

THE UNIVERSITY OF CALGARY

SPREADSHEET BASED EXPERT SYSTEMS

by

Nabil Yacoub Haddad

A THESIS

**SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE**

DEPARTMENT OF ELECTRICAL ENGINEERING

CALGARY , ALBERTA

SEPTEMBER , 1990

© N. Haddad 1990



National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

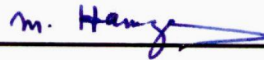
L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-79226-4

Canada

THE UNIVERSITY OF CALGARY
FACULTY OF GRADUATE STUDIES

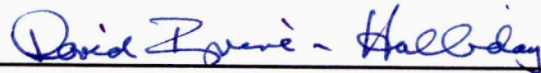
The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled, "Spreadsheet Based Expert Systems" submitted by Nabil Yacoub Haddad in partial fulfillment of the requirements for the degree of Master of Science.



Supervisor, Dr. M.H. Hamza
Department of Electrical Engineering



Dr. N.D. Rao
Department of Electrical Engineering



Dr. D. Irvine-Halliday
Department of Electrical Engineering



Dr. B. Gaines
Department of Computer Science

Date September 10, 1990

Abstract

Recent developments to spreadsheets have added a number of sophisticated features, all confined in one environment. These features include, but are not limited to, matrix operations, database functions, string functions and a programming language comparable in power to those used in conventional languages such as FORTRAN, Pascal and C. Moreover, nowadays a large number of individuals, businesses and firms are equipped with personal computers and popular software packages.

In an effort to take advantage of these two facts, a Spreadsheet Based Expert Shell was developed to construct expert systems. This shell is written using the popular Lotus 1-2-3 TM software, and hence provides an inexpensive and efficient way to develop expert systems, requiring no prior experience in knowledge system technology and no special programming capability. Any current user of spreadsheets may extend his skills to operate soundly this Shell, with a minimal effort.

Following the philosophy of spreadsheets, this Shell is completely menu driven, with specially designed screens to guide and prompt the user all through a session. In addition an on-line help facility is available.

The Shell carries two methods for developing expert systems. The first is the procedural method, which is rule based, having the form of IF-THEN-ELSE statements, with confidence levels assigned to the results. The second method is the declarative method, where the system is "taught" how to handle decisions, by building a number of examples. These two methods may be used separately or in conjunction while developing an expert system. A generous number of rules may be assigned to a single knowledge base, which can in turn be chained to other bases, allowing even a much larger knowledge base.

A valuable feature of expert systems is the ability to explain its reasoning process and justify conclusions. This Shell capitalizes on the database features of 1-2-3 by extracting the relevant rule, to reason why a question is being asked, and queries all "history" behind any specific conclusion, and can explain why any result was ruled out.

The rules are entered as sentences in any Latin based language. They are then parsed into segmented phrases of knowledge analyzed by the inference engine.

A number of debugging tools are available, to pinpoint missing information. Modifications may then be easily carried out, by using the powerful edit tool.

The system can exchange data with a number of programs including Lotus 1-2-3 (the host program), Symphony and dBase IV. In addition a temporary exit to DOS allows the execution of other programs and software.

Because of the integrated availability of three environments, the visible spreadsheet screen environment (including cells that house formulas), the Lotus Programming Language and the expert shell environment, this package holds a considerable promise for developing easy powerful and user friendly expert systems.

To demonstrate the flexibility and power of the Spreadsheet Based Expert Shell, some practical applications were developed. The first application is for a security advisor to recommend the most appropriate indoor and outdoor intrusion detection devices, cameras and access control for particular applications. The second application is a numerical integration advisor. Depending on the equation, degree of complexity and accuracy required, this expert system selects the most appropriate numerical integration method from among seven possible methods. The third expert system recommends the auxiliaries to be used for a generating set, advises on their cost and sizes the generator according to the load requirement.

Acknowledgment

The author wishes to express his utmost appreciation and gratitude to Dr. M.H. Hamza for his patience, guidance and support.

Special thanks to Dr. J.W. Haslett for giving me the opportunity to join the program with a generous financial support.

The author is indebted to the Department's professors for the knowledge they have provided. Special thanks to Dr. S. Lagu for the information he provided and for keeping my spirit high.

Thankful appreciations are extended to the special friends Amer, Abed, Anwar, Marwan, Correy, Elli, Mazen, Wafa, Ziad, Maha, Eyad, Lara, Yasser, Samira, Mike, Jasbir, Muna, Dena, and the rest for the moral support.

To Beirut,

***a city that died a thousand times,
and a thousand times was resurrected***

Table of Contents

Title Page	i
Approval Sheet	ii
Abstract	iii
Acknowledgment	v
Dedication	vi
Table of Contents	vii
List of Tables	xi
List of Figures	xii
 1 INTRODUCTION	 1
1.1 Thesis Objective	1
1.2 Thesis Overview	1
1.3 Thesis Outline	6
1.4 Thesis Contributions	7
 2 EXPERT SYSTEMS, EXPERT SHELLS AND SPREADSHEETS	 8
2.1 Introduction	8
2.2 Expert Systems	8
2.2.1 General	8
2.2.2 Features and Applications	12
2.2.3 Advantages and Disadvantages	13
2.3 Expert Shells	15
2.3.1 General	15
2.3.2 Commonly Used Programs And Shells	15

2.3.3 1st. Class Expert Shell	17
2.3.3.1 General	17
2.3.3.2 System Overview	17
2.3.3.3 Advantages and Disadvantages	18
2.3.4 Exsys Expert Shell	19
2.3.4.1 General	19
2.3.4.2 System Overview	20
2.3.4.3 Advantages and Disadvantages	21
2.4 Spreadsheets	22
2.4.1 General	22
2.4.2 Commonly Used Spreadsheets	23
2.4.3 Lotus 1-2-3 Spreadsheet	25
3 SPREADSHEET BASED EXPERT SHELL (SES)	27
3.1 Introduction	27
3.2 System Overview	27
3.3 Method Selection	33
3.3.1 General	33
3.3.2 Procedural Method	34
3.3.3 Declarative Method	36
3.3.4 Install Program	39
3.4 File Selection	40
3.4.1 General	40
3.4.2 New File	41
3.4.3 Old File	41

3.5 Utility Functions	42
3.5.1 General	42
3.5.2 Run and Again	43
3.5.3 Save	43
3.5.4 Edit	43
3.5.5 Print	45
3.5.6 DOS Access	52
3.6 Operation Functions	52
3.6.1 General	52
3.6.2 Advisor	53
3.6.3 Rule	58
3.6.4 Map	58
3.6.5 Statistics	59
3.6.6 Utility	60
3.6.7 Display	60
3.6.8 Compile	61
3.7 Help Functions	61
3.8 Validation of User Friendliness	62
3.9 Comparison with other Shells	63
3.9.1 General	63
3.9.2 Comparison of Procedural Method with Exsys	64
3.9.3 Comparison of Declarative Method with 1 st Class	65
3.10 System Advantages and Disadvantages	68
4 APPLICATION TO SECURITY EXPERT SYSTEMS	73
4.1 Introduction	73

4.2 System Overview	73
4.3 System Operation	74
4.4 Outdoor Intrusion Advisor	75
4.5 Indoor Intrusion Advisor	81
4.6 Camera Selection Advisor	85
4.7 Access Control Advisor	88
4.8 Discussion	90
5 TWO APPLICATIONS	92
5.1 Introduction	92
5.2 Numerical Integration Expert System	92
5.2.1 General	92
5.2.2 System Overview	93
5.2.3 Numerical Integration Advisor	93
5.2.4 Discussion	97
5.3 Generating Set Cost and Sizing Advisor	98
5.3.1 General	98
5.3.2 System Overview	98
5.3.3 Generating Set Advisor	99
5.3.4 Discussion	103
6 CONCLUSIONS AND SUGGESTIONS FOR FURTHER RESEARCH	104
6.1 Conclusions	104
6.2 Suggestions for Future Research	105
REFERENCES	108
APPENDIX A: Program Flow Diagram	112

List of Tables

2.1. Executive Summary of Spreadsheet ratings	24
3.1. SES and Exsys Speed Comparison	64
3.2. SES and 1 st Class Speed Comparison	66
4.1. Outdoor Detector Systems Ratings	81

List of Figures

1.1.	Analogy Between Programs and Expert Systems	3
2.1.	Expert Systems Building Blocks	9
2.2.	Expert Systems Major Components	10
3.1.	Schematic of Program	30
3.2.	Example of a Non Formatted Rule	30
3.3.	Example of a Formatted Rule	31
3.4.	Example of a Special Guideline Screen	31
3.5.	Sample of a Menu Macro	32
3.6.	Sample of a Macro	32
3.7.	Program's Logo	33
3.8.	Tree Structure	36
3.9.	Definition Screen Schematic	37
3.10.	Example Screen Schematic	38
3.11.	Install Screen	39
3.12.	File Selection Screen	40
3.13.	Utility Selection Screen	42
3.14.	Secondary Utility Selection Screen	44
3.15.	Print-out Screen	45
3.16.	Sample Report Screen	46
3.17.	Sample Output Screen	47
3.18.	Sample Report Screen	49
3.19.	Sample Definition Screen	50
3.20.	Sample Example Screen	50
3.21.	Sample Exhaustive Search Screen	51

3.22.	Sample Optimum Search Screen	51
3.23.	Sample Statistics Screen	52
3.24.	Sample Map Screen	52
3.25.	Operation Screen	53
3.26.	Pie Plot of Results	55
3.27.	Reasoning Screen	56
3.28.	Search Methods Screen	57
3.29.	Pie Plot of Statistics	60
3.30.	Help Index	62
3.31.	Exsys Output for the Access Control Advisor.....	64
3.32.	Partial Wine Selection Database.....	65
3.33.	1 st Class Wine Advisor output.....	65
3.34.	SES Wine Advisor output.....	66
3.35	Plot of Wine Advisor Output.....	67
4.1.	Outdoor Intrusion Question-Answer	77
4.2.	Outdoor Intrusion Summary	77
4.3.	Indoor Intrusion Question-Answer	84
4.4.	Indoor Intrusion Summary	84
4.5.	Camera Selection Question-Answer	87
4.6.	Camera Selection Summary	88
4.7.	Access Control Question-Answer	90
4.8.	Access Control Summary	90
5.1.	Numerical Integration Session Summary	94
5.2.	Numerical Integration Partial Examples	95
5.3.	Numerical Integration Output Screen	95

5.4.	Numerical Integration Partial Tree	96
5.5.	Numerical Integration Statistics Screen	97
5.6.	Generator Advisor Session Summary	99
5.7.	Generator Advisor Output Screen.	100
5.8.	Generator Sizing Program Output Screen.	102
5.9.	Onan Generator Sizing Program Output Screen.	103

1 INTRODUCTION

1.1 Thesis Objective

The aim of this thesis is to present a new expert shell developed within a spreadsheet environment, and to demonstrate its uses and new ideas through some practical examples.

1.2 Thesis Overview

Human knowledge if translated to machines may be preserved, secured, duplicated, enhanced and may be distributed to parts of the world where human experts do not exist or where they wish not to go. Hence comes the idea of expert systems to perform tasks and to provide advice in much the same way that it would be obtained and used if a human expert were available.

Expert systems is a domain of research, application and instruction concerned with programming computers [1]. The vital pre-requisite for developing expert systems is the availability of a person who is an expert in the subject area to draw on his knowledge and experience, and then to codify this information into computer language. This code should be designed in such a way to perform a search strategy and explain how a given result is obtained. A period of use is then required to verify and validate the system, while extra knowledge may be added at any later stage.

This scenario seems rather easy and straight-forward. However problems arise from difficulties in knowledge acquisition and modeling, which are so far primitive. As

an example, a chess game expert system, having thousands of rules, is still defeated by humans. This is explained by the fact that current intelligence modeling procedures do not reflect the exact methodology followed by humans. The human brain is the most complex computing device. The brain's powerful capabilities in thinking, interpreting, remembering and problem solving may be simulated with simplified computer models, but still humans have the ability of pattern recognition in addition to logical rules.

Despite these shortcomings some expert systems have proven to be quite handy and powerful, that many companies which are implementing them are finding them commercially useful which in turn justifies further research work.

To date there exists no general purpose tool to develop expert systems. A number of special programming languages and "expert shells" exist to aid in developing expert systems, each following its own peculiar approach, and is appropriate for specific type of applications.

An electronic spreadsheet on the other hand is a software that combines a straight-forward output oriented user interface with a number of features such as database, string functions and a built-in programming language. Spreadsheets have gained immense popularity. It is estimated that one developer has sold eight million copies [2].

A spreadsheet based expert shell (SES) was developed to capitalize on the spreadsheet features for constructing expert systems.

Most computer programs today perform tasks according to the decision-making logic of conventional programs. Programs consist of two distinct parts, algorithms and data. Algorithms determine how to solve specific kinds of problems, and data characterizes parameters in the particular problem at hand. Unfortunately human knowledge does not fit this model, because it consists of elementary fragments of know-how, pattern recognition and experience to define a decision.

Expert systems collect these fragments of information in a knowledge base and then access the knowledge base to reason about each specific problem through an inference engine.

An analogy for understanding these concepts of regular programs and expert systems is expressing them in an equation form as shown in Fig.1.1.

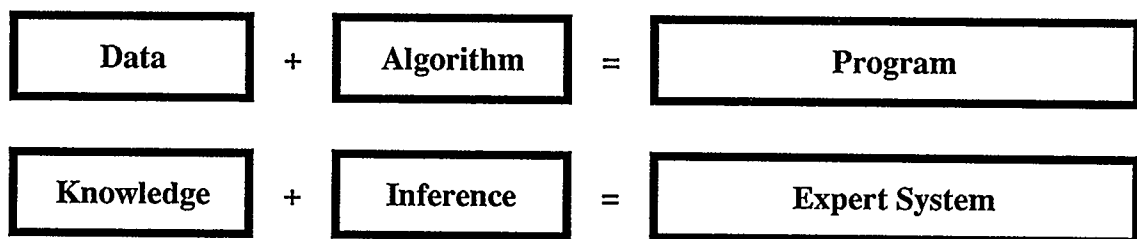


Fig.1.1. Analogy Between Programs and Expert Systems

Traditional research on expert systems focuses on the second equation. However some problems require the mix of the two concepts. Considering a plant operation as an example, on-line data is collected. This data needs then to be analyzed through an algorithm, which in turn reveals some computed values. Based on these values the knowledge base may automatically advise or perform some action.

For such applications one of the advantages of a SES prevails. This advantage is that algorithms, data, knowledge and inference are all confined to a single environment, facilitating the construction of expert systems and programs or a combination of both. Add to the above advantage, the user friendly environment and the large number of spreadsheet users, we can assume that this package may hold a considerable promise for future research and practical applications.

Following the philosophy of spreadsheets, the SES is completely menu driven, with specially designed screens to guide and prompt the user all through a session. In addition an on-line help facility is available.

The SES carries two methods for developing expert systems. The first is the procedural method, which is rule based, having the form of IF-THEN-ELSE statements, with confidence levels assigned to the results. The second method is the declarative method, where the system is given a database of examples. These two methods may be used separately or in conjunction while developing an expert system. The latter technique proves to be of great value. A generous number of rules (8150) may be assigned to a single knowledge base, which can in turn be chained to other bases, allowing even a much larger knowledge base.

A valuable feature of an expert system is its ability to explain its reasoning process and justify conclusions. This Shell capitalizes on the database features of 1-2-3, by displaying the relevant rule to reason why a question is being asked, and queries all "history" behind any specific conclusion, and can explain why any result was ruled out.

In order to allow professionals develop directly their knowledge bases, the rules are entered in English sentences, which are then parsed into segmented phrases of knowledge, which is then analyzed by the inference engine.

A number of debugging tools are available, to pinpoint missing information. These include a display of all possible routes in a form of a tree, or only those branches of the tree that lead to a result. Modifications may then be easily carried out, by using the powerful edit tool.

The system can exchange data with a number of programs including Lotus 1-2-3 (the host program), Symphony and dBase IV. In addition a temporary exit to DOS allows the execution of other programs and software.

Because of the integrated availability of three environments, the visible spreadsheet screen environment (including cells that house formulas), the Lotus Programming Language and the expert shell environment, this package holds a considerable promise for developing easy, powerful and user friendly expert systems.

This Spreadsheet Based Expert System inherits most of the characteristics of the environment it was written in, and is simply an add-on program that allows knowledge bases to be easily constructed.

Demonstrating the scope, power and features, several practical expert systems were developed. The first recommends the most suitable equipment for security purposes, the second selects the appropriate mathematical integration method to follow, while the last advises on the cost of a standby generating set and sizes it

according to the load requirements.

1.3 Thesis Outline

Chapter two of this thesis is a brief about expert systems, expert shells and spreadsheets. It follows the recent developments, and highlights the features and advantages of each system and analyzes a few of these tools available in the market. Finally it emphasizes the reasons behind developing a spreadsheet based expert shell (SES).

Chapter three details the SES, explains how to operate the shell and develop expert systems. It also tutors on operation of the shell's tools and the navigation between the available levels and demonstrates the various screens displayed while running the program.

Chapter four is the description of the first practical application of the SES. It is a security systems advisor, which recommends the most appropriate indoor and outdoor intrusion detection devices, cameras and access control protection.

In chapter five two expert systems provide additional examples of the use of the SES. The first selects the most suitable numerical integration method, while the second advises on the cost of a generating set, and then links to an actual sizing program.

Conclusions and future work are finally presented in chapter six.

1.4 Thesis Contributions

The most important contribution of this thesis is the development of the spreadsheet based expert shell SES. This ties the spreadsheet environment to that of expert shells and gives millions of spreadsheet users access to this expert shell.

The second contribution is the availability of two distinct expert system development methods, in the SES. The first method uses proper language sentences, entered as the system's knowledge, which is then automatically parsed into important pieces of information. The other method teaches the SES, in the form of examples, the logic or the sequence of decision making followed by the experts. The use of any, or the combination of the two development methods is permissible. In addition the SES provides links to spreadsheets and other programs.

The development of some practical expert systems is the third contribution. The systems studied may be applied in industry or utilized for educational purposes.

2 EXPERT SYSTEMS, EXPERT SHELLS AND SPREADSHEETS

2.1 Introduction

This chapter is intended as a brief about expert systems, expert shells and spreadsheets. It follows the recent developments, and highlights the features and advantages of each. Finally it emphasizes the reasons behind developing a spreadsheet based expert shell (SES).

2.2 Expert Systems

2.2.1 General

One definition of artificial intelligence is that it is the study of ways in which computers can be made to perform tasks that, if performed by a human being, would be considered intelligent [3]. Artificial intelligence research has branched into more specialized areas one of which is expert systems, the others being speech processing, vision, robotics, cognitive modeling, knowledge representation, problem solving and inference [4].

Research in artificial intelligence began in the mid-1950s shortly after the first digital computers emerged. Computers were primarily designed to carry out numerical computations in an efficient way, but then it was observed that computers were not limited to numbers, but were capable of general processing of all kinds of symbols or patterns. Artificial intelligence exploits these capabilities.

Expert systems in the narrow sense is a part of computer science, aimed at exploring the range of tasks over which computers can be programmed to behave intelligently. It produces intelligent responses to some specific tasks. Expert systems (also known as knowledge systems) are providers of advice in much the same way that it would be obtained and used if a human expert were available [5]. They are intended and designed to emulate human knowledge and expertise.

Fig.2.1 illustrates the primary building blocks of expert systems. The base level consists of those techniques that underlie nearly all applications. Level two shows the most frequently used forms of knowledge representation, while the third level reflects the methods of searching, reasoning and organizing the flow of the program. Finally the cap-stone of the expert systems is the ability of communication with knowledge engineers, experts, consumers, databases and other computer systems [6].

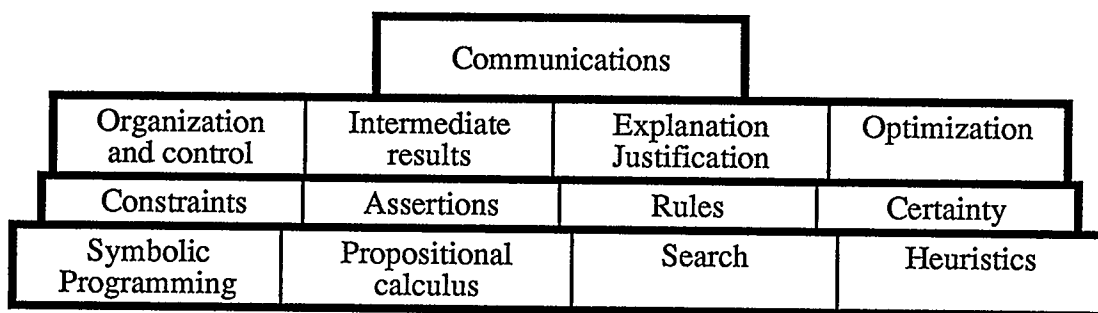


Fig.2.1. Expert Systems Building Blocks.

Fig.2.2 illustrates the major components of contemporary expert systems. The figure depicts the knowledge system as a computer application with distinctive development and operational environments. For development of an expert system, tools are needed to acquire, maintain, validate and interface with the user. Using these tools the knowledge engineer constructs knowledge systems that incorporate the three key components, the knowledge base, the inference engine and the user interface. Once a knowledge system completes development, it enters operation. In this environment it ordinarily accesses databases, connects to various communication networks, transfers to or integrates with existing installed equipment, and may receive data directly from sensor systems.

DEVELOPMENT ENVIRONMENT		User Interface	OPERATION ENVIRONMENT
		Natural Language-Graphics Explanation	
	Acquisition Tools	Knowledge Base - Inference Engine	
	Maintenance Tools	Representation-Problem Solving	
	Validation Tools	Programming Environment	
	User Interface Design Tools	Programming Language System	
		Operating System	Communication Networks
		Computer	Installed Equipment
			Databases
			Sensors

Fig.2.2. Expert Systems Major Components.

In the building of expert systems, it is necessary to have available, and to draw on, the knowledge and experience of a person who is already an expert in the subject area. Knowledge acquired to build an expert system is derived from experts, who are providers of information, problem solvers and who can explain the reasoning behind

their method chosen [7].

Knowledge Representation in expert systems is contained and given by the rules in the knowledge base rather than computer coding as in conventional programs. A collection of rules derive the knowledge of many expert systems rather than algorithmic procedures. Expert systems tend to work with information in a symbolic form (such as natural languages) rather than being calculations performed in numeric form.

In seeking a solution to a problem, expert systems examine, select and manipulate rules in the knowledge base by means of search strategies, such as forward and backward chaining. The portion of the system performing the logic functions is known as an inference engine.

Expert systems are often written in list processing languages such as LISP, logic programming languages such as PROLOG, or expert shells (which will be discussed in subsection 2.3), rather than procedurally oriented ones such as FORTRAN. An important attribute of expert systems is their ability to explain how a given result was obtained. PROLOG, LISP and usually the expert shells all have the facilities to trace how a decision was reached and provide reasoning behind a conclusion.

In order to verify and validate an expert system, it is to be subjected to a period of rigorous apprenticeship and testing.

The knowledge base is separate from the search logic, and hence it is possible to add new rules or modify existing rules without having to review or reorganize the existing structure.

The increasing use of expert systems for diverse applications provides a good index for the growth of this field. While early applications were largely developed by universities on powerful machines, nowadays most expert systems are being developed by companies for in-house applications, utilizing workstations or personal computers.

2.2.2 Features and Applications

Expert systems provide many features [8]. Among these features is relieving the burden on experts. This diminishes the queries directed to them, and multiplies the expert knowledge, leading to redistribution of expertise. Expert systems also secure, duplicate and distribute the knowledge of human experts.

Expert systems are ideal for certain training purposes, where for example an introduction of a new product in a company leads to a high training cost. The problem solving and explanatory capabilities of the expert system can tutor and teach the special skills required. In addition, as the volume of available knowledge grows, expert systems are not bewildered by growing amounts of data, but on the contrary function better. They investigate all hypotheses which can be pursued, and do not overlook any lead, and can give information and explain why they have chosen a certain strategy and rejected another.

Finally expert systems allow for an efficient reduction of outcome to a few possible solutions, where a fast and reliable decision is required.

Currently there are four major areas where expert systems have been proven successful [9]. These are the following:

- Diagnosis and repair, which is data dependent with fixed alternatives, and is based on question-answer dialogue. Medical diagnosis and car repairs are good examples of such systems.
- Event or data driven procedures, where intricate details and complex data-dependent ordering is important.
- Modeling and simulation of organizations and mechanisms, which uses a declarative knowledge representation and inference technique.
- Design and planning, which is goal directed and problem solver.

2.2.3 Advantages and Disadvantages

Among the pros of expert systems [10] is that they allow the automation of tasks that could not be effectively automated before. Expert systems are a productivity multiplier, by replicating the expertise of human experts. In addition they act as a "corporate memory" for the knowledge held and promise increased efficiency in developing future systems, because of the potential of reusing knowledge from existing ones.

Expert systems can carry out tasks in locations where human experts do not exist, where they do not wish to go, and where communication with human experts is not feasible.

Expert systems can handle complex fast changing situations and, when compared to other computational approaches, are more robust and designed to deal with problems exhibiting uncertainty, ambiguity, inaccuracy and missing data. They provide the facilities to easily modify and upgrade the knowledge, and allow an on-line explanation of reasoning and conclusions.

Another advantage of expert systems is that they can cope with imprecision of human decision making, through the use of models of uncertainty, and can interact intelligently with the users, through a natural language understanding. Expert systems are more efficient than other automation techniques, by virtue of their ability to understand the critical aspects of the current problem.

Unfortunately there are disadvantages to expert systems. Among them is that expert systems are expensive to build. They tie up the best experts for a long time, while developing the system.

No general purpose intelligence exists and hence expert systems are feasible in narrow domains. Up till now they have not yet proven to have many "successes". To date the learning process of expert systems is primitive.

Expert systems can not be left alone to run autonomously for long periods, and require a different development methodology than other programs.

2.3 Expert Shells

2.3.1 General

An expert system allows the user to engage in a question and answer consultation concerning a problem pertinent to an application [11]. A valuable feature of expert systems is the ability to explain its reasoning process and justify conclusions. At present there are several powerful software tools available, generically called "shells", which provide an inexpensive and efficient way to develop an expert system. They require no prior experience in knowledge system technology and no special programming capability. They allow the knowledge engineer to concentrate on building the knowledge base.

2.3.2 Commonly Used Programs And Shells

The availability of a large number of tools and computers for this industry, suggests that there is not a single artificial intelligence machine or tool that is best for everybody, and clearly the best selection is application dependent, and each machine and software has its own niche.

The personal computer is becoming a popular tool for artificial intelligence programming due to the low cost involved, the large number of personal computer users and the introduction of powerful and fast computers [12]. The primary uses are for experimental prototype small-scale expert systems and knowledge bases, training of personnel and troubleshooting.

LISP [13] and PROLOG [14] "dialects" as well as expert shells for building expert systems specifically for personal computers are now available from numerous suppliers. Some of this software is quite powerful and has been applied in real-world problems.

An expert system may be implemented by using special programming languages such as LISP and PROLOG [15], but such activity normally requires a great deal of effort. Although these languages are well structured, they require extensive programming efforts. Even an experienced programmer finds difficulty in adapting to these languages, due to the completely different structure, reasoning and execution processes.

Shells offer an alternative, where a user friendly environment exists. Usually these shells are menu driven, easy to debug and knowledge can be revealed and is simple to understand. However due to the large number of shells sweeping the market, caution should be exercised in selecting the most appropriate one. As mentioned earlier the key element in the selection process is the application required. A careful review is necessary, to find out the limitations of each shell. Some of the shells available are Guru, 1st Class, Exsys, Framentac, Comex, Nexpert, OP5, and Rosie. An in-depth analysis of two of them follows in subsections 2.3.3 and 2.3.4, to contrast the approaches followed by expert shell producers, and to provide the opportunity to compare these shells to the SES. Due to financial constraints, only two shells were purchased and analyzed in this thesis.

2.3.3 1st. Class Expert Shell

2.3.3.1 General

The goal of 1st Class expert system is to provide a tool that allows the end users to create an expert system without having to learn a whole new set of skills [16]. The development of an expert system is achieved through a series of menus using spreadsheet and database layouts.

2.3.3.2 System Overview

With 1st Class shell an expert system is built in four distinct phases. In phase one, key factors are identified into the spreadsheet format. The head of each column is the title that identifies the specific factor, while the rows below it hold values that can be assigned (for example switch-position is a factor, **on** and **off** are the values or definitions that may be assigned to the factor). The right-most column is allocated for the results, outcome or possible actions and recommendations.

After entering the definitions, the system moves on to phase two, where examples illustrating how a decision is reached are entered. In phase three, the system asks the user to select one of six methods available for the search technique. The first five of these methods define and determine the order of, and the number of the questions that are necessary to be asked, in order to achieve a result. The sixth is a customized expert system , where the user graphically edits or

creates the rules.

Phase four allows the expert system developer to experiment on the operational prototype model he created, then to modify the examples and the search method until his initial aim is achieved.

2.3.3.3 Advantages and Disadvantages

Overall this package is quite impressive [17]. Some of the advantages it offers are:

- Good documentation.
- Comes with several sample databases.
- Good educational tool for introducing expert system knowledge.
- Easy to use.
- Use of natural languages to develop knowledge bases.
- Clarity of knowledge representation.
- Offers a number of easy to use debugging, testing and editing tools.

However a number of disadvantages prevail. Among these are:

- Limited knowledge base size. This shell allows only 32 factors and 256 examples. To put this into correct perspective, considering a knowledge base made out of only four factors each having four definitions, would entitle 256 examples to completely represent the full knowledge and hence the full size available. Developers advise chaining the files as a remedy, but that proves quite cumbersome even for a reasonably sized expert system.
- Not all knowledge may be easily represented. Unless the knowledge is frame-based (defined as objects), it is hard to draw conclusions.
- The search techniques described in subsection 2.3.3.2 may be misleading. For example the optimize search uses the least number of questions to draw a conclusion. If the knowledge base is missing some examples, the result may be one of the outcomes instead of "no-available-data".
- Even though the edit functions are so convenient, the fact that large knowledge bases need to be split to fit the limitations imposed, makes editing so cumbersome and hard to follow.

2.3.4 Exsys Expert Shell

2.3.4.1 General

Exsys is a generalized expert system development package. Its developers suggest the possibilities of developing expert systems, varying from "automating complex regulations, aiding customers in selecting from among a group of

products, automated assistance and identification of diseases or problems" [18].

2.3.4.2 System Overview

Expert systems maybe developed with Exsys for any problem that involves a selection from among a group of choices, where decision is based on logical rules. The rules can involve relative probabilities of a choice being correct. Exsys deals with knowledge rather than data and rules are in the form of IF-THEN-ELSE statements. Truth of the IF condition determines if the statements in the THEN or ELSE parts are considered and added to what the system knows. This is achieved through a series of questions asked about the rules until a conclusion is reached.

The search for results in the knowledge base may be achieved in one of the five options:

- Normal Backward Chaining, where the program only questions the validity of those rules that have a result or lead to an outcome. It puts in order the rules that yield the same outcome, and then checks if they are dependent on other rules; if so a backward chaining occurs where the validity of the latter rule is checked, and information about the original dependent rule is derived.
- Final Pass Option, which is similar to backward chaining, except it also tests the rules that do not lead to a result.
- Forward Option, where the difference between this option and the normal backward chaining option is that the rules are tested in the natural order they

were originally built in, while allowing backward chaining to take place. No ordering of the rules is allowed to take place as is the case in the backward chaining option

- Nobackward Option, which like the backward chaining orders the rules that lead to a result, but does not allow backward chaining to occur, even if information is needed from other rules.
- Both Nobackward and Forward Option, which is identical to the nobackward chaining option, except it asks the questions in the order they are originally inputted.

2.3.4.3 Advantages and Disadvantages

This package yields a number of positive features, among which are:

- Good documentation.
- Comes with several sample databases.
- Use of natural languages to develop knowledge bases.
- Clarity of knowledge representation.
- Allows for a generous size of rules and qualifiers.

However a number of disadvantages prevail. Among these are:

- Long time required to input the rules. The package does not allow the copying or replicating of the rules.
- Editing or modifying is not a simple task, as looking for a certain qualifier (addressed by a number) can be difficult to find and time consuming.

2.4 Spreadsheets

2.4.1 General

Electronic spreadsheets are one of the most ingenious software developments; they combine a straight-forward, output oriented user interface with the computer's ability to calculate and display numbers and strings. Recent developments to spreadsheets have added a number of sophisticated features, all confined in one environment. These features include, but are not limited to, the following:

- A built in programming language with simple commands most of which are English words (referred to as Macros). This language has the full capabilities and is comparable in power to conventional programming languages such as Fortran, Pascal and C.
- Some powerful mathematical functions, (such as matrix operations where inversion and multiplication of a generously sized 90 x 90 matrix may be applied).

- Some powerful string functions, such as string search, string comparison, and splitting into smaller strings on demand.
- A handful of database functions of extreme power and speed.
- A number of easy to generate graphs of different patterns.

2.4.2 Commonly Used Spreadsheets

Although new spreadsheets are endowed with a huge assortment of powerful features, these features come at a price. Ironically, some of the new richness of features comes at the expense of simplicity, user interface, speed and hardware requirements.

Many packages are now available in the market, such as Excel, Quattro, Lucid, Lotus 1-2-3, WingZ, and Supercalc. However Lotus 1-2-3 has reigned supremely for years; none of its contenders has inherited its mandate, but some pose a serious challenge. Competition to gain larger market shares is split among "heavy duty" users, who are equipped with the state of the art machines (386 computers with expanded memory), and low end users (the older XT computers). As a consequence, spreadsheet developers such as Lotus are marketing two products suitable for each category. The first takes advantage of the high performance offered by the new machines (release 3), while the second suits the consumers still equipped with the older machines (release 2.2).

Although Lotus 1-2-3 new releases offer a stunning array of powerful features, it leaves novices struggling with a difficult interface. Lotus's archrival Excel has upped the spreadsheet stakes with powerful reporting capabilities and a graphic interface that is both potent and intuitive. SuperCalc has stayed in the game by improving on the lower end Lotus 1-2-3 release 2 platform with a host of extra features.

Different reviews come out with conflicting results as to which is the best package [2][19][20]. This is attributed to the difference in the testing procedures, as no standard test exists, and this is a sign that each package offers some better aspects. However most reviewers agree that, Lotus 1-2-3, Quattro, and Excel are the top packages. Table 2.1 is an extracted executive summary comparing three of these packages.

<u>FEATURE</u>	<u>Lotus 1-2-3</u>	<u>Excel</u>	<u>SuperCalc</u>
Spreadsheet	E	E	G
Database	E	G	F
Reporting & Graphs	G	E	E
Macros	G	E	G
Performance	E	G	F
Overall Value	E	E	G
<u>Rating:</u> E = Excellent G = Good F = Fair			

Table 2.1. Executive Summary of Spreadsheet ratings.

Subsection 2.4.3 focuses on Lotus 1-2-3 spreadsheet, and expands on some of its features.

2.4.3 Lotus 1-2-3 Spreadsheet

Lotus 1-2-3 spreadsheet delivers a realistic way to organize, analyze and manage complicated applications. It offers powerful data management capabilities, graphics, full network support, programming tools and rich documentation and indoctrination. In addition a three dimensional spreadsheet is available for those users equipped with 286 or 386 personal computers [21].

The software, which is written mostly in C language, currently runs on IBM personal computers, using either DOS or OS/2 operating systems. Recently a new release was introduced to run on the IBM S/370 and DEC Vax/VMS machines [22]. As a consequence transfer of data and programs is now possible from Personal Computers to larger systems and vice versa.

This spreadsheet is handy for both novices and advanced programmers, as the skills to use it are easily acquired. A single keystroke activates the main menu, where operations pertinent to the whole worksheet or a range are possible. These operations may be to format in a certain style, copy or move some cells, perform a database or plot a graph in any style.

The built-in programming language may automate any manual spreadsheet operation or can be used as a typical procedural programming language. Recalculation or printing can take place in the background, while the user is still operating the system.

Based on assessments in this and the previous subsection 2.4.2, and as Lotus sets the standards in this industry, and 1-2-3 is available on larger machines and has the biggest market share, Lotus 1-2-3 was selected to fully develop the Spreadsheet Based Expert Shell (SES).

A survey of the available shells in the market showed that some of these shells have a "spreadsheet format", and may link data to spreadsheets, but none is a spreadsheet itself.

3 SPREADSHEET BASED EXPERT SHELL (SES)

3.1 Introduction

From the early days of personal computers, spreadsheets were popular and this software was one of the leading in this industry. Faced with an array of success and high demand, the producers of these programs have constantly improved on their products, adding new features, and making spreadsheets available on larger systems.

Unfortunately many engineers and scientists have often looked down on spreadsheets as being too simple and limited. They based their judgement on the early spreadsheets which only catered to a few useful scientific operations. However, recent developments to this software have added a number of features, that anyone who ignores them is perpetually losing a handy, simple and powerful tool.

Yet in the very excess of power and features lies the simplicity and user friendliness of spreadsheets [23]. Moreover, nowadays a large number of individuals, businesses and firms are equipped with personal computers and popular software packages such as the spreadsheets.

3.2 System Overview

In an effort to take advantage of the facts that advanced and sophisticated spreadsheets are now available, and quite a large number of spreadsheet users exist and is growing steadily, a Spreadsheet Based Expert Shell (SES) was developed to construct expert systems. This shell is written using the popular TM Lotus 1-2-3

software, and hence provides an inexpensive and efficient way to develop expert systems, requiring no prior experience in knowledge system technology and no special programming capability. Any current user of spreadsheets may extend his skills to operate soundly this Shell, with a minimal effort.

Following the philosophy of spreadsheets, this Shell is completely menu driven, with specially designed screens to guide and prompt the user all through a session. In addition an on-line help facility is available.

The Shell carries two methods for developing expert systems. The first is the procedural method, which is rule based, having the form of IF-THEN-ELSE statements, with confidence levels assigned to the results. The second method is the declarative method, where the system is "taught" how to handle decisions, by building a number of examples. These two methods may be used separately or in conjunction while developing an expert system. The latter technique proves to be of great value. A generous number of rules (8150) may be assigned to a single knowledge base, which can in turn be chained to other bases, allowing even a much larger knowledge base.

This Shell capitalizes on the database features of 1-2-3, by extracting the relevant rule to reason why a question is being asked. SES also queries all "history" behind any specific conclusion, and can explain why any result was ruled out.

The rules are entered in English sentences, which are then parsed into segmented phrases of knowledge, which are then analyzed by the inference engine.

A number of debugging tools are available, to pinpoint missing information. These include a display of all possible routes in the form of a tree, or only those branches of the tree that lead to a result. Modifications may then be easily carried out, by using the powerful edit tool.

The system can exchange data with a number of programs including Lotus 1-2-3 (the host program), Symphony and dBase IV. In addition a temporary exit to DOS allows the execution of other programs and software.

Because of the integrated availability of three environments, the visible spreadsheet screen environment (including cells that house formulas), the Lotus Programming Language and the expert shell environment, this package holds a considerable promise for developing easy, powerful and user friendly expert systems.

This Spreadsheet Based Expert System inherits most of the characteristics of the environment it was written in, and is simply an add-on program that allows knowledge bases to be easily constructed.

Four levels of menus are available. Navigation between these menus is very simple, and an on-line Help is always available. These four levels of menus are the method selection, the file selection, the utility and the operation. An indicator, which is located in the upper right corner of the screen, shows the mode or level the program is in at any time. The following subsections will discuss each in detail. The program schematic is shown in Fig.3.1.

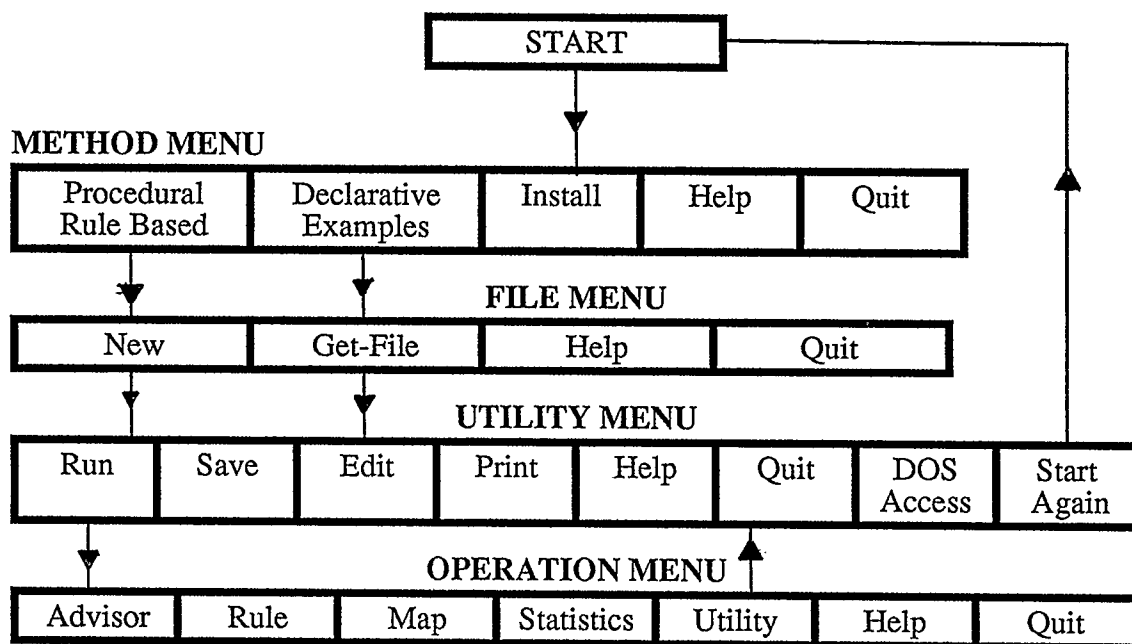


Fig.3.1. Schematic of Program

This Spreadsheet Based Expert System capitalizes on most of the features offered by Lotus 1-2-3. First a specially designed template allows the user to input an English structured sentence, Fig.3.2, then string commands in the template split the sentence into useful information, Fig.3.3, which are then stored in the database.

Rule No. 1
<p>-----</p> <p>The policeman said : "If you cross the street on red light then you will get a ticket and you might have an accident else you are a good citizen."</p> <p>According to him the probability is 30,10 else 100</p> <p>He seems to be right (but I will still run a yellow light!)</p> <p>*****</p>

Fig.3.2. Example of a Non Formatted Rule

Rule No. 1		
=====		
IF	you cross the street on red light	
THEN	you will get a ticket	Probability % 30
and	might have an accident	Probability % 10
ELSE	you are a good citizen	Probability % 100
NOTE	He seems to be right (but I still will run a yellow light!)	

Fig.3.3. Example of a Formatted Rule

Special screens for the beginning of the program, error messages, guideline on what to expect next, prompting the user for an action or a delay, advice and help, are all available to enhance the user friendliness [24]. Fig.3.4 is an example of such screens.

WELCOME TO S.E.S.	

Please select one of the following choices, by pressing one of the letters or using the arrow keys (<- or ->) then pressing return.	
(P)	Procedural (Rule Based Expert System)
(D)	Declarative (Example Based Expert System)
(I)	Install (Select File Directory)
(H)	Help (Get Help)
(Q)	Quit (End Session)

Fig.3.4. Example of a Special Guideline Screen

The major part of the SES is written as Macros, which are Lotus 1-2-3 subroutines [25]. They are divided into two categories: one to allow menu driving of the program, Fig.3.5, where the user selects using arrow keys, or typing one letter, the

desired flow of the program; the other is to allow loops or repetition of procedures, reading data, user interaction, processing of information and controlling screens,

Fig.3.6.

CHOICE	Procedural	Declarative
	Rule Based Expert System	Example Based E. S.
	{LET Type,1}~	{LET Type,2}~
	/cT41..W43~DI40~	{GOTO}A281~
	{GOTO}A281~	{RETURN}~
	{RETURN}~	

Fig.3.5. Sample of a Menu Macro

MAIN	{GETLABEL Title of Knowledge Base (File Name)...,CurrentFile}~
	{GETLABEL Designer's Name... ,D38}~
	{GETLABEL Date... ,D39}~
	/rtD37..D39~DI1~
	/xl Text to be Displayed at Beginning....~BeginTxt~
	/xl Text to be Displayed at End....~EndTxt~
	{GETNUMBER Number of Rules... ,RuleNo}~
	{IF Type=2} {BRANCH GetData2}~
	{HOME}~
	{IF Type=1} {FOR RuleCounter,1,RuleNo,1,GetData}~
	{Process}~
	{INDICATE Utilt}~
	{MENUBRANCH Utility}~

Fig.3.6. Sample of a Macro

Having to deal with knowledge bases, one of the most important aspects is to search for a certain result using definitive criteria. This program capitalizes on the built in feature of Lotus of matching data using the Database Functions, which are extremely fast [26]. Sections 3.3 through 3.6 present a detailed description of the shell.

3.3 Method Selection

3.3.1 General

As soon as the program is loaded the logo is displayed for five seconds, Fig.3.7, after which the user is prompted to select one of the methods he desires to be used: Procedural (rule based expert system) or Declarative (example based expert system). It is of utmost importance to have an in depth perception of these methods, so as to select the most appropriate method for a particular knowledge base. The following subsections will detail each method. An install program is also available, to indicate to SES the directory where the files are to be saved and retrieved. Help and ending the session, through a quit command, may be acquired as well at this point.

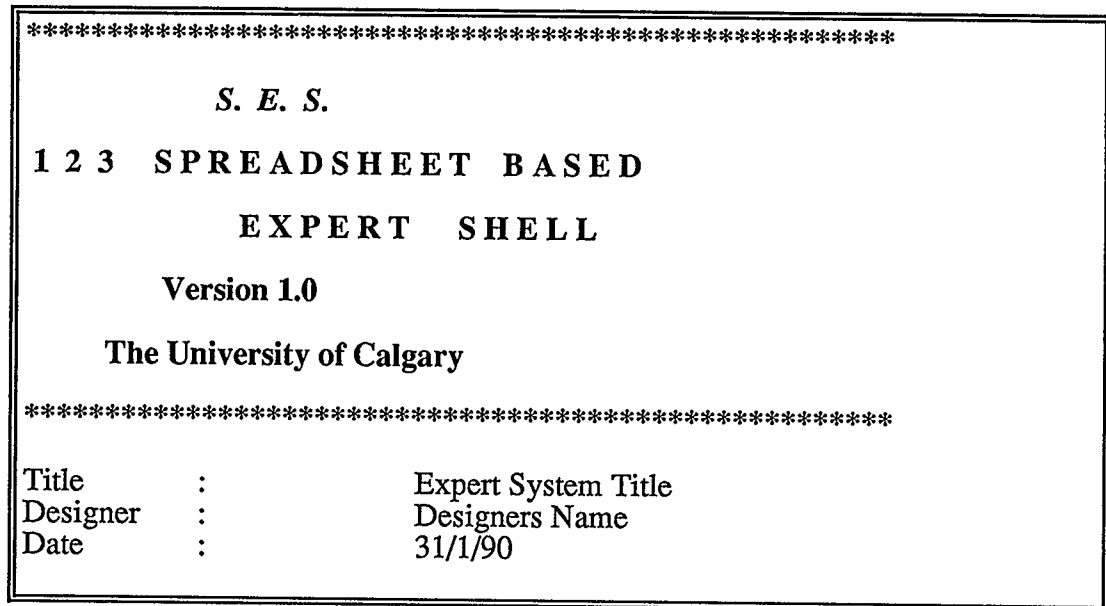


Fig.3.7. Program's Logo

3.3.2 Procedural Method

The procedural method is rule based. The rules are in the form of If... Then ... Else..., with logical operators in between. For better documentation and in order to allow the professionals to express themselves in English sentences, a rule may be written in this form:

The expert Mr. X said in New York on the fourth of July : " If a then b and c else d."

Automatically the template splits this sentence into the important knowledge information: The truth of **a** implies the truth of **b and c**; otherwise **d** is true. The other information such as who said this information, where and when, is stored as a reference for the end user (after the knowledge base has been constructed) to assess the quality of the information and its sources.

The second step is to assign confidence to the knowledge acquired. This also is entered in an English sentence as follows:

According to Mr. X the probability is 50,60 else 90

The values will be assigned by the template to correspond respectively to the phrases in the **THEN** and **ELSE** parts. In this example **b** will be assigned fifty percent, **c** sixty percent, while **d** (in the else part) ninety percent.

Caution should be exercised that the moment a probability is assigned the corresponding phrase is converted into a result or an outcome. Should the phrase not be a result, a probability should not be assigned. In the latter case the phrase is used as a logical statement to derive information.

To determine the overall confidence in a certain result, several methods exist [27][28]. In general they depend upon whether the outcomes are dependent on each other or not. The problem is very complex [29] and great care has to be used in the determination of the overall confidence.

The computation method used to calculate the overall value can have a considerable effect on the outcome [30]. For this package the average method is adopted. It is basically a ranking method where the average is used to determine the order of the outcome, and is only an indication of which system is more suitable under particular conditions. Such an approach is also used in other shells [18].

Additional reference to help the end user may be attached in the form of a note, so whenever the rule is displayed, this note appears as an extra piece of information.

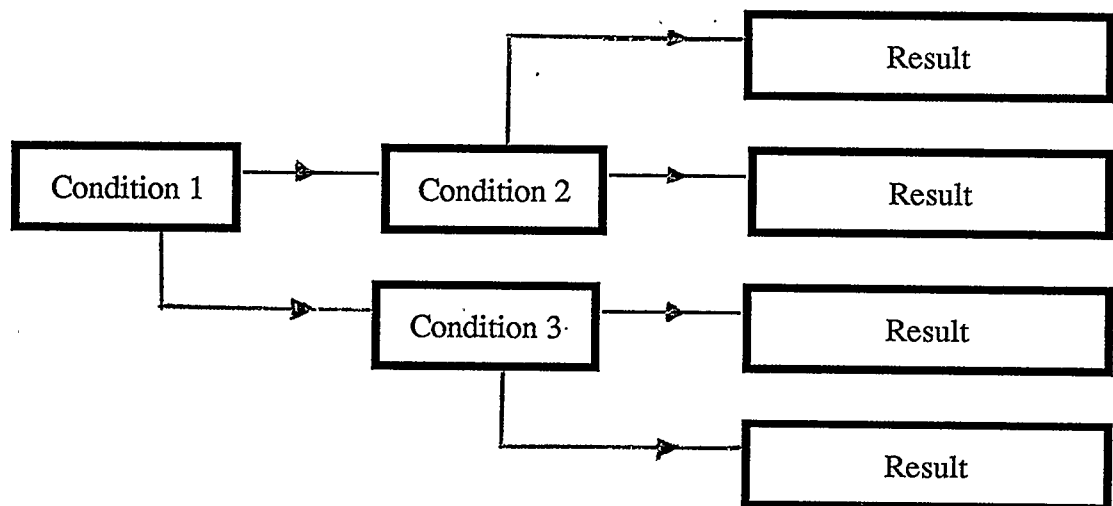
As a further step to facilitate entering the rules, a menu pops up on the screen prompting the rules to be entered in one of the following ways:

- As a new rule, hence fresh typing.
- Similar to the last entered rule, hence editing last rule.
- Similar to any rule in knowledge base; rule will be recalled and edited.

As soon as all rules are entered, "processing" these rules takes place. This is when the program checks which rules are similar, which rules depend on each other to derive information and what are the anticipated results. During this time a wait screen is displayed for a brief moment.

3.3.3 Declarative Method

Unlike the procedural method, the declarative method is based on examples which are "taught" to the system. These examples are based on the knowledge of an expert, and how he performs a certain task considering so many conditions. A tree structure would best allow the visualization of the concept as shown in Fig.3.8. Conditions branch the tree into options, until a result is achieved.



Note: Results may be similar or different

Fig.3.8. Tree Structure

To put this concept in words, it can be described as a number of rules of this form:

If Factor 1 and Factor 2 and Factor 3 and then Result

Each of these factors consists of a specified number of definitions. For example a Factor may be a "traffic signal light", with its three definitions "red", "yellow" or "green". Another factor may be "Pedestrians", with the definitions that it may take "crossing the street" or "not crossing the street". A rule can thus be formed as such:

If the traffic signal light is *green* and pedestrians are *not crossing the street* then a driver can make a right turn.

The factors are underlined while the definitions are shown in italic. To input this into the program, key factors are identified and entered first, as the head of each column. The rows below each factor hold the definitions that the factor can take on. Fig.3.9 shows a schematic of a definition screen.

FACTOR 1 -----	FACTOR 2 -----	RESULTS -----
Definition 1.1	Definition 2.1	Result 1
Definition 1.2	Definition 2.2	Result 2
:	:	:

Fig.3.9. Definition Screen Schematic

Having established these basic definitions, examples are then entered to "teach" the system the logic behind the selection process. For each example to be filled, a number of menus equal to the number of factors, appear sequentially on the screen.

Each of these menus prompts the selection of a certain definition. Using the same above example, a question "What is the color of the traffic signal light" appears, along with a menu with options "red", "yellow" and "green". Pressing "g" from the keyboard, or pointing to "green" using the arrow keys and pressing return, fixes this first example's condition to

If the traffic signal light is *green*

The process is then repeated for the remaining factors, and finally a result is selected in the same manner.

As many examples as specified may be entered to a limit of 8150 examples. This process allows a considerable time saving, compared to manually entering the data. A sample example screen is shown in Fig.3.10.

FACTOR 1	FACTOR 2	RESULTS
-----	-----	-----
Condition 1.1	Condition 2.3	Result 1 Confidence
Condition 1.2	Condition 2.5	Result 2 Confidence
:	:	

Fig.3.10. Example Screen Schematic

Caution must be exercised while filling these examples. The more examples provided, the more complete the knowledge base is. Undefined knowledge will lead to a "no data" result. Now that the examples are prepared, running this expert system entitles just a matching between what the end user has answered and the database to achieve a certain result.

3.3.4 Install Program

Install is a number of procedures that informs the system where the files are to be stored and retrieved. In addition the user may modify a few pre-defined reserved words. Once activated the install screen is displayed, Fig.3.11. In this figure, the program is set to search for French words.

----- SELECT INSTALL FUNCTION -----		
Please select one of the following choices, by pressing one of the letters or using the arrow keys (<- or ->) then pressing return.		
	Currently Defined as -----	
(D)	Change File Directory	FILES
(I)	Modify IF reserved word	si
(T)	Modify THEN reserved word	puis
(E)	Modify ELSE reserved word	sinon
(A)	Modify AND reserved word	et

Fig.3.11. Install Screen

A menu at the top of the screen prompts the user to select among one of the following items to modify:

- The directory path in which the files are to be saved and retrieved.
- The reserved search word for IF in the procedural method. An example would be to define the French word SI, so French sentences may be constructed for the knowledge base instead of English (after redefining all reserved search words as shown in Fig.3.11).

- The reserved search word for THEN in the procedural method.
- The reserved search word for ELSE in the procedural method. As an example OTHERWISE may substitute ELSE.
- The search word for AND in the procedural method.

Caution: provision of a blank before and after the word, ensures that the string is distinct and unique, not to be confused with a similar string in the middle of a word such as **vandal** with **and**.

3.4 File Selection

3.4.1 General

The file selection is at the second menu level in the program. It allows the selection of one of four options: a new file, retrieval of an old file, help and quit options. Fig.3.12 shows the file selection screen.

SELECT TYPE OF FILE

Please select one of the following choices, by pressing one of the letters or using the arrow keys (<- or ->) then pressing return.

(N)	Start a New File
(G)	Get an Existing File
(H)	Help (Get Help)
(Q)	Quit (End Session)

Fig.3.12. File Selection Screen

3.4.2 New File

New file creates a new knowledge base. The title of the knowledge base, the designer's name, and the date are first required. Then text to be displayed in the first and last screens while running the expert system advisor, is prompted to be entered.

In order to reserve the space allocation and perform the proper loop repetitions, the number of rules (or examples) is to be entered. From this point a split in the program flow occurs depending on the choice of method made at the first menu level (declarative or procedural), different menus and questions will appear on the top of the screen to guide the entering of proper information. The method of entering these data was described in subsections 3.3.2 for the procedural method and 3.3.3 for the declarative method. Having entered the data the program proceeds automatically to the Utility Menu level after processing the data.

3.4.3 Old File

This facility permits the retrieval of the old files, in order to run their knowledge bases and to update their data. A list of existing files is exhibited on the top line of the screen, to allow easy selection of the desired file by simply pointing to it and then pressing return. These files are located in the locked up directory as defined to the program in the install procedure. Having retrieved the data the program proceeds automatically to the Utility Menu level.

3.5 Utility Functions

3.5.1 General

The utilities of any software are one of the important foundations for its success, for they provide the hard copies, allow modifications, updating and upgrading of knowledge, as well as access to other programs, languages and software.

The utilities developed for this Spreadsheet Based Expert System, offer a handful of operations, all of which are so easy to perform. The following sub-sections detail each utility operation. Note that Help and Quitting the program are also attainable at this level. Fig.3.13 shows the utility screen.

SELECT UTILITY FUNCTION	

Please select one of the following choices, by pressing one of the letters or using the arrow keys (<- or ->) then pressing return.	
(R)	Run (Knowledge Base)
(S)	Save File
(E)	Edit Data (Modify,Delete,Add)
(P)	Print-out (Report,Output,Rules,Definitions, Examples,Search Statistics,Map)
(H)	Help (Get Help)
(Q)	Quit (End Session)
(A)	Start All over Again
(D)	DOS Access

Fig.3.13. Utility Selection Screen

3.5.2 Run and Again

These are natural links to interconnect the different menu levels available in the program. **Run** links to the operations menu (level Four of the menus), while the **Again** causes the program to start again from the very beginning (first menu level, the method choice selection).

3.5.3 Save

This utility saves the constructed knowledge base, in the locked up directory. This may be achieved by pointing the cursor to overwrite an existing file, among a list that appears as a menu on the top line of the screen, or by typing a fresh new file name. Prior to overwriting an existing file, another chance is given to the user to cancel the command. A spreadsheet file extension format will be automatically assigned in both cases.

3.5.4 Edit

The edit utility leads the program to one of two secondary utility menus depending on the method selected at the beginning of the program (procedural or declarative). Refer to Fig.3.14 for the secondary utility screen.

SELECT EDIT FUNCTION

Please select one of the following choices, by pressing one of the letters or using the arrow keys (<- or ->) then pressing return.

(M)	Modify a Rule
(E)	Erase or Delete a Rule
(R)	Add a Rule
(F)	Add a Factor
(D)	Modify a Definition
(Q)	Return to Utility Menu

Fig.3.14. Secondary Utility Selection Screen

Whenever the user is through with modifications, both menus will go back to the utility menu, after processing and assessing the new modified data. For the procedural method, the menu choice is among the following:

- Modify an existing rule.
- Erase an existing rule.
- Add a new rule.
- Quit edit and return to the main utility menu.

For the declarative method a slightly different menu exists, which allows in addition to those mentioned above, to add or modify either a factor or a definition. Note that should these last two edit operations be performed, the whole database needs to be re-adjusted to include them.

3.5.5 Print

Again this utility leads the program flow to one of two print-out menus, depending on the method selected (procedural or declarative). Fig.3.15 shows the print-out selection screen.

SELECT PRINT FUNCTION	
Please select one of the following choices, by pressing one of the letters or using the arrow keys (<- or ->) then pressing return.	
<u>PROCEDURAL METHOD</u>	<u>DECLARATIVE METHOD</u>
(R) Print-out Report on Session	(R) Print-out Report on Session
(O) Print-out Output Screen	(D) Print-out Definitions
(F) Print-out Rules in Formatted Form	(E) Print-out Examples
(U) Print-out Rules in Non-Formatted Form	(X) Print-out Exhaustive Search
(Q) Return to Utility	(O) Print-out Optimum Search
	(S) Print-out Statistics
	(M) Print-out Map of Chained Files
	(Q) Return to the Utility

Fig.3.15. Print-out Screen

For the procedural method the allowed options of printing-out are:

- Report, which is the "dialogue" that took effect during the session, between the user and the computer. Fig.3.16 is a sample dialogue.
- Output Screen, which shows the results or the outcomes the system advises, with the corresponding confidence in each result. Fig.3.17 is a sample output screen.
- Formatted Rules, which are the knowledge base rules parsed by the template to maintain the important knowledge. Fig.3.2 in subsection 3.2 is a sample.

- Non-Formatted Rules, which are the actual rules entered into the system by the knowledge engineer. A sample was shown in Fig.3.3 in subsection 3.2.
- Quit, ends the print-out session and returns to the Utility Menu.

SESSION SUMMARY	

Wind velocity is usually less than 50 km/hr	True
Wind velocity is usually between 50&120 km/hr	False
Wind velocity frequently goes above 120 km/hr	False
Rain is frequent	True
Snow is frequent	False
Fog is frequent	False
Noise is high	False
Small animals are around	True
Medium animals are around	False
Large birds are around	False
Lightning is frequent	False
Overhead transmission cables are close	True
Underground transmission cables are close	False
There is no clear line of sight between detectors	False
Terrain is not flat	False
Low maintenance is required	True
Low cost is desired	False
Ground is soft	True
Seismic effects are around(railways,etc.)	False
Aesthetics are important	True

Fig.3.16. Sample Report Screen

RESULT SCREEN	

<u>RESULT</u>	<u>PROBABILITY %</u>
Use microwave detection system	73
Use infrared detection system	79
Use electric field detection system	0
Use fence motion detection system	0
Use taut wire detection system	0
Use seismic detection system	0
Use magnetic detection system	64
Use seismic-magnetic detection system	65
Use ported coaxial detection system	83
Use video motion detection system	63
Use laser detection system	92

Fig.3.17. Sample Output Screen

As for the print-out operations available for the procedural method, the options are:

- Report, which is the "dialogue" that took effect during the session, between the user and the computer. Fig.3.18 is a sample dialogue, for a shipping advisor. According to the weight of the package, the speed of delivery, delivery day requirement and delivery time, a specific shipping company is recommended. This example advises the use of Federal Express, in accordance to the data entered.
- Definitions, which are the factors, definitions and results that the knowledge engineer enters. Fig.3.19 is a sample of the print-out of the definitions for the same example described above. In this example the knowledge engineer has defined four factors, the weight of the package *WEIGHT*, the speed of package delivery *SPEED*, whether Saturday delivery is required or not *SAT-DEL*, and whether the time of delivery required is before ten *TIME*. Under each of these

factors, a few definitions are used. The last two columns are reserved for the results and comments. A list of normally utilized shipping companies is entered in the result column, with a specific comment about each entered in the comments column.

- Examples, which is the knowledge "taught" to the system in the form of examples. Fig.3.20 is a sample of the print-out of examples. Using the same shipping advisor described above, example one in Fig.3.20 reads as follows:

If the **WEIGHT** of the package is *Below One Kilogram* and the **SPEED** of Delivery is *One Day* and **SATURDAY DELIVERY** is *Required* and the **TIME** the Package is Required is By *Ten* then **SHIP BY Express** with a **CONFIDENCE** of one.

- Exhaustive, which are all the possible combinations of search presented in a tree formation. Fig.3.21 is a partial sample of the print-out of exhaustive search.
- Optimum, which only prints-out those branches of the tree that would lead to a result. During this process, the SES performs a database search, to extract the conditions that lead to each of the results. It starts with the first result, from the list of the results. For each result, SES identifies the conditions that have to be met in order to achieve this particular result. Fig.3.22 is a sample of the print-out of an optimum search.

- Statistics, prints-out the performed statistics on the database of the system. Fig.3.23 is a sample of the print-out of statistics, using the same example of the shipping advisor. It shows the total number of examples in the knowledge base, the total weight of the confidence of all the examples, how many times did each example occur, the average weight of each example, the frequency of occurrence, and in which examples does each result occur.
- Map, prints-out out the road map of how files are chained together. Fig.3.24 is a sample of the print-out of map. In this example the current file called *F_CHAIN* is forward chained to two files *!Otis* and *!Schindler*.

RULES TO BE SEARCHED			

<u>WEIGHT</u>	<u>SPEED</u>	<u>SAT_DEL</u>	<u>TIME</u>
below-1-kg	one-day	yes	ten
	<u>Ship By</u>	<u>Confidence</u>	
	Federal	4	
Comments : Cost is expensive			
SESSION SUMMARY			

What is the weight of the package in kgs?		below-1-kg	
How fast should package be delivered?		one-day	
Is Saturday delivery required?		yes	
What time is package required?		ten-o'clock	

Fig.3.18. Sample Report Screen

DEFINITION SCREEN

<u>Weight</u>	<u>Speed</u>	<u>Sat_Del</u>	<u>Time</u>	<u>Ship_By</u>	<u>Comments</u>
below-1-kg	one-day	yes	ten	Express	This very cheap
one&2-kg	two-days	no	later	Courier	Best Service
two&20-kg	three-days			Parcel	Explanation
over_20-kg	six-days			DHL	Only in Europe
				UPS	Another reason
				Mail	Cheapest
				Federal	Expensive

Fig.3.19. Sample Definition Screen

<u>No</u>	<u>Weight</u>	<u>Speed</u>	<u>Sat_Del</u>	<u>Time</u>	<u>Ship_By</u>	<u>Confidence</u>
1	below_1	one	yes	ten	Express	1
2	below_1	one	yes	later	Courier	1
3	one&2	one	yes	ten	Express	1
4	one&2	one	yes	later	Parcel	1
5	two&20	one	no	ten	Express	1
6	two&20	one	no	later	DHL	12
7	two&20	two	yes	ten	UPS	2
8	two&20	two	no	later	Mail	23
9	two&20	three	no	later	Mail	3
10	one&2	one	yes	ten	Federal	4
1	over_20	one	yes	ten	Parcel	5
12	over_20	two	no	later	Mail	1
13	over_20	six	no	later	Mail	2
14	over_20	six	no	ten	Parcel	2

Fig.3.20. Sample Example Screen

EXHAUSTIVE SEARCH SCHEMATIC				

<u>Weight</u>	<u>Speed</u>	<u>Sat Del</u>	<u>Time</u>	<u>Ship By</u>
below_1	one	yes	ten	-----Express
			later	-----Courier
		no	ten	-----Federal
			later	-----no_data
	two	yes	ten	-----no_data
			later	-----no_data
		no	ten	-----no_data
			later	-----no_data
	three	yes	ten	-----no_data
			later	-----no_data
		no	ten	-----no_data
			later	-----mail
six	yes	ten	-----mail	
		later	-----mail	
	no	ten	-----mail	
		later	-----no_data	
one&2	one	yes	ten	-----Express
~~~~~				

Fig.3.21. Sample Exhaustive Search Screen

<u>Weight</u>	<u>Speed</u>	<u>Sat Del</u>	<u>Time</u>	<u>Ship By</u>
below_1	one	yes	ten	-----Express
one&2	one	yes	ten	
two&20	one	no	ten	
below_1	one	yes	later	-----Courier
one&2	one	yes	later	-----Parcel
over_20	one	yes	ten	
over_20	six	no	ten	
two&20	one	no	later	-----DHL
two&20	two	yes	ten	-----UPS
two&20	two	no	later	-----Mail
two&20	three	no	later	
over_20	two	no	later	
over_20	six	no	later	
one&2	one	yes	ten	-----Federal

Fig.3.22. Sample Optimum Search Screen

STATISTICS SCREEN				
*****				
Number of Examples: 14		Total Weight: 59		
<u>Result</u>	<u>Occurrence</u>	<u>Avg.Wt</u>	<u>Freq.</u>	<u>Occurred In Examples</u>
Express	3	1.0	21.4%	1,3,5
Courier	1	1.0	7.1%	2
Parcel	3	2.7	21.4%	4,11,14
DHL	1	12.0	7.1%	6
UPS	1	2.0	7.1%	7
Mail	4	7.3	28.6%	8,9,12,13
Federal	1	4.0	7.1%	10

Fig.3.23. Sample Statistics Screen

MAP SCREEN	
*****	
Backward Chained to Current File Forward Chained to	
F_CHAIN	-----!Otis
	-----!Schindler

Fig.3.24. Sample Map Screen

### 3.5.6 DOS Access

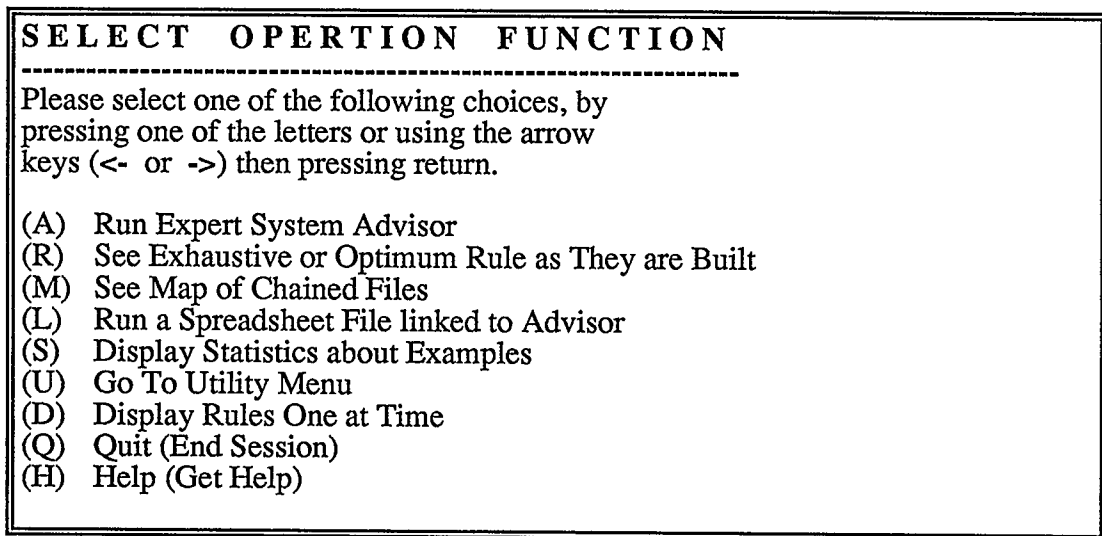
This utility opens a temporary access to DOS. A batch file may be written in DOS to execute certain commands, or run a program as an external link to the expert system. The last command of the batch file should be "exit" to return to the Utility Menu.

## 3.6 Operation Functions

### 3.6.1 General

The Operation functions are at the last menu level. They constitute the core of

this program. The inference engine, which is the driving search mechanism to reach a certain outcome, is accessed through the advisor. Other functions result in acquiring some information about the knowledge base, which may be very important. Fig.3.25 is the screen of the Operation menu.



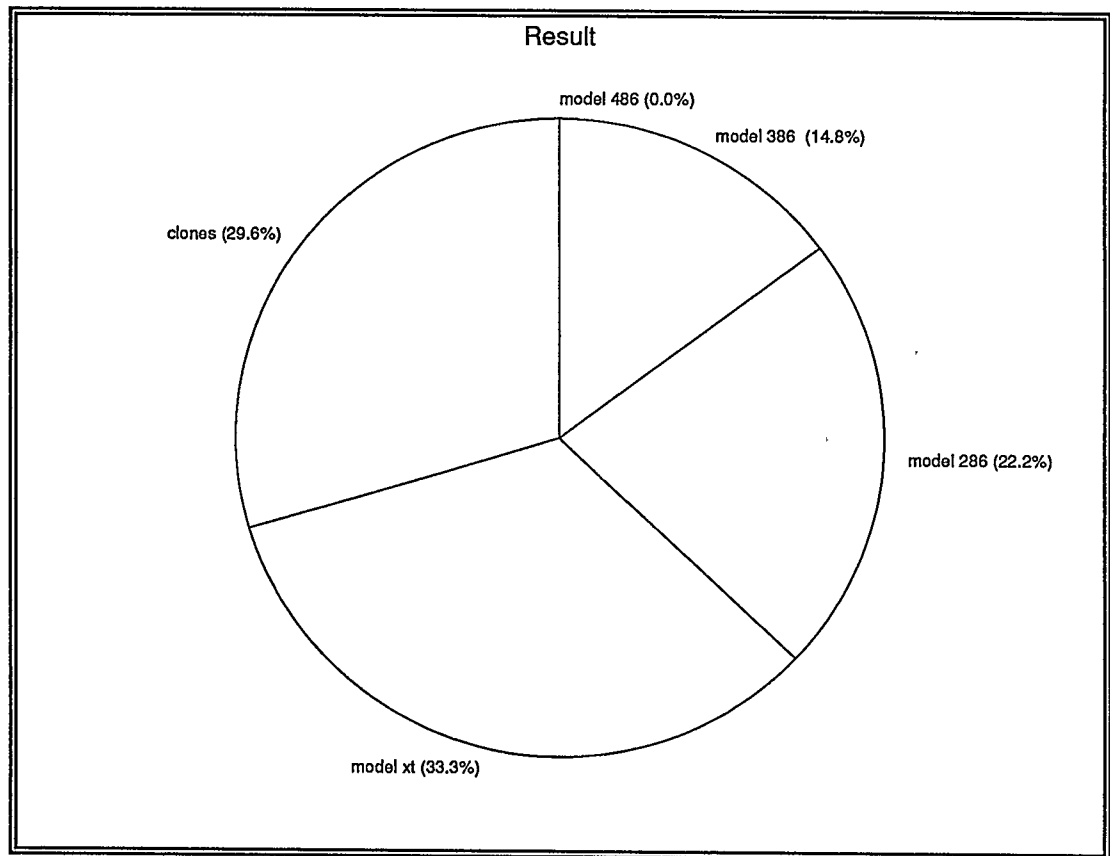
**Fig.3.25. Operation Screen**

### **3.6.2 Advisor**

The advisor for both the procedural and the declarative methods, is structured in the following format:

- a- Display the beginning text screen, describing the context of and expectations from the expert system.
- b- Man-Machine interaction, where the user answers a few questions according to the particular situation he faces. Fig.3.16 and Fig.3.18 in subsection 3.5.5 are samples of such "dialogues".

- c- Search for a result(s) by the inference engine in accordance with particulars asked for by the user.
- d- Display of end text screen, which explains what to expect from the results.
- e- Actual display of results or outcomes, with associated confidence levels in each. Fig.3.17 and Fig.3.18 in subsection 3.5.5 are samples of such outcomes.
- f- A menu pops up on top of the screen prompting if an automatic plot of the results or reasoning of any of the results is needed. If plot is chosen, a pie chart is displayed with each sector being one of the results. Overall confidence levels comparing particular results to the rest are also indicated. Fig.3.26 is a typical result pie plot.



**Fig.3.26. Pie Plot of Results**

If the reasoning option is selected from the menu, the cursor movement keys permit choosing one of the results. The "history" behind the choice made is then displayed on the screen, listing all the rules that lead to this result and how these rules were answered. Fig.3.27 is an example, where the computer explains why it has given the choice "Microwave Detection System", the probability of 73 percent. A list of the number of the rules that lead to this outcome is displayed. The user is then to press return to display those rules one at a time, at the bottom portion of the screen.

**REASONING SCREEN**

*****

The Result Under Question is :  
 Information About this Result was  
 derived from examples :

*Use Microwave Detection System*  
*1, 5, 9, 13, 17, 21, 25, 29, 33,*  
*39, 43, 44, 49, 55, 56.*

Rule No *13* was Answered *True*, Stated:

If Rain Is frequent Then *Use Microwave Detection System* and Use Infrared  
 detection System.

The Probability is 70 and 80 percent respectively.

PRESS RETURN TO DISPLAY NEXT RULE.....

**Fig.3.27. Reasoning Screen**

While running the procedural method advisor, in addition to the above mentioned steps, further options of the method of search are available:

- Backward Chaining.
- Forward Chaining.
- No Backward Chaining.

Fig.3.28 shows the selection screen. Considering the same two examples shown in Fig.3.28, the backward chaining search first determines where the results are. Rule number one has a result, and hence SES tries to validate this rule. In doing so the system first checks if validity of this rule may be derived from others. In this case rule number two is used to derive information about the first rule. Truth about the c condition implies truth about a. Hence rule number two is asked and **RESULT** is thus obtained.

In the forward search option, all the rules are checked in the order they are entered, and hence in this example, rule one is first asked then rule two. The no

backward search option on the other hand, discards all the rules that do not lead to a result. In this example rule number two is not asked, instead validity of rule number one is directly checked.

### SEARCH METHODS

*****

(B) Backward Search: Searches those rules that will  
produce a result. Allows Backward Chaining.

Example: Consider these rules :

1) if a then **RESULT**    2) if c then a

System asks about c if true or not to get the **RESULT**

(F) Forward Search: Searches all rules in order.

(N) No Backward Search: Searches those rules that will  
produce a result. Does NOT  
Allow Backward Chaining.

Example: Consider the same rules as above:

System asks about a if true or not to get the **RESULT**

**Fig.3.28. Search Methods Screen**

As for running the declarative method advisor, if a factor title has an exclamation mark at the first letter (!, example: !Color), this would entail the system to back chain to a knowledge base (!Color in this example), to derive information about the current factor. However, a manual override is available to discard such a back chain if the user already knows the outcome. Similarly an exclamation mark at the first letter of a result, would cause a forward chaining to occur. This feature would entitle knowledge bases to be constructed in such a manner as to ask a few questions, get a result, then chain to another file to ask more particulars and get specific answers. It also permits construction of smaller knowledge bases, which are easier to debug. Manual overriding the chaining is also permissible in this case. Attention should be made to have all these chained files in the active file directory as



set in the install procedure.

### **3.6.3 Rule**

This function is only available for the declarative search method. Two options exist: an exhaustive or an optimum search, refer to Fig.3.21 and 3.22 in subsection 3.5.5. The former is a complete tree of the knowledge base, where all possible combinations whether leading to a result or not are shown, while the latter displays just the branches in which a result exist. This is an important tool for debugging knowledge bases; the tree structure points out where no data was taught to the system, hence these knowledge gaps may be filled, while the optimum points out the routes that lead to a specific result, so any missing route may be easily spotted then added.

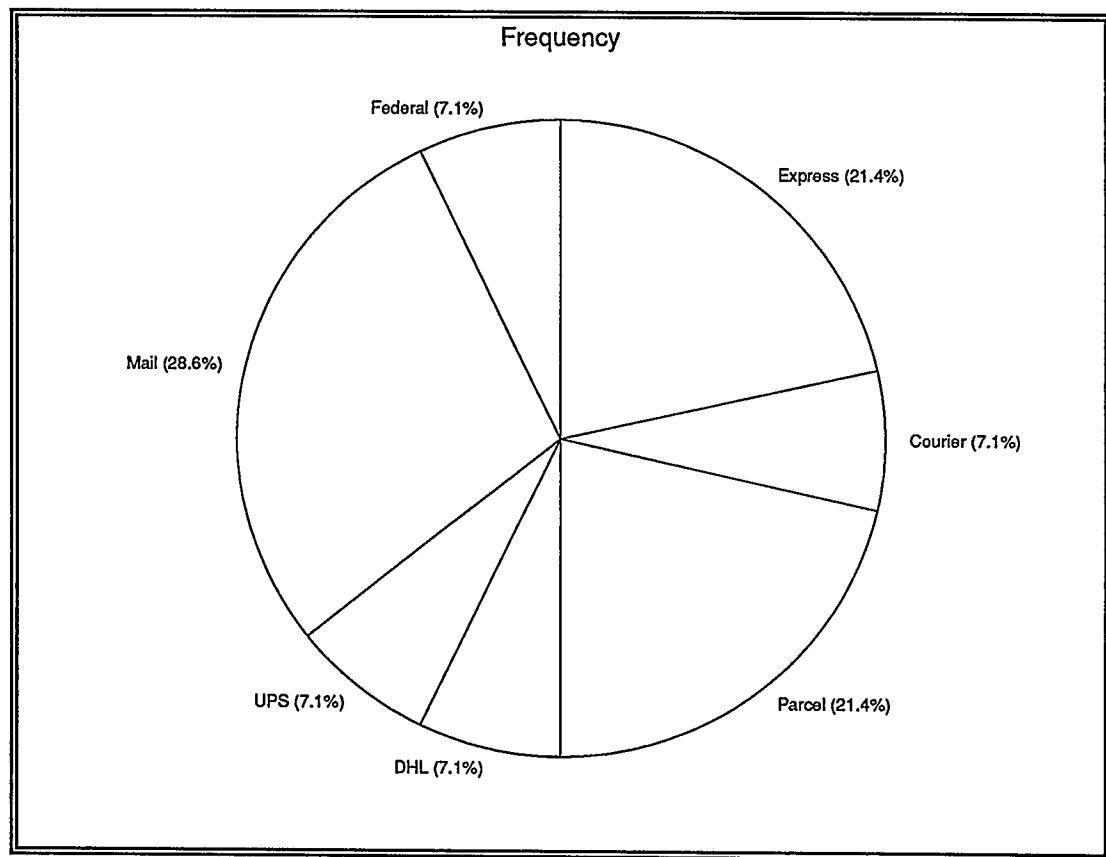
### **3.6.4 Map**

This function is also only available for the declarative search method. It constitutes a map displaying which files are back chained or forward chained to the current file, refer to Fig.3.24 in subsection 3.5.5 for a sample. These files are clearly identified, for the user to make sure that they are available in the working directory, otherwise backward and forward chaining will cause an error screen. Also they provide a clear view for the user to foresee where the program is leading him.

### 3.6.5 Statistics

Again this function is only available for the declarative search method. Activating the statistics function yields these options:

- Screen, where information about all the database is disputed, such as the total number of examples, the total confidence (weight) of the examples, the results with how many times and where they occurred, their percentage of occurrence and average weight. Fig.3.23 in subsection 3.5.5 is a sample statistics screen.
- Window, opens a window in the screen to provide statistical information about one particular result. Specially designed menus allow navigation in between the screens of definitions, examples, exhaustive and optimum search. Pointing at any of the results display all statistics about it in the window.
- Plot, a pie chart plot may be automatically generated for the results, with average weight or frequency of occurrence as the portions of the sector. A sample plot is shown in Fig.3.29.
- Quit, ends the statistics menu and returns control to the Operation menu.



**Fig.3.29. Pie Plot of Statistics**

### **3.6.6 Utility**

This function simply permits backing up to the Utility Menu to perform any of the utility functions. It is a part of the navigation links in the system.

### **3.6.7 Display**

The display function is only available for the procedural search method. It is a means to display on screen all rules for checking purposes. Rules are displayed in ascending order in both the non-formatted (the way they were entered) and formatted

forms. A hard copy is attainable through the print-out option.

### **3.6.8 Compile**

Once an expert system is completed, this function may compile the knowledge base. The file hence becomes directly executable. The file may be retrieved through the Lotus 1-2-3 file retrieve command, and automatically the expert system starts to acquire particulars about the specifics of the project. Modifications to the expert system or rule additions, may not be carried through a compiled file. The knowledge base has to be re-loaded into SES, then modifications to be carried out, and finally the file needs to be re-compiled.

## **3.7 Help Functions**

Help is attainable at any level. This on-line facility once activated provides the required help, as it detects the location where it was asked for in the program, and displays the relevant screen of help pertinent to that part. However additional help about any part of the program may also be obtained through a help index, Fig.3.30, and a schematic of the program may be displayed to ease navigation.

HELP INDEX			
*****			
SCHEMATIC...		Operation.....	Advisor
		.....	Rule
METHOD.....	Procedural	.....	Map
.....	Declarative	.....	Statistics
.....	Install	.....	Utility
FILE.....	New		
.....	Get_File		
		SEARCH	
		METHOD.....	Backward
			:Forward
			:No_Backward
UTILITY.....	Run		
.....	Save		
.....	Edit.....	Modify	
.....	Print-out	..... Delete	
.....	Again	..... Add	
.....	DOS		

Fig.3.30. Help Index

### 3.8 Validation of User Friendliness

Two individuals who have never developed expert systems were asked to try the SES. The first was a graduate student who was not familiar with Lotus 1-2-3, while the second was an employee in a life insurance company that had a fair command of Lotus 1-2-3.

Both individuals were asked to develop two simple knowledge bases, using in the first advisor the procedural method and the declarative method in the second. The advisors recommend the best personal computer for different applications. There are three possible outcomes and five factors influencing the decision.

The graduate student required eleven minutes to develop the knowledge base, as opposed to thirteen for the life insurance employee. Both required around four minutes to run the advisor.

The declarative method consumed fourteen and seventeen minutes respectively to develop the knowledge base, and only two minutes to run the advisor. The other facilities such as obtaining statistics and displaying the tree structure were easily conducted.

Both individuals expressed their content and thought that SES is easy to use and friendly. It is worthwhile to note that after the experiment the graduate student was able to immediately identify and enter a similar application of his own.

### **3.9 Comparison with Other Shells**

#### **3.9.1 General**

Utilizing a Maxar 386 micro computer (IBM compatible) running at sixteen kilohertz, two knowledge bases were prepared; one a rule based while the other an example based. These knowledge bases were then entered into SES and one in each of 1st Class and Exsys. Subsections 3.9.2 and 3.9.3 show the results of the comparison.

### 3.9.2 Comparison of the Procedural Method with Exsys

An access control advisor was used. The advisor is fully described in subsection 4.7. The same rules were entered into the two systems, and the two advisors were run. The questions and answers for both expert systems are shown in Fig.4.7. The output of Exsys is shown in Fig.3.31, while that of SES in Fig.4.8.

Values based on 0 - 10 system		VALUE
1	Weigand	7
2	Magnetic Stripe	3

**Fig.3.31. Exsys Output for the Access Control Advisor**

Comparing the two outputs reveals that the results are identical, except that Exsys uses a scale of one to ten and suppresses the display of ruled out outcomes such as barium ferrite cards, keys and proximity cards. Table 3.1 lists a comparison in speed of operation.

	Exsys	SES
Data Entry	12 min	9 min
Data Compilation	2.4 sec	9.6 sec
Obtaining Results	2.7 sec	11.2 sec

**Table 3.1. SES and Exsys Speed Comparison**

It is noted that Exsys outperforms SES in the speed of compilation and obtaining results. However SES requires less time in data entry. This is attributed to the simplistic approach of entering data used in SES on the expense of longer processing time.

### 3.9.3 Comparison of the Declarative Method with 1st Class

A wine selection advisor consisting of two files back-chained to each other, having 63 and 12 rules respectively, was prepared and entered into 1st Class and SES, Fig.3.32. The advisor selected a suitable wine depending on personal taste and type of meal. The two advisors were then run. The output screens are shown in Fig.3.33 and Fig.3.34. respectively.

	<u>Color</u>	<u>Sweetness</u>	<u>Body</u>	<u>Spiciness</u>	<u>Suggestion</u>	<u>Weight</u>
1	red	medium	medium	spicy	gamay	0.8
2	red	medium	medium	not-spicy	gamay	0.8
3	red	sweet	medium	spicy	chablis	1.0
4	red	sweet	medium	not-spicy	chablis	1.5
5	white	dry	light	spicy	sauv-blanc	1.0
~~~~~						

Fig.3.32. Partial Wine Selection Database

The wine(s) that you should consider for your meal are:
 chardonney with confidence of 54.55%
 -- or --
 sauvignon-blanc with confidence of 45.45%

Fig.3.33. 1stClass Wine Advisor Output

RULES TO BE SEARCHED			

<u>!COLOR</u>	<u>SWEETNESS</u>	<u>BODY</u>	<u>SPICINESS</u>
white	dry	medium	spicy
		<u>SUGGESTION</u>	<u>Confidence</u>
		Sauv-blanc	1
		Chadonnay	1.2
Comments : Excellent choice of white wine.			

Fig.3.34. SES Wine Advisor Output

Again the results are identical except that 1st Class expresses the final confidence in per cent. In SES the outcomes are expressed in percentage when the plot of the results is conducted, Fig.3.35. Table 3.2 lists a comparison in speed of operation.

	1 st Class	SES
Data Entry	17 min	13 min
Data Compilation	0.5 sec	2.9 sec
Obtaining Results	0.2 sec	0.4 sec

Table 3.2. SES and 1stClass Speed Comparison

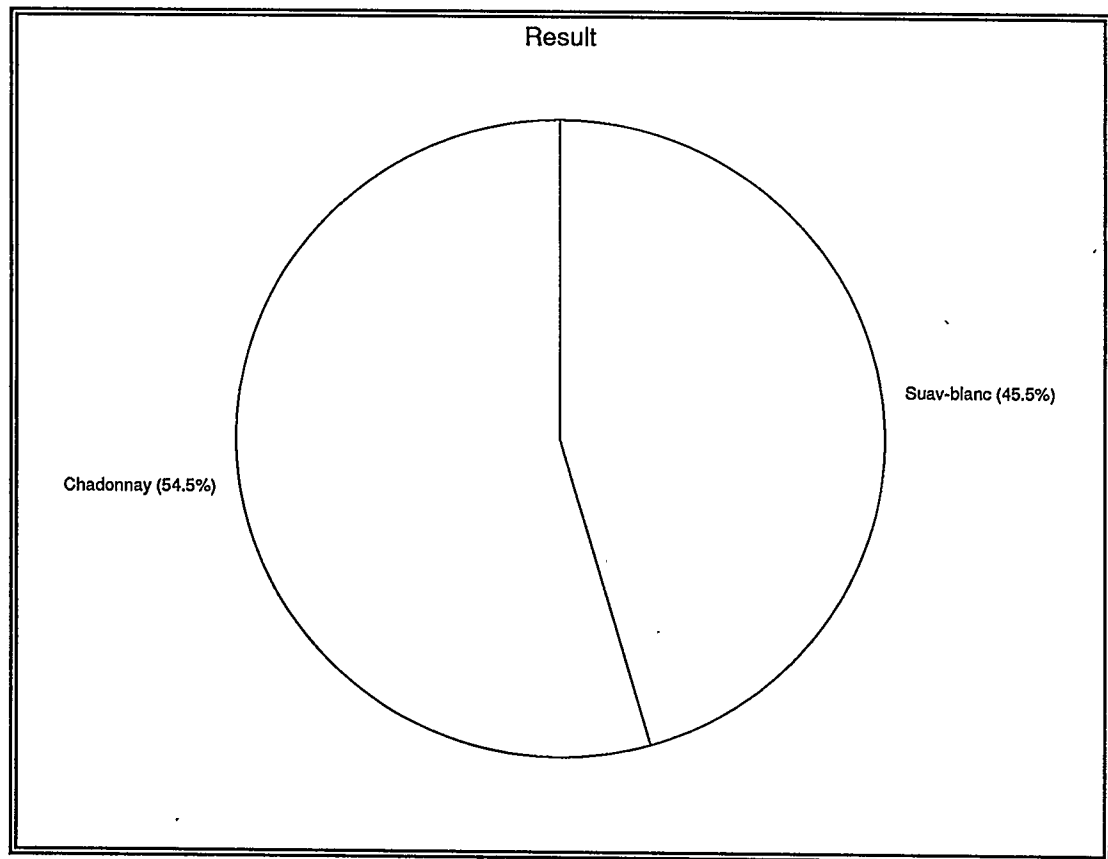


Fig.3.35. Plot of Wine Advisor Output

1st Class is a little faster than SES in speed of compilation and obtaining results. However, SES requires less time in data entry. As in the case of Exsys, this is attributed to the simplistic approach of entering data used in SES on the expense of longer processing time. It should be noted that without chaining for both SES and 1st Class, the maximum SES knowledge base size is around 32 times larger than that of 1st Class.

3.10 System Advantages and Disadvantages

The advantages of the SES may be divided into two parts. The first set of advantages are due to the fact that it is written in a spreadsheet environment, the second set arises from the new features that the SES offers. The advantages due to the spreadsheet environment are as follows:

- There are eight million Lotus 1-2-3 spreadsheet users, plus a large number of users utilizing compatible spreadsheets to that of Lotus 1-2-3, who may easily add and utilize this package.
- SES may be used wherever Lotus 1-2-3 is installed. The software, which is written mostly in C language, currently runs on IBM personal computers, using either DOS or OS/2 operating systems. With the recent new release that was introduced, it now runs on the IBM S/370 and DEC Vax/VMS machines. As a consequence SES may be used as well in these larger machines.
- SES is written using the built-in Lotus 1-2-3 programming language. Hence the software can be upgraded to any future release without any modification (since all newer releases can read the code of their predecessors).
- SES provides one single confined environment where data, algorithms, knowledge and an inference engine coexist.
- SES follows the strategy of spreadsheets of user friendliness, availability at all times of an on-line help facility and depending on menus to interact with the user.

- One of the important features that many expert system shells stress is compatibility of data transfer with Lotus 1-2-3. SES is built in Lotus 1-2-3 and hence is one hundred percent compatible. In addition this feature ensures reading the knowledge of these shells that can exchange data with 1-2-3.

The advantages offered by the specifics of the SES are the following:

- SES provides a utility subroutine that permits the development of expert systems in any Latin based language.
- The rules are entered into SES as natural language sentences. They are then automatically parsed into pieces of information by the software.
- SES provides two distinct methods for developing expert systems, the procedural rule and the declarative example based methods. Expert systems may be developed using any one or a combination of the two methods.
- The declarative method allows 8,150 examples and 256 factors in any single file, which may also be chained to other equally sized files to form an expert system. Other shells utilizing the same development method, have usually a much smaller capacity.
- Searching for an example in the knowledge base of the declarative method, is carried out using the matching criteria of the database functions of Lotus 1-2-3. This search method proves to be extremely fast, and independent of the size of the database.

- SES permits communication, data transfer and execution of other programs while running an expert system.
- In some of the other shells, a limit of eight characters in defining any result, factor or definition is imposed. On the other hand SES automatically detects the length of the result, factor or definition, and sizes the column accordingly.
- A generous number of debugging, validating and reporting facilities are provided in SES.
- SES occupies only around 120 kilo-Bytes of memory. A typical average sized knowledge base file requires around 50 kilo-Bytes. As a consequence, the availability of around 200 kilo-Bytes of free memory space is usually enough.

SES on the other hand has some limitation and disadvantages. These are the following:

- When used on Lotus 1-2-3 release 3, the procedural method of search in the SES is slow in execution. This is attributed to the fact that this new release of Lotus 1-2-3 is faster in few operations, but much slower in the database extract commands that SES depends on. This slow execution is not noticed while using releases 2.0, 2.1 or 2.2.
- Spelling mistakes while developing a procedural rule based expert system, cause a string mismatch. A consequence is that backward chaining does not occur in this case. For example considering these two rules:

IF **string one** THEN **string two**

IF **misspelled string two** THEN *Result*

Usually if there is no spelling mistake, backward chaining forces SES to validate **string one** to derive information about the *Result*. However with a spelling mistake **string two** and **misspelled sting two** are not identical according to SES, and the system in its effort to derive information about the *Result*, tries to validate **misspelled string two** instead.

- Only two logical operators are allowed in between the phrases of the THEN and ELSE parts. This limit is a compromise between the size of a rule and the speed of the search. To overcome this shortcoming, a large rule may be split into parts and entered as smaller rules. For example considering this rule:

IF condition THEN **a** and **b** and **c** and **d** and **e** ELSE **f**

may be split as such:

IF condition THEN **a** and **b** and **c** ELSE **f**

IF condition THEN **d** and **e**

- For computing the confidence of a certain result in the procedural method, only the average of all particular probabilities is available. This is purposely done to ensure a safe means of obtaining feasible confidence in the results, whether the rules are dependent or independent.

- In the declarative method SES asks all possible questions in the tree structure. Other shells provide an optimum search method where few questions may be dropped out. Consider the following simple medical example,

<u>FEVER</u>	<u>RUNNING NOSE</u>	<u>RESULT</u>
Yes	Yes	Flu
No	Yes	Cold

This is an incomplete knowledge base. Using the optimum search method in these shells, asking the second question is not necessary, since both examples have a yes for the running nose condition. A person who has no fever is directly concluded to have a cold. SES in turn asks both questions, and returns the label "no-data" for a person that has neither a fever nor a running nose. This ensures that missing databases cause no wrong answers.

4 APPLICATION TO SECURITY EXPERT SYSTEMS

4.1 Introduction

To demonstrate the flexibility and power of the Spreadsheet Based Expert Shell, some practical applications were developed.

The first application is for a security advisor to recommend the most appropriate indoor and outdoor intrusion detection devices, cameras and access control for any particular application. This expert system consists of four separate advisors for each of the above mentioned security systems, developed using the procedural method. These four advisors are controlled by a manager program developed with the declarative method. This expert system is an example of the use of the combination of the two methods, where the declarative method's capability of file chaining is utilized to manage and control decisions.

4.2 System Overview

The knowledge of these systems was acquired while designing and installing a number of security projects.

The integrity of the complete security system may be jeopardized, in the process of design, by just overlooking one "hole" in the system. The complete system reliability is as good as its weakest part, and hence judgement that the design is a success or a failure depends on every single component.

Proper design of a security system involves a complex interweaving of multiple concurrent considerations. These include, but are not limited to, environmental conditions, specifics about country, location and activities peculiar to each site, the weather conditions, the physical layout, the type of application, the levels in the terrain and type of soil, and aesthetics. Without an overall prescribed procedure, the system design can become quite segmented, with penetration possible and target accessible [31].

4.3 System Operation

To operate the system, Lotus 1-2-3 must be in the ready mode. SES is then loaded into memory by retrieving the file called *SES*. This is achieved by pressing the slash (/), File then **R**etrieve (Lotus commands) from the keyboard, then pointing to and selecting the file *SES* from the menu that appears.

The program's logo is then displayed, after which the **D**eclarative method and **G**et-File options are to be selected from the menus that appear. The next step is to choose the file *Security* from among the list of files exhibited on the screen. This file acts as the manager to control the flow of the program. This is achieved through a series of questions that appear on the screen asking if individual security systems, such as outdoor or indoor intrusion detection, are required. A choice of whether to chain to these files or to discard the request is possible, depending upon whether these systems are required or not. For example, if no monitoring in the security system is needed, the camera selection advisor is to be discarded.

Chaining to any of the particular security advisors, places the advisor's file in memory, ready to be executed. This process starts by displaying particular information about the advisor and expectations from the outcome. The user is then to select from among the methods of search available, the backward, forward or no-backward chaining. According to the way the knowledge base was prepared in this application, the backward chaining method is recommended.

Having selected the method, the program starts asking questions about the peculiar specifics of the project. The user is expected to answer these questions from the menu choice that appears on the screen, having the facility of finding out why any certain question is being asked. These questions-answers are shown in Figures 4.1, 4.3, 4.5 and 4.7 for each of the security advisors. Depending on the answers, the system assesses the probability or confidence of each of the possible outcomes and displays them in the result screens as shown in figures 4.2, 4.4, 4.6 and 4.8. Plotting these results is then possible in the form of a pie chart.

This process is repeated for the four security advisors after which the program is halted.

4.4 Outdoor Intrusion Advisor

Eleven different outdoor intrusion detection systems are usually considered in the design process [32]. Each offers certain advantages under specific conditions peculiar to the site under consideration. The optimum system is the one that offers the least false alarm rate under normal conditions, and highest detection rate when intrusion

occurs. To achieve this, environmental conditions pertinent to the locale, existence of animals and birds, electrical power cables, nature of terrain, soil and level of ground, cost, maintenance, seismic effects around and aesthetics are considered.

Each of these constraints increases, decreases or eliminates the chances of selecting a system over another. This suggests that a certain probability is assigned to each of these constraints, and hence the procedural method is the most appropriate, as it caters for a probability over each condition, with the overall confidence revealed at the end.

The questions asked by the system are displayed in Fig.4.1, while a typical outcome (in accordance with the answers) in Fig.4.2.

SESSION SUMMARY

Wind velocity is usually less than 50 km/hr	True
Wind velocity is usually between 50&120 km/hr	False
Wind velocity frequently goes above 120 km/hr	False
Rain is frequent	True
Snow is frequent	False
Fog is frequent	False
Noise is high	False
Small animals are around	True
Medium animals are around	False
Large birds are around	False
Lightning is frequent	False
Overhead transmission cables are close	True
Underground transmission cables are close	False
There is no clear line of sight between detectors	False
Terrain is not flat	False
Low maintenance is required	True
Low cost is desired	False
Ground is soft	True
Seismic effects are around(railways,etc.)	False
Aesthetics are important	True

Fig.4.1. Outdoor Intrusion Question-Answer

RESULT SCREEN

<u>RESULT</u>	<u>PROBABILITY %</u>
Use microwave detection system	73
Use infrared detection system	79
Use electric field detection system	0
Use fence motion detection system	0
Use taut wire detection system	0
Use seismic detection system	0
Use magnetic detection system	64
Use seismic-magnetic detection system	65
Use ported coaxial detection system	83
Use video motion detection system	63
Use laser detection system	92

Fig.4.2. Outdoor Intrusion Summary

The following is a brief description of the principle of operation of each of the eleven systems, with some of their advantages and disadvantages:

- Microwave intrusion detection devices, of which there are three main types, all of which employ antennas. All microwave systems have a good range with a good beam spread. They are easy to set up, difficult to evade and, apart from cleaning, have a low maintenance factor. Disadvantages arise through the wide beam (space required), line of site restrictions, and moving objects in the vicinity.
- Infrared intrusion detection device, which offers a good narrow beam, easy to set up, hard to defeat and requires low maintenance. Problems arise with fog, snow, infrared sources and moving objects in the vicinity.
- Electric field intrusion detection device, which can be either above or below the ground. It operates on either one of the principles of a change in capacitance in wires, or when the electric field coupling is distorted. Again there is a space requirement, a "sterile" area is vital (no trees or any moving objects), maintenance is time consuming and aesthetically is not elegant.
- Fence motion intrusion detection device, which consists of wires or cables mounted on a separate fence. This system operates on one of three principles mechanical, electromechanical or strain sensitive cable. Climbing the fence causes tension or break in the wires, and hence an alarm condition. Wires must be broken or badly distorted to cause an alarm. Vandalism and accidental breaks are a nuisance, since the wires need to be fixed or changed, re-installed and the system re-tested.

- Taut wire intrusion detection device, which is similar to the fence motion sensor, but is itself a fence, free standing or as a part of another structure. It can operate on either a spring action tension wires or fibre optics. This system is immune to radio frequencies, highly tamper resistant, has long range and requires virtually no maintenance. However, as in the case of a fence motion detection, wires must be broken or badly distorted to cause an alarm. Vandalism and accidental breaks are a nuisance, since the wires need to be fixed or changed, re-installed and the system re-tested.
- Seismic intrusion detection device, is an underground system, which may be buried under hard surfaces. It operates on the principle of detecting a seismic disturbance. It is not very effective in soft grounds, is affected by frost, is a problem where there is static water and is very sensitive to surrounding noise from trains and high speed traffic.
- Seismic-Magnetic intrusion detection device, employs in addition to the seismic detection system, the ability to pick up ferro-magnetic material. This is an improvement over the former system, but still suffers from similar disadvantages.
- Magnetic intrusion detection device, where an alarm condition is created when ferro-magnetic material (for example a weapon carried by an intruder) is moved near to the wire loop buried coplanar with the earth's surface, thus causing disturbance in the local magnetic field. This system is a more positive system from both described above, but still it is not as selective in detection.

- Ported Coaxial Cable intrusion detection device. In this system two parallel cables are buried, with a transmitter and a receiver at the edges. Electromagnetic energy from the transmitter cable couples into the receiver cable producing a static field. An alarm condition occurs when the coupling is distorted. Again it needs a "sterile" area where normal pedestrians do not pass by. Running water and vibrations may also prove to be a nuisance causing alarms.
- Video Motion intrusion detection device, which needs a CCTV system as a prerequisite to function. An alarm condition is caused by any sudden change in the brightness composition of the image on the monitor video screen. Even though it is a very effective security tool, it has many prejudices which produce false alarms, such as sudden change of light (a passing cloud shadowing the scenery), and movement of objects in view (tree branches shaking due to blowing wind).
- Laser intrusion detection device, which is relatively new, but offers long range and absolute accuracy with little maintenance, but is expensive and needs a clear line of site.

Table 4.1 tabulates the level of susceptibility of each of the eleven systems to hazards from environmental and other conditions.

<u>HAZARD</u>	<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>E</u>	<u>F</u>	<u>G</u>	<u>H</u>	<u>I</u>	<u>J</u>	<u>K</u>
Wind to 50 Km/hr	1	1	2	2	1	2	2	1	0	4	1
Wind 50-120 Km/hr	2	2	3	4	1	3	3	2	0	5	1
Wind 120+ Km/hr	2.5	2.5	4	5	2	4	4	2	1	6	1
Rain	2	2	3.5	3	1	2	2	2	3	4.5	1
Snow	3	3.5	3	3	0	2	2	2	2	6	3
Fog	2	3	0	0	0	0	0	0	0	0	3
Noise	0	0	0	0	0	5	5	2	0	0	0
Small Animals	3	3	3	1.5	0	2	2	1	1	1	1
Large Animals	6	6	6	6	3.5	6	6	3	3	6	3
Large Birds	3	3	3	2	1	1	1	1	1	3	3
Lightning	2.5	2	3	1	1	2	4	4	4	5	0
O/H Lines	2	1	1	1	1	2	3	4	1	0	0
U/G Lines	1	1	1	1	1	3	4	4	1	0	0

DETECTOR LEGEND :

A: Microwave

B: Infrared

C: Electric Field

D: Fence Motion

E: Taut Wire

F: Seismic

G: Seismic-Magnetic

H: Magnetic

I: Ported Coaxial Cable

J: Video Motion

K: Laser

0=minimum, 6=maximum

Table 4.1. Outdoor Detector Systems Ratings.

4.5 Indoor Intrusion Advisor

Indoor intrusion systems are employed as a last resort to detect intrusion. The virtue that their geographic location is too close to the protected object or person, imposes that they should have a high rate of detection and low false alarms rate. Fortunately the fact that they are placed in a controllable indoor environment raises their detection ability, and if properly placed and used the nuisance alarms are kept to a very low rate.

This expert system considers the eight most used modern indoor detection devices. These are as follows.

- The microwave intrusion detection device, where low cost, low power reliable hardware in the last decade has produced a proliferation of devices for numerous applications, one of which is security. The microwave signal sent by a detector is reflected by walls. The bounced back signal by the moving objects in the field of view is interpreted by the processor to detect movement.
- The infrared intrusion detection device, which is now the most widely used intrusion detection device in commercial security. The detector houses a transducer which converts the optical energy in the infrared band into electrical signals. The transducers employed are either photo-diodes (photoelectric semi-conductors sensitive in the infrared region) or thermopiles (arrays of miniature thermocouples).
- The ultrasonic intrusion detection device. This device emits ultrasonic frequencies outside the range of human ear sensitivity. The unit then detects the Doppler shift from its own reflected ultrasonic emissions. Any acoustically reflective object in its environment, that moves along any component of an arc, causes a shift in the reflected signal, and hence an alarm condition. Stationary or slow moving objects are filtered out as non-threatening.
- The dual intrusion detection device, which consists of two systems out of the above mentioned three, confined in one detector. This system reduces drastically the nuisance alarms, but on the other hand it is more costly and less sensitive to intrusion since it requires two conditions to be met to set an alarm.

- The video motion intrusion detection device, where a closed circuit television system is a prerequisite. An alarm condition is caused by any sudden change in the brightness composition of the image on the monitor video screen.
- The capacitance intrusion detection device, where a protection loop is connected from the conductive object being protected (conductivity is a pre-condition) to the tuned circuit in the control unit. A change in capacitance between the protection loop and earth disturbs the condition of resonance.
- The vibration intrusion detection device, where either vibration or acoustical energy is converted into electrical signals, or as a mechanical transducer to open or close a switch. This kind is usually fixed to glass, walls or such surfaces.
- The switches intrusion detection devices, which protect doors and windows. Two models are utilized, the micro and the magnetic. The micro-switch is actually a switch itself placed in the lock's cavity. Locking the door with the key exerts a pressure on the device and switches it to the on position. The magnetic-switch is composed of two magnetic pieces, a spring and a switch. One magnet is placed on the swinging part of the door, while the other on the fixed door frame along with the switch and the spring. If the door is opened, the magnets are far away and the spring causes the switch to open, causing an alarm. Closing the door on the other hand, causes a magnetic effect which is a more powerful force than the spring tension, and hence the switch closes.

The questions asked by the system are displayed in Fig.4.3, while a typical outcome (in accordance with the answers) in Fig.4.4.

SESSION SUMMARY	

Cost is important	False
Heat sources are in the vicinity of protected area	False
Sunlight enters room (through window or openings)	True
Moving air is common in the protected area	False
High frequency tones occur in the area	False
Protected area is not confined (next to hallways)	True
Protected area has fluorescent lighting	True
Protected object is conductive	False
There are a lot of vibrations in the vicinity	False
There is a CCTV system	True
There is a video recording system	True
Aesthetics are very important	True
Low maintenance is required	True
Country has high attack risk	True
Target is a high attack risk	True
Application is for a bank	False
Guards are available 24 hours a day 7 days a week	True
During regular hours, area is open to public	True
Low false alarm rate is required	True

Fig.4.3. Indoor Intrusion Question-Answer

RESULT SCREEN	

<u>RESULT</u>	<u>PROBABILITY %</u>
Use microwave detection	0
Use infrared detection	0
Use ultrasonic detection	70
Use video motion detection	93
Use capacitance detection	0
Use vibration detection	79
Use switches (micro & magnetic)	84
Use dual detection system	70
Shut system down during regular hours except CCTV	100

Fig.4.4. Indoor Intrusion Summary

4.6 Camera Selection Advisor

A closed circuit system is composed of three basic components. These are the camera, the cabling and the monitor [33]. This expert system concentrates on the selection of the cameras as the selection of the latter two is a straight forward process.

In order to achieve the proper choice of equipment, the designer is to first determine the purpose of the system, funding available, existing lighting levels or achievable ones and the level of security needed. The layout and variance in the lighting dictate the lens format and lens size, while environmental conditions determine the camera housing type and accessories.

Modern cameras are categorized into two classes, the pick-up tube and the charge-coupled device (CCD) also known as the chip camera. The former has been used extensively in the past while the latter is a newcomer, but persistently gaining solid ground in the market. An enhancement to both versions is the use of an intensified tube or chip, which allows for an excellent performance under much lower lighting conditions (starlight conditions, around 10^{-4} lux, where human beings are virtually blind).

Cheap tube cameras use the vidicon pick-up technology. This type is known to have a high propensity to burn or retain images, and is best suited for low security level indoor locations, where constant bright light conditions prevail. For heavy duty, higher resolution, at locales with drastically varying light conditions, the newvicon or ultracon tube technologies are used.

The chip cameras come with different qualities that can replace the applications of the tube cameras. They require less energy, space, maintenance and have longer life. Still to date they do not offer the higher resolutions of their counterparts, and require an infrared filter to enhance day pictures, and hence hampers the camera's sensitivity.

The larger the camera's lens format the better the resolution is. There are three standard lens formats, the one half, the two thirds and the one inch. For high level security monitoring jobs, where a video recording may be used as an evidence of identifying the intruder, the larger formats are recommended.

The distance between the camera and the monitored object and the angle of view determine if a standard, telephoto or wide angle is required. The further the object is, or if a small view angle is needed, then the larger size of lens is required. For higher security levels, where indoctrinated guards are on site, the use of a zoom lens with a pan and tilt head is recommended.

If the light is constant, as is the case in some indoor applications, then a manual or auto-iris is acceptable. However, for continuously varying light, an auto-iris is a must to control the amount of light entering the camera and hence obtaining a clear image.

The environment conditions play an important role in determining the accessories. The temperature forces the addition of a heater in cold locations and a fan in hot ones. To avoid dirt or excessive dust accumulating on the lens, a washer and wiper may be used or regular cleaning must be performed.

A sample question and answer session is shown in Fig.4.5 and a sample output in Fig.4.6.

SESSION SUMMARY	

Light is bright & constant	False
Cost is important	False
Low maintenance is desired (including change of tube)	True
Life expectancy is important	False
Space occupied by camera is important	False
Cast starlight is only available at some times	True
Clear starlight is only available at some times	True
Full moonlight is only available at some times	True
Twilight or better lighting is available at all times	False
Camera will be used to identify persons	True
Vandalism is a problem	True
Environment is hazardous	False
Temperature drops below 0 deg C (32 deg F)	False
Temperature goes above 27 deg C (80 deg F)	True
Locale is dirty	False
Light level changes with time	True
High security level is required	True
Camera is close to monitored object	False
Camera is far away from monitored object	True

Fig.4.5. Camera Selection Question-Answer

RESULT SCREEN	

<u>RESULT</u>	<u>PROBABILITY %</u>
Use vidicon tube	0
Use CCD chip	0
Use newvicon or ultracon tube	0
Use intensified tube (SIT)	63
Use intensified chip	73
Use ultra-intensified tube (ISIT)	65
2/3" format lens	20
1" format lens	90
Use a high resolution camera	100
Put camera in a vandal proof enclosure	100
Use fan in camera enclosure	100
Use an auto-iris lens	100
Use a zoom lens	95
Use a pan & tilt head	70
Use a wide angle lens	10
Use a telephoto lens	85
Use a standard lens	65

Fig.4.6. Camera Selection Summary

4.7 Access Control Advisor

Access control describes any technique used to control passage into or out of an area [34]. A properly designed access control system reduces the risks of losses and security, protects assets, and provides a secure environment for a workplace.

Over the years, access control systems have become more and more sophisticated. Today they incorporate the latest advances in microprocessor technology, and are based on cards, the size of a standard credit card [35]. This expert system selects the most appropriate card and reader, for the specifics of the application, from among current technology. These are the following:

- Magnetic Stripe Card. Its major advantage is low cost, field encoding and the large quantity of data stored on it. Disadvantages include its relative susceptibility to counterfeiting, vulnerability to card slot vandalism, problems with magnetic stripe abrasion and relatively high reader maintenance requirements.
- Barium Ferrite Card, which is another type of magnetic card offering better security and resistance to duplication and cheaper cost than the former card, but this older technology has relatively shorter card life, and both card and reader are subject to wear.
- Weigand Card, which is named after its inventor, uses a series of small diameter wires, that when subjected to a changing magnetic field, induces a voltage output in the sensing coil. This card is factory encoded and is difficult to copy, counterfeit or alter. It has the ability to store a large number of code and is moderately priced. However, because the card has to be passed through the slot, card wear and life can be a problem.
- Proximity Card, offers a hands-free operation and hence extends the life of both the card and reader. They are also encased in vandal-proof, dirtproof, weather-proof and shock resistant housing. Cards are factory encoded and are difficult to copy or counterfeit. Disadvantages are card thickness and high cost.
- Keys, the traditional access control are not to be overruled for low security level jobs.

A sample question and answer session is shown in Fig.4.7 and a sample output in Fig.4.8.

SESSION SUMMARY	

Cost is important	False
Quantity of data to be stored on card is high	False
Counterfeiting is to be avoided highly	True
Vandalism is a problem	True
Wear is to be avoided	False
Low maintenance is required	True
Card orientation is a problem	False
Aesthetics are very important	True
Life expectancy of system is important	False
In-house encoding is a must	False
Weather resistance is a condition	True
Magnetic field resistance is desired	False
Application is for a parking	False
Card size is crucial (less than 3mm thick)	True
High security level is required	True

Fig.4.7. Access Control Question-Answer

RESULT SCREEN	

<u>RESULT</u>	<u>PROBABILITY %</u>
Magnetic Stripe	32
Barium Ferrite	0
Keys	0
Weigand	67
Proximity	0

Fig.4.8. Access Control Summary

4.8 Discussion

These security expert systems offer an overall view where the security engineer may experiment on this model, and modify criteria acquiring sound advice each time,

until achieving the most safe and reliable system. With such an advice the engineer can rest assured that the selected equipment are right for the application, having low false alarm rate, detecting and preventing intrusion.

Additional knowledge may at any time be incorporated into the system. For example, if new equipment is used in industry in the future, new rules having the characteristics of these devices may be added to the system.

Other security system advisors may as well be prepared and added to this expert system. These may include advisors for physical barrier selection, control room equipment and camera location.

5 TWO APPLICATIONS

5.1 Introduction

In addition to the security systems advisors, two more expert systems were developed to demonstrate the uses of the SES. The first application is a numerical integration advisor. Depending on the available resources, degree of complexity and accuracy required, this expert system selects the most appropriate numerical integration method from among seven.

The second expert system is a demonstration of the capability of linking to other programs, whether developed in the spreadsheet environment or completely strange to it. It is an advisor of the approximate relative cost of a selected generating set with its auxiliaries, compared to a basic generating unit. The next step is the actual sizing of the generating set in accordance with the load requirements, which is carried out using either an external software or a program written in Lotus 1-2-3.

5.2 Numerical Integration Expert System

5.2.1 General

Elementary calculus focuses on two important processes of mathematics: differentiation and integration [36]. Differentiation is rather a straight forward procedure. Integration from a standpoint of numerical mathematics is examined in this expert system. An advisor is introduced for the selection of the most appropriate

integration method. The declarative method was adopted to develop this system.

5.2.2 System Overview

It is customary to distinguish between two types of integrals, the definite and the indefinite. The indefinite integral of a function is another *function* or class of functions, where as the definite integral of a function over a fixed interval is a *number*. An example of an indefinite integral is

$$\int x^2 dx = x^3/3 + C$$

where C is a constant. An example of a definite integral on the other hand is

$$\int_0^2 x^2 dx = 8/3$$

The definite integral of a function is interpreted as the area under the curve, and may always be computed regardless whether an anti-derivative can be found or not [37]. This advisor helps in the selection of the most appropriate mathematical integration method.

5.2.3 Numerical Integration Advisor

In the same manner as the security advisors, to operate the system, Lotus 1-2-3 must be in the ready mode. SES is then loaded into memory by retrieving the file

called *SES*. This is achieved by pressing the slash (/), File then **Retrieve** (Lotus commands) from the keyboard, then pointing to and selecting the file *SES* from the menu that appears.

The program's logo is then displayed, after which the **Declarative** method and **Get-File** options are to be selected from the menus that appear. The next step is to choose the file *Integral* from among the list of files exhibited on the screen.

At this stage, the *SES* is at the utility menu level. Pressing **Run**, then **Advisor** starts executing the expert system. A few questions are asked by the expert system as displayed in Fig.5.1, the session summary.

SESSION SUMMARY	

Are the data points few (less than 10) or many?	Few
Is spacing of points even, equal irregular or may be chosen by user's discretion?	Even
Which is more important, accuracy or simplicity?	Simplicity
Is a computer available or not?	No

Fig.5.1. Numerical Integration Session Summary

SES searches its database, Fig. 5.2, to find a match. In accordance with the answers of Fig. 5.1, the first three examples meet the criteria. *SES* thus recommends in the output screen, Fig. 5.3, that these integration methods be used.

EXAMPLE SCREEN						

No	Data-Points	Spacing	Accuracy	Computer Available?	Method	Weight
1	few	even	simplicity	no	Midpoint	1
2	few	even	simplicity	no	Simpson	1
3	few	even	simplicity	no	Trapezoidal	1
4	few	even	simplicity	yes	Midpoint	1
5	few	even	simplicity	yes	Simpson	1
6	few	even	simplicity	yes	Trapezoidal	1
7	few	even	accuracy	no	Midpoint	1
8	few	even	accuracy	no	Simpson	1
9	few	even	accuracy	no	Trapezoidal	1
10	few	equal	accuracy	yes	Romberg	1
11	few	equal	accuracy	yes	Newton	1
~~~~~						
61	many	irregular	accuracy	yes	Romberg	1
62	few	even	accuracy	yes	Romberg	1
63	few	even	accuracy	yes	Newton	1

Fig.5.2. Numerical Integration Partial Examples

THE RULES TO BE SEARCHED			
*****			
<u>Data-Points</u>	<u>Spacing</u>	<u>Accuracy</u>	<u>Computer</u>
few	even	simplicity	no
RESULT OF SEARCH...			
*****			
Recommended Methods			
*****			
	<u>Method</u>	<u>Confidence</u>	
	Midpoint	1	
	Simpson	1	
	Trapezoidal	1	
<u>Comments :</u>	These are simple to use integration methods		

Fig.5.3. Numerical Integration Output Screen

At this stage, the user may plot the different confidence levels of each integration method, as a sector of a pie chart, compared to the recommended remaining methods. In this case, each sector of the pie chart occupies one third of the circle, as all three outcomes have equal confidence levels.

Having established the recommended integration methods, the user may experiment on the system. Running the advisor again, only this time changing the replies to the questions, is a means to find out the recommended integration methods having different resources, such as a computer for example. The user may as well ask for a tree display of the examples, Fig. 5.4, to have an overall view of the rules. This is achieved by pressing **Run**, **Rule** and **Exhaustive** options.

<b>EXHAUSTIVE SEARCH SCHEMATIC</b>				
*****				
<u>Data-Points</u>	<u>Spacing</u>	<u>Accuracy</u>	<u>Computer</u>	<u>Method</u>
few	even	accuracy	yes	-----Romberg
				& - - -Newton
			no	-----Midpoint.
		simplicity		& - - -Simpson
				& - - -Trapezoidal
			yes	-----Midpoint.
	equal	accuracy		& - - -Simpson
				& - - -Trapezoidal
			no	-----Midpoint.
		accuracy		& - - -Simpson
				& - - -Trapezoidal
			yes	-----Romberg
			no	-----Romberg
~~~~~				

Fig.5.4. Numerical Integration Partial Tree

In addition the user may ask for more information, such as statistics on the database, Fig. 5.5. In subsections 3.5 and 3.6 the utility and operation functions, detail of the different possibilities to acquire information and graphs available in the SES were described.

STATISTICS SCREEN				

No. of Examples: 63 Total Weight: 61				
<u>Result</u>	<u>Occurrence</u>	<u>Avg. Wt</u>	<u>Freq.</u>	<u>Occurred In Examples.</u>
Gauss	12	1.0	19.0%	14,15,16,17,46,47,48,49,50,51,52
Interpole.	4	1.0	6.3%	18,19,20,21
Midpoint.	11	1.0	17.5%	1,4,7,38,39,40,41,42,43,44,45
Newton	2	1.0	3.2%	11,13
Romberg	10	1.0	15.9%	10,12,54,55,56,57,58,59,60,61
Simpson	11	1.0	17.5%	2,5,8,21,23,24,25,26,27,28,29
Trapezoidal	11	1.0	17.5%	3,6,9,30,31,32,33,34,35,36,37

Fig.5.5. Numerical Integration Statistics Screen.

5.2.4 Discussion

This expert system advises on the selection of a numerical integration method from among seven different methods. A future addition may be to actually carry out the calculation of the integration advised, according to the users' problem. Seven different separate programs are to be written, using Lotus 1-2-3 macros or any other program, and a forward chaining process links to these programs to provide the automatic execution of the actual integration. The first letter in the name of these programs need to start with an exclamation mark (!), to indicate to SES that a forward chaining is required.

5.3 Generating Set Cost and Sizing Advisor

5.3.1 General

A proliferation of companies are now offering standby generating sets to a bewildering degree of different specifications. The engineer is to carefully study the literature of these companies and to decide on the most appropriate set for his peculiar application [38]. In addition, he is to determine what auxiliaries does the locale require, such as a totally enclosed housing for outdoors installation, sound-proofing where codes require a damped noise level, and the mode of starting the set.

5.3.2 System Overview

The cost of auxiliaries can add up considerably to the overall cost of a generating set. For example, having an outdoor totally enclosed set costs around fifteen percent more. If soundproofing is necessary another thirty percent is required. The mode of generator starting also has some affects on the cost.

This expert system is a tool to aid the engineer in advising the relative cost of a specific chosen set compared to a basic standard unit. The expert system is then automatically chained to one of two actual generating set sizing programs, to demonstrate the various capabilities of this shell. The first is developed by Microsoft Corporation for Onan Corporation, while the second is developed utilizing a Lotus 1-2-3 spreadsheet format.

5.3.3 Generating Set Advisor

To operate the system, Lotus 1-2-3 must be in the ready mode. SES is then loaded into memory by retrieving the file called *SES*. This is achieved by pressing the slash (/), File then **Retrieve** (Lotus commands) from the keyboard, then pointing to and selecting the file *SES* from the menu that appears.

The program's logo is then displayed, after which the **Declarative** method and **Get-File** options are to be selected from the menus that appear. The next step is to choose the file *GenSet* from among the list of files exhibited on the screen.

At this stage, the SES is at the utility menu level. Pressing **Run**, then **Advisor** starts executing the expert system. A few questions are asked by the expert system as displayed in Fig. 5.6.

SESSION SUMMARY	

What is the percentage of load to be served?	Partial
What is the method of starting up desired?	Automatic
Is there a space for the set to be installed indoors?	Outdoor
Is soundproofing required?	Yes
Back-chain to find proper rating	Yes

Fig.5.6. Generator Advisor Session Summary

Having established these basic facts, the program searches its database to determine a match to the user's answers, and displays the output results, Fig.5.7. In this example, the cost is 1.9 times more than that for a basic standard generating set.

THE RULES TO BE SEARCHED				

<u>LOAD</u>	<u>START-UP</u>	<u>LOCATION</u>	<u>SOUND</u>	<u>RATING</u>
Partial	Automatic	Outdoor	Yes	Chain to file
RESULT OF SEARCH...				

The relative cost is displayed. Press RETURN to Chain to the actual sizing program				

	<u>COST</u>	<u>CONFIDENCE</u>		
	c=1.9	1		
<u>Comments</u> :relative cost to indoor/hand-start/no sound proof generator				

Fig.5.7. Generator Advisor Output Screen.

At this stage the SES asks if a chaining to an external program or to a spreadsheet based program is required for the actual generator sizing process. This is done through a menu to select one of the two options, or to discard the chaining completely.

Selecting the Lotus 1-2-3 sizing file, loads this file automatically into memory. This file is built using the 1-2-3 macro commands, and has an auto-execute feature. Hence it starts acquiring information about the specifics of the project, Fig.5.8. Project reference and number and the current date are first asked. The system's operating voltage (column 3), and the accumulated non-motor loads, are then entered. After that, the motors' horsepower (column 1), motor's nameplate code (column 2) and means of starting (column 4) are entered for each of the motors, in the sequence they are deemed to start.

According to the motor's nameplate code entered, the program extracts from its database the specifics of the motor characteristics. These characteristics, along with the motor's horsepower rating and the reduced voltage starting method determine, in both kVA and kW units, the motor's starting load (locked rotor load, columns 5 and 6), and the running motor load (columns 7 and 8). The accumulated load plus the motor starting load (columns 9 and 10), and the total loads (columns 11 and 12) are then computed.

Finally the program extracts from all these steps the maximum starting kVA and kW loads, as well as the total accumulated loads. With these four values, the user is then to search the manufacturers specifications, to select a generating set that has the capacity to drive these motors.

A link to an external program, may be achieved with the selection of the second chaining option. For demonstration purposes a program prepared by Microsoft Corporation for a leading generating sets manufacturer, Onan Corporation, is provided. A copy of this program and a license to use it are required.

Generator Selection For Electric Motor Loads									
Project Reference: Trial					Date: 11/5/90				
Project Number : 90-1									
Motor Information From Nameplate				Reduced Voltage Motor Starting		Motor Starting Load (Locked Rotor)			
---	1	2	3	4	---	5	---	---	---
---	HP	Code	V	Tap	Mult	LR kVA	PF	---	LR kW
---	---	---	---	---	---	---	---	---	---
1	150	F	415	Auto	0.64	550.	0.28		154.
2	50	F	415	Y/D	0.33	94.5	0.36		34.0
Motor Running Load		Total Load+Next Motor Start		Total Load		Accumulated Load+Next Motor Start		Accumulated Load	
---	---	---	---	---	---	---	---	---	---
7	8	9	10	11	12	13	14	15	16
kVA	kW	Max kVA	Max kW	Cont kVA	Cont kW	Max	Max	Cont	Cont
---	---	---	---	---	---	---	---	---	---
140	123	595	194	45	40	<u>kVA</u>	<u>kW</u>	<u>kVA</u>	<u>kW</u>
49.4	41	279	197	185	163				
				234	204				
Select A Generator Set Having These Specifications:						595	197	234	204

Fig.5.8. Generator Sizing Program Output Screen.

A DOS batch file permits the link to such a program. As is the case in most programs, generalities about the project are first entered. The different loads are then entered. The nature of these loads are specified, such as motor, fluorescent lighting, resistive or computer loads. After entering all the data, computations are carried out in a similar fashion to that described in the 1-2-3 developed program. Finally the recommended generating set reference is displayed, along with its characteristics, from Messrs. Onan database and the actual steady and surge loading, Fig.5.9.

This is the Recommended Generator Set Based on Information supplied:		Total Loading:
ONAN	250DFM	Max Surge kW= 200
Max Surge:	260 kW	Max Surge @ Step 3
Max Surge:	864 kVA	-----
Max Steady:	250 kW	Max Surge kVA= 534
Max Steady:	312 kVA	Max Surge @ Step 2
Generator Voltage:	480 V/60 Hz	-----
Nominal Amps:	378 A	Max Steady kW= 210
		Max Steady kVA= 230

Fig.5.9. Onan Generator Sizing Program Output Screen.

5.3.4 Discussion

This expert system computes the approximate ratio of increased cost due to the additional auxiliaries needed, over a standard basic generating set unit. It then chains to programs, to size the generating set according to the actual load. Programs prepared by different companies may be chained to this expert system, by simply modifying the DOS batch file. Developing more expert systems to advise or help on the selection of other generating set aspects, may be added, and chained to this one.

6 CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH

6.1 Conclusions

In this thesis, a new spreadsheet based expert shell called SES was developed and introduced. This shell was written using Lotus 1-2-3 spreadsheet Macros.

SES is menu driven, has an on-line help facility and is user friendly.

SES is a natural add-on to existing 1-2-3 programs. It is efficient and not expensive.

SES may be used wherever Lotus 1-2-3 can be installed, and hence benefits from all Lotus 1-2-3 capabilities. Since newer releases can read the code of their predecessors, therefore SES can be upgraded to future 1-2-3 releases without major modification.

SES provides one single environment where data, algorithms, knowledge and an inference engine coexist.

Possibility of developing expert systems in any Latin based language was shown. The rules are entered into SES as natural language sentences. They are then automatically parsed into pieces of information by the software.

The two distinct methods for developing expert systems, the procedural rule and the declarative example based methods, were demonstrated. Expert systems may be developed using any one or a combination of the two methods mentioned above. In addition SES has a generous number of debugging, validating and reporting facilities.

Three practical expert systems were constructed utilizing SES. The first is a security system advisor, which consists of four separate advisors. It aids the design engineer in the process of selection of the required equipment. The security systems that are the optimum, most effective, having the lowest false alarm rate and the most suitable for the location and environment, are recommended.

The second advisor selects the most appropriate numerical integration method, for a particular application.

The last advisor recommends the auxiliaries needed for a generating set, provides the corresponding cost, and sizes the generator according to the load.

6.2 Suggestions for Future Research

When used on Lotus 1-2-3 release 3, the procedural method of search in SES is slow in execution. This is attributed to the fact that this new release of Lotus 1-2-3 is faster in few operations, but much slower in the database extract commands that SES depends on. Re-coding some of the procedures, to reduce their dependency on the database extract commands, may prove to be one solution to overcome this limitation.

Spelling mistakes while developing a procedural rule based expert system, cause a string mismatch. A consequence is that backward chaining does not occur in this case. A possible remedy is to include another option for entering the rules that uses phrases, which can be recalled at any instance to input the rules. The phrases guarantee that no string mismatch may occur, as the same phrase is used over and over.

Allowing more logical operators in the phrases, is to be made possible. Currently this is limited to two operators in each phrase, as a compromise for acceptable search speed. The number of logical operators may be increased upon the release of faster spreadsheets.

For computing the confidence of a certain result in the procedural method, only the average of all particular probabilities is available. This is purposely done to ensure a safe means of obtaining feasible confidence in the results, whether the rules are dependent or independent. Other confidence methods may be added to the system, on condition that proper warning screens alert the user to the consequences of improper use.

Provision of an optimum search method, to reduce the number of questions asked by the system while running a knowledge base, may be added. However, properly warning the expert system developer of the consequences of using this method when having an incomplete knowledge base is also to be provided.

For the security systems advisors, additional or more recent knowledge may be incorporated into the system. For example, if new equipment is used in industry in the future, new rules having the characteristics of these devices may be added to the system.

Other security system advisors may as well be prepared and added to this expert system. These may include advisors for physical barrier selection, control room equipment and camera location.

A future addition to the numerical integration advisor, is to provide forward chaining links to programs that carry out the numerical integrations. The first letter in the name of these programs needs to start with an exclamation mark (!), to indicate to SES that a forward chaining is required.

As for the generating set advisor, programs prepared by different companies may be chained to this expert system, by simply modifying the DOS batch file. Developing more expert systems to advise or help on the selection of other generating set aspects, may be added, and chained to this one as well.

REFERENCES

- [1] D. Nebendahl, *Expert Systems, Introduction to the Technology and Applications*, Siemens Aktiengesellschaft and John Wiley and Sons Limited, Berlin, 1988.
- [2] C. Stinson, " Spreadsheet Heavyweights take on 1-2-3", *PC Magazine*, Volume 9, Number 8, April 24,1990, pp.97-156.
- [3] L. Widman, K. Loparo, N. Nielsen, *Artificial Intelligence, Simulation and Modeling*, John Wiley & Sons, N.Y., 1989.
- [4] R. Levine, D. Drang, B. Edelson, *Comprehensive Guide to Artificial Intelligence and Expert Systems*, McGraw-Hill Book Company, N.Y., 1986.
- [5] Canada's National Research Council, "*The Social Context of Artificial Intelligence*", Ottawa, 1989.
- [6] S. Shapiro, D. Eckroth, G. Vallasi, *Encyclopedia of Artificial Intelligence*, John Wiley & Sons, N.Y., 1987.
- [7] A. Hart, *Knowledge Acquisition for Expert Systems*, McGraw-Hill Company, N.Y., 1986.
- [8] "*Advantages of Expert Systems*", Logicware Inc., Toronto, 1988.
- [9] R. Miller, T. Walker, *Artificial Intelligence Applications in Engineering*, SEAI Technical Publications and Fairmount Press Inc., Georgia, 1988.
- [10] "*Pros and Cons of Expert Systems*", Advanced Decision System, San Antonio, CA, 1988.
- [11] A. Gupta, B. Passad, *Principles of Expert Systems*, Cambridge MA, M.I.T. Press, 1988.

- [12] A. Gupta, B. Passad, *Microcomputer Based Expert Systems*, IEEE Publications, N.Y., 1988.
- [13] Y. Shirai, J. Tsuji, *Artificial Intelligence Concepts, Techniques and Applications*, John Wiley & Sons Ltd., N.Y., 1984.
- [14] W. Clocksin, C. Mellish, *Programming in PROLOG*, Springer-Verlag, Berlin Heidelberg, New York, 1981.
- [15] A. Walker, M. McCord, J. Sowa, W. Wilson, *A Logical Approach to Expert Systems and Natural Language Processing, Knowledge Systems and PROLOG*, Addison-Wesley Publishing Company Inc., Reading MA, 1987.
- [16] W. Thomas, *1st Class Expert Systems Reference Manual*, 1st Class Expert Systems, Inc., Wayland MA, 1989.
- [17] D. Strickland, "Product Review, 1st Class Expert System", *IEEE Expert*, Summer 1988, pp. 69-71.
- [18] *Exsys Reference Manual*, Exsys Inc., Albuquerque NM, 1989.
- [19] R. Scoville, "Powerful Spreadsheets: A New Standard?", *PC World*, October 1989, pp. 96-110.
- [20] R. Scoville, "Seven Sensible Spreadsheets", *PC World*, April 1990, pp. 116-130.
- [21] *Lotus 1-2-3 Reference Manual*, Lotus Development Corporation, Cambridge MA, 1989.
- [22] M. O'Brien, "Lotus Moves up from Micros", *Computer Data*, Volume 15, Number 4, April 1990, pp. 1-4.
- [23] M. Cambell, *1-2-3 Release 3, The Complete Reference*, McGraw-Hill, N.Y., 1989.

- [24] A.T. Williams, *Lotus 1-2-3 Release 3*, John Wiley & Sons, N.Y., 1989.
- [25] J. Mueller, *Illustrated Lotus 1-2-3 Release 3*, Wordware Publishing Inc., N.Y., 1990.
- [26] D. Bookbinder, *The Lotus Guide to 1-2-3 Release 3*, Addison Wesley Publishing Inc., N.Y., 1989.
- [27] B. Gnedenko, *The Theory of Probability*, Chelsea Publishing Company, N.Y., 1967.
- [28] K. Weichselberger, S. Pohlmann, *A Methodology for Uncertainty in Knowledge Based Systems*, Springer, Berlin, 1990.
- [29] I. Goodman, H. Nguyen, *Uncertainty Models for Knowledge-Based Systems*, North-Holland, Amsterdam, 1985.
- [30] M. Gupta, *Approximate Reasoning in Expert Systems*, North-Holland, Amsterdam, 1985.
- [31] H. Noble, "A Step by Step Guide to Security", *International Security Review*, U.K., a Series of Articles from 1983 Until 1986.
- [32] H. Noble, "Perimeter Defence, Final Thoughts", *International Security Review*, U.K., February 1988, pp.35-42.
- [33] C. Pierce, "Executive Protection, CCTV", *Security Management*, American Society for Industrial Security, November 1989, pp. 69-78.
- [34] "What Every Executive Should Know About the Ins and Outs of Access Control", Shlage Electronics, Part of Worldwide Ingersoll-Rand, Santa Clara, CA, 1989.
- [35] R. Mourey, "It's in the Card", *Security Management*, American Society for Industrial Security, July 1989, pp. 17-20.

- [36] S. Yakowitz, F. Szidarovszky, *An Introduction to Numerical Computation*, Macmillan Publishing Company, N.Y., 1989.
- [37] W. Cheney, D. Kincaid, *Numerical Mathematics and Computing*, Brooks/Cole Publishing Company, Monterey CA, 1985.
- [38] J. Ridout, "Selection in Standby Plant", *Gulf Construction*, March 1985, pp. 34-41.

APPENDIX A: PROGRAM FLOW CHART

