1989

RESCUE: expert system for mine rescue operations based on gas analysis

Samuel Osei-Tutu
University of Wollongong

RECOMMENDED CITATION
NOTE

This online version of the thesis may have different page formatting and pagination from the paper copy held in the University of Wollongong Library.

UNIVERSITY OF WOLLONGONG

COPYRIGHT WARNING

You may print or download ONE copy of this document for the purpose of your own research or study. The University does not authorise you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site. You are reminded of the following:

Copyright owners are entitled to take legal action against persons who infringe their copyright. A reproduction of material that is protected by copyright may be a copyright infringement. A court may impose penalties and award damages in relation to offences and infringements relating to copyright material. Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.
RESCUE: EXPERT SYSTEM FOR MINE RESCUE OPERATIONS BASED ON GAS ANALYSIS

A thesis submitted in partial fulfilment of the requirements for the award of the degree of

MASTER OF ENGINEERING (HONOURS)

from

THE UNIVERSITY OF WOLLONGONG

by

SAMUEL OSEI-TUTU

DEPARTMENT OF CIVIL AND MINING ENGINEERING

(1989)
ACKNOWLEDGEMENTS

I wish to thank Dr. E.Y. Baafi for his able guidance and sound advice in the development of the research described in this dissertation and for his assistance in directing my program of study. I also wish to thank Mr Paul MacKenzie-Wood of the Southern Mine Rescue Station, Bulli, New South Wales and all the staff for their assistance with the development of the knowledge-based system component of the research.

I would like to thank The Australian Development Assistance Bureau (ADAB), and the Department of Civil and Mining Engineering, The University of Wollongong for the opportunity of studying at the department as a fellow of the Special Commonwealth African Assistance Plan (SCAAP).

I thank the following personalities for taking care of my children while I was away from Ghana: Mr J.A. Assumang, Mr F. Kuman and Mr E.G. Aboagye.

Finally, I would like to thank my wife, Comfort and my son Albert for encouraging me to buy a personal computer.
# TABLE OF CONTENTS

| LIST OF FIGURES ........................................ | (a) |
| LIST OF TABLES .......................................... | (b) |
| ABSTRACT .................................................. | (c-d) |

## CHAPTER 1 - GENERAL INTRODUCTION .................................. 1

1.1 Historical Accounts .............................................. 1
1.2 Research Into Fire Emergency ................................... 6
1.3 Definition of Problem ........................................... 8
1.4 Scope of Work ................................................... 9

## CHAPTER 2 - MINE FIRES AND RESCUE OPERATIONS ..................... 11

2.1 The Nature of Fire .................................................. 11
2.2 Flammability and Explosibility Ranges .......................... 14
2.3 Fire Extinguishing Methods ...................................... 15
   2.3.1 Fuel Removal (Starving) .................................. 16
   2.3.2 Oxygen Removal (Smothering) .............................. 16
   2.3.3 Heat Reduction (Cooling) ................................ 17
2.4 The Fire Ratios ................................................... 18
   2.4.1 The Carbon Monoxide Index ................................ 22
   2.4.2 Airfree Ratios ............................................. 22
   2.4.3 The Trickett's Ratio ...................................... 24
   2.4.4 Graham's Ratio ............................................ 25
2.5 Rescue Organisational Procedure ................................. 26

## CHAPTER 3 - ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEMS ........ 30

3.1 The Knowledge-Base ............................................... 31
3.2 The Inference Engine ............................................. 33
   3.2.1 Backward Chaining Inference Engine .................... 33
   3.2.1 Forward Chaining Inference Engine ...................... 34
3.3 The Working Memory ............................................... 34
3.4 Natural Language Processor ...................................... 35
3.5 Knowledge Acquisition .......................................... 35
   3.5.1 The Method of Familiar Tasks ............................ 37
   3.5.2 The Unstructured Interview ............................... 37
   3.5.3 The Structured Interview ................................ 39
   3.5.4 Limited Information Tasks ............................... 40
   3.5.5 Constrained Processing Tasks ............................ 40
      3.5.5.1 The Method of Simulated Familiar Tasks ........ 41
      3.5.5.2 The Method of Scenarios ............................ 42
   3.5.6 Combined Constrained .................................... 42
   3.5.7 The Method of Tough Cases ............................... 43
### 3.6 Comparison of the Methods

3.6.1 Task and Material Simplicity 44
3.6.2 Task Brevity 45
3.6.3 Task Flexibility 45
3.6.4 Task Artificiality 45
3.6.5 Data Format 46
3.6.6 Data Validity 46
   - 3.6.6.1 Validity: The Importance of the Data 47
   - 3.6.6.2 Validity: Data Reliability or Consensus 47
   - 3.6.6.3 Validity: The Completeness of the Database 48

### 3.7 Tools and Languages for Expert Systems 48

3.7.1 The LISP Programming Language 50
3.7.2 The PROLOG Programming Language 53
3.7.3 The Expert System Shell 56
   - 3.7.3.1 Special Features of VP-Expert 56
   - 3.7.3.2 VP-Expert Inference Engine 57
   - 3.7.3.3 VP-Expert Knowledge-base 59
      - 3.7.3.3.1 The ACTIONS Block 59
      - 3.7.3.3.2 The VP-Expert Rules 59

### 3.8 Expert System Application in the Mining Industry 60

### CHAPTER 4 - STRUCTURE AND DESIGN OF RESCUE SYSTEMS 64

4.1 Model Formulation 64
   - 4.1.1 The FORTRAN Program 65
   - 4.1.2 The Lower Explosive Limits 68

4.2 Extinctive Volumes of Nitrogen and Carbon Dioxide 70

4.3 The FORTRAN Program Algorithm 72
   - 4.3.3 Computational Sequence 73

4.4 Development of Database Management Systems 80
   - 4.4.1 Addgas.prg 81
   - 4.4.2 Kunted.prg 85
   - 4.4.3 Forgraph.prg 85
   - 4.4.4 Fortimpt.prg 88
   - 4.4.5 Choselis.prg 88
   - 4.4.6 LIST1...4.prg 89
   - 4.4.7 Zapall.prg 89

4.5 The "BASIC" Programs for Graphics Display 90
   - 4.5.1 Functions of the "BASIC" Programs 92
   - 4.5.2 Design Philosophy 92
      - 4.5.2.1 Converting Time Interval to Hours 94

### CHAPTER 5 - SYSTEM VALIDATION 97

5.1 Operational and Computing Requirement for RESCUE 97
   - 5.1.2 Using RESCUE 98

5.2 Consultation 100

5.3 Confidence Factors 104
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>The Tetrahedron of Fire</td>
<td>12</td>
</tr>
<tr>
<td>2.2</td>
<td>Sample Plot of CO vrs. Time</td>
<td>23</td>
</tr>
<tr>
<td>2.3</td>
<td>Sample Plot of Airfree vrs. Time For CO</td>
<td>23</td>
</tr>
<tr>
<td>2.4</td>
<td>A Flowchart of Rescue Operations</td>
<td>27</td>
</tr>
<tr>
<td>3.1</td>
<td>Structure of Expert System</td>
<td>32</td>
</tr>
<tr>
<td>3.2</td>
<td>Artificial Intelligence Tools</td>
<td>49</td>
</tr>
<tr>
<td>4.1</td>
<td>Structure of &quot;RESCUE&quot; System</td>
<td>66</td>
</tr>
<tr>
<td>4.2</td>
<td>The Coward Diagram</td>
<td>75</td>
</tr>
<tr>
<td>4.3</td>
<td>Connections Between Files</td>
<td>85</td>
</tr>
<tr>
<td>4.4</td>
<td>Hypothetical Arrangement of Sample Points in Panels</td>
<td>87</td>
</tr>
</tbody>
</table>
### List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>List of Recorded Underground Fires in Queensland (1920-1979)</td>
</tr>
<tr>
<td>1.2</td>
<td>List of Recorded Underground Fires in New South Wales (1920-1986)</td>
</tr>
<tr>
<td>3.1</td>
<td>Knowledge Acquisition Methods</td>
</tr>
<tr>
<td>3.2</td>
<td>Some Salient Advantages and Disadvantages of Knowledge Acquisition Methods</td>
</tr>
<tr>
<td>4.1</td>
<td>Explosive Limits of Gases</td>
</tr>
<tr>
<td>4.2</td>
<td>Extinctive Volumes of Gases</td>
</tr>
<tr>
<td>4.3(a)</td>
<td>Structure of GASAMPLE.DBF Database File</td>
</tr>
<tr>
<td>4.3(b)</td>
<td>Structure of KONTER.DBF Database File</td>
</tr>
<tr>
<td>5.1</td>
<td>Sample Output</td>
</tr>
</tbody>
</table>
This thesis addresses a practical problem associated with the coal mining industry: fire outbreak due to dust explosions, short-firing, coal outbursts and other causes have plagued the industry for a long time.

During such emergencies lives and equipment are at risk if proper remedial procedures are not adhered to. The adherence to these life-saving procedures calls for the availability of experts as well as the proper input. Such input consists of both human and material resources. One of the material input, may be in the form of a computer program that may assess the changing conditions of the vitiated atmosphere in order to determine the atmosphere's fitness to accommodate rescuers.

Hitherto, computer programs have been written to compute the explosibility status of the atmosphere surrounding a fire, the results of which are used to plan rescue strategies.

These programs do work satisfactorily. As the life of a fire is prolonged, and more and more samples are analysed, these programs reach their limit due to the absence or inadequacy of data handling facilities.
This problem has been addressed in this thesis by incorporating a database management system with these programs. With this enhancement the amount of data that can be handled by the package is practically unlimited. Thus this system will not be overburdened with data no matter how long a fire fighting process may take.

In addition, an expert system has been attached to assist personnel with making decisions.

The computer programs for this thesis were developed in three different languages:

(1) FORTRAN
(2) BASIC
(3) dBASE III+

The expert system was developed in the VP-Expert Shell. Each of these languages has some features the others lack. For example dBASE III+ is a database management development tool. On the other hand all the graph-plotting programs were written in the BASIC language since BASIC is particularly suited for graphics on the personal computer. The rest of the programs were written in FORTRAN.
CHAPTER ONE

1.0 GENERAL INTRODUCTION

1.1 Historical Accounts

It is clear throughout its history that coal mining has been regarded as a hazardous and dangerous industry. It is only in the last fifty years, however, that greater attention has been focused on the safety and health problems associated with coal mining.

The instance of coal mine disasters in Australia is large and the causes and circumstances relating to these disasters are varied. The severity and magnitude of these disasters are, however, considered small compared to those that have occurred in the United Kingdom, the United States of America and other coal-producing countries of the world.

The beginning of the 20th century saw a greater upsurge in the coal industry throughout the world. To meet the rapidly rising demand for coal, new mines were opened up, old workings in existing mines were extended and the industry was seen to have come of age. The increase in the number of mines coupled with the lack of proper legislation governing the working practices of the industry combined to increase the number of accidents. Between 1960 and 1970, over 3200 men were killed in underground coal mine explosions, and between 1972 and
1979 four major methane and coal dust explosions occurred in New South Wales (NSW) and Queensland with two coal mines totally lost and large loss of life in three of the mines.

The early mines in Australia were developed from the same concepts and design principles as those in Britain. The work-force in those early coal mines was mainly unskilled labour. This fact is evident in the history of the early disasters. These temporary difficulties were soon overcome. Each state in Australia established formal state regulation to oversee that standards of safety and health in mines were maintained at an adequate level. Along with legislation came the gradual introduction of on-the-job training and education for workmen and officials.

In Australia a number of coal mining disasters have occurred with most disasters falling into two aspects. These are:

(1) Fatalities caused as a result of underground fire, gas or coal dust explosions and,

(2) Fatalities caused as a result of gas and coal outburst.

There is an apparent disparity between the deaths caused by these disasters in Queensland (more deaths) and New South Wales (fewer deaths), and this is attributed to the greater attention paid to safer working practices in
NSW coal mines. In general the total number of deaths in coal mines in Queensland and New South Wales is high and should be of concern to employers, employees and government authorities.

A list of the prominent ones of these disasters are shown chronologically in Tables 1.1 and 1.2

<table>
<thead>
<tr>
<th>LOCATION</th>
<th>YEAR</th>
<th>TYPE OF INCIDENT</th>
<th>FATALITIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mount Mulligan</td>
<td>1921</td>
<td>Coal Dust Explosion</td>
<td>76</td>
</tr>
<tr>
<td>Redbank</td>
<td>1928</td>
<td>Gas Explosion</td>
<td>4</td>
</tr>
<tr>
<td>Hart's Aberdare</td>
<td>1936</td>
<td>Gas Explosion</td>
<td>4</td>
</tr>
<tr>
<td>Wood End</td>
<td>1945</td>
<td>Gas Explosion</td>
<td>4</td>
</tr>
<tr>
<td>Aberdare</td>
<td>1954</td>
<td>Gas Explosions</td>
<td>6</td>
</tr>
<tr>
<td>Collinsville</td>
<td>1954</td>
<td>Outburst of CO$_2$</td>
<td>7</td>
</tr>
<tr>
<td>Ipswich</td>
<td>1972</td>
<td>Gas &amp; Dust Explosion</td>
<td>17</td>
</tr>
<tr>
<td>Kianga</td>
<td>1975</td>
<td>Gas &amp; Dust Explosions</td>
<td>13</td>
</tr>
<tr>
<td>Leichardt</td>
<td>1978</td>
<td>Gas Outburst</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 1.1: Chronological list of recorded incident involving underground fires, explosions and outbursts, resulting in more than two fatalities in Queensland coal mines (1920-1979)

<table>
<thead>
<tr>
<th>LOCATION</th>
<th>YEAR</th>
<th>TYPE OF INCIDENT</th>
<th>FATALITIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bellbird Colliery</td>
<td>1923</td>
<td>Fire &amp; Explosion</td>
<td>21</td>
</tr>
</tbody>
</table>
Table 1.2: Chronological list of recorded incident involving underground fires, explosions, and outbursts, resulting in more than two fatalities in New South Wales coal mines (1920-1986).

<table>
<thead>
<tr>
<th>Date</th>
<th>Location</th>
<th>Event Description</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>1925</td>
<td>Metropolitan Coll.</td>
<td>Outburst of Coal</td>
<td>3</td>
</tr>
<tr>
<td>1926</td>
<td>Readhead Colliery</td>
<td>Gas Explosion</td>
<td>5</td>
</tr>
<tr>
<td>1954</td>
<td>Metropolitan Coll.</td>
<td>Outburst of CO₂</td>
<td>3</td>
</tr>
<tr>
<td>1965</td>
<td>Bulli Colliery</td>
<td>Underground fire</td>
<td>4</td>
</tr>
<tr>
<td>1979</td>
<td>Appin Colliery</td>
<td>Underground explosion</td>
<td>14</td>
</tr>
<tr>
<td>1986</td>
<td>Moura Colliery</td>
<td>Underground Explosion</td>
<td>12</td>
</tr>
</tbody>
</table>

The gradual evolution of the mining industry in Australia, the advance of knowledge of the causes of mine hazard, the adoption of innovative safety methods and the growth of mechanization and automation have played a significant role in reducing casualties in coal mines. Coal mining in the last 50 years has seen rapid changes in attitude of workers and management. Planning and development of mines followed strict guidelines of safety and placement of greater emphasis on job training and education generally became a reality. All these factors have contributed significantly to the standard of safety in Australian mines in recent years. Although the coal mining industry in Australia has not been slow in implementing on-the-job training in the area of accident prevention and safety in mines, there still appears to be deficiencies in these areas.
The laws and regulations written to safeguard health and safety differs from state to state thus raising some confusion and non-uniformity on the national scale. State laws written for state conditions are generally adequate with respect to personal safety. However, variations from state to state introduce a degree of incongruity which has often played an important part in reducing the overall progressive development and improvement of accident prevention in coal mines. Thus, albeit the industry has seen some sort of improvement in accident prevention, practically it is impossible to eliminate accidents completely.

1.1. Factors Arising Out of The Disasters

A careful dissection of the facts of the accidents of tables 1.1 and 1.2 revealed that most of the accidents were caused by human negligence, incompetence, inexperience and/or other combination of human elements.

The accidents of tables 1.1 and 1.2 could be divided into two broad categories:

1. Accidents caused by neglect, ignorance, carelessness and a general casual disregard for safety conditions. These accidents, could have been avoided without great difficulty and were termed "preventable accidents" because knowl-
edge and the state-of-the art were such that they could have been avoided or minimised.

2. The other category of accidents were caused by unpredictable factors or by causes about which either little or nothing was known at the time. Some of these accidents could have been avoided if sufficient research and investigations were conducted prior to the commencement of operations. These accidents were termed "unpredictable accidents" because knowledge and the state-of-the-art were such that they could not be foreseen or predicted at the time.

Most of the accidents were placed in the first category and even the few that fell into the second could still have been either completely avoided or minimised in effect if sufficient emphasis had been placed on research.

1.2 Research Into Fire Emergency

Since these accidents, a number of investigators and organisations involved with mine rescue operations in one way or the other, particularly the Southern Mine Rescue Station at Bulli in New South Wales, have made efforts to
improve their efficiency both in terms of equipment and training and awareness. In New South Wales, the Southern Mine Rescue (SMR) have upgraded their equipment and technical services as a result of their "below-standard" performances at the Avon Colliery fire emergency which occurred on the 29th August, 1975 and the Appin emergency on the 24th July, 1979. (Reference: A Critical Review of Coal Mine Disasters In Queensland and New South Wales Since 1920, P.K Chatterjee, University of Queensland).

Among the improvement done in the last few years at the SMR are:

a) the purchase of communication and gas monitoring equipment.

b) the setting up of technical services branch within the station. The technical services are responsible for the proper calibration, repairing and servicing of all gas monitoring equipment as well as all emergency equipment, provision of technical back-up to the gas monitoring team and the provision of consummate gas analysis services if required in an emergency.

Other improvement for the SMR include the evaluation and introduction of nomex flame-proof clothing for use in an emergency. The station has also a mobile emergency winder and accessories for recovering
personnel from the bottom of shaft with no winding facilities.

1.3 Definition of Problem

A number of individual researches and organisations, notably the U.S. Bureau of Mines, the CSIRO and others in Germany, United Kingdom and Japan have done extensive research into the use of computers in mine emergency and have also done further work on the fire ratios.

In fire emergencies in coal mines, the computer is used mainly for data storage, calculations for the fire ratios to access the progress of the fire, graphing of the important gas ratios such as CO/CO₂ as well as determination of the explosibility of the mine atmosphere. In other applications of computers in fire emergencies, the computer was used in various ways in assisting the officials in charge of the fire make important decisions to successfully extinguish the fire. Computer based ventilation simulations are invaluable tools. First the computer was used to predict the reduction of ventilation in the fire area if a particular service fan was taken out. Secondly, the computer can also predict the migration of methane to other parts of the mine. Also the computer can be used to select the optimum blade setting of fans for adequate ventilation. *(Interpretation of the State of a mine Fire by Computers, J.N. Fairbanks and R.G. Robinson, proceedings of the First*
The use of the computer in mine emergencies has concentrated on algorithm-based problems. Most practical problems, however, are very complex and require considerable amount of human expertise. If computers are to be used in solving problems of this nature, the machine should attempt to "mimic" the performance of a human expert in some intellectual endeavour. Despite the extensive use of computers in the mining industry, very little has been done in areas which incorporates the expert's intuitions and experience. This is the general problem area of concern for this thesis.

1.4 Scope of Work

This investigation is a software development of a suite of practical computer programs which incorporate the judgement, experience, rules of thumb, and mine environmental conditions as well as human intuitions in a mine rescue operation.

The software development for this research is divided into four main divisions. The first three of these programs follow conventional software development approach. The main purpose here is to allow users to supply gas sample results for analysis. This is achieved in a relational database manner. The addition of database man-
agement system allows the storage of large amount data, easy access and updating. The database system used was dBASE III+.

Owing to dBASE III+ inability to handle complicated calculations in general and trignometric functions in particular, a FORTRAN program was added to perform the fire ratio analysis. The results of the analysis are stored in the database management system together with the original input data.

In fire emergency cases graphs of fire ratios with time for the atmosphere are indispensable for decision making. These graphs are drawn with BASIC programs. The user is guided step by step through screen prompts throughout the operation. In addition to the graphs, users have a choice to ask for hard or soft copies of results of analysis in a tabular form.

The last program is an expert system which was developed in the VP-Expert shell. The choice to consult the expert system is among other choices from the menu. The consultation develops a dialogue between the user and the computer. The user answers relevant questions posed by the computer pertaining to the problem under investigation. The dialogue will proceed until the expert system arrives at a decision or the user decides to quit.
CHAPTER TWO
MINE FIRES AND RESCUE OPERATIONS

Mine explosions and coal outbursts are usually accompanied by fire. The presence of fire means that rescuers usually have two-fold task: that of rescuing people in danger, as well as dealing with the fire.

Accurate knowledge of the nature of fire is therefore necessary for the rescuer. This chapter explains the chemistry of fire and how the components of fire can be controlled in achieving this end.

2.1 The Nature of Fire

Fire is a chemical reaction, known as combustion, in which combustible material is oxidized rapidly. This causes a release of energy in the form of heat and light. For many years this was graphically depicted as triangle consisting of oxygen, fuel, and heat. (Extinguishing A Fire, A.V. Allan, 1985 Victorian Occupational Health and Safety Convention).

Recently, a refinement has been developed which promotes understanding of both burning and extinguishment. (Fire Loss Control, Global Guardian, pp 381-384) In this new concept the triangle has been replaced by a four-sided geometric figure called a "tetrahedron" or triangular pyramid as shown in Figure 2.1.
Fig. 2-1 The Tetrahedron of Fire.

Oxidizing Agent

Fuel (Reducing Agent)

Temperature

Uninhibited Chain Reaction
This fourth component may be described as "Uninhibited Chain Reaction". Thus the four sides are oxygen, fuel, heat and the chain reaction. Removal of any of these elements rapidly extinguishes the fire.

**Fuel**— This element is defined as any material that can be oxidized rapidly, such as wood, paper, oil, grease, certain metals, dust, gases and so forth. Fires are classified according to the type of fuel involved.

**Oxygen**— Fires need only 16% of oxygen by volume to burn. Normal atmosphere contains 21% oxygen. So fires are normally surrounded by plenty of oxygen to support burning. The more oxygen available the more intense the fire becomes. Some fuel contain sufficient oxygen within their make-up to support burning and, therefore, may burn in an oxygen-free environment.

**Heat**— Combustion is initiated when energy is applied to a system so that the temperature of a part of it rises to a stage where rapid reaction between an oxidizing agent and a reducing agent occurs with the liberation of heat. The heat liberated shows up a temperature rise in the combustion gases, and as radiated heat. If
the final temperature is above the level necessary to sustain combustion, and if enough of the radiant heat is utilized to vaporize replacement fuel in quantities greater than that originally used, then combustion will continue to an even increasing degree. Otherwise the fire will go out.

Chemical Chain Reaction—This reaction begins very early in the burning process and grows in intensity, feeding the fire. As the fire begins, "atomic molecule" or "free radicals" are thrown off and immediately drawn back into the base of the fire. These preheated atoms intensify the blaze by providing additional fuel, gases and oxygen. The fuel is initially heated, putting off gas or vapour that is ignited; but as the fire sets up the chain reaction, the fuel itself begins to burn, and may continue until it is all consumed.

2.2. Flammability and Explosibility Ranges

The proportion in which flammable gases are mixed is critical for ignition to take place. For example, the flammable range of motor spirit is from about 1.2% to 7.5% by volume in air. Outside this range the fuel/air
mixture is said to be either "too rich" or "too lean" for combustion. The boundary of the flammable range is not an exact value; it varies with a number of factors including the ambient temperature and pressure, as well as the volatility of the particular fuel. (A Manual On Mines Rescue, Safety and Gas Detection, J. Strang and P. MacKenzie-Wood, 1985).

Inside the flammable range lies the explosive range of a flammable gas. Within these limits, if vapour and air are intimately mixed, combustion may take place with explosive violence. If the concentration of flammable gas is below this range, then the mixture is too lean to sustain combustion. At concentrations above this range, insufficient oxygen is present to allow propagation of an ignition. The concentration of flammable gas corresponding to the lowest and highest points of this range are termed the lowest and highest explosive limits (LEL, UEL), respectively.

2.3 Fire Extinguishing Methods

The best method of stopping a fire depends on its size and the type of fuel involved. Some fires are best fought by attacking the supply of oxygen. With others, it is more practical to remove the fuel or heat. Certain types of fuel require very detailed plans that combine several techniques like cooling, oxygen depletion and dilution of the fuel, all in sequential steps. (Extinguishing A
The danger of explosion must also be considered, since some fuel will react violently to some extinguishing agents. An example is water applied to burning liquids or metals. The common fire fighting methods are briefly discussed below.

2.3.1 Fuel Removal (Starving)

This method ranges from the closing of a valve to the physical removal of objects likely to burn. It can also involve the wetting down of potential fuel. Starving may be achieved by the use of fire fighting foam, particularly with flammable liquids. The foam forms a layer on the surface of the fuel involved in the fire. This layer then prevents the formation of vapours to burn and the fire then goes out.

2.3.2 Oxygen Removal (Smothering)

Smothering extinguishes fire by separating or excluding the oxygen from the other elements that make a fire. Fires that do not depend on external fuel to burn, such as plastics (cellulose nitrate), metals (titanium) and certain other fuel cannot be put out by smothering.

Since normal atmospheric air contains 21% oxygen and a fire needs only 16% to burn, it means that most fuels are surrounded by sufficient oxygen for burning. So to some
extent, the exclusion of oxygen should be considered for proper control of all fires. It must also be recognized that while a fire needs 16% oxygen to burn freely, it may burn slowly and smoulder for a long time when less oxygen is available. In the case of smouldering in an area behind a seal, a fire may virtually explode into a ferocious fire when the door of the sealing is opened, providing a new source of oxygen.

2.3.3 Heat Reduction (Cooling)

This is actually controlling the temperature of the fire to the point where the fuel is not hot enough to give off the gas vapours which burning requires. Water is the most commonly used and practical way of cooling a fire. It is best applied as either a spray or a fog to take full advantage of its cooling properties. Of all the extinguishing agents, water absorbs more heat per volume than any other agent. A major effect of the use of water is that the steam formed from contact of water with hot surfaces acts as smothering agent by reducing oxygen access. One litre of water, when heated, produces 1244 litres of steam and in the process absorbs a very large amount of heat.

2.3.4 Interrupting The Chain Reaction

The fourth method of fire extinguishment is the interruption of the chain reaction in which the atomic molecule or
free radicals that have been preheated are thrown off from the blaze and drawn back into the base of the fire, rapidly increasing the fire intensity.

Certain chemical substances can break up this reaction. When introduced into the fire in the proper amounts, these substances inhibits the atoms and prevents the flame from burning. The most commonly used substances are Halon gases such as Halon 1301, Halon 1211, and Halon 2402. Halon is an odourless, clear gas that quickly replaces the free radicals or molecules. (Extinguishing A fire, A.V. Allan).

2.4 The Fire Ratios

The term 'fire ratios' is defined as 'the ratio of gases produced by a spontaneous heating and which are used to determine the state of the mine fire behind the fire seals'. (A Manual On Mine Rescue, Safety And Gas Detection, J. Strang and P. MacKenzie-Wood, 1985). Fire ratios are invaluable in deciding the proper action to take in fire emergencies. These ratios have seen gradual modification with time as more knowledge was gained of the chemistry of fire. Attempt is made in this section to recount some of the works done before, even though many of these ratios are not known to the present mining engineer.
Rhead & Wheeler (1910)

Rhead and Wheeler were the first researchers to consider the importance of the ratios of the product of combustion. The two researchers stated that the ratio \( \text{CO}_2/\text{CO} \) decreased as the temperature increased 2.(1).

Winmill & Graham (1913-16)

These reporters found that:

1. The ratio \( \text{CO}/\text{O}_2 \) increased as temperature increased. 2.(2)
2. The ratio \( \text{CO}_2/\text{CO} \) increased with time. 2.(3)
3. The ratio \( \text{CO}_2/\text{CO} \) decreased as the temperature increased. 2.(4)

Porter & Ralston (1914)

Porter and Ralston found that as the temperature rises the ratio \( \text{CO}_2/\text{CO} \) is more or less constant. 2.(5)

Bone (1918)

The researcher found that after 100 hours (from start of fire) the ratio \( \text{CO}_2/\text{CO} \) was produced in nearly the same relative proportion, and that \( \text{CO}_2/\text{CO} = 2.5 \) 2.(6)
Partington (1919)

Partington found that there was a fall in the ratio CO$_2$/CO as the temperature increased. 2.(7).
He further stated that the ratio N$_2$/oxides of carbon increased as temperature increased. 2.(8)

Graham (1921)

Graham proved that: As the temperature increased so does the ratio CO produced: percent O$_2$ absorbed (as noted earlier in 2.(2) above). This ratio, CO production/O$_2$ absorption, has been known to the world as Graham's ratio and has been used for many years to indicate the state of a mine fire behind the seals. 2.(9).

Storrow & Graham (1924)

The authors stated that CO$_2$ produced/O$_2$ absorbed increased as temperature increased, 2.(10)
and that CO produced/O$_2$ absorbed increased with temperature. These early fire ratios may be summarised as follows:

Statement 1:
The ratio CO$_2$/CO decreases as the temperature increases.
(From 2.(1), 2.(4), 2.(7))
Statement 2

The ratio CO₂/CO increases with time.  
(From 2.(3))

Statement 3

The ratio CO₂/CO is more or less a constant as the temperature rises.  
(From (5)).

Statement 4

The ratios CO₂/CO is a constant of 2.5 after 100 hours.  
(From 2.(6))

Statement 5

The ratio CO/CO₂ increases as the temperature increases.  
(From 2.(2), 2.(9), 2.(10), 2.(11))

Statement 6

The ratio N₂/[oxides of carbon] increases as the temperature increases.  
(From (8))

(Reference: R. Morris, _A New Fire Ratio for Determining Conditions in sealed Areas_).

In addition to the empirical formulas mentioned above, there are a number of numerical ratios which go to establish the condition of the atmosphere. The ratios discussed in this section are calculated using the volume percent concentrations of gases as defined below.

\[
\begin{align*}
\text{CO} & = \text{Carbon Monoxide, } \% \\
\text{O}_2 & = \text{Oxygen, } \% 
\end{align*}
\]
2.4.1 The Carbon Monoxide Index

Carbon monoxide index has been used for monitoring fires because the effects of sample dilutions are eliminated. The CO index is calculated as follows:

\[
\text{CO index} = \frac{\text{CO} \times 100.0}{(0.265 \times N_2) - O_2}
\]

An increase in the CO index indicates an increase in oxidation and vice versa. An increase in the CO index from some stable level should be investigated and corrective action taken to reverse the trend. A sample plot of CO vs. time is shown as Figure 2.2

2.4.2 Airfree Ratios

Air free ratios of the various fire gas concentrations also eliminate the effect of sample dilution because the O\(_2\) concentration has been factored out. The airfree CO concentration is of interest and is computed as follows:
Figure 2.2 A Sample Carbon Monoxide vrs. Time Plot

Figure 2.3 Typical Airfree vrs. Time Plot
2.4.3 The CO/CO₂ Ratio

The Carbon Monoxide/Carbon Dioxide ratio is critical for determining the state of a coal mine fire. When the CO/CO₂ ratio reaches a value of between 0.10 and 0.15 the fire is thought to be in the flaming stage.

2.4.4 The Trickett Ratio

Trickett's ratio is considered an indicator of diluted samples and the type of fuel which is burning. The Trickett's ratio is defined as follows when air to the fire contains normal oxygen.

\[
\text{Trickett's Ratio} = \frac{\text{CO}_2\% + 0.75\text{CO}\% - 0.2\text{H}_2\%}{0.265\text{N}_2 - \text{O}_2} \quad (3)
\]

When the ratio exceeds 1.6, the sample should be considered invalid. The following ranges have been given as indicative of the primary type of fuel burning.

A typical Airfree value for CO is shown in Figure 2.3
Trickett’s Ratio = 0.4 - 0.5  Methane
= 0.5 - 1.0  Coal, Oil, Conveyor belt
= 0.9 - 1.6  Wood
= 0.0 - 0.87  Coal dust
> 1.6  Invalid.

(Source: J. Strang and P. MacKenzie-Wood)

a) Trickett’s ratio will not work if the intake air is oxygen deficient through the injection of nitrogen or carbon dioxide or through a high methane make.

b) Fresh air dilution of the fire products has no effect on Trickett’s ratio.

c) If Trickett’s ratio is less than 0.4 then there is no fire and the gases are residual rather than active.

2.4.5 Graham’s Ratio (CO/O₂ Deficiency Ratio)

This ratio increases with the speed and intensity of a heating and is of great importance. It is the ratio between the percentage of carbon monoxide in the air and the percentage of oxygen absorbed at a given sampling point.

\[
\text{Graham’s Ratio} = \frac{79.03 \times 100.0 \times \text{CO}\%}{(20.93 \times N_2) - (79.03 \times O_2)} \quad (4)
\]
The carbon monoxide content can be affected by fumes from diesel engines and shotfiring. Graham's ratio can be used to differentiate between a fire and a heating as follows:

**Fresh Coal**

- **Graham's Ratio:** 0.5 - 1.0
  - Represents a heating
- 2.0 - 10.0
  - Represents a fire

**Old Coal**

- **Graham's Ratio:** 1.0 - 2.0
  - Represents a heating
- 10.0 - 20.0
  - Represents a fire.


**2.5 Rescue Organisational Procedure**

The likely sequence of events following an outbreak of fire is depicted by a flow chart as Figure 2.4. *(Source: Beatty, John, University of Wollongong, Personal Consultation).*

1. **ALARM** - The fire can either be observed in its early stages or detected by monitors or remote sensing instruments in which case an urgent site inspection is made to confirm the event. Smoke and heat have built up, atmosphere is fouled with contaminants which circulate towards the exhausting fans.
Fire Emergency Procedure

- **EVACUATION OF MEN**
- **ALARMD**
- **FIRST AID FIRE FIGHTING**

- **MINE RESCUE STATION ON STANDBY**
- **MANAGER'S ASSESSMENT**
- **MANAGER DECLARES EMERGENCY CONTACTS RESCUE STATION, UNION, DIRECTOR**

- **ENGINEERS & FOREMEN CHECK SERVICE FACILITIES & OTHER EQUIPMENT ARE IN GOOD CONDITION**

- **FIRE FIGHTING OPERATIONS**
- **ESTABLISH OPERATIONS ROOM EXPERT SYSTEM & OTHER PROGRAMS AVAILABLE**
- **GAS ANALYSIS LABS**

- **PROGRESS INFORMATION AND FEED BACK**

**Figure 2.4**
The observer will inform an official on the surface and evacuation of men is commenced. Underground official(s) observe and record the fire and commence fire fighting.

2. DECLARATION OF EMERGENCY: This may happen on the first alarm but certainly as the fire fighting attempts are reported as having little effect, and will be done by the manager or most senior official available. The report is made to the Rescue Station which marshals trained rescue teams immediately to the mine. The mine activates its manual aid scheme. Specialized services are put on standby and staff attend to preparation functions.

3. EVACUATION OF THE MINE: The evacuation of the mine is carried out as per the statutory rules of the manager. The employees of each colliery follow defined paths. Within one half hour, either all men will have been accounted for or will be presumed to be in peril and will have a full scale rescue operation mounted for them. It could be that rescue and initial fire fighting work could go on simultaneously.

4. MINE RESCUE SERVICES - GAS MONITORING:

Rescue teams from adjacent mines are summoned. When two teams have arrived, orders from the manager in the operation room direct their work. This will be finely defined by the Rescue Superintendent prior to the first team departing for underground; the second team will be
team departing for underground; the second team will be on stand-by. A gas monitoring team is likely to depart with this team. The team is utilised to establish flammable gas percentages in and around the fire. The gas monitoring team will, by use of portable analysers, monitor fire products for this duration.
ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEMS

An expert system is a program that mimics the performance of a human expert in some intellectual endeavour. The archetypal expert system attains this high level of proficiency by embodying the heuristics, informally framed knowledge of the human expert, along with the expert's method of reasoning in the problem domain.

Expert systems can be distinguished from other forms of artificial intelligence program in that:

1. It deals with subject matter of realistic complexity that normally requires a considerable amount of human expertise;

2. It must exhibit high performance in terms of speed and reliability in order to be a useful tool;

3. It must be capable of explaining and justifying solutions and recommendations in order to convince the user of the correctness and reliability of its reasoning.

Expert systems are a special class of the State-space search paradigm, (Artificial Intelligence, P.H. Winston). Here the focus is on the "knowledge" that goes into the definition of heuristics. Expert System programs search
for a "path" from a possible class definition back to observable data (search space is implicitly a completely specified graph defined by "rules", this graph is actually dynamically reconstructed in solving each given problem). (Artificial Intelligence, P.H. Winston).

The basic structure of an expert system, as depicted in Figure 3.1 consists of the following:

1. A knowledge-base
2. An inference engine or control structure
3. A working memory
4. A natural language processor

3.1 The Knowledge-Base

This is generally a collection of empirically modifiable rule which contain the knowledge and expertise about a domain, elicited from an expert or a group of experts. (Introductory Readings in Expert Systems, Donald Michie). It contains general facts such as methane gas is explosive, Class-A fire involves carbonaceous materials, etc. It also contains heuristics, rules of thumb or rules of good judgement that the expert uses in arriving at a decision. The knowledge-base constitutes the long term storage of the Expert System program. An example of a heuristic is "since deeper coalbeds are usually gassier, the Appin Coalbed (depth = x metres) PROBABLY has higher methane emission rate than the
Figure 3-1. Architecture of Expert System
Westcliff Coalbed (depth = y metres), where x is assumed to be numerically greater than y. This type of knowledge is stored in production rules, which in the case of this thesis takes the form of IF.....THEN statements. The most important part of the anatomy of any knowledge-based system is the knowledge base, because the performance of a knowledge-based system depends on the knowledge it has.

3.2 The Inference Engine

The inference engine stands between the user and the knowledge-base and provides two functions. First, it provides the strategy for drawing inferences from the production rules stored in the knowledge-base and new data provided by the user. Its second function is to control the order in which the expert system draws inferences and communicates with the user. Inference strategies in rule-base knowledge systems can be goal driven (backward chaining) or data driven (forward chaining), or a combination of these.

3.2.1 Backward Chaining Inference Engine

In the backward chaining method of inference strategy, one starts by considering a goal, in this case a fact whose truth value is desired. The key step is to find a way of determining the truth values of each of the rule's antecedents (the IF part of the IF....THEN rule). This
reformulation of problems into subproblems is iterated until each of the subproblem can be solved directly, either by finding facts in the knowledge base, or by asking questions of the user. For this reason backward chaining is also called *goal-driven reasoning*.

### 3.2.2 Forward Chaining Inference Engine

The second inference control strategy is called *forward chaining*. In forward chaining, one compares the facts in the knowledge base with the antecedents of the various rules, trying to fire any rules possible. When a rule fires, its conclusion are added to the knowledge-base and can potentially trigger other rules. Because attention is focused on matching antecedents against the facts that are known, this method is also called *data-driven reasoning*.

### 3.3 The Working Memory

The working memory is the "*scratch pad*" where dynamic information is kept about the specific problem being analysed. For example, gas sampling data and results for mine rescue operations and other information such as panel name, number of sample points etc. and any inferences drawn about the problem, (for example, gas atmosphere around location D4 in panel NORTH-2-SECTION is potentially explosive) would be stored here.
3.4 Natural Language Processor

The natural-language processor provides a simple method of querying the user for input then supplying him back with results that are easily understood. For example, the user may be queried in various ways in an attempt to determine relevant information: multiple choices, TRUE or FALSE and YES and NO questions. A simple TRUE or FALSE question may be:

is it true that:

The colour of the smoke is black?

TRUE FALSE

3.5 Knowledge Acquisition

An expert system acts like a real expert because it has embedded in it the factual and experiential (heuristic) knowledge of a real expert. Getting the knowledge out of the real expert and into a form that can be used by a computer is a difficult job. For one thing, it requires an expert who is both cooperative and articulate. For another, it requires a person who knows something about computers, artificial intelligence, psychology, statesmanship, and the expert’s application area in order to interview the expert intelligently and convert the knowledge acquired from the interviews into a computer-
usable form.

The person who interviews the expert and acquires and represents the knowledge is known as a knowledge engineer. The process of acquiring the knowledge and representing it for the computer is known as knowledge engineering. Knowledge engineering is a labour-intensive process that takes place over a long period of time.

At the moment knowledge acquisition is not (yet) a science which means that the person building the knowledge system must devise his own method and style of dealing with the expert. A number of suggestions and techniques are available however to assist knowledge engineers with knowledge acquisition. Table 3.1 summarizes the known techniques.
<table>
<thead>
<tr>
<th>METHOD CATEGORY</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>METHOD OF &quot;FAMILIAR&quot; TASK</td>
<td>Analysis of the tasks that the expert usually performs.</td>
</tr>
<tr>
<td>STRUCTURED AND UNSTRUCTURED INTERVIEW</td>
<td>The expert is queried with regard to knowledge of facts and procedures</td>
</tr>
<tr>
<td>LIMITED INFORMATION TASKS</td>
<td>A familiar task is performed, but the expert is not given certain information that is typically available</td>
</tr>
<tr>
<td>CONSTRAINED PROCESSING TASKS</td>
<td>A familiar task is performed, but the expert must do so under time or other constraints.</td>
</tr>
<tr>
<td>METHOD OF &quot;TOUGH CASES&quot;</td>
<td>Analysis of a familiar task that is conducted for a set of &quot;tough case&quot; for the expert.</td>
</tr>
</tbody>
</table>

Table 3.1 Types of methods that can be used to extract knowledge of an expert. (Source: AI Magazine, Summer 1988, volume 9. No 2).

3.5.1. The Method of Familiar Tasks

The method of familiar task involves studying the expert while he is engaged in the relevant task. Looking across a set of an expert’s specific tactics and procedures, one should see commonalities in terms of goals, information the expert likes to have available, and the data or records that are produced.
Psychologically, the task that an expert typically performs involves at least the following: (1) the analysis of complex stimuli into relevant features or cues based on a process called "perceptual learning", (2) the analysis of conceptual categories in terms of relevant features (the perception of similarities and differences), (3) the analysis of features and the categories in terms of relevant underlying causal laws (involving "concept formation processes"), and (4) ability to infer and test hypotheses.

An analysis of familiar tasks (including an analysis of available texts and technical manuals) can be used to generate a "first pass" at a data base. What the expert knows is represented as a categorized listing of statements cast in some sort of formal language (such as propositions) using terms and categories that are meaningful and related to the domain at hand. Such propositions can express observation statements or facts as well as implications or potential IF-THEN rules of inference.

3.5.2 The Unstructured Interview

The development of most existing expert systems has been accomplished with unstructured interviews of the expert. In an unstructured interview, the knowledge engineer asks more-or-less spontaneous questions of the
expert while the expert is performing (or talking about) a familiar task. For instance, the interviewer might ask the expert a question such as "How do you know that?" whenever the expert seems to tap into knowledge or make an inference. The engineer should have the ability to follow the expert's reasoning lines and to ask relevant questions when necessary. Knowledge engineers sometimes make an audiotape of the expert's ruminations; this recording is called a verbal protocol. The trick is to train the expert into thinking aloud.

Audiotapes are necessarily a partial representation of the key information. An expert's facial expression and gestures can also reveal inference-making processes.

3.5.3 The Structured Interview

An alternative to the unstructured interview, the structured interview combines an analysis of familiar tasks with an unstructured interview. In order to add structure to an interview, the knowledge engineer initially makes a first pass at a data base by analysing the available texts and technical manuals, or by conducting an unstructured interview. The expert then goes over the first-pass data base one entry at a time, making comments on each one.

A structured interview in one way or another forces the expert to systematically go back over the knowledge. Any given comments can have a number of effects on the data
base. It can lead to (1) the addition or deletion of entries, (2) the qualification of entries, (3) the reorganisation of the hierarchical or categorical structure of the data base, or (4) the addition or deletion of categories. The results is a second pass of the data base.

3.5.4. Limited Information Tasks

Limited-information tasks represent the application of basic scientific methods: to understand how something works in nature, it should be tinkered with. Limited information tasks are similar to the familiar tasks but the amount or kind of information that is available to the expert is somehow restricted. For example, an expert aerial-photo interpreters like to have all sort of maps available during the interpretation of photos. In the limited-information task, such contextual information can be withheld, forcing the expert to rely heavily upon (and hence provide additional evidence about) his knowledge and reasoning skills.

The limited-information task can be used to provide information about sub-domains of the expert's knowledge, to fill in any gaps in the data base or a set of inference rules.

3.5.5 Constrained processing tasks

Constrained processing tasks are like limited-information
tasks in that both involve tinkering with the familiar task. Constrained processing tasks involve deliberate attempts to constrain or alter the reasoning strategies that the expert uses. One simple way to achieve this goal is to limit the amount of time that the expert has in which to absorb information or make judgements. For example, the interpretation of aerial photos takes hours, but in a constrained processing task the expert might be allowed, say, two minutes to inspect a photo and five minutes to make judgement about it.

Another way to constrain the processing is to ask the expert a specific question rather than to require the full analysis that is conducted during the familiar task. Two subtypes of a constrained processing task that involves this single-question processing are what can be called the method of simulated familiar tasks and the method of scenarios.

3.5.5.1 The Method of Simulated Familiar Tasks

Here, a familiar task is performed using archival data. The expert has to do a diagnosis but in displaced real time. In a simulated familiar task the clock can be stopped at any point and the expert queried with regard to reasoning strategies or subdomain of knowledge as incoming data are monitored and predictions are made.
3.5.5.2. The Method of Scenarios

While analysing cases, experts draw analogies to previously encountered situations or cases. A given case is explored in terms of any relevant or salient similarities and differences relative to a previously encountered case. For example, expert mechanical and electrical engineers design new components by analogy to other components that perform similar functions. Expert weather forecasters make predictions about the growth of storms based on analogies to previously experienced storms.

One type of constrained-processing task involves having the interviewer deliberately encourage the use of scenarios during the performance of a familiar task. Such a practice should evoke evidence about the expert's reasoning for the kinds of scenarios involved in the data at hand.

3.5.6. Combined Constrained

A task can involve combining limited information constraints with processing constraints. For example, the expert aerial-photo interpreter could be asked to interpret a photograph without the benefit of maps and with only two minutes to view the photos.
3.5.7. The Method of Tough Cases

Subtle or refined aspects of an expert's reasoning are often manifested when an expert encounters a tough case, a case with unusual, unfamiliar, or challenging features. Almost by definition, tough cases are rare.

The engineer is not likely to be present when an expert encounters a tough case; so, the expert is equipped with a small tape recorder and is instructed how to make a verbal protocol of personal ruminations when encountering a tough case.

3.6. Comparison of The Methods

One way to compare the above methods is in terms of their relative advantages and disadvantages. Table 3.2 summarizes these points.

<table>
<thead>
<tr>
<th>METHOD</th>
<th>ADVANTAGE</th>
<th>DISADVANTAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANALYSIS OF FAMILIAR TASKS</td>
<td>The expert feels comfortable.</td>
<td>Can be fairly time consuming.</td>
</tr>
<tr>
<td>INTERVIEWS</td>
<td>For a first and second pass at a data base, it can generate much information.</td>
<td>Typically very time consuming.</td>
</tr>
</tbody>
</table>
LIMITED INFORMATION TASKS Can be tailored to extract information on selected subdomains of knowledge, on the expert's strategies. Expert feels uncomfortable and is hesitant to make judgement.

CONSTRAINED PROCESSING Can be tailored to extract information of knowledge, on the

ANALYSIS OF "TOUGH CASES" Can yield information about refined reasoning. Occurs unpredictably. The knowledge engineer may not be present.

Table 3-2. Some salient advantages and disadvantages of the various methods for extracting an expert's knowledge. (Source: AI Magazine, Summer 1988, volume 9, No 2).

3.6.1 Task and Material Simplicity The method of familiar tasks and the method of tough cases are essentially equal in terms of the simplicity of the task and the materials, as are the limited-information and constrained-processing tasks. For all these methods, the instructions involved are about a page or so long, and the materials will be one or a few pages of information, displays, graphs, and so on. The structured interview stands out because it requires a first-pass data-base, which can itself consist of a relatively large and complex set of materials.
3.6.2 Task Brevity

Ideally, one wants to disclose the expert's reasoning as quickly as possible. Familiar task can take anywhere from a few minutes (gas sample analysis) to an hour (weather forecasting) to an entire day (aerial-photo interpretation). Interview takes on the order of days or even weeks. Although a time consuming process, interviews can yield a great deal of information, especially when used in the initial phase of developing a database. Limited information and constrained processing tasks can be intensive and should be designed to take somewhere between 15 and 45 minutes. Although they are not very time consuming, these tasks do require time for transcribing and coding the audiotapes.

3.6.3 Task Flexibility

Ideally, the task should be flexible: it should work with different sets of materials and with variations in instructions. For some experts, abundant information about reasoning can be evoked by the simplest of questioning. For others, the verbalisations might be less discursive. For some experts, a tape recorder might be absolutely necessary; for others, short-hand notes might suffice.

3.6.4 Task Artificiality

The task should not depart too much from the familiar
task. The further the task departs from the usual problem-solving situation, the less it tells the knowledge engineer about the usual sequence of mental operations and judgements. However, deliberate violations of the constraints that are involved in the familiar tasks (that is, tinkering with the familiar tasks) can be used to systematically expose the expert's reasoning.

3.6.5 Data Format

Ideally, the data that results from any method should be in a format ready to be input into the data base. Only the structured interview has this characteristic. The other methods can result in verbal protocol data, which need to be transcribed and analyzed.

3.6.6 Data Validity

The data that results from any method should contain valid information, valid in that it discloses truths about the expert's perceptions and observations, the expert's knowledge and reasoning, and the method of testing hypotheses. In addition to validity in the sense of truth value, one would like to feel that a data-base is complete— that it covers all the relevant subdomains of knowledge. One would also like to know that different experts agree about the facts. This agreement is validity in the sense of reliability or consensus across experts. Finally, one would like to have some assurance that a set of facts include the most important ones.
3.6.6.1 Validity: The Importance of The Data

Some of the expert's statements will obviously be irrelevant (e.g. "Do you want a cup of coffee?"). Excepting these statements, the knowledge engineer generally takes it for granted that a given statement by the expert is relevant to the domain of expertise being studied. However, just because a given statement (or datum) is relevant does not mean that it is at all important.

Validity in the sense of relative importance of a given fact or rule can be assessed in a number of ways. One way is the direct approach - ask the expert some more-or-less structured questions to get his judgement of the relative importance of a data-base entry. Another way to assess importance is to see how early or how often a given fact crops up in the data from various experts.

3.6.6.2 Validity: Data Reliability or Consensus

The running of additional experts in a knowledge-extraction task presumably has its major effect the generation of lots of information which is redundant in that it repeats data which are already in the data base (having come from earlier studies of some other experts). The redundancy can be taken as evidence of the validity of the data, validity in the sense of agreement or reliability across different experts.
3.6.6.3 Validity: The Completeness of The Data-base

Having more than one expert perform various tasks can also help assure validity - in the sense of completeness of the data-base. In addition to their idiosyncratic knowledge, various experts may also have idiosyncratic strategies. Thus, having more than one expert go through the various knowledge-extraction tasks should have the effects of "pushing" the data-base towards completeness by filling in the gaps.

3.7. Tools and Languages for Expert Systems

Expert system tools are programming systems that simplifies the job of constructing an expert system. They range from very high-level programming languages to low-level support facilities. These tools improve the ability to transfer the technology and shorten the training time. The power of these tools lie in their ability to be used by people with no Artificial Intelligence expertise. Expert system tools can be divided into four major categories (Yoshiaki Shirai et al: Artificial Intelligence; Concepts, techniques, and application) as shown in Figure 3.2.

A programming language is an artificial (as opposed to natural) language developed to control and direct operation of a computer. One class of programming language is
Figure 3.2. Types of tools available for expert system building.
known as knowledge engineering languages. They are especially designed for constructing and debugging expert systems. Knowledge engineering languages are good for the building of expert systems, but compared with ordinary programming languages, they are less flexible with regard to how knowledge can be represented and manipulated.

The programming languages used for expert system applications are generally either problem-oriented languages, such as FORTRAN and PASCAL, or symbolic-manipulation languages, such as LISP and PROLOG. Problem-oriented languages are computer languages designed for a particular class of problems, for example, FORTRAN, designed for efficiently performing algebraic computations, and COBOL, with features for business record keeping. Symbolic-manipulation (or processing) languages are designed for artificial intelligence applications; for example, LISP has mechanisms for manipulating symbols in the form of list structures. A list, in this context, is simply a collection of items enclosed in parentheses, where each item can be either a symbol (example of symbols are 56, methane-gas) or another list.

3.7.1. The LISP Programming Language

LISP is a computer programming language designed specifically to manipulate symbols rather than numeric data. A LISP data element is a list of symbols that may represent any object, including its own list processing functions. A
LISP program essentially consists of collections of independent procedures called functions.

Four key points differentiate LISP from traditional programming languages. First the basic programming units are known as S-expressions (for "symbolic expressions"). An S-expression may be an atomic symbols such as H, 4, PLUS, or GASCOLOUR. Or it may be a list of atomic symbols (or other S-expression, thus making this definition recursive).

The atomic symbols or list are grouped together in parentheses such as (PLUS 3,4), (BLACK GAS COLOUR), (PANEL No4).

LISP which stands for LIST Programming suggests that a main feature of LISP is its ability to manipulate lists. The first two lists above can be evaluated directly (where "evaluation" means "performing the function indicated by the first atom in the list"). The last two lists (which describes gas colour and a panel number) might represent knowledge in a frame base expert system. The second key point about LISP is that LISP uses what is called prefix notation to express functions. In prefix notation, operators, such as PLUS, are written first, followed by operands (such as numbers to be added). In contrast, algebra uses infix notation, such as 4 plus 3 where the operator is located between the operands. LISP has a host of primitive functions (such as PLUS).
The programmer uses these primitive functions to define other functions which can be called and used as any other primitive function. For example, one can define a function PLUS10 (any function name will suffice) which when operated on a number will always add 10 to the operand. The function PLUS10 is defined as follows:

(defun plus10 (x)
  (plus x 10)).

It can be seen that PLUS10 has been defined in terms of the primitive function PLUS. Any number on which the function PLUS10 is operated upon will yield the results of adding 10 to the number. Thus (PLUS10 20) will yield 30 as the answer, and (PLUS10 25) gives 35. The third point is that expressions in LISP are independent functions that state an action or purpose. Because LISP programs are composed of independent functions, and each function contains independent separate S-expressions, many sets of parentheses are required to clarify LISP programs for computers. In addition to mathematical functions such as PLUS, there are other functions called predicates. A predicate evaluates to true or false. Example of predicate functions are EQUAL, LESS etc.

LISP functions are evaluated independent as soon as a function is entered. This fact makes LISP a highly interactive language which gives fast answers.
The fourth key point about LISP is that lists of LISP can be broken down into lists and atoms. For example CAR of (PLUS 3,4) gives the atom PLUS and the CDR gives the list (3,4).

There are different (non-standard) versions of LISP. The major versions include Maclisp, developed at the Massachusetts Institute of Technology and Interlisp, developed at Bolt Beranek Research Centre both in the United States. An extension of the Maclisp version of LISP is Zetalisp. An attempt of standardisation of the various versions have led to a version called the Common LISP. A host of other LISP dialect exist. These include FranzLisp which runs under Unix, Portable Standard LISP, Cambridge LISP, Golden LISP and others.

3.7.2 The PROLOG Programming Language

PROLOG stands for "PROgramming in LOGic" and is a computer programming language that was invented around 1970 by Alain Colmerauer and his colleagues at the University of Marseille, France. There was not a lot of interest in PROLOG until the Japanese decided to use the language as the main language for their Fifth-generation computer project. The major advantage of PROLOG are its reliability, and relational data-base, natural-language interface, and parallel processing capabilities. Like logic statements, it is easy to read PROLOG statements, declaratively, as English statements about what is true in the
world. For example statements such as:

explosive (methane-gas)
poisonous (carbon-monoxide)
colourless (carbon-monoxide)
colourless (carbon-dioxide)
colourless (water)
gas (carbon-dioxide)
gas (carbon-monoxide)
liquid (water)
?- colourless (X)
?- colourless (X), poisonous (X), gas (X)

are typical PROLOG declarative statements. These statements can form part of a PROLOG relational data-base. The last two statements are queries. The last but one statement asks for all colourless variables. Since PROLOG has in its data-base that water, carbon-monoxide and carbon-dioxide are all colourless these three items will be produced as PROLOG answers to the query. On the other hand the last query asks for all items that are colourless, poisonous and gas. Only one variable, namely carbon-monoxide meets all the conditions. Even though PROLOG knows that water is colourless and carbon-dioxide is both colourless and a gas, these two will not be given as answers to the query because they are not stated as poisonous. The "X" in the statements indicates the missing information that the query is trying
to find. Logic statements such as these make it easy to implement a data-base and query language in PROLOG. Unlike conventional data-base which only stores conventional data, PROLOG data-bases may also contain rules.

Another advantage of PROLOG is its suitability for writing natural language front ends for data-bases. The reason is that the PROLOG syntax allows the rules of natural-language grammar to be expressed directly as PROLOG statements. The PROLOG processor automatically parses the English sentences and applies the appropriate rules in order to represent the meaning of the sentence. Like LISP there are different dialects of the PROLOG language, the most popular for the personal computers are MICRO-Prolog and TURBO-Prolog.

3.7.3. The Expert System Shell

Expert system shells are computer packages that simplify the effort required in building an expert system. These shells basically contain inference engines and various user interfaces. In addition to these basic requirements, others contain knowledge acquisition aids. The shells, however, lack knowledge-base; in other words the expert system shells are empty expert systems. The addition of a specific knowledge-base transforms the shell into an expert system. The expert system shells can be liken to electronic spreadsheets; when the user of an electronic spreadsheet enters financial data, he creates
a system that will do a specific financial projections just as the knowledge engineer uses a shell to create an expert system that will offer advice about a specific type of problem.

The Expert System for this thesis has been developed in the VP-EXPERT SHELL. VP-EXPERT is a system development tool that provides the inference engine, the user interface, the commands-- indeed, everything needed to develop a working expert system.

3.7.3.1 Special features of VP-Expert include:

* The ability to exchange data with dBASE data-base files, worksheet files and ASCII text files.

* An induced command that automatically creates a knowledge-base from a table contained in a text, data-base, or worksheet files.

* An inference engine that uses backward and forward chaining for problem solving.

* Optional development windows that allows observation of behind the scenes paths of the inference engine as it navigates the knowledge-base to solve problems during a consultation.
Confidence factors.

Simple English rule construction.

The ability to explain its actions during a consultation.

A built in text editor

Automatic question generation.

The ability to record and graphically display the rule-by-rule search pattern used behind the scene during consultation.

Floating point mathematical function.

The ability to execute external DOS programs.

3.7.3.2. VP-Expert Inference Engine

The problem-solving method used by VP-Expert is the backward-chaining inference engine. The inference engine starts by identifying the goal variable and then moves through the sequence of rules until it finds a value that can help it assign a value to the goal variable. This is the pattern it follows:
RULE 1

IF B = value
THEN A = value;

The first rule showing A in its conclusion is found. The variable B (shown in its premise) is not known, so the inference engine looks for the first rule showing B in its conclusion.

RULE 2

IF C = value
THEN B = value;

This rule naming B in its conclusion is found. The variable C (shown in its premise) is not known, so the inference engine looks for the first rule showing C in its conclusion.

RULE 3

IF D = value
THEN C = value;

This rule naming C in its conclusion is found. The variable D (shown in its premise) is not known so the inference engine looks for the first rule showing D in its conclusion.
3.7.3.3. VP-Expert knowledge-base

VP-Expert's knowledge-base file contains three basic elements:

The ACTIONS block.

Rules.

Statements.

A fourth element, Clauses, are contained within the ACTIONS block and rules of the knowledge-base.

3.7.3.3.1. The ACTIONS block

The ACTIONS block (which consists of the keyword "ACTIONS", one or more clauses, and a semi-colon) defines the problems ("goals") of the consultation and the sequence of their solution. In other words the ACTIONS block tells the inference engine what it needs to find out, and in what order. This is accomplished with the FIND clauses that instruct VP-Expert to "find" the value or values of one or more "goal variables."

3.7.3.3.2. VP-Expert Rules

Rules, stated as IF/THEN propositions, contain the actual "knowledge" or "expertise" of the knowledge-base.
For example, "if the animal has hair and bears young ones then it is a mammal." Stated as proper rule, this information may appear like this:

RULE 1

IF hair = yes AND birth = live

THEN animal = mammal;

Rules have four essential elements:

1. The rule name
2. The rule premise
3. The rule conclusion
4. A semi-colon at the end of the rule.

3.8. Expert System Application in the Mining Industry

The potential application of artificial intelligence technology, especially expert systems is quite promising. Some potential application areas may overlap with other areas of application such as planning, monitoring and scheduling. The following are all possible application areas in mining:
1. Mine Machines monitoring
   a. Trucks
   b. Draglines
   c. Shovels
   d. Continuous miners
   e. Conveyor belts
2. Power systems
3. Ventilation and methane drainage
4. Transportation
5. Geology
6. Face Geometry
7. Personnel
8. Maintenance
9. Robotic systems
10. Mine Rescue Operations


With monitoring, expert systems can be used to control and monitor the many operational parameters that can increase safety and production. However, the sheer volume of information produced by these monitoring systems makes it difficult for human operators to handle them efficiently to the best possible advantage to the mine. In addition there is no exception to the rule that these information need expert analysis for inherent harbingers of otherwise imperceptible problems for corrective measures to be taken (on the piece of equipment being monitored) if necessary.
Many mining companies have automatic monitoring systems installed on important piece of equipment. On a shovel or dragline (say), devices can be installed to monitor parameters such as: -

a) Swing time  
b) power consumption  
c) diggability  
d) ground conditions  
e) downtime and maintenance requirement

This information can be used to plan for things like: 

a) production scheduled  
b) drill hole pattern design  
c) maintenance scheduled etc.

With monitoring for example the flow of data from the monitor (shovel, say) to the expert system can be in one of two forms: -

1) Continuous, in which the expert system continuously updates its knowledge-base and determines if changes to monitored equipment requires action on its part.

2) On as-as-needed bases, in which data flow occurs only when a decision is requested by the operator.
In the first case, the expert system continuously determines the optimum strategy for the shovel conditions. Changes are recommended or carried out to make sure that economic and safety goals are satisfied. This situation is more of a real-time expert system. In the latter case, the operator will initiate the data flow to the expert system; hence the decision-making process is initiated when a problem arises. Here, data flow to the expert system will be made up of the current shovel status plus historical data. In this case the expert system will be used when necessary.

In either case, once the decision-making process is initiated and an action for the situation determined, that action could be carried out manually or automatically.
CHAPTER FOUR

STRUCTURE AND DESIGN OF RESCUE SYSTEMS

4.1 Model Formulation

Many computer programs have been developed to aid mine rescue personnels make decisions and determine the explosibility status of the affected mine’s atmosphere.

For efficient execution of any rescue operation detailed and up-to-date information is required of the changing conditions of the fire and the surrounding atmosphere. This requires that a large number of data be taken at close time intervals. For a fire which proves to be difficult to handle, its control can take days or even weeks to achieve. In this situation, since a lot of data will be required, existing computer programs will be less useful since they do not have data storage facilities.

This shortcoming has been taken care of in this research. In addition to the availability of a database system, an expert system has also been incorporated to aid in decision making.

The complete system is made up of four programs, each performing a specific task and they are interconnected via the expert system shell.

The first of these programs is written in dBASE III+ data
base management system and allows input of raw gas sample data.

The second program is written in FORTRAN and this program accepts data from the database management system after the data have been transformed into a "System Data Format" (SDF) file which is essentially an ASCII code. After analysis by the FORTRAN the results of the computation are again appended to a database file.

The third program is written in BASIC. This program consists of a number of small subprograms, each one plotting a specific graph for explosibility interpretation.

The last program of the system is written in VP-EXPERT, an Expert System Shell, and this contains the main Expert System program. Figure 4.1 illustrates the overall structure of the system.

4.1.1 The FORTRAN Program

Even though raw gas sampling data is entered directly into the database system the computational analysis is done by a FORTRAN program. This is because the analysis is quite extensive and requires functions such as trigonometric functions which are not available within DBASE III+.
Figure 4.1 A flowchart of the program.
The listing of the FORTRAN program, called GASANAL.FOR (gas analysis), is given in Appendix A. In designing the algorithm for the program, three classes of variables were identified:

** Variable Class 1: User supplied variables.

These variables include all the gas input, namely methane, carbon monoxide, hydrogen, oxygen, carbon dioxide and nitrogen. The percent concentrations of these gases in the atmosphere will be measured directly and will be supplied to the FORTRAN program via the database. Other user-supplied variables are not mentioned among the class 1 variables because even though they are supplied by the user, no computation is performed on them. These variables include temperature, pressure, humidity, time and date.

** Variable Class 2: Constants.

The following constants which have been established by experiments were used in the computation for gas analysis:

LELCO: Lower Explosive Limit for CO
LELH2: Lower Explosive Limit for H2
LELCH4: Lower Explosive Limit for CH4
UELCO: Upper Explosive Limit for CO
UELH2: Upper Explosive Limit for H2
UELCH4: Upper Explosive Limit for CH4
N1CO: Nose Limit for CO in air and excess nitrogen
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1H2</td>
<td>Nose Limit for H₂ in air and excess nitrogen</td>
</tr>
<tr>
<td>N1CH4</td>
<td>Nose Limit for CH₄ in air and excess nitrogen</td>
</tr>
<tr>
<td>N2CO</td>
<td>Nose Limit for CO in air and excess CO₂</td>
</tr>
<tr>
<td>N2H2</td>
<td>Nose Limit for H₂ in air and excess CO₂</td>
</tr>
<tr>
<td>N2CH4</td>
<td>Nose Limit for CH₄ in air and excess CO₂</td>
</tr>
<tr>
<td>V1CO</td>
<td>Volume of N₂ required to render unit volume of CO extinctive</td>
</tr>
<tr>
<td>V1CH4</td>
<td>Volume of N₂ required to render unit volume of CH₄ extinctive</td>
</tr>
<tr>
<td>V1H2</td>
<td>Volume of N₂ required to render unit volume of H₂ extinctive</td>
</tr>
<tr>
<td>V2CO</td>
<td>Volume of CO₂ required to render unit volume of CO extinctive</td>
</tr>
<tr>
<td>V2H2</td>
<td>Volume of CO₂ required to render unit volume of H₂ extinctive</td>
</tr>
<tr>
<td>V2CH4</td>
<td>Volume of CO₂ required to render unit volume of CH₄ extinctive</td>
</tr>
</tbody>
</table>

4.1.2 The Lower Explosive Limits

Mixture of flammable gases with air exhibit a concentration range over which the mixture, upon ignition, will propagate a flame independently of an external heat source.

If the concentration of flammable gas is below this range, then the mixture is too lean to sustain combustion. At concentrations above this range insufficient oxygen is present to allow propagation of ignition. The concentration of flammable gas corresponding to the lowest and highest
points of this range are termed the lower and upper explosive limits (LEL, UEL). Table 4.1 shows the accepted values of the lower and upper explosive limits of methane, carbon monoxide and hydrogen in air under ambient conditions. These ranges apply only to mixtures of the gas with air. For mixtures of flammable gas with an atmosphere deficient in oxygen, the explosive range is narrowed.

<table>
<thead>
<tr>
<th>GAS</th>
<th>LEL</th>
<th>UEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH₄</td>
<td>5.0</td>
<td>15.0</td>
</tr>
<tr>
<td>CO</td>
<td>12.5</td>
<td>74.0</td>
</tr>
<tr>
<td>H₂</td>
<td>4.0</td>
<td>75.0</td>
</tr>
</tbody>
</table>

In general, the lower explosive limit for a particular gas is raised only slightly but the upper limit decreases markedly with increase in oxygen deficiency of the mixture. The situation is further complicated if a mixture of flammable gas is considered, the components of which have widely varying explosive ranges. Such mixtures may be encountered in coal mines when, as a result of an explosion, heating or mine fires, carbon monoxide, hydrogen, and higher hydrocarbons are present in addition to methane.

The nose point is the flammable gas/oxygen concentration on a Coward diagram where the upper and lower flammable
limits coincide and hence explosions cannot occur.

Experiments have shown that this oxygen value decreases linearly as the temperature increases. For methane/air mixtures, the decreasing rate is approximately 1% by volume per 100 degrees Celsius rise in temperature. For a mixture of explosive gases, the nose points for the constituent gases are combined using an established formula to yield a single nose point for the mixture.

4.2. Extinctive volumes of Nitrogen and Carbon Dioxide

The last set of constants supplied to the GASANAL program is the Nitrogen and Carbon Dioxide volumes required to render unit volume of each explosive gas in the mixture extinctive.

Commonly inert gas components are also present. These may be excess Nitrogen (from oxygen depletion of original air) or Carbon Dioxide (from combustion) or both. These have a dumping effect on the explosibility of other gases. Figures expressing this effect as the volume of inert gas required to render a unit volume of combustible gas extinctive are given in Table 4.2. It can be seen from Table 4.2 that Carbon Dioxide is a more effective extinctive agent than Nitrogen.
Table 4.2 Extinctive Volumes

<table>
<thead>
<tr>
<th>Combustible</th>
<th>Inert</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N₂</td>
</tr>
<tr>
<td>CH₄</td>
<td>6.0</td>
</tr>
<tr>
<td>CO</td>
<td>4.15</td>
</tr>
<tr>
<td>H₂</td>
<td>16.55</td>
</tr>
</tbody>
</table>

** Variable Class 3

The last class of variables used by the programs are the results of computation performed on the first two classes of variables, namely the user-supplied variables and the constants. The calculated parameters are given below:

* ΣC: This is the sum of the constituent explosive gases
* PCO: Concentration of CO as percent of total combustible gases.
* PH₂: Concentration of H₂ as percentage of total combustible gases.
* PCCH₄: Concentration of CH₄ as percentage of total combustible gases.
4.3 The FORTRAN Program Algorithm

The steps involved in the explosibility computation are outlined below. The following format is used:

STEP N:
CALCULATED: CA,CB,CC,.....
INPUT: IA,IB,IC,....... 
FORMULA: ..........

In this format the N represents the Nth step, that is the point in the sequence of calculation.

CA,CB,CC,..... represent the parameters to be calculated in step N

IA,IB,IC,..... represent the input variables required for the computation

FORMULA: is followed by the FORMULA used.

The equations follow the calculation procedure analogous to that of Ellicott and they include the extinctive capacity of Carbon Dioxide.
4.3.1 Computational Sequence

STEP 1

CALCULATED: $\Sigma C$

INPUT: $CH_4\%$, $CO_2\%$, $H_2\%$

FORMULA: $\Sigma C = CH_4 + CO_2 + H_2$

STEP 2:

CALCULATED: $PCO$, $PH_2$, $PCH_4$

INPUT: $\Sigma C$, $CH_4\%$, $H_2\%$, $CO\%$

FORMULA: $P(gas) = (gas)/\Sigma C$ where $CH_4$, $H_2$, $CH_4$ are substituted for $(gas)$ as appropriate. For example $PCO = \%CO/\Sigma C$

STEP 3

CALCULATED: LELMIX

INPUT: LELCO, LELH_2, LELCH_4, PCO, PCH_4, PH_2

100.0

FORMULA: $LELMIX = \frac{(P(gas)/LEL(gas))}{100.0}$

where the appropriate gas is substituted for $(gas)$
STEP 4

CALCULATED: UELMIX

INPUT: UELCO, UELH₂, UELCH₄, PCO, PCH₄, PH₂

FORMULA: \[
UEL_{MIX} = \frac{100.0}{(P_{gas}/UEL_{gas})}
\]

STEP 5

CALCULATED: N₂

INPUT: CH₄, CO, H₂, O₂, CO₂, 100.0

FORMULA: \[N₂ = (100.0 - (CH₄ + CO + H₂ + O₂ + CO₂))\]

STEP 6

CALCULATED: T

INPUT: N₂, CO₂

FORMULA: \[T = N₂ + CO₂ \text{ (sum of extinctive gases)}\]

STEP 7

CALCULATED: C_N \text{ (combustible content at Nose point)}

INPUT: 100.0, PCH₄, PH₂, N₁CO, N₁CH₄, N₁H₂, T, N₂, CO₂, N₂CH₄, N₂CO, N₂H₂
FORMULA

\[ C_N = \frac{N_2}{T} \left[ \frac{100.0}{\Sigma (P_{(gas)}/N_1{_(gas)})} \right] + \]

\[ \frac{100.0}{\Sigma (P_{(gas)}/N_2{_(gas)})} \]

\[ \frac{CO_2}{T} \]

STEP 8a

CALCULATED: \( X_L \)

INPUT:

\( V_1CH_4, V_1H_2, V_1CO, V_2CH_4, V_2H_2, V_2CO, N_1CO, N_1CH_4, N_1H_2, N_2CO, N_2CH_4, N_2H_2 \)

FORMULA

\[ X_L = \frac{N_2}{T} \left[ \frac{(V_1{_(gas)} P_{(gas)})}{\Sigma (P_{(gas)}/N_1{_(gas)})} \right] + \]

\[ \frac{CO_2}{T} \left[ \frac{100.0}{\Sigma (P_{(gas)}/N_2{_(gas)})} \right] \]

STEP 8b

CALCULATED: \( O_N \)

INPUT: \( 0.2093, 100.0, X_L \)

FORMULA: \( O_N = 0.2093(100.0 - X_L - C_N) \)

An abbreviated form of the Coward Diagram is shown in Figure 4.2. In this diagram point X represents a sample composition shown here for convenience as potentially explosive.
Figure 4.2 The Coward Diagram
STEP 9

1. CALCULATED: Co-ordinates of points B, C, E and X

INPUT: UELMIX, LELMIX, -0.209, 20.93%, O₂, ΣC

FORMULAE:

\[ B = (\text{LELMIX}, -0.209 \times \text{LELMIX} + 20.93) \]
\[ C = (\text{UELMIX}, -0.209 \times \text{UELMIX} + 20.93) \]
\[ E = \left[ \begin{array}{c}
-20.93 \times \text{CN} \\
\text{ON-20.93}
\end{array} \right], 0 \]

\[ X = (\Sigma C, O₂) \]

2. Co-ordinate transformation to move the origin to the Nose point.

\[(X, Y) \rightarrow (X-C_N, Y-O_N)\]

The transformed co-ordinates of points B, C, E, and X are obtained by subtracting \(C_N\) from the X-coordinates of these points and \(O_N\) from the Y-coordinates. Thus after the transformations have taken place, the origin of the Coward Diagram will move to the Nose point. This is obtained by subtracting \(C_N\) from 0 and \(O_N\) from 0. Thus \((0,0)\) moves to the point \((C_N,O_N)\). To keep the relative positions of these points, the same quantities are subtracted from points B, C, E. Thus the coordinates of point B moves from \((\text{LELMIX},(-0.209\times\text{LELMIX} + 20.93))\) to \((\text{LELMIX}-C_N, (-0.209 \times \text{LELMIX} + 20.93 - O_N))\). Similar computation is performed on the coordinates of C, E and X to obtain their transformed positions. These new coordinates are designated \(B_1, C_1, E_1, \text{ and } X_1\)
STEP 10

Conversion of new coordinates of points $B_1, C_1, E_1$, and $X_1$ to polar form with the origin at the Nose point.

**CALCULATED:** $r_b, \theta_b$

**INPUT:** $B_1$

Assuming the co-ordinate of point $B_1$ is $(X_b, Y_b)$, then

$$r_b = \sqrt{(X_b)^2 + (Y_b)^2} \quad \text{and} \quad \theta_b = \tan^{-1}(X_b, Y_b)$$

Thus $B_1$ is transformed to $r_b, \theta_b$. Points $C, E, X$ are transformed in similar fashion.

STEP 11

Calculation of modified angular co-ordinates $r_m, \theta_m$ for the sample point $X$. 
(a) Explosive Case: \(0_b \leq \theta_x > \theta_c\)

\[
\begin{align*}
    r_m &= r_x \\
    \theta_m &= \left[ \frac{\theta_x - \theta_c}{(\theta_x - \theta_c) + (\theta_b - \theta_x)} \right]
\end{align*}
\]

(b) Potentially Explosive Case: \(\theta_c > \theta_x\) or \(\theta_x > \theta_e\)

\[
\begin{align*}
    r_m &= r_x \\
    \theta_m &= \left\{ \left[ \frac{\theta_x - \theta_e}{(\theta_x - \theta_e) + (\theta_c - \theta_x)} \right] 0.90 \right\} + 270
\end{align*}
\]

(c) Non-Explosive Case

\[
\begin{align*}
    r_m &= r_x \\
    \theta_m &= \left\{ \left[ \frac{\theta_m - \theta_b}{(\theta_x - \theta_b) + (\theta_e - \theta_x)} \right] 0.180 \right\} + 90
\end{align*}
\]

Conversion of modified angular coordinates to regular (cartesian) coordinates for sample point:
The coordinates $(X_m, Y_m)$ then convey the explosive states of the gas mixture according to:

<table>
<thead>
<tr>
<th>STATE</th>
<th>$X_m$</th>
<th>$Y_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explosive</td>
<td>$+$</td>
<td>$+$</td>
</tr>
<tr>
<td>Potentially Explosive</td>
<td>$+$</td>
<td>$-$</td>
</tr>
<tr>
<td>Non Explosive</td>
<td>$-$</td>
<td>$+$</td>
</tr>
<tr>
<td>Explosive</td>
<td>$-$</td>
<td>$-$</td>
</tr>
</tbody>
</table>

These points are then plotted on the Ellicott Diagram for time trend observations.

4.4 Development of the Database Management Systems

The main function of the database programs is to accept data for storage and transfer them to the other programs when necessary. The idea here is to relieve the FORTRAN program, which does all the computations, the task of storing the data. The database programs also provides
the user-friendly interface between the user and the computer. Without this provision for data storage and retrieval, all the gas samples would be directly supplied to the FORTRAN by the user. This would be a severe limitation to a prolonged use of the system since in such situations a large number of data manipulation is likely to be encountered. Additionally, the low level language characteristics of DBASE III+ makes the development of user-friendly programs less complicated.

Altogether, there are ten command files for the database management system. The names of these files and a brief description of their functions are given below:

4.4.1 ADDGAS.PRG:

The source code for this file is given as Appendix B. ADDGAS.PRG, a procedure file, allows a database (.DBF) file to accept and store input of data from the user and then transforms this data to a "Text" file for the FORTRAN to use as its input. This program "USES" the GASAMPLE.DBF and KONTER.DBF database files. The custom format file, which is a customized display on the monitor for input, is the GASFMT.FMT. The structures for GASAMPLE.DBF and KONTER.DBF database files are shown below as tables 4.3(a) and 4.3(b).
### Table 4.3(a) Structure of GASAMPLE.DBF Database File

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Width</th>
<th>Dec</th>
<th>Field Name</th>
<th>Type</th>
<th>Width</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>METHANE</td>
<td>Numeric</td>
<td>5</td>
<td>2</td>
<td>HUMIDITY</td>
<td>Numeric</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>CARBON_MON</td>
<td>Numeric</td>
<td>6</td>
<td>3</td>
<td>NITROGEN</td>
<td>Numeric</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>HYDROGEN</td>
<td>Numeric</td>
<td>5</td>
<td>2</td>
<td>ATM_PRESS</td>
<td>Numeric</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>OXYGEN</td>
<td>Numeric</td>
<td>5</td>
<td>2</td>
<td>SAMPLER</td>
<td>Character</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>CARBON_DOX</td>
<td>Numeric</td>
<td>5</td>
<td>2</td>
<td>AIR_VELOCI</td>
<td>Numeric</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>TEMP</td>
<td>Numeric</td>
<td>6</td>
<td>1</td>
<td>TYPE</td>
<td>Character</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>TIME</td>
<td>Numeric</td>
<td>5</td>
<td>2</td>
<td>COMMENT</td>
<td>Character</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>DATE</td>
<td>Date</td>
<td>8</td>
<td></td>
<td>COMMENT1</td>
<td>Character</td>
<td>35</td>
<td></td>
</tr>
</tbody>
</table>

Enter the field name.

Field names begin with a letter and may contain letters, digits and underscores.
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Width</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 KONT</td>
<td>Numeric</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>2 LOCATION</td>
<td>Character</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>3 HEIGHT</td>
<td>Numeric</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>4 WIDTH</td>
<td>Numeric</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>5 DIAMETER</td>
<td>Numeric</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>6 COLIERY</td>
<td>Character</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>7 PANEL</td>
<td>Character</td>
<td>15</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4.3(b) Structure of KONTER.DBF Database File
One of the fields of KONTER.DBF is called KONT. This numeric field accepts a number from the user and this number represents the number of sample sets available for input. Thus essentially the number stored as a variable for the KONT field is used as a counter for the number of times a blank screen format is displayed for input. Each time a blank format screen is displayed the KONT field variable is reduced by one until it reaches zero and this will signify the end of data input.

When data input is finished, a temporary database file, SAMPLE.DBF is created and values for a selected fields for GASAMPLE.DBF are copied into the temporary database file. SAMPLE.DBF file is not a permanent file. It is created on the run and erased after use. All the fields and their values for GASAMPLE.DBF are copied to SAMPLE.DBF except the COMMENT field, the TYPE (of sample) field and the SAMPLER field. Without the creation of a temporary file like SAMPLE.DBF, it will be necessary to convert all the field values in the source file, namely GASAMPLE.DBF to a "text" file and this will include all the variables which are not needed for computation, namely the TYPE, SAMPLER and the COMMENT fields. This will be uneconomical in terms of computing memory and time consumed in creating "text" files for these variables every time the program is run. The temporary file allows only the necessary variables to be transformed as "text" file for the FORTRAN. The temporary file, SAMPLE.DBF is deleted after its use. The relationship between these files is
4.4.2 KUNTED.PRG

This command file is the very first program that is used even before the ADDGAS.PRG discussed above. KUNTED.PRG command file also uses the KONTED.DBF as its database file. The custom format it uses is the KONTFMT.FMT. This command file allows all the variables for the KONTER.DBF fields to be transformed to a "text" file called DBASE.TXT. The listing of KUNTED.PRG program is provided in Appendix C.

4.4.3 FORGRAPH.PRG

This program allows the FORTRAN program to append the results of its computation to a database file. These results includes the X, Y co-ordinates for the Ellicott Diagram
and all the gas ratios. A number of text files have been created for different graphs and the relevant information for a particular graph are extracted automatically.

Control is transferred automatically to this program immediately after the execution of the FORTRAN program. The design philosophy behind the FORGRAPH.PRG command file is that in a fire situation different graphs are plotted for investigation, each requiring a unique set of input data. In addition to the variety of the graphs, different panels must be specified. Within a specific panel a graph can only be plotted for a specific location. Figure 4.4 limns this idea.

One of the database files (Forgraph.dbf), however, contains the names of all the panels supplied to it and their locations within the panels. In plotting a graph the user specifies the name of the panel. This will cause all the locations for the selected panel to be displayed. These variables (that is, the selected panel and location) are stored in a database file with same field names, that is PANEL and LOCATION. In other words, the value for the PANEL field will be the name of the selected panel and the LOCATION field will contain the selected location. Using these variables as a key, the source database file is opened. All records with PANEL and LOCATION fields matching the key will have the values of their fields copied to the target database file. Non common fields will
### Colliery "Z"

#### PANEL "A"

<table>
<thead>
<tr>
<th>0</th>
<th>04</th>
</tr>
</thead>
<tbody>
<tr>
<td>05</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

#### PANEL "B"

<table>
<thead>
<tr>
<th>1</th>
<th>05</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

#### PANEL "C"

<table>
<thead>
<tr>
<th>01</th>
<th>06</th>
<th>07</th>
<th>08</th>
</tr>
</thead>
<tbody>
<tr>
<td>02</td>
<td>03</td>
<td>04</td>
<td>05</td>
</tr>
</tbody>
</table>

#### PANEL "D"

<table>
<thead>
<tr>
<th>01</th>
<th>04</th>
<th>05</th>
<th>08</th>
</tr>
</thead>
<tbody>
<tr>
<td>02</td>
<td>03</td>
<td>07</td>
<td>09</td>
</tr>
</tbody>
</table>

* Sample location

---

**Figure 4.4 Hypothetical Arrangement of Sample Points**
not be affected. In other words the two database files should have common fields before transfer of data can be effected. The retrieval of data pertaining to the graph will leave the user with the last task: specifying the type of graph.

Specification of the graph is accomplished in the VP-Expert Shell. In a menu form the user is prompted to make a selection for the graph type. For example, the user may choose to plot a "Gas content vs. Time" graph for "Oxygen". These variables ("Gas-content-Time" and "Oxygen") are also appended to a database file called GTYPE.DBF. The variables are then made into a "text" file for the graph-plotting program to read. They are then used as conditions for the program to read the appropriate data for the graph. Appendix D shows the FORGRAPH.PRG program listing. A summary of the sequence described above is given below:

4.4.4 FORTIMPT.PRG

This command file displayed as Appendix E allows the results from the FORTRAN to be appended to different database files. These results are grouped into four for convenience, and each group of data is destined for different database files for storage and use.

4.4.5 CHOSELIS.PRG

This program lists the different groups of results to
print or display on the screen depending on the user's choice. The choice is made from a menu. This program is a simple manifestation of DBASE III's power of interactive programming. CHOSELIS.prg listing is shown as Appendix F.

4.4.6 LIST1...4.PRG

Depending on the user's choice one of these four command files is executed after CHOSELIS.PRG. Each of the programs executes a different REPORT format. User's are given the option to indicate a choice of hard copy and the number of copies. Otherwise the results are displayed on the screen. LIST1.prg is shown as Appendix G.

4.4.7 ZAPALL.PRG

ZAPALL.PRG command file is the last database program. Its function is to "ZAP", that is, empty all the database files of their contents. In addition all system created files, including "text" files are removed permanently. The choice to effect the execution of this program is found in the main menu. To avoid any inadvertent erasure of files, the user is always asked to confirm or cancel this choice. The listing of this command file is shown as Appendix H.
4.5 The "BASIC" programs for graphics display

A graphical representation of the changing conditions, with respect to time, of the vitiated atmosphere resulting from underground fire outbreak is invaluable for timely and accurate interpretation of the fire's responses to fire fighting techniques. So, in addition to the tabular display of results, a number of graphs have been plotted for this purpose. Altogether seven different graphs are plotted. These graphs are listed below.

Only one specific gas/time graph can be plotted at a time. For example $O_2$/time graph cannot be plotted simultaneously with any other graph.

Other graphs plotted by this BASIC programs are:

2. Airfree vs. Time
3. Temperature vs. Time
4. Pressure vs. Time
5. The Coward Diagram
6. The Ellicott Diagram

4.5.1 Functions of the "BASIC" Programs

A representative code for these programs is shown as Appendix I(a). The function of each one is given below:
1. GASVAL.BAS

   This program plots graphs of the gases O₂, N₂, CO₂, H₂, CH₄, vs. Time.

2. AIRFREE.BAS

   Plots Air Free values vs. Time for each of the gases.

3. TEMPRES.BAS

   Plots graphs of Temperature vs. Time.

4. COWD.BAS

   Plots the Coward Diagram

5. ELLICO.BAS

   Plots the Ellicott Diagram.

6. RATIOS.BAS

   Plots the Trickett's and Graham's Ratios.

The following flowchart also shows the relationship between the user and system functions.
1. Selects type of graph from menu (e.g. "gas content vs. time")

Appends selected graph to a database file. Makes a TEXT file for BASIC to read.

2. Selects gas to plot for (e.g. "Oxygen")

3. Selects panel from menu (e.g. "North Panel")

Appends selected panel to a target database file.

4. Selects location of sample (e.g. "A4")

Opens source database file and selects matching records

Appends selected items (from source database file) to target database file and makes "TEXT" file.

Control goes back to BASIC program to plot graph using as inputs the resulting TEXT file created above.

4.5.2 Design Philosophy

The graph-plotting programs will not be discussed individually, instead an overview of their design philosophy will be presented in this section.

The most important input to the programs are gas concentrations, the X- and Y-coordinates for the Ellicott 92
Diagram, time, date and sample identification parameters (panel and location). In all the programs, the user is prompted to supply the date, which is used to identify the plotting date and may or may not be the same as the sampling date.

In developing the program the important consideration was the time trend of the graphs: the temperature of the fire at 2pm is not as important as the trend, that is, the behaviour of the temperature with time.

With these facts in mind, the first sample time was taken as time zero, and all other times are measured relative to the first. Thus if the first time was at 3pm and the second 4pm, 3pm was taken as time zero. The interval between samples one and two is therefore one hour. A third sample taken on 5pm the same day will be 2 hours from the start of the fire. Thus to get the time interval between the first and any other subsequent sample, the first sample time was deducted from the other time.

The most complicated graphs are those involving time. Since fires can occur at any time of the day, of the month, and of the year, all possibilities must be considered.

Complications begin to appear with the time trend when the life of a fire extends into days and samples have to be taken periodically. To make this point clearer, the time interval between samples taken on day
one at 11.55pm and another taken on day two at 12.05am is NOT 1 day (24 hours) but only 10 minutes. The complication is compounded if the two samples fall into different years. For example, the time interval between a sample taken on day one at 11.05pm on 31/12/87 and another taken on day two at 12.05am on 01/01/88 is NOT 1 year.

4.5.2.1 Converting time intervals to hours

To overcome this problem, the time intervals between the first sample and all other samples were converted to hours. The following procedure were followed:

Two different dates representing the worse case possible will be used as illustration.

1. Assume sample one was taken at 11:30pm on 31/12/87 and sample two at 00:10am on 01/01/88. A simple calculation reveals that the elapse time between the two samples is 40 minutes. It is not always enough to consider only the times; for example if the second sample was taken at 12:10am on 03/01/88 (instead of 01/01/88), subtracting the first time from the second will not give the correct time interval between the two samples. Thus the sample dates must also be considered.

2. Split the sample date into day, month and year and treat them as ordinary numbers. Thus 31/12/87 and 01/01/88 become 31, 12, 87 and 1, 1, 88 respectively.
3. Subtract the year part of the first date from the year part of the other.

Thus $88 - 87 = 1$ year ------------------------ (4.1)

4. Convert the answer from (1) to months by multiplying by 12.

Thus $1 \times 12 = 12$ months ------------------------ (4.2)

5. Subtract the month part of the first date from the month part of the other.

Thus $1 - 12 = -11$ months ------------------------ (4.3)

6. Add (2) and (3) to give the total number of months

Thus $12 + (-11) = 1$ month ------------------------ (4.4)

7. Convert (4) to days

Thus $1 \times 31 = 31$ days ------------------------ (4.5)

8. Subtract the day part of the first date from the day part of the other date

Thus $1 - 31 = -30$ days ------------------------ (4.6)
9. Add (5) and (6) to get the total number of days
   Thus \(31 + (-30) = 1\) day ------------------------- (4.7)

10 Convert (7) to hours
   Thus \(1 \times 24 = 24\) hours ----------------------- (4.8)

11 Subtract the first time from the second time.
   Thus 00:10am - 11:30pm = -11.2 ------------------ (4.9)
   Therefore the elapse time = (4.8) + (4.9) = 12.8 hours
CHAPTER FIVE

System validation

In this chapter the system validation is presented which includes the operational and computing (hardware) requirement of the system, as well as a sample consultation.

5.1. Operational and computing requirement for "RESCUE"

The system has been developed to handle fire emergency situations in which gas sample results and knowledge of other environmental conditions are required as basis for decision making. The data needed for consultation is typical of what mine rescue personnel would have readily available such as gas contents, temperature, date and time of sampling, pressure, geological information (e.g. coal rank, heat propagation rates, ventilation requirement etc.). Also if the user lacks certain data "RESCUE" equates the value of the variable to "UNKNOWN" and the consultation proceeds.

To make the accessibility of the program feasible for the mining industry, it was deemed appropriate that the system be developed for use on personal computers.

The development tool selected ("shell") serves as an advanced development environment and delivery vehicle for production rule base. This software has the capability to activate other executable programs.
(written in other languages) during execution of a knowledge base. For example, it is able to activate a compiled (using CLIPPER) dBASE III+ so that the user can manipulate data in the database environment (such as putting sample data in the database) and when ready returns the user to the expert system environment for continuation of the consultation. This capability also allows the expert system to "CALL" the FORTRAN program which utilizes the data stored in the database file to calculate for the explosibility status of the atmosphere and other parameters required for decision making. These results are passed back to the expert system for its inferences and conclusion.

The recommended hardware configuration is a minimum of 640K bytes of RAM and a minimum of 2M bytes of hard disk. This configuration is to ensure that sufficient memory will be left to run externally activated programs.

Knowledge of "RESCUE" is embodied within facts specified by production rules and a database of information (gas contents, airway dimensions, air velocities) available for access by the system while it is conducting a consultation session.

5.1.2 Using "RESCUE"

Fig 4.1 is the structure of operation of RESCUE. The system begins in the expert system environment and the fol-
The following menu is presented to the user:

1) Input/add gas sample data
2) Delete data
3) Plot/display graph
4) Display input/output (results)
5) Consult expert system
6) Quit

If the system is being used for the first time the only logical choice is option 1 since the system by this time has not processed any data for the other options to be meaningful. When option 1 is selected, the user is immediately taken into the database environment to put in his gas sampling data. These input are transformed into system data format (SDF) files. A FORTRAN program is then "called" which uses the created TEXT file as its input for analysis of the explosibility status of the atmosphere. When this stage is completed, control is transferred back to the expert system which then activates a database program and "APPENDS" the gas analysis results (from the FORTRAN) to a database file for storage and subsequent utilization during the expert system consultation process. The user is then taken back to the main menu where the process described above can be repeated (ie input more sample data.) or any of the other options may be selected.
In addition to plotting graphs, the system stores all the input and output results and can present them to the user when required. Due to the large number of input and output involved, the results are grouped into four.

GROUP 1 Contains all gas input, X and Y coordinates and Trickett's ratio.

GROUP 2 Contains all the input and Air Free values.

GROUP 3 Contains all gas input, CO/min., and all gas ratios.

GROUP 4 Contains all input including pressure, humidity and temperature.

Table 5.1 is a typical computer run of group 2 above.

Finally option 5 (from main menu) takes the user into the expert system consultation mode.

5.2 Consultation
Option 5 in the main menu takes the user into the consultation mode. The main goal of the system is to assist decision makers as to the course of action to take following the outbreak of fire. In view of the complexity and the multiple number of decisions that can be reached in such a situation, it was decided that the backward chaining, depth-first search method of
<table>
<thead>
<tr>
<th>DATE</th>
<th>TIME</th>
<th>O2</th>
<th>H2</th>
<th>CH4</th>
<th>CO</th>
<th>CO2</th>
<th>AFU</th>
<th>AFU</th>
<th>AFU</th>
<th>AFU</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(gmt)</td>
<td>(%)</td>
<td>(%)</td>
<td>(%)</td>
<td>(%)</td>
<td>(%)</td>
<td>(H2)</td>
<td>(CH4)</td>
<td>(CO)</td>
<td>(N2)</td>
</tr>
<tr>
<td>02/01/88</td>
<td>0.30</td>
<td>13.46</td>
<td>0.83</td>
<td>3.10</td>
<td>0.830</td>
<td>2.62</td>
<td>2.32</td>
<td>8.68</td>
<td>2.320</td>
<td>79.34</td>
</tr>
<tr>
<td>02/01/88</td>
<td>0.45</td>
<td>17.00</td>
<td>2.20</td>
<td>1.20</td>
<td>0.600</td>
<td>3.40</td>
<td>11.70</td>
<td>6.38</td>
<td>3.190</td>
<td>60.66</td>
</tr>
<tr>
<td>02/01/88</td>
<td>1.45</td>
<td>14.00</td>
<td>1.20</td>
<td>1.00</td>
<td>1.200</td>
<td>3.20</td>
<td>3.62</td>
<td>3.02</td>
<td>3.620</td>
<td>88.08</td>
</tr>
<tr>
<td>02/01/88</td>
<td>2.50</td>
<td>2.80</td>
<td>4.90</td>
<td>3.30</td>
<td>4.300</td>
<td>13.80</td>
<td>5.66</td>
<td>3.81</td>
<td>4.960</td>
<td>69.64</td>
</tr>
</tbody>
</table>

**Panel**: NO2-East-Panel

* Location: A3

<table>
<thead>
<tr>
<th>DATE</th>
<th>TIME</th>
<th>O2</th>
<th>H2</th>
<th>CH4</th>
<th>CO</th>
<th>CO2</th>
<th>AFU</th>
<th>AFU</th>
<th>AFU</th>
<th>AFU</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(gmt)</td>
<td>(%)</td>
<td>(%)</td>
<td>(%)</td>
<td>(%)</td>
<td>(%)</td>
<td>(H2)</td>
<td>(CH4)</td>
<td>(CO)</td>
<td>(N2)</td>
</tr>
<tr>
<td>08/03/88</td>
<td>16.40</td>
<td>13.46</td>
<td>0.83</td>
<td>3.10</td>
<td>0.830</td>
<td>0.00</td>
<td>2.32</td>
<td>8.68</td>
<td>2.320</td>
<td>86.67</td>
</tr>
<tr>
<td>08/03/88</td>
<td>17.00</td>
<td>16.00</td>
<td>1.00</td>
<td>2.00</td>
<td>1.000</td>
<td>3.00</td>
<td>4.24</td>
<td>8.48</td>
<td>4.240</td>
<td>70.32</td>
</tr>
</tbody>
</table>

**Panel**: North-Z-Section

* Location: A2

Table 5-1. Gas input and partial results
knowledge representation would be most appropriate for the inference engine. The following are examples of the type of rules RESCUE will utilize albeit the syntax of representation in the shell is not strictly followed here.

RULE 5

IF fire is too intense for fire fighting OR
fire is expanding more quickly than fire fighting can cope OR
fire is inaccessible for fire fighting OR
fire location is unsafe for fire fighting AND
direct air access to the fire bed is limited AND
gas ignition risk is too low

THEN apply a high expansion foam plug using generator in the opening for 24 hours, monitor the fire gas products for lower oxygen and increased flammable gas content, and control air such that flammable gas in main return does not exceed 2%.

RULE 6

IF gas analysis is as follows:

\[ \text{CO}_2\% \leq 5.00 \]
\[ \text{CH}_4\% \leq 0.50 + \text{normal emission} \]
\[ \text{CO}\% \leq 0.50 \]
H₂% ≤ 0.50
O₂% ≤ 10.00
N₂% is balance to 100%

THEN The fire atmosphere is confirmed as oxygen rich.
Fire fighting can proceed.

Assuming rule 5 is fired, the inference engine will try to establish all the sub-goals that lead to the conclusion of the fired rule (in this case rule 5). The engine will look for rules that concern a sub-goal. If there are no rules to validate or invalidate a particular sub-goal, the user will be requested to supply a value for the sub-goal. Specifically in rule 5, the system will try to establish the intensity of the fire. Since there are no rules to establish this, the user will supply a value for the intensity of the fire and a question such as "What is your estimate of the intensity of the fire?" could be asked. To help the user with possible answers expected by the system, a list of possible answers will be supplied along with the question. Again to establish the sub goal "gas ignition risk too low", a rule such as the following will need to be provided:

RULE 5A

IF flammable composition ≤ 1.5%
THEN gas ignition risk is low
Now the system should establish the value for this sub-goal (i.e. value for flammable composition). At this point the system will go into the database (where results of gas analysis are stored) and will look for out the value of the combustible.

5.3 Confidence factors

In real life, very little is known with absolute certainty. Even "experts" frequently resort to likely conclusions and best guesses when handing out advices.

RESCUE makes room for uncertain responses by the user by allowing confidence factors. A confidence factor is an integer (between 0 and 100) indicating the degree of certainty that a particular conclusion is valid. The system allows the user to enter confidence factors when responding to a prompt. As an example consider the following rule:

\[
\text{IF smoke is dark AND smoke is dense AND no ventilation control over fire zone is possible AND smoke back-up is observed AND fire is intense AND fire is well established}
\]
THEN there is evident of fuel-rich fire
tending to recirculate  CNF 100.

This rule says that if the antecedent are all true
then the conclusion will be true with 100% confidence. However, if the user is not absolutely sure of his answers
to the prompts, then the confidence factor of the conclusion reduces correspondingly. For example if
responses to the above rule were to be:

smoke is dark  CNF 90
smoke is dense  CNF 100
fire is well established  CNF 100
fire is intense  CNF 100
smoke back-up is observed  CNF 80

These confidence factors from the user will be combined in a formula \((0.9 \times 1 \times 1 \times 1 \times 0.8 = 0.72)\) and this will be the new confidence factor for the conclusion of the rule.
CHAPTER SIX

SUMMARY AND CONCLUSIONS

6.1 SUMMARY

This thesis is a software development aimed to be used by mine emergency personnel both as a training tool and as an aid to fire fighting decision-making process. The software incorporates both conventional programs and artificial intelligence (expert system) algorithm which contains knowledge extracted from mine emergency personnel.

In developing the package some factors were taken into consideration:

a) Personal computers are now reasonably powerful in both speed and memory capacity. Since personal computers are much cheaper than Mainframe, the system was developed exclusively for use on personal computers.

b) Prerequisite for a successful fire fighting and control is a combination of many skills. These skills are only acquired after a long exposure to emergency situations. In addition gas sampling results and analysis is invaluable for determining the trend of the fire. In the absence of an experience personnel, an expert sys-
tem becomes a substitute and in the presence of such a person the system can be used as a coadjutor in making decision.

6.2 Conclusions

This thesis addresses a practical problem associated with the coal mining industry: fire outbreak due to dust explosions, short-firing, coal outbursts and other causes have plagued the industry for a long time.

During such emergencies lives and equipment are at risk if proper remedial procedures are not adhered to. The adherence to these life-saving procedures calls for the availability of experts as well as the proper input. Such input consists of both human and material resources. One of the material input, may be in the form of a computer program that may assess the changing conditions of the vitiated atmosphere in order to determine the atmosphere's fitness to accommodate rescuers.

Hitherto, computer programs have been written to compute the explosibility status of the atmosphere surrounding a fire, the results of which are used to plan rescue strategies.

These programs do work satisfactorily. As the life of a fire is prolonged, and more and more samples are analysed, these programs reach their limit due to the absence
This problem has been addressed in this thesis by incorporating a database management system with these programs. With this enhancement, the amount of data that can be handled by the package is practically unlimited. Thus, this system will not be overburdened with data no matter how long a fire fighting process may take.

In addition, an expert system has been attached to assist personnel in making decisions.

The computer programs for this thesis were developed in three different languages:

1. FORTRAN
2. BASIC
3. dBASE III+

The expert system was developed in the VP-Expert Shell. Each of these languages has some features that others lack. For example, dBASE III+ is a database management development tool. On the other hand, all the graph-plotting programs were written in the BASIC language since BASIC is particularly suited for graphics on the personal computer. The rest of the programs were written in FORTRAN.

6.3 Running the programs

In running the system, the user starts in the expert
system environment, and a menu is displayed for selection. When sampling results are supplied to the system (via the database program), the data is analysed and the results automatically appended to a database file. The user is then returned to the main menu. On return to the main menu each of the menu choices becomes available for selection.

6.3.1 Graph Plotting

Graphs are plotted automatically upon selecting the choice and as soon as the specification of the type of graph is transmitted to the computer. Graphs are not saved or written to disk so the only way to preserve a graph is to print a hard copy. Also graphs are plotted one at a time. These graphs are important in that they provide a visual display of the changing atmospheric conditions.

6.3.2 The Expert System

Knowledge in the expert system were extracted from experts at the Southern Mine Rescue Station of Bulli, New South Wales, Australia. The knowledge were encoded in the VP-Expert shell. When the choice to run the expert system is executed, the computer begins a consultation with the user until a solution is reached or the user decides to quit.

The shell takes care of the inference control, the
natural language interface and the confidence factors. The production rule strategy of knowledge representation was employed. This comprises a set of rules of the form IF....THEN statements. In this representation the conclusion or the THEN substatement is considered valid if the antecedent or the IF substatement can be proved to be valid. The validation of the antecedent is done by the user answering a question to this effect or can be done through the system's working memory.

6.4 Suggestions for further work

The programs developed for this thesis is intended to be used as a training tool for mine rescue personnels and for analysis of mine gas sampling during mine emergencies. In developing the expert system many important factors were not considered for lack of time. In mine emergencies, in addition to sampling results that are taken into account, many other important factors must also be considered. These include the ventilation network and the layout or plans of the colliery. The ventilation network, showing air distribution, is necessary for controlling the flow of air to specific sections of the mine. The mine plan is essential for rescue strategies.

Any further work along the lines of this research, should incorporate these important factors for completeness. In particular the mine plans should be incor-
porated in the expert system.

Dynamic simulation of the mine plans as well as the ventilation distribution could be displayed on the screen. Thus locations for rescue activities could be shown on the screen. In this case changes could be done in the simulated mine and its probable effect in the real world may be predicted.
REFERENCES

Bundy, A., Bryd, L. Expert System In The Micro-Electronic Age (Edited by Michie D.) (pp 50-64)

P.H. Winston, Artificial Intelligence (Addison Wesley) (pp 87-155)

S. Alim and J. Munro, Prolog-based Expert System in Civil Engineering (pp 182-197)

D. Michie, Introductory Readings In Expert Systems (pp 15-34, 45-67)


Y. Shirai and J. Tsujii Artificial Intelligence: Concepts Techniques, and Applications (pp 232-367)

D.A. Waterman Guide to Expert Systems

P.K. Chatterjee A Critical Review of Coal Mine Disasters in Queensland and New South Wales Since 1920 (University of Queensland, Department of Mining and Metallurgy, (pp3-22)


C.W. Ellicott Assessment of the Explosibility of Gas Mixtures and Monitoring of Sample-Time
P. Harmon and D. King

Artificial Intelligence in Business (pp 64-77)

Artificial Intelligence (American Association for Artificial Intelligence):

Volume 30 (1986) (pp 77-89)
Volume 20 (1983) (pp 215-251)
Volume 15 (1980) (pp 179-222)

dBASE III+ Handbook

VP-Expert Manual

QUICKBASIC Manual
APPENDIX A: GASANAL.FOR

This is a Fortran program that analyses the original gas input.
PROGRAM gasanal ASSESS THE EXPLOSIBILITY OF GAS MIXTURES
THE MONITORING OF SAMPLE-TIME TREND. THE METHOD OF
COMPUTATION IS BASED ON ELLICOTT EXTENSION OF COWARD'S
ASSESSMENT PUBLISHED ON PAGE 20-1 OF - IGNITION, EXPLOSIONS
& FIRES, EDITED BY A J HARGRAVES, AusIMM (ILLAWARRA
BRANCH), 1981.

INTEGER DATE
CHARACTER * 10 LOCAT,FILIN,FILEIN,FILEOT,
1 FLOUT,FILOUT,FOUT
CHARACTER * 15 PANEL,COL

COMMON/OTP/LOCAT,DATE,TIME
COMMON/GAS1/COPMIN,Q,XCORD,YCORD
COMMON/GAS2/CH4,H2,CO,CO2,O2,GN2,SIGMAC

DATA IPUT,IOUUT,IIOUT,IIIOUT,IOUT/10,11,12,13,14,15/
DATA FILIN,FILEIN,FILEOT/'BBASE.TXT', 'BBASE3.TXT',
1 'GASOT.TXT'/
DATA FLOUT,FILOUT,FOUT/'GASOUT.TXT', 'GSOUT.TXT', 'GOUT.TXT' /
DATA XLCH4,XLH2,XLCH4/12.5,4.0,5.0/
DATA UPCH4,UPCH4,UPH2/74.0,15.0,75.0/
DATA XI1CO,XI1CH4,X1H2/13.78,5.93,4.30/
DATA X2CO,X2CH4,X2H2/18.0,6.66,5.73/
DATA DELTA,PI/0.001,3.1415927/
DATA CONS1,CONS2,CONS3/4.776,20.93,0.209/
DATA V1CH4,V1CO,V1H2/6.0,4.15,16.55/
DATA V2CH4,V2CO,V2H2/3.2,2.16,10.2/

DEFINITION OF VARIABLES

1. USER INPUTS IN %.

CH4 = METHANE (GAS)
CO = CARBON MONOXIDE (GAS)
H2 = HYDROGEN (GAS)
CO2 = CARBON DIOXIDE (GAS)
GN2 = NITROGEN (GAS)
O2 = OXYGEN (GAS)

2. CONSTANTS

XLJ = LOWER EXPL. LIMIT FOR GAS,J.
UPJ = UPPER EXPL. LIMIT FOR GAS,J.
X1J = NOSE LIMIT FOR GAS,J, IN
AIR AND EXCESS NITROGEN
X2J = NOSE LIMIT FOR GAS,J, IN
AIR AND EXCESS CO2.
V1J = VOLUME OF N2 REQUIRED TO
RENDER UNIT VOLUME OF GAS,J,
EXTINCTIVE.
V2J = VOLUME OF CO2 REQUIRED TO RENDER
UNIT VOLUME OF GAS,J, EXTINCTIVE.

3. CALCULATED.

SIGMAC = TOTAL COMBUSTIBLES
PJ = CONCENTRATION OF COMBUSTIBLE
GAS,J, AS A PERCENTAGE OF
TOTAL COMBUSTIBLES
XLMIX = LOWER EXPL. LIMIT OF MIXTURE
UPMIX = UPPER EXPL. LIMIT OF MIXTURE
CN = COMBUSTIBLE CONTENT AT THE NOSE POINT FOR THE MIXTURE
ON = OXYGEN CONTENT AT THE NOSE POINT FOR THE MIXTURE

OPEN COUNTER FILE
OPEN(UNIT=IINPUT,FILE=FILIN,STATUS='OLD')
READ(IINPUT,10) NN,COL,PANEL,LOCAT,WIDTH,HGT,DIA
FORMAT(13,2A15,A4,3F5.2)

! CALCULATE X-SECT. AREA OF DRIVE !

IF(DIA .LT. DELTA) THEN
AREA = HGT*WIDTH
ELSE
AREA = PI*(DIA*DIA)/4.0
ENDIF

OPEN BASIC FILES
OPEN(UNIT=IINPUT,FILE=FILIN,STATUS='OLD')
OPEN(UNIT=IOUTPUT,FILE=FILEOUT,STATUS='NEW')
OPEN(UNIT=IIOUTPUT,FILE=FLOUT,STATUS='NEW')
OPEN(UNIT=IIOUTPUT,FILE=FIOUTPUT,STATUS='NEW')
OPEN(UNIT=IIOUTPUT,FILE=FOUT,STATUS='NEW')

READ GAS DATA VALUES FROM filein
DO 100 1=1,NN
READ(IINPUT,20) H2,CH4,CO,CO2,GN2,
1HUM,TEMP,PRESS,TIME,VEL,DATE
FORMAT(3F5.2,F6.3,2F5.2,3F6.2,2F5.2,18)

STEP1: CALCULATE TOTAL COMBUSTIBLES
INPUTS: CH4,H2,CO
CALL TCOMB(PCO,PCH4,PH2)

STEP2: CALCULATE LOWER EXPLOSIVE LIMIT OF MIXTURE
INPUTS: LELCO,LELH2,LELCH4,PCO,PCH4,PH2
XLMIX = 100.0/((PCO/XLCO) + (PCH4/XLCH4)
1+ (PH2/XLH2))

STEP3: CALCULATE UPPER EXPLOSIVE LIMIT OF MIXTURE
INPUTS: UPCO,UPH2,UPCH4,PCO,PCH4,PH2
UPMIX = 100.0/((PCO/UPCO) + (PCH4/UPCH4)
1+ (PH2/UPH2))

STEP4: CALCULATE NITROGEN CONTENT IF UNKNOWN
INPUTS: CH4,CO,H2,O2,CO2
IF(GN2 .LT. DELTA) GN2 = 100.0-(CH4+CO+H2+O2+CO2)

STEP5: CALCULATE TOTAL N2 AND CO2
INPUTS: GN2, CO2

T = GN2 + CO2

STEP6: CALCULATE COMBUSTIBLE CONTENT AT NOSE POINT OF MIXTURE

ACN1 = GN2/T
ACN2 = CO2/T
ACN3 = 100.0/((PH2/X1H2)+(PCO/X1CO)+(PCH4/X1CH4))
ACN4 = 100.0/((PH2/X2H2)+(PCO/X2CO)+(PCH4/X2CH4))
CN = (ACN1*ACN3)+(ACN2*ACN4)

CALCULATE: OXYGEN CONTENT AT NOSE POINT

AXL1 = (V1CH4*PCH4)+(V1CO*PCO)+(V1H2*PH2)
AXL2 = (PCH4/X1CH4)+(PCO/X1CO)+(PH2/X1H2)
AXL3 = (V2CH4*PCH4)+(V2CO*PCO)+(V2H2*PH2)
AXL4 = (PCH4/X2CH4)+(PCO/X2CO)+(PH2/X2H2)
AXL5 = AXL1/AXL2
AXL6 = ACN1*AXL5
AXL7 = AXL3/AXL4
AXL8 = ACN2*AXL7
XL = AXL6 + AXL8

ON = CONS3*(100.0-XL-CN)

STEPS IN EXTENDED COWARD DIAGRAM:

(A) CALCULATION OF COORDINATES OF POINTS B, C, E.

BX = XLMIX
BY = (-CONS3*XLMIX) + CONS2
CX = UPMIX
CY = (-CONS3*UPMIX) + CONS2
EX = (-CONS2*CN)/(ON-CONS2)
EY = 0
XX = SIGMAC
XY = 02

(B) COORDINATE TRANSFORMATION TO MOVE ORIGIN TO THE NOSE POINT.

BXT = BX-CN
BYT = BY-ON
CXT = CX-CN
CYT = CY-ON
EXT = EX-CN
EYT = EY-ON
XXT = XX-CN
XYT = XY-ON

(C) CONVERSION OF NEW CO-ORDINATES OF
POINTS B, C, E, AND X TO POLAR FORM
WITH THE ORIGIN AT THE NOSE POINT.

RB = DIST(BXT,BYT)
THETAB = ANGLE(PI,BXT,BYT,ZZ)
RC = DIST(CXT,CYT)
THETAC = ANGLE(PI,CXT,CYT,GG)
RE = DIST(EXT,EYT)
THETAE = ANGLE(PI,EXT,EYT,AA)
RX = DIST(XXT,XYT)
THETAX = ANGLE(PI,XXT,XYT,SS)

CALCULATION OF MODIFIED ANGULAR
CO-ORDINATES FOR SAMPLE POINT, X

XMC = THETAX-THETAC
XME = THETAX-THETAE
CMX = THETAC-THETAX
XMB = THETAX-THETAB
EMX = THETAE-THETAX
BMX = THETAB-THETAX

! (A) EXPLOSIVE CASE (0B > 0X > 0C) !
IF(THETAB .GT. THETAX .AND. THETAX .GT. THETAC) THEN
THETAM = (XMC/(XMC+BMX))*90.0

! (B) POTENTIALLY EXPLOSIVE CASE (0C > 0X OR 0X > 0E) !
ELSEIF(THETAC .GT. THETAX .OR. THETAX .GT. THETAE) THEN
THETAM = ((XME/(XME+CMX))*90.0)+270.0

! (C) NON-EXPLOSIVE CASE (WHEN (A) AND (B) DO NOT APPLY)! ELSE
THETAM = ((XMB/(XMB+EMX))*180.0)+90.0 ENDIF
XCORD = RX*COS(THETAM*PI/180.0)
YCORD = RX*SIN(THETAM*PI/180.0)

CALL TGRAM(TR,GR)
Q = AREA*VEL

CALCULATION OF LITRES
PER MIN. OF CO

PPMCO = 10000.0*CO
COPMIN = PPMCO*Q*6.0/100.0
AFVF = CONSl*02
AFVF1 = 100.0-AFVF
AFVFAC = 100.0/AFVF1
AFVCH4 = AFVFAC*CH4
AFVCO = AFVFAC*CO
AFVH2 = AFVFAC*H2
AFVCO2 = AFVFAC*CO2
SUMAFV = AFVCO+AFVCH4+AFVH2+AFVCO2
AFVN2 = 100.0-SUMAFV

OUTPUT RESULTS
WRITE(IOUT,22) DATE,TIME,02,H2,CH4,CO,CO2,
1XLMIX,UPMIX,XCORD,YCORD,TR,LOCAT,COL,PANEL,NN,ON,CN,
1BY,CY,EX,EY,SIGMAC
22 FORMAT(1X,I8,4(1X,F5.2),1X,F6.3,3(1X,F5.2),2(1X,F6.2),1X,
1F5.2,1X,A4,2(1X,A15),1X,I3,7(1X,F5.2))
WRITE(IOUT,28) DATE,TIME,02,H2,CH4,CO,CO2,GN2,
1AFVCH4,AFVH2,AFVCO,AFVN2,AFVCO2,LOCAT,COL,PANEL
28 FORMAT(1X,I8,4(1X,F5.2),1X,F6.3,7(1X,F5.2),1X,A4,2(1X,A15))
WRITE(IIOUT,43) DATE,TIME,02,H2,CH4,CO,CO2,GN2,
1C0PMIN,Q,GR,TR,LOCAT,COL,PANEL
43 FORMAT(1X,I8,4(1X,F5.2),1X,F6.3,2(1X,F5.2),1X,
1F7.0,1X,F7.2,2(1X,F5.2),1X,A4,2(1X,A15))
WRITE(IIOUT,91) DATE,TIME,02,H2,CH4,CO,CO2,
1HUM,TEMP,PRESS,LOCAT,COL,PANEL
91 FORMAT(1X,I8,4(1X,F5.2),1X,F6.3,2(1X,F5.2),3(1X,F6.2),1X,
1A4,2(1X,A15))
100 CONTINUE
CLOSE(IINPUT)
CLOSE(IPUT)
CLOSE(UNIT=IOUT,STATUS='KEEP')
CLOSE(UNIT=IIOUT,STATUS='KEEP')
CLOSE(UNIT=IIIOUT,STATUS='KEEP')
CLOSE(UNIT=IIOT,STATUS='KEEP')
STOP
END

FUNCTION dist CALCULATES DISTANCE BETWEEN VECTORS
FUNCTION dist(AA,BB)
DIST = SQRT(AA*AA+BB*BB)
RETURN
END

FUNCTION angle CALCULATES ANGLE BETWEEN TWO LINES
FUNCTION angle(PI,HH,FF,GG)
GG = (FF/HH)
CTOD = 180.0/PI
ANGLE = ATAN(GG)*CTOD
RETURN
END

SUBROUTINE tcomb CALCULATES PERCENT CONCENTRATION
SUBROUTINE tcomb(PCO,PCH4,PH2)
COMMON/GAS2/CH4,H2,CO,CO2,02,GN2,SIGMAC
SIGMAC = CH4 + H2 + CO
STEP2: CALCULATION OF CONC. OF EACH COMBUSTIBLE
AS A PERCENTAGE OF TOTAL COMBUSTIBLES
INPUTS: SIGMAC, CH4, CO, H2

PCO = (CO/SIGMAC)*100.0
PCH4 = (CH4/SIGMAC)*100.0
PH2 = (H2/SIGMAC)*100.0
RETURN
END

SUBROUTINE tgram CALCULATES BOTH THE GRAHAM &
TRICKETT'S RATIO

COMMONE/GAS2/CH4,H2,CO,CO2,O2,GN2,SIGMAC
DATA X1,X2,X3,X4,X5/0.75,0.25,0.265,20.93,79.03/

TR = (CO2+(X1*CO)-(X2*H2))/((X3*GN2)-O2)

GR = CO*100.0/((X4*GN2/X5)-O2)
RETURN
END
APPENDIX B: ADDGAS.PRG

This database program accepts the second set of data from the user. The data comprises the gas sample results and other environmental conditions (e.g., temperature and humidity). The data is transformed into a System Data File and is used as input for a Fortran program.
** Program: ADDGAS.PRG *****
*Add a set of sample readings to GASAMPLE.DBF

SELECT 1
USE GASAMPLE
SELECT 2
USE KONTER
SELECT 2
*** Reading number of sample sets *****
***** KONT is the field name for No of sample sets *****
STORE KONT TO KOUNT
SELECT 1
USE GASAMPLE

* Clean database file before adding new data.
* --------------------------------------------
ZAP
***** Use custom format GASFMT.FMT *******
***** Read data. No of times controlled by KONT *******
SET FORMAT TO GASFMT
SET COLO TO R,GR+/B,W+/RB,BG
SET DATE BRITISH
APPEND BLANK
SET COLO TO B*
?KOUNT
SET COLO TO R,GR+/B,W+/RB,BG
DO WHILE KOUNT > 0
IF KOUNT = 0
EXIT
ELSE
READ
CLEAR GETS
SET COLO TO *G
@3,10 SAY "Are your entries correct? (Y/N)"
SET COLO TO R,GR+/B,W+/RB,BG
WAIT" " TO CHOICE
DO CASE
CASE UPPER(CHOICE) = "Y"
KOUNT=KOUNT-1
CLEAR
APPEND BLANK
SET COLO TO B*
?KOUNT
SET COLO TO R,GR+/B,W+/RB,BG
LOOP
OTHERWISE
CLEAR
SET COLO TO RB
@12,12 SAY " PLEASE PRESS ANY KEY....MAKE CORRECTIONS" + ;
" AND PRESS ^END"
WAIT " "
SET COLO TO R,GR+/B,W+/RB,BG
CLEAR
LOOP
ENDCASE
ENDIF
ENDDO
CLOSE DATABASES
@12,21 SAY "PLEASE WAIT......."
USE GASAMPL
APPEND FROM GASAMPLE
CLOSE DATABASES
ERASE SAMPLE.DBF
USE GASAMPLE
**** COPYING SELECTED FIELDS TO INPUT FILE (ASCII) ****

COPY TO SAMPLE.DBF FIELDS OXYGEN, HYDROGEN, METHANE,;
CARBON_MON, CARBON_DOX, NITROGEN, HUMIDITY,;
TEMP, ATM_PRESS, TIME, AIR VELOCI, DATE
CLOSE FORMAT
CLOSE DATABASES
USE SAMPLE

* Create text. files for Fortran to read as its input *

ERASE BBASE3.TXT
COPY TO BBASE3 SDF

* After creating the text file erase the database file
* from which the text file was created.

ERASE SAMPLE.DBF
CLOSE FORMAT
CLOSE DATABASES
CLEAR
RETURN
APPENDIX C: KUNTED.PRG

This program accepts the initial set of input from the user. This input includes:

1) No. of sets of samples
2) Name of colliery
3) Name of panel
4) Location of sample(s)
5) Dimensions (width, height) of driveway.

The program further transforms the input into a System Data File which is read as input to a Fortran program.
* This program accepts data from the user and transfers some of these values to the Fortran program for analysis.

* -----------------------------------------------
* Copy selected files to Albert.dbf.
* * -----------------------------------------------
* -----------------------------------------------
* Transform the data to text files for the Fortran program.
* * -----------------------------------------------

COPY TO ALBERT.DBF FIELDS KONT,COLLIERY,PANEL,LOCATION,;
WIDTH,HEIGHT,DIAMETER
CLOSE DATABASES

* -----------------------------------------------
* After the transformation erase Albert.dbf so that next time the name can be used again.
* * -----------------------------------------------
APPENDIX D: FORGRAPH.PRG

Forgraph.prg extracts the relevant data from the database file. The extracted data is then transformed into a TEXT file for a BASIC program to plot the appropriate graph.
* This program allows the relevant text file transformation for a particular graph to be plotted

CLOSE DATABASES
SET COLO TO G
@12,21 SAY "PLEASE WAIT..."
SET COLO TO B*
@12,35 SAY "...PROGRAM RUNNING"
USE FORGRAPH
INDEX ON LOCATION+PANEL TO GRAF
STORE LOCATION TO LOKATE
STORE PANEL TO PAN
ZAP
GO TOP

* Collect only the values for a particular panel and location

APPEND FROM FORTVAL1 FOR LOCATION=LOKATE .AND. PANEL=PAN

* Make a text file of the values for the graph-plotting program to read.

COPY TO GRAPH.TXT DELIMITED
ZAP
RELEASE ALL
CLOSE DATABASES
USE AIRFREE
INDEX ON LOCATION+PANEL TO GRAF
STORE LOCATION TO LOKATE
STORE PANEL TO PAN
ZAP
GO TOP
APPEND FROM FORTVAL2 FOR LOCATION=LOKATE .AND. PANEL=PAN

* Afree.txt contains data for plotting airfree graphs.

COPY TO AFREE.TXT DELIMITED
ZAP
RELEASE ALL
CLOSE DATABASES
USE TEMPR
INDEX ON LOCATION+PANEL TO GRAF
STORE LOCATION TO LOKATE
STORE PANEL TO PAN
ZAP
GO TOP
APPEND FROM FORTVAL4 FOR LOCATION=LOKATE .AND. PANEL=PAN
COPY TO TEMPR.TXT DELIMITED
ZAP
RELEASE ALL
CLOSE DATABASES
USE FORGRAF
INDEX ON LOCATION+PANEL TO GRAF
STORE LOCATION TO LOKATE
STORE PANEL TO PAN
ZAP
GO TOP
APPEND FROM FORTVAL2 FOR LOCATION=LOKATE .AND. PANEL=PAN
COPY TO GRAF.TXT DELIMITED
ZAP
RELEASE ALL
CLOSE DATABASES
USE GTYPE
GO TOP

* *---------------------------------------------------------------------
* Gknown.txt is a text file that contains
* the graph to be plotted and for which gas.
* *---------------------------------------------------------------------

COPY TO GKNOWN.TXT DELIMITED
ZAP
CLOSE DATABASES
USE RATIOS
INDEX ON PANEL+LOCATION TO GRAF
STORE LOCATION TO LOKATE
STORE PANEL TO PAN
ZAP
GO TOP
APPEND FROM FORTVAL3 FOR LOCATION=LOKATE .AND. PANEL=PAN

* *---------------------------------------------------------------------
* Ratio.txt contains data for Graham's
* and Trickett's ratio graphs
* *---------------------------------------------------------------------

COPY TO RATIO.TXT DELIMITED
ZAP
RELEASE ALL
CLOSE DATABASES
SELECT 1
USE GKOUNT
SELECT 2
USE FORTVAL2
SELECT 3
USE GRAPHING
SELECT 1
INDEX ON LOCATION+PANEL TO GRAF
STORE LOCATION TO LOKATE
STORE PANEL TO PAN
ZAP
SELECT 2
COUNT FOR LOCATION=LOKATE .AND. PANEL=PAN TO GRAFKT
SELECT 3
REPLACE KONT WITH GRAFKT

* *---------------------------------------------------------------------
* Grafko.txt contains number of plotting points.
* *---------------------------------------------------------------------

COPY TO GRAFKO SDF
RELEASE ALL
CLOSE DATABASES
CLEAR
RETURN
APPENDIX E: FORTIMPT.PRG

This database program takes the output from the Fortran and appends them into the appropriate database file.
* Gasot.txt, Gasout.txt, Gsout.txt and Gout.txt
* are the output files created by the Fortran
* program. In this program, the data from Fortran
* are appended to the database file. For example
* FORTVAL1.dbf accepts data from Gasot.txt.
* ---------------------------------------------------------------

CLOSE DATABASES
CLEAR
SET COLO TO *G
@12,21 SAY "PLEASE WAIT....."
USE FORTVAL1
INDEX ON LOCATION+PANEL+COLLIERY TO LIST1
APPEND FROM GASOT DELI WITH BLANK
CLOSE DATABASES
USE FORTVAL2
INDEX ON LOCATION+PANEL+COLLIERY TO LIST2
APPEND FROM GASOUT DELI WITH BLANK
CLOSE DATABASES
USE FORTVAL3
INDEX ON LOCATION+PANEL+COLLIERY TO LIST3
APPEND FROM GSOUT DELI WITH BLANK
CLOSE DATABASES
USE FORTVAL4
INDEX ON LOCATION+PANEL+COLLIERY TO LIST4
APPEND FROM GOUT DELI WITH BLANK
SET COLO TO R,GR+/B,W+/RB,BG
CLOSE DATABASES
CLEAR
RETURN
APPENDIX F: CHOSELIS.PRG

This database program displays a menu. Each menu choice displays a different set of output of the gas analysis.
** This program lists the results of sample analysis
* It allows the user to select results to print or
* view on the screen.
* -------------------------------------------------------------------
SET DATE BRITISH
SET TALK OFF
SET SAFETY OFF
SET COLO TO R,GR+/B,W+/RB,BG
STORE " " TO CHOICE
DO WHILE .T.
CLEAR
?
?
?
".
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?
?

<table>
<thead>
<tr>
<th>Task Code</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;A&gt;</td>
<td>GAS INPUT / X,Y CORD / TR RATIO</td>
</tr>
<tr>
<td>&lt;B&gt;</td>
<td>GAS INPUT / AIR FREE VALUES</td>
</tr>
<tr>
<td>&lt;C&gt;</td>
<td>GAS INPUT / CO PER MIN / GR &amp; TR RATIOS</td>
</tr>
<tr>
<td>&lt;D&gt;</td>
<td>GAS INPUT / PRESS / TEMP / HUM / ETC</td>
</tr>
<tr>
<td>&lt;Q&gt;</td>
<td>QUIT (RETURN TO MAIN MENU)</td>
</tr>
</tbody>
</table>

WAIT " Enter your choice (A-Q) from above:" TO CHOICE
** ------ Branch to appropriate program.
DO CASE
CASE UPPER(CHOICE)="A"
  DO LIST1
  WAIT
CASE UPPER(CHOICE)="B"
  DO LIST2
  WAIT
CASE UPPER(CHOICE)="C"
  DO LIST3
  WAIT
CASE UPPER(CHOICE)="D"
  DO LIST4
  WAIT
CASE UPPER(CHOICE)="Q"
  QUIT
  ENDCASE
  ENDDO
CLOSE DATABASES
RETURN
APPENDIX G: LIST1.PRG

This program displays the first of the four different outputs. The others are displayed by LIST2.PRG, LIST3.PRG and LIST4.PRG. These choices are displayed by the CHOSELIS.PRG (Appendix F).
CLOSE DATABASES
USE FORTVAL1
INDEX ON PANEL+LOCATION TO LIST1
CLEAR
@10,16 SAY "DO YOU WANT A HARD COPY? (Y/N)"
WAIT " " TO CHOICE
DO WHILE .T.
DO CASE
  CASE UPPER(CHOICE) = "N"
    REPORT FORM LISTING1
    RETURN
  CASE UPPER(CHOICE) = "Y"
    SET COLO TO G
    INPUT "HOW MANY COPIES?" TO ANSWER
    CLEAR
    @12,12 SAY "PREPARE PRINTER .. PRESS ANY KEY WHEN READY"
    WAIT " ".
    DO WHILE .T.
      IF ANSWER = 0
        RETURN
      ELSE
        REPORT FORM LISTING1 TO PRINT
        ANSWER = ANSWER - 1
      LOOP
    ENDDO
  OTHERWISE
    CLEAR
    SET COLO TO RB
    @10,16 SAY "PLEASE ANSWER (Y)es OR (N)o"
    SET COLO TO R
    @12,16 SAY "DO YOU WANT A HARD COPY? (Y/N)"
    WAIT " " TO CHOICE
    LOOP
  ENDCASE
ENDDO
SET COLO TO R,GR+/B,W+/RB,BG
CLOSE DATABASES
RETURN
APPENDIX H: ZAPALL.PRG

ZAPALL.PRG command file is the last database program. Its function is to "ZAP", that is, empty all the database files of their contents. In addition all system created files, including "text" files are removed permanently.
* This program deletes all the data in all the database files. It also deletes all files. It must therefore be used with care.

CLEAR
@13,13 SAY "ARE YOU SURE YOU WANT TO CLEAN" + ; " DATABASES ? (Y/N)"
WAIT " " TO CHOICE
DO WHILE .T.
DO CASE
CASE UPPER(CHOICE) = "N"
CLEAR
EXIT
CASE UPPER(CHOICE) = "Y"
USE GASAMPL
ZAP
CLOSE DATABASES
USE KONTER
ZAP
CLOSE DATABASES
USE FORTVAL1
ZAP
CLOSE DATABASES
USE FORTVAL2
ZAP
CLOSE DATABASES
USE FORTVAL3
ZAP
CLOSE DATABASES
USE FORTVAL4
ZAP
CLOSE DATABASES
USE GASAMPLE
ZAP
CLOSE DATABASES
ERASE BBASE.TXT
ERASE BBASE3.TXT
ERASE GASOUT.TXT
ERASE GSOUT.TXT
ERASE GASOT.TXT
ERASE GOUT.TXT
CLEAR
EXIT
OTHERWISE
@15,13 SAY "PLEASE ANSWER (Y)es OR (N)o"
WAIT " " TO CHOICE
CLEAR
LOOP
ENDCASE
ENDDO
RETURN
Appendix I is one of the BASIC graph-plotting programs
**THIS PROGRAM PLOTS GRAPHS FOR AIR FREE VALUES VRS TIME **

** JANUARY 1988 **

** VARIABLES USED **

- TM(100) -- Array variable for sample time
- GAS(100) -- Array variable for the gas to be plotted
- AFVXXX(100) -- Array variable for Air Free Value for gas XXX
- YR(100) -- Array variable for year of date
- MT(100) -- Array variable for month of date
- DAY(100) -- Array variable for day of date
- PX(100) -- Array variable for X-Cordinate of graph
- PY(100) -- Array variable for Y-Cordinate of graph

SCREEN 2, 0: CLS : KEY OFF
DIM TM(100), GAS(100), AFVH2(100), AFVCH4(100)
DIM YR(100), MT(100), DAY(100), AFVCO(100), AFVN2(100), AFVCO2(100)
DIM PX(100), PY(100)

'Ask User to input date
LOCATE 10, 15: PRINT "Enter Today's date below:
LOCATE 14, 15: INPUT "Day:", Day$
LOCATE 16, 15: INPUT "Month:", Month$
CLS

'Store date to the variable SAM
SAM$ = Day$ + "/" + Month$ + "/" + Year$

'Draw box to contain name of panel
LOCATE 3, 4: PRINT SAM$
LINE (10, 10)-(10, 190)
LINE (10, 190)-(630, 10)
LINE (630, 10)-(630, 190)
LINE (140, 30)-(140, 140)
LINE (140, 140)-(550, 140)
LOCATE 20, 39: PRINT "Time (Hours)"
A = 8
LOCATE 6, 15: PRINT "%"

'Open Input file
-- GKNOWN.TXT is a TEXT file containing the name
of gas to plot for
OPEN "I", #3, "GKNOWN.TXT"
INPUT #3, GS$, GRAF$
SS = LEN(GS$)
IF SS > 11 THEN SS = 10
FOR I = 1 TO SS
PP$ = MIDS(GS$, I, 1)
LOCATE A, 15: PRINT PP$
A = A + 1
NEXT I
CLOSE 3

'Open Input file
OPEN "I", #3, "GKNOWN.TXT"
INPUT #3, gas$, GRAPH$
CLOSE 3
LOCATE 3, 30: PRINT "AIR FREE VALUE GRAPH"
LOCATE 4, 30: PRINT "---------------------"

'The file "GRAFKO.TXT" contains the number of samples.
The number is read and then used as a counter for a loop
OPEN "I", #4, "GRAFKO.TXT"
OPEN "I", #1, "AFREE.TXT"  ' Open file containing Air Free Values
OPEN "I", #2, "GFOTT.TXT"   ' Open file containing date

\[ \text{Input air free values for the first sample set} \]
\[
\text{INPUT } #1, \ DT, \ TM(1), \ AFVH2(1), \ AFVCH4(1), \ AFVCO(1), \ AFVCO2(1), \ AFVN2(1), \ LC$, \ COL$, \ PAN$
\]
\[ \text{ Input the data of first sample} \]
\[
\text{INPUT } #2, \ YR(1), \ MT(1), \ Day(1)
\]

\[ \text{The initial time (ie time for first sample) is} \]
\[ \text{made the reference time. This time is referred to as} \]
\[ \text{time zero.} \]
\[ \text{all other times are referenced to the initial time} \]

\[
X1 = TM(1)
\]
\[
YR1 = YR(1)
\]
\[
MT1 = MT(1)
\]
\[
DAY1 = Day(1): \text{CLOSE 1: CLOSE 2}
\]
\[ \text{OPEN "I", #1, "AFREE.TXT"} \]
\[ \text{OPEN "I", #2, "GFOTT.TXT"} \]

\[ \text{FOR } N = 1 \text{ TO } NN \]

\[ \text{'Now read the other air free values And} \]
\[ \text{The times for these samples} \]
\[ \text{INPUT #1, DT, TM(N), AFVH2(N), AFVCH4(N), AFVCO(N), AFVCO2(N), AFVN2(N), LC$, COL$, PAN$
\]
\[ \text{INPUT #2, YR(N), MT(N), Day(N)} \]
\[ \text{IF } N > 1 \text{ GOTO 10} \]

\[ \text{'PAN$' variable containing name of panel for the plot} \]
\[ \text{'LC$' variable containing name of location for plot} \]
\[ \text{LOCATE 23, 20: PRINT "PANEL: "; PAN$, " LOCATION: "; LC$} \]

\[ \text{'GAS$ is instantiated to the user's choice of the gas} \]
\[ \text{to be plotted. For example if the user wants to plot} \]
\[ \text{'the air free value for hydrogen then} \]
\[ \text{GAS$ is instantiated to "HYDROGEN"} \]
\[ \text{'Now select the appropriate graph} \]

\[ \text{10 SELECT CASE gas$} \]
\[ \text{CASE "Hydrogen"} \]
\[ \text{gas(N) = AFVH2(N)} \]
\[ \text{CASE "Methane"} \]
\[ \text{gas(N) = AFVCH4(N)} \]
\[ \text{CASE "Carbon_monoxide"} \]
\[ \text{gas(N) = AFVCO(N)} \]
\[ \text{CASE "Carbon_dioxide"} \]
\[ \text{gas(N) = AFVCO2(N)} \]
\[ \text{CASE ELSE} \]
\[ \text{gas(N) = AFVN2(N)} \]
\[ \text{END SELECT} \]

\[ \text{Subtract the initial date from all subsequent dates} \]
\[ \text{This makes the dates and times relative to the starting} \]
\[ \text{dates and times.} \]
\[ \text{Thus we speak of X hours of elasp time from} \]
\[ \text{the start of the fire} \]
\[ \text{DIFF1} = YR(N) - YR1 \]
\[ \text{DIFF2} = MT(N) - MT1 \]
\[ \text{DIFF3} = Day(N) - DAY1 \]

\[ \text{xvii} \]
```plaintext
TM(N) = TM(N) - XI

'Change the elapse time into days

MM1 = 12 * DIFF1
MM2 = 31 * (MM1 + DIFF2)
MM3 = 24 * (MM2 + DIFF3)
TM(N) = TM(N) + MM3
PX(N) = TM(N)
PY(N) = gas(N)
NEXT N

FOR K = 1 TO NN
  IF K > 1 GOTO 500
  XMAX = PX(K)
  YMAX = PY(K)
  IF K = 1 GOTO 550

  'Get the maximum of the X and Y coordinates
  'for scaling

500 IF XMAX < PX(K) THEN XMAX = PX(K)
  IF YMAX < PY(K) THEN YMAX = PY(K)

550 NEXT K

XSCALE = 410 / XMAX
YSCALE = 105 / YMAX

FOR L = 1 TO NN
  FOR J = 1 TO 20000: NEXT J
  'Make sure that the points fall within the axes

  PX = 140 + (TM(L) * XSCALE)
  PY = (140 - (gas(L) * YSCALE))
  IF PX > 550 THEN PX = 550
  IF PX < 140 THEN PX = 140 + PX(L)
  IF PY > 140 THEN PY = 140 - PY(L)
  IF PY < 30 THEN PY = 30
  IF L = 1 GOTO 700

  'Draw circles at the sample points and
  'connect points with lines

700 CIRCLE (PX, PY), 2.5
  LINE -(PX, PY)
  NEXT L

CLOSE 1: CLOSE 2

800 IF INKEY$ = "" THEN 800
SCREEN 0: WIDTH 80: KEY ON: END
```

APPENDIX J: This is the listing of the Expert System Code.
   It was written in the VP-Expert Shell.
FIND graph_to_plot
FIND send_to_dbf;

RULE 7a
IF decision = plot
THEN plotting = yes
FIND which_graph;

RULE 8a
IF plotting = yes AND
which_graph = ellicott
THEN graph_to_plot = ellicott_dia
MENU which_panel,ALL,fortvall,panel
FIND which_panel
MENU give_location,which_panel = panel,fortvall,location
FIND give_location;

RULE 8b
IF plotting = yes AND
which_graph = Coward
THEN graph_to_plot = Coward_dia
MENU which_panel,ALL,fortvall,panel
FIND which_panel
MENU give_location,which_panel = panel,fortvall,location
FIND give_location;

RULE 9a
IF plotting = yes AND
which_graph = air_free_vrs_time
THEN graph_to_plot = air_free
gastype
MENU which_panel,ALL,fortvall,panel
FIND which_panel
MENU give_location,which_panel = panel,fortvall,location
FIND give_location;

RULE 10a
IF plotting = yes AND
which_graph = gas_content_vrs_time
THEN graph_to_plot = gas
gastype
MENU which_panel,ALL,fortvall,panel
FIND which_panel
MENU give_location,which_panel = panel,fortvall,location
FIND give_location;

RULE 11a
IF plotting = yes AND
which_graph = temp_vrs_time
THEN graph_to_plot = temp
MENU which_panel,ALL,fortvall,panel
FIND which_panel
MENU give_location,which_panel = panel,fortvall,location
FIND give_location;
RULE 12a
IF plotting = yes AND
which_graph = tr_ratio_vrs_time
THEN graph_to_plot = tratio
MENU which_panel,ALL,fortvall.panel
FIND which_panel
MENU give_location,which_panel = panel,fortvall,location
FIND give_location;

RULE 13a
IF plotting = yes AND
which_graph = gr_ratio_vrs_time
THEN graph_to_plot = gratio
MENU which_panel,ALL,fortvall.panel
FIND which_panel
MENU give_location,which_panel = panel,fortvall,location
FIND give_location;

RULE 14a
IF plotting = yes AND
which_graph = co_per_min_vrs_time
THEN graph_to_plot = copermin
MENU which_panel,ALL,fortvall.panel
FIND which_panel
MENU give_location,which_panel = panel,fortvall,location
FIND give_location;

RULE 15a
IF graph_to_plot = ellicott_dia
THEN send_to_dbf = okay
GET ALL,forgraph,ALL
panel = (which_panel)
location = (give_location)
APPEND forgraph
CLOSE forgraph
GET ALL,gkount,ALL
panel = (which_panel)
location = (give_location)
APPEND gkount
CLOSE gkount
GET ALL,gtype,ALL
gass = (gastype)
graph_type = (which_graph)
APPEND gtype
CLOSE gtype
CALL forgraph
CALL Q6EX;

RULE 15b
IF graph_to_plot = Coward_dia
THEN send_to_dbf = okay
GET ALL, forgraph, ALL
panel = (which_panel)
location = (give_location)
APPEND forgraph
CLOSE forgraph
CLOSE GET
GET ALL, gkount, ALL
panel = (which_panel)
location = (give_location)
APPEND gkount
CLOSE gkount
CLOSE GET
GET ALL, gtype, ALL
gass = (gastype)
graph_type = (which_graph)
APPEND gtype
CLOSE gtype
CLOSE GET
CALL forgraph
CALL COWD;

RULE 16a
IF graph_to_plot = air_free
THEN send_to_dbf = okay
GET ALL, airfree, ALL
panel = (which_panel)
location = (give_location)
APPEND airfree
CLOSE airfree
CLOSE GET
GET ALL, gkount, ALL
panel = (which_panel)
location = (give_location)
APPEND gkount
CLOSE gkount
CLOSE GET
GET ALL, gtype, ALL
gass = (gastype)
graph_type = (which_graph)
APPEND gtype
CLOSE gtype
CLOSE GET
GET ALL, forgraf, ALL
panel = (which_panel)
location = (give_location)
APPEND forgraf
CLOSE forgraf
CLOSE GET
CALL forgraph
CALL GRAPH
CALL AFREE;

RULE 17a
IF graph_to_plot = gas
THEN send_to_dbf = okay
GET ALL, forgraf, ALL
panel = (which_panel)
location = (give_location)
APPEND forgraf
CLOSE forgraf
CLOSE GET
GET ALL, gkount, ALL
panel = (which_panel)
location = (give_location)
APPEND gkount
CLOSE gkount
CLOSE GET
GET ALL, gtype, ALL
gass = (gastype)
graph_type = (which_graph)
APPEND gtype
CLOSE gtype
CLOSE GET
CALL forgraph
CALL GRAPH
CALL GASVAL;

RULE 18a
IF graph_to_plot = temp
THEN send_to_dbf = okay
GET ALL, tempres, ALL
panel = (which_panel)
location = (give_location)
APPEND tempres
CLOSE tempres
CLOSE GET
GET ALL, gkount, ALL
panel = (which_panel)
location = (give_location)
APPEND gkount
CLOSE gkount
CLOSE GET
GET ALL, gtype, ALL
gass = (gastype)
graph_type = (which_graph)
APPEND gtype
CLOSE gtype
CLOSE GET
GET ALL, forgraf, ALL
panel = (which_panel)
location = (give_location)
APPEND forgraf
CLOSE forgraf
CLOSE GET
CALL forgraph
CALL GRAPH
CALL TEMP;
RULE 19a
IF graph_to_plot = tratio OR
graph_to_plot = gratio OR
graph_to_plot = copermin
THEN send_to_dbf = okay
GET ALL, ratios, ALL
panel = (which_panel)
location = (give_location)
APPEND ratios
CLOSE ratios
CLOSE GET
GET ALL, gkount, ALL
panel = (which_panel)
location = (give_location)
APPEND gkount
CLOSE gkount
CLOSE GET
GET ALL, gtype, ALL
gass = (gastype)
graph_type = (which_graph)
APPEND gtype
CLOSE gtype
CLOSE GET
GET ALL, forgraf, ALL
panel = (which_panel)
location = (give_location)
APPEND forgraf
CLOSE forgraf
CLOSE GET
CALL forgraph
CALL GRAPH
CALL RATIOS;

RULE 20a
IF want = Consult_ES
THEN decision = expert_system
others = ok
FIND cause
FIND action1
FIND action2
FIND action3
FIND action4;

RULE 21a
IF co_oxy_def_ratio = rising AND
sweating = yes AND
source = fall
THEN cause = fall
BECAUSE "A rise of CO/02 deficiency ratio indicates a possible
heating. The heating will be accompanied by sweating.~";

RULE 21b

IF co_oxy_def_ratio = steady OR
   co_oxy_def_ratio = decreasing

THEN cause = no
CLS
DISPLAY "

You cannot have fire or heating with no increase in the CO/02 deficiency ratio. Check the trend of the CO/02. If it is rising and you have detected smoke then it is likely you have got fire. If the trend is on the increase with no accompanying smoke then heating may be probable, otherwise there is no cause for alarm. Keep the situation under observation.""

CLS;

RULE 22a

IF co_oxy_def_ratio = rising AND
   sweating = yes AND
   coal_condition = fresh_coal AND
   graham_ratio = 0.5_1.0

THEN situation = heating;

RULE 22b

IF co_oxy_def_ratio = rising AND
   sweating = no AND
   confirmed = Yes

THEN situation = heating
CLS
DISPLAY "

You have not observed any abnormal sweating from workers, but since you have confirmed the rise of CO/02 ratio, I will treat this as heating or fire Please continue to answer my questions. Press any key.~"

CLS;

RULE 22c

IF co_oxy_def_ratio = rising AND
   sweating = no AND
confirmed = No
THEN situation = none
CLS
DISPLAY

"Since you haven't got any abnormally high temperature (no sweating), and you could not confirm a rise of CO/O2 ratio, heating or fire cannot be confirmed."

CLS;
RULE 23a

IF co_oxy_def_ratio = rising AND
   sweating = yes AND
   coal_condition = old_coal AND
   graham_ratio = 1.0_2.0
THEN situation = heating;

RULE 24a

IF situation = heating AND
   locat = intake AND
   fall_location = in_old_workings
   THEN answer = yes
       action1 = seal_the_area
       action2 = nil
       action3 = nil
       action4 = nil
   CLS
   DISPLAY "

You have {situation} in an {fall_location} area
Since there is no activity in this area, my advise to you is to:

{action1} completely
continue to monitor the situation~"
RULE 25a

IF situation = heating AND
   locat = intake AND
   fall_location = maj_dev_area

   THEN answer = no;

RULE 26a

IF answer = no AND
   roof_cond = bad

   THEN roof_type = bad
   CLS
   action1 = inertise_and_seal
   action2 = seal_off
   action3 = nil
   action4 = nil
   CLS
   DISPLAY "You have got {situation} in a
   {roof_type} ground. Since the
   problem is in a {fall_location},
   you have two options:-

   1. {action1}
   2. {action2}

   The option you take will depend on
   how soon you want normal activities
   to resume in this {fall_location}.~"

   CLS;

RULE 27a

IF answer = no AND
   roof_cond = good

   THEN roof_type = good;

RULE 28a

IF roof_type = good AND
   cause = fall AND
   fall_size = small

   THEN vent = do_not_ask
   heating_stage = do_not_ask
   CLS
   action1 = cool_with_water
   action2 = load_out
   CLS
   DISPLAY "It is likely that you have got {situation}
   You may take the following actions:
1. {action1} and  
2. {action2}  
3. ALSO  
continue to monitor the gas concentrations. We expect a (gradual) decrease in oxygen concentration.  
IF the concentrations of the gases (H2, CO, CH4 CO2) are increasing at a higher rate than the decrease in concentration of oxygen then consider sealing off ~"  
RULE 29a  

IF  
roof_type = good AND  
cause = fall AND  
fall_size = large AND  
secured_roof = no  

THEN  
action1 = sealing  
action2 = nil  
action3 = nil  
action4 = nil  
CLS  
DISPLAY "  
You have got a {fall_size} {cause} in a {fall_location} Even though your roof condition is good, you have told me that the inbye and outbye edges of the fall are not secured and well supported. I will therefore advise you to consider {action1}.~"  
CLS;  

RULE 30a  

IF  
roof_type = good AND  
cause = fall AND  
fall_size = large AND  
smoke_detected = no  

THEN  
CLS  
action1 = secure_the_roof  
action2 = cool_with_water  
action3 = load_out  
action4 = nil  
CLS  
DISPLAY "
I have detected that you have got {situation} in its early stage. Your fall is large and edges are secured. You are advised to perform the following sequence of actions:

1. {action1} further
2. {action2}
3. {action3} and then...
3. Continue to monitor the atmosphere~"
THEN situation = inconsistent
FIND report;

RULE 31c
IF situation = inconsistent
THEN report = yes
CLS
DISPLAY "

Your answers are not consistent with facts. Fresh coal should have Graham Ratio of between 0.5 and 1.0. Old coal should have a ratio of between 1.0 and 2.0. You said you have {coal_condition} but the ratio does not correspond to that of {coal_condition}."

CLS;

RULE 32a
IF roof_type = good AND
  cause = fall AND
  fall_size = small
THEN action1 = cool_area_with_water
  action2 = load_out
CLS
DISPLAY"

Your fall is small and roof conditions are good. These are ideal conditions to:

1. {action1}
2. {action2}"

CLS;

RULE 33a
IF situation = heating AND
  locat = return AND
  fall_location = maj_dev_area AND
  roof_cond = good AND
  fall_size = small AND
  smoke_detected = no
THEN CLS
  action1 = cool_area_with_water
  action2 = load_out
  action3 = nil
  action4 = nil
CLS
XXX
You have got {situation} in its early stage.
Your roof condition is {roof_cond}.
You may consider the following:

1. {action1} and
2. {action2}~"
I suggest you do the following:

1. {action1} and

2. apply {action2}
3. Since the problem is in a production ground you may consider also to {action3}.

CLS;

RULE 36a
IF
situation = heating AND
locat = return AND
fall_location = maj_dev_area AND
fall_size = large AND
roof_cond = good AND
smoke_detected = yes
THEN
CLS
action1 = seal
action2 = sandstone
action3 = inertize
action4 = nil
CLS
DISPLAY"

You have got {situation} in a {roof_cond} ground. Since you have detected smoke it means the {situation} is in an advanced stage. You may consider the following:

1. {action1} and
2. apply {action2}
3. Since the problem is in a production area you may consider also to {action3}.

CLS;

RULE 37a
IF
situation = heating AND
locat = return AND
fall_location = maj_dev_area AND
fall_size = large AND
roof_cond = good AND
smoke_detected = no
THEN
CLS
action1 = cool_area_with_water
action2 = load_out
action3 = nil
action4 = nil
CLS
DISPLAY"

You have got {situation} in a {roof_cond} ground. Since you have not detected smoke it means the {situation} is in a normal stage. You may consider the following:

1. {action1} and
2. apply {action2}
3. Since the problem is in a production area you may consider also to {action3}.
{situation} is in an early stage. You may consider the following:

1. {action1} and
2. {action2}~"
RESET graph_to_plot
RESET gastype
RESET APTOGRK
RESET gass
RESET kalled
RESET panel
RESET location
RESET graph_type
RESET cause
RESET action1
RESET report

RESET action2
RESET action3
RESET action4
RESET co_oxy_def_ratio
RESET sweating
RESET confirmed
RESET source
RESET coal_condition
RESET graham_ratio
RESET situation
RESET locat
RESET fall_location
RESET answer
RESET roof_cond
RESET roof_type
RESET fall_size
RESET vent
RESET heating_stage
RESET secured_roof
RESET smoke_detected
!CLS
FIND decision
END;

ASK want :"Please select the task you wish to perform";
ASK which_panel: "Choose the panel you want";
ASK give_location: "And choose your location";
ASK which_graph: "what type of graph do you want to plot?";
ASK gastype: "Which of the gases do you want a plot for?";
CHOICES want: Input_data,Plot_graph,Print_results,
Clean_databases,Consult_ES;

CHOICES which_graph : Ellicott,Air_free_vrs_time,
Gas_content_vrs_time,Temp_vrs_time,
Gr_ratio_vrs_time,Tr_ratio_vrs_time,
co_per_min_vrs_time,Coward;

CHOICES gastype : Oxygen,Hydrogen,Methane,Nitrogen,
Carbon_monoxide,Carbon_dioxide;

ASK co_oxy_def_ratio: "How is the CO/O2 deficiency ratio ?";
CHOICES co_oxy_def_ratio: rising, steady, decreasing;
ASK sweating: "Is the atmosphere causing sweating?";
CHOICES sweating: yes, no;
ASK coal_condition: "Estimate the condition of the coal";
CHOICES coal_condition: fresh_coal, old_coal;
ASK graham_ratio: "What is the figure for the Graham's Ratio?";
CHOICES graham_ratio: 0.5_1.0, 1.0_2.0;
ASK smoke_detected: "Have you detected any smoke?";
CHOICES smoke_detected: yes, no;
ASK source: "What is the source of the problem?";
CHOICES source: fall, spoil_heap;
ASK locat: "What is the location of the {cause}?";
CHOICES locat: intake, return;
ASK fall_size: "Estimate the size of the fall?";

CHOICES fall_size: small, large;
ASK roof_cond: "Give me the condition of your roof";
CHOICES roof_cond: good, bad;
ASK ventilation: "Try and tell me something about the ventilation";
CHOICES ventilation: normal, reduced;
ASK fall_location: "In what area is the problem situated?";
CHOICES fall_location: in_old_workings, maj_dev_area;
ASK secured_roof: "Are the inbye and outbye edges of the fall secured well supported?";
CHOICES secured_roof: yes, no;
ASK confirmed: "Are you really sure the CO/O2 is on the rise?";
CHOICES confirmed: Yes, No;