which support (ie. are consistent with) the current argument. Here, Optimist cites several appraisals where the same (skeleton) warrants were applied with similar strengths in other
Figure 4.18: The modified argument for the claim ‘trap is effective’.

Figure 4.19: An abbreviated list of precedents supporting the current argument.

appraisal arguments (Figure 4.19).

Perhaps more importantly, clicking the Argue against button causes Optimist to search for stored appraisals inconsistent with the current appraisal (Figure 4.20). A number are highlighted where the same (skeleton) warrants were applied but with greatly differing strengths, reflecting that significantly different judgements were made. In response, the user may choose to explore some of these precedent arguments, loading one into the system to view the complete argument and the backing for the warrant’s differing strength.

These argumentation functions by default scan the whole of the database of appraisal arguments. The user, however, may wish to constrain the search to those prospects considered most relevant to the current prospect by constraining the similarity metric used in the search (described in Section 4.9.3). Some of the previous cases are highly relevant, and similar risk judgements should be made. Others are not so relevant (eg. if the rock strata involved are of greatly differing ages). To constrain the search, the user selects how broad or narrow a search should be performed by selecting parameters as illustrated in Figure 4.21.

The user selects the parameters to obtain a sufficiently large but also manageable sample of relevant previous cases. To explore the appraisal arguments that Optimist has alerted the
Hey! That doesn’t sound right!

What about the risks for prospects:
Broken Seal
prospect e/4-e (top claymore): Mike, 03-Mar-90 Broken seal risked at 60%
prospect e/4c-3 (w3, top fulmar): Dave, 05-May-90 Broken seal risked at 60%
prospect d/5-y (w32, top paleocene): Tim, 27-Feb-90 Broken seal risked at 60%
etc. etc.

traptype=facies
prospect p2/7-8 (w54, top paleocene): Nigel, 04-Sep-90 facies trap risked at 61%
prospect p5/t-3 (w82, magnus sand): Dave, 13-Jul-90 facies trap risked at 80%
prospect 2w/4-d (w3-8, aptian): Mike, 03-Aug-90 facies trap risked at 83%
etc. etc.

Figure 4.20: An abbreviated list of precedents conflicting with the current argument.

Search for cases, constrain by:

<table>
<thead>
<tr>
<th>Trap type</th>
<th>Structural (dips &amp; faults)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon</td>
<td>All horizons</td>
</tr>
<tr>
<td>Distance</td>
<td>&lt;600km away</td>
</tr>
<tr>
<td>Name</td>
<td>any</td>
</tr>
</tbody>
</table>

No. of prospects covered: 6

Figure 4.21: Defining the similarity metric for searching for precedents.

<table>
<thead>
<tr>
<th>Prospect</th>
<th>Well</th>
<th>Reservoir unit</th>
<th>AppNo</th>
<th>TotRisk</th>
<th>Who</th>
<th>When</th>
</tr>
</thead>
<tbody>
<tr>
<td>p2/7-8</td>
<td>w276</td>
<td>top fulmar/piper</td>
<td>59</td>
<td>8%</td>
<td>Peta</td>
<td>22-Jan-91</td>
</tr>
<tr>
<td>pa/5-32</td>
<td>w12</td>
<td>top sele</td>
<td>20</td>
<td>12%</td>
<td>Alan</td>
<td>31-Jan-90</td>
</tr>
<tr>
<td>8/a-43</td>
<td>w1805</td>
<td>base fulmar/piper</td>
<td>21</td>
<td>22%</td>
<td>Jim</td>
<td>27-Mar-90</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>Dave</td>
<td>52%</td>
<td>52%</td>
<td>Dave</td>
<td>27-Oct-89</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>Alan</td>
<td>41%</td>
<td>41%</td>
<td>Alan</td>
<td>26-Oct-89</td>
</tr>
<tr>
<td>7/r-re</td>
<td>w1808</td>
<td>top fulmar/piper</td>
<td>12</td>
<td>15%</td>
<td>Steve</td>
<td>21-Nov-89</td>
</tr>
<tr>
<td>7/r-re</td>
<td>w1814</td>
<td>top maureen</td>
<td>25</td>
<td>6%</td>
<td>Gary</td>
<td>14-May-90</td>
</tr>
<tr>
<td>7/3-2a</td>
<td>w2010</td>
<td>top forties</td>
<td>64</td>
<td>2%</td>
<td>Tim</td>
<td>14-Mar-91</td>
</tr>
<tr>
<td>7/3-2a</td>
<td>w2010</td>
<td>top forties</td>
<td>62</td>
<td>3%</td>
<td>Tim</td>
<td>14-Feb-91</td>
</tr>
</tbody>
</table>

Figure 4.22: Comparison of qualifications (‘TotRisk’) to the main claim ‘oil is present’ in selected arguments.
Figure 4.23: Generation of a special warrant for the claim that adequate gross is present.

user to, he or she can view them individually (retrieving one from the database as the new ‘current argument’ to explore). Alternatively, Optimist allows the user to view parts of several arguments on the same page of display. These functions can display and compare items including the strengths of the same (skeleton) warrant in each argument, their backings, data about the relevant prospects, the actual outcome of drilling the prospect, and the overall qualification to a claim (eg. ‘oil is present’) in each argument. An example is shown in Figure 4.22, where the user has asked to view the overall qualification that oil is present (‘TotRisk’) for several selected previous arguments.

4.12.8 Special Warrants

A similar procedure is carried out to calculate the risk contributed by source presence and maturity to the overall claim that oil is present. However, for reservoir presence and communication, special warrants are used to conclude risk. A special warrant collects a sample of wells and calculates a relevance for each (generating a relevance argument for each well, which the user can similarly inspect and argue with). Then statistical methods are applied to this sample, with user assistance, for inferring a warrant strength and associated parameter value for use in the current argument. This was described in Section 4.7.4 and the algorithm shown in Table 4.7.

Figure 4.23 illustrates the generation of a special warrant. Here, the user has estimated a value of gross of 200ft, and has complete confidence that an adequate thickness of gross sand will be found (a warrant strength of 1.0). This special warrant is included in the appraisal argument, and the sample of wells and statistics form the backing for this warrant. If the argument is stored, this sample and plot will also be stored with it.

4.12.9 Storing the Appraisal Argument

Having worked through the six risk components to the user’s satisfaction, the final qualification (= probability) for the claim that oil is present is calculated. The summary of the complete argument is shown in Figure 4.24. Here, the probability of oil is very low (0.03). The complete argument is then stored, and added to the summary of appraisals of the well indexed by the
Figure 4.24: Summary of the complete appraisal argument.

username and date.

In this way, the system and user interact to construct an argument for an appraisal. The interaction is an ongoing exchange of information – the user supplying backings describing new, relevant data influencing his or her risk judgement and the system providing consistency information using other appraisal arguments and data. The argument is specific to the particular user. Others may reach different conclusions using their own models and by disputing different parts of arguments. The computer's capacity for faithful memory and search, combined with AI techniques for representing chains of reasoning (appraisal arguments) as well as facts, provides the system with a potentially powerful decision support capability to assist users. In the next chapter, we evaluate how successful these techniques have been at meeting this goal in practice.
Chapter 5

Evaluation

5.1 Introduction

We have described a model of argumentation and an implementation for assisting geologists in the task of oil exploration. We now evaluate our model against our goal that it can provide useful support to users in decision-making. Because a model cannot be evaluated in the abstract, we base our evaluation on its application in Optimist, described in the previous Chapter. We discuss issues involved in applying the model in alternative ways in Chapter 6.

In Section 2.5 we discussed several issues concerning the evaluation of decision-assisting systems. We reiterate the main points:

1. Evaluation of decision-aiding systems is difficult, as the benefits of such systems are often intangible. Criteria for measuring this benefit are generally indirect.

2. Appropriate criteria to use (in terms of both what defines success and what is feasible to measure) are domain- and user-dependent, and thus must be chosen to suit the application context of the system.

3. Because of the indirect nature in which benefit is assessed, the use of several criteria is important for proper system evaluation.

Following from these, we first discuss the particular context for evaluation in which we are working and its implications for what criteria are measurable. We then overview the criteria which we use and how they act as indicators of the success or otherwise of the system. Following this we describe details of the evaluation itself.

5.2 The Evaluation Context

5.2.1 Assessing Performance Improvement

Two evaluation criteria which we discussed in Section 2.5 were the accuracies of system and user decisions. In this domain, however, these metrics cannot easily be applied. Because geologists estimate a probability of finding oil, it is impossible to judge the correctness of a single estimate even after the result of drilling is known. For example, a prospect estimated to have 20% chance of containing oil, later found to be dry, does not imply the estimate of 20% was too high. Instead, prediction accuracy can only be measured statistically over a large number of appraised prospects. However, collecting a large enough sample for statistical analysis would take several years, and even then the interpretation of the results would be complicated by the fact that the appraisals were by different geologists using different methods, and that only a small fraction of the appraised wells were actually drilled. A second complication is that there can be a substantial
time lag (between a few months and over a year) between making an appraisal and finding the result of drilling.

Within the exploration area in which Optimist is being used, around 20-40 new prospects will be appraised per year by the team covering it and around 5-10 test wells drilled (these numbers vary year to year, but provide an indication of the sample sizes involved). Note that the the drilled prospects are not necessarily those with highest oil probability; other factors such as the expected volume of oil present and the drilling costs affect the decision whether to drill, as discussed in Section 4.2.1. In addition, some prospects may be drilled as a legal requirement for obtaining exploration rights in an area, and may have very low oil probability. Of the 31 prospects appraised with Optimist, at last information, six have been drilled. One (prospect p8) found oil and two (p18, p19) found evidence that some conditions for oil were present (the three prospects found to be dry were p9, p10, and p20). Such low figures makes statistical validation of risk estimates difficult. Even if the statistics were more reliable, they still would not be that revealing about the contribution made by Optimist, or any other individual factor in appraisal, as we now discuss.

5.2.2 Influences in Risk Appraisal

Optimist is only one of many influences in risk appraisal. The overall quality of risk assessment is influenced by many factors such as the skill of the geologists, the quality of available data, the effectiveness of the exploration group as a team, the quality of other exploration software tools, and the general organisation of the company. Because of the long-term nature of validating risk judgements (as just discussed), it is thus almost impossible to isolate the contribution of any one particular factor in the appraisal process directly. The implications for evaluation are that indirect measurements of Optimist’s utility and performance are needed.

5.2.3 Pragmatic Constraints

Finally, this application domain imposes several practical constraints on evaluation. Geologists are highly trained, their time is valuable, and they are frequently busy. The number specialising in a particular area is also small (eg. the team covering Optimist’s prospects constitutes five people at present), although a large number of other personnel also contribute indirectly to appraisal (eg. those involved in digitising seismics, data collection etc.). These factors, and again the difficulty of assessing the risk judgements objectively, constrain what is testable. For example, large control group experiments are infeasible, due to the limited availability of sufficient numbers of trained experts. These pragmatic constraints are not uncommon in system evaluation, and must be taken into consideration during evaluation design [BWB87, O’K89].

5.3 Evaluation Criteria

Given this evaluation context, we now set out the claims which we wish to establish and the assessment methods which we apply.

Our thesis goal is to develop a practical, computational model of argumentation which can usefully support users in decision-making. As we cannot prove success, it is useful to consider what criteria might indicate that the project had failed:

**System Usage** One indication of failure would be if the system did not achieve regular use. Although this may be for reasons peripheral to the argumentation model (Section 2.5.5), it would also be suggestive that a sufficient level of usability or utility had not been achieved.

A related indicator, which partially takes into account social barriers to usage, would be if usage tailed off after an initial period. This would suggest that although the system was usable it was not sufficiently aiding users to merit the overhead of running the system.
Measures of Variability  A second class of indicators arises from the specific goals of the argumentation model, namely to reduce variability (i.e. random fluctuation) in risk judgements for similar events and to facilitate a useful exchange of information between the system and user.

One indicator that this had not been achieved would be if risk estimates varied greatly without any meaningful reason being supplied. A second would be if the models of opinion within the system, representing typical risk judgements, were ‘unstable’, i.e. their performance fluctuated greatly with time.

These metrics concern the meaningfulness of measures of belief (strengths and qualifiers) to the users of the system. High levels of fluctuation and instability would indicate they had little meaning to users, thus questioning their value as a representation of belief and a useful source of information for users.

We note that we are seeking to reduce variability through the argumentation approach, requiring some form of comparative study to properly evaluate this. We examine issues involved in this shortly.

Acquisition of Knowledge  A third indicator of failure, related to our goal of information exchange, would be if users were unable to supply meaningful backings to the system when requested. Our claim for system utility is based on the construction and manipulation of arguments as a useful resource of information – hence if meaningful backings were rare, this claim would be undermined.

These indicators are all closely related, and test different aspects of the system’s decision support. Metrics of sustained usage most strongly indicate the system is assisting the users in some way; the other metrics test more specific claims about the nature of that assistance. It is of course possible that the system could achieve sustained usage but fail to show evidence of meaningful information being exchanged, hence our decision to use several evaluation methods to assess the quality and impact of that exchange.

Following from the above discussion, we state the main claims against which we evaluate Optimist, and summarise the assessment methods used:

1. The model can be practically and usefully applied.
2. The representation of different opinions (stored in models of opinions) can be established and maintained by non-computer-skilled users.
3. Optimist helps reduce variability in risk judgements.
4. As part of the exchange of information during dialogue, Optimist acquires geological knowledge from users.

The first of these concerns the general utility of the applied system. The three other claims test specific aspects of the argumentation approach which contribute to achieving useful decision support.

We briefly overview the main assessment criteria which we employ:

System Usage: Usage is assessed from the log which the system keeps and from the records of appraisals conducted with Optimist.

Stability of Models: The stability of a model of opinion is assessed by charting how well it matches the owner’s opinion, as expressed in his or her appraisals, with time.

Measures of Variability: To assess change in variability in risk judgements, we perform two comparative studies:
1. Comparisons of variability in risk judgements with and without Optimist, taken over a set of manual appraisals and appraisals within Optimist.

2. Comparison of pairs of appraisals of the same prospects by the same user, each pair containing appraisals made with and without Optimist.

**Information Exchange:** We informally assess whether geological knowledge is being exchanged by examining the backings contained in Optimist’s database.

Finally, we reiterate two points of terminology for clarity:

- **An appraisal** is an argument about the probability of finding oil at a prospect. Its claim is ‘oil is present’ and its qualifier is the estimated probability of oil (ie. that the claim is true). The *process* of appraisal refers to the whole activity of constructing, disputing, and modifying an argument about oil probability.

- In Optimist, as qualifiers are equated with subjective probability estimates, we often refer to the qualifier of a claim $X$ as simply “the probability of $X$”.

## 5.4 System Usage

### 5.4.1 Context and Overview

We first describe the context in which the system is being used in order to provide some background for assessing its usage:

**Potential Users** Although the company we are working with performs exploration work all over the world, Optimist only covers the one area of the North Sea. The exploration team with which we have been working is responsible for the Northern part of this area, and has had on average five members over the past two years. Two other similar sized teams cover other areas of the North Sea and thus could also use the system, but have less awareness of its existence and function.

**Level of Usage** Optimist is not expected to be in continuous use; it takes about an hour to construct an appraisal argument with the system, compared with many weeks’ work in collating data, locating potential sites etc. In addition, demand for usage is expected to vary considerably from month to month, reflecting the slow cyclical processes of exploration (acquiring exploration rights, exploring an area for potential prospects, appraisal, and making drilling decisions).

**Compulsion for Usage** Using the system is not a mandatory part of working practice: the system is available as an aid to geologists who wish to use it.

We first summarise the system’s application status. The system was installed in July 1989 and a logging mechanism added in December 1989. Since the logging mechanism was added, until June 1991, 76 sessions (on 52 separate days) have been run each lasting on average just over 1 hour.

Most significantly, it has achieved use on a sustained basis by 2 geologists (we detail this shortly). Four other geologists have also used it to conduct at least one appraisal – it is important to note that none of these stopped using Optimist for reasons concerning the system itself. A summary of the appraisals and their authors is given in Appendix C. We denote the geologists by the letters A to F. Geologists A and B were originally involved in the system’s development; shortly following its installation, geologist A moved from exploration to field development, and subsequently to the computing group. Geologists B and C are the two regular users. Geologist D moved to Norway in early 1990. User E is a geophysicist, with primary responsibility for
specific aspects of appraisal (e.g. seismic interpretation), and is rarely involved in performing an overall risk appraisal itself. Geologist F only recently started using the system in the last two months (May and June 1991), too short a time to class as ‘sustained usage’.

As of June 1991, the system contains 46 appraisal records of 31 prospects (summarised in Appendix C). (There are also an additional 4 incomplete appraisals recorded; one was by geologist B in December 1989 containing only one warrant. Two were fictitious appraisals by geologist A for demonstration purposes. The fourth is completely empty and created midway through a session, caused by the user hitting the ‘save’ button by mistake.) These figures and the time log exclude test runs made during maintenance of the system, i.e. are all expert appraisals of real-world prospects. They include the 8 appraisals conducted away from the company for the comparative study described later in Section 5.6.3, but these eight are still real-world appraisals under consideration for drilling by the company.

5.4.2 Data Collection

Description

The main method for collecting usage data comes from a log which the system maintains. Additionally, the records of appraisal arguments partially illustrate usage of the system, summarised in Appendix C. The log provides a summary of the users’ actions when using the system, generated by lines being written to a log file of the form:

< UserName, Date, Time, CurrentProspect, CurrentAppraisal, Action >

Only actions recording the main appraisal events are recorded, Action being one of: start/edit appraisal, load appraisal, save appraisal, update appraisal, load ruleset, save ruleset, updated ruleset, exit appraisal summary. Thus the log provides a coarse overview of the system’s activity. Details of the appraisals themselves are not recorded in the logging mechanism, but the completed appraisal arguments themselves are stored in the system’s database (summarised in Appendix C). The time gap between entries in the log during system activity is on average 10 minutes, the longer time gaps occurring during appraisals themselves. During appraisals, usage is evidenced by the creation or updating of a new appraisal in the database, and also the log will record the user breaking to view prospect data during the appraisal (to do this, the user exits and then re-enters the appraisal part of the system as described in Section 4.10.2). Thus there is a possibility that a small part of the ‘usage’ time corresponds to short interruptions (e.g. a phone call), but it is felt that this would not alter the time records substantially.

Three other points should be made. First, the initial entry in the log appears when the user enters or views an appraisal: thus the initial time spent in logging on, loading maps, and data entry (describing the prospect) is not included in the log record. Second, if the user only views the summary of existing appraisal arguments for a prospect but does not enter the part of the system for making appraisals, only a single entry is recorded in the log (exit appraisal summary), appearing as a session of zero time. These sessions are included in the usage plots in Figures 5.1, 5.2 and 5.3. Third, if the user only enters/views the prospect description but does not view or perform an appraisal, then the session will not be recorded in the log (the users report this is very rarely done). In these respects the logging mechanism may underestimate the total usage time.

Finally, although Optimist was installed in July 1989, the logging mechanism was not added until December of that year and thus usage in the first five months is not recorded. Examination of the appraisal records (Appendix C) indicates that the system was used at a comparable level in the two months prior to logging as 9 appraisals were conducted during this time. In the three months before this the system was unused; these three months spanned a summer period, after which a short visit to give on-site training was arranged. This is reflective of the difficulties
Crosses show the number of sessions, blocks show the number of hours usage.

Figure 5.1: Cumulative usage of Optimist.

which can be encountered in encouraging awareness and experimentation with new software
tools.

Figure 5.1 summarises the usage of the Optimist as recorded in the log from December 1989
(month 1) to June 1991 (month 19). Figures 5.2 and 5.3 show the usage plots for the two
main geologists, B and C. The main point of note in these latter figures is the ratio of session
length to session total; for geologist B, the average session is approximately 45 minutes whereas
for geologist C it is approximately 1 hour 20 minutes. This difference is consistent with their
different positions within the company, geologist B being the more senior, busy, and experienced
geologist. This also explains why the session-to-time ratio changes in Figure 5.1, geologist C
starting use of Optimist in month 7.

5.4.3 Discussion

This data provides two strong indicators that Optimist is providing useful support to the users.
First, it has achieved sustained usage by two users. Second, of the geologists who have used
the system, none of these have stopped using it for reasons concerning the system itself. The
number of appraisals in Optimist’s database shows that the majority of prospects appraised
by the two regular users over the past 21 months have been ‘argued about’ with Optimist (as
mentioned earlier, the team they are part of appraises approximately 20-40 prospects per year
in total). This again is a strong indicator of the system’s utility. The fact Optimist is being
used is particularly significant given the generally low success rate of expert systems in the oil
industry (Section 2.2.2) and elsewhere ([Joh84, DH88]).
Crosses show the number of sessions, blocks show the number of hours usage.

Figure 5.2: Cumulative usage of Optimist (Geologist B).

Crosses show the number of sessions, blocks show the number of hours usage.

Figure 5.3: Cumulative usage of Optimist (Geologist C).
5.5 Stability and Maintenance of Models

5.5.1 Introduction

A second indicator we use is that of the stability of the models of opinion within Optimist. We wish to know whether users are able to maintain these models themselves, without the intervention of a trained programmer. The argumentation model imposes two main constraints on models of opinions: they must share the same skeleton warrants, and users can only modify their strengths (not the skeleton warrants themselves). This evaluation thus indicates whether these constraints sufficiently simplify knowledge maintenance while still providing enough flexibility so that users can customise models to reflect their own opinions more accurately.

Because of the geological context in which we are working, evaluation of a model’s stability is a long-term activity. Of the two regular users, geologist B has performed most of his appraisals with a single model (number 15) permitting its stability to be examined. The other models of opinion are not sufficiently developed to properly assess their stability at present (we discuss this shortly).

5.5.2 Metrics

We require some metric of how well a model fits the user’s opinion in order to examine whether users have been able to ‘tune’ their models.

Initially, a model of a user’s opinion starts as a copy of the ‘default’ model, i.e. the set of warrant strengths established during Optimist’s development. During argumentation, the user may modify the strengths in the model \( M \) being used. After construction of and argumentation about \( n \) appraisal arguments (constructed with \( M \)), we denote the state of model \( M \) to be \( M(n) \). The initial model (equal to the default model) is thus \( M(0) \).

A stored argument represents the user’s opinion about a specific prospect. We wish to know how well a user’s tuned model and the default model reflects this opinion. We consider applying the user’s model again to appraise the prospect, and compare the concluded probability with the actual probability in the stored argument. (Note they may not be the same, as the user may have subsequently modified the model or changed warrants in the argument only). We define a model \( M \)’s error on an appraisal as the difference between these two probabilities. After \( n \) appraisals with \( M \), we define the model’s error as the average error of \( M(n) \) on each appraisal. We compare the error for a user’s tuned model \( M(n) \) with that of the original default model \( M(0) \), to see if the user has been able to tune the model to better match his or her opinion. Note that improvement is not guaranteed, as tuning the model for one argument may simultaneously ‘untune’ it for arguments constructed earlier.

Secondly, we compare the degree of tuning of different models within Optimist, defined as the proportion of warrants whose strengths differ from the strengths in the default model.

5.5.3 Results

Table 5.1 shows which warrant strengths have been changed in the different, tuned models in Optimist, and the degree of tuning.

Figure 5.4 shows the evolution of the model which geologist B has mainly used (number 15), comparing its average error with that of the default model on the prospects appraised with it. These sixteen prospects include the eight used in the comparative study described shortly (Section 5.6.3). As can be seen, the tuned model has lower error than the default model. We also note that geologists A and B were responsible for the original default model, constructed through a number of trial runs with Optimist on real-world prospects during system development (Section 4.10.1). Thus the fact that further tuning was possible on a model already partially matching his opinion is significant.
This plot illustrates the maintenance of one model of opinion within Optimist (for geologist B). The upper line (crosses) shows the average error (difference of oil probabilities \( \times 100 \)) of the default model \( M(0) \) as more prospects are appraised, the lower line (blocks) showing that for the tuned model \( M(n) \) with increasing \( n \). The figure shows that after model modification, the tuned model \( M(n) \) is a better performer than the initial default model.

Figure 5.4: Reduction of error in a user’s model of opinion.
<table>
<thead>
<tr>
<th>Warrant → Model No ↓</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>10</th>
<th>11</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td>90%</td>
<td>94%</td>
<td>60%</td>
<td>70%</td>
<td>59%</td>
<td>87%</td>
<td>24%</td>
<td>84%</td>
<td>17%</td>
</tr>
<tr>
<td>10 (D)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11 (B)</td>
<td>85%</td>
<td>51%</td>
<td>25%</td>
<td>85%</td>
<td></td>
<td>50%</td>
<td>33%</td>
<td>30%</td>
<td>10%</td>
</tr>
<tr>
<td>12 (E)</td>
<td>85%</td>
<td>51%</td>
<td>25%</td>
<td>85%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14 (B)</td>
<td>85%</td>
<td>70%</td>
<td>50%</td>
<td>85%</td>
<td>100%</td>
<td>80%</td>
<td></td>
<td>50%</td>
<td>33%</td>
</tr>
<tr>
<td>15 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32 (A)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33 (A)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34 (A)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35 (A)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36 (A)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37 (A)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>39 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>45 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>46 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>47 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>48 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>49 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>51 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>52 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>53 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>54 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>55 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>56 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>57 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>58 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>59 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>61 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>62 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>63 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>65 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>66 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>67 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>68 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>69 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>70 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>71 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>72 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>73 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>74 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>75 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>76 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>77 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>78 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>79 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>81 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>82 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>83 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>84 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>85 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>86 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>87 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>88 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>89 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>90 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>91 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>92 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>93 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>94 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>95 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>96 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>97 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>98 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>99 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100 (B)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

No figure implies the default strength is taken. Only the subset of warrants (16 out of 31) whose strengths were altered in at least one model are shown. The degree of tuning is the proportion of the warrant set whose strengths differ from the default strength. The owner is shown in brackets.

Table 5.1: Modification of Warrant Strengths for Different Users’ Models.
<table>
<thead>
<tr>
<th>Appraisal no.</th>
<th>Oil probability by re-applying:</th>
<th>Target:</th>
<th>Would yield profit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Default User's Tuned</td>
<td>User's</td>
<td>(Default</td>
</tr>
<tr>
<td></td>
<td>Model Model M(16) appraisal</td>
<td></td>
<td>model</td>
</tr>
<tr>
<td>10</td>
<td>55%</td>
<td>55%</td>
<td>52%</td>
</tr>
<tr>
<td>18</td>
<td>14%</td>
<td>21%</td>
<td>11%</td>
</tr>
<tr>
<td>19</td>
<td>19%</td>
<td>22%</td>
<td>14%</td>
</tr>
<tr>
<td>22</td>
<td>19%</td>
<td>18%</td>
<td>11%</td>
</tr>
<tr>
<td>24</td>
<td>2%</td>
<td>10%</td>
<td>7%</td>
</tr>
<tr>
<td>16</td>
<td>10%</td>
<td>8%</td>
<td>2%</td>
</tr>
<tr>
<td>36</td>
<td>4%</td>
<td>7%</td>
<td>7%</td>
</tr>
<tr>
<td>51</td>
<td>25%</td>
<td>13%</td>
<td>11%</td>
</tr>
<tr>
<td>52</td>
<td>14%</td>
<td>14%</td>
<td>14%</td>
</tr>
<tr>
<td>4</td>
<td>59%</td>
<td>56%</td>
<td>56%</td>
</tr>
<tr>
<td>53</td>
<td>20%</td>
<td>19%</td>
<td>20%</td>
</tr>
<tr>
<td>54</td>
<td>15%</td>
<td>6%</td>
<td>6%</td>
</tr>
<tr>
<td>55</td>
<td>3%</td>
<td>6%</td>
<td>8%</td>
</tr>
<tr>
<td>56</td>
<td>6%</td>
<td>19%</td>
<td>19%</td>
</tr>
<tr>
<td>57</td>
<td>1%</td>
<td>13%</td>
<td>13%</td>
</tr>
<tr>
<td>58</td>
<td>27%</td>
<td>27%</td>
<td>29%</td>
</tr>
<tr>
<td>AvErr=5.8%</td>
<td>AvErr=2.6%</td>
<td></td>
<td>168.168</td>
</tr>
</tbody>
</table>

Geologist B's actual opinion, the target, is taken as the conclusion of the argument which he concluded with Optimist. AvErr is the error of the model compared with the target probability, averaged over the 16 appraisals (as defined in Section 5.5.2). The profit yield is calculated using a simple profit-and-loss model described in Appendix B.

Table 5.2: The accuracy of the default and geologist B's tuned model after 16 appraisals.

Table 5.2 compares the conclusions of arguments constructed by re-applying the final (current) state $M(16)$ of the model $M$ and the default model on the 16 prospects originally appraised with $M$. Again, the tuned model has lower error than the default model. For interest, we apply a simple profit-and-loss model to see how the two sets of oil probability estimates affect profit. The model is described in Appendix B and the profits shown in Table 5.2 (all multiplied by 1000 for legibility). This shows the tuned model would yield a greater profit than the default model if applied to these drilling decisions. (The profits shown are with profit-and-loss parameters $k = c = 1$, but the tuned model reaps higher profit for any values of these parameters).

The other regular user’s appraisals were conducted with several different models (numbers 52, 53, and 58). As can be seen from Table 5.1, these models are almost unchanged from the default model $M(0)$. Examination of his appraisals shows he preferred to alter warrant strengths within individual appraisal arguments but not propagate those changes into the models of his opinion. As a result, the models for this user are virtually untuned (at present).

Table 5.1 shows the degree of tuning of models is generally low (at most, only 39% of the warrant set was changed). We discuss this below.

### 5.5.4 Conclusions and Discussion

The main conclusion is that, in the case where stability can be tested, the model of opinion was stable, i.e., controllable by the user. This is most clearly shown in Figure 5.4, which shows that geologist B was able to tune Optimist’s model of his opinion to match his own more closely than the original default model.

This provides several important indicators for our evaluation. First, the fact the model could
be controlled supports our claim for the practicality of our argumentation model. Second, the stability and improved performance of the user’s model is an indicator that he was able to relate it to his own opinion. This is an important conclusion regarding the system’s role as an information resource.

The data also indicates that the user was able to adequately perform maintenance of the system’s knowledge of his opinions himself, without the intervention of a trained programmer. This indicates the value of the argumentation approach as assisting in knowledge acquisition. However, we also note that larger changes (eg. changing the warrants themselves) still required a programmer: in the 21 months Optimist has been used, this has been necessary twice for minor changes (adding 3 new values for an attribute, and merging 8 old warrants into 5). We detail these changes in Section 6.4.3, but our point here is that users can perform the bulk of the knowledge maintenance themselves. Because few of the oil industry expert systems achieved regular use (Section 2.2), and none of those reported maintenance requirements for either the overall software or the knowledge-base in particular, comparisons are difficult. (Even if figures were available, fair comparisons would be impossible due to the differing size, complexity, and subject matter of the encoded knowledge). However, we note that the programmer-requiring maintenance is low in absolute terms, that the demand on the programmer has been reduced by the users’ ability to participate in maintenance themselves.

While our claim that users can tune models is supported, the related question of whether they do in practice must also be asked. The lack of model tuning for the other regular user reveals two ‘loop-holes’ by which users can avoid tuning models: First, tuning can be avoided by only altering warrant strengths in specific arguments and not in the models used to derive them. Second, users tend only to tune warrants which they encounter during appraisals and with which they disagree. We discuss this further in the section on future work in Chapter 6.

5.6 Measures of Variability

5.6.1 Overview

A third claim we wish to test is that Optimist will help reduce variability of risk judgements among appraisals. By variability, we refer to fluctuations in the way the ‘same’ risk judgement is expressed in appraisals (also referred to as ‘reliability’ elsewhere in the literature [WB83]). Variability reduces the value of recorded appraisals as meaningful representations of decision-making. In the extreme case where the expressed judgements are completely random, their value as information to the user is lost.

Variability is particularly difficult to evaluate because of the problem of identifying ‘similar’ risk judgements. We address this in two ways. First, we assess variability among sets of appraisals described using the same description language. Although variability is to be expected (apparently similar situations may in fact have differences inexpressible in the language), we are interested in comparing variability with and without Optimist. We perform two comparative studies for this purpose. Following this, in Section 5.7, we examine backings of warrants with large differences in strength to assess whether the differences were caused by variability in judgements about the ‘same’ event or whether the backings distinguish the events as different in a geologically meaningful way.

The two comparative studies we perform here are as follows:

1. Variability among manual appraisals is compared with variability among those conducted with Optimist.

2. The variability between eight pairs of appraisals, each pair consisting of appraisals with and without Optimist of the same prospect, is examined.
We make two important points. First, as discussed in Chapter 2, comparative studies are particularly important in evaluation: absolute measures of performance alone are difficult to interpret without any form of control against which to compare. Second, the acquisition of such control data is very difficult within the pragmatic constraints of our application (Section 5.2.3). As a result, the two studies below have had to be carefully designed to work within these constraints. Although the comparisons are thus limited and tentative, the fact that feasible comparative studies can be designed at all is a point of note.

5.6.2 Variability of Manual and Optimist Appraisals

Metrics

The first study we perform compares variability of risk judgements expressed in manual appraisals with variability in Optimist-assisted appraisals. Some metric of variability within a set of appraisals is thus required. For brevity, we simply use the word ‘variability’ below to refer to ‘variability in expressed risk judgements’.

One metric would be to take pairs of appraisal arguments, and for every skeleton warrant they have in common, compare the strengths of that warrant in both arguments. If the strengths are the same, we say there is no variability in the judgements. Otherwise, we can define the degree of variability as the difference in their strengths. The variability of the two arguments can be defined as the average variability among skeleton warrants they have in common. The variability of the set of arguments can be defined as the average variability per argument pair, averaged over all possible pairs (for $N$ arguments, there are $(N/2) \times (N - 1)$ possible pairs).

There is an immediate difficulty preventing us applying this metric to manually appraised prospects. In manual appraisals, there are no equivalents to warrant strengths which are recorded. Instead, only the overall probabilities of the six risk components are recorded, (trap presence/effectiveness, reservoir presence/communication, source presence/maturity), corresponding to the combination of separate warrants in Optimist. Because of this we cannot simply compare individual (equivalents of) warrant strengths among manual appraisals, and some alternative metric must be used.

To overcome this problem, two alternative metrics can be devised which consider variability between the combined strengths of a set of warrants in two arguments. These are as follows.

1. Prospects with identical properties affecting oil probability:

If two prospects have identical properties affecting oil probability, then we would also hope that this probability would be the same in their risk appraisals. If so, we say there is no variability among the risk judgements in the two appraisals. If not, we define the degree of variability $V_{ident}$ as the difference in the two oil probabilities $p_1$ and $p_2$:

$$V_{ident} = |p_1 - p_2|$$  (5.1)

To identify the properties affecting oil probability, we use Optimist’s set of skeleton warrants. This set serves as an approximate model of which properties (the grounds) determine oil probability. Note that these properties might be facts supplied by the user (eg, the trap is dip closure) or relationships between these facts (eg, the fault age is less than the seal age).

To denote the properties affecting oil probability we simply list the skeleton warrants whose grounds are satisfied and which conclude oil is present. For a pair of appraisals where these sets are identical, we can assess their variability using Equation 5.1. If the sets are not identical, we cannot say anything about the variability of the risk judgements.

In practice, some variability is to be expected as Optimist’s warrants only approximately model the relation between a prospect’s description and oil likelihood – what are apparently identical situations being assessed may in fact be different. However, as discussed earlier, we are not so much interested in whether there is variability (we expect there to be some) but
in comparing the degree of variability among manual and among Optimist appraisals. The approximation that the language always discerns differing risk situations applies equally to variability among Optimist appraisals and manual appraisals, and thus does not affect the comparison. Our claim is that the explicit use of a model of appraisal in Optimist, combined with argumentation functions alerting users to previous appraisal judgements, will reduce variability of risk judgements among appraisals.

2. Appraisals where one prospect is more likely to have oil than another:
The previous metric can only be applied when two prospects have identical properties affecting oil. The rarity of this occurring makes it of limited value (in our analysis in Table 5.3, only 2 of the 105 pairs of manual appraisals meet this condition). To devise a more generally applicable metric, we exploit the fact that, for some appraisal pairs, we can identify that one prospect should have a higher (or equal) probability of oil than the other in the pair. This can be done because, for some properties, a clear ordering of ‘riskiness’ can be given. This was only done for properties with an unambiguous, graduated scale of risk (eg. ‘reservoir mapping’ has values ‘perfect’, ‘good’, and ‘uncertain’). For properties which could not be clearly ordered in this way, including those requiring geological expertise to decide an appropriate ordering, we still demand both prospects have identical occurrences of them for the comparison to be made.

If, for a pair of prospects P1 and P2, all the properties of P1 affecting oil are either identical or ‘less risky’ than those of P2, then the appraised probability of oil at P1 should be greater than or equal to that at P2. If so, we say there is no evidence of variability in the risk judgements. If not, then some variability is evidenced: either the user has under-estimated the probability of P1 and/or over-estimated that of P2 (again we note this is apparent variability according to the description language – there may be good but inexpressible reasons for the difference). In the latter case, we define the degree of variability $V_{\text{better}}$ as the difference in their appraised probabilities. Thus:

$$V_{\text{better}} = \begin{cases} 
0 & \text{if } p_1 \geq p_2 \\
 p_2 - p_1 & \text{if } p_1 < p_2 
\end{cases} \quad (5.2)$$

where $p_1$ and $p_2$ are the appraised probabilities of oil for P1 and P2.

For example, if the warrants (hence properties affecting oil) which would apply for P1 are \{sw1, sw3\}, and for P2 \{sw1, sw5\}, and the grounds of sw3 are unambiguously less risky those of sw5, then we expect the probability of oil at P1 to be greater than or equal to that at P2. However if the skeleton warrants applying at P2 were, for example, \{sw1, sw5, sw9\}, or \{sw2, sw5\}, we would be unable to say anything about the variability in risk judgements between the two appraisals.

Method

We compared variability among 15 manual prospect appraisals and among the 36 up-to-date Optimist appraisals (ie. excluding those where there was also a more recent appraisal for the same prospect by the same user in Optimist). All the prospects were described using the same (Optimist’s) prospect description language. The 15 manual appraisals were supplied by geologists B (8 supplied) and F (7 supplied), who filled in a specially designed questionnaire for each prospect covering first a description of the prospect using Optimist’s description language, and second the risk appraisal itself from company records (the probabilities of the six components of risk and overall oil probability). For the prospects supplied by geologist B, a second geologist was also involved in making the recorded appraisals. The 8 prospects supplied by geologist B were later appraised separately with Optimist and are included in the 36 Optimist appraisals used (we compare these 8 specific appraisals in more detail shortly).

Of the six components, we examine variability of the probabilities for trap presence, trap effectiveness, and their product (total trap). We do not repeat this analysis for reservoir or
<table>
<thead>
<tr>
<th></th>
<th>Manual Appraisals</th>
<th>Appraisals with Optimist</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number $N$ of appraisals used</td>
<td>15</td>
<td>36</td>
</tr>
<tr>
<td>No. $N_p$ of possible appraisal pairs ($= N/2 \times (N - 1)$)</td>
<td>105</td>
<td>630</td>
</tr>
</tbody>
</table>

**Overall Trap:**
Of the $N_p$ pairs, no. $N_{ident}$ having identical properties affecting the probability of ‘trap’

**Av. variability** among these $N_{ident}$ pairs (Eqn 5.1)

|                                | 2                  | 16                        |
|                                | 24.4%              | 22.4%                     |

Of the $N_p$ pairs, no. $N_{better}$ having one prospect identifiably more likely to have a trap than the other
Of these $N_{better}$, no. $N_{inc}$ which are inconsistent (the prob. of trap in the first prospect was lower)

**Av. variability** among these $N_{better}$ pairs (Eqn 5.2)

|                                | 13                 | 224                       |
|                                | 13 (23%)           | 224 (9%)                  |

**Trap Presence:**
Of the $N_p$ pairs, no. $N_{better}$ having one prospect identifiably more likely to have ‘trap presence’ than the other
Of these $N_{better}$, no. $N_{inc}$ which are inconsistent

**Av. variability** among these $N_{better}$ pairs (Eqn 5.2)

|                                | 13                 | 224                       |
|                                | 3 (23%)            | 18 (9%)                   |

**Trap Effectiveness:**
Of the $N_p$ pairs, no. $N_{better}$ having one prospect identifiably more likely to have ‘trap effective’ than the other
Of these $N_{better}$, $N_{inc}$ are inconsistent

**Av. variability** among these $N_{better}$ pairs (Eqn 5.2)

|                                | 13                 | 224                       |
|                                | 5 (38%)            | 23 (10%)                  |

|                                | 10.9%              | 2.5%                      |
|                                | 8.4%               | 0.5%                      |

|                                | 10.5%              | 3.7%                      |

Table 5.3: Variability of Manual and Optimist Appraisals.

source, as their probabilities are more specific to the local prospect conditions (eg, the use of special warrants in Optimist), and thus comparisons would be less valid due to the differing geographical distributions of manual and Optimist-appraised prospects.

To identify which prospects have the same properties affecting trap presence, we locate prospects sharing the same set of properties affecting the total trap (ie. compare the sets of applicable warrants concluding both ‘trap is present’ and ‘trap is effective’). We do this because there is some ambiguity about whether a property affecting the trap should contribute to trap presence, or effectiveness, or both, so we include all possible trap properties to ensure all trap presence properties are shared (similarly for trap effectiveness).

**Results and Conclusions**
The average variabilities in the set of manual and Optimist-appraised prospects are shown in Table 5.3.

It should first be noted that these comparisons are tentative, and complicated by many factors: both sets of prospects have differing geographical distributions (although all from the same general area), were appraised by different geologists at different times, and the sample sizes are small. In addition, the metrics used only measure variability in a limited way (only examining appraisal pairs where the ‘same’ or ‘better-than’ relation holds). However, from the comparison
shown in Table 5.3, with the same tentativeness, we can conclude that Optimist appears to be helping reduce variability among appraisals, based on the assumption that the samples of manually and Optimist-appraised prospects are reasonably typical. For all the metrics of variability, Optimist’s appraisals have less variance with each other than the manual appraisals have among themselves. For example, of the 224 pairs where the better-than relation holds in Optimist, the average variability per pair for trap presence was less than 1%. We discuss the significance of this result in more detail shortly, after describing the second comparative study.

5.6.3 Comparison of Appraisals with and without Optimist

Method and Metrics

Our second method for evaluating variability is to compare appraisals conducted with and without Optimist for the same prospects. For each prospect we wish to know if it is being appraised in the same way both with and without Optimist, or whether Optimist is affecting the appraisal and if so, in what way.

Eight of the manual appraisals were supplied by geologist B. To perform this comparison, he was later asked to perform the same appraisals using Optimist. As described in Section 5.4, the appraisals made with Optimist were performed away from the company at the Turing Institute (using an identical, up-to-date copy of Optimist’s database) by geologist B working alone i.e. with no intervention from ourselves, between the 21st and 22nd of January 1991 (appraisals 51 to 58 in Appendix C). We reiterate that these eight are, like the other appraisals made with Optimist, also real-world appraisals under consideration by the company, i.e. are not fictitious prospects.

The conditions of appraisal are thus similar (same prospect, same geologist), although some differences still remain. Firstly, there was a substantial time gap (approximately 12 months) between the manual and Optimist appraisals; this ensured that he was not simply reciting the manual appraisals from memory, but additional information in the intervening time may have altered his judgement. Secondly, another geologist also contributed to the manual appraisals, reducing the similarity in appraisal conditions. The comparison of the probabilities for the six components of the appraisal are shown in Table 5.4.

Results, Conclusions, and Discussion

Because of the differences in appraisal conditions just mentioned, care is needed to interpret this data. To help distinguish effects of Optimist on appraisals from effects due to these different conditions of appraisal, we additionally interviewed the geologist to discuss the differences.

Table 5.4 shows substantial differences in the probabilities concluded with and without Optimist, but the absolute differences in the overall probabilities are less than those for the six components of risk.

Discussion with the geologist provided evidence that Optimist is mainly affecting the breakdown of the overall oil probability into its six components, while leaving the overall probability relatively unchanged. The geologist considered the differences in overall probability to be mainly caused by the different appraisal conditions rather than Optimist’s influence. This is supported by his comments on the manual conclusions, prior to being reminded of the conclusions he found with Optimist. He offered comments on 5 of the 8 prospects, and in all but one of these cases he made comments suggesting that the Optimist-derived conclusion correctly represented his current opinion about the prospect (e.g. when shown the manual conclusion of 10% in appraisal 54, he reported that it sounded a bit high and that 6%-8% would be more appropriate, consistent with his Optimist-derived probability). This suggests that the use of Optimist is mainly reflecting rather than affecting the geologist’s overall conclusion. A second piece of evidence supporting this is that geologists using Optimist will often reach a conclusion and then return to
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>51 Manual with Optimist</td>
<td>90</td>
<td>33</td>
<td>69</td>
<td>69</td>
<td>84</td>
<td>84</td>
<td>10</td>
</tr>
<tr>
<td>difference</td>
<td>5</td>
<td>16</td>
<td>21</td>
<td>21</td>
<td>11</td>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td>52 Manual with Optimist</td>
<td>80</td>
<td>90</td>
<td>10</td>
<td>100</td>
<td>90</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>difference</td>
<td>38</td>
<td>10</td>
<td>35</td>
<td>9</td>
<td>5</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>53 Manual with Optimist</td>
<td>47</td>
<td>30</td>
<td>72</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>10</td>
</tr>
<tr>
<td>difference</td>
<td>38</td>
<td>20</td>
<td>24</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>54 Manual with Optimist</td>
<td>50</td>
<td>60</td>
<td>50</td>
<td>95</td>
<td>70</td>
<td>100</td>
<td>10</td>
</tr>
<tr>
<td>difference</td>
<td>20</td>
<td>27</td>
<td>12</td>
<td>20</td>
<td>25</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>55 Manual with Optimist</td>
<td>47</td>
<td>30</td>
<td>76</td>
<td>95</td>
<td>100</td>
<td>100</td>
<td>10</td>
</tr>
<tr>
<td>difference</td>
<td>38</td>
<td>9</td>
<td>9</td>
<td>20</td>
<td>5</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>56 Manual with Optimist</td>
<td>47</td>
<td>33</td>
<td>86</td>
<td>83</td>
<td>95</td>
<td>100</td>
<td>13</td>
</tr>
<tr>
<td>difference</td>
<td>38</td>
<td>3</td>
<td>4</td>
<td>17</td>
<td>5</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>57 Manual with Optimist</td>
<td>35</td>
<td>35</td>
<td>90</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>11</td>
</tr>
<tr>
<td>difference</td>
<td>50</td>
<td>9</td>
<td>10</td>
<td>21</td>
<td>5</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>58 Manual with Optimist</td>
<td>42</td>
<td>100</td>
<td>72</td>
<td>95</td>
<td>100</td>
<td>100</td>
<td>29</td>
</tr>
<tr>
<td>difference</td>
<td>42</td>
<td>16</td>
<td>16</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>

The numbers express probabilities as percentages (eg, 43 means a probability of 0.43 or 43%). The probabilities shown are for the six appraisal components, and the overall probability of oil (the product of the six).

Table 5.4: Comparison of appraisals of 8 prospects with and without Optimist.

‘tweak’ one or two probabilities to obtain the answer they wanted if they felt the original answer was wrong. This reflects that a degree of variability remains in the warrant strengths which the users can exploit to express their own opinions. This and other geologists have reported that this ‘tweaking’ behaviour is also common in the manual appraisal process: our comparative study in Section 5.6.2 indicates that Optimist is helping to reduce the variability that may result.

A second feature in Table 5.4 is the large differences in the breakdown of oil probability among the six components. Additionally, the direction of the difference varies within appraisals (ie, whether the manual probability is greater or less than the Optimist-derived probability). Unlike the overall probability, the geologist attributed these differences mainly to Optimist’s influence, in particular because Optimist imposes a fixed, clearly defined set of relationships between a prospect’s description and the six appraisal components. The geologist attributed some variations in the manual prospect’s probability breakdown to ambiguity about which prospect properties affected which component. For example, the large difference in probability of trap presence in appraisal 54 was attributed to the manual appraisal treating a particular prospect property as affecting both trap presence and effectiveness, whereas Optimist considered it to affect effectiveness alone. “If you talk to different people,” the geologist reports, “some people might say well, it [a twin fault feature] is not a problem of trap [presence] definition, it’s
only a problem of trap effectiveness, and some other people might say it’s only a problem of definition and it’s not a problem of effectiveness.” In addition, there were smaller differences due to his own opinion now differing from the earlier manual appraisals.

This ambiguity problem is reduced with Optimist, as its warrants explicitly state how properties affect the six appraisal components and thus enforces a consistent allocation of risk among components. This consistency is essential if the probabilities for the six components are to have meaning to other geologists. From the data in Table 5.4 and the discussions with the geologist, we conclude Optimist has mainly influenced how the oil probability is distributed among the six components, while influencing the final oil probability to a lesser extent. The evaluation in Section 5.6.2 suggests this redistribution has produced appraisals with less variability in expression of risk judgement, thus helping other geologists to understand the appraising geologists’ reasoning and compare appraisals. This is a valuable achievement of the system.

We also note that ambiguity in terminology is only one cause of disagreement among users. Other causes include different geological knowledge, or simply differences in judging or expressing probability, and thus we would not expect removal of the ambiguity problem to result in perfect agreement among users. We analyse these additional causes of disagreement shortly in Sections 5.7 and 5.8.

5.6.4 Summary

To summarise the main points:

1. The comparative studies indicate that Optimist is helping to reduce the variability in expressed risk judgements.

2. From the second study and interview, part of that reduction is caused by reduced ambiguity about how risk should be attributed to the six appraisal components. This is achieved by the warrant set helping to clarify the definition of terms and the structure of an appraisal.

We also make some final comments concerning the view of recorded appraisals as expressing rather than deriving an answer; this considers appraisal as the two processes of oil probability assessment and the translation of the reasoning in that assessment into a numeric form (six probabilities). Compton and Jansen [CJ89, CHQ+89] have advocated a similar view of knowledge acquisition, regarding it as a post hoc justification of inarticulatable reasoning. Although numeric expressions of belief are known to be somewhat unnatural for experts, some written, quantitative expression of the reasoning behind appraisals is essential for appraisals to be compared and understood by others. From the presented data, Optimist appears to be helping to reduce the variability in those expressions of reasoning, with consequent benefits for users in understanding and comparing appraisal arguments. This contributes to our claim of the value of computer argumentation in decision support.

5.7 Acquisition of Extra Knowledge

5.7.1 Introduction

Finally, we examine the ability of geologists to justify the changes they make to warrant strengths. This again relates to the issues of variability and value of information in an argument: a large difference in strength could be caused either by imprecision in expressing belief or genuine geological differences in the two situations. To evaluate this, we inspect the backings to examine whether users were able to justify their changes with geological information.

We inspect the backings in two ways. First, we examine what proportion of backings contain geological information (as opposed to tautological or meaningless information). Second, we
<table>
<thead>
<tr>
<th>Category</th>
<th>No. of Backings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geological Knowledge (general)</td>
<td>162 (68%)</td>
</tr>
<tr>
<td>Geological Knowledge (precedents)</td>
<td>46 (19%)</td>
</tr>
<tr>
<td>Tautological</td>
<td>17 (7%)</td>
</tr>
<tr>
<td>Absent</td>
<td>15 (6%)</td>
</tr>
</tbody>
</table>

Table 5.5: Distribution of backings among categories.

informally examine warrants where substantial variation in strengths exist, to see if users were able to justify those changes in a geologically meaningful way.

5.7.2 Categorisation of Backings

A full list of the backings in Optimist is given in Appendix D. (Names have been changed to preserve company confidentiality). The backings are deliberately concise as users are allowed only one line of text. (They have declined to have this maximum length extended, as they consider one line sufficient and the conciseness useful). A number of categories of (not mutually exclusive) backings can be identified:

**Geological Knowledge (general):** Backings which express general geological knowledge about the prospect under examination or the geographical area.

**Geological Knowledge (precedents):** Geological backings based on evidence from specific, named prospects, oil sources etc.

**Tautological:** where the expert has simply reiterated his judgement (eg. ‘risk of 50 too low’).

**Absent:** where the expert had difficulty in formulating a meaningful justification (eg. ‘I don’t know’).

We separate geological backings into ‘general’ and ‘precedent’ for interest to see how often precedent backings are used (we claim that precedent backings are important in geology). Apart from the very early appraisals, Optimist always insists the user give some backing should he or she override the system’s argument.

Table 5.5 summarises the number of backings in the different categories (repeated backings are only counted once). This data shows that, in most cases, users were able to supply extra geological knowledge to justify their opinions during appraisals. Only 13% of the backings were in the non-useful categories ‘tautological’ or ‘absent’. Thus this data supports our claim that, through argumentation, Optimist is acquiring extra useful information from the users.

5.7.3 Justification for Strength Differences

To further explore this issue, Table 5.6 provides a second illustration of geologists’ use of backings in Optimist. Here, the backings attached to warrants with substantial variation in strength are shown alongside that strength. Again, the proportion of geologically meaningful backings are consistent with the figures in Table 5.5, with uninformative backings being infrequent (eg. “Initial value too high”, “regional risk”). This shows that users are in the majority of cases able to justify the changes in warrant strengths.

5.7.4 Summary

The main conclusion here is that users were able, in most cases, to supply geologically meaningful backings for changes to warrant strengths. This again suggests that Optimist is acquiring useful
This table shows the backings for warrants where there was substantial variation in the warrant strength ($s_{\text{max}} - s_{\text{min}} > 40\%$). Italics show the default warrant strength. For very early appraisals, it was not compulsory to provide a backing hence occasional backings are missing above.

Table 5.6: A sample of the backings used in appraisals in Optimist.

<table>
<thead>
<tr>
<th>Warrant</th>
<th>Strength</th>
<th>Backing</th>
<th>In appraisal argument nos.</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>50</td>
<td>reservoir not well defined</td>
<td>(9)</td>
</tr>
<tr>
<td></td>
<td>51</td>
<td>Initial value too high</td>
<td>(19, 26, 68, 69, 78)</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>Seismic OK, corellation uncertain</td>
<td>(54)</td>
</tr>
<tr>
<td></td>
<td>74</td>
<td></td>
<td>(77)</td>
</tr>
<tr>
<td>11</td>
<td>94</td>
<td>other Fields in the region have faulting&lt;seal age.</td>
<td>(6, 8, 32)</td>
</tr>
<tr>
<td></td>
<td>95</td>
<td>always risk with downturned prospects</td>
<td>(67)</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>facies sealed by base moravican</td>
<td>(15)</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>fault is pre-migration</td>
<td>(9, 22, 23, 24)</td>
</tr>
<tr>
<td>13</td>
<td>50</td>
<td>major faults break surface</td>
<td>(13, 16, 17, 18, 19, 51, 53, 55, 56, 57, 74, 87)</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>regional risk</td>
<td>(8)</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>Most Madeline fields are tilted normal fault blocks</td>
<td>(9, 16, 18, 19, 23, 24, 51, 79, 83, 84, 85)</td>
</tr>
<tr>
<td></td>
<td>87</td>
<td>local discoveries suggest favian fault seal</td>
<td>(55)</td>
</tr>
<tr>
<td></td>
<td>95</td>
<td>regional data suggests cross-fault seal</td>
<td>(88)</td>
</tr>
<tr>
<td>15</td>
<td>30</td>
<td>downturned closures are leaky</td>
<td>(6, 32)</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>Fault seal evidence from Baries/Washton/ Berk</td>
<td>(87)</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>no regional precedent for this seal being effective</td>
<td>(88)</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>no regional precedent to suggest seal is effective</td>
<td>(13, 15, 17)</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>regional risk</td>
<td>(68)</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>Downthrowing against Moravian basement.</td>
<td>(74)</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>Upthrows G.Hub. claystones.</td>
<td>(74)</td>
</tr>
<tr>
<td></td>
<td>75</td>
<td>Downthrowing against Moravian Basement.</td>
<td>(78)</td>
</tr>
<tr>
<td></td>
<td>85</td>
<td>consider this seal to work in region, ie Kivia Field.</td>
<td>(67)</td>
</tr>
<tr>
<td>17</td>
<td>10</td>
<td>facies didn’t seal in df/s-s</td>
<td>(16)</td>
</tr>
<tr>
<td></td>
<td>33</td>
<td>Facies works at Washton/Shalian</td>
<td>(54)</td>
</tr>
<tr>
<td></td>
<td>33</td>
<td>sand pinching out into shale and there is an element of dip closure, thus reducing risk.</td>
<td>(67)</td>
</tr>
<tr>
<td></td>
<td>61</td>
<td>Facies = closure is a comb. of sand pinchout &amp; dip.</td>
<td>(32, 34)</td>
</tr>
<tr>
<td></td>
<td>65</td>
<td>Closure has element of dip so improving eff.</td>
<td>(68)</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>Facies applies to upside case/Dip closure applies in the most likely case.</td>
<td>(78)</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>Guillar has facies closure 25 kms away</td>
<td>(69, 77)</td>
</tr>
<tr>
<td></td>
<td>81</td>
<td>evidence from garavian sands</td>
<td>(7)</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>I’ve changed this trap direction to be a trap.</td>
<td>(5)</td>
</tr>
<tr>
<td>25</td>
<td>74</td>
<td>more than one potential source</td>
<td>(6)</td>
</tr>
</tbody>
</table>

This table shows the backings for warrants where there was substantial variation in the warrant strength ($s_{\text{max}} - s_{\text{min}} > 40\%$). Italics show the default warrant strength. For very early appraisals, it was not compulsory to provide a backing hence occasional backings are missing above.
information from the users, supporting the claim that the system's utility is based on its its value as an information acquirer and resource.

Finally, we do not claim that users necessarily form backings easily. Users report that sometimes it requires some thought and reflection to provide backings, but that this is a valuable activity stimulated by using Optimist. Again the importance of this task of evoking backings should be emphasised, as it constitutes an important part of the information exchange during argumentation.

5.8 Differences of Opinion

Finally, we make some brief analysis of differences of opinion within Optimist in order to illustrate the use of backings and argument analysis in locating the cause(s) of disagreement. Of the 31 appraised prospects, there are 5 which have been appraised by more than one user and 10 where the appraisal has been updated and stored as a new appraisal (as opposed to overwriting the old appraisal).

Table 5.7, an extraction from Appendix C, summarises the 5 prospects with appraisals by more than one user. The boxes highlight where warrant strengths differ between the two users. In all cases, the second appraisal in the pair was made at a later date than the first. For the first and last pairs (3-4 and 15-53), there is a time gap of approximately 15 months between the dates members of the pair were most recently updated. For the others, the time difference was within two days of each other.

There are two sources of difference: differences in warrant strength and (for pairs 7-10 and 6-8) also differences in the prospect description itself. Both these sources relate to differing geological opinion, the former being expressed in backings and the latter in the prospect’s description. We briefly summarise the causes of difference in each case. For appraisal pairs 3-4 and 15-53 new data had become available in the intervening 15 months from drilled wells. The major difference in 3-4 is warrant S4, backed by “wx/91-s dry hole on prospect edge” causing reduced confidence in communication with an oil source. The major differences in 15-53 are warrants 15 and S4, backed in appraisal 53 by “Fault seal evidence from Baries/Washton/Berx” and ‘Oil present in ab/3-d & we/2b-z” respectively, again describing newly available data. Appraisals 7-10 differ in the interpretation of the trap from seismic, described in the prospect description. Appraisals 6-8 differ in mapping confidence (again a prospect description attribute), and in warrants 13, 25 and S4. Again these warrants are backed with geological justifications (“major faults break surface”, “more than one potential source” and “reservoir is juxtaposed against source”). Appraisals 11-12 are in close agreement; the marginally differing warrant S4 here (with an unusually low strength) is backed by “no proven migration past boundary fault” and, less meaningfully, “50% risk for unknown shows is too low!!” for appraisals 11 and 12 respectively.

The main point we make is that it possible to perform this analysis, illustrating that geological knowledge has been captured within these appraisal arguments. This again substantiates our claim that the argumentation framework is fulfilling its intended function.

5.9 Summary and Conclusion

We now summarise our main points of conclusion:

1. The application has achieved sustained usage over a period of 21 months. This strongly supports the utility and practicality of the argumentation approach.

2. The models of opinion are controllable, evidenced by examination of the most extensively used model. The other models of opinion are not sufficiently developed to assess their stability at present. This indicates that the model is both comprehensible and geologically meaningful to the user, and that he can perform the maintenance of his model himself.
<table>
<thead>
<tr>
<th>Appraisal no. (User)</th>
<th>Warrant number</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>8</th>
<th>9</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>15</th>
<th>17</th>
</tr>
</thead>
<tbody>
<tr>
<td>(default →)</td>
<td></td>
<td>100</td>
<td>94</td>
<td>90</td>
<td>77</td>
<td>100</td>
<td>59</td>
<td>100</td>
<td>87</td>
<td>84</td>
<td>61</td>
</tr>
<tr>
<td>3 (C)</td>
<td></td>
<td>90</td>
<td>100</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 (B)</td>
<td></td>
<td>85</td>
<td>100</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 (A)</td>
<td></td>
<td>94</td>
<td>100</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>10 (B)</td>
<td></td>
<td>94</td>
<td>100</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 (D)</td>
<td></td>
<td>90</td>
<td>100</td>
<td>59</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>87</td>
</tr>
<tr>
<td>8 (B)</td>
<td></td>
<td>94</td>
<td>77</td>
<td>100</td>
<td>59</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>50</td>
</tr>
<tr>
<td>11 (E)</td>
<td></td>
<td>90</td>
<td>100</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12 (B)</td>
<td></td>
<td>90</td>
<td>100</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15 (D)</td>
<td></td>
<td>90</td>
<td>100</td>
<td>95</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>60</td>
</tr>
<tr>
<td>53 (B)</td>
<td></td>
<td>85</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>50</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Appraisal no. (default →)</th>
<th>Warrant number</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>25</th>
<th>27</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>(default →)</td>
<td></td>
<td>95</td>
<td>50</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>72</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>95</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>85</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>95</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td>70</td>
<td>56</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>95</td>
<td>100</td>
<td>100</td>
<td>82</td>
<td>89</td>
<td>80</td>
<td></td>
<td></td>
<td></td>
<td>41</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>95</td>
<td>100</td>
<td>100</td>
<td>82</td>
<td>89</td>
<td>80</td>
<td></td>
<td></td>
<td></td>
<td>52</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>33</td>
<td>0</td>
<td>100</td>
<td>100</td>
<td>75</td>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>70</td>
<td>74</td>
<td>100</td>
<td>100</td>
<td>75</td>
<td>91</td>
<td></td>
<td></td>
<td></td>
<td>7</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>95</td>
<td>100</td>
<td>100</td>
<td>80</td>
<td>90</td>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td>14</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>95</td>
<td>100</td>
<td>100</td>
<td>80</td>
<td>90</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>95</td>
<td>100</td>
<td>75</td>
<td>70</td>
<td>93</td>
<td>74</td>
<td></td>
<td></td>
<td></td>
<td>17</td>
</tr>
<tr>
<td>53</td>
<td></td>
<td>95</td>
<td>100</td>
<td>75</td>
<td>70</td>
<td>93</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td>20</td>
</tr>
</tbody>
</table>

This table summarises appraisals of the five prospects appraised by more than one user. This data is extracted from Appendix C, each pair being appraisals of the same prospect by different users. Each line represents an appraisal argument, and the numbers show the strengths of warrants which it contains. Boxes highlight where differences in warrant strength occur.

Table 5.7: Analysis of differences between appraisals by different users.
3. The argumentation approach has helped to reduce variability in expression of risk judgements. This was achieved partly because the argument structure clarifies definitions of terms. This indicates an improvement in the information value of the appraisal records compared with manual appraisals, a significant benefit for decision-support.

4. Users were able to justify their reasoning with geologically meaningful backings in most cases. This again indicates that Optimist is acquiring geological information, essential for its role in decision support.

We also reiterate the qualifications to these evaluations, in order to make their level of significance clear. The most robust indicator is that of sustained usage, by virtue of the long time period it covers. For concluding model stability we have only been able to examine a single model by one user; it is possible that other users may have more difficulty in controlling the models, and thus this result is only an indicator of general model stability for all users. For measuring variability, the nature of manual appraisals and pragmatic issues for acquiring data have constrained the comparisons to be over small samples, thus again qualifying their value. The subsequent interview discussing differences in the eight appraisal pair comparison helped to strengthen the conclusions about variability which were drawn. Finally, although the analysis of backings is qualified by our ability to assess their full geological significance, the large number and proportion containing geological knowledge is a strong indicator of the system’s value as an active information resource.

Despite these qualifications, the results together provide support our argumentation model can provide a framework for a practical and useful aid in decision-making. The strongest support comes from the application’s usage and the analysis of backings. Additionally, the other ‘weak’ indicators are consistent with this conclusion.

Finally, as described in Chapter 4, we note that the user interface was developed informally during the system’s construction. Although the interface itself was not formally evaluated, we refer the reader back to Section 4.10.2 for a discussion of the issues which arose during development.

We have so far focussed in detail on one particular application of the model. We now turn to the model’s generality, discussing what general requirements are needed for its application and the issues which must be addressed for applying it in alternative ways.
Chapter 6

Generality and Extensions

6.1 Introduction

6.1.1 Overview of the Chapter

In this Chapter, we explore the generality and scope of our argumentation model. First, we indicate its domain-independent components by abstracting out a simple argumentation ‘shell’ from Optimist. We then examine the issues involved in applying the model in alternative ways, asking whether our particular application is the only one possible which achieves our goals. Following this, we discuss in more detail how the constraint of a fixed skeleton warrant set could be relaxed.

As illustration of an alternative application of the model, we outline its application in a domain very different to oil exploration, namely that of assessing and advising on the use of machine learning software. Our objective is to demonstrate the utility of the functions which the model provides, and to consider its value as a replacement for a normal expert system. Finally, we discuss areas for future research in the development and application of our model.

6.1.2 A General Argumentation ‘Shell’

The domain-independent features of our argumentation model and its implementation in Optimist can be abstracted out and expressed as a simple ‘shell’, the main procedures of which are given in Figure 6.1. The purpose of the shell is to demonstrate the viability of extracting a domain-independent structure from Optimist and illustrate what that structure might contain. For illustrative purposes the shell has been kept simple and does not include the full range of comparison, search, and analysis functions which would fully exploit the argument database. The structure of arguments which this particular shell constructs is constrained in a similar way to Optimist (Section 4.7.3), with only the warrants which directly conclude the main claim having strengths which the user can modify.

A complete listing of the shell in pseudo-code and Prolog is given in Appendix E, including two example ‘knowledge bases’, one for oil exploration and the other for software recommendation. A trace of its operation is also provided.

6.1.3 Basic Functions Supported by the Model

In order to highlight what our argumentation model can offer in other domains, we describe its properties and functions in more general, domain-independent terms.

The basic process supported by our argumentation model is the construction, discussion and storage of arguments. The simplifying principle of our model is the use of a common set of skeleton warrants by all users with which to construct arguments. Because all arguments are constructed from this common pool of warrants, they share the same general structure and
procedure go:
1. Get the name of the current user
2. Get an attribute-value description of the new case Case to work on
3. Ask the user to select the appropriate model Model of his or her opinion to use
4. Interactively construct an Argument for the claim of interest in Case using Model
5. save the final argument, and any changes made to Model

procedure solve(Argument, Model, Case) returning Argument and UpdatedModel:
1. generate an initial argument InitArgument for Claim using Model
2. discuss InitArgument to obtain Argument and UpdatedModel.

procedure discuss(Argument, Model) returning UpdatedArg and UpdatedModel:
1. describe the argument to the user
2. if the user agrees
   then return Argument and Model
   elseif the user wants a Warrant in Argument checked
   then cite cases attacking Warrant, then discuss Argument again
   elseif the user disagrees with Argument
   then dispute Argument; in the process, Argument will be updated to become MidArgument and Model to MidModel.
   Discuss MidArgument and MidModel to find UpdatedArg and UpdatedModel.

procedure dispute(Argument, Model) returning UpdatedArg and UpdatedModel:
1. ask the user which warrant in the argument he or she disagrees with
2. cite cases defending the warrant
3. if the user still disagrees
   then ask what the new warrant strength should be, and why
   revise Argument replacing the strength and backing of the disputed warrant with the user’s answer.
   Similarly revise Model if the user so desires.

procedure cite_cases(defending/attacking, Warrant):
1. forall other arguments where the same (skeleton) warrant was applied:
   if it was applied with approx. the same/different strength as Warrant
   then cite this in defence/attack of Warrant, also presenting the backing which justified why this strength was considered appropriate.

The complete (and more formal) pseudo-code, Prolog code and examples are given in Appendix E.

Figure 6.1: The top level procedures of a simple argumentation shell.
vocabulary. We can say they are all expressed in the same language, defined by the skeleton warrant set.

The growing database of arguments, expressed in a common language, is a primary resource of an argumentation system. It provides a machine-analysable record of decisions which have been made, offering benefits including the provision of a reference guide, consistency checking and improvement, comparisons; selective data retrieval and statistical analysis. The model is thus well suited to meeting a general requirement of users in many domains, namely that they be selectively informed about what others have done in similar problems, without being swamped by information. Cast in this light, our approach appears potentially applicable to a large number of domains.

6.2 Alternative Applications of the Model

6.2.1 Introduction

The implementation we have described represents one way in which our general argumentation model can be applied. We now discuss possible alternative applications. In particular, we ask whether our specialisation of the model is the only one possible which allows practical argumentation.

The main characteristics of our particular application are:

1. A simple strength calculus was used.

2. Normal warrants (i.e., with modifiable strengths) do not chain together (each chain, i.e. branch, in the argument tree contains at most one such warrant).

3. Grounds are not qualified.

4. The structure of the argument tree is tightly constrained by the two logical warrants at its top (Figure 4.4).

A strength of our application is its simplicity, contributing to its comprehensibility by users. This is a critical concern for meaningful argumentation to occur, and has also been recognised more generally as a fundamental issue in successfully applying AI technology [Lan86]. Our evaluation indicated that this level of simplicity was adequate for decision-support in a complex domain, a significant result for the argumentation model.

However, in other domains these characteristics may be inappropriate, either due to the domain itself or the way users perform problem-solving. We now discuss whether these constraints can be relaxed, asking what problems might arise and what additional machinery might be used to overcome such problems.

6.2.2 General Requirements

Arguments within an application of our model must encode users’ opinions about the problem at hand. Two requirements for this are that:

1. The argument structure must reflect a meaningful way of solving the problem to the users, in order that they can express opinions within it.

2. The strength calculus must be reasonably transparent and strength combination rules should be reflective of the way users combine belief.

Additionally, commonalities between argument structure and content are required in order that they can be compared and exploited by the system.
These requirements are necessary for the users to be able to properly critique the system’s arguments, and for the useful manipulation of arguments by the system. It follows from these that the appropriateness of a representation is both domain- and user-specific. Our simplified model application met these requirements for several reasons:

- Geologists were already familiar with the use of probability.
- The overall structure of appraisal is company-determined and agreed (the six components of risk).
- The multi-process nature of oil formation helped to make the independence assumption reasonable.

However, if we wished to extend the model’s use in this domain, or apply it in another domain, then the simplifying constraints above may need to be relaxed. This must be done in such a way as to preserve the above requirements. For example:

- The structure of different arguments may be less constrained (hence may differ more).
- Warrants may chain together.
- Grounds may be qualified.
- Combination functions may be more complex.

It is important to note that there is nothing intrinsic about these generalisations which prohibits them being applied in our model provided that our two requirements remain met: the argument structure must correspond to a meaningful way of solving the problem to the users, and the strength calculus be reasonably intuitive.

Although in principle these generalisations can be made, there may in practice be additional issues which arise. Whether in fact these issues occur is again domain- and user-dependent. We now discuss them and ways in which they could be addressed.

### 6.2.3 Greater Variations in Argument Structure

In our application, arguments share considerable similarities in structure: for example, they all have the same logical warrant at the top of the argument tree (Figure 4.4). One question concerns what would happen if the argument structure varied more among arguments. For example, some users may prefer to use a different top level warrant, perhaps resulting in their arguments being represented as completely different tree structures (this may happen if we wished to compare arguments across different companies, for example). Two ways this may arise are first if different users established a claim by different methods or second if they used terminology differently. We discuss these below.

#### Differences in Method

If two users establish a claim by different routes, there are two possible ways this can be handled:

1. The differences could be encoded in the backings. For example:

   $A \rightarrow B$ (in Fred’s argument, backed by $C$)
   
   $A \rightarrow B$ (in Fred’s argument, backed by $D$)

2. Different warrants could encode the different methods, resulting in different arguments eg:

   $A \rightarrow C$ & $C \rightarrow B$ (Fred’s way of concluding $B$)
   
   $A \rightarrow D$ & $D \rightarrow B$ (Joe’s way of concluding $B$)
In our application we adopt the former choice. However, we now consider whether such differences could be encoded within the warrants themselves. This would allow our model to also be applied in domains where there was little common structure between users’ ways of addressing the problem.

First, some simple extensions to the machinery for constructing arguments may be necessary. In our application, all relevant warrants for a claim are always applied; however in other applications users may wish to disable warrants from applying to save them critiquing every possible line of reasoning leading to a conclusion. In this example, Joe may simply not want to include $C \rightarrow B$ in his argument (although he may be happy that Fred’s method is a valid alternative to his).

Second, arguments can still be straightforwardly compared where commonalities exist (providing terminology is used consistently; we discuss this shortly). For example, even if two arguments have very different structures:

- Warrants with the same skeleton can still be compared.
- Common subclaims within the arguments can be compared.
- Subtrees which do share a common structure can be compared.

Our application only compares primitive components of arguments (warrants, claims and qualifiers). Additional comparison functions could also be envisaged which would also compare aggregated sections of arguments. In the Fred-Joe example, such a comparison function might say “Fred thinks $A$ strongly implies $B$, whereas Joe does not” thus combining the two chains of warrants $A \rightarrow C \rightarrow B$ and $A \rightarrow D \rightarrow B$. This would be an interesting area of future work, providing analyses of arguments at different levels of abstraction.

Third, in our application, backings attach only to single warrants. However, if we allow users to select different lines of reasoning, there would be scope for additional backings which justify those selections. This again can be seen as a task of manipulating arguments at different levels of abstraction.

Fourth, if two argument trees were completely different in structure and content, comparison would be more difficult. This of course is not just a problem specific to the computational framework; if people disagree completely, without any common points of reference, then argumentation is difficult as a social phenomenon also.

Fifth, we note the practicality issue which arises when more detail is included in the argument tree, namely that the user should not be over-burdened. There is a trade-off here between a complex tree (allowing disagreements to be finely located) and a simple tree (which is easy for the user to critique). This can be seen as a task of selecting the appropriate level of granularity to suit the domain and target users.

There is a related trade-off also to consider, of open-endedness versus detail of analysis. A simple tree, where much knowledge is encoded in backings, permits users flexibility in expressing how their conclusions were reached but limits the detail with which arguments can be compared. Conversely, a complex tree allows points of disagreements to be finely located, but also the encoding of users methods in warrants constrains the ways the users can express solutions (eg. if the user invokes a new method, new warrants would have to be added). Finding the appropriate balance is a domain-specific issue. Again, this trade-off could be eased by machinery for working with arguments at different levels of abstraction: for example, if no warrants existed for encoding details of the user’s method (eg. $A \rightarrow E$ & $E \rightarrow B$), then a more abstract warrant could be used instead (eg. $A \rightarrow B$) with the method encoded in the backing.

We finally emphasise that we are not just concerned with comparing arguments of different users, but also by the same user at different times in order to encourage consistency. For the latter comparisons, it can be expected that a single user’s arguments will be similar in structure unless his or her approach to the problem is very eratic.
Differences in Terminology

A second issue is that of differences in terminology. For comparisons to be valid, terms must be used in a consistent way. One advantage of the computer argumentation approach is that definitions are made more explicit; for example, warrants explicitly state which grounds are relevant to the definition of a claim. We can say the computer clarifies the frame of discernment over which a model of belief is defined.

In our application, users were happy to work with the system's definitions of ambiguous terms as encoded in the skeleton warrants. (Section 5.6.2). We consider how the argumentation model could operate if this was not the case, for instance (taking the geological example) if Fred felt strongly that leaky faulting affected trap presence according to his definition of trap presence, but Joe did not.

Again there is no reason in principle why the argumentation model could not handle such differences, but additional machinery would be required. Most simply, the two different concepts could be given different terms ('trap presence according to Fred', '... according to Joe') allowing different users to solve the problem according to their own method as discussed previously. A more ambitious extension would be to encode the relation between different users' definitions, allowing a user to view arguments using his terminology. However, we emphasise that in our application ambiguity was met with flexibility rather than conflict on behalf of the users: it is an open issue as to whether users in other domains would be strongly reluctant to work with a particular, clearly defined definition of a term.

Finally, we note terminological differences could also arise elsewhere, eg. in the definition of a ground in an argument. Alternative applications of our model could usefully include machinery for clarifying such definitions eg. on-line text, examples, illustrations.

6.2.4 Chaining of Warrants

In our application, only one warrant with tunable strength could occur in any branch of an argument. In an alternative application, several such warrants could chain together. A chain of warrants serves to break up the confidence in a claim (given some grounds) into constituent parts; in fact, this is only one particular way in which some overall belief in a claim can be decomposed (combining warrants in parallel, as in our application, is another example). We now discuss the issues which arise.

First, to critique an argument, elements within it must be meaningful to the user. Users may not be accustomed to think of some overall confidence that A implies B as separate confidences in individual steps of reasoning (eg. $A \rightarrow_{s_1} C$ & $C \rightarrow_{s_2} B$), and thus it may require them to consider the components of the problem in more detail. This is not necessarily something to be avoided; indeed, the purpose of problem-solving guidelines as used within many organisations is precisely to encourage consistent, thorough analysis of decisions. Similarly, computer argumentation can assist in this task while additionally being reactive to the user's expressed reasoning.

Second, the pragmatic issue of not over-burdening the user with large numbers of warrant strengths to evaluate applies here. With long chains of inference, the user may be tempted not to evaluate the whole chain providing its conclusion was correct. This would weaken the argument as an accurate representation of opinion.

Third, we require that the argument be stable to users' expressions of belief. Users can only quantify belief to a certain level of precision - thus small changes in qualifiers and strengths which do not meaningfully represent a different level of belief to the user should only have small changes in the rest of the argument. For some strength calculi, there may be a danger of 'error amplification' where meaningless variation in a strength somewhere has large effects elsewhere, the variation amplified by propagation through several chained warrants. In this case, the calculus is not adequately reflecting the user's reasoning and either should not be used or the length of chains limited.
6.2.5 Qualified Grounds

A useful extension of our application would be to allow grounds to be qualified by the users. Uncertainty in grounds is handled in a limited way within it (Section 4.5), with some grounds being assumed certain.

The two considerations for this are as follows:

1. An extra level of complexity in the strength calculus is added for the user to grasp.

2. If the user is asked to specify a value for a ground from an exhaustive, mutually exclusive set of options, then machinery would be required to allow users to specify a distribution of qualifiers over the options (eg. a set of probabilities summing to 1.0). This moves towards a fuzzy logic representation of uncertainty, similar to that used in the SPII-2 prospect appraisal system (Section 2.2.7).

Again, the key requirement is that the users can maintain an understanding of the system’s argument.

6.2.6 Alternative Models of Strength Combination

As discussed in Chapter 4, a suitable model of strength combination should fulfil the following requirements:

1. It is local, ie. the combination rules are functions of the warrants’ strengths and the grounds’ qualifiers only.

2. It is easy to understand, ie. 
   
   - the warrant strengths and qualifications should represent opinions in a way transparent to users.
   
   - the combination rules should reflect the way users’ combine evidence.

In our application, our calculus fulfilled these requirements as geologists were already reasonably familiar with probabilities through normal working practice, and the physics of oil formation allowed the use of simple combination rules.

The review in Chapter 2 and other literature suggest several alternative calculi which could be used, for example:

- Modified Bayesian inference (eg. Prospector and Explorationist, in Section 2.2).

- fuzzy logic (eg. SPII-2, in Section 2.2).

- Other expert system calculi eg. Mycin’s certainty factors [Sho76].

- A modified version of our calculus where risk is assessed in relation to a ‘typical’ rather than ‘ideal’ prospect ie. the initial probability of oil is taken as a prior probability (eg. $q_o = 0.2$) rather than $q_o = 1.0$ (Section 4.6.2). Here, warrant strengths $s$ greater than as well as less than 1.0 would be used, resulting in a more Bayesian-like calculus.

- A calculus using qualitative estimates of warrant strength (we illustrate an example of this later for another application domain in Section 6.5).

- A calculus using a boolean representation of belief (eg. true/false).

The most important point is that there is no reason in principle why the above calculi could not be used providing the locality and comprehensibility requirements are met. The appropriateness of these different calculi depends on several factors:
The domain: in some domains, simple combination rules may suffice. In others, evidence may combine in more complex ways.

The users: users may find some calculi more easy to understand than others by virtue of their background or experience.

User interface design: Comprehensibility to the user is partly affected by interface design. In Prospector, for instance, users found conditional input probabilities easier to estimate when they were rescaled in the range -5 (evidence is absent) through 0 (prior probability of evidence is unchanged) to +5 (evidence is present). Similarly, graphical presentation (eg. bar charts, etc.) can make the calculus more transparent.

6.2.7 Summary

Thus there are several alternative ways in which the model can be applied. For their successful application, we must preserve the meaningfulness of represented arguments to the users and commonality between arguments to allow their exploitation by the computer. The psychological issues of the representation’s presentation and comprehensibility are thus important; further work would be required to fully assess the suitability of different representations in these respects, and whether alternative trade-offs between simplicity and representational power could be made. The most important point of summary is that there is nothing inherent in these alternatives which prevents their use; instead, their ability to meet our requirements for argumentation is dependent on the particular application domain and users.

6.3 Additional Requirements for Application

We now make some additional brief comments on requirements for generally applying the model.

1. The problem is normally solved by application of knowledge rather than extensive search (ie. the task is knowledge-based).

2. The problem can be expressed as one requiring a claim to be established rather than a solution to be constructed (eg. constructing a plan).

3. Disagreements with the system can be localised, ie. attributed to specific warrants used in the argument.

We discuss these below.

6.3.1 The Task is Knowledge-Based

Our argumentation model is founded on a knowledge-based approach to problem-solving, reaching conclusions mainly by the application of knowledge rather than extensive search. As argumentation concerns the interaction of different lines of reasoning, we require that such knowledge-rich lines of reasoning exist in the first place. This does not mean, of course, that search is not required for argumentation; indeed, our argumentation model can be viewed as enhancing normal knowledge-based problem-solving with search-intensive functions for supporting or challenging that knowledge (eg. argument comparison and retrieval, consistency checking).

6.3.2 The Task is of Establishing a Claim

The model is also designed for problems requiring a claim to be established, rather than a solution to be constructed (ie. the task is not one of synthesis). We point out, though, that this is mainly a requirement of problem formulation, and that many tasks requiring constructed
6.3.3 Disagreements are Localised

Finally we require that disagreements with the system can be *localised*, i.e., can be broken down ('factorised') into separate points of disagreement with individual warrants. This requirement is necessary because our argumentation model cannot represent a 'spreading of blame' over several warrants (a backing can only attach to a single warrant).

As an example of unlocalised disagreement, consider the user saying “I think this group of five warrants should strongly imply X, so I’ll spread this influence equally among each warrant, adjusting their strengths to be equal and combine so that X is strongly implied”. Two problems arise. Most significantly, our model assumes that a warrant’s strength is a function of its backing alone, and in particular is independent of the argument within which it is embedded (Sections 3.4.3 and 4.6.1). This assumption underlies the whole concept of users being able to assign strengths to warrants independent of the arguments in which they are used, and validates argument comparison operations based on warrants’ strengths. However, with some strength calculi, this ‘spreading the blame’ reasoning will violate this assumption as the warrant strength will then also be a function of the number of warrants among whom the blame is spread. A second problem is that, if large numbers of warrant changes are made for every argument, the selectivity of some argument comparison functions may degrade. For example, if many warrant changes result in many inconsistencies then search for old arguments inconsistent with the current argument may tend towards retrieving them all, thus over-burdening the user.

This requirement is more of a constraint on how a system based on our model is used, rather than on how the problem is represented. It can be regarded as a requirement that the correct semantics of warrants are properly adhered to — by definition, a warrant’s strength should not depend on how many other warrants apply in an argument. It is also a significant requirement; in Optimist users were observed to partly ‘work backwards’ from a target conclusion to assign warrant strengths (Section 5.6.3), indicating a danger that choices of warrant strengths were partly influenced by what was needed to reach a conclusion. An argumentation system’s response to this can be viewed positively, as it will highlight where warrant’s strengths appear inconsistent with the evidence by looking at previous arguments.

6.4 The Skeleton Warrants: Strengths, Constraints and Extensibility

6.4.1 Introduction

We have discussed alternative ways in which the model can be applied. We now turn to the issue of maintaining such applications, and in particular how to relax the requirement that the set of skeleton warrants is fixed. We consider maintenance for the case where users share the skeleton warrant set; similar issues also arise if users only partially share warrant sets (Section 6.2.3).

A strength of having a fixed, shared set of skeleton warrants is that this provides a common language with which to express arguments. This allows a level of comparison, retrieval and search facilities to be provided which otherwise would not be possible. It also constrains knowledge maintenance operations by the users in such a way that this commonality is preserved.

We now discuss the limitations and generality of this property, as it represents one of the constraints as well as benefits of our model. First, we describe the degree of freedom this constraint permits for users to express disagreements at a general level. Second, we discuss how to relax this constraint, allowing the alteration of the skeleton warrants themselves. The
main requirement for this is that the system can translate arguments constructed using an old warrant set into the language of a revised set. This is necessary to preserve argument comparison operations. The ability to tackle this issue affects the generality and extensibility of our argumentation model directly.

### 6.4.2 The Flexibility of the Model for Expressing General Changes

We first discuss the flexibility which the user has to make general changes to the system’s model of his or her opinion, without altering the skeleton warrants themselves. A user may alter a warrant’s strength for two reasons:

1. it does not reflect his or her opinion (its initial value is ‘wrong’).
2. while the user generally agrees with the strength, there is some anomaly about the particular case under consideration which suggests the strength should be different in this case.

The latter case can be viewed as a deficiency in the warrant set; there is some attribute of the case which influences this warrant’s strength which is not being tested on. For example, consider a user altering a warrant stating ‘a geological fault strongly implies no oil’ because of a particular anomaly in the current case, eg. ‘faulting is strike-slip’.

In our argumentation model, there are various levels of generality with which a user can express his or her disagreement with a warrant:

1. Altering its strength as applied in the current argument under consideration, while leaving the general warrant (in the model of opinion used to construct that argument) unchanged. This change is thus effected for the current argument only.
2. Altering its strength also in the model of opinion used to construct the argument, hence affecting future arguments constructed with that model.
3. Propagating this change also to any other of the user’s models of opinion.

We have stated that a user may own several models. In Optimist, these models are organised hierarchically, corresponding to a general opinion about increasingly larger geographical areas. Thus a user can change a warrant for a particular prospect, or for the local area containing the prospect, or globally in all his or her models. In all events, the user’s description of the anomaly is stored as the backing to the changed warrants.

We make two points:

1. The argumentation model can handle deficiencies in the set of warrants; however, corrections are expressed not in the warrant itself but in the warrant’s backing.
2. The hierarchical organisation of a user’s models allows him or her some (limited) control over the degree of generality with which those changes can be made. However, this degree of generality is constrained to be only along the dimension with which the user’s models are hierarchically organised. (In Optimist, this dimension is geographical area).

To go beyond this, the warrants themselves would have to be modified. We discuss this below.

### 6.4.3 Changing the Set of Skeleton Warrants Themselves

Although users can perform the day-to-day maintenance of the models of their opinions, we also expect occasions to arise when the set of skeleton warrants itself does need to be changed.
In order for a system to continue making comparisons between arguments, some means is needed of translating arguments constructed with an out-of-date warrant set into a form compatible with the current set. We can view changes to the set as a change in language, requiring a corresponding translator to maintain compatibility.

We now describe three categories of change which might be made to a set of skeleton warrants, discussing the machinery required to accommodate them.

**Extension of the Warrant Set**

A new warrant can be added without requiring any translation of existing, stored arguments if it is inapplicable in all those arguments. This may occur when a user, noting a hitherto-unseen anomaly about a new case, realises a warrant for it would be appropriate. It also might occur if the scope of the system was to be extended eg. in geology from oil in Alaska to oil anywhere.

The new skeleton warrant is added to the existing set of skeleton warrants, and also to all models of opinion with some default strength assigned to it. If a new property of cases is also being introduced, then the default must be that the property does not apply in previous cases. The user may prefer to introduce a second, new warrant for this default value which has no strength reflecting the assumption made in previous arguments.

One of the two programmer-requiring changes made to Optimist’s warrants so far was of this type, when extra values of an attribute were added.

**Information-Preserving Modifications to the Warrant Set**

A second type of change is one which makes some previous, stored arguments ‘incompatible’ with the modified warrant set, but where there is sufficient information in the old arguments to permit automatic translation to a form compatible with the modified warrant set. (By ‘compatible’ with a warrant set we mean the argument is expressed in the language defined by that set).

Examples of such modifications include renaming attributes and renumbering warrants. This occurred in Optimist when two attributes A and B (both with possible values \(v_1, v_2, v_3\) and \(v_4\)) were combined to a single, new attribute C with values \(v_1, v_2, v_3, v_4\) and \(v_5\), where a one-to-one mapping existed from the old values of A and B to the new value for C (only five combinations of A’s and B’s values were ever used, mapping onto C’s five values. The corresponding new warrants are numbers 6, 7, 8, 9 and 10 in Appendix C, replacing the previous warrants for values of A and B).

The important point we make is that this type of modification can be tolerated in our model by encoding a small translation procedure for converting arguments in the old form to the new form, hence maintaining compatibility between all stored arguments.

**Information-Requiring Modifications to the Warrant Set**

Information-requiring modifications are those where old arguments do not map to a unique new argument (the old arguments have become ambiguous in the new argument language). These modifications are more difficult to handle, as fully automatic translation of arguments from the old to new argument language are not possible.

Examples include changes of vocabulary (eg. recategorising geological faults as strike-slip/tectonic/compression instead of downthrown/upthrown), specialisation of warrants (eg. splitting a single warrant into two, depending on the value of a newly introduced attribute), and more major structural changes to the argument language.

Modifications such as these could be handled in two ways:

1. Accept a degree of incompatibility among arguments. However, this would mean some comparisons between some arguments would no longer be possible.
2. Interactively translate old arguments into new arguments, the system asking the user for the missing information to make the translation. Such translation could occur in one pass (simply step through all the old arguments), or at run-time as and when needed. In the latter case, a hypothetical dialogue might look:

“In the current case, you’ve considered tectonic faulting to strongly reduce the likelihood of oil. However, this may be inconsistent with prospect e/21-1a, where faulting had almost no effect on oil. In e/21-1a you described the faulting as downthrown—how would you describe it in terms of strike-slip/tectonic/compression?”

6.4.4 Summary

The constraint imposed and flexibility permitted by using a common warrant set is an important issue concerning the generality of our argumentation model. The two key points we make are that:

1. A fixed warrant set still allows users significant flexibility to express their opinions. Users may still completely disagree about the reasons underlying warrants (the backings), and users have a (limited) degree of control over the level of generality at which their opinions can be expressed to the system.

2. We can easily envisage machinery for allowing the warrant set itself to be changed, suggesting further flexibility of the argumentation model can be introduced.

6.5 Application in Another Domain: The ML Advisor

6.5.1 Introduction

We now consider how to apply the argumentation model to a very different task, namely that of advising on software usage. Our objective is to demonstrate the generality of our argumentation model by illustrating its application to this very different domain. Following this, in Section 6.6 we briefly consider applying the model instead of a normal expert system for the task of medical diagnosis, to further illustrate the model’s generality and advantages.

In this first application, the task is to advise users, interested in applying machine learning techniques to solve some problem, as to whether the particular machine learning tool ID/CN (implementing algorithms ID3 [Qui83] and CN2 [CN89, CB91]) is appropriate or not. This domain has been chosen firstly because the author has sufficient expertise within it to address the task realistically, and secondly because the domain’s substantial differences with risk assessment help considerably in scoping the generality and limits of our argumentation approach.

6.5.2 The Problem Task and an Argumentation Approach

The particular tool ID/CN generates rules for classifying examples, taking as input a set of already-classified ‘training’ examples. ID/CN’s successful application depends primarily on these training examples being expressed in a suitable form (as opposed to any intrinsic properties of the domain itself). Machine learning experts are familiar with a number of ‘tricks’ by which data in an unsuitable form can be transformed into a suitable form. This advice-giving task, then, can be described as follows: Given some data in a particular (possibly unsuitable) form, how can it be transformed to a form which is suitable for ID/CN?

There are two main difficulties to be overcome in order to apply our model:

• The task is one of giving advice. To apply our model, this task must be recast in a framework based on establishing a claim.
• The user is not expected to be an expert in the domain, as was the case with Optimist.

These features constrain the way the argumentation model can be used.

Concerning the first point, we can re-express the problem as one of establishing whether ID/CN is appropriate. The model then offers a method for advice-giving based on the organisation and retrieval of backings in arguments supporting or attacking ID/CN's suitability in previous cases. It is these backings which express advice about how the data can (or cannot) be transformed into an appropriate form for ID/CN. Cast in this framework, the general requirements of users in geology and this domain are similar, namely to be selectively informed about what others have done with similar problems, without being swamped by information.

Concerning the second point, although non machine learning experts cannot 'argue back' with the system initially, they will obtain knowledge after following the system's advice. At this point they can argue back should the advice turn out to be incorrect, thus correcting and expanding the system's database of arguments.

The argumentation model thus offers several important benefits:

1. Non machine learning experts can run the system and enjoy the benefits of the inter-case comparison functions to provide them with information about the domain

2. Non machine learning experts can expand the system's argument database by arguing back about cases after following the system's advice

3. The expanding record of arguments provides useful information for domain experts to later improve the system's knowledge base

In this way the argumentation approach offers incrementality to the advice-giving system, allowing both experts and non experts to maintain the system's knowledge. This is essential in a domain such as this where the advice will necessarily be simple and approximate – due to the limits of AI techniques for representing the full details of the machine learning application in question – and makes it imperative that the system is responsive to and can accommodate disagreement with the user.

6.5.3 A Sketch of an Implementation

One way of providing these functions with our argumentation model is as follows. Given a particular machine learning application, a description of the form of the available data to learn from is given to the system. This constitutes the case description about which an argument will be constructed. Following this, a model of (a machine learning expert's) opinion is used to determine whether ID/CN can be applied to the data in this form. We call the resulting argument (concerning ID/CN's suitability) the base argument.

If the base argument concludes that ID/CN is unsuitable, the system then tries to refute the argument by focusing on its weakest component(s), namely the warrant(s) which most strongly imply ID/CN is unsuitable. A refutation is accomplished by challenging this warrant (the system 'argues with itself'). It is possible to issue a challenge if this warrant has been previously challenged in another case. A challenge to a warrant is a case where the same (skeleton) warrant was applied but with a reduced strength about ID/CN's unsuitability, backed by a statement normally of the kind:

"This warrant does not hold when it is possible to perform the following transformation on the data..."

It is this backing which advises the user how his or her own data might be transformed into a form more suitable for ID/CN. A repertoire of such 'advice cases' is initially created by an expert during construction of the system.
Alternatively, if the base argument indicates that ID/CN is suitable, it is possible that once a trial has been run the user will judge the results a failure. If so, the user may then ‘argue back’ with the base argument, indicating which component was most at fault by altering its associated warrant and providing a backing describing the problem which arose. The modified argument is then added to the system’s repertoire of arguments for future use.

To summarise:

1. We simplify by using only a single model of opinion advising on ID/CN’s suitability, encoded by a machine learning expert.

2. Non machine learning experts can query, dispute and modify the arguments produced with this model but not the model itself.

3. After generating a base argument suggesting ID/CN is inapplicable, the system ‘argues with itself’, attempting to refute it as a means of locating the appropriate advice in precedent cases.

4. After following the system’s advice, the users can add to the system’s knowledge and advice by arguing back (with hindsight) with the original advice.

6.5.4 Hypothetical Interaction

To illustrate this, we now sketch a hypothetical dialogue which might occur. The user’s replies are underlined, and explanatory comments are shown in italics.

Data Entry

*Initially the system asks for a description of the current case to assess ID/CN’s suitability for*

Please enter a description of your problem:

What is the name of the domain? fault-diagnosis
Please give a brief description of the domain:

Diagnosis of faults in a simple electrical circuit, given voltage readings taken at several junctions in the circuit

Please answer the following questions about the data you currently have:
Can an expert easily provide more examples manually? no
Can an expert manually classify the examples, using no other data? no
Do experts always agree on the classification? yes
Do decisions require knowledge of relations between attributes eg. mathematical relns (as opposed to depending on simple logical combinations of attributes)? yes
Is information external to the example relevant (eg. its environment, its context)? yes

Approximately how many attributes are there? 20
Approximately how many examples are there? 10000
Approximately how many classes are there? 5
Are you running on a Sun3 or a Sun4? Sun3
What is the maximum acceptable run-time (min)? 10

*A full list of questions for characterising a dataset is given in Appendix F.1*
6.5. APPLICATION IN ANOTHER DOMAIN: THE ML ADVISOR

Figure 6.2: The Base Argument for ID/CN’s use in Circuit Fault Diagnosis.

Generation of the Base Argument

Now the system applies the warrants to the data to construct an argument concerning ID/CN’s appropriateness. An example skeleton warrant is shown below:

```
skel-warrant 1 ::
    if 'manually provide exs' = n
    and 'examples available in db' = n
    then ID/CN is inappropriate.
```

A full list of skeleton warrants for this problem is given in Appendix F.2, and the model of (my) opinion about the strengths of these warrants is given in Appendix F.3. The result of applying this model is the base argument, sketched in Figure 6.2. Having generated an argument, the system describes those warrants which most strongly concluded that ID/CN was inappropriate. The user’s main interest will be in the backings for those warrants.

Assessing the appropriateness of ID/CN to this problem domain:

From the description you have given me, I conclude that ID/CN is likely to be inappropriate with the data in this form.

The characteristics of the problem domain of most concern to me are:

1. Complex relations exist between attributes. (warrant 4)
   Reason: (i.e. the backing) ID/CN uses a constrained language for expressing its rules, which is unable to express relations, and hence unable to take them into account for classifying new examples.

2. Contextual information is relevant to the classification. (warrant 5)
   Reason: ID/CN can only generate rules which test on the information provided. If additional information is needed for accurate classification, but not expressed in the example set, then the resulting classificational power of the induced rule set will also be diminished.
3. The learning time is too long (warrant 21).
   Reason: You have too many attributes and examples to allow learning to complete within
   the time you specified!

**Refutation of the Base Argument**

*Having identified the major causes of concern about the current form of the data (repre-
   sented by those warrants which strongly imply ID/CN is inappropriate), the system
   now looks for previous arguments where this concern was alleviated (i.e. the same
   (skeleton) warrant only weakly implied ID/CN was inappropriate). In each refuting
   argument, the warrant’s backing describes the reason why the data set characteristic
   was not problematic, describing a transformation of the data which successfully
   overcame the problem. It is this description of transformations which constitute the
   advice to the user.*

We illustrate the system’s refutation of the first cause of concern, namely the existence of complex relations

Attempting to refute the warrant for complex relations...  
Hey! Gary successfully used ID/CN in the domain of chess, where complex relations exist.  
Also, Pete successfully used ID/CN in the domain of credit rating, where complex relations exist.

Do you want to see more information?  yes

*Here the system replays the backings associated with these warrants in these arguments, describing how the problems of complex relations were avoided.*

**Case 43: The Credit Rating Domain:**
Complex relations were not a problem because (backing for warrant 4, shown below):

<table>
<thead>
<tr>
<th>BankAcc</th>
<th>Savings</th>
<th>House</th>
<th>OKHist?</th>
<th>CreditRating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applicant 1</td>
<td>£12,000</td>
<td>£300</td>
<td>£30,000</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In this particular example, the sum of BankAcc + Savings + House were relevant to giving a CreditRating. This sum could simply be expressed as an extra attribute, ie.

<table>
<thead>
<tr>
<th>BankAcc</th>
<th>Savings</th>
<th>House</th>
<th>Total</th>
<th>OKHist?</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appt 1</td>
<td>£12,000</td>
<td>£300</td>
<td>£42,300</td>
<td>yes</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Thus you’re advised to try and construct similar transformations to make important relations explicit as additional attributes in your problem.
Peter Clark
2.7.90

**Case 71: The Chess Endgame Domain:**
Complex relations were not a problem because:
It is possible to transform from the initial descriptions to a form in which the complex relations are added as extra attributes. The original data appeared:

<table>
<thead>
<tr>
<th>WhiteKing</th>
<th>WhiteRook</th>
<th>BlackKing</th>
<th>IsLegalPosn?</th>
</tr>
</thead>
<tbody>
<tr>
<td>(X, Y)</td>
<td>(X, Y)</td>
<td>(X, Y)</td>
<td></td>
</tr>
<tr>
<td>board1 a</td>
<td>4 d</td>
<td>8 g</td>
<td>yes</td>
</tr>
</tbody>
</table>

There are important relations between the position, eg. if the black king and white rook have the same X (rank) or Y (file). The transformed data set looked of the form:

<table>
<thead>
<tr>
<th>Kings</th>
<th>BK and WR</th>
<th>BK and WR</th>
<th>IsLegal?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Adjacent</td>
<td>SameRank?</td>
<td>SameFile?</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>board1</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
</tbody>
</table>

You should also consider whether similar transformations are possible with your particular data set. Note: you may decide to leave out the original attributes.

More information about this can be found in “Structured Induction in Expert systems” (1987) A. D. Shapiro, Wokingham: Addison-Wesley.

Peter Clark
12.12.90

Thus the system has given advice about how the problem of complex relations might be avoided based on previous, stored arguments about other cases. Appendix F.4 describes a number of possible cases whose arguments could be annotate with advice and form the initial database of arguments in the system.

Do you think you will be able to follow this advice to alleviate the problem of complex relations? yes

Modifying the base argument...

If the user is persuaded that he or she can overcome this problem given this advice, then the (initially strong) warrant against ID/CN in the base argument is weakened. The user repeats this for the other factors of concern identified by the system also.

‘Postmortem’ Argumentation

Later, after following the advice, the user may return to ‘argue back’ with the system’s original opinion and add the resulting argument about ID/CN’s appropriateness to his or her problem into the argument database.

How successful do you consider your attempt with ID/CN? unsuccessful

The original base argument I generated also suggested ID/CN would be unsuccessful. These were the main points of concern in that argument:

1. Complex relations exist between attributes
2. Contextual information is relevant to the classification.
3. The learning time is too long

However,
I was also able to offer refutations to these factors, all of which you accepted!

Name a factor which still caused you problems: 1

Please describe why:
Although you advised me that problems of complex relations could be avoided by adding new attributes expressing those relations, it turned out impossible to do so in this domain because we could not identify what those relations were.

Having added several known important relationships as new attributes, the algorithm still did poorly. There remain a number of other relationships which we know exist but cannot adequately define.

Thus a corollary should be added to this advice, namely that the domain must be sufficiently well understood that the important complex relations can be identified.

*As a result of this challenge by user (a ‘refutation of the refutation’), the warrant in the argument is changed again, strengthened to imply ID/CN is inappropriate.*

Any other factors which still caused you problems? *no*

*The final argument is stored, thus making the user’s comments expressed in the above backing available for future users who also encounter data sets where complex relations between attributes are relevant to classification.*

### 6.5.5 Summary

We have thus sketched out how our argumentation model could be applied to a problem with substantially different characteristics to risk assessment. Our main point is that the basic processes it supports – namely the construction, storage, and selective retrieval of records of the reasoning performed by different users in different problems – appears applicable to a wide variety of domains, including those with characteristics substantially different to risk assessment such as the advice-giving task described here. This domain also illustrates the model applied using an alternative, qualitative strength calculus.

### 6.6 Application in Expert System Domains

To further discuss the model’s generality, we briefly consider using the model instead of a normal rule-based expert system for a knowledge-based task (as opposed to tasks where expert systems cannot easily be applied). We take medical diagnosis as an example domain, considering replacing a diagnostic expert system such as Mycin (say) [Sho76] with an argumentative version (‘A-Mycin’, say). For conciseness we do not sketch a full implementation but merely highlight the important points of comparison.

The process by which an expert system produces diagnostic advice is sketched in Figure 6.3. An argumentation version of the system can be viewed as a generalisation of the original rule-based architecture, in which rules (now warrants) can be queried and challenged, and where records of each diagnostic argument are stored and can be searched.

We summarise several advantages this would offer over a normal rule-based system:

**Overcoming an Incomplete Description Language:** Nuances of diagnostic problems, outside the scope of the system’s language for describing patients, can be included in the final diagnostic argument by the expert arguing with the system and describing them in backings. For example, a particular peculiarity in an X-Ray may be relevant to diagnosis, but not expressible in the patient’s description language.

**Case Retrieval:** Previous, relevant diagnostic problems and experts’ opinions can be retrieved and presented to the user. This serves to highlight where alternative diagnoses should be considered, indicate ‘special cases’ which should be watched out for, and indicate how successful diagnoses following the same reasoning were in the past (see ‘postmortem analysis’ below),
Figure 6.3: Producing Diagnostic Advice using an Expert System.

‘Postmortem’ Analysis: (of arguments not patients!). Retrospective analyses of the accuracy of diagnoses can be made, allowing the diagnosis process to be further refined and improved.

Knowledge Maintenance: Users can perform the day-to-day maintenance of the system’s models of their opinions themselves.

Record Keeping: Records of the reasoning behind different diagnoses are constructed and stored.

Thus there are a number of advantages to be gained by generalising a normal rule-based approach into an argumentation approach based on our model. It is also interesting to note that during the evaluation of Mycin, significant differences of opinion among therapists were encountered; for example, on average only 42% of therapies recommended by 9 experts were rated as acceptable by 8 other (independent) experts [YFB+84]. Although Mycin (as a 10th recommender) fared well (in fact best, scoring 52%), this still low score highlights the significance of differences of opinion in this domain. This suggests an argumentation approach, which can account for these differences, would be valuable here.

6.7 Future Work

We have argued that our argumentation model can be usefully applied to a wide variety of domains, operating within the requirements for its use set out in Section 6.3. We now turn to the interesting question of how the model and its implementation might be further developed, improving its support for users and allowing these requirements to be relaxed or met more easily. Some of these developments can be seen as improved implementation techniques, others as a development of the argumentation model itself.
6.7.1 Representing Semantics of Text Backings

One extension to the model would be to replace text backings with a formal representation of their semantics. A backing could itself be a complete argument, claiming that the warrant is valid. This would provide a hierarchical structure of arguments, allowing a more sustained dialogue to occur as advocated by Stutt [Stu87] (eg. the backings themselves could be argued about).

As discussed in Section 1.5.2, the issue here is not whether the improved functionality would be desirable, but how it can be constrained so that practicality can be maintained. Again, the knowledge acquisition problem appears; we desire a formal structure for backings but do not want to insist users learn to program. We also wish to keep the representation of backings ‘open-ended’, not constrained to a particular vocabulary of concepts (Section 3.9.2). Further research is required to find feasible solutions to these problems.

We also suggest simpler, more limited ways in which the use of text backings can be extended. At the simplest level, users could be encouraged to re-use backings, thus enabling semantically similar backings to be identified. This could be implemented by offering users a list of existing backings to select from when a new backing was required. For example, following the user altering the strength of a warrant about geological faulting, the dialogue requesting a backing might proceed:

And why do you consider the faulting worrying in this prospect?
1. it is strike-slip
2. it is in a sub-sedimentary tertiary environment
3. (add a new reason)

Reason: 3

What is the new reason? it cuts sand-bedded aeolian deposits

Here the user has been offered existing backings from previous applications of this warrant to select from. In this case he or she has decided to add a new backing.

Being able to identify semantic equality of backings would refine the comparison and analysis functions possible. Given a particular backing is provided, all other arguments where the same warrant and the same backing can be selected for retrieval and consistency analysis. It may also be possible to incorporate the backings at the data entry stage of the system’s use, thus extending the problem description language.

6.7.2 Changing the Warrant Set

A second extension of the implementation would be to allow users to make some changes to the skeleton warrants themselves. Two issues need to be addressed for this:

1. Compatibility of already stored arguments with new arguments should be maintained (as discussed earlier in Section 6.4.3).

2. The user should not require programming skills.

To address the first issue, some means for users to express the relation between the old and new warrant language would need to be designed. Or, in the simplest case, users could be constrained to only make changes of the type which did not damage compatibility (Section 6.4.3).

Concerning the second issue, recent advances in knowledge acquisition techniques offer possible ways forward. The recent ‘ripple-down rules’ knowledge acquisition methodology for example, enables a system to list possible corrections to buggy rules itself by comparing examples of the rule’s correct and incorrect behaviour, and can then ask the user to select the appropriate correction from that list. In this way the user does not need programming skills. However,
we should also point out that there are still several limitations with this methodology which need to be overcome, for example an adequate vocabulary must be provided to the system for ripple-down rules to work. The reader is referred to [SCM+91] for a more detailed discussion of this technique.

6.7.3 Development of Models of Opinion

Models of opinions are approximate expressions of users' subjective judgements about a domain. There are several ways in which their representation, use, and validation can be further enhanced which we now discuss.

Statistical Validation

In domains where the task is to make a prediction, the implementation could be extended to include some statistical validation of the user’s warrant strengths, based on the known outcomes of previous cases. If the user considers X a major influence on some claim, but statistically this can be shown not to be true, the user should be informed.

While this development seems straightforward, we make a point of caution. In domains where different expert opinions exist, it is often precisely the lack of any objective means of determining the “right” answer which has lead to those differences arising in the first place. Thus in domains where opinions differ, it may often also be the case that an objective statistical approach has limited value.

Qualitative Representations of Strength

It is well known that users can have difficulty in precisely quantifying degrees of belief (eg. [Sa87, DKSZ79]). Although our model will tolerate a degree of imprecision, an alternative would be to ask users to specify belief in a qualitative manner (eg. “this warrant is very strong”). We illustrated a simple use of qualitative measures of warrant strength in the ML Advisor application (Section 6.5). There is evidence that some form of qualitative measures could be appropriate also for the geological application (for example, Appendix C shows that users often express warrant strengths in multiples of 5%).

Another area for future work would be to investigate whether, with time, users’ were able to improve their own precision of strength estimates with time. There is evidence that feedback can help in this [WB83], and the assessment of this in the context of feedback from argumentation would be a valuable test of the approach. In the oil exploration domain, this could be assessed directly by asking users to quantify their degree of uncertainty in the concluded probability of oil (eg. “+/-.5”) and investigating whether this decreased with time. Alternatively, indirect measures as discussed in Section 2.5.6 could be applied. This would assess whether the resolution of users’ judgements was becoming more precise (and thus less qualitative) through argumentation.

Relational Representations of Strength

A related suggestion is to allow the use of relational expressions of belief (eg. “this warrant is more influential than that warrant”, “these two warrants should have the same strength” etc.). Similarly, the system’s comments to the user could be rephrased in relational terms (eg. “but you said the risk here should be greater than there…”). This form of dialogue may prove more natural to users, hiding the absolute warrant strengths from them. Given a description of this kind, the system could then seek to assign strengths to warrants in a way consistent with the expressed relationships. Similar techniques were used during the construction of Prospector’s models: the tool KAS [Reb81] was used to find appropriate assignments of LS/LN factors in Prospector’s models such that correct results on a set of training examples were obtained.
Computing Models of Opinions

A final development would be to compute the model of a user’s opinion from the arguments which he or she had constructed. For example, the strength of a warrant could be computed by taking some average of its strengths in the user’s arguments. An initial set of arguments would be required to initiate this process. Two advantages of this are:

1. The model may reflect the user’s actual judgements more accurately. At present, the user can state a particular belief in the model (i.e. assign a particular strength to a warrant), but in practice always over-ride that warrant when it applies without changing the model. Here, the user may not realise that there are so many exceptions to the warrant that, in fact, it does not reflect their general opinion at all. Computing the model from arguments would avoid these discrepancies.

2. At present, users can have several models reflecting their opinions in different parts of the problem domain. If a single model was computed as needed, however, this partitioning of the problem domain could be avoided. Instead, a metric of the relevance of existing arguments to the current problem could be encoded, and the model of opinion computed from those arguments, weighted by relevance.

As a result, models would not need to be explicitly stored but instead generated at run-time. For efficiency, only the parts of the model required for the current problem would need to be generated (‘lazy generalisation’ [Aha90], p4). This is a move to a case-based model of a user’s opinion. A corresponding limitation, though, would be that there would no longer be place for users to express non-case-specific opinions.

6.7.4 Unconstrained Use of Strengths and Qualifiers

Our implementation constrained the use of strengths and qualifiers in arguments (Section 4.7.3). This was done to keep arguments simple enough for users to understand and modify them as they desired.

These constraints could be relaxed in other implementations of our model, allowing qualifiers to propagate through arbitrary chains of warrants. However, this may make the arguments more difficult to understand and modify. Further investigation is needed of the benefits and costs of relaxing this constraint.

6.7.5 More Extensive Postmortem Analysis

In the implementation described, we have considered comparison functions to search the database of arguments. However, some arguments may be out of date (superseded by a more recent argument by the same user for a different solution), some may have been proved incorrect (if the argument made a predictive claim which subsequently was found wrong) and others may have been proved correct.

A straightforward but important development of the current implementation would be to account for the different statuses of arguments during search. Additionally, if a predictive argument was later shown wrong, analysis about which parts of that argument were in error could be made given knowledge about what outcome occurred in practice (‘postmortem analysis’).

6.7.6 Personalised Problem Descriptions

Expert opinions about the problem description itself may sometimes differ. This may arise when the problem description includes not only factual data but interpretations of that data – interpretations which may differ. A small implementation extension is to personalise the problem description for different users, requiring them to give justifications for the interpretative parts
of the description they provide. These justifications can be considered as backings for implicit warrants (ie. not in the skeleton set) connecting the raw data with the interpretative descriptors used to describe the case.

6.7.7 Recording the Dynamics of the Argument

Our model stores arguments, constructed interactively with the user, but does not record any of the dynamics of that construction process (eg. which warrants were queried, which changed, which precedents were most influential etc.). An extension of the model would be to also record these dynamics, and for the system to use them for argumentation. Several of the hypertext systems reviewed in Section 2.4 did this, but with no active argumentation element in the systems.

6.8 Summary

In this Chapter, we have discussed the generality of our argumentation model and the requirements for its use. The most important points of conclusion are as follows:

1. The argumentation model meets a requirement common to users in many domains, namely that they be selectively informed about what others have done in similar problems, without being swamped by information. This requirement is met by virtue of the basic processes which the model supports, namely the construction, discussion, and storage of arguments expressed in a common language.

2. There are alternative ways the model can be applied, compared with the specific application presented in Chapter 4. For their successful application, we must preserve the meaningfulness of represented arguments to the users and commonality between arguments to allow their exploitation by the computer. Providing this can be achieved, alternative applications with different characteristics are also possible.

3. The model’s possible application using an alternative, qualitative model of strength has been illustrated by sketching its use in the task of giving software advice (Section 6.5).

4. The model can be further developed in several ways, extending its existing functionality.

Thus we have sought to characterise the model’s generality and its potential for further improvement. We now return to the wider context of AI research, from where we started in Chapter 1, to discuss the argumentation model’s contributions to the field.
Chapter 7

Conclusion

7.1 Summary of the Thesis

The work in this thesis was originally motivated by the difficulties encountered in applying expert system techniques to geological risk assessment. In response to these difficulties, we have developed an approach to problem-solving based on argumentation and applied it to this domain. This approach can be viewed as an alternative to an ‘autonomous expert’ model of problem-solving, in which problem-solving is considered essentially as a cooperative activity based on the interaction of different, possibly conflicting chains of reasoning. It can also be viewed as a natural development of existing expert system technology, in which computational models of explainable reasoning are augmented with methods for justifying conclusions, responding to criticisms and accounting for different and changing opinions.

Our goals were to develop and implement a practical computational model of this process. To achieve this goal we developed a model of argumentation closely related to Toulmin’s ideas, and evaluated an implementation for assisting experts in geological risk assessment. The implementation has been in routine, commercial use for over a year, providing strong evidence of the utility and practicality of our approach. The evaluation showed that in this particular application the model has also helped experts to formulate more consistent arguments for their conclusions.

We have also addressed the issues of the generality and extensibility of our approach by characterising the problems to which it could be applied. This characterisation suggests the model could be used in a wide variety of domains in addition to that of risk assessment, the application domain of our implementation.

We now return to the wider context of AI research, to discuss the place and contribution of our work to the field.

7.2 Contributions of the Work

7.2.1 Expert System Technology

The field of expert systems is still rapidly progressing in many ways. New developments include the use of improved knowledge representation schemes, the incorporation of ‘deep’ or ‘causal’ domain models into reasoning, integration with database technology and improved human-computer interaction techniques. The work presented here can also be seen as a practical development of existing techniques but along a different dimension, namely that of modelling the eristic or argumentative problem-solving among cooperating experts, and providing computational techniques for incorporating machines in this process. The issue is not so much the desirability of supporting argumentation but of how techniques for this can be sufficiently constrained to make such support practical while still being useful. Our model offers one solution
7.2. CONTRIBUTIONS OF THE WORK

to this.

The model itself is straightforward, using a simple rule-like knowledge representation scheme (warrants). This simplicity allows the model to be comfortably implemented with current knowledge-based system technology.

This work thus contributes to expert system technology, providing a model with which the computer can usefully assist in cooperative, eristic problem-solving among experts. In particular, the model allows users to maintain a system’s knowledge themselves in a practical but constrained way. This contribution is important, as practical knowledge maintenance remains a significant problem for expert systems.

7.2.2 Case-Based Reasoning

The storage of arguments within an implementation of our model leads naturally to case-based reasoning operations being used during argumentation (eg. similarity assessment, argument retrieval and learning by acquiring new cases). This work contributes to research in case-based reasoning in the following ways.

First, our model augments case descriptions (the grounds) with models of reasoning about those descriptions (the arguments). This provides a knowledge-rich way to identify relevant cases, and contrasts with statistical methods sometimes used in case-based reasoning eg. [Aha89, Aha90]. A model of opinion can be seen as a personalised domain theory, containing a dynamic component controllable by the user (the warrant strengths). This work thus contributes a practical theory of how dynamic and encoded similarity assessment methods can be combined for case-based reasoning. The routine use of the implementation in industry illustrates the value and practicality of such a combination.

This work also relates Toulmin’s ideas about argumentation to case-based reasoning. The significance of this is that it offers a means of integrating rule-based and case-based reasoning together while maintaining flexibility. In case-based reasoning, the desire to incorporate general domain knowledge (eg. rules) can conflict with the desire not to restrict learning by introducing a fixed, static body of knowledge. Toulmin’s approach alleviates this conflict by regarding rules as argument components which may be modified by case evidence. Our model provides a practical embodiment of Toulmin’s ideas.

Finally, Optimist can be seen as a mature implementation of a case-based tutoring system (Bareiss et al. discuss this concept in [BFF91]). The basic process of selectively informing users about what others have done in similar, previous problems can be used to tutor new users in a domain. The functions for argumentation allow a sustained, mixed initiative dialogue to follow specific points of interest.

7.2.3 Knowledge Acquisition

This work offers a partial step forward in the area of automated knowledge acquisition. The argumentation model allows users to maintain knowledge without intervention of a programmer, but achieves this by constraining the ways this can be performed. Because of these constraints, we prefer to call this process ‘knowledge maintenance’ rather than knowledge acquisition.

In this research we have explored a particular trade-off between generality and practicality of knowledge acquisition, adopting an approach less general but more practical than other more complex systems for knowledge acquisition (eg. Protos [BPM89]). Our implementation contributes a ‘data point’ to this trade-off, illustrating a point at which practicality is achieved. It also presents challenges for further improving the knowledge acquisition functions it supports while maintaining usability.

Finally, the evaluation supports the view of knowledge acquisition as a modelling rather than extraction process. It is now broadly accepted that the term ‘knowledge acquisition’ is somewhat of a misnomer. Clancey writes
“Knowledge acquisition, in particular, is a process of developing computer models, often for the first time, not a process of transferring or accessing statements or diagrams that are already written down and filed away in an expert’s mind.” p39 [Cla89]

Others have made similar comments. Compton et al. suggest experts are engaged in the following processes:

“Firstly, he [the expert] identifies the correct interpretation for the case, and secondly he justifies this interpretation. The justification...is what the expert communicates. The insight...is not available.” p378 [CHQ+89]

Seen from this perspective, the activity of constructing arguments (both with or without a computer) is a post hoc translation process, translating inarticulatable reasoning into a communicable form. This translation is necessary for exchange of information between people, however it is also error-prone, as people’s ability to ensure judgements are consistently expressed is limited. The argumentation model thus contributes assistance to this process, helping to ensure consistency of expression and language is maintained by exploiting the computer’s strengths of memory and search.

7.3 Closing Remarks

The process of argumentation – the critical exchange and development of knowledge – is a general one, and surely must be a fundamental process in any ‘intelligent’ system. In this thesis we have presented a model of this process, building upon work in related areas of expert systems, case-based reasoning and knowledge acquisition. While the utility of this model has been demonstrated, there are still many ways in can be further extended. It remains to be seen how this area of research develops in the future.
Bibliography


## Glossary Index

<table>
<thead>
<tr>
<th>Term</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>adversarial</td>
<td>49</td>
</tr>
<tr>
<td>appraisal</td>
<td>53, 91</td>
</tr>
<tr>
<td>argument</td>
<td>35</td>
</tr>
<tr>
<td>argumentation</td>
<td>1, 35</td>
</tr>
<tr>
<td>arguments, analytic</td>
<td>42</td>
</tr>
<tr>
<td>arguments, substantial</td>
<td>42</td>
</tr>
<tr>
<td>backing</td>
<td>38</td>
</tr>
<tr>
<td>claim</td>
<td>38</td>
</tr>
<tr>
<td>components</td>
<td>54</td>
</tr>
<tr>
<td>conflict</td>
<td>43</td>
</tr>
<tr>
<td>consensus</td>
<td>49</td>
</tr>
<tr>
<td>consistency</td>
<td>43</td>
</tr>
<tr>
<td>containment</td>
<td>4</td>
</tr>
<tr>
<td>critiquing systems</td>
<td>8</td>
</tr>
<tr>
<td>distributed problem-solving</td>
<td>5</td>
</tr>
<tr>
<td>element, of an argument</td>
<td>38</td>
</tr>
<tr>
<td>eristic problem-solving</td>
<td>1</td>
</tr>
<tr>
<td>field of discourse</td>
<td>36</td>
</tr>
<tr>
<td>grounds</td>
<td>38</td>
</tr>
<tr>
<td>ill-structured</td>
<td>4</td>
</tr>
<tr>
<td>inference</td>
<td>35</td>
</tr>
<tr>
<td>initial presumption</td>
<td>58</td>
</tr>
<tr>
<td>knowledge maintenance</td>
<td>48</td>
</tr>
<tr>
<td>modifier</td>
<td>58</td>
</tr>
<tr>
<td>party</td>
<td>36</td>
</tr>
<tr>
<td>prospect</td>
<td>52</td>
</tr>
<tr>
<td>prospect appraisal</td>
<td>53</td>
</tr>
<tr>
<td>qualifier</td>
<td>38</td>
</tr>
<tr>
<td>reasoning</td>
<td>35</td>
</tr>
<tr>
<td>rebuttal</td>
<td>38</td>
</tr>
<tr>
<td>risk components</td>
<td>54</td>
</tr>
<tr>
<td>skeleton warrant</td>
<td>40</td>
</tr>
<tr>
<td>special warrants</td>
<td>62</td>
</tr>
<tr>
<td>strength</td>
<td>40</td>
</tr>
<tr>
<td>structure, of an argument</td>
<td>40</td>
</tr>
<tr>
<td>tree, representing an argument</td>
<td>40</td>
</tr>
<tr>
<td>warrant</td>
<td>38</td>
</tr>
</tbody>
</table>
Index

adversarial 49
argument 35, 40, 44
  construction 67
  content 41
  structure 37, 40, 48, 59
  tree 40
argumentation 1, 6, 11, 35, 48, 67, 70
backings 38, 42, 105, 130
case representation 47, 135
case-based reasoning 10, 42, 47, 48, 70, 128, 132, 135
case-based tutoring 135
case 38, 42
conflict resolution 47, 49
consensus 49
consistency 43, 99
containment of knowledge 4
cooperative systems 3
critiquing systems 8
decision-support systems 5, 24
determinations 50
distributed problem-solving 3, 5
endorsements 50
eristic problem-solving 1, 3, 5, 6
expert systems 2, 3, 128, 134
explanation-based learning 42
field of discourse 36
financial marketing 12
geology 1, 10, 52
grounds 38, 42, 62
hypertext 27, 48
ill-structured domains 4
inconsistency 99
inference 35
Internist 2
justification 10, 47, 48
knowledge acquisition 9, 47, 50, 95, 105, 135
knowledge maintenance 47, 50, 95
knowledge modelling 105, 135
learning apprentices 9
legal reasoning 13
machine learning 9, 70, 122
model, of opinion 40, 44, 63, 66
multi-expert systems 9
Mycin 2, 128
oil exploration 14, 52
Oncocin 8
open textured semantics 41
open-ended representations 46, 130
Optimist 52
parallel processing 5
party, in an argument 36
planning 3
precedes 42, 70
Prospector 2, 9, 11, 17, 58
Protos 9, 50
qualifier 38, 57, 62, 132
qualitative modelling 3
rebuttal 38, 39
relevance 40, 50, 63
ripple-down rules 9, 50, 130
risk assessment 1, 10, 53
similarity assessment 48, 63, 70, 71
skeleton warrants 40, 45, 50, 63, 119
special warrants 62, 71
specificity 49
strength, of a warrant 40, 57, 62, 132
Teiresias 9
terminology 41
Toulmin 38, 40, 135
truth maintenance 49
warrant 42, 63
  logical 59
  grouping 66
XCon 2
Appendix A

Data Structures for Storing Arguments

The data structures used to store an argument in Optimist are most easily described by an example. Figure A.1 shows an argument tree, simplified in extent but illustrating all the essential structure. This tree would be represented in Optimist as follows.

![Argument Tree Diagram]

Figure A.1: An example argument.

1. The qualifications of the main claim (oil present) and the sub-claims in the two top logical warrants are stored. Strictly this information is redundant as the qualifications can be recalculated from the rest of the tree, but it simplifies and speeds up search when scanning arguments.

<table>
<thead>
<tr>
<th>Claim</th>
<th>Qualification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oil is present</td>
<td>0.18</td>
</tr>
<tr>
<td>Trap is present</td>
<td>0.50</td>
</tr>
<tr>
<td>Reservoir is present</td>
<td>0.70</td>
</tr>
<tr>
<td>Gross is adequate</td>
<td>0.70</td>
</tr>
<tr>
<td>Net is adequate</td>
<td>1.00</td>
</tr>
<tr>
<td>Source is present</td>
<td>0.50</td>
</tr>
</tbody>
</table>
2. The warrants with user-tunable strengths that were applied are stored. ‘Forced?’ implies that the user has changed the warrant strength for this particular argument only, leaving the strength in the model that was applied unchanged. The user would do this when the warrant strength in the model correctly reflected his general belief about the grounds influence on the claim, but when there was also some prospect-specific anomaly that led him or her to consider this general influence not to hold in this particular case. The anomaly would be described in the backing.

<table>
<thead>
<tr>
<th>Warrant</th>
<th>Claim</th>
<th>Strength</th>
<th>Backing</th>
<th>Forced?</th>
</tr>
</thead>
<tbody>
<tr>
<td>w1</td>
<td>Trap present</td>
<td>0.90</td>
<td>“18th round showed effective”</td>
<td>no</td>
</tr>
<tr>
<td>w7</td>
<td>Trap present</td>
<td>0.55</td>
<td>“Rietger’s theory applicable here”</td>
<td>yes</td>
</tr>
<tr>
<td>w11</td>
<td>Sorc present</td>
<td>0.50</td>
<td>“Proven Volgian field 100km south”</td>
<td>no</td>
</tr>
</tbody>
</table>

3. The special warrants are also stored. The warrant for ‘Gross’, for instance, claims that there will be sufficient gross thickness of reservoir in the prospect. These special warrants also have an associated value (eg. for gross, the expected gross thickness). Two types of backing are stored: firstly, a pointer to the statistical prediction method and plot (if any), and secondly a text comment from the user.

Plots are stored in a plot database, and thus can be redisplayed when viewing the argument.

<table>
<thead>
<tr>
<th>Claim</th>
<th>Strength</th>
<th>Assoc. Value</th>
<th>Backing</th>
<th>Additional Text Backing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sufficient gross</td>
<td>0.70</td>
<td>100ft</td>
<td>plot32, histogram</td>
<td>“Mean of histogram”</td>
</tr>
<tr>
<td>Sufficient net</td>
<td>1.00</td>
<td>80ft</td>
<td>plot71, interpolation</td>
<td>“Good correlation”</td>
</tr>
</tbody>
</table>

4. The data used for these statistical predictions comes from a set of relevant wells, extracted from a well database by Optimist. Optimist stores the list of relevant wells used and their degree of relevance. The relevance of each well is in fact the qualification to the claim ‘this well is relevant’, found by generating a relevance argument for each well using a set of relevance warrants. These relevance arguments are not stored.

<table>
<thead>
<tr>
<th>Well</th>
<th>Relevance</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/3-b</td>
<td>1.00</td>
</tr>
<tr>
<td>w/3-c</td>
<td>0.95</td>
</tr>
<tr>
<td>s/76-d</td>
<td>0.50</td>
</tr>
</tbody>
</table>
Appendix B

A Simple Profit-and-Loss Model

We present a simple profit-and-loss model to investigate how errors in predicted oil probability affect profits.

In this model, we assume a distribution of prospect sizes given by the formula:

\[ p(s) = \text{Probability of Prospect Size } s = k e^{-ks} \]

Where \( k \) is a constant. Thus small prospects are common, but large ones are rarer as reflected in a normal geological environment. We also assume the cost \( c \) of drilling a prospect is a constant, and that the financial gain from a prospect with oil in is proportional to its size \( s \).

Thus, given a prospect of size \( s \), with probability \( p_{oil} \) of containing oil:

\[
\begin{align*}
\text{Cost} &= c \\
\text{Gain} &= s \, p_{oil}
\end{align*}
\]

and thus it is cost effective to drill prospects where \( Gain > Cost \), i.e. \( s > c/p_{oil} \).

Given a geologist’s estimate of the probability of oil, \( p_{est} \), it is thus considered beneficial to drill the well if it’s size \( s > c/p_{est} \). If we drill all such wells, the average cost and gains per well will be:

\[
\begin{align*}
\text{Cost} &= \int_{c/p_{est}}^{\infty} c \, p(s) \, ds \\
&= c \, e^{-ck/p_{est}} \\
\text{Gain} &= \int_{c/p_{est}}^{\infty} s \, p_{oil} \, p(s) \, ds \\
&= k \, p_{oil} \left[ \frac{1}{k^2} \left( -ks - 1 \right) e^{-ks} \right]_{c/p_{est}}^{\infty} \\
&= \left( \frac{c \, p_{oil}}{p_{est}} + \frac{p_{oil}}{k} \right) e^{-ck/p_{est}} \\
\text{Profit} &= G ain - C ost \\
&= \left( \frac{c \, p_{oil}}{p_{est}} + \frac{p_{oil}}{k} - c \right) e^{-ck/p_{est}}
\end{align*}
\]

Differentiating with respect to \( p_{est} \) shows \( Profit \) is maximum when \( p_{est} = p_{oil} \), as expected. The relationship between \( Profit \), \( p_{est} \) and \( p_{oil} \) for \( k = c = 1 \) is shown in Figure B.1.
Figure B.1: How the difference in estimated and actual oil probability affects profit.
Appendix C

Summary of Appraisals

Tables C.1, C.2 and C.3 summarise the appraisal arguments within Optimist. Each line is a separate appraisal, showing which warrants occurred in the appraisal argument and their strengths in the argument. The last four warrants in Table C.2 are the special warrants, generated at run-time for each appraisal (see Chapter 4.7.4). Nine other warrants, not shown here, have never had their grounds satisfied in any appraised prospect so far. Table C.3 gives the prospect number which the appraisal concerns, the author and the appraisal’s creation date.

Appraisal numbers in brackets indicate out-of-date appraisals, ie. those where a more recent appraisal was later made for the same prospect by the same user with Optimist and saved as a new appraisal (as opposed to updating the old appraisal).

The overall probability in Table C.2 is approximately the product of all the warrant strengths (strengths are expressed as percentages here). Deviations from this are firstly because a warrant may be applied more than once in the same argument, and secondly for arguments 6, 25, 27, 33, 53 and 58 the user altered a qualifier of a claim within the argument rather than a warrant strength.

In the three Tables, each line corresponds to an appraisal argument (Arg No. is its code number). In Tables C.1 and C.2, the line marked ‘Default’ shows the default strength of warrants in the system’s default model. ‘Overall Prob.’ is the concluded probability of oil (the qualifier to the claim ‘oil is present’).

Table C.3 shows, for each appraisal, the prospect it concerns, the owner and the appraisal’s creation date (also given as number of months since logging of usage began, eg. Dec-89 was month 1 of logging).
<table>
<thead>
<tr>
<th>Arg No.</th>
<th>Warrant number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Default</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>90</td>
</tr>
<tr>
<td>4</td>
<td>85</td>
</tr>
<tr>
<td>6</td>
<td>90</td>
</tr>
<tr>
<td>7</td>
<td>94</td>
</tr>
<tr>
<td>8</td>
<td>94</td>
</tr>
<tr>
<td>9 (9)</td>
<td>50</td>
</tr>
<tr>
<td>10</td>
<td>94</td>
</tr>
<tr>
<td>11</td>
<td>90</td>
</tr>
<tr>
<td>12</td>
<td>90</td>
</tr>
<tr>
<td>13 (13)</td>
<td>90</td>
</tr>
<tr>
<td>15</td>
<td>90</td>
</tr>
<tr>
<td>16</td>
<td>100</td>
</tr>
<tr>
<td>17 (17)</td>
<td>85</td>
</tr>
<tr>
<td>18</td>
<td>85</td>
</tr>
<tr>
<td>19</td>
<td>51</td>
</tr>
<tr>
<td>20</td>
<td>85</td>
</tr>
<tr>
<td>22</td>
<td>85</td>
</tr>
<tr>
<td>23 (23)</td>
<td>85</td>
</tr>
<tr>
<td>24</td>
<td>85</td>
</tr>
<tr>
<td>25</td>
<td>85</td>
</tr>
<tr>
<td>26 (26)</td>
<td>51</td>
</tr>
<tr>
<td>27 (27)</td>
<td>94</td>
</tr>
<tr>
<td>32</td>
<td>94</td>
</tr>
<tr>
<td>33</td>
<td>94</td>
</tr>
<tr>
<td>34</td>
<td>90</td>
</tr>
<tr>
<td>35 (35)</td>
<td>85</td>
</tr>
<tr>
<td>36</td>
<td>85</td>
</tr>
<tr>
<td>51</td>
<td>85</td>
</tr>
<tr>
<td>52</td>
<td>50</td>
</tr>
<tr>
<td>53</td>
<td>85</td>
</tr>
<tr>
<td>54</td>
<td>70</td>
</tr>
<tr>
<td>55</td>
<td>85</td>
</tr>
<tr>
<td>56</td>
<td>85</td>
</tr>
<tr>
<td>57</td>
<td>85</td>
</tr>
<tr>
<td>58</td>
<td>50</td>
</tr>
<tr>
<td>67</td>
<td>85</td>
</tr>
<tr>
<td>68 (68)</td>
<td>51</td>
</tr>
<tr>
<td>69</td>
<td>51</td>
</tr>
<tr>
<td>74</td>
<td>85</td>
</tr>
<tr>
<td>77</td>
<td>70</td>
</tr>
<tr>
<td>78</td>
<td>51</td>
</tr>
<tr>
<td>79</td>
<td>85</td>
</tr>
<tr>
<td>84 (84)</td>
<td>85</td>
</tr>
<tr>
<td>85</td>
<td>85</td>
</tr>
<tr>
<td>87</td>
<td>85</td>
</tr>
<tr>
<td>88</td>
<td>85</td>
</tr>
</tbody>
</table>

Table C.1: The strengths of warrants (1-17) occurring in appraisal arguments in Optimist.
<table>
<thead>
<tr>
<th>Arg No.</th>
<th>Warrant number</th>
<th>Special warrants</th>
<th>Overall Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>18 20 21 22 25 27 29 30 S1 S2 S3 S4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default</td>
<td>35 95 50 20 20 100 100 100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>95 100</td>
<td>100 100 100 100 85</td>
<td>72</td>
</tr>
<tr>
<td>4</td>
<td>95 100</td>
<td>100 100 100 70</td>
<td>56</td>
</tr>
<tr>
<td>6</td>
<td>33 0</td>
<td>100 100 75 50</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>95 100</td>
<td>100 100 82 89 80</td>
<td>41</td>
</tr>
<tr>
<td>8</td>
<td>70 74</td>
<td>100 100 75 91</td>
<td>7</td>
</tr>
<tr>
<td>(9)</td>
<td>35 100</td>
<td>100 100 86 90</td>
<td>8</td>
</tr>
<tr>
<td>10</td>
<td>95 100</td>
<td>100 100 82 89 80</td>
<td>52</td>
</tr>
<tr>
<td>11</td>
<td>95 100</td>
<td>100 80 90 24</td>
<td>14</td>
</tr>
<tr>
<td>12</td>
<td>95 100</td>
<td>100 80 90 25</td>
<td>15</td>
</tr>
<tr>
<td>(13)</td>
<td>95 100</td>
<td>70 84 33 90</td>
<td>8</td>
</tr>
<tr>
<td>15</td>
<td>95 100</td>
<td>75 70 93 74</td>
<td>17</td>
</tr>
<tr>
<td>16</td>
<td>100 100</td>
<td>100 100 100 35</td>
<td>2</td>
</tr>
<tr>
<td>(17)</td>
<td>95 100</td>
<td>100 100 75 89</td>
<td>19</td>
</tr>
<tr>
<td>18</td>
<td>95 100</td>
<td>100 100 75 89</td>
<td>11</td>
</tr>
<tr>
<td>19</td>
<td>95 100</td>
<td>90 90 76 68</td>
<td>14</td>
</tr>
<tr>
<td>20</td>
<td>95 100</td>
<td>75 50 100 50</td>
<td>12</td>
</tr>
<tr>
<td>22</td>
<td>100 100</td>
<td>100 90 74 70</td>
<td>11</td>
</tr>
<tr>
<td>(23)</td>
<td>100 100</td>
<td>85 40 90 80</td>
<td>7</td>
</tr>
<tr>
<td>24</td>
<td>100 100</td>
<td>85 40 90 80</td>
<td>7</td>
</tr>
<tr>
<td>25</td>
<td>75 100</td>
<td>95 50 100 50</td>
<td>6</td>
</tr>
<tr>
<td>(26)</td>
<td>95 100</td>
<td>95 100 95 71</td>
<td>13</td>
</tr>
<tr>
<td>(27)</td>
<td>95 100</td>
<td>50 75 90 55</td>
<td>12</td>
</tr>
<tr>
<td>32</td>
<td>95 100</td>
<td>50 50 95 99</td>
<td>4</td>
</tr>
<tr>
<td>33</td>
<td>95 100</td>
<td>50 75 90 55</td>
<td>12</td>
</tr>
<tr>
<td>34</td>
<td>95 100</td>
<td>75 90 99 70</td>
<td>9</td>
</tr>
<tr>
<td>(35)</td>
<td>35 95 100</td>
<td>50 100 95 100</td>
<td>3</td>
</tr>
<tr>
<td>36</td>
<td>50 95 100</td>
<td>50 100 85 91</td>
<td>7</td>
</tr>
<tr>
<td>51</td>
<td>95 100</td>
<td>100 95 95 90</td>
<td>11</td>
</tr>
<tr>
<td>52</td>
<td>95 100</td>
<td>100 50 90 81</td>
<td>14</td>
</tr>
<tr>
<td>53</td>
<td>95 100</td>
<td>75 70 93 100</td>
<td>20</td>
</tr>
<tr>
<td>54</td>
<td>95 100</td>
<td>85 70 65 75</td>
<td>6</td>
</tr>
<tr>
<td>55</td>
<td>95 100</td>
<td>100 75 90 75</td>
<td>8</td>
</tr>
<tr>
<td>56</td>
<td>95 100</td>
<td>100 100 86 83</td>
<td>19</td>
</tr>
<tr>
<td>57</td>
<td>95 100</td>
<td>100 95 85 79</td>
<td>13</td>
</tr>
<tr>
<td>58</td>
<td>95 100 100</td>
<td>89 81 100 95</td>
<td>29</td>
</tr>
<tr>
<td>67</td>
<td>100 100</td>
<td>95 80 90 100</td>
<td>13</td>
</tr>
<tr>
<td>(68)</td>
<td>95 100</td>
<td>75 45 95 85</td>
<td>3</td>
</tr>
<tr>
<td>(69)</td>
<td>95 100</td>
<td>55 85 95 50</td>
<td>8</td>
</tr>
<tr>
<td>74</td>
<td>95 100</td>
<td>80 67 87 54</td>
<td>14</td>
</tr>
<tr>
<td>77</td>
<td>95 100</td>
<td>95 50 95 50</td>
<td>12</td>
</tr>
<tr>
<td>78</td>
<td>95 100</td>
<td>75 45 95 85</td>
<td>3</td>
</tr>
<tr>
<td>79</td>
<td>95 100</td>
<td>90 60 80 75</td>
<td>12</td>
</tr>
<tr>
<td>(84)</td>
<td>95 100</td>
<td>95 95 90 100</td>
<td>45</td>
</tr>
<tr>
<td>85</td>
<td>95 100</td>
<td>90 80 75 90</td>
<td>5</td>
</tr>
<tr>
<td>87</td>
<td>65 95 100</td>
<td>70 75 75 89</td>
<td>9</td>
</tr>
<tr>
<td>88</td>
<td>65 95 100</td>
<td>90 75 70 90</td>
<td>11</td>
</tr>
</tbody>
</table>

Table C.2: The strengths of warrants (18 onwards) occurring in appraisal arguments in Optimist.
<table>
<thead>
<tr>
<th>Arg No.</th>
<th>Prospect</th>
<th>User</th>
<th>Initially created</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>p7</td>
<td>C</td>
<td>25-Oct-89 (pre-log)</td>
</tr>
<tr>
<td>4</td>
<td>p7</td>
<td>B</td>
<td>25-Oct-89 (pre-log)</td>
</tr>
<tr>
<td>6</td>
<td>p9</td>
<td>D</td>
<td>26-Oct-89 (pre-log)</td>
</tr>
<tr>
<td>7</td>
<td>p8</td>
<td>A</td>
<td>26-Oct-89 (pre-log)</td>
</tr>
<tr>
<td>8</td>
<td>p9</td>
<td>B</td>
<td>27-Oct-89 (pre-log)</td>
</tr>
<tr>
<td>(9)</td>
<td>p10</td>
<td>B</td>
<td>27-Oct-89 (pre-log)</td>
</tr>
<tr>
<td>10</td>
<td>p8</td>
<td>B</td>
<td>27-Oct-89 (pre-log)</td>
</tr>
<tr>
<td>11</td>
<td>p11</td>
<td>E</td>
<td>21-Nov-89 (pre-log)</td>
</tr>
<tr>
<td>12</td>
<td>p11</td>
<td>B</td>
<td>21-Nov-89 (pre-log)</td>
</tr>
<tr>
<td>(13)</td>
<td>p12</td>
<td>B</td>
<td>8-Dec-89 (1)</td>
</tr>
<tr>
<td>15</td>
<td>p12</td>
<td>D</td>
<td>14-Dec-89 (1)</td>
</tr>
<tr>
<td>16</td>
<td>p10</td>
<td>B</td>
<td>23-Jan-90 (2)</td>
</tr>
<tr>
<td>(17)</td>
<td>p13</td>
<td>B</td>
<td>24-Jan-90 (2)</td>
</tr>
<tr>
<td>18</td>
<td>p13</td>
<td>B</td>
<td>24-Jan-90 (2)</td>
</tr>
<tr>
<td>19</td>
<td>p14</td>
<td>B</td>
<td>25-Jan-90 (2)</td>
</tr>
<tr>
<td>20</td>
<td>p6</td>
<td>B</td>
<td>31-Jan-90 (2)</td>
</tr>
<tr>
<td>22</td>
<td>p15</td>
<td>B</td>
<td>4-Apr-90 (5)</td>
</tr>
<tr>
<td>(23)</td>
<td>p16</td>
<td>B</td>
<td>14-May-90 (6)</td>
</tr>
<tr>
<td>24</td>
<td>p16</td>
<td>B</td>
<td>14-May-90 (6)</td>
</tr>
<tr>
<td>25</td>
<td>p17</td>
<td>B</td>
<td>14-May-90 (6)</td>
</tr>
<tr>
<td>(26)</td>
<td>p18</td>
<td>C</td>
<td>31-Jul-90 (8)</td>
</tr>
<tr>
<td>(27)</td>
<td>p19</td>
<td>C</td>
<td>31-Jul-90 (8)</td>
</tr>
<tr>
<td>32</td>
<td>p18</td>
<td>C</td>
<td>1-Aug-90 (9)</td>
</tr>
<tr>
<td>33</td>
<td>p19</td>
<td>C</td>
<td>1-Aug-90 (9)</td>
</tr>
<tr>
<td>34</td>
<td>p20</td>
<td>C</td>
<td>3-Aug-90 (9)</td>
</tr>
<tr>
<td>(35)</td>
<td>p21</td>
<td>B</td>
<td>30-Oct-90 (11)</td>
</tr>
<tr>
<td>36</td>
<td>p21</td>
<td>B</td>
<td>30-Oct-90 (11)</td>
</tr>
<tr>
<td>51</td>
<td>p31</td>
<td>B</td>
<td>21-Jan-91 (14)</td>
</tr>
<tr>
<td>52</td>
<td>p32</td>
<td>B</td>
<td>21-Jan-91 (14)</td>
</tr>
<tr>
<td>53</td>
<td>p12</td>
<td>B</td>
<td>22-Jan-91 (14)</td>
</tr>
<tr>
<td>54</td>
<td>p33</td>
<td>B</td>
<td>22-Jan-91 (14)</td>
</tr>
<tr>
<td>55</td>
<td>p34</td>
<td>B</td>
<td>22-Jan-91 (14)</td>
</tr>
<tr>
<td>56</td>
<td>p35</td>
<td>B</td>
<td>22-Jan-91 (14)</td>
</tr>
<tr>
<td>57</td>
<td>p36</td>
<td>B</td>
<td>22-Jan-91 (14)</td>
</tr>
<tr>
<td>58</td>
<td>p37</td>
<td>B</td>
<td>22-Jan-91 (14)</td>
</tr>
<tr>
<td>67</td>
<td>p42</td>
<td>C</td>
<td>11-Mar-91 (16)</td>
</tr>
<tr>
<td>(68)</td>
<td>p43</td>
<td>C</td>
<td>11-Mar-91 (16)</td>
</tr>
<tr>
<td>(69)</td>
<td>p41</td>
<td>C</td>
<td>11-Mar-91 (16)</td>
</tr>
<tr>
<td>74</td>
<td>p47</td>
<td>C</td>
<td>13-Mar-91 (16)</td>
</tr>
<tr>
<td>77</td>
<td>p41</td>
<td>C</td>
<td>15-Mar-91 (16)</td>
</tr>
<tr>
<td>78</td>
<td>p43</td>
<td>C</td>
<td>15-Mar-91 (16)</td>
</tr>
<tr>
<td>79</td>
<td>p46</td>
<td>C</td>
<td>21-Mar-91 (16)</td>
</tr>
<tr>
<td>(84)</td>
<td>p50</td>
<td>F</td>
<td>22-May-91 (18)</td>
</tr>
<tr>
<td>85</td>
<td>p50</td>
<td>F</td>
<td>22-May-91 (18)</td>
</tr>
<tr>
<td>87</td>
<td>p51</td>
<td>F</td>
<td>4-Jun-91 (19)</td>
</tr>
<tr>
<td>88</td>
<td>p52</td>
<td>F</td>
<td>26-Jun-91 (19)</td>
</tr>
</tbody>
</table>

Prospect number denotes the prospect appraised, users are denoted by A to F. The initial creation date is also given as the equivalent month number in the system log. (Note that these dates are not the only 'login' dates: geologists also use the system to view, compare, and update appraisal arguments).

Table C.3: Additional information about the appraisals.
Appendix D

Backings Provided by Users

This appendix lists the backings provided by users for appraisals and models of opinions in Optimist. Names have been changed to preserve company confidentiality. A number in brackets indicates the backing occurred in more than one appraisal/model.

D.1 Backings Used in Appraisal Arguments

5 dry wells in e-t/rb !! (2)
50% risk for unknown shows is too low!!
Closure has element of dip so improving eff.
Dagaland sand regionally has good porosity.
Dip closure is slightly less than perfect here.
Downthrowing against Moravian Basement.
Downthrowing against Moravian basement.
Due to close vertical proximity to Proximal source. (2)
Facies = closure is a comb. of sand pinchout & dip. (2)
Facies applies to upside case/Dip closure applies in the most likely case. (2)
Facies works at Washton/Shalian
Fairly confident that Lower Midian Sediments are present.
Fault seal evidence from Baries/Washton/Berx (2)
Garath says so !! (2)
Gas has migrated this far north as evidenced by offset wells ,particularly ab/a-2b.
Good offset well tie of Mideric into block,
Good porosity in fg/1-s
Guilar has facies closure 25 kms away
I’ve changed this trap direction to be a trap.
I’ve changed this trap direction to be mean upthrown by a normal fault.
If net reservoir is sandy matrix then this is a best guess at expected porosity from offset welldata.
If net reservoir present then this porosity can be expected,
If reservoir present then should have fairly high N/G
If sand is present it’s likely to be OK
If sand present then will be of good reservoir quality. (2)
If sands there it will be porous !!
Initial value too high
Initially too high (2)
Initially too low a value. As good as upthrown normal
Lack of well tie data for Guiler sand.
Lack of well tie into block means i’m not sure if the sand has been deposited there.
Lack of well tie into block, Nearest well control in block 998/-. 

156
Lack of well tie into block, unsure about sand distribution.
MDinian present in all wells in area
Madeleine here is quite deeply buried (2)
Madeleine is most prolific N Sea reservoir
Madeleine porosity proven in York/Down/Washton
Madeleine present in all wells in store
Most Madeleine fields are tilted normal fault blocks
Mud prone Jurine fan’s round here
No L.Bat fields north of Moreal Ground Graben
No data
No relevant data (2)
No well control tying into seismic anomaly
No wells with Madeleine sand absent
Not proven in QWE9
Offset well data suggests Filistair only improves (porosity) with depth. Also Kh are rather poor.
Offset well data suggests this will be expected porosity.
Oil present in ab/3-4
Oil present in ab/3-d & we/2b-z
Oil tested from effective por in rd/t-R
Oil tested in rd/e-t
Poor porosity preservation at 150000 ft
Poor seismic evidence only
Present in E. Yemen & Saudi (2)
Present in all wells
Present in all wells in store (2)
Prospect’s in proven Madeleine Play
Rarely preserved below 110500’
Reasonably confident a rock package has been mapped at this level.
Requires Madeleine seal to fail!!
Reservoir and source are interbedded.
SEE PLOT (2)
Sand not proven on Mideristic High
Sealing faults may work at Washlon & Marvan (4)
Seismic OK, corelation uncertain
Shallow depth of burial
Simian gas field 40km north, also Tertiary
Simian gas field has same source
Some net sand in all wells
Source rock is known to be present and mature, proven by nearby wells.
Tober is regionally present
Updip of oil, downturn of water !!
Upthrows G.Hub. claystones.
Well defined seismic anomaly at Simian level
a/b-s & d had oil
adjacent to oil fields
always risk with downthrown prospects
asdf
based on regional depth vs porosity graph and local well control (2)
because graham woods said so (3)
because i say so
because im not reaaly sure
certain (4)
chalk at crest of structure is faulted
congrations have to be sandy matrix to be good reservoir.
consider this seal to work in region, i.e. Kivia Feild.
depth somain
depth somain has poor poro-perm
discoveries in area indicate presence of source
dodgy first value
down dip of er/s-e
dowthrown closures are leaky (2)
downthrown prospect undrilled (2)
evidence from garavian sands
facies didn’t seal in df/s-s
facies sealed by base moravinan (4)
fault sealed by base alang (8)
faulting is pre-migration (12)
geologist guess
good seismic evidence
good well control (2)
good well evidence for communication with source
gut feeling that m fabriian reservoir will be present
gut feeling!
hot line fit (3)
i am confident that a foroan contamiing structure is present based on local well data and seismic
i’ve checked offset wells/Mirian is a major fm in area
i’ve got an oil well to the southeast of the prospect and a well with minor shows to the northwest so the
risk of actually communicating should be minimal
if duriasity present then likely to have porosity = 18.5 if horian present then likely to have n/g = 50 if sand
present then good n/g inadequate data
initial value too high (16)
initially too low for new basin
lack of seismic well tie + hodian sand likely to pinchout northwards.
local discoveries indicate mature source for gas/cond
local discoveries suggest favian fault seal
local well control and value justification also depth should not be too much of a problem
low risk as value based on sc/2-1 this value is also consistent with the change on going along the ridge
containing the prospect to sc/12-2b (2)
low risk based on local well control and on-prospect well (2)
major faults break surface
mapped thick on seismic
marial sand regionally has good porosity at this depth (2)
mele production in cardi
mirian is usually porous (2)
more than one potential source
more than one potential source
most wells have some pay
near-by well (2)
near-by well info (2)
near-by wells drilled off-structure/see dd/as-22
nearby discoveries indicate presence of source
nearby to re/er-1
net in most wells
no effective seal
no obvious inversion (2)
no proven migration past boundary fault
no proven this far south (2)
no regional precedent for this seal being effective
no regional precedent to suggest seal is effective
no well data
no well data in Mediterranean Ocean
no wells tie this Rith. thick
not seen on seismic
other Fields in the region have faulting seal age.
pay is in source-rock
possibility of x.vur volc. developed (2)
possibility of x.vurvolc developed
present in all wells on block (2)
present in most wells
proven by df/2-2a (2)
proven by well data (2)
proven in df/s-h (4)
r/3-2 is on-prospect and has proven hydrocarbons (2)
regional data suggests cross-fault seal
regional overpressure (4)
regional reduction in poroperm vs depth (2)
regional risk (14)
regional welldata tells me all sands have porosity
relies on early oil gen. + dubious fault migration. (2)
reservoir is juxtaposed against source
reservoir not actually mapped (3)
reservoir not well defined (5)
reservoir unit not actually seen in nearby wells
risk increased because faults cut base chalk
robin’s too optimistic; source always dodgy
sand pinching out into shale and there is an element of dip closure, thus reducing risk,
sands clearly visible on seismic(see scan) (3)
see ds/2-3, re/r-t oil migration into area (2)
see net vs gross plot (2)
see net: gross plot
see plot (5)
see scatter on plot (2)
see special plot
see value justification i can’t be bothered to type it all out again
see we/r-y
seismic pick /well thickness correlation
seismically well defined (2)
should have good sand development
some Madeline always present
source dead cert (5)
stonking line fit
strike slip faults, no sealing
this prospect undrilled (2)
trinavian field adjacent
trinavian field average
unconformities work in XRT (2)
unknown basin (2)
up-dip wells only have 3600'
updip erosion
upper dinjian thickening suggests golian presence
v low SD (3)
vertical seal may not exist (2)
very small std from this value
well control suggest ave porosity of 98 well control/my knowledge=uncon. sand/v.g good por.
well data=sand+oily area
wells in area have gas/cond in m humeric
wx/91-s dry hole on prospect edge

D.2 Backings In Users’ Models

50 too low (3)
Facies works at Washton/Shalian
Fault seal is proven in Baries/Washton/Berx
Initial value too high
Initially too high
Initially too low a value. As good as upthrown normal
Most Madeline fields are tilted normal fault blocks
Possibility for some H/C generation
Sealing faults may work at Washton & Marvan
This prospect has downthrown Birian mudstones form seal down to most-likly case.
boundary faults (2)
chalk at crest of structure is faulted
d.n. fault in er/e-r sealed
downthrown closures are leaky (6)
facies didn’t seal in df/t-yt (6)
fault seal risk similar for side and midthrown faults (2)
fault sealed by base serbian
faulting is pre-migration (3)
hurvan seal works in ab-2/2 and ab/1-23 (2)
increase (3)
initial value too high (11)
initially too high for new basin
initially too low for new basin (12)
local discoveries suggest mid fault seal
local discoveries suggest up and midthrown seal same
nearby to dr/a-n
no regional precedent for midian seal being effective
no regional precedent to suggest fault seal is effective
poor seismic resolution (6)
proven h/c (4)
regional data indicates up and downthrown risk is same
regional data suggests cross-fault seal
regional overpressure (6)
regional risk (2)
reservoir not well defined (6)
residual oil only
risk increased because faults cut base chalk
source dead cert
unconformities work in XDF (2)
Appendix E

A Domain-Independent Shell

Pseudo-code and Prolog code for a simple argumentation shell are presented in this appendix. Example knowledge-bases for two domains are given in Section E.2.2 and Section E.2.3. A trace of the shell running with these knowledge-bases is given in Sections E.3.1 and E.3.2.

E.1 Pseudo-Code

Data structures used:

- Argument = { CaseName, Qualifier, Claim, UserName, Warrants }
- Model = { Number, Owner, Name, Warrants }
- Warrants = { Warrant1, ..., WarrantN }
- Warrant = { WarrantNo, Grounds, Claim, Backing, Strength }
- Grounds = { Ground1, ..., GroundN }
- Claim, Ground = Atti=Valj
- Case = { CaseName, Data }
- Data = { Att1=Val1, ..., AttN=Valj }

To access an element in a structure, the function lookup is used.

**procedure** go:

Pre-defined, global structures:

- Claim := the overall claim to establish (domain-specific)
- Atts := { Att1, ..., Attn } (for describing a case)
- Cases := { Case1, ..., CaseN } (the case database)
- Models := { Model1, ..., ModelN } (the available models of opinion)
- Arguments := { Argument1, ..., ArgumentN } (previous, stored arguments)

let UserName := login
let CaseName := newcase
let Data := data_entry(CaseName, Atts)
let Model := select_model(Models)
let FinalArgument and UpdatedModel := solve(Claim, Model, Case)
add CaseName and UserName to Argument
add FinalArgument to Arguments
add { CaseName, Data } to Cases
replace Model in Models with UpdatedModel.

**procedure** login returning UserName:

write “what is your name?”
read UserName.

**procedure** newcase returning CaseName:

write “What is the name of the new case you wish to work on?”
read CaseName.
procedure data_entry(CaseName, Atts) returning Data:
    let Data := {}
    forall attributes Att in Atts
        write “What is the value of attribute ” Att “ for case ” CaseName “ ?”
        read Val
        add Att=Val to Data.

procedure select_model(Models) returning CurrModel:
    forall Model in Models
        lookup Number, Owner, Name in Model
        write “Model ” Number “; ” Name “, owned by ” Name
        write “Which model do you wish to use?”
        read MNoToUse
        lookup CurrModel in Models where MNoToUse is in CurrModel.

procedure solve(Claim Model, Case) returning Argument and UpdatedModel:
    let InitialArgument := construct_argument(Claim Model, Case)
    let Argument and UpdatedModel := discuss(InitialArgument, Model).

procedure discuss(Argument, Model) returning UpdatedArg and UpdatedModel:
    present(Argument)
    write “Do you agree, want to challenge an item, or shall I check the argument?”
    read Agreement
    if Agreement = agree
        let UpdatedArg := Argument
        let UpdatedModel := Model
    elseif Agreement = check
        write “Which warrant shall I check?”
        read WarrantNo
        lookup Warrant in Argument where WarrantNo is in Warrant
cite_cases(attacking, Warrant)
        let UpdatedArg and UpdatedModel := discuss(Argument, Model).
    elseif Agreement = disagree
        let MidArgument and MidModel := dispute(Argument, Model)
        let UpdatedArg and UpdatedModel := discuss(MidArgument, MidModel).
    present(InitialArgument):
        forall Warrant in InitialArgument
            lookup WarrantNo, Grounds, Claim, Strength, Backing in Warrant
            write WarrantNo “; ” Grounds “ affected ” Claim “ with strength ”
            Strength “ because ” Backing.

procedure dispute(Argument, Model) returning UpdatedArg and UpdatedModel:
    write “Which warrant to you disagree with?”
    read WarrantNo
    lookup Warrant in Argument where WarrantNo is in Warrant
    lookup Strength in Warrant
    write “The strength of the warrant was ” Strength
cite_cases(defending, Warrant)
    write “Do you still disagree? ”
    read Disagreement
    if Disagreement = yes
        let UpdatedWarrant := respond_to_disagreement_with(Warrant)
        let UpdatedArg := Argument, but with Warrant replaced by
        UpdatedWarrant
        write “Make the change to the model being used also?”
read Answer
if Answer = yes
then let UpdatedModel := Model, but with Warrant replaced
   by UpdatedWarrant
else let UpdatedModel := Model
else let UpdatedArg := Argument
let UpdatedModel := Model.

procedure cite_cases(About, Warrant):
(This procedure uses the global structure Arguments, the set of stored arguments)
lookup WarrantNo in Warrant
for all Argument in Arguments
for all OtherWarrant in Argument where WarrantNo is-in OtherWarrant
if check_relation(Warrant, About, OtherWarrant)
then lookup Owner, CaseName in Argument
lookup Strength, Backing in OtherWarrant
write “but " Owner " thought the strength should be "
   Strength “in case " CaseName “ because " Backing “!

procedure check_relation(Warrant1, About, Warrant2) returning true/false:
lookup Strength1 in Warrant1
lookup Strength2 in Warrant2
if About = defending and |Strength1 - Strength2| ≤ Threshold
then return true
elseif About = attacking and |Strength1 - Strength2| > Threshold
then return true
else return false.
   (Note, choice of Threshold depends on the strength calculus used)

procedure respond_to_disagreement_with(Warrant) returning UpdatedWarrant:
write “What should the strength of this warrant be, then?”
read NewStrength
write “Why?”
read NewBacking
lookup WarrantNo, Grounds, Claim in Warrant
let UpdatedWarrant :=
   { WarrantNo, Grounds, Claim, NewBacking, NewStrength }.

procedure construct_argument(Claim Model, Case) returning Argument:
   let ApplicableWarrants := {}
for all Warrant in Model where Claim is-in Warrant
   if for all Ground in Grounds, establish(Ground, Model, Case)
   then add Warrant to ApplicableWarrants
let Qualifier := combine_strengths_of(ApplicableWarrants)
   (this combination function depends on the strength calculus used)
let Argument := { Qualifier, Claim, ApplicableWarrants }.

procedure establish(Ground, Model, Case) returning true/false:
if Ground is the negation of some other ground Ground2
then if not establish(Ground2) then return true else return false
elseif Ground is-in Data in Case then return true
else let SubClaim := Ground
   if there is a LogicalWarrant in Model where SubClaim is-in
      LogicalWarrant, and, for all SubGround in LogicalWarrant,
      establish(SubGround, Model, Case)
   then return true
else return false.
E.2 Prolog Code

E.2.1 The Argumentation Shell

The following Prolog code implements the basic argumentation shell, constructing arguments subject to the constraints described in Section 4.7.3.

```
% File: shell.pl
% Author: Peter Clark
% Purpose: A simple, domain-independent argumentation shell
% This loosely follows the preceding pseudo-code, although some of the data
% (CaseName, UserName, Data, and Model) are in global rather than local variables.

% The following imported predicates import the domain-specific knowledge:
% :/2, a_case_isa/1, attribute/3, combine_strs/2, explanation/3, model/4,
% text_to_present_overall_fact/3, text_to_present_str/3,
% text_to_present_str_alone/3, top_level_claim/1, w_strength_and_backing/4.

% The following simple predicates are not defined here, but defined in loaded library
% files. Their functionalities should be clear from their use in the code below.
% count/1, intread_atom/2, menu_get/4, nmember/4, read_atom/1, read_atom/2,
% tab/0, yread_atom/1, bagof/0/3, datatimeatom/1, write/1, write/2,
% is_list/1, member/2, nmember/3, \=/2.

:- op(900, xfy, ;).
:- op(890, fy, [skeleton_warrant, logical_warrant, if]).
:- op(880, xfy, then).
:- op(870, xfy, and).
:- op(865, xfy, or).
:- op(860, fy, any).

:- dynamic curr_username/1, curr_case/1, curr_model/1, wm/2, arg_summary/5, arg_branch/2.

% ------------ Top level control
run :- login, go.

go :-
    a_case_isa(Thing),
    writeln("\nTop level options:"),
    writeln("\n=================================\n"),
    writeln("What do you want to do:\n"),
    writeln("\t1. Examine a new %\n", [Thing]),
    writeln("\t2. View arguments for the current %\n", [Thing]),
    writeln("\t3. Generate a new argument for the current %\n", [Thing]),
    writeln("\t4. Quit\n"),
    writeln("\t\tOption -> "),
    read_atom(Option),
    go(Option).

go(1) :- newcase, data_entry, go.
go(2) :- view_arguments, go.
go(3) :- select_model, solve, go.
go(4).

% ------------ Login
login :-
    writeln("What is your name? ")
    read_atom(Who),
    retractall(curr_username(_)),
    assertz(curr_username(Who)).
```
% ------------ Get a new case name

case :-
  a_case_isa(Thing),
  writef("What is the name of the %w you want to work on? ", [Thing]),
  read_atom(C),
  retractall(curr_case(_)),
  assertz(curr_case(C)).

% ------------ Data entry

data_entry :-
  writef("\nData entry:")
  writef("\nnext
")
  curr_case(C),
  wm_get(Case),
  !,
  a_case_isa(Thing),
  writef("(I already know about %w %w)\n", [Thing,Case]).

data_entry :-
  curr_case(C),
  wm_get(C),
  attribute(Att, Options, Question),
  writef(Options),
  get_answer(Options, Val),
  wm_put(C, Att=Val),
  fail.

data_entry.

get_answer(integer, Val) :-
  read(Val),
  integer(Val),
  !.

get_answer(integer, Val) :-
  !,
  writef("ERROR! Please type an integer!\n"),
  get_answer(integer, Val).

get_answer(string, Val) :-
  !,
  read_atom(Val).

get_answer(Options, Val) :-
  is_list(Options),
  !,
  nl,
  menu_get(’, Options, _ Val).

get_answer(ATYPE, _ ) :-
  writef("ERROR! Don’t understand answer type %w!\n", [ATYPE]).

select_model :-
  writef("\nSelect a model of opinion to use:\n")
  writef("\next models:\n")
  list_models,
  writef("\nWhich model do you want to use? ")
  read(MNo),
  retractall( curr_model(_) ),
  assertz( curr_model(MNo) ).

list_models :-
  writef("Available models are as follows:\n")
  model(MNo, Who, Name, Date),
  writef("%w, %w, %w, %w\n", [MNo,Name,Who,Date]),
  fail.
% ------------ Main argumentation control

solve :-
    curr_model(Model),
    top_level_claim(Claim),
    construct(InitialArgument, /*for*/ Claim, Model),
    discuss(InitialArgument, Model, FinalArgument),
    store(FinalArgument).

discuss(Argument, Model, FinalArgument) :-
    write("\nThe current argument is as follows:\n"),
    write("\n================================******/\n"),
    present(Argument),
    write("\nDo you agree, wish to challenge an item, or wish me to check"),
    write(" the argument?\n\t(agree/disagree/check)? "),
    read_atom([agree,disagree,check], Answer),
    discuss1(Answer, Argument, Model, FinalArgument).

discuss1(agree, Argument, Model, FinalArgument) :- !.
    write("Which item do you disagree with? "),
    intread_atom(0-100, N),
    nummember(Warrant, Argument, N, RestArgument),
    explain_parts([Warrant], N),
    discuss_war(Warrant, Model, RevisedWarrant),
    nummember(RevisedWarrant, RevisedArgument, N, RestArgument),
    % put it back
discuss(RevisedArgument, Model, FinalArgument).

discuss1(check, Argument, Model, FinalArgument) :- !,
    write("Which item shall I check? "),
    intread_atom(0-100, N),
    nummember(Warrant, Argument, N),
    cite_cases(attacking Warrant),
    discuss(Argument, Model, FinalArgument).

% ------------ Argue with the user about a warrant

discuss_war(Warrant, MNo, RevisedWarrant) :-
    cite_cases(defending Warrant),
    write("\nDo you still wish to revise this conclusion (y or n)? ") ,
    intread_atom(y, !),
    write("What do you think the revised contribution should be? ", []),
    read([NewWStr], write("Why? ", !),
    read_atom(NewBacking),
    write("\nDo you want to make this change to:\n"),
    write("\n1. The current argument only\n"),
    write("\n2. The current argument and to the general model of opinion used\n"),
    write("\n\tOption (1 or 2) -> "),
    read_atom([1,2], Answer),
    % get the warrant number
    Warrant = WNo--
    RevisedWarrant = WNo--NewWStr--NewBacking,
    ( Answer = 1 -> true
    ; retract( w_strength_and_backing(MNo,WNo--), % change the model
    assert(z w_strength_and_backing(MNo,WNo,NewWStr,NewBacking) ) ).

discuss_war(Warrant, WNo--).

% -------------- Attack/defend a warrant

% About = {attacking,defending}
% lookup warrant no
% A similar warrant has the same WNo

% About = "WNo--"
bagof0(ArgNo, % bagof0/3: all vars are univ quantified
    ( arg_branch(ArgNo,SimilarWarrant),
        check_relation(SimilarWarrant, /*us*/ About, Warrant ) ),
    ArgNos),
member(ArgNo, ArgNos),
arg_summary(ArgNo, Case, Who, _),
arg_branch(ArgNo, SimilarWarrant),
SimilarWarrant = WNo=AltWStr=Backing,
explanation(skeleton_warrant WNo, Text, Arg),
a_case_isa(Thing),
writef("\n\nHey! For \%w \%w, it's also true that ", [Thing,Case]),
writef(Text, Arg),
( curr_username(Who) -> Name = you ; Name = Who ),
text_to_present_str(AltWStr, Text2, Arg2),
write_intro(About, Name),
writef(Text2, Arg2),
may_be_write_old_strength(About, WStr),
writef("\n\nThe reason given was: \%w.\n", [Backing]),
writef("\n\nSee any more \%w cases (y or n)? ", [About]),
yread_atom(n),
).
cite_cases(About, _) :-
    writef("No (more) \%w cases found.\n", [About]).
write_intro(defending, Name) :- writef("\n\nHere, \%w ALSO concluded that ", Name).
write_intro(attacking, Name) :- writef("\n\nHOWEVER, here \%w concluded that ", Name).
may_be_write_old_strength(attacking, WStr) :-
text_to_present_str_alone(WStr, Text3, Arg3),
writef(" (not "), writef(Text3, Arg3), writef(" )").
may_be_write_old_strength(defending, _).

% ----------- Present an argument to the user

present(Argument) :-
    combine(Argument, Qualfn),
    explain_parts(Argument),
    text_to_present_overall_qualfn(Qualfn, Text, Args),
    writef("\n\nThus ", writef(Text, Args), nl).
combine(Argument, Qualfn) :-
gather_branch_strs(Argument, Stm),
combine_strs(Stms, Qualfn).

gather_branch_strs([], []).
gather_branch_strs([ArgBranch|ArgBranches], [Str|Strs]) :-
    ArgBranch = _Str->_
    gather_branch_strs(ArgBranches, Strs).

explain_parts(Argument) :-
    explain_parts(Argument, 1),
explain_parts([], _).
explain_parts([ArgBranch|ArgBranches], N) :-
    ArgBranch = WNo=WStr=Backing,
explanation(skeleton_warrant WNo, Text, Args),
writef("\%w\n", [N]),
writef(Text, Args), nl tab,
text_to_present_str(WStr, Text2, Args2),
writef(Text2, Args2),
writef("\n\nbecause: \%w\n", [Backing]),
NewN is N + 1,
explain_parts(ArgBranches, NewN).
% ---------------- Construct an initial argument

construct(Argument, /*for*/ Claim, Model) :-
  bagof(ArgBranch, construct1(ArgBranch, Claim, Model), Argument).

% Note, SubTree is thrown away for simplicity
construct1(ArgBranch, /*for*/ Claim, Model) :-
  skeleton_warrant WNo :: if Grounds then Claim,
  establish(Grounds, Model, SubTree), % don't store the subtrees for space
  w_strength_and_backing(Model, WNo, WStr, Backing),
  ArgBranch = WNo- WStr- Backing.

establish(G and Gs, M, [Arg|Args]) :- !,
  establish1(G, M, Arg),
  establish1(Gs, M, Args).

establish1(not G, M, not G) :- !,
  \ + establish1(G, M, \).

establish1(any G, M, Arg) :-
  establish1(G, M, Arg), !.

establish1(G, \ from _db(G)) :-
  curr_case(Core),
  wm_get(Core, G),
  \ !.

establish1(A < B, \ A < B) :- !, A < B.

establish1(A >= B, \ A >= B) :- !, A >= B.

establish1(SubClaim, Model, Argument) :-
  logical_warrant WNo :: if Grounds then SubClaim,
  establish(Grounds, Model, Argument).

% ---------------- List stored arguments about the current case

view_arguments :-
  a_case_isa(Thing),
  curr_case(Core),
  write("\nFor the current %v %v: %n\n", [Thing, Case]),
  arg_summary(ArgNo, Case, Who, Qualfn, When),
  ( curr_username(Who) \( -> Who2 = you ; Who2 = Who \),
  text_to_present_overall_qualfn_Qualfn(Text, Args),
  write("%v. %v concluded that \", [ArgNo, Who2],
  write(Text, Args), write(" on %v\n", [When]),
  fail.

view_arguments :-
  write("\nWhich argument to view (0 to end) -> "),
  read_atom(ArgNo),
  ArgNo \(= 0 !, 
  get_argument(ArgNo, Arg),
  write("\nThe details of this argument are as follows:",
  write("\n="*100\n"\n"),
  present(Arg),
  write("\n="*100\n"\n"),
  view_arguments.

view_arguments.

get_argument(ArgNo, Argument) :-
  bagof(ArgBranch, arg_branch(ArgNo, ArgBranch), Argument).

% ---------------- Add an argument to memory

store(Argument) :-
curr_use_name(Who),
curr_case(Case),
count(ArgNo),             % ArgNo = 1,2,...  by repeated backtracking
.onClick_summary(ArgNo, Case),
dateTimeAtom(TimeStamp),
store_branches(Argument, ArgNo, Who, Case),
combine(Argument, Qualfn),
assertz( onClick_summary(ArgNo, Case, Who, Qualfn, TimeStamp) ).

store_branches([], []).
store_branches([ArgBranch|ArgBranches], ArgNo, Who, C) :-
  assert( onClick_branch(ArgNo, ArgBranch) ),
  store_branches(ArgBranches, ArgNo, Who, C).

% ---------------- INTERFACE TO THE WORKING MEMORY

wm_put(C, A) :- assert(wm(C,A)).              % INTERFACE TO THE WORKING MEMORY
wm_get(C, A) :- wm(C,A).
wm_clear(C) :- retractall(wm(C, _)).
E.2.2 A Knowledge Base for Prospect Appraisal

This code illustrates the form of a simple knowledge base for prospect risk appraisal; a complete knowledge base would contain more warrants, attributes and models.

A trace of the shell running with this knowledge base is given in Section E.3.1.

```prolog
:- dynamic w_strength_and_backing/4, model/4.
:- ensure_loaded(library(math)).

% ----------------- Attributes for describing the problem
attribute(trap, [dip,fault,other], "What is the trap of the prospect?").
attribute(reservoir, [cretaceous,kimmeridge,forties], "What is the name of the reservoir?").
attribute('faulting age', integer, "What is the age of the most recent faulting? ").
attribute('source age', [old,young], "Is the source old or young? ").

% ----------------- The set of skeleton warrants
skeleton_warrant 1 :: TrapType ::
    if trap=TrapType
    then oil=present.

skeleton_warrant 2 ::
    if trap=fault
    and 'reservoir age' = ResAge
    and 'faulting age' = FaultAge
    and ResAge < FaultAge
    then oil=present.

logical_warrant 5 :: if reservoir=cretaceous
    then 'reservoir age'=200 / *MYrs*/.

logical_warrant 6 :: if reservoir=kimmeridge
    then 'reservoir age'=250 / *MYrs*/.

logical_warrant 7 :: if reservoir=forties
    then 'reservoir age'=300 / *MYrs*/.

% ----------------- A model of opinion

% Model 1:
model(1, 'Pete', 'W. W. Sea', '3-Dec-89').

% w_strength_and_backing(ModelNo, SkelWarNo, Strength, Backing).
w_strength_and_backing(1, 1-dip, 100, 'dips always seal').
w_strength_and_backing(1, 1-fault, 40, 'faults almost always leak').
w_strength_and_backing(1, 1-other, 70, 'other seals usually work').
w_strength_and_backing(1, 2, 60, 'reservoir may be penetrated by faults').

% ----------------- The main claim to establish
top_level_claim(oil=present).

% ----------------- Rules for combining warrant strengths together
combine_strs(Strs, ICombinedStr) :-
    combine_strs(Strs, 100, CombinedStr),
    ICombinedStr is integer(CombinedStr).

combine_strs([], Str, Str).
combine_strs([Str|Strs], StrSoFar, CombinedStr) :-
```
NewStr is StrSoFar * Str / 100,  
combine_strs(Sts, NewStr, CombinedStr).

% -------------- Definition of consistent/inconsistent warrants

check_relation(Warrant, /*öis*/ About, SimilarWarrant) :-  
  Warrant = WNo−WStr1−,  
  SimilarWarrant = WNo−WStr2−,  
  distance_between_strs(WStr1, WStr2, AbsD),  
  consistency_threshold(Th),  
  ( About = defending -> AbsD =< Th  
  ; About = attacking -> AbsD > Th ).

consistency_threshold(20). % if =< 20% difference, the warrants are approx. consistent

distance_between_strs(Str1, Str2, AbsD) :-  
  D is Str1 − Str2,  
  abs(D, AbsD).

% -------------- Information for pretty text formatting

a_case_isa(prospect).  
text_to_present_str(V, "this contributed a probability of %v", [V]).  
text_to_present_overall_qualify(V, "the overall probability is %v", [V]).  
text_to_present_str_alone(V, "%v", [V]).

explanation(skeleton_warrant 1−D, "the trap of the prospect is %v", [D]).  
explanation(skeleton_warrant 2.  
  "nearby faulting is more recent than the reservoir age", []).
E.2.3 A Knowledge Base for a Software Advisor

This code similarly illustrates the form of a knowledge base for assessing the suitability of the machine learning tool ID3 to a particular data set. Again, a complete knowledge base would contain many more warrants, attributes and warrants.

A trace of the shell running with this knowledge base is given in Section E.3.2.

:- dynamic w_strength_and_backing/4, model/4.

% ---------------- Attributes for describing the problem

attribute('easy encoding', [easy,difficult],
  "How easily can the expert encode examples? ").
attribute('experts agree', [yes,no],
  "Do the experts always agree on the classification? ").
attribute('unique class', [yes,no],
  "Is there always precisely one class assigned to each example? ").

% ---------------- The set of skeleton warrants

skeleton_warrant 1—Ease ::
  if 'easy encoding'='Easy
  then 'id3 appropriate'='yes.
skeleton_warrant 2—Agreement ::
  if 'experts agree'='Agreement
  then 'id3 appropriate'='yes.
skeleton_warrant 3 ::
  if 'unique class'='no
  then 'id3 appropriate'='yes.

% ---------------- A model of opinion

% Model 1:
model(1, 'Pete', 'Use of MLT Algorithms', '3-Nov-90').

% w_strength_and_backing(ModelNo, SkelWarNo, Strength, Backing).
w_strength_and_backing(1, 1—easy, easily, 'plenty of data is available').
w_strength_and_backing(1, 1—difficult, 'with difficulty',
  'data is needed, but difficult to acquire').
w_strength_and_backing(1, 2—eas, easily, '
  the classification rules will be acceptable to all the experts').
w_strength_and_backing(1, 2—no, 'moderately easily',
  'the classification rules may be acceptable to some experts').
w_strength_and_backing(1, 3, 'with difficulty',
  'ID3 can only predict precisely one class for an example').

% ---------------- The main claim to establish

top_level_claim('id3 appropriate'='yes).

% ---------------- Rules for combining warrant strengths together

combine_strs(Strs, 'with difficulty'):-
  memberchk('with difficulty', Strs).
combine_strs(Strs, 'moderately easily'):-
  memberchk('moderately easily', Strs).
combine_strs(Strs, easily).

% ---------------- Definition of consistent/inconsistent warrants

check_relation(Warrant, /*is*/ About, SimilarWarrant) :-
  Warrant = WNo—WStr1—
  SimilarWarrant = WNo—WStr2—
  distance_between_strs(WStr1, WStr2, D),
( About = defending \rightarrow D = 0
; About = attacking \rightarrow D = 1 ).

distance_between_strs(A, A) :- !.
distance_between_strs(_, _).

% ------------ Information for pretty text formatting

a_case_isa(domain).
text_to_present_str(V, "ID3 can be applied \%w", [V]).
text_to_present_overall_qualifn(V, "overall ID3 can be applied \%w", [V]).
text_to_present_str_alone(V, "\%w", [V]).

explanation(skeleton_warrant 1-
-Ease,
   "The experts find the examples \%w to encode", [Ease]).
explanation(skeleton_warrant 2-
-yes,
   "The experts always agree on the classification", []).

explanation(skeleton_warrant 2-
-no,
   "The experts don’t always agree on the classification", []).

explanation(skeleton_warrant 3,
   "Sometimes more than one class is assigned to an example", []).
E.3 Trace of Usage

Here a trace of the shell running with the two previous knowledge bases is given. Marginal notes comment on the interaction.

E.3.1 Prospect Appraisal

<table>
<thead>
<tr>
<th>?- run.</th>
<th>What is your name? steve</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find the name of the</td>
<td>Top level options:</td>
</tr>
<tr>
<td></td>
<td>=============</td>
</tr>
<tr>
<td>What do you want to do:</td>
<td>1. Examine a new prospect</td>
</tr>
<tr>
<td></td>
<td>2. View arguments for the current prospect</td>
</tr>
<tr>
<td></td>
<td>3. Generate a new argument for the current prospect</td>
</tr>
<tr>
<td></td>
<td>4. Quit</td>
</tr>
<tr>
<td>Option -&gt; 1</td>
<td>What is the name of the prospect you want to work on? p27</td>
</tr>
<tr>
<td></td>
<td>Data entry:</td>
</tr>
<tr>
<td></td>
<td>=============</td>
</tr>
<tr>
<td></td>
<td>Having selected a</td>
</tr>
<tr>
<td></td>
<td>What is the type of the prospect?</td>
</tr>
<tr>
<td></td>
<td>work dr. disp user de-</td>
</tr>
<tr>
<td></td>
<td>scribe if fault sys-</td>
</tr>
<tr>
<td></td>
<td>tem. 3. other</td>
</tr>
<tr>
<td></td>
<td>Option -&gt; 2</td>
</tr>
<tr>
<td></td>
<td>1. cretaceous</td>
</tr>
<tr>
<td></td>
<td>2. kimmeridge</td>
</tr>
<tr>
<td></td>
<td>3. forties</td>
</tr>
<tr>
<td></td>
<td>Option -&gt; 2</td>
</tr>
<tr>
<td></td>
<td>Is the source old or young?</td>
</tr>
<tr>
<td></td>
<td>1. old</td>
</tr>
<tr>
<td></td>
<td>2. young</td>
</tr>
<tr>
<td></td>
<td>Option -&gt; 2</td>
</tr>
<tr>
<td></td>
<td>=============</td>
</tr>
<tr>
<td></td>
<td>What do you want to do:</td>
</tr>
<tr>
<td></td>
<td>1. Examine a new prospect</td>
</tr>
<tr>
<td></td>
<td>2. View arguments for the current prospect</td>
</tr>
<tr>
<td></td>
<td>3. Generate a new argument for the current prospect</td>
</tr>
<tr>
<td></td>
<td>4. Quit</td>
</tr>
<tr>
<td></td>
<td>Option -&gt; 3</td>
</tr>
<tr>
<td></td>
<td>Select a model of opinion to use:</td>
</tr>
<tr>
<td></td>
<td>=============</td>
</tr>
<tr>
<td></td>
<td>Available models are as follows:</td>
</tr>
<tr>
<td></td>
<td>1. N. W. Sea (Pete, 3-Dec-89)</td>
</tr>
<tr>
<td></td>
<td>Which model do you want to use? 1.</td>
</tr>
<tr>
<td></td>
<td>The current argument is as follows:</td>
</tr>
<tr>
<td></td>
<td>=============</td>
</tr>
<tr>
<td></td>
<td>1. the trap of the prospect is fault</td>
</tr>
<tr>
<td></td>
<td>this contributed a probability of 40%,</td>
</tr>
<tr>
<td></td>
<td>because: faults almost always leak</td>
</tr>
</tbody>
</table>

(Here, there’s only one model to select from. In a full system there would be more)

The system’s initial argument.
2. nearby faulting is more recent than the reservoir age
   this contributed a probability of 60%,
   because: reservoir may be penetrated by faults

Thus the overall probability is 24%

Do you agree, wish to challenge an item, or wish me to check the

argument? (agree/disagree/check)? disagree

Which item do you disagree with? 1

1. the trap of the prospect is fault
   arguments contributed a probability of 40%,
   faults. The system almost always leak
   now searches for proper cases, assume,
   this item.

Hey! For prospect p12, it's also true that the trap of the
   prospect is fault: Here, John ALSO concluded that this
   contributed a probability of 40%.
   The reason given was: around the mid-Simian region, calcite
   precipitation appears to improve fault sealing quality.

See any more defending cases (y or n)? y

Hey! For prospect p14, it's also true that the trap of the
   prospect is fault. Here, John ALSO concluded that this
   contributed a probability of 40%.
   The reason given was: Statistics from the Garvin region (see
   file s32) suggest fault sealing rate is approximately 2 in 5.

See any more defending cases (y or n)? n

Do you still wish to revise this conclusion (y or n)? y
   the user still disagrees
   considering
   Why? Faults in the Cen. Median appear very poor; 20 consecutive
   faulted wells were dry
   risk of the prospect,
   and thus modifies the

Do you want to make this change to:
   warrant strength and
   1. the current argument only
   2. the current argument and to the general model
   of opinion used
   Option (1 or 2) -> 1

The current argument is as follows:

1. the trap of the prospect is fault
   this contributed a probability of 10%,
   because: Faults in the Cen. Median appear very poor; 20
   consecutive faulted wells were dry
2. nearby faulting is more recent than the reservoir age
   this contributed a probability of 60%,
   because: reservoir may be penetrated by faults

Thus the overall probability is 6%

Do you agree, wish to challenge an item, or wish me to check the
   argument? (agree/disagree/check)? check

Which item shall I check? 1

See any more attacking cases (y or n)? y

Hey! For prospect p12, it's also true that the trap of the
   prospect is fault. HOWEVER, here Pete concluded that this
   contributed a probability of 80% (not 10%)
   according with this conclu-
E.3. TRACE OF USAGE

The reason given was: Faults in Cent. Median region seem to seal better, as evidenced by prospects p81 and p92.

In two previous See any faults attacking considered so worry-
for prospects, it's also true that the trap of the prospect is fault. HOWEVER, here Pete concluded that this contributed a probability of 90% (not 10%). The reason given was: Cent. Median regional faults seem effective from statistics db.

See any more attacking cases (y or n)? n

The current argument is as follows:
==================================
1. the trap of the prospect is fault
   this contributed a probability of 10%,
   because: Faults in the Cen. Median appear very poor; 20 consecutive faulted wells were dry
2. nearby faulting is more recent than the reservoir age
   this contributed a probability of 60%,
   because: reservoir may be penetrated by faults

Thus the overall probability is 6%

Do you agree, wish to challenge an item, or wish me to check the argument? (agree/disagree/check)? disagree

Which item do you disagree with? 1
1. the trap of the prospect is fault
   this contributed a probability of 10%,
   because: Faults in the Cen. Median appear very poor; 20 consecutive faulted wells were dry

No (more) defending cases found.

Do you still wish to revise this conclusion (y or n)? y
What do you think the revised contribution should be? 25.

Why? Faults appear poor (20 consecutive dry wells), but also evidence from p81 and p92 suggests there is a chance the fault *may* just be okay.

Do you want to make this change to:
1. The current argument only
2. The current argument, and to the general model of opinion used

Option (1 or 2) -> 1

The current argument is as follows:
==================================
1. the trap of the prospect is fault
   this contributed a probability of 25%,
   because: Faults appear poor (20 consecutive dry wells),
   but also evidence from p81 and p92 suggests there is a chance the fault *may* just be okay.
2. nearby faulting is more recent than the reservoir age
   this contributed a probability of 60%,
   because: reservoir may be penetrated by faults

Thus the overall probability is 15%

Do you agree, wish to challenge an item, or wish me to check the argument? (agree/disagree/check)? agree

Finally the user is satisfied. The argument is stored.
What do you want to do:
1. Examine a new prospect
2. View arguments for the current prospect
3. Generate a new argument for the current prospect
4. Quit
   Option -> 1

What is the name of the prospect you want to work on? p12
wishes to see details of another argument,
Data entry:
->
(I already know about prospect p12)
Top level options:
What do you want to do:
1. Examine a new prospect
2. View arguments for the current prospect
3. Generate a new argument for the current prospect
4. Quit
   Option -> 2

For the current prospect p12:
1. john concluded the overall probability is 24% on 24/3/91
5. pete concluded the overall probability is 48% on 18/5/91
   Which argument to view (0 to end) -> 5

The details of this argument are as follows:
->
1. the trap of the prospect is fault appraisal arguments,
   this contributed a probability of 80%,
   because: Faults in Cent. Median region seem to seal better, as evidenced by prospects p81 and p92
2. nearby faulting is more recent than the reservoir age
   this contributed a probability of 60%,
   because: reservoir may be penetrated by faults

Thus the overall probability is 48%

For the current prospect p12:
1. john concluded that overall probability is 24% on 24/3/91
5. pete concluded that overall probability is 48% on 18/5/91
   Which argument to view (0 to end) -> 0

Top level options:
What do you want to do:
1. Examine a new prospect
2. View arguments for the current prospect
3. Generate a new argument for the current prospect
4. Quit
   Option -> 4

Finish.

yes
E.3.2 Advice about Using ID3

| ?- run. |
| What is your name? john |

Top level options:
===================
What do you want to do:
1. Examine a new domain
2. View arguments for the current domain
3. Generate a new argument for the current domain
4. Quit

Option -> 1

What is the name of the domain you want to work on? Anemia diagnosis

Data entry:
============
How easily can the expert encode examples?
1. easy
2. difficult

Option -> 1

Do the experts always agree on the classification?
1. yes
2. no

Option -> 1

Is there always precisely one class assigned to each example?
1. yes
2. no

Option -> 2

Top level options:
===================
What do you want to do:
1. Examine a new domain
2. View arguments for the current domain
3. Generate a new argument for the current domain
4. Quit

Option -> 3

Select a model of opinion to use:
==============================
Available models are as follows:
1. Use of MLT Algorithms (Pete, 3-Nov-90)

Which model do you want to use? 1.

The current argument is as follows:
==================================
1. The experts find the examples easy to encode
   ID3 can be applied easily,
   because: plenty of data is available
2. The experts always agree on the classification
   ID3 can be applied easily,
   because: the classification rules will be acceptable to all
   the experts
3. Sometimes more than one class is assigned to an example
   ID3 can be applied with difficulty,
   because: ID3 can only predict precisely one class for
   an example

Thus overall ID3 can be applied with difficulty

Do you agree, wish to challenge an item, or wish me to check the
argument? (agree/disagree/check)? check

Which item shall I check? 3

Hey! For domain hardware fault diagnosis, it’s also true that

Sometimes more than one class is assigned to an example.

HOWEVER, here Pete concluded that ID3 can be applied easily
(not with difficulty)
The reason given was: Instead of predicting non-exclusive
classes, the following you can transform the data to generate
different training sets, one predicting C1 or NotC1, one
predicting C2 or Not C2, etc. This overcomes the
non-exclusiveness problem.

See any more attacking cases (y or n)? n

The current argument is as follows:
===============================================================================
1. The experts find the examples easy to encode
   ID3 can be applied easily,
   because: plenty of data is available
2. The experts always agree on the classification
   ID3 can be applied easily,
   because: the classification rules will be acceptable to all
   the experts
3. Sometimes more than one class is assigned to an example
   ID3 can be applied with difficulty,
   because: ID3 can only predict precisely one class for
   an example

Thus overall ID3 can be applied with difficulty

Do you agree, wish to challenge an item, or wish me to check the
argument? (agree/disagree/check)? disagree

Which item do you disagree with? 3

3. Sometimes more than one class is assigned to an example
   ID3 can be applied with difficulty,
   because: ID3 can only predict precisely one class for
   an example

No (more) defending cases found.

Do you still wish to revise this conclusion (y or n)? y

What do you propose the revised contribution should be? easily.

Why is the transformation in the hardware fault diagnosis case
also be applied here?

Do you want to make this change to:
1. The current argument only
2. The current argument, and to the general model of opinion used
   Option (1 or 2) -> 1

The current argument is as follows:
===============================================================================
1. The experts find the examples easy to encode
   ID3 can be applied easily,
   because: plenty of data is available
2. The experts always agree on the classification
   ID3 can be applied easily,
   because: the classification rules will be acceptable to all
   the experts
3. Sometimes more than one class is assigned to an example
   ID3 can be applied easily,
   because: the transformation in the hardware fault diagnosis can also be applied here

Now the user asks the system to look for refutations of (part of) this argument, cases where ID3 was successfully applied.
Thus overall ID3 can be applied easily

Do you agree, wish to challenge an item, or wish me to check the argument? (agree/disagree/check)? agree

Top level options:

What do you want to do:

1. Examine a new domain
2. View arguments for the current domain
3. Generate a new argument for the current domain
4. Quit

Option -> 4

yes

The final argument is stored.
Appendix F

Details of the ML Advisor Application

This Appendix lists some of the domain-specific information required for the ML Advisor application of our argumentation model. The application is described in Section 6.5, including a hypothetical dialogue of the interaction which might occur.

Here we present the problem description language (Section F.1), the full set of skeleton warrants (Section F.2) and a corresponding model of opinion for assessing ID/CN suitability (Section F.3). Finally, several ‘advice cases’ and an outline of the corresponding arguments for ID/CN’s suitability to them are given (Section F.4).

F.1 Problem Description Language

The following table summarises the problem description language, used for summarising the case (a data set) under question. All attributes have boolean values apart from the first two (text strings) and the last 5 (numeric and text).
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Question to ask</th>
</tr>
</thead>
<tbody>
<tr>
<td>'domain name'</td>
<td>What is the name of the domain?</td>
</tr>
<tr>
<td>'domain description'</td>
<td>Please give a brief description of this domain</td>
</tr>
<tr>
<td>'manually provide exs'</td>
<td>Can an expert easily provide examples manually?</td>
</tr>
<tr>
<td>'manually classify exs'</td>
<td>Can an expert manually classify the examples, using no other data?</td>
</tr>
<tr>
<td>'experts agree on class'</td>
<td>Do experts always agree on the classification?</td>
</tr>
<tr>
<td>'complex relns between atts'</td>
<td>Do decisions require knowledge of relations between attributes eg. mathematical rels as opposed to depending on simple logical combinations of attributes?</td>
</tr>
<tr>
<td>'environment/context import.'</td>
<td>Is information external to the example relevant (eg. its environment, its context)?</td>
</tr>
<tr>
<td>'examples available in db'</td>
<td>Is your data from a database?</td>
</tr>
<tr>
<td>'missing attribute values'</td>
<td>Are there missing attribute values?</td>
</tr>
<tr>
<td>'exactly one class per ex'</td>
<td>Is there exactly one class per example?</td>
</tr>
<tr>
<td>'some atts hierarchical'</td>
<td>Can attributes be hierarchically organised?</td>
</tr>
<tr>
<td>'classes are hierarchical'</td>
<td>Can classes be hierarchically organised?</td>
</tr>
<tr>
<td>'some att vals are numeric'</td>
<td>Any numeric attribute values?</td>
</tr>
<tr>
<td>'some att vals are ordered'</td>
<td>Any non-numeric, ordered attribute values?</td>
</tr>
<tr>
<td>'some classes have no exs'</td>
<td>Any classes with zero examples?</td>
</tr>
<tr>
<td>'ordering of exs important'</td>
<td>Ordering of examples important?</td>
</tr>
<tr>
<td>'att vals fully specified'</td>
<td>Can all the attribute values be precisely specified?</td>
</tr>
<tr>
<td>'symmetries exist'</td>
<td>Any obvious symmetry in the examples?</td>
</tr>
<tr>
<td>'any uncertainty in class'</td>
<td>Is expert ever uncertain about the classification?</td>
</tr>
<tr>
<td>'class values are discrete'</td>
<td>Are class values discrete (as opposed to continuous)?</td>
</tr>
<tr>
<td>'number of atts'</td>
<td>Approximately how many attributes are there?</td>
</tr>
<tr>
<td>'number of exs'</td>
<td>Approximately how many examples are there?</td>
</tr>
<tr>
<td>'number of classes'</td>
<td>Approximately how many classes are there?</td>
</tr>
<tr>
<td>'hardware for ID'</td>
<td>Are you running on a Sun3 or a Sun4?</td>
</tr>
<tr>
<td>'max acceptable runtime'</td>
<td>What is the maximum acceptable run-time?</td>
</tr>
</tbody>
</table>

## F.2 The Skeleton Warrant Set

The set of skeleton warrants for assessing the suitability of ID/CN to a particular data set are as follows:

**skel-warrant 1 :: "No data"
if 'manually provide exs' = n
and 'examples available in db' = n
then ID/CN is inappropriate.

**skel-warrant 2 :: "Incomplete data"
if 'manually classify exs' = n
and 'examples available in db' = n
then ID/CN is inappropriate.

**skel-warrant 3 :: "undefined answer"
if 'manually classify exs' = y
and 'experts agree on class' = n
and 'examples available in db' = n
then ID/CN is inappropriate.

**skel-warrant 4 :: "ID/CN rule language too poor"
if 'complex relns between atts' = y
then ID/CN is inappropriate.

skel-warrant 5 :: "missing context data"
if 'environment/context import.' = y
then ID/CN is inappropriate.

skel-warrant 6 :: "missing att values"
if 'missing attribute values' = y
then ID/CN is inappropriate.

skel-warrant 7 :: "No unique target class"
if 'exactly one class per ex' = n
then ID/CN is inappropriate.

skel-warrant 8 :: "att desc. could be improved"
if 'some atts hierarchical' = y
then ID/CN is inappropriate.

skel-warrant 9 :: "class desc. could be improved"
if 'classes are hierarchical' = y
then ID/CN is inappropriate.

skel-warrant 10 :: "numeric desc. could be improved"
if 'some att vals are numeric' = y
then ID/CN is inappropriate.

skel-warrant 11 :: "ordering could be expressed"
if 'some att vals are ordered' = y
then ID/CN is inappropriate.

skel-warrant 12 :: "lack of data for some classes"
if 'some classes have no exs' = y
then ID/CN is inappropriate.

skel-warrant 13 :: "missing ordering information"
if 'ordering of exs important' = y
then ID/CN is inappropriate.

skel-warrant 14 :: "vals could be better organised"
if 'att vals fully specified' = y
then ID/CN is inappropriate.

skel-warrant 15 :: "missing symmetry information"
if 'symmetries exist' = y
then ID/CN is inappropriate.

skel-warrant 16 :: "uncertainty in classes"
if 'any uncertainty in class' = y
then ID/CN is inappropriate.

skel-warrant 17 :: "non-discrete prediction task"
if 'class values are discrete' = n
then ID/CN is inappropriate.

skel-warrant 18 :: "too few exs or too many atts"
if 'number of atts' < 'number of exs' / 2
then ID/CN is inappropriate.
F.3. A Model of Opinion

A model of opinion for assessing ID/CN’s suitability comprises the skeleton warrants above plus an assignment of strengths to those warrants, listed below. This model is based upon my own opinion of ID/CN’s likely success following my and other people’s experience of its application in different domains.

Note that the calculus for warrant strengths is simple and approximate, reflecting the limited way in which ‘suitability’ can be quantified. Given several warrants apply, strengths would be combined by simply taking the maximum (strongest) strength applying. It should also be noted that such approximations are not expected to be problematic; the roles of the model and arguments are to record and organise knowledge, and to answer users’ queries at the level of precision which they require.
<table>
<thead>
<tr>
<th>Skeleton Warrant</th>
<th>Strength in Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (&quot;No data&quot;)</td>
<td>strong</td>
</tr>
<tr>
<td>2 (&quot;Incomplete data&quot;)</td>
<td>strong</td>
</tr>
<tr>
<td>3 (&quot;undefined answer&quot;)</td>
<td>weak</td>
</tr>
<tr>
<td>4 (&quot;id3 rule language too poor&quot;)</td>
<td>medium</td>
</tr>
<tr>
<td>5 (&quot;missing context data&quot;)</td>
<td>medium</td>
</tr>
<tr>
<td>6 (&quot;missing att values&quot;)</td>
<td>weak</td>
</tr>
<tr>
<td>7 (&quot;No unique target class&quot;)</td>
<td>weak</td>
</tr>
<tr>
<td>8 (&quot;att desc. could be improved&quot;)</td>
<td>weak</td>
</tr>
<tr>
<td>9 (&quot;class desc. could be improved&quot;)</td>
<td>weak</td>
</tr>
<tr>
<td>10 (&quot;numeric desc. could be improved&quot;)</td>
<td>weak</td>
</tr>
<tr>
<td>11 (&quot;ordering could be expressed&quot;)</td>
<td>weak</td>
</tr>
<tr>
<td>12 (&quot;lack of data for some classes&quot;)</td>
<td>weak</td>
</tr>
<tr>
<td>13 (&quot;missing ordering information&quot;)</td>
<td>strong</td>
</tr>
<tr>
<td>14 (&quot;vals could be better organised&quot;)</td>
<td>weak</td>
</tr>
<tr>
<td>15 (&quot;missing symmetry information&quot;)</td>
<td>weak</td>
</tr>
<tr>
<td>16 (&quot;uncertainty in classes&quot;)</td>
<td>medium</td>
</tr>
<tr>
<td>17 (&quot;non-discrete prediction task&quot;)</td>
<td>strong</td>
</tr>
<tr>
<td>18 (&quot;too few exs or too many atts&quot;)</td>
<td>medium</td>
</tr>
<tr>
<td>19-10 (&quot;too many classes&quot;)</td>
<td>weak</td>
</tr>
<tr>
<td>19-20 (&quot;too many classes&quot;)</td>
<td>weak</td>
</tr>
<tr>
<td>19-50 (&quot;too many classes&quot;)</td>
<td>strong</td>
</tr>
<tr>
<td>20 (&quot;too many classes or too few exs&quot;)</td>
<td>medium</td>
</tr>
<tr>
<td>21 (&quot;expected runtime too long&quot;)</td>
<td>medium</td>
</tr>
</tbody>
</table>

An example argument constructed with this model is illustrated in Figure 6.2.

### F.4 Advice Cases

Initially, several cases and their corresponding arguments about ID/CN’s suitability would be provided to the system during its construction. These cases are those in which some of the problems, identified by the model of opinion, have been overcome by the application of some transformation of the data set. Their arguments describe the revised opinion about ID/CN’s suitability, along with backings encoding advice about how the problems can be alleviated or removed.

Seven such ‘advice cases’ are summarised below, taken from real-world applications where problems with the data format were both encountered and solved:

**Soya** Diagnosis of soybean diseases from descriptions of the plant [MC80]. This database is contained in the UCI Repository of Machine Learning Databases.

**HWFault** Fault diagnosis of computer hardware, given hexadecimal sensor readings on the machine [Cla90a].

**Testes** Medical treatment recommendation for patients with maldescended testicles [Cla90b].

**Protein** Prediction of a protein’s secondary structure given its primary structure sequence [Kin87].

**Satellite** Fault diagnosis of the electrical sub-system of a satellite [Pea88].

**Mushroom** Identification of poisonous mushrooms from a description of their observable characteristics. This database is contained in the UCI Repository of Machine Learning Databases.
**Dynamic Control** Prediction of a human's control actions used to balance an inverted pendulum, given examples of the pendulum's state and the human's action with the pendulum in that state.

Descriptions of the form of the data sets, as originally provided, are given below.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Soya</th>
<th>HFault</th>
<th>Testes</th>
<th>Prot.</th>
<th>Sat.</th>
<th>Mush.</th>
<th>Ctrl</th>
</tr>
</thead>
<tbody>
<tr>
<td>manually provide exs</td>
<td>y</td>
<td>n</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>y</td>
<td>y</td>
</tr>
<tr>
<td>manually classify exs</td>
<td>y</td>
<td>n</td>
<td>y</td>
<td>n</td>
<td>y</td>
<td>y</td>
<td>y</td>
</tr>
<tr>
<td>experts agree on class</td>
<td>y</td>
<td>n</td>
<td>y</td>
<td>n</td>
<td>y</td>
<td>y</td>
<td>n</td>
</tr>
<tr>
<td>complex rels between atts</td>
<td>n</td>
<td>y</td>
<td>n</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>environment/context import.</td>
<td>n</td>
<td>y</td>
<td>n</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>examples available in db</td>
<td>y</td>
<td>n</td>
<td>y</td>
<td>n</td>
<td>y</td>
<td>y</td>
<td>n</td>
</tr>
<tr>
<td>missing attribute values</td>
<td>n</td>
<td>y</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>exactly one class per ex</td>
<td>y</td>
<td>n</td>
<td>y</td>
<td>n</td>
<td>y</td>
<td>y</td>
<td>n</td>
</tr>
<tr>
<td>some atts hierarchical</td>
<td>n</td>
<td>y</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>classes are hierarchical</td>
<td>n</td>
<td>y</td>
<td>n</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>some att vals are numeric</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>some att vals are ordered</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>some classes have no exs</td>
<td>n</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>ordering of exs important</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>att vals fully specified</td>
<td>y</td>
<td>n</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>n</td>
</tr>
<tr>
<td>symmetries exist</td>
<td>n</td>
<td>n</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>any uncertainty in class</td>
<td>n</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>class values are discrete</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>number of atts</td>
<td>35</td>
<td>13</td>
<td>11</td>
<td>3</td>
<td>75</td>
<td>23</td>
<td>4</td>
</tr>
<tr>
<td>number of exs</td>
<td>307</td>
<td>140</td>
<td>185</td>
<td>200000</td>
<td>4000</td>
<td>8120</td>
<td>1040</td>
</tr>
<tr>
<td>number of classes</td>
<td>19</td>
<td>30</td>
<td>9</td>
<td>3</td>
<td>112</td>
<td>2</td>
<td>200</td>
</tr>
<tr>
<td>hardware for ID</td>
<td>Sun4</td>
<td>Sun3</td>
<td>Sun3</td>
<td>Sun4</td>
<td>Sun4</td>
<td>Sun4</td>
<td>Sun3</td>
</tr>
<tr>
<td>max accept. runtime (sec)</td>
<td>3600</td>
<td>3600</td>
<td>3600</td>
<td>3600</td>
<td>3600</td>
<td>3600</td>
<td>3600</td>
</tr>
</tbody>
</table>

For each data set, an argument about ID/CN's applicability can be constructed. The warrants which apply for each data set are summarised below. For each case, the applicable warrants would combine into an argument similar to that shown in Figure 6.2.
<table>
<thead>
<tr>
<th>Skeleton Warrant</th>
<th>Used in argument?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (&quot;No data&quot;)</td>
<td>✓</td>
</tr>
<tr>
<td>2 (&quot;Incomplete data&quot;)</td>
<td>✓</td>
</tr>
<tr>
<td>3 (&quot;undefined answer&quot;)</td>
<td>✓</td>
</tr>
<tr>
<td>4 (&quot;ID/CN rule language too poor&quot;)</td>
<td>✓ ✓ ✓</td>
</tr>
<tr>
<td>5 (&quot;missing context data&quot;)</td>
<td>✓ ✓ ✓</td>
</tr>
<tr>
<td>6 (&quot;missing att values&quot;)</td>
<td>✓</td>
</tr>
<tr>
<td>7 (&quot;No unique target class&quot;)</td>
<td>✓</td>
</tr>
<tr>
<td>8 (&quot;att desc. could be improved&quot;)</td>
<td>✓ ✓ ✓</td>
</tr>
<tr>
<td>9 (&quot;class desc. could be improved&quot;)</td>
<td>✓</td>
</tr>
<tr>
<td>10 (&quot;numeric desc. could be improved&quot;)</td>
<td>✓</td>
</tr>
<tr>
<td>11 (&quot;ordering could be expressed&quot;)</td>
<td>✓</td>
</tr>
<tr>
<td>12 (&quot;lack of data for some classes&quot;)</td>
<td>✓</td>
</tr>
<tr>
<td>13 (&quot;missing ordering information&quot;)</td>
<td>✓</td>
</tr>
<tr>
<td>14 (&quot;vals could be better organised&quot;)</td>
<td>✓</td>
</tr>
<tr>
<td>15 (&quot;missing symmetry information&quot;)</td>
<td>✓ ✓ ✓</td>
</tr>
<tr>
<td>16 (&quot;uncertainty in classes&quot;)</td>
<td>✓</td>
</tr>
<tr>
<td>17 (&quot;non-discrete prediction task&quot;)</td>
<td>✓</td>
</tr>
<tr>
<td>18 (&quot;too few exs or too many atts&quot;)</td>
<td>✓</td>
</tr>
<tr>
<td>19-10 (&quot;too many classes&quot;)</td>
<td>✓ ✓ ✓</td>
</tr>
<tr>
<td>19-20 (&quot;too many classes&quot;)</td>
<td>✓</td>
</tr>
<tr>
<td>19-50 (&quot;too many classes&quot;)</td>
<td>✓ ✓ ✓</td>
</tr>
<tr>
<td>20 (&quot;too many classes or too few exs&quot;)</td>
<td>✓</td>
</tr>
<tr>
<td>21 (&quot;expected runtime too long&quot;)</td>
<td>✓</td>
</tr>
</tbody>
</table>

Some warrants state that the data set property described in the warrant’s grounds makes ID/CN inapplicable to some degree. However, in practice ID/CN was made applicable by transforming the data set. This would be reflected by the strength of those warrants being weakened, and a description of how the data set was transformed to overcome the problem given. These descriptions are stored as the warrant’s backings, and constitute the advice to future users.

An example backing is shown below, for warrant 7 ("no unique target class") in the argument concerning ID/CN’s applicability to the hardware fault diagnosis data set. Originally, the warrant weakly implied ID/CN would be inappropriate, but because the data set could be transformed this warrant’s strength was changed in the argument to have no effect on ID/CN’s suitability.
In this dataset, there was often more than one class per example. To handle this here, several new data sets were derived performing a binary prediction task of ‘C’ or ‘not C’, for each class C. The original data set appeared:

<table>
<thead>
<tr>
<th>Dial1:</th>
<th>Dial2:</th>
<th>Dial3:</th>
<th>FailureClass:</th>
</tr>
</thead>
<tbody>
<tr>
<td>e1:</td>
<td>00</td>
<td>EF12</td>
<td>y</td>
</tr>
<tr>
<td>e2:</td>
<td>0B</td>
<td>EF11</td>
<td>n</td>
</tr>
<tr>
<td>e3:</td>
<td>00</td>
<td>EF11</td>
<td>n</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Processor1 AND Memory</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Processor1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Memory</td>
</tr>
</tbody>
</table>

The two transformed datasets were (for Processor1 and Memory respectively):

<table>
<thead>
<tr>
<th>Dial1:</th>
<th>Dial2:</th>
<th>Dial3:</th>
<th>Processor1Failure?</th>
</tr>
</thead>
<tbody>
<tr>
<td>e1:</td>
<td>00</td>
<td>EF12</td>
<td>yes</td>
</tr>
<tr>
<td>e2:</td>
<td>0B</td>
<td>EF11</td>
<td>yes</td>
</tr>
<tr>
<td>e3:</td>
<td>00</td>
<td>EF11</td>
<td>no</td>
</tr>
</tbody>
</table>

and

<table>
<thead>
<tr>
<th>Dial1:</th>
<th>Dial2:</th>
<th>Dial3:</th>
<th>MemoryFailure?</th>
</tr>
</thead>
<tbody>
<tr>
<td>e1:</td>
<td>00</td>
<td>EF12</td>
<td>no</td>
</tr>
<tr>
<td>e2:</td>
<td>0B</td>
<td>EF11</td>
<td>yes</td>
</tr>
<tr>
<td>e3:</td>
<td>00</td>
<td>EF11</td>
<td>yes</td>
</tr>
</tbody>
</table>

Peter Clark
2.5.90

Space does not permit a full list of all backings for all the warrants whose strengths would be changed for these initial advice cases. The general principle, though, of how advice cases and arguments are thus constructed for the initial advice system should now be clear to the reader.