Safety in Mines Research Advisory Committee

Final Project Report

Numerical Modelling of Mine Workings

N Lightfoot
&
M Maccelari

Research Agency : CSIR Division of Mining Technology
Project Number : GAP 415
Date : March 1999

Executive Summary

This report covers the production of a general reference handbook of numerical modelling for the South African mining industry, with particular reference to the gold and platinum industries. It also covers a research sub-project investigating the use of expert systems and intelligent user interfaces to aid in the process of numerical modelling.

Although the bulk of the project involved the production of the numerical modelling handbook, most of this report concerns the expert system and user interface sub-project. This is because the handbook is essentially self-documenting, and the expert system and user interface sub-project is not documented elsewhere.

The final output of the numerical modelling handbook consists of two published volumes:

Volume I	Principles of Numerical Modelling,
Volume II	Example Applications of Numerical Modelling

In addition, there is a CD-ROM with electronic versions of both volumes in Portable Document Format (PDF) that can be read by software industry standard readers such as Adobe Acrobat Reader 3. Finally, a World Wide Web site has been constructed to provide interactive access to, and feedback from, this project.

Volume I of the handbook includes five chapters covering fundamental aspects of numerical modelling. These chapters are:

- Overview of Numerical Modelling
- Solid Mechanics
- Modelling Criteria
- Material Models
- Solution Methods

Following this are chapters covering the five most commonly used numerical modelling programs in the South African gold and platinum mining industries. These programs are

- BESOL/MINAP_97
- MINSIM-W
- BESOL/MS
- MAP3D
- FLAC

This volume is intended to provide a self-contained reference to the application of these programs. Enough theory and relevant equations and transformations are given to cover most of what is required for a practising rock mechanics engineer to be able to use any of these five programs to solve practical mining problems. The chapters on specific programs discuss their individual strengths and weaknesses and highlight certain idiosyncrasies. These sections are not intended as replacements to the program manuals, but rather add value to the existing documentation in the South African mining context.

Volume II is a series of worked examples illustrating both principles and applications of numerical modelling in the context of the South African gold and platinum mining industries. This includes an example that utilises a number of different numerical modelling programs to solve a single problem. This particular example describes the application of a methodology for numerical modelling of mine design. Further examples cover the application of specific programs including guidelines on how to select actual input parameters for given situations.

The work on expert systems covers a review of expert systems in general followed by a brief synopsis of specific examples related directly to mining. It concludes with a more detailed example of a specific system associated with numerical modelling. The intelligent interface work covers the development of prototypes of both a Modeller Wizard and a Visual Modeller. Both provide interactive, domain specific interfaces to assist with the preparation of numerical modelling simulations. They are intended to guide the rock mechanics engineer constructing input for a numerical simulation in such a way as to avoid incorrect input. The contention is that the engineer can be isolated from certain control parameters and calculations the input of which can be automated. In addition, particular parameters can be assessed for validity at an early stage to avoid incorrect input. Furthermore, conflicting states can be avoided if they are captured at the construction point.

Such intelligent user interfaces are individually designed to address specific problem domains or types. They are not designed to be program specific, but rather can help identify the most suitable numerical modelling program available for solving particular problems, and can then create the necessary input data.

This phase of the project concludes that numerical modelling remains sufficiently in the realm of art rather than science, such that expert systems are not well suited to aiding in the facilitation of valid numerical modelling. The concept of a well designed input engine to help construct the model for simulation is a more pragmatic approach to resolving problems associated with complicated numerical models. The interface can be used to capture and filter bad input. It can be designed to ensure that any input submitted to the numerical modelling program is valid.

Acknowledgements

We would like to express our gratitude to SIMRAC for financial support for project GAP415.

We would like to thank Ms. Helen Semenya and Dr. Steve Webber for their contribution to the multimedia content of this project. We would like to thank the contributing authors who provided the basis for either whole chapters or particular sections of chapters. Specifically these include Mr. Tony Leach of Itasca Africa, Mr. Willem Theron of Mining Stress Systems and Mr. Roger Johnson of Snowden. We are indebted to numerous colleagues who gave their time willingly to provide peer review, particular gratitude goes to Dr. John Napier and Ms. Lindsey Anderson. We would like to thank both our colleagues at Miningtek and the members of the GAPREAG committee who gave constructive feedback and active encouragement throughout the course of this project.

Particular thanks go to the staff at GCC who acted to ensure we had a quality document ready on time for publication. Finally we would like to thank Mrs. Wendy Smith for her exhaustive proofreading of the document and Mrs. Beverly Webster for her generous contributions towards typing.

Table of Contents

List of Figures

List of Tables

Glossary of abbreviations, symbols and terms

Abbreviations

API	Application Programmer’s Interface
ASCII	American Standard Character Information Interchange
CAI	Computer Aided Instruction
CAT	Computer Aided Training
DTP	Desk Top Publishing
GDI	Graphics Device Interface
GUI	Graphical User Interface
ICAT	Intelligent Computer Aided Training
IFE	Intelligent Front End
IO	Input Output (system)
OOP	Object Oriented Programming
PDF	Portable Document Format
RAD	Rapid Application Development
URL	Universal Resource Location

1 Introduction

A total of five enabling outputs were planned for project GAP415, however, one of these, the intelligent front end, was sub-divided from the rest of the project. As a result of this initial sub-division, this report will deal first with the other four directly related enabling outputs, and will then discuss the development of the intelligent front end.

As the guidebook in itself provides documented output, this report will only briefly discuss the work that went into its production. It will describe the methodology used to create the document and it will cover the resources involved. However, there is no intention to repeat the contents of the guidebook itself.

As the intelligent front end is not documented anywhere else, but in this report, a fair amount of detail will accompany this aspect of the report. However, as both the application and the source code are freely available to SIMRAC, issues that are best described by these are not covered in this report.

1.1 Problem Statement

For a number of years now it has been recognised by many senior members of the South African rock engineering fraternity that there is a general problem when it comes to the use of numerical modelling to solve mine design problems. Field engineers are unsure of which numerical modelling programs are best suited to solving the problem at hand. Even if they can identify the most suitable modelling programs and are aware of practical modelling methodologies, they are often unsure of what values to use for input parameters. Indeed, they are often unsure about what the input parameters themselves mean.

The literature and folklore surrounding rock engineering numerical modelling is diverse and confusing. It is often difficult to find the answer to simple specific questions. In some cases, different texts give different answers to the same question. Often the terminology used is inappropriate, ambiguous or simply incorrect. Certain terms that have gained colloquial use in the South African mining industry have a completely different meaning in the international arena.

The number of available textbooks relating directly to the use of numerical modelling in hard rock, shallow to deep, underground mining is disappointingly small. Many journal articles that refer to numerical modelling simulations for solving specific problems tend to ignore, or gloss over, the details of the modelling process employed. Furthermore, in some cases, such articles describe modelling processes that are either inappropriate or incorrect.

It has become necessary to collate all valuable information relating to the use of numerical modelling to solve mine design problems in the South African gold and platinum industries. In addition, it has become necessary to correct the ambiguities and to cull the incorrect and misleading information.

In addition to collecting and correcting existing information, a need was recognised to address a number of important questions that have dogged the industry since it was first introduced to numerical modelling. Some questions are very general while others are more specific. An example of a more general question is how do test laboratory results relate to the *in situ* rock mass? A more specific question is how does non-linear material behaviour affect path dependency in otherwise linear elastic models?

Even from the outset the scope of this project was recognised to be large. Exactly how large an undertaking it really was only became apparent during the course of the project itself.

1.2 Objectives of this Project

This project set out to solve the problem of the inappropriate use of numerical modelling for solving mine design problems in a number of ways. Five primary solutions were proposed prior to the outset of actual work, these were:

a) a handbook of available numerical modelling programs,
b) a guidebook containing examples of mine design problem solving using numerical modelling,
c) expert systems to help in the implementation process of numerical modelling,
d) intelligent user interfaces for existing numerical modelling programs, and
e) training courses to teach both the background to and the use of numerical modelling for solving mine design problems.

When the proposed project resources were considered it became clear that not all of these solutions could be addressed individually. After consideration it was decided to combine solutions c) and d) and to investigate the use of intelligent user interfaces that incorporate aspects of expert system technology, and not to provide training courses as a direct output from this project.

1.3 Numerical Modelling Guidebook

The project was originally planned for a total of 518 man-days over a period of 2 years. In total, this equates to 2.4 man-years over that time, or slightly more than one individual’s time for the entire project period.

The actual enabling outputs for the guidebook were broken down into four sections in the original project proposal. These sections are:

Enabling Output 1:	Internet Web Site,
Enabling Output 2:	Reference Handbook – Draft Copy,
Enabling Output 4:	CD-ROM, and
Enabling Output 5:	Reference Handbook – Final Copy.

Although the third enabling output dealt with the intelligent front end, the four enabling outputs relating to the guidebook will be reported on first in this document.

1.4 Intelligent Front End / Expert System

The intelligent front end sub-project was 55 man days in duration and completed as a whole in the first year of the project. The intention was to produce a prototype system to illustrate the potential of what could be achieved with this technology. The project was completed in the given time, by the given date, and within the given budget.

1.5 Other SIMRAC Outputs

It was initially anticipated that this project would feed directly into GAP414 ‘A reference work of current rock engineering knowledge’. However, it soon became apparent during the course of the two projects that the objectives and goals were so dissimilar that almost nothing could be taken from this project and used in GAP414. In the event, the work in GAP414 relating to numerical modelling became a stand-alone under-taking.

2 Internet Web Site

Enabling Output 1: An interactive web site relating to advice and information regarding the use of numerical modelling for improved mine design of South African gold and platinum mines.

Milestone Date: 09/97
Actual Date: 09/97

2.1 Web Site Development

The Internet World Wide Web Site, or simply web site, was intended to provide a facility for interaction between rock mechanics engineers in the industry and numerical modelling experts at facilities such as CSIR Miningtek and Itasca Africa. The intention was to have a usable web site in place before the majority of engineers in the industry came online. Although an initial skeleton site was created very early on in this project a number of problems were duly encountered. These ranged from actual content, through security issues to hardware problems concerning access times. In addition, some internal confusion arose about the URL address to allocate to this site. As a result, low key development of the site continued, but it was kept offline until these problems could be resolved.

The problems encountered in developing this site resulted from the early exploitation of a new technology that had not been well tried or tested in this field. The goal of providing an interactive online service has proven to be more difficult and has taken longer than anticipated. However, it was not a mistake to undertake this work at this time. The process of transferring the content of the numerical modelling handbook to the web site at the final stages of completion was greatly facilitated by the initial work. The web site address is http://www.simrac.csir.co.za which resides within the CSIR domain.

2.2 Milestone Delivery

An interactive Internet web site was made available online by mid 1997, some time before the scheduled milestone. However, of a number of problems relating to this site the most significant problem was that of content. The site was primarily intended to provide access to aspects of the numerical modelling handbook, however, the handbook content was not completed as a quality document until March 1999. It was decided to keep the web site for internal access only until the content was of sufficient quality. This has been achieved in the first quarter of 1999.

3 Reference Handbook – Draft Copy

Enabling Output 2: Draft copy of the reference handbook, containing the basic structure of the book and as much data as can be compiled to this point. This will be provided for review and consideration of future work.

Milestone Date: 12/97
Actual Date: 12/97

3.1 Numerical Modelling Programs

After discussion with the GAPREAG committee it was decided to limit the numerical modelling programs included in the document to only the most commonly used linear, elastic programs, with one exception. This meant including the four linear elastic programs

• BESOL/MINAP_97	Mining Stress Systems,
• MINSIM	CSIR Division of Mining Technology,
• BESOL/MS	Mining Stress Systems, and
• MAP3D	Mine Modelling Limited

The 2D-continuum plasticity program FLAC from Itasca Consulting Group is also included.

This mutual decision eliminated the necessity to discuss aspects such as discontinuum modelling and dynamic analyses.

3.2 Handbook Layout

At the outset of this project much thought and discussion went into the actual layout and format of the hard-copy guidebook document. It was recognised at an early stage that a document dealing with a cutting edge technology, such as engineering computer modelling, would be likely to be out of date very rapidly. Furthermore, it was recognised that different aspects would progress at different rates and would not evolve synchronously. In order to avoid rapid obsolescence of the document, it was recognised that provision should be made at an early stage for easy updating of the text. Two alternative approaches were considered:

- Infrequent scheduled updates of the document as a whole, and
- Frequent ad-hoc updates of relevant parts of the document

It was decided that the second approach, ad-hoc updates of relevant parts of the document, would provide the greatest benefit to engineers in the field. In order to facilitate future implementation of this process it was necessary to develop a document format that would be flexible in allowing addition to and replacement of individual portions while leaving the remaining sections intact.

If individual sections are to be added or replaced without affecting the remainder of the document, then clearly a fully bound document is not practical. A decision was made to distribute the document in loose-leaf, A4 format. Additionally, it was decided that the document should be comprised of individual sections each with its own isolated page numbering scheme. The only thing that is fixed then is the number and order of each of the sections, but not the content or size.

Nine sections in total were proposed. These sections were headed:

1. An Overview of Numerical Modelling
2. Solid Mechanics
3. Solution Methods
4. Numerical Modelling Programs
5. BESOL/MINAP_97
6. MINSIM-W
7. BESOL/MS
8. Map3D
9. FLAC

3.3 Milestone Delivery

A draft copy of Volume I was distributed to the GAPREAG committee for comment and feedback in January 1998. The deadline for feedback was set by that committee to be the end of March 1998. Miningtek received no official feedback by that date and subsequently requested verbal acceptance of the format of the document at the mid-year progress meeting. This verbal approval was given and minuted.

4 CD-ROM

Enabling Output 4: An organised CD-ROM containing example data files, result files and illustrated examples of the application of certain numerical modelling tools to the solution of South African gold and platinum mine design problems.

Milestone Date: 12/98
Actual Date: 12/98

4.1 CD-ROM Content

The CD-ROM contains a number of applications. The primary application is a copy of the guidebook in electronic Portable Document Format (PDF). This document can be read with Adobe Acrobat 3 Reader, which is freely available from Adobe Systems Incorporated (http://www.adobe.com) at no cost. There are a number of advantages in using PDF for electronic distribution of the document. These include:

- a free document reader from Adobe that includes sophisticated document navigation tools,
- an industry standard format,
- read-only distribution, preventing end users from modifying the document, and
- format independent resolution, prints can be made according to any particular printer's resolution

The electronic document can be easily navigated by means of the Adobe Acrobat Reader interface. At this point, this involves such features as section expansion and contraction, and search facilities.

A limited number of CD-R disks have been written using a standard Yamaha CD-R writer. Although this method (CD-R) is well suited to the fast creation and deployment of a small number of disks in standard CD-ROM readable format it is not well suited to mass production of CD-ROM disks. Actual pressing of CD-ROMs is better suited to the distribution of large numbers of discs. No provision was made for mass distribution of this CD-ROM in the original project proposal. CSIR Miningtek has no facility for such mass production. A number of CD-R 'masters' have been issued to SIMRAC.

4.2 Milestone Delivery

The original test versions of the CD-R were written for the December 1998 deadline, however, this included only a partially complete version of the guidebook as the guidebook delivery date had been rescheduled to March 1999. Final versions of the CD-R could only be produced on completion of the guidebook in March 1999.

5 Reference Handbook – Final Copy

Enabling Output 5: Reference handbook for rock engineering numerical modelling of tabular mines. An overview of relevant programs and an appraisal of the strengths and weaknesses of each regarding the South African gold and platinum mining rock engineering perspective. Examples of the use of numerical modelling for mine design. This will include a discussion of the choice of input parameters and adjustments applied to laboratory measurements for various modelling applications.

Milestone Date: 12/98
Actual Date: 03/99

5.1 Document Version Control

During the course of development of the final copy of the guidebook it became apparent that a number of authors and multimedia developers required access to the same sections of the document. It was important to ensure that all persons were able to access the most current version of any section but that only one person could access a specific section at any specific time. Without actually passing the appropriate files on disk the only way to ensure this could be achieved was by employing a proper file version control system over a distributed network.

It was decided to use Microsoft Visual Source Safe (VSS) running on a Windows NT 4.0 Server controlling a domain of Windows 95/98 and Windows NT workstations. Although VSS is specifically designed for controlling ASCII source files for programming applications it is well suited to version control of any files. The users access the VSS server through individual VSS clients installed on their workstations. VSS controls access to the files, ensuring that only one person has read-write access at any one time while any number of users can simultaneously acquire read-only access.

This type of version control has the added benefit of storing a backup of each file on the server that can be restored to any workstation if a particular workstation file is lost or corrupted. In addition, it provides the facility to recover previous versions of a file if unwanted or incorrect modifications are made and saved.

A powerful version control system, such as Microsoft Visual Source Safe proved indispensable in the management of the production of this document.

5.2 Document Publication

Although all original material was created using Microsoft Word 97 problems with this package prevented its use for final publication of the document. The format chosen for the document proved too complex for MS Word to handle in a stable manner. Formatting features were regularly lost and objects such as diagrams and equations were often corrupted. Eventually a professional desktop publishing (DTP) company was approached to help resolve the problems experienced with Word 97. Ultimately, this company advised using Word 97 for draft only but retyping and recompiling the document as a whole using Corel Ventura, a professional DTP package. This resolved all technical publication problems.

The review process consisted of an initial internal (i.e. to the project team) review that involved both proofreading and technical evaluation. This was followed by a peer review from relevant parties to check technical content. After any changes were made an initial external proofreading was undertaken. Subsequent to this the draft document was submitted to the DTP group for final layout. The final layout was reviewed by the project team. Finally, this was submitted to a professional proofreader. After final changes had been made, individual masters were made of each chapter.

5.3 Content of Volume I

Although nine chapters were originally proposed for Volume I at the draft stage of the project, it became apparent that a further two chapters would have to be added to make the document complete in its current form. The additional two chapters are ‘Modelling Criteria’ and ‘Material Models’. These were inserted between the ‘Solid Mechanics’ and ‘Solution Methods’ chapters. This resulted in a total of five chapters covering the principles of numerical modelling. These are followed by a brief chapter to overview the separate numerical modelling programs covered in this volume. Finally each of the individual numerical modelling programs is covered in its own chapter. The five chapters covering the principles of numerical modelling are briefly described below.

5.3.1 An Overview of Numerical Modelling

This section is intended to provide an overview of both the philosophy and the methodology of numerical modelling relating to rock engineering in the South African gold and platinum industries. It was adapted from a paper presented at a SANGORM symposium on numerical modelling given in 1994 (Lightfoot and Napier, 1994).

The overview given draws on both the authors’ (Lightfoot and Napier) and other authors (Cundall and Starfield, 1988) experiences in numerical modelling in rock engineering. In addition it draws on the experiences of a number of published authors in such fields as electrical engineering, medicine, nature conservation and architecture.

It describes the concept of modelling in general and explains what models are and why they are useful. It goes on to narrow the discussion to modelling in rock engineering only. A brief discussion follows on the type of numerical modelling tools that are available to the rock engineer. Finally, the section closes with a proposed methodology for the use of numerical modelling to solve rock engineering problems.

5.3.2 A Chapter on Solid Mechanics

Numerical modelling in rock engineering is based principally on the fundamental concepts of solid mechanics. This chapter gives an overview of these principles and provides numerous equations that are of value to the numerical modeller. The chapter is fairly heavy on theory and mathematics and will probably not be read by all numerical modellers using the handbook. However, it provides important detail for the modeller who is attempting to extend beyond the superficial. An attempt has been made to group a body of knowledge into a single reference that is otherwise only available in numerous separate sources.

5.3.3 A Chapter on Modelling Criteria

Numerical modelling in engineering design is all about the application of relevant design criteria. The proposed design is tested against prescribed criteria, if it passes the design is deemed suitable, if it fails then the design is not sufficient.

This chapter describes the principles and application of five common design criteria used in conjunction with numerical modelling. These are:

- Rock strength and failure,
- Average Pillar Stress,

- Energy Release Rate,
- Excess Shear Stress, and
- Rock Condition Factor.

5.3.4 A Chapter on Material Models

All of the programs discussed in this volume make use of one or more non-linear material models for modelling such features as backfill and crush pillars. One of the programs goes further and actually incorporates plastic material behaviour. This chapter introduces linear elastic material models, non-linear elastic material models and simple plasticity models. Where necessary relevant equations are provided.

A section is included on downgrading laboratory measurements for use in numerical models.

5.3.5 A Chapter on Solution Methods

The chapter on solution methods describes a number of simple analytical solutions available to the rock mechanics engineer. These can be used for providing initial solutions to problems. The actual equations are given where applicable. In some cases references are given to the text that originally described the techniques. If errors have been identified in such texts the corrections are provided relating directly to the specific texts (i.e. page numbers, etc are given to enable the corrections to be implemented).

The three numerical methods used in rock engineering are described briefly. The solution methods are:

- Boundary Element Method,
- Finite Element Method, and
- Finite Difference Method.

These are not covered in detail, as this would not be possible within the constraints of this project. However, enough information is given to provide a cursory familiarity with each method, such that the engineer can appreciate the advantages and disadvantages of each method.

5.3.6 The Chapters on the Numerical Modelling Programs

There are six chapters covering the specific numerical modelling programs available to the South African gold and platinum mining industries. The first chapter provides a summary overview of all the programs used in South African rock engineering. It then provides an additional limited amount of information concerning the five most commonly used programs, other commercial programs and finally special purpose research programs are listed.

The following five chapters are each dedicated to one of the individual commonly used programs. These programs are:

• BESOL/MINAP_97	boundary element method
• MINSIM-W	boundary element method
• BESOL/MS	boundary element method
• Map3D	boundary element method
• FLAC	finite difference method

A general overview is given of each program, followed by a description of the user interface. There follows a section covering verification problems relevant to the gold and platinum mining industries. This is followed by a section covering specific modelling considerations relevant to that program when solving practical problems. This might include a discussion of any bugs or problems found with the program. It might also highlight specific areas where the program is optimally suited to solving particular problems.

It is anticipated that these chapters covering the specific programs will require frequent changes and additions. Changes will be primarily driven by updates and corrections to the actual programs. Additions will be provided by input from users of the programs both in the practical mine design context and at research institutions.

5.4 Content of Volume II

Volume II includes a number of worked examples illustrating both principles and applications of numerical modelling in the context of the South African gold and platinum mining industries. This volume is far less formal than Volume I. The intention is to help the reader work through specific problem types in a friendly manner.

A chapter is included on mine plan capture, as this is an important and time consuming aspect of numerical modelling in mine design. The most common method of mine plan capture is via a digitising tablet. There are two commonly used programs for digitising mine plans in rock engineering in South Africa. Both programs have their strengths and weaknesses, but both are quite adequate for capturing mine plans if they are used properly. However, it is often the case that the person digitising the mine plan approaches the process in an ill-prepared manner. This chapter describes a step wise process to mine plan digitisation that is designed to eliminate as many problems and mistakes as possible. Sections are included covering details of the actual use of the two common digitising packages.

The first example utilises a number of different numerical modelling programs to solve a single problem. The problem chosen exposes a common misconception found amongst rock engineers and uses qualitative numerical modelling to illustrate why this misconception is incorrect. This particular example describes the application of a methodology for numerical modelling of mine design. Further examples cover the application of specific programs including guidelines on how to select actual input parameters for given situations.

5.5 Milestone Delivery

The original delivery date for the two volumes of the numerical modelling handbook was the end of December 1998. However, due to formatting problems it was agreed with SIMRAC (letter dated 8th December 1998) to postpone this to the end of March 1999. The final launch of the numerical modelling handbook has been scheduled for 14th April 1999.

6 Intelligent Front End / Expert System

Enabling Output 3: A prototype Intelligent Front End (IFE) for a generic practical problem type. This will setup the input requirements to run the most appropriate program while interfacing with the user using general mining terminology.

Milestone Date: 12/97
Actual Date: 12/97

The intention of this part of the project was to investigate the possibility of developing expert systems or interface programs with 'intelligence' to help engineers use numerical modelling programs for mine design. This area of the project represented 55 man-days of work.

The work concentrated on four potential aspects of user interface development for numerical modelling. The first of these was to review the potential for expert system technology to provide tools for decision support when using particular numerical modelling programs. The second was to consider a 'wizard' type approach to create input files for commonly used numerical modelling programs. The third and largest area of work concerned the design and development of code-generic, domain-specific intelligent interfaces to create input files for numerical modelling programs. Finally, some aspects of alternative methods for displaying results were given brief consideration.

Most expert systems are developed using special purpose development shells that produce rule-based program logic. This can be highly efficient for decision support programs where the underlying control of the program stems primarily from the encapsulation of human knowledge that cannot be well represented by computer algorithms. Most successful expert systems have been developed to deal with limited problem, or knowledge, domains, for example, completing income tax returns or fixing automobile engines. The key to keeping the system tractable is to reduce the scope of the domain in which the system must operate.

In the scope of this project a review was made of expert system technology in general. The application of expert systems to mining problems in general and rock engineering in particular was then considered. Finally the development of an expert system for decision support in rock engineering numerical modelling was considered.

An alternative approach to expert systems is the development of interface applications that incorporate a degree of intelligence. Such interfaces are developed without expert system shells but rather with more traditional programming languages such as C/C++, Delphi, or Visual Basic. The intelligent interface described here incorporates expert system techniques such as heuristic logic (in addition to algorithmic logic) and domain specific problem solving. As with expert systems, domain specific problem solving is fundamental to the design of an intelligent interface that can help engineers interact on an understandable level with complex numerical modelling programs.

Finally, a very brief evaluation of interactive post-processing was attempted. The aim of this was to develop a simple 2-dimensional viewing system that would allow design criteria threshold values to be altered on the fly. The intention was to assess whether such systems provide more visual clues to the identification of design problems than simple static visualisation systems.

6.1 An Overview of Expert Systems

Artificial intelligence (AI) techniques and in particular expert system technology has, over recent years, been proposed for a number of applications in the field of rock engineering in the South African gold mining industry. It has been suggested that expert systems might provide useful decision support tools to help with the use of numerical modelling for mine design. However, almost no examples of the actual use of this technology exist in this country. The only way to assess the applicability of expert systems or other artificial intelligence techniques to help address numerical modelling problems that exist in the South African mining industry is to attempt to apply the technology to specific problem domains that exist in the mining industry today.

6.1.1 What are Expert Systems

The primary goal of AI research was for many years the development of computer programs that could in some way think, that is, programs that could solve problems in a way that would be considered intelligent if done by humans (Waterman, 1986).

Throughout the 1960s these researchers concentrated on the quest for programs that could solve any type of problem. Perhaps the best example of this was the General Problem Solver (GPS) developed by Newell, Shaw and Simons from 1957 onwards (Simons, 1984). GPS was the first problem-solving program to separate the knowledge specific to the problems it was set to solve from the approach it took to problem solving in general. Although GPS provided invaluable insights into how humans solve problems and how to make computers emulate them in this field, it also showed that the goal of general problem solving was unobtainable. It was identified that ‘the more classes of problems a single program could handle, the more poorly it seemed to do on any individual problem’ (Waterman, 1986).

In the 1970s AI researchers concentrated their effort on techniques of knowledge representation and search. The problem of knowledge representation requires a solution to capturing the full range of knowledge required for intelligent behaviour in a formal language that is suitable for computer manipulation. Search is a problem solving technique that systematically explores a space of problem states. Problem states may be defined as successive and alternative stages in the problem solving process. New algorithms that could speed the search through complex knowledge spaces were explored. It was realised that if computers were ever to be made to think like humans then efficient search strategies would be essential.

Eventually, in the late 1970s it was realised that the single most important factor within a problem-solving program was the knowledge it possessed relating to the problem to be solved. General problem solving strategies are not sufficient alone to solve real problems. This realisation led directly to the development of problem solving computer programs that contained a great deal of expert knowledge in very narrow problem areas. This type of program is known as an expert system. Expert systems are a very successful approximate solution to the classic problem of programming intelligence into a machine (Giarratano and Riley, 1989). Expert systems are said to be domain specific because any single expert system contains only knowledge from a very specific area of knowledge or problem domain. The important thing about expert systems is that, although the knowledge they contain is limited to a very narrow domain, it is of extremely high quality and it is extensive within the domain. To solve different classes of problems requires the use of more than one expert system. A different system must be used for each knowledge domain for which there is a problem to be solved.

This specialisation to specific fields of expertise is a far cry from the general problem solving approach of the 1960s, but it still relies on the idea that was first encapsulated in GPS of separating knowledge about the domain from the problem solving techniques. In modern expert systems the knowledge is contained in a knowledge base and this is operated upon by an inference engine. It is the job of the inference engine to draw inferences between individual packets of knowledge encoded in the knowledge base. It is the job of a knowledge engineer to encode sufficient quality data into the knowledge base for the inference engine to operate on.

6.1.1.1 Benefits of expert systems

Expert systems are perceived to have a number of advantages over human expertise. The most commonly cited advantages are given by Waterman (1986) and are sited here in Table 6-1. Clearly these advantages could be of benefit to rock engineers using numerical modelling for mine design.

Table 6-1: The benefits of artificial expertise (expert systems) in comparison to human expertise (after Waterman, 1986).

Human Expertise	*Artificial Expertise*
Perishable	Permanent
Difficult to transfer	Easy to transfer
Difficult to document	Easy to document
Unpredictable	Consistent
Expensive	Affordable

6.1.1.2 Characteristics of expert systems

Expert systems are called systems rather than programs because they contain more than one component. An expert system actually consists of the expert system itself and a support system or environment. The support environment helps the user interact effectively with the expert system. It may contain, for example, a rule editor, debugging tools, trace facilities and sophisticated explanation facilities to name but a few components. The expert system itself contains the knowledge base and the inference engine (Figure 6-1).

The term user is used here in a context that is more general than that usually applied to conventional programs. In the case of an expert system the user may be operating in any of the following modes (Rolston, 1988):

- Tester: The user attempts to verify the validity of the system behaviour.
- Tutor: The user provides additional knowledge to the system or modifies knowledge already present in the system.
- Pupil: The user seeks to rapidly develop personal expertise relative to the subject domain by extracting organised, distilled knowledge from the system.
- Customer: The user applies the system’s expertise to a specific real task.

In the case of conventional programs the user adopts only the role of customer.

Waterman (1986) states four main characteristics that expert systems must possess. These are expertise, symbolic reasoning, depth and self-knowledge (Figure 6-2). By expertise is meant that ‘an expert system must perform well, that is, achieve the same levels of performance in the domain of interest that human experts can achieve.’ (Waterman, 1986)

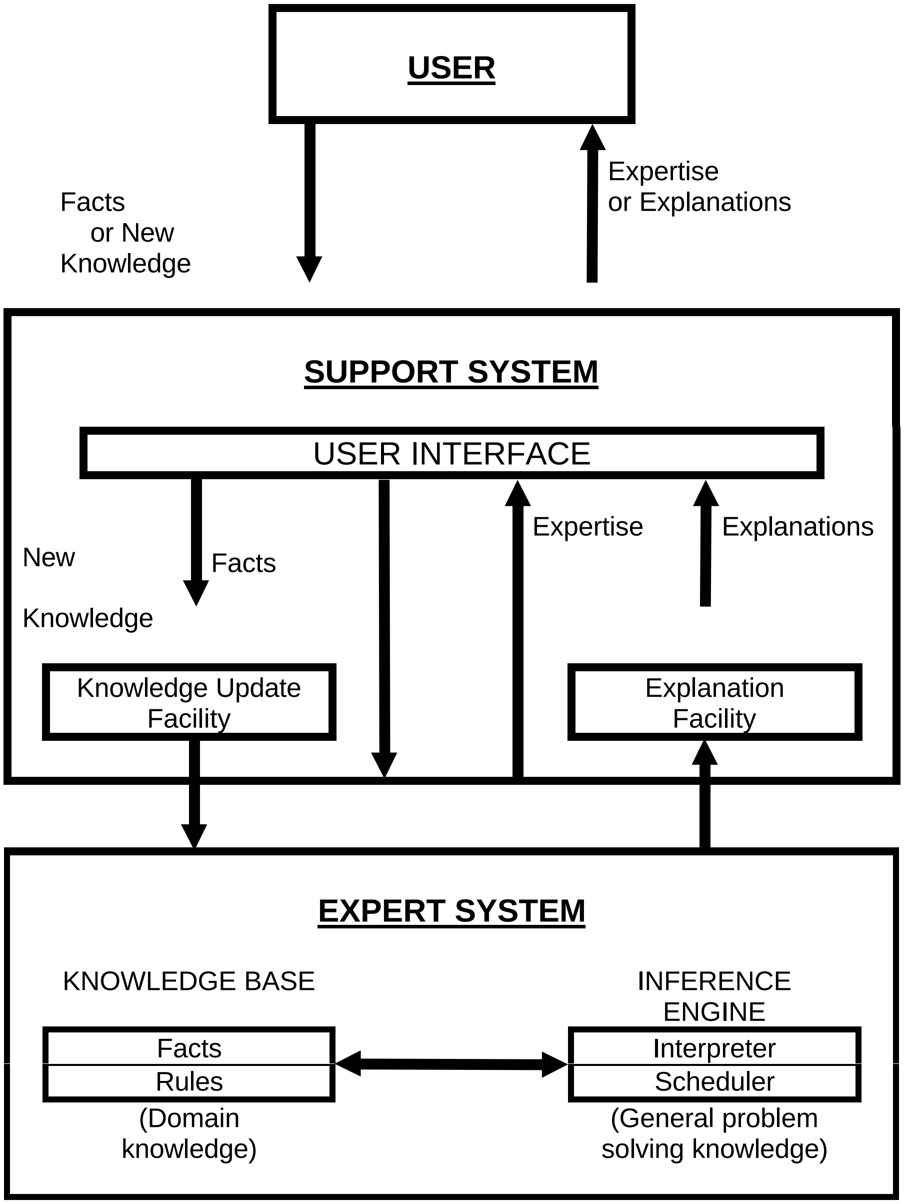


Figure 6-1: The structure of an expert system showing the support system, or environment, and the interaction with the user.

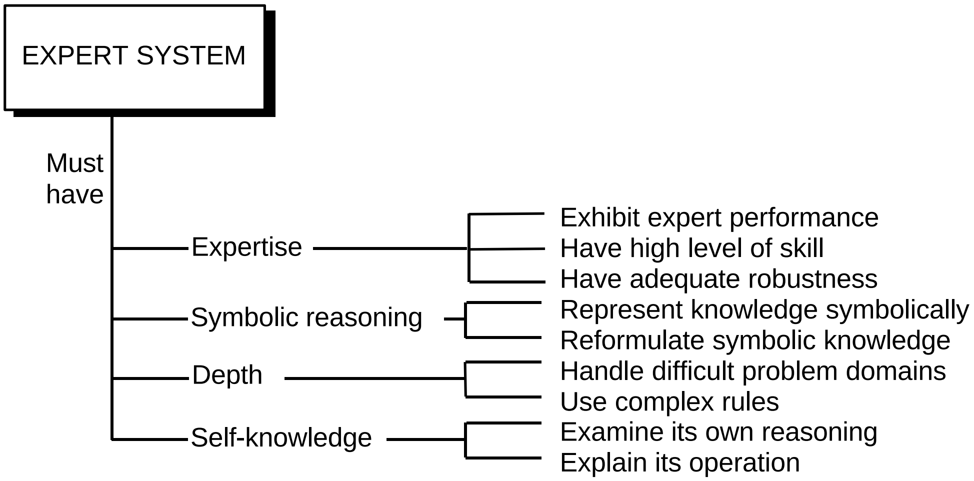


Figure 6-2: Characteristics of an expert system that distinguish it from a conventional program (from Waterman, 1986).

6.1.1.3 The knowledge base: knowledge representation

Knowledge can be represented formally in one of two ways for computer manipulation. Either, it can be encoded as production rules or it can be encoded in the form of graphs or frames (semantic nets and neural networks are sub-classes of frames).

The knowledge encapsulated in an expert system is not exhaustive of all human knowledge in that problem domain. The domain knowledge is a sub-set of all human knowledge and the knowledge base is a sub-set of the domain knowledge (Figure 6-1). This means that expert systems always have the potential to grow or ‘learn’. It also means that expert systems are not privy to all knowledge, but few experts in any domain know or can remember everything there is to know about their field of expertise.

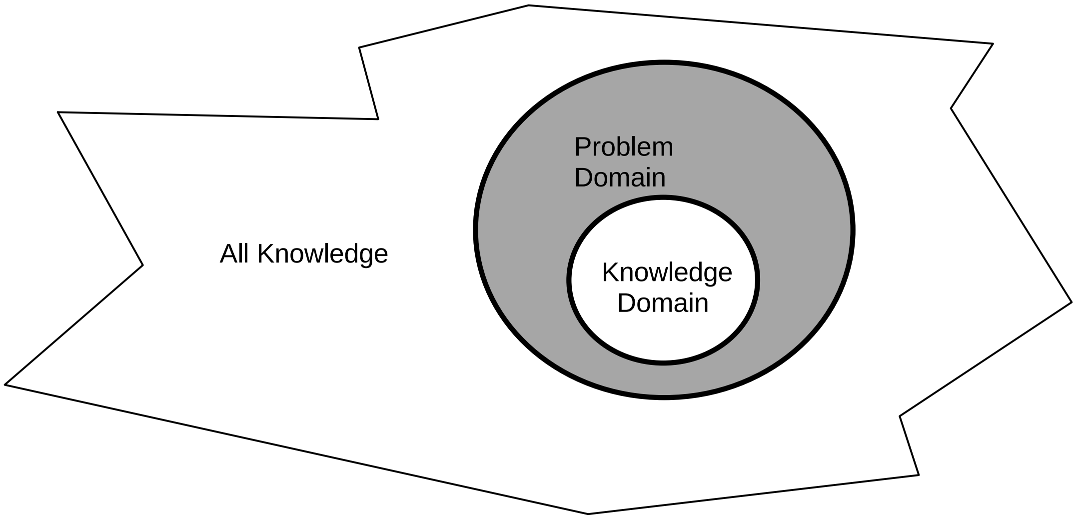


Figure 6-1: The relationship between global knowledge, the knowledge in the problem domain and the knowledge contained in the expert system - the knowledge domain.

6.1.1.4 The inference engine: knowledge manipulation

The inference engine is the domain independent part of the expert system that is responsible for manipulating the domain knowledge based on the specified domain rules in such a way that new knowledge can be inferred from the existing knowledge. This can be best explained by means of a simple example. Consider the example of three generations of a family, one production rule and two statements of knowledge.

Rule:
- The father of a son's father is the son's grandfather.

Knowledge:
1. Brian is John's father
2. Peter is John's son

Inference:
- Brian is Peter's grandfather.

In this case no direct link is specified between Brian and Peter, however from one single rule and two statements of knowledge it is possible to unambiguously infer the appropriate relationship. This is a simple example of what an expert system inference engine does.

There are two important aspects regarding the way the inference engine manipulates a rule based knowledge base to solve a problem. The first concerns the method used to achieve the goals necessary to solve the problem and the second involves the path it uses to scan the knowledge base. There are two alternate strategies for each of these aspects. In the case of goal solving the inference engine can use either forward or backward chaining. When scanning through the rule base it can use either top-down or breadth-first search strategies. The most common implementation in modern production rule systems is to use backward chaining with a depth-first search strategy.

6.1.1.5 Languages and shells

In the field of modern expert system construction there are three possible types of tools for development. Firstly there are the third generation procedural languages (e.g. C, Fortran, Cobol, etc), secondly there are the fourth generation AI languages (e.g. LISP, and its variants, and PROLOG) or there are expert system shells.

When it comes to choosing between a procedural or AI language and an expert system shell to build commercial expert systems there are many advantages to choosing the route of expert system shell. These include improved productivity, ease of developing and maintaining the system, ease of portability between shells and environments, and the vendor support and training that is available (Carrico, et al, 1989). Carrico, et al, (1989) go on to strongly recommend the choice of a rule-based shell whenever possible.

6.1.2 Expert Systems in Mining and Rock Engineering

The first expert system introduced to any part of the international mining industry was Prospector (Waterman, 1986). Prospector was developed by Stanford Research Institute between 1974 and 1983 to aid exploration geologists in their search for ore deposits.

More recently expert systems have been developed for mining applications such as fragmentation system design (Paul and Gershon, 1988) and mining method selection (Bandopadhyay and Venkatasubramanian, 1988).

In the field of rock engineering expert systems have been developed to operate in the fields of coal mining strata control (Ozan, 1990) rock mass classification (Butler and Franklin, 1990) and choice of stress analysis programs for excavation design (Coulthard and Ciesielski, 1991).

Dershowitz and Einstein (1984) suggest applications for expert systems in such varied rock engineering fields as the analysis of rock wedge stability, rock fracture flow, large strain discontinuum mechanics and underground construction.

Johnson (1988) discusses the application of expert systems to aid in the more appropriate use of the COMRO (Chamber of Mines Research Organisation, now the Division of Mining Technology, CSIR) computer program MINSIM. He suggests that there is considerable misuse of MINSIM in the practical mining environment due to a skills shortage in the industry. A possible solution to this problem is to use expert system technology to help rock engineers make the most of the MINSIM suite of programs.

Cichowicz (1993) describes a PC based expert system written to provide risk assessment when mining in potentially rockburst prone ground. Webber (1996) describe another expert system called RockRisk that was also developed for rockburst risk assessment. This system is based on a Microsoft Windows platform.

6.1.3 An Example Expert System

In order to assess the applicability of expert system technology to numerical modelling in rock engineering it was necessary to consider a specific case study. This case study involved selecting a rock engineering mine design problem domain. Once this was done a practical development tool was selected and finally a prototype expert system was developed. Finally, a limited appraisal was made of the effectiveness and efficiency of the use of this system in this domain.

6.1.3.1 The problem domain

Initially two proposals for relevant domains were investigated. The first involved decision support for the design of tunnel support and the second covered decision support for use of the numerical modelling program MINSIM. It was decided to pursue the second proposal as it involved less on the fly analysis and more heuristic principles.

Hence, the domain of expertise the system must address can be defined as a decision support tool to help persons decide whether MINSIM is a useful tool for solving some particular problem they have in mine design. As far as the knowledge engineer is concerned, the only knowledge he has of the problem the user will specify is that it is related to some rock mechanics aspect of mine design or diagnostics involving the MINSIM program.

6.1.3.2 Identification of the user

As much as an expert system is domain specific, it is specific to a certain class of user. It is essential to decide on the class of user that must be addressed before any work can begin on expert system development. To avoid added complexity and the need for additional areas of research it was decided to address the lowest possible level of competence and experience in a professional rock mechanics practitioner working in the South African gold mining industry.

The expert system was aimed at strata control officers with little or no experience of numerical modelling. This category of user allows the knowledge engineer to assume that he is at least familiar with the mining environment and the jargon of both the industry and of rock mechanics. However, it avoids any necessity to get involved with detailed background information both in numerical modelling, in general, and boundary element methods, in particular. Such a user can be expected to be concerned with how MINSIM can be used to solve the current problem and not the specifics of how it will go about solving it.

6.1.3.3 The expert system development tool

In main stream expert system development the choice of development tool is generally driven by the nature of the knowledge that is to be encapsulated in the knowledge base. However, other considerations such as time and budget are also important. The constraints applied to this project were, time, primarily and in addition, it was necessary for the development software to be able to run on a desktop personal computer

Within the bounds outlined a series of options were available for the project. Artificial Neural Nets were not even considered, as they are best suited to domains that contain abundant empirical data (Giarratano and Riley, 1989). Each main class of development tool was represented by at least one option. The options available are given in Table 6-2.

Table 6-2: Available expert system development tools.

Procedural Language	Microsoft Fortan Microsoft Visual Basic ANSI C
Declarative Language	Borland Turbo PROLOG PROLOG Development Corporation PROLOG
Expert System Language	Clips
Multimedia Authoring Sytsem	Asymmetrix Toolbook
Expert System Shell	EXSYS Expert System Shell VP- Expert Rule-Based Expert System Development Tool

All of the tools considered were immediately available hence the choice was made on available time and greatest potential for assessing actual expert system technology. That is, which tool would allow the greatest amount of progress on the prototype within a short period of time, and would also provide the best insight into the fundamentals of expert system technology.

Familiarity with the three procedural languages sited suggested that an approach using one of these would involve the shortest learning curve. However, third generation procedural languages are poorly suited to the development of expert systems, and do not provide any direct insights into expert system development technologies. Furthermore hand coding makes subsequent modifications to and debugging of a knowledge base difficult (Buchanon *et al*, 1983). All procedural languages were ruled out as a tool of choice.

Some experimentation was undertaken with the declarative language PROLOG. The Borland Turbo Prolog compiler was used initially, but this was subsequently replaced with the updated Prolog Development Corporations Prolog compiler. A simple experimental expert system was built using a series of predicates and rules that would be operated on by the Prolog inference engine. During this experimental stage it became increasingly clear that although Prolog provides a powerful method of developing an expert system there are significant overheads in its use.

The expert system language that was considered briefly was CLIPS (Giarratano and Riley, 1989). CLIPS is an expert system language that was developed at the NASA/Johnson Space Centre using the C programming language. The name is an acronym for C Language Integrated Production System. It is a forward chaining rule-based language which is related to OPS5 and ART. It became apparent that this course was somewhat out of the main stream of expert system development and there did not appear to be significant benefits to using this tool to achieve the specified goals of the expert system prototype.

A number of expert system shells were considered as viable programming tools for development of the expert system. A public domain version of the EXSYS expert system development shell was considered. There appeared to be some confusion with the programs provided free by Public Brand Software for this system as not all the programs appeared to be present on the diskettes. This option was not pursued any further.

The VP-Expert expert system shell quickly proved to be easy to learn and was capable of providing all that was required for the project including direct assessment of a specific expert system development tool. Although the product is MS-DOS based it provides moderate graphics capabilities. The expert system is rule-based and is shipped with its own inference engine and development environment. It quickly became apparent that VP-Expert would be the best development tool to achieve the goals of the project in the time allocated.

6.1.4 The Prototype Expert System

Reference has been made in the past to the use of expert systems to help in the use of numerical modelling in rock engineering in South Africa (e.g. Johnson, 1988). However, the coverage has been purely speculative and no such systems have ever been developed specifically for the field. Buchanan et al (1983) identify lack of early system programming as a common problem in expert system development. They suggest that the knowledge engineer begin programming the system immediately he has sufficient understanding of the domain that is to be tackled. A prototype system is an excellent means of judging the applicability of early assumptions. It is also a good way to demonstrate the potential of expert systems to those not familiar with the discipline of artificial intelligence.

The name XMIN is short for Expert MINSIM User. The purpose of XMIN was to evaluate the benefits of an expert system in terms of both decision support (DS) and computer aided instruction (CAI). The system was built using the VP Expert expert system shell and a series of Microsoft FORTRAN text and graphics routines that are launched by the shell. Knowledge bases were developed in the expert system shell to handle the non-deterministic aspects of the system. The knowledge bases consist of sets of rules that are operated on by the system shell's inference engine.

The goal of an individual XMIN session is to determine if a particular type of rock mechanics problem can be addressed using the MINSIM suite of programs. The user is asked questions and the system follows an operational path dependent on the response. Mostly, the response can only consist of a yes or no answer or some multiple-choice option. At certain points in the system path short lessons are called that inform the user of certain rock engineering principles. At these points no response is required from the user who is simply being lectured. This is where the application of CAI. is most prevalent. However, as the system makes no attempt to enter into any form of interaction with the user it is clear that its functionality in this area is extremely primitive. In fact it is arguable that the monologue approach to training adopted in XMIN cannot be called CAI. it may be better to define an alternative term such as computer aided training (CAT.). It is clear that this CAT. approach is very much restricted when compared to CAI or ICAI, however, to improve on the approach adopted in XMIN would require a considerable amount of additional work that is beyond the scope of this project.

The experimentation undertaken with CAT in the development of XMIN and the literature pertaining to CAI and ICAI certainly indicate that there is a considerable amount of scope for the development of such training techniques for use in the South African mining industry.

In the process of developing an expert system to be an expert MINSIM user it is necessary to fully understand exactly what MINSIM is. Once this has been achieved it is possible to build a model of the MINSIM program that can be incorporated in the expert system. This model should contain only elements relevant to the development and use of the expert system, but it is essential that no facts of relevance be omitted.

In addition to MINSIM, the expert system is concerned with solving practical problems that occur in the field of rock engineering on a production gold mine. To address this issue, all the types of problems that can occur, and the ones relevant to analysis with MINSIM, must be identified. To identify all classes of rock engineering problems that can occur on a deep level gold mine it is necessary to fully understand all aspects of a deep level gold mine. Again, in order to achieve this, it is necessary to build a model of a mine that incorporates all elements that are relevant to rock engineering. This model can then be used to develop smaller, more refined, models that are applicable to the domain of the expert system.

6.1.5 Use of XMIN for Decision Support

XMIN is designed specifically to provide a decision on whether the MINSIM program should be used to help solve a specific mining problem. It will make one of two decisions for each consultation: yes or no. In the course of a consultation it uses scripts to provide advice and instruction to the user in how best to use MINSIM to help address the problem at hand.

In all but the most rudimentary areas of theoretical rock engineering and modelling methodology XMIN is very weak at providing decision support and expert advice. The reason for this is that the expert advice that must be programmed into the system knowledge base does not exist. Where expertise does exist it is often contentious or simply non-deterministic. This is a problem that must occur in any young science where the design expertise that is required is still at the cutting edge of research. In a number of years time research may provide the expertise, only then will it be possible to use the techniques of knowledge engineering to dissipate expert knowledge throughout the rock engineering community.

If XMIN fails as a significant decision support tool how does it stand-up in the field of computer aided training? The XMIN scripts provide no interaction as in CAI and there is no attempt at all to improve the programs student model through mixed initiative dialogue as is the norm in ICAI. In fact the XMIN scripts are more an example of primitive ad-hoc frame oriented computer teaching that Carbonell (1970) criticised as being able to do 'little more than what a programmed textbook can do'. In practice this means that these scripts may provide a limited amount of instruction but they are likely to be inadequate and it is unlikely that even a novice rock mechanics engineer would wish to consult with them more than once.

The simple training by what-if modelling that is employed in the Mohr-Coulomb training script in the interactive module MC2D incorporated as a part of XMIN may provide more promising avenues of development, but to make this effective requires a considerable amount of additional work. The module MC2D needs to be equipped with at least simple query capabilities to initiate some degree of student teacher dialogue. It is possible that a relatively simple context sensitive help system with hypertext capabilities could be used in this regard. This would not require excessive coding and could be used as a template for other such modules. Certainly, the simple what-if modelling capabilities provided by MC2D are a valuable way of awakening user awareness to aspects of more complex models such as MINSIM.

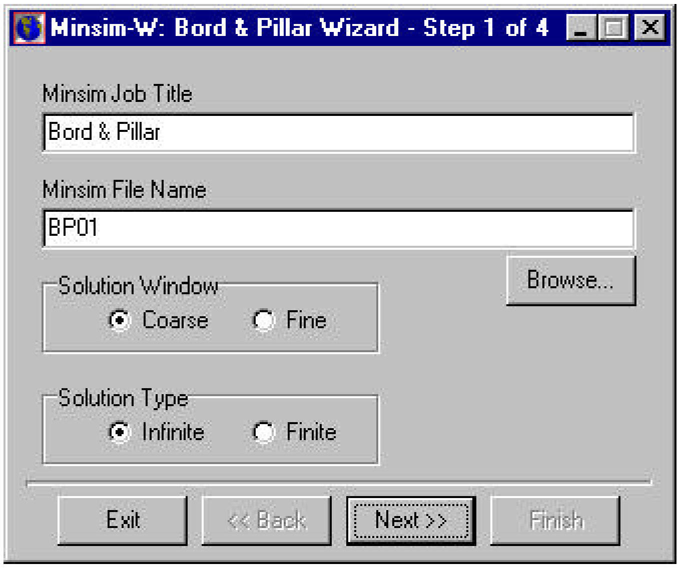
From this discussion it can be concluded that XMIN is fairly limited when it comes to decision support and is unlikely to be put to more than infrequent use in a mining environment. This is not to say that it cannot be used, but highlights the potential difficulties involved in developing an expert system or series of expert systems that could provide more than very rudimentary decision support in the field of rock engineering. Furthermore, the XMIN training scripts are fairly inadequate at providing the training that was proposed at the outset of this project. Certainly, with more development, many of the inadequacies could be resolved, but there are problems of efficiency. Is it wise to pursue the expert system and ICAI routes with all the necessary work involved to resolve these problems, or is there a more direct and immediately implementable approach to the problem?

6.2 Modeller Wizards

The idea of modeller wizards was inspired by the type of wizards available in many aspects of the Microsoft Windows 95 system and applications. Essentially, wizards comprise of a series of sequential windows that ‘walk’ the user through some process in a systematic way. It was proposed that such an approach might be of benefit in setting up specific numerical simulations.

For the sake of experimentation with the modeller wizard concept a very simple regular geometry was chosen and the target program was MINSIM. The specific geometry is a single reef, bord and pillar layout. The construction of such layouts using the current MINSIM pre-processor is extremely tedious, despite the fact that the number of parameters actually required is very limited. The reason for the current level of difficulty is that the whole philosophy of creating MINSIM input relies upon a drawing (usually digitised) of the mine plan being submitted to the solution ‘engine’ of the MINSIM program. In the case of ideal regular pillar layouts, it is generally easier to describe the parameters in words and numbers than it is to have to draw each and every pillar.

A simple modeller wizard prototype was developed in Borland C++ Builder Version 1.00. The Minsim Bord & Pillar Wizard comprises 4 windows that each address a different aspect of the problem being considered. The first sets up the main job parameters, such as job title and solution type (Figure 6-1). The second window describes the reef or seam orientation and position (**Error! Reference source not found.**).The third describes the pillar geometry in terms of size and spacing (**Error! Reference source not found.**). Finally, the fourth window describes the rock mass characteristics (Figure 6-4). These are the only parameters required to create a Minsim input file to run this particular type of problem.



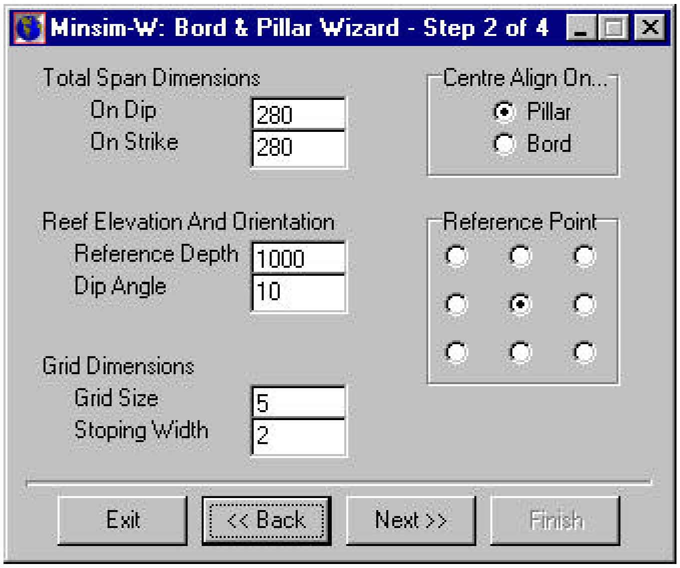
Minsim Job Title
Bord & Pillar

Minsim File Name
BP01

Solution Window
● Coarse ○ Fine

Solution Type
● Infinite ○ Finite

Figure 6-1: The first window of the MINSIM Bord and Pillar Wizard designed to accept general problem parameters.



Total Span Dimensions
On Dip 280
On Strike 280

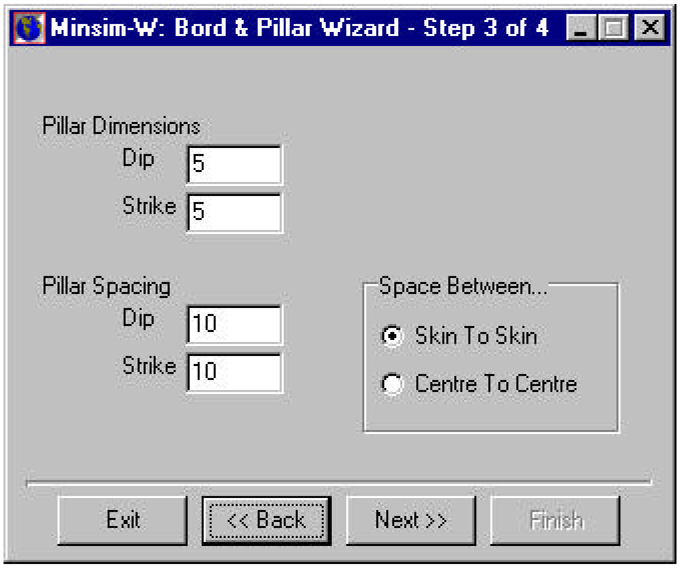
Centre Align On...
● Pillar
○ Bord

Reef Elevation And Orientation
Reference Depth 1000
Dip Angle 10

Reference Point
○ ○ ○
○ ● ○
○ ○ ○

Grid Dimensions
Grid Size 5
Stoping Width 2

Figure 6-2: The second window of the MINSIM Bord and Pillar Wizard limited to the actual reef geometry and the overlaying computational grid.



Pillar Dimensions
Dip 5
Strike 5

Pillar Spacing
Dip 10
Strike 10

Space Between...
● Skin To Skin
○ Centre To Centre

Figure 6-3: The third window of the MINSIM Bord and Pillar Wizard dealing only with parameters actually relating to the pillars.

Young's Modulus 50
Poisson's Ratio .2
Rock density 2700
K Ratio 0.6

Figure 6-4: The final window of the MINSIM Bord and Pillar Wizard isolated to rock mass properties.

The modeller wizard approach does show good potential for the type of problem that was considered here, namely a regular geometry that can be described with a limited number of input parameters. This approach should be considered for implementation in future numerical modelling pre-processor interfaces for use with certain problem types.

6.3 Intelligent Interfaces

Arguably, any engineering design process can be sub-divided into a set of discrete problem types or domains. This is true of mine design where the broad problem domains consist of such aspects as tunnel layouts, stope layouts, support design, etc.

These broad domains can themselves be further split into even more specific problem domains such as tunnel roof bolt design or over-stoping of tunnels. If the overall evolutionary process of excavation design throughout the life of a mine can be routinely classified into such domains then it is possible to provide a toolkit of modules to help the engineer analyse each of these problem domains as and when necessary. In this case the toolkit comprises a set of domain specific user interfaces for generating input definitions for numerical modelling programs. These domain specific interfaces must minimize the amount of input required from the engineer while producing useful models for the analysis of mine design. Such domain specific interfaces have been called intelligent interfaces.

The goal of the intelligent interface is to simplify the numerical modelling process for the design engineer. The engineer is confronted with modelling the actual problem in question whilst being shielded from the precise workings of the numerical modelling program. For example, while the rock mechanics engineer may be free to control the tunnel shape and size, the rock type and the type and length of roof bolt, he does not have to design a finite difference, finite element or boundary element mesh. It should not be necessary to have to provide such parameters as damping factors or boundary conditions. Indeed, it is unnecessary for the engineer to even have to know which particular numerical analysis program is used for the actual simulation.

In addition to confining the engineer to the world with which he is familiar the intelligent interface can also help to ensure consistency in modelling. By presenting the engineer with a choice of materials such as rock, backfill or roof bolt type by name, rather than requiring him to enter the detailed parameters that define the material behaviour according to some constitutive model, the modelling process becomes less prone to error. Furthermore, this approach would allow an experienced rock engineer to prescribe an initial model that could then be given to a less experienced, more junior, engineer for sensitivity analysis. In other words, the system could help enforce safe delegation of the workload.

Consider an example of such a delegation process where the senior engineer could define and lock all parameters apart from the roof bolt length and spacing. The junior engineer could then perform a sensitivity analysis on these parameters where he need only enter two different (and well defined) input fields. At the conclusion of the analysis, the senior engineer can be certain that no other parameters were accidentally modified or the wrong units used for one or more of the parameters.

Note: In this section, the term ‘application’ is used to refer to the SIMRAC Intelligent Interface program. The term ‘program’ is used to refer to the target numerical modelling programs, such as FLAC, UDEC, etc. The term ‘code’ is used for the programmed instructions used to create the SIMRAC Intelligent Interface.

6.3.1 Requirements for an Intelligent Interface

At the outset of development of the intelligent interface prototype a number of basic requirements were specified. These were:

- access through a simple graphical user interface that uses general mining terminology rather than terminology specific to a particular program,
- the interface must not produce invalid models,
- it must be possible to acquire complete consistency and repeatability between models,
- the application must be code generic, i.e. it must not be designed for a single, specific program, but must be able to provide input for a number of different programs,

- complete flexibility to incorporate, new domain definitions with relevant user interfaces, new mine objects, additional numerical modelling programs, and
- easy maintenance by developers other than the original authors.

In order to achieve these objectives it was necessary to undertake a careful design process that laid out a structured approach to the application development.

6.3.2 Intelligent Interface Design

A number of implementation approaches were considered prior to the actual intelligent interface prototype development. The first involved developing a script language that could be interpreted by a meta-interface, and the second involved using object oriented programming to develop flexible, compilable modules. The first approach has the advantage that new domains can be added without having to recompile the entire program. The second approach has the advantage of being quicker and easier to implement.

Ultimately, the program was designed as a series of modules that allows for the addition, removal and modification of modules, at any time with a minimum of disruption to the rest of the application code. It was necessary to identify where pre-defined databases of parameter values could be substituted for direct user input in order to simplify and add consistency to the modelling process. Finally, it was recognised that, if the interface is to provide a high degree of robustness when creating actual models it is necessary to perform a large amount of error checking directly at user input.

6.3.2.1 The script language approach

The idea behind the script language approach was to develop a script language that could be used to create the user interface on the fly. The language would be designed to allow for the manipulation of mining objects such as tunnels, stopes, support and geological features. The script file would be read through a parser by the intelligent interface application at initialization time. It would use the information contained in the script to create the user interface for any given problem domain. This facilitates a large degree of flexibility within the system, allowing for the creation of new user interfaces or the adaptation of existing ones, without the necessity for recompiling any code. Any person familiar with the script language could create an ASCII script file describing the basics of a problem domain. This would allow a person familiar with the script language to create and add new problem domains by writing a script file using an ASCII text editor, such as Windows Notepad, without having to alter the application code.

The main disadvantage of the script language approach is the amount of time it would take to implement. In addition, it is still necessary to develop all of the underlying mining objects such that they can be created from the script by the application. The best way to create these objects would again be by using an object oriented programming approach. Hence, that particular phase of work would have to be undertaken regardless of whether the script language was implemented or not. After careful consideration, it became apparent that the time required to develop the script language and implement the parser was beyond the scope of this small sub-project.

6.3.2.2 The object oriented programming approach

In this approach the user interface is hard-coded and then compiled into the final application executable. In order to add new domains it is necessary to write additional code and then recompile the application. This is less flexible than the script language approach, but considerably easier to implement in a short time span. Certain design considerations are necessary to maintain a significant degree of flexibility and to ensure that new problem domains can be added to the application with a minimum of effort.

Code reuse is a significant consideration in the development of this application as it is envisaged that future domains will inherit a great deal of functionality from the code developed for existing domains in the application. The object oriented programming (OOP) paradigm offers considerable benefits over more traditional programming paradigms where code reuse is important. In addition, using OOP it is possible to encapsulate all mining features, such as stopes, tunnels, faults, etc. as individual object classes that have their own properties and methods. These can be defined as collections, making it easy to add new instances of any of the objects as and when required.

The implementation of mining features as objects makes it possible to divorce responsibility for all manipulation of any given object from the main code and to encapsulate it within the object's definition. This includes responsibility for the object to draw itself to the graphical user interface and write its definition to the program-input file. Encapsulation of mining objects in this manner means that new objects can be added with a minimum of disruption to the main application code.

6.3.2.3 Program modules

In order to ensure maximum future flexibility, with a minimum of disruption to existing code, the application design incorporates a large degree of modularization. The modules exist on three primary levels:

- project manager
- domain manager, and
- object library.

The project manager controls the choice of domain and all other application wide considerations, such as which programs are available at the current site. The application has only one project manager. The domain manager controls aspects local to the currently selected domain, such as which mining objects are available. It is also responsible for providing the graphical user interface and processing any events generated by this interface. There exists one domain manager for each problem domain available to the application. The object library makes object definitions available to both the project manager and the domain managers. These definitions exist as class modules that can be used as templates for new instances of objects to be created by any of the 'managers'. The object library contains two types of objects:

- application objects and
- mining objects.

The application objects provide the application with functionality such as file and registry access. The mining objects encapsulate the actual mining features such as stopes, tunnels, faults, etc.

In order to add a new domain at some future time it is only necessary to add code for a new domain manager with the accompanying user interface. New mining features can be added by creating new class definitions. This need be done only once for each new feature to be added. Generating a new instance of the object from the class definition can easily create multiple features of any given type in any given problem domain.

6.3.2.4 Material property database

All numerical analysis programs require input parameters to describe rock mass material behaviour such as elastic constants describing a linear elastic rock mass or a detailed stress-strain table describing complex strain softening behaviour. In programs that can accommodate an inhomogeneous rock mass it may be necessary to supply multiple sets of rock mass material parameters. In addition, it is often necessary to supply material parameters for supports such as rock bolts, timber packs or elongate props.

In the case of a linear, isotropic, elastic, continuum rock mass there are few problems in deciding on values to use for the elastic constants. However, in the case of a rock mass represented by a plasticity constitutive model it can be much more difficult to determine appropriate values. The more complicated the constitutive model, generally, the more difficult this becomes. In addition to the problem of determining appropriate values, even in the linear elastic case, it can be difficult to ensure consistency in the choices made between different engineers modelling the same rock mass. There can be consistency problems when the same engineer models the same rock mass at different times. Such difficulties in the choice of appropriate values for material parameters can become further exacerbated in the case of complex support units composed of more than one material such as grouted rock bolts.

Introducing the concept of a material property database can alleviate the problem of choosing appropriate material properties for both the rock mass and the support units. In this case, the user is presented with a choice from a list of pre-defined rock types or support units. The entry in the list acts as an index into the appropriate database from where the application can transparently acquire the exact values required for parameters necessary to describe the material in question. This avoids the necessity for the site engineer to have to define complex input and it ensures consistency in choice of values across the whole user base.

The rock mass material property database could be extended to incorporate geotechnical areas. Based on the geotechnical area selected and some other factors such as the depth specified, the application could, transparently, select the appropriate rock type, or types, and then access the correct material properties.

6.3.2.5 Error trapping

In order to ensure maximum stability of the intelligent interface when in use it is necessary to incorporate a large amount of error trapping code into the application. Potential errors can arise from two main sources:

- bad input, and
- system problems.

Bad input might be something like entering a negative Young’s Modulus or a non-integer value for the number of mining steps. System problems include such things as attempting to write an output file when the disk is full, or trying to read a non-existent input file.

It is possible to trap a large amount of bad input at the user interface level. A high level of success can be achieved by testing the user input on two separate levels. Firstly the keyboard input is restricted to only valid keystrokes, for example letters cannot be typed in numeric input fields and decimal points cannot be typed in numeric integer fields. On the second level, specific variables can be tested for validity, for example friction angles can be rejected if they are negative or are given as 90° or more (bearing in mind that 90° is not an allowable friction angle).

In order to help with correctness of input the intelligent interface application has been designed to reside in a continual state of valid input. This means that all fields of the user interface consistently contain values that are valid in their own right, and in terms of the global model that is to be created. To achieve this, the application must start up with all fields containing values that can create a valid model for analysis. A program file could be created without altering any of the start-up input variables. The question arises as to what model to start the application with. The first time the application is run for any specific domain there are no clues as to what model the user might be interested in solving. In this case default values are used that have been hardwired into the applications executable code. However, for all future use of the application it seems appropriate to start with the last model specified by the user. The application has been designed to 'remember' the values that were in the input fields when the application was last terminated.

6.3.3 Implementation of the Intelligent Interface

A prototype intelligent interface has been developed to demonstrate the principles involved in such an application. The prototype is not intended to be fully functional as the implementation of full functionality is beyond the scope of the limited resources of this project.

The current prototype contains a number of components. These components include a project manager, a partially functional user interface and a domain manager for a single problem domain, a second non-functional user interface and domain manager for a different domain and a number of application objects and mining objects. These components are described in more detail below.

The domain chosen to demonstrate the principles of the intelligent interface is the situation of mining in the vicinity of a seismically active fault. This problem domain was chosen for its simplicity: only three mining objects are required, these being two stopes and a fault. This circumvented the necessity to implement object collections at this stage, as the number of each object required is known at the time of compilation.

6.3.3.1 Development tools

The original intention was always to use the 32 bit version of Microsoft Windows as implemented in the Microsoft Windows 95 and Windows NT operating systems as the platform of choice for the development of the intelligent interface. These platforms provide a number of development tools that include systems known as RAD, or Rapid Application Development, tools. The time constraint on this project meant that it would be necessary to use one of the available RAD tools. The main RAD tools available for these operating systems include such products as Microsoft Visual Basic, Borland Delphi and Borland C++ Builder amongst others. Microsoft Visual Basic 5.00 was chosen as the development tool.

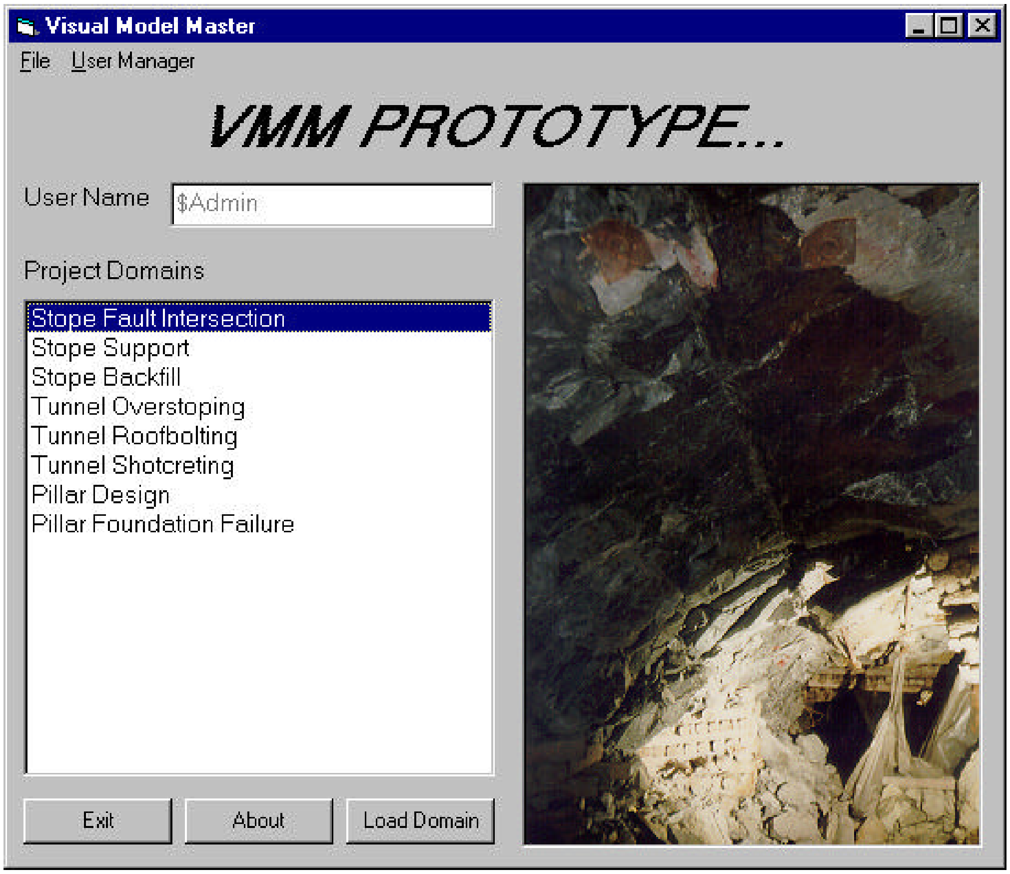
6.3.3.2 Coding conventions

In order to facilitate future development by persons other than the original authors it was necessary to decide upon certain coding conventions for the naming of variables and object classes. It seemed prudent to adopt existing conventions that can be found in the Microsoft Visual Basic 5.00 Programmers Guide, Appendix B. The naming conventions are adhered to for all variables that extend beyond the scope of an individual procedure (i.e. sub-routine or function).

6.3.3.3 The graphical user interface (GUI)

The Windows GUI is implemented through the use of Microsoft Visual Basic (VB) Forms. These are essentially object classes with a user interface that can be built using a visual toolbox of controls provided by the VB development environment. For the prototype a number of independent forms were developed in addition to a form template that can be reused for creating the domain manager user interface (DMUI) main form. The DMUI is copied for each new instance of a domain that is added to the application. Relevant controls and code are added to the form to provide the necessary functionality for each separate domain. At this stage only one complete DMUI has been developed although a second 'blank' instance has been linked into the prototype for the purposes of providing an example of how this is implemented. Additional file input/output (IO) control windows are provided through the system common dialogue control that forms part of the Microsoft Win32 API. Microsoft Windows help functionality is provided by calls to the Win32 API WinHelp function. In addition, Win32 API functionality is used to read system information that could be useful for debugging and maintenance purposes after code distribution. In all of these cases the Microsoft Win32 API is responsible for providing relevant dialogue windows. The most important of the windows developed specifically for the prototype are described below.

A simple Login window controls user access to the application. This is consistent with the concept of dividing users into the two categories of 1) Administrator and 2) Normal. At application load it is necessary to distinguish into which of these categories a particular user falls. If an administrator has not created a user by name then that user cannot login to the system.



User Name $Admin

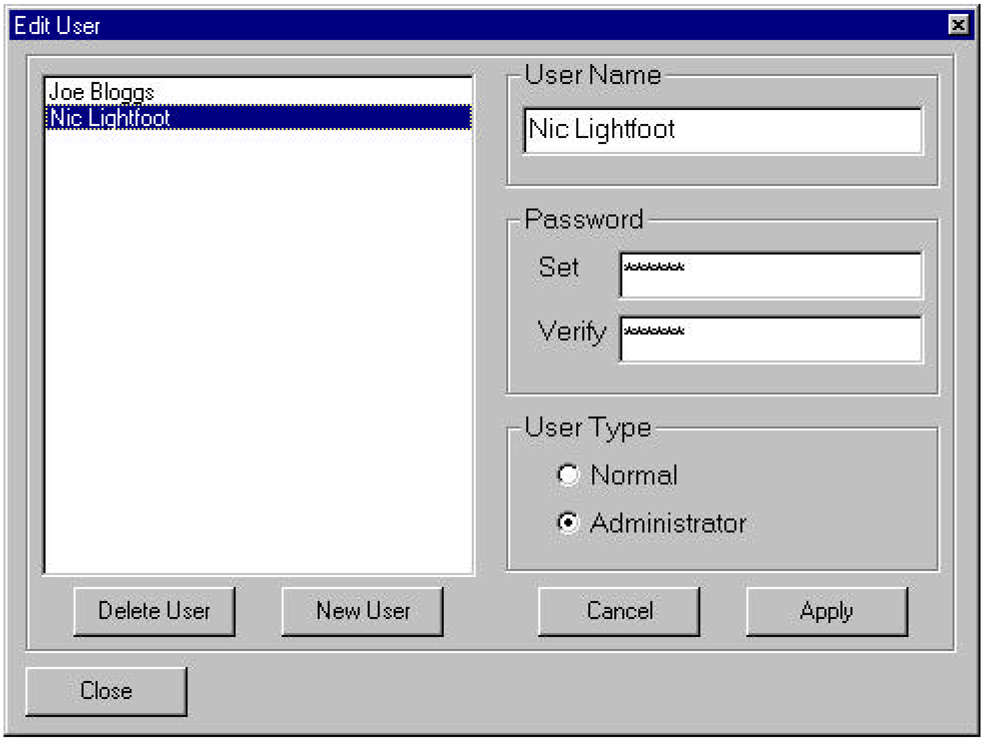
Project Domains

- Stope Fault Intersection
- Stope Support
- Stope Backfill
- Tunnel Overstoping
- Tunnel Roofbolting
- Tunnel Shotcreting
- Pillar Design
- Pillar Foundation Failure

Figure 6-1: The Project Manager Window.

The Project Manager window is usually the first window presented to the user by the application after a successful login. The Project Manager window (Figure 6-1) controls which domain will be loaded for the user. It also gives an administrator access to the user manager.

The Edit User window (Figure 6-2) provides an administrator with the ability to edit a previously defined user's profile, delete a previously defined user or create a new user. When editing a current user the user name is selected in the left hand list box and the profile is then presented by the application in the text boxes to the right of the list box. Altering the details in any of the text boxes alters the profile for the selected user. The delete button will delete the currently selected user unless this is the last user in the list. The User Manager does not allow the deletion of the last user as this would prevent future logins on the current computer. In order to ensure full system functionality in the event of only one user profile being available the system automatically updates this user profile to have administration privileges if that was not already the case. The Add User button opens the Add User window which simply provides a form that replicates the right hand side of the edit form and allows for the addition of new users to the user list.



Joe Bloggs
Nic Lightfoot

User Name: Nic Lightfoot

Password
Set: ********
Verify: ********

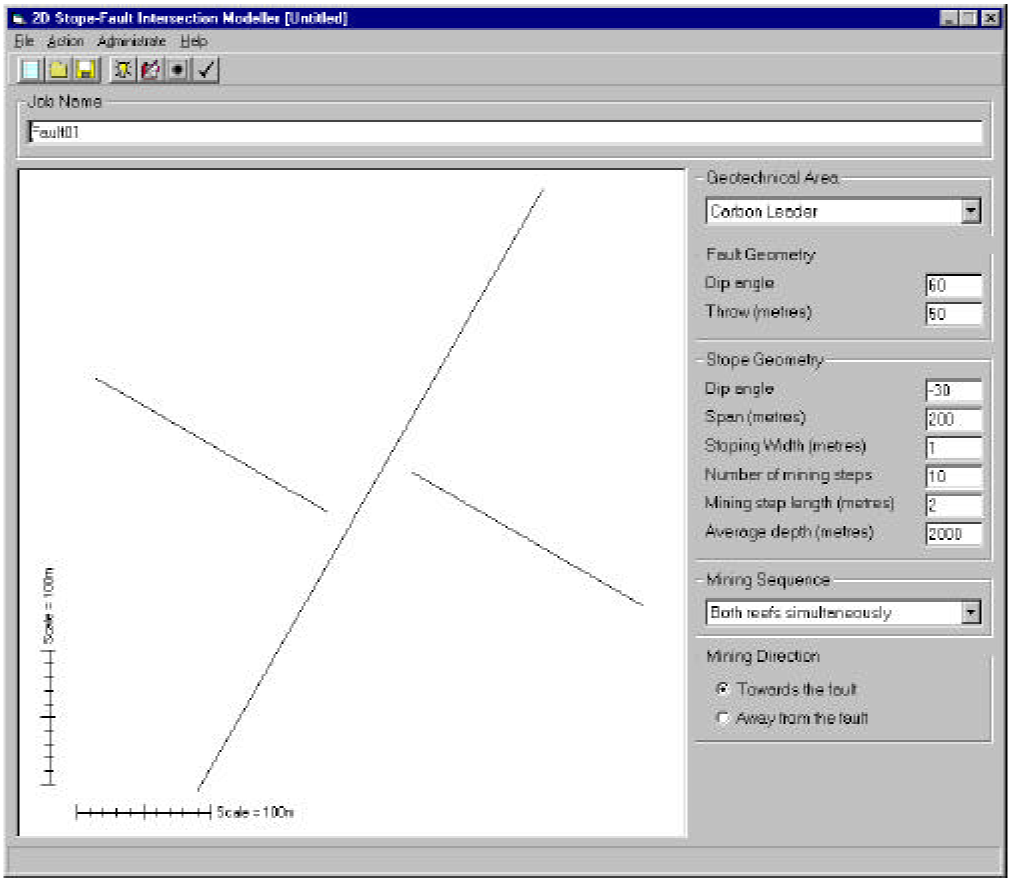
User Type
○ Normal
● Administrator

Figure 6-2: The User Manager Window. In this example only two user profiles are available and the second has been selected for editing.

The 2D Stope-Fault Intersection Modeller Domain Manager is the actual window that captures the input parameters necessary to build the numerical model. To date only this Domain Manager window has been programmed. However the system is designed such that future Domain Manager windows are constructed from an existing template.

The current Domain Manager addresses the problem of mining a stope in the vicinity of a geological fault (Figure 6-3). It considers only two-dimensional programs, and consists of a main graphic region that shows a sketch of the problem being

represented. To the right of this are the required input fields for the domain in question. The menu bar at the top provides the standard user with functionality to save the current work space, retrieve a previous work space, create the numerical modelling input file and exit or return to the Project Manager. An administrator is provided with additional access to lock specific controls on the form or to control which numerical modelling programs will be targeted by the Domain Manager on the current system
.



Job Name: Fault01

Geotechnical Area: Carbon Leader

Fault Geometry
Dip angle: 60
Throw (metres): 50

Stope Geometry
Dip angle: -30
Span (metres): 200
Stoping Width (metres): 1
Number of mining steps: 10
Mining step length (metres): 2
Average depth (metres): 2000

Mining Sequence: Both reefs simultaneously

Mining Direction
● Towards the fault
○ Away from the fault

Figure 6-3: An example of a Domain Manager User Interface (DMUI) window. In this case the domain is that of mining in the vicinity of a geological fault.

The Lock Control window is only available to administrators. It allows an administrator to disable any of the input fields on the associated Domain Manager window. This prevents a ‘normal’ user from modifying these fields enabling the administrator to enforce model consistency when this is desired. For instance, in the case of the 2D Stope-Fault Intersection Modeller the stope parameters may be well defined, but the fault throw and dip angle are known to vary across a given mining area. In this case all input fields other than those relating to the fault can be disabled. This concept could be extended to not just locking the controls but rather specifying bounds on the acceptable input values. For example, in this case all input fields but the fault dip and throw are disabled, the fault dip and throw can be limited to values within relevant ranges.

The Program List window contains a Check Box list of all the numerical analysis programs available to the current system. The application will select the most appropriate program from those checked in the list. This can be used to force the application to create input for a specific program by ‘unchecking’ all programs other than the required program. The programs available within the Program List window are those given by ‘anding’ the total available program list (as defined by the project manager) and the domain specific program list (as defined by the domain manager). Although this window is available in the prototype, at this stage, neither the project manager nor the domain managers have been programmed to manage program lists.

6.4 Interactive Post-Processing

A very brief phase of work was undertaken to assess alternatives to conventional analysis post-processing packages. Conventional post-processing facilities (MINSIM-W, BESOL/MS and BESOL/MINAP_97) facilitate the display of design criteria as static contours, vectors or shaded cells. In general the contour ranges can be modified through system dialogues. However, the whole process is static. It is not possible to hide or highlight features in real time. It was proposed that a facility that can modify display parameters in real time could emphasise regions of interest, or problem areas. The question was whether such real time modifications could be achieved on a real data set and if so would it highlight such features.

A very simple post-processor prototype was developed in Microsoft Visual Basic Version 5.0 (Figure 6-1). Although this prototype operates on a fixed MINSIM-W grid of static data it is useful to illustrate the main principles of interactive post-processing. In this case, the data represents the ESS values on an off-reef benchmark sheet placed on the surface of a geological fault. In traditional analysis, the friction angle would be set, and fixed, at the solution phase of the analysis. The ESS values would then be contoured and a threshold contour between acceptable and unacceptable ESS values would be selected at the final post-processing phase. In the case of the interactive post-processor the fault friction angle is not fixed, but can be varied by the user in real time. In addition the actual ESS threshold can also be varied in real time. The results of making variations in friction angle and/or threshold level are reflected instantaneously by the post-processor. In the case illustrated the values are shaded as either acceptable (green) or unacceptable (red) for maximum impact. However, it is quite feasible to provide a graded scaling between the extremes. Again, this option could be selected interactively in the post-processor by the user.

It is believed that this type of interactive post-processing facility offers interesting and valuable possiblities for future numerical modelling analysis programs. The impact of real time variation in parameters and design thresholds can be of significant advantage.

The post-processing prototype developed here was implemented in Microsoft Visual Basic, which is not renowned for producing fast executable programs. However with a data set of 4096 elements, the application provided more than adequate speed on an Intel 200MHz Pentium Pro with a Matrox Milenium I grahics card. Enhanced speed of graphics display was achieved in Visual Basic by making direct calls to the Windows GDI API and circumventing the standard Visual Basic functionality.

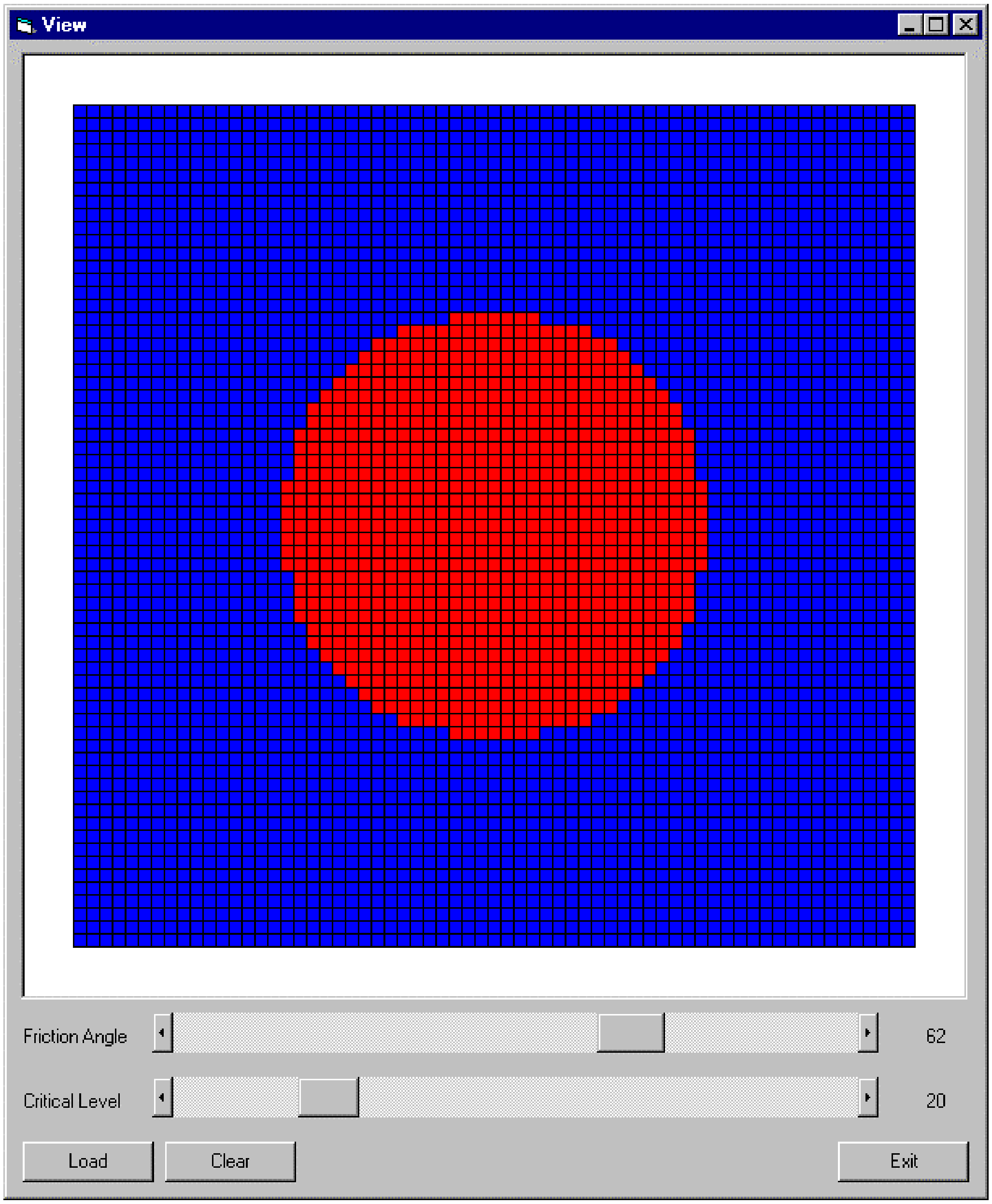


Figure 6-1: A prototype interactive post-processing system. A 64x64 element mesh is shown with ESS values divided between acceptable (green = dark grey) and unacceptable (red = light grey). The ESS friction angle can be changed interactively using a slider system. In addition the acceptable ESS threshold can also be modified with a second slider.

6.5 Expert Systems or Intelligent Interfaces?

A number of expert systems have been developed within the mining industry. The technology seems to offer the most potential benefit in the area of hazard risk assessment. However, there seems to be only limited, if any, potential for the direct application of the technology in numerical modelling. This is not to say that certain aspects of expert system technology cannot be used to enhance the field of numerical modelling. There is tremendous potential to use certain principles of expert system technology to improve numerical modelling user interfaces for both pre- and post-processing.

Pre-processing Modeller Wizards could help in the capture of routine problem data and regular mining geometries. Interfaces such as the Visual Model Master described here provide tremendous scope for user-friendly numerical modelling user interfaces. Both of these are strongly influenced by the expert system principle of domain specialisation.

With the advent of faster computers and better graphics systems, interactive post-processing is likely to become the norm rather than the exception. The additional insight provided by interactive manipulation of simulation results cannot be achieved by passive rendering.

Subsequent to the completion of the work undertaken in this sub-project two interesting new developments have emerged in the geotechnical modelling field. The first is that of PanTechnica Corporation in the USA who have developed a visual tool called PT Workshop for interfacing with the numerical modelling programs developed by Itasca Consulting Group. PT Workshop is relatively code specific and domain general, which conflicts with the approach proposed here. However, it does offer interesting possibilities for visual model development and data visualisation. The second development is more close to home. The CSIR Miningtek's new stress-strain analysis program and its data visualisation package provides the kind of facility discussed in the section on interactive post-processing. However, it takes the concept much further, moving into the realm of true interactive 3D graphics. In this arena, the limits of 3D geotechnical visualisation is bounded only by the imagination.

6.6 Milestone Delivery

This work was completed on schedule, by the end of December 1997. The work was reported in an interim SIMRAC progress report submitted in January 1998.

7 Conclusion

It was initially appreciated that the scope of this project was very large, but only during the course of the project's development did the true enormity of the scope become clear. Simply gathering and collating the varied literature proved to be an enormous task which was compounded by difficulties that arose when different texts provided different answers to the same problem. The true scale of the unknowns relating to aspects of both numerical modelling in general and actual numerical modelling programs in particular still has to be fully appreciated. However, the work within the scope of the original project was completed relatively timeously. It became apparent that an additional body of work (e.g. dynamics and discontinuum mechanics) required consideration: this is covered in a new SIMRAC project scheduled to commence in 1999.

The work covering the development of expert systems and intelligent user interfaces for augmentation of the numerical modelling process has shown that the greatest promise lies in the field of integrated intelligent user interfaces rather than stand-alone expert systems.

The opportunities offered by breaking traditional paradigms related to the design and development of numerical modelling program user interfaces is vast. It is perhaps prudent for the industry to concentrate less effort on the development of new numerical solution systems and focus more effort on user interface design and development.

8 References

Bandopadhyay S. and Venkatasubramanian P. 1988. A rule-based expert system for mining method selection. In *Computer Applications and Process Control.* CIM Bulletin 81(919): 84-88.

Buchanon B.G., Barstow D., bechtel R., Bennet J., Clancey W., Kulikowski C., Mitchell T. and Waterman D.A. 1983. Constructing an expert system. In *Building Expert Systems,* Hayes-Roth F., Waterman D.A. and Lenat D.B. (ed.), p. 127-167. Addison Wesley Publishing Company.

Butler A.G. and Franklin J.A. 1990. Classex: An expert system for rock mass classification. In *Static and Dynamic Considerations in Rock Engineering,* Brummer (ed.), A.A. Balkema.

Carbonell J.R. 1970. AI in CAI: An artificial-intelligence approach to computer-assisted instruction. *IEEE Transactions on Man-Machine Systems*, MMS-11(4):190-202.

Carrico M.A., Girard J.E. and Jones J.P. 1989. *Building Knowledge Systems: Developing and Managing Rule-Based Applications*. Intertext Publications, McGraw Hill Book Company.

Cichowicz A. 1993. Expert system for seismic hazard analysis. *Trans. Instn Min. Metall. (Sect A: Min. industry) 102*, The Institution of Mining and Metallurgy.

Coultard M.A. and Ciesielski V. 1990. An expert system to choose a stress analysis program for rock excavation. *Computer Methods and Advances in Geomechanics*, Beer, Booker and Carter (ed.), A.A. Balkema.

Dershowitz W.S. and Einstein H.H. 1984. Application of artificial intelligence to problems of rock mechanics. In *Rock Mechanics in Productivity and Protection,* 25th U.S. Symposium on Rock Mechanics, North Western University, p. 483-494.

Giarratano J.C. and Riley G.D. 1989. *Expert Systems Principles and Programming*. PWS-Kent Publishing Company.

Johnson R.A. 1988. Micro-computer based expert systems as and the use of numerical modelling programs for rock mechanics applications. *SANGORM Symposium: Rock Mechanics in Africa.*

Lightfoot N. and Napier J.A.L. 1994. On the road to a methodology of modelling in rock mechanics. In *The Application of Numerical Modelling in Geotechnical Engineering.* International Society For Rock Mechanics: South African National Group.

Ozan T.T. 1990. Expert systems and an application in coal mining strata control. *COMRO* Internal Note C10/90, CC8S04.

Paul J. and Gershon M. 1988. An expert system for fragmentation system design. S*ociety of Mining Engineers*.

Simons G.L. 1984. Introducing Artificial Intelligence. National Computing Centre.

Starfield A.M. and Cundall P.A. 1998. Towards a methodology for rock mechanics modelling. *Int. J. Rock Mech. Min. Sci. & Geomech. Abstr.* 25(3): 99-106.

Waterman D.A. 1986. *A Guide to Expert Systems*. Addison Wesley Publishing Company.

Webber S.J. 1996. Rockburst risk assessment on South African gold mines: An expert system approach. *Eurock '96,* Barla (ed.), A.A. Balkema, Rotterdam.