THE UNIVERSITY OF CALGARY

EXPERT SYSTEMS AND APPLICATIONS

by

Anwar M.F. Soukaria

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

DECEMBER, 1987

ISBN 0-315-42409-5

THE UNIVERSITY OF CALGARY

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommended to the Faculty of Graduate Studies for acceptance, a thesis entitled, "*Expert Systems and Applications*" submitted by Anwar M.F. Soukaria in partial fulfillment of the requirements for the degree of Master of Science.

Supervisor - Prof. M.H. Hamza
Dept. of Electrical Engineering

Interm Supervisor - Dr. R.B. Streets, Jr.
Dept. of Electrical Engineering

Dr. M. Fattouche
Dept. of Electrical Engineering

Dr. G.S. Hope
Dept. of Electrical Engineering

Dr. J. Graham
Faculty of Management

Date: Dec. 21, 1987

Abstract

The application of expert systems in the fields of electrical engineering and marketing management is investigated. The design and build-up of an expert system process using an expert system shell, and the knowledge acquisition and representation, are analysed.

Three expert systems for laboratory instruction in control systems were developed as computer-aided instruction tools. The systems assist a student to reduce the error in performing three experiments in control.

An expert system was developed as an application of these systems in marketing management. This system assists an entrepreneur in developing a marketing plan for an imitative durable product used in households.

Design and build-up procedures for expert systems with a potential for generalization were developed. These procedures offer a compromise solution between the existing procedures that range from complex algorithms that involve programming structural analysis to simple procedures that visualize the design process through a general build-up procedure.

The expert systems developed were tested, analysed and evaluated. The evaluation gave an insight to possible applications and hinted at constructive considerations and conclusions about expert systems application in the field of electrical engineering and marketing management.

Acknowledgements

The author wishes to express his sincere gratitude to Dr. M. H. Hamza for his guidance and support throughout the course of this work, and for providing me with the opportunity to research in the field of expert systems.

The author is indebted to Dr. R. B. Streets, Jr. for his appreciated patience and for his valuable advice during the preparation of this manuscript.

The invaluable assistance and contribution of Mr. B. Yeung, Mr. M. I. Itani and Ms. Bettina Wyss are deeply appreciated.

Thankful appreciations are extended to Dr. M. T. Fattouche and electrical engineering graduate students for their constructive suggestions and support.

The author is especially grateful to Dr. J. W. Haslett for the financial support he arranged through the Department of Electrical Engineering in the form of graduate teaching and research assistantships.

Finally, I would like to thank Laura Millan for typing this thesis.

To My Parents

and My Brothers

Table of Contents

Page No.

Table of Contents (continued)

List of Tables

List of Figures

LIST OF SYMBOLS

A	Total attribution level
BC	Backward chaining
CA	Circuit adjustment
Cal	Transfer function parameter adjustment
D	Display of system response
E(s)	$L[e(t)]$
Ex	explanations
EN	End node or result
Ep	External program
F	External file
FC	Forward chaining
G(s)	Control plant transfer function
H	Feedback factor
I	Introduction or general instructions
IN	Inactive node
K	Pre-amplifier output gain
Ke	Potentiometer error factor
Km	motor gain constant
Kv	velocity error constant

M(s)	$L\ [m(t)]$
N	Node
N_1	Output potentiometer-motor coupling gear ratio
Q	Question
Q_1	Annual potential market roles in dollars
R(s)	$L\ [r(t)]$
y(s)	$L\ [y(t)]$
T	Total annual market share of potential competitors in dollars
Th	Theoretical review
V	Value
a	Actual annual market share in dollars
a_i	Importance weight of attribution i.
a_1	Estimated number of accessable customers in %.
a_2	Esimated number of customers aware of the product in %.
a_3	Estimated number of customers intending to by the product in %.
a_o, b_o, b_1, b_2	Second order system transfer function parameters
b	Actual number of buyers
b_i	Atribution level of a product
$e(t)$	Error signal
$g(t)$	$L^{-1}\ [G(s)]$

m	Available market share in dollars
$m(t)$	Controller output signal
n	Estimated total number of buyers
p	Average price per unit in dollars
q	Product quantity purchased annually by an average buyer
$r(t)$	Reference input
u	Estimated number of units to be manufactured per year
v_a	dc motor armature voltage
$y(t)$	Contol system output signal or response
$\Theta_i(t)$	Angular displacement reference input
$\Theta_o(t)$	Angular displacement output of the dc motor shaft
τ_m	Motor time constant
$\omega(t)$	Angular velocity
ω_n	Undamped natural frequency
ζ	Damping ratio
#	Inference indicator

CHAPTER 1

EXPERT SYSTEMS

1.0. Introduction

Continuing developments in the field of artificial intelligence (AI) have contributed considerably to the extensive attention that expert systems have received. With the magnitude of accomplishments that the application of expert systems promise, many companies and businesses are implementing expert systems and are reaping enormous harvests in terms of profitability, competitive edge and cost reduction [1]. A record of successful applications of expert systems exists, and is enough to justify research and continued interest in determining feasible application domains.

The purpose of any application of expert systems is to emulate human knowledge and intelligence in a certain expertise domain, and to use these systems to obtain expert advice. The present attempts to achieve this purpose are at best primitive, and no evidence is available to prove that it is possible to emulate real human intelligence on today's computers [1]. Evidently, there are narrow application areas with constrained knowledge requirements and restricted usage environment in which the goal of emulating human knowledge and intelligence is achieved.

The objective of this thesis is to investigate expert systems and examine the possibility of applying them in the fields of electrical engineering and marketing management. The background and interests of the author and the thesis supervisor determined these choices. The applications selected in electrical engineering were such as to yield a result of practical value. The application in marketing management required a different approach in decision making and is in general a much larger problem. Three expert systems were developed for laboratory instruction in control. One expert system was developed that enables an entrepreneur to develop a marketing plan.

1.1. What is an Expert System?

Due to the different perceptions of the nature of expert systems and what these systems can achieve, many definitions of expert systems exist [2]. Some of these definitions list the features of expert systems (expertise knowledge domain, inference mechanisms, etc.), while others try to point out particular programming features (e.g. use of logic programming techniques) and outline the limits of what an expert system is supposed to accomplish. Expert systems are by-products of the AI field. Ideally, the goal of AI is to produce a machine that can emulate intelligent human activities [1]. Hence, expert systems should express the same ideas that AI tackles.

The British Computer Society's Committee of the Specialist Group on Expert Systems adopted the following definition of an expert system: "An expert system is the embodiment within a computer of a knowledge-based component from an expert skill in such a form that the machine can offer intelligent advice or take an intelligent decision about a processing function. A desirable additional characteristic, which many would regard as fundamental, is the capability of the system on demand to justify its own line of reasoning in a manner directly intelligible to the enquirer. The style adapted to attain these characteristics is rule-based programming." Such a definition focuses on programming, allows a wide range of applications, and hints at an "explanation of reasoning" capability.

The IEEE Standard Dictionary of Electrical and Electronics Terms [3] does not include a definition of expert systems, which may serve as an indicator of the relative recency of the field. A simple definition proposed by Johnston [4], states that an expert system is a computer system which emulates human expertise by making deductions from given information using the rules of logical reference.

Whatever the definition is, the characteristic of knowledge-based programming and the capability to emulate human experts in a certain domain are shared by all the definitions. As mentioned earlier, the field of expert systems is relatively new, and this may explain the variety of definitions available. It should be noted that no discrepancies exist to undermine a general understanding of what an expert system is and what is to be expected from its application. Hence, in this thesis an expert system is understood to be a knowledge-based program designed to provide an expert advice in a certain domain to an end user. An expert system has the following characteristics [5].

1. A knowledge base drawn from experts in a certain domain. This domain should itself be rational and can be represented using reason.
2. Usage of specific knowledge representation techniques that can emulate human reasoning. These techniques vary according to their suitability for the expertise domain tackled in the expert system.
3. The capability of explaining the reasoning that led to an advice upon request. This explanation may not be required depending on the nature of the advice and the expertise domain.

In addition, an expert system should handle complexity of reasoning through logical rules that are the base for inferring any advice to the end user. It also should be easy to use and modify.

1.2. Basic Architecture of Expert Systems

In general, expert systems have the same basic architecture with certain minor alterations depending on the expertise domain to be emulated. A comprehensive expert system would include the following components [1].

1. A knowledge-acquisition module designed to "extract" the required knowledge from the experts.
2. The knowledge base embodying in symbolic form the knowledge of an expert, or experts, in a certain domain.
3. An inference engine that establishes the interactive mode between the user and the knowledge base.

4. The user or other external sources of information like data files, external programs, sensors, etc.

There are various ways of representing the basic architecture of expert systems. Each way corresponds to a certain stage in the expert system build-up, i.e. the stages in such a build-up determine what components are added and what are absent. Most of these representations may as well converge to the model in Fig. 1.1.

If the stage of knowledge acquisition is included, the representation would be modified to the model in Fig 1.2 [6].

The definition of an expert system that a researcher adopts plays an important role in the composition of an expert system. This variety of definitions have an impact on the degree of complexity of the models but does not introduce any discrepancies among these models, and accordingly, the model in Fig. 1.1 will be considered as the major reference for representing an expert system. This model parallels the adopted definition of an expert system.

1.2.1. The Knowledge Acquisition Module

Knowledge must be obtained from an expert in the expertise domain. A module that can extract this knowledge and prepares it to be processed to build the knowledge base is called a knowledge acquisition module. This module can serve as a component used to update information for self-learning expert systems and as a separate block for other expert systems with a similar field of application.

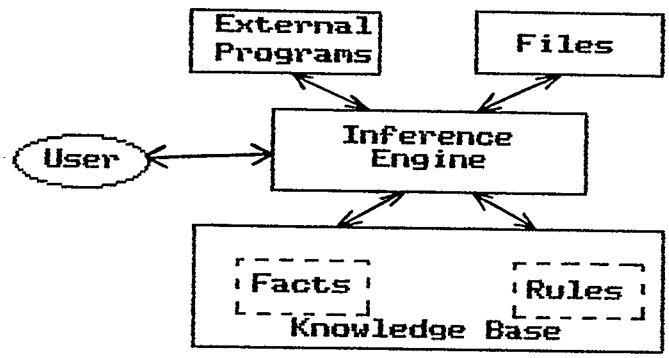


Figure 1.1 Expert system general structure

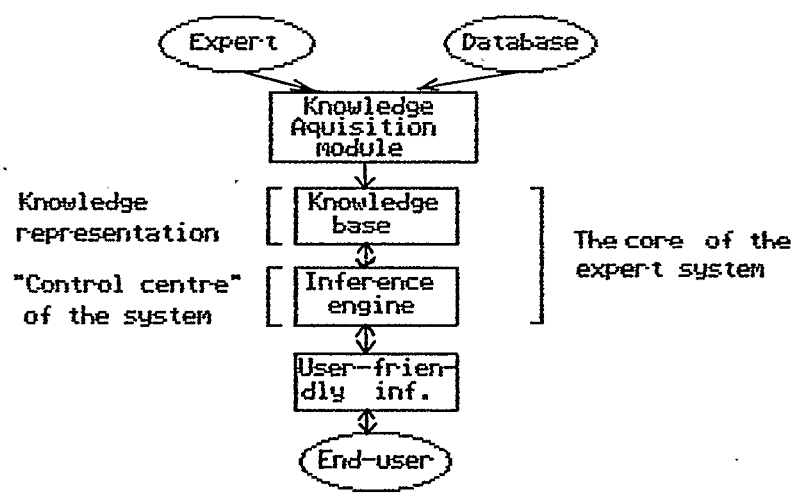


Figure 1.2 Expert system modified general structure

Knowledge acquisition will be discussed in more detail in Chapter 2 of this thesis.

1.2.2. The Knowledge Base

The knowledge base embodies the knowledge required to provide expert advice in a certain expertise domain. It contains facts and rules. Facts are short term information that combine to form the expert's knowledge. Rules are the set of relationships that determines the combination of facts leading to a specific expert advice. This approach differs from the conventional data base methodology in the aspect of creativity. Facts in a data base are normally passive in nature, i.e. they are either present or not available. On the other hand, a knowledge base actively tries to fill in the missing information through the rules [6]. Due to the nature of the expertise domain, the information in the knowledge base is subject to change. Hence, the knowledge base undergoes a long-term maintenance process in which the information is updated. This modification is realized by either the occasional supervision of the knowledge engineer or by build-up the expert system to possess a self-learning capability. Self-learning implies that the system is capable of updating the facts and rules of the knowledge base by interacting with the available information sources [1]. The facts that a knowledge base contains are not solely fragmented pieces of information. These facts are acquired from expertise sources through knowledge acquisition modules. The knowledge base does not contain these facts in a random fashion. On the contrary, this knowledge is represented in the knowledge base according to specific knowledge representation schemes. These schemes vary depending on the suitability and the degree of optimization that the chosen knowledge representation scheme have with respect to emulating the expert's knowledge of the expertise domain.

In addition, a knowledge base can be composed of several knowledge bases interacted in a manner that does not disrupt the unity of the expertise domain.

1.2.3. The Inference Engine

A control mechanism is needed to establish the interactive mode between the knowledge base with its facts and rules, external programs and files, and the end user. The inference engine establishes such a mode by inferring the rules embodied in the knowledge base that match the facts required to obtain an expert advice (Fig. 1.3). The inference engine infers the facts and reproduces them in a different form by questioning the user and interpreting the appropriate rules of relationship [2]. Forward and backward chaining are used as overall inference strategies. Forward chaining involves reasoning that establishes the status of facts in order to obtain the expert advice. The user is questioned about several facts, then the advice is projected. Backward chaining involves reasoning that develops the advice from previous incomplete facts and then prompts the user to establish the status of the facts related to that advice.

Expert systems employ either one of these inference strategies or a combination of both through their inference engines. The nature of these strategies implies that incomplete sets of facts can still lead to advice. Hence, a means of dealing with uncertainty is achieved through using expert systems which helps in dealing with the real world rather than an idealized abstraction as in the case of traditional data-systems.

1.2.4. The Information Sources

Information sources are the user, external programs, files, sensors, etc. in any combination. These resources provide the guidelines for the expert system rule-fact matching in order to give an expert advice. In the process of giving this advice upon a consultation, the expert system will obtain the sequence of facts required to match specific embodied rules from such sources. The variety of information sources widens the range of expert systems applications. The expert system has to adapt to the user, and this distinguishes knowledge-based programs from traditional data-based programs. The adaptation takes the form of flexibility in questioning the user and the ability of the expert system to interact with the user and give an expert advice even with an incomplete set of facts.

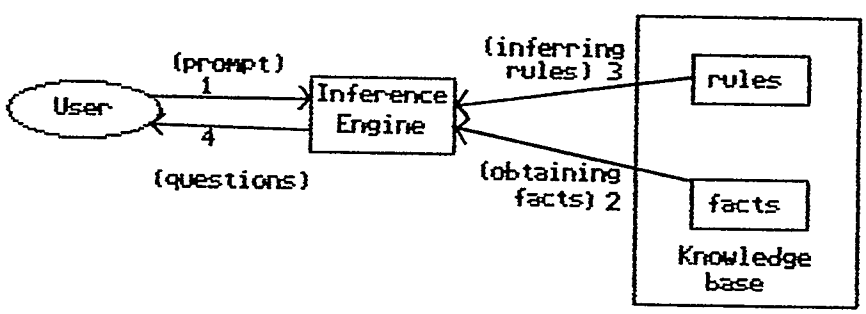


Figure 1.3 The inference cycle

1.3. Features of Expert Systems

The knowledge-based approach to system design introduced a significant change that led to impressive consequences. This approach replaced the software tradition of [6]

$$\text{Data} + \text{Algorithm} = \text{Program} ,$$

to a new form of

$$\text{Knowledge} + \text{Inference} = \text{System} \cdot$$

These consequences are evident in the features of expert systems. Some of the general intuitive features include high quality of performance of expert systems, relative speed of performance, and the implementation of heuristics or rules of thumb in the rules structure of the knowledge base. Heuristics are among the set of rules that an expert uses to advice on a problem. Other more characteristic features are as follows.

(1) The specialized and narrow domain of expertise.

Specialization is a natural outcome of comparing depth of knowledge to its width. It is apparently more difficult to tackle broad knowledge domains than narrow domains. For instance, expertise in technical fields is available, while it is not in everyday activities such as natural language or visual senses [5]. This feature has a direct impact on the spectrum of expert systems applications.

(2) The ability to reason with uncertainty.

This feature is in direct contrast with the traditional data-base system design which handles complete and certain data inputs. Even with an incomplete set of input facts, an expert system should be capable of providing an output expert advice. This is an outcome of the nature of knowledge-based system design which separates facts and rules enabling the designer to group either into sets that yield results despite of uncertainty levels.

(3) The inference mechanism is a separate entity from the knowledge base.

The facts in the knowledge base are not contained within the programming procedures that control the expert advice outcome. In other words, the facts and the inference mechanism are separated but both interact within the same system [6].

(4) An expert system is typically rule-based.

If-then rules with various formulations are the reasoning tool of the system. These rules contained in the knowledge base form the frame through which facts can be weighed and advice given.

(5) Expert systems are capable of explaining the advice.

The need for explanation depends on the advice given. In certain cases no explanation is needed, or one explanation can be given to all advices. Explanation can take several forms. The user can be prompted, upon request, with the rules and facts leading to the advice, or by an explanatory text or by any other form of explanation.

(6) The expert system output is typically an advice for a problem that the user faces in the expertise domain.

This advice can be a table of figures or graphic displays, but has to be typically formalized as an advice.

Other features whose effects on the concept of expert systems are still unclear, resemble those of a human expert such as the ability to learn from further experience, acquire general knowledge reasoning by analogy etc., but these features cannot be taken into account as they exist by the virtue of intelligence rather than expertise.

1.4. Expert Systems Application

Expert systems can be applied in any field where it is needed and is feasible to assist or replace human expertise for any justifiable reason, on condition that such an expertise can be embodied in an efficient expert system. This implies that such applications are numerous due to the immense variety of human intellectual activity. Because the success of expert systems is relatively recent, the current enthusiasm for expert systems technology implementation focuses on looking for applications of these systems in any direction. The right attitude is to apply expert systems for problems that need solutions once such an application is acceptable [1].

1.4.1. Selection of Expert Systems Applications

A criteria has to be developed upon which applications of such systems are to be chosen. The factors that are considered in selecting expert systems applications are basically the same as those considered in selecting any system project. The criteria for choosing applications of expert systems in this thesis are as follows.

(1) Worth of application.

This factor projects the feasibility of the application in its pay off and expected performance.

(2) Expert shell efficiency.

An expert shell is the system program that assists in build-up the knowledge base of the expert system. It also provides the inference engine. The flexibility, ease of use and knowledge base build-up procedure of the expert shell have a major impact on the chosen application (refer to Chapter 3 for more details on the expert shell).

(3) Nature of the expertise domain.

The easiest expert systems to build are characterized by their narrow, deep and deterministic knowledge domain [7].

(4) Availability of the experts.

Expert knowledge sources are naturally a basic element in build-up an expert system. Their availability, attitude, number and creditability are crucial for the success of the expert system application and build-up process.

(5) Availability of other expert systems build-up tools.

Many expert shells do not provide additional features that complement the expert advice output. For instance if graphic display output is required to complement the expert advice and the expert shell doesn't provide such a facility, then external simulation programming packages should be available to the expert system.

1.4.2. Expert Systems Fields of Application

Today, successful expert systems are employed in almost every industrial and commercial sector. The variety of applications is a direct consequence of extensive research and actual need for this aspect of AI [2]. Johnson [4], suggests that the promising fields for developing expert systems are as follows.

1. Computing, electronics and communications.
2. Oil exploration and extraction.
3. Financial services.

In medicine many expert systems have been developed and some with great success. Such expert systems include MYCIN, CASNET, INTERNSIT, PUFF which is a pulmonary-function program, HODGKINS which is a system for performing diagnostic planning for Hodgkins disease, VM which is an intensive care monitor, and many others.

In geology PROSPECTOR which is an expert system that helps geologists investigating in hard-rock mineral exploration, is one of the best known system developed for this field.

In chemistry, the DENDRAL expert system can identify the possible molecular structures of constituent atoms that could account for the given spectroscopic analysis of the module under investigation. Another system, CRYSALIS, focuses on protein crystallography, aiming to integrate various knowledge source to match the crystallographer's performance in electron-density-map interpretation.

In mathematics the system MACSYMA can tackle complicated mathematical problems and is already successfully operational in many academic institutes.

In engineering diagnostics CRIB is designed for computer fault diagnosis.

In computer aided design the list is long including DEC, XCON, EURISTO, and many others.

In business and finance the system TAXADVISOR can recommend tax plans for businessmen and other users. IMACS can aid in manufacturing while XSEL helps the salesman to develop system orders.

Military applications include AIRPLAN which plans military air-traffic movement and TATR is used for tactical air targeteering.

From this short list of successful applications in various fields we notice that expert systems are used for complex problems. Reports of the successes of these expert systems indicate that they can in many instances outperform the expert. The sources of error in the software are mainly due to faulty information programmed into the knowledge base (facts or rules), or a fault in the build-up process of the system. Otherwise, the expert system is not exposed to problems of fatigue or any other form of human error sources. Hence, expert systems are applied in commercial and industrial sectors with increasing success, growing trust and creditability.

CHAPTER 2

DESIGNING AND BUILDING EXPERT SYSTEMS

2.0. Introduction

A survey of available references on expert systems which investigate the design and build-up procedures of such systems shows that they fail to converge to a standard formalism of these procedures. The procedures presented range from being rather simple and intuitive in nature to complicated descriptions of system structures. Most of the design procedures are extracted from traditional systems analysis in data-based programming while the build-up procedures take into account the particular structure of expert systems and the tools used to implement them [8]. The experience acquired executing this work hints at the inevitability of assuming what follows prior to carrying out designing and building the expert system.

1. The designer possesses an adequate understanding of data-based systems and structural analysis, expert systems in general, and a background in programming. Additional requirements depend on the problem that the expert system is intended to handle. For instance, to build an expert system that advises on types of faults occurring on power transmission lines and suggests the most feasible and efficient ways to handle them, an electrical power engineer may be capable of an optimal performance in designing and building such a

system. The main factor contributing to this additional requirement is the expected ability of this engineer to understand the knowledge domain communicated from the expert in this field. Hence, a better design and more efficient structure can be hoped for.

2. The expert system build-up tools are available, and the designer has a fair idea of their performance and capacity.

2.1. Expert System Build-Up Tools

The expert systems market is accumulating a wide variety of means or tools used for building these systems. The purpose of this section is to categorize these tools and explain briefly the methodology of their use. The majority of expert systems build-up tools currently available fall into three different classes [9].

(1) Expert systems shells.

An expert shell is a program that assists the user to build the knowledge base and includes the inference engine. The knowledge base is built according to a certain knowledge representation scheme particular to the expert shell. The inference engine would implement either a forward or backward chaining inference strategy or a combination of both depending on the expert shell capabilities. The expert shell can provide a set of features such as the ability to link to programs written in certain programming languages or provide a built-in graphics or simulation facility or other features that can enhance emulating the expert and creating a user-friendly mode of operation. The knowledge representation capabilities, the inference strategies used, the user-friendliness, speed of operation and the features that the expert system provides, determine the level of complexity and degree of emulation that an expert system being built using the shell can accommodate. Expert shells can be used by non-programmers as the shell provides the basic structure constructed by experienced programmers who have formulated a solution for a problem similar to that of the shell user. The user may also require a background in programming to complement the expert shell.

Such a tool can in principle be adopted to a wide range of expertise domains. What enables expert shells to have this variety of applications is the universality of the basic inference strategies. These strategies can be applied to any form or method of representing knowledge. Some shells already on the market were based on rules developed for an existing successful expert system like MYCIN. Other shells are original and are developed without any expert system precedent. Many shells should be run on special-purpose workstations such as KEE, LOOPS and ART which require Lisp machines while others may require mainframe computers or just microcomputers like GURU, NEXPERT and 1st-Class expert shell. Examples of expert system shells available on the market include: APES, ART, BUGGY, COPE, CRIB, DMS, EMYCIN, EXPERT-EASE.

(2) High level programming languages.

These are general-purpose representation languages developed for knowledge-based programming paradigms. They are not as constrained as expert shells as they do not possess any particular framework or paradigms to represent knowledge or infer rules from knowledge bases [5]. Examples of high level programming languages include ROSIE, OPS5, RLL and HEARSAY-III.

(3) General purpose programming languages.

Using this class of expert systems tools requires extensive programming background. The expert system is built from scratch with symbolic computation oriented languages such as LISP and PROLOG in any of their dialects or with traditional languages such as FORTRAN or PASCAL. Symbolic computation languages are the principal programming languages of artificial intelligence. Using traditional programming languages in building expert systems is rather limited as they are data-based, but they can be used to complement the expressive ability of the system, especially if the advice output of the expert system is data related.

These tools in any combination are used to build expert systems. Special hardware, or workstations, have to be utilized in certain cases to accommodate this type of software limiting the availability of these tools. Recently, expert shells are being focused on, due to their general applications prospects, ease of use and commercial success. It has to be noted, however, that developing a large expert system with a high level of complexity requires employing symbolic computation languages such as PROLOG or LISP.

2.2. Expert Systems Design and Build-Up Procedures

Design and build-up procedures are the main stages of development in any system analysis, and expert systems are not an exception. These procedures are interactive in nature because the design is continuously modified according to the feedback from problems arising and new findings in the build-up process. Once the design is modified, the build-up is affected in such a way to yield an expert system that coincides with the initial objectives of the project as a whole. Prior to executing these procedures, two steps have to be taken [10].

(1) Project selection.

Out of several projects considered for development, one is chosen to be implemented as an expert system.

(2) Feasibility study.

In this study, the goals hoped for by building an expert system are outlined, the possible outcomes of employing the system are investigated and the main requirements and features of the system are defined.

The procedures to be discussed were specifically developed for designing and building an expert system using an expert shell as the basic tool. Generalization of the procedures is avoided as more research would be required which lies beyond the scope of this work.

2.2.1. Expert System Design Procedure

The main stages in the design procedure are as follows.

(1) Identification of the expert system.

With the chosen project in mind, the domain of expertise to be processed into an expert system is defined. The scope of work, problems that the expert system is expected to solve and the operating environment are specified as well.

(2) Identification of the expert knowledge sources.

These sources provide the expertise knowledge needed to build the knowledge base. As the expertise domain is well defined with all its projected constraints, the expert knowledge sources must be identified in order to determine their availability, accessibility and degree of contribution to the expert domain. These sources can be human experts, books and printed material written or edited by experts or any other communicable form that can be a medium for knowledge exchange in the identified expertise domain.

(3) Assessment of the expert shell.

This logically implies that the expert shell is available and is to be used as the major tool for building the expert system. The expert shell must be examined and tested thoroughly to determine its features, general capabilities, advantages and disadvantages. It is strongly recommended that a small expert system prototype is to be built as a testing mechanism in order to assess the expert shell. Despite the magnitude of the expected error, efforts must be invested to build this prototype expert system as similar as possible to the originally proposed one.

(4) Expert system modelling.

The last step in the design process is to construct the initial system model. This model should reflect the relationship between the sources of expert knowledge, the chosen expertise domain and the expected output using block diagrams or any other suitable form of relationship representation.

The design process is continuously modified as building of the expert system progresses. This modification is a natural consequence of the actual build-up which will refine the constraints of the expertise domain and detail the relations within that domain with respect to the abilities of the expert shell.

2.2.2. Expert System Build-Up Procedure

The main stages in the build-up process are as follows.

(1) Knowledge acquisition.

To supply the knowledge base with a set of facts and rules of the chosen expertise domain, knowledge has to be acquired from the proper sources. This stage will be discussed thoroughly in a separate section.

(2) Knowledge partitioning and refinement.

The set of facts and rules is carefully investigated then partitioned into problem groups. A problem group is a subset that includes the necessary information that can identify a problem and that outputs an advice within the constraints of that problem. A process of elimination or addition to the initial set of facts and rules depending on the constraints originally realized for the problems that the expert system is intended to handle, refines the acquired knowledge. This step aids in finding which form of knowledge representation is most suitable for implementing the expert system.

(3) Knowledge representation.

Once the expert knowledge is partitioned into problem groups, the expert system developer must represent it using one of the acknowledged representation schemes. Knowledge representation will be discussed in a separate section, but it is important to note that every expert system shell has a specific knowledge representation scheme and this must be taken into account prior to implementing the expert system.

(4) Knowledge base build-up.

Each expert shell usually has a knowledge base build-up facility. Once the expertise knowledge is represented according to the scheme required by the expert shell, the expert system developer can utilize the build-up facility to construct the knowledge base with its facts and rules. If the expert shell lacks the expressive ability required to complement the expert system, external programs may be used to compensate for this disadvantage.

(5) Expert system implementation.

At this stage the knowledge base and external programs are ready. With the expert shell providing the inference engine, all these components are grouped or integrated to form the expert system. This grouping cannot be successful unless the knowledge representation and rules of the knowledge base are compatible with the inference strategies employed by the inference engine of the available expert shell.

(6) Expert system testing and evaluation.

The expert system is tested and evaluated for efficiency, degree of emulation of the expert, correctness and validity of its advice and creditability.

After testing, if the expert system demonstrates acceptable levels of performance, then it is said to be operational. The knowledge base should be occasionally modified to update the expert knowledge in order to project any advances or changes in the expertise domain. Figure 2.1 illustrates the different levels of the design and build-up procedures of an expert system using an expert shell as a basic build-up tool.

2.3. Knowledge Acquisition for Expert Systems

Knowledge acquisition for expert systems is the transfer and transformation of knowledge from an expert knowledge source to a program which primarily outputs advice [5]. The person who performs this function is a knowledge engineer. Knowledge acquisition or elicitation can be automated using acquisition models and modules. The expertise knowledge can be elicited from several sources, mainly human experts. Other sources include empirical data, case studies, textbooks or documentation of the human expert's own knowledge. The process of knowledge acquisition is time consuming, tedious and the logistics of which are not well understood [9]. Many references on expert systems describe this process as being the "bottleneck" in the construction of expert systems. It is of critical importance for the expert system developer to have a substantial understanding of

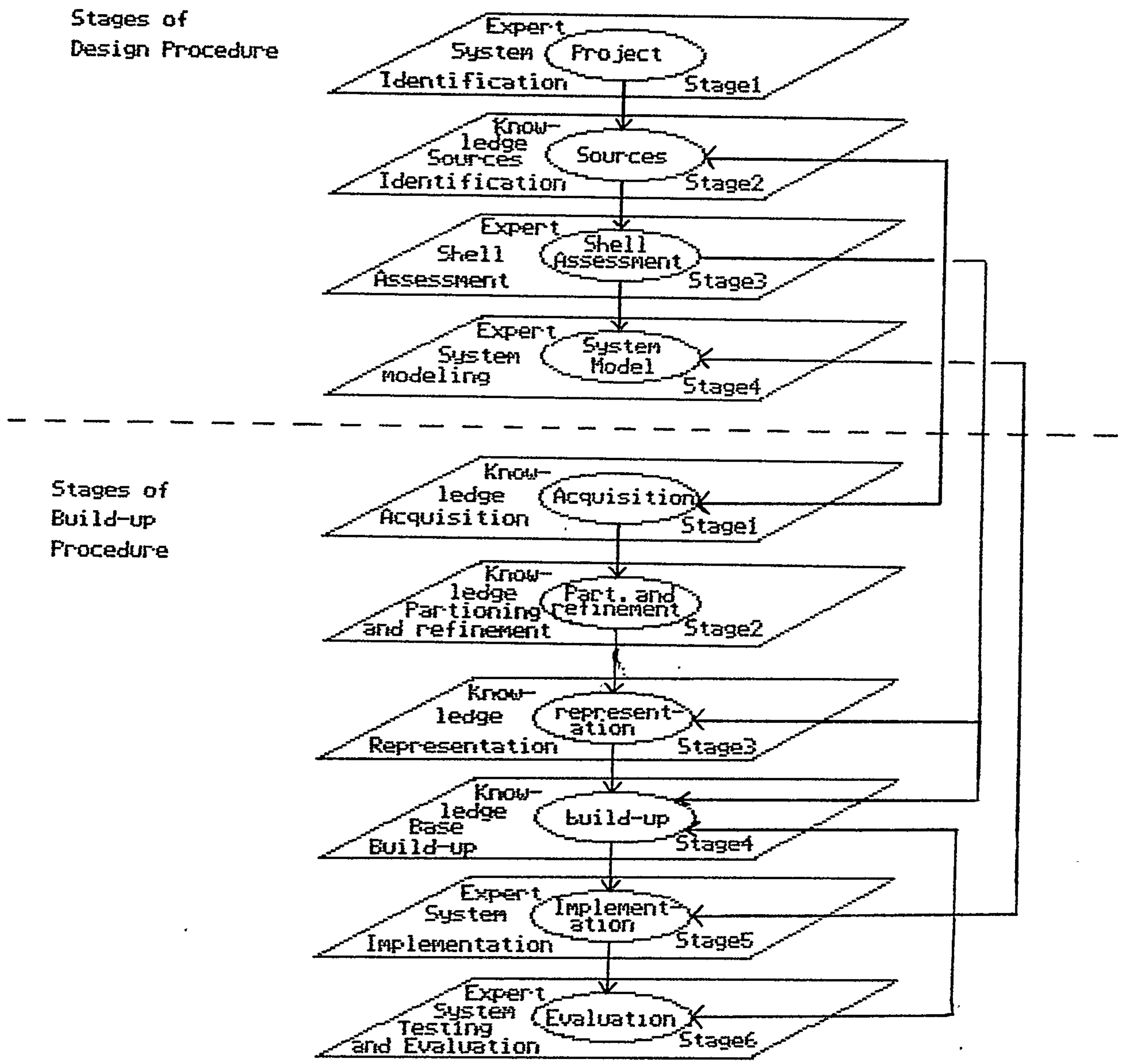


Figure 2.1 Stages of design and build-up procedures

knowledge representation in general and the knowledge representation scheme of the expert shell to be used as a major build-up tool prior to knowledge acquisition.

Some references [5,9,10] offer an analysis of knowledge acquisition integrated in a process model of how to build an expert system. The experience of actually building an expert system within the scope of this work showed that separating the two processes (knowledge acquisition and expert system build-up) simplifies integrating the acquired knowledge in the knowledge base. While knowledge is being collected in the form of facts and rules, special care has to be taken to represent it in a way that matches the knowledge representation scheme that the expert shell may accommodate. As mentioned earlier, an expert shell may be capable of accommodating several knowledge representation schemes. Hence, it is the duty of the knowledge engineer to research into several possibilities of representation and choose the most suitable scheme. The criteria that determine this choice will be discussed in the knowledge representation section.

2.3.1. Means for Knowledge Acquisition

Knowledge of an expert can be elicited in several means or ways. Group discussions, private interviews, questionnaires, sampling records, extensive observation, conducting tests once required, face-to-face discussions with expected users of the expert system and individuals involved in the expertise domain or any other suitable means that can accomplish the knowledge transfer from the source, are ways of eliciting expert knowledge. The effective combination of these means vary according to the expertise domain and types of information sources.

Many problems ranging from lack of cooperation of the experts to misunderstanding of the knowledge due to inadequate backgrounds on part of the expert system developers may infest the knowledge transfer process. These problems must be overcome by the knowledge engineer who apparently should possess good communication skills and trouble-shooting capabilities. Analysis of such problems, their remedies and the comprehensive characteristics that a knowledge engineer must possess in order to execute the acquisition in an efficient manner is an area undergoing extensive research efforts, but is beyond the scope of this work.

2.3.2. Modes and Modules of Knowledge Acquisition

There are two modes to acquire knowledge: indirect and automated.

(1) Indirect knowledge acquisition mode.

The knowledge engineer converses with the expert and extracts the relevant information about the knowledge domain from the available sources of information. This mode is indirect as the knowledge engineer is the transfer link who separates the domain knowledge from the rest of the knowledge acquired from all the information sources [5]. The knowledge is then conceptually grouped into simple information structures that follow a chosen knowledge representation scheme to form the knowledge base. Figure 2.2 shows the indirect mode of knowledge acquisition [5].

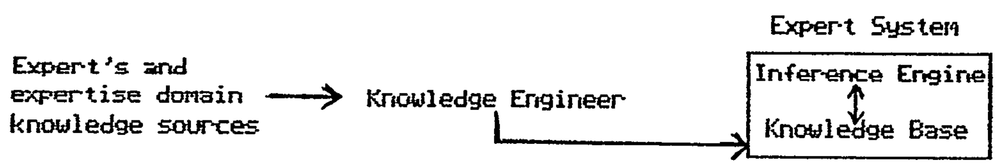


Figure 2.2 Indirect knowledge acquisition mode - Knowledge domain to knowledge base via a knowledge engineer.

(2) Automated knowledge acquisition mode.

Knowledge can be elicited directly from a source using a program that interacts with the knowledge source and builds the required knowledge base. This automated process is also referred to as machine learning, i.e. the machine learns from the expert knowledge source through special programs. Prototype machine learning programs exist, but very few are in general use for automatic knowledge acquisition in expert systems [10]. One of the major problems that faces this type of programs is the very wide range of technical vocabulary and the variety of expert knowledge sources. A knowledge acquisition program can elicit knowledge from a specific source or adapt to different types of sources. In addition, these programs can be utilized for a particular application area in an expertise domain or be flexible and have the ability to tackle a group of applications. Figure 2.3 illustrates the automated mode of knowledge acquisition [5].

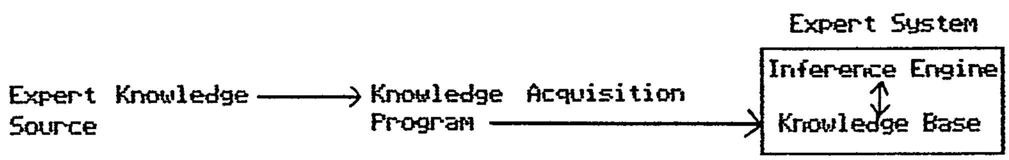


Figure 2.3 Automated knowledge acquisition mode - Knowledge domain to knowledge base via a knowledge acquisition program

A knowledge acquisition module is a standard procedure, indirect or automated, developed to elicit knowledge in a certain domain. A generalization of the module to cover different expert systems applications of the same expertise domain is possible provided that these applications share the same knowledge sources. A module is developed during the process of knowledge acquisition and is finalized once the expert system is operational.

2.4. Knowledge Representation in Expert Systems

Knowledge representation is of vital importance in building on expert system. The nature of expert systems requires that three criteria be met for knowledge representation [11].

1. The knowledge representation should be capable of expressing all the particularities and distinctions of the expertise knowledge domain. For example consider a knowledge domain that requires the use of "if not" statement to express a particularity or distinction of the domain. A knowledge representa-

tion technique must include "if not" statement in a simple form so as to be utilized to express all that knowledge domain.

2. All the knowledge domain should be accessible, even if the user is not familiar with programming. In other words, the knowledge must be available, through a representation, in a form and language that is understood by a user familiar with the knowledge domain, but not with the programming.

3. The knowledge representation should capture the nature of the structure of the knowledge domain. Experts utilize heuristics, or rules of thumb, in giving advice. The representation technique should be able to include these rules, and associated facts in an easy, comprehensive way.

Due to the large variety of human intellectual activities, knowledge can be represented using several techniques, such as frames, semantic nets, production rules, logic, common sense or suitable combinations of these methods. Most of the existing methods would be under one of the following principal schemes [11].

(1) Rule-based productions systems.

These are unstructured systems which group the facts and rules of the knowledge domain in sets that do not follow an organized structure. A production system consists of three parts.

(a) A global data base or working memory which holds permanent and dynamic information.

(b) Production rules. These rules are data structures of the form if

(A) then (B).

(c) Control system. The control system combines (a) and (b). It reads the global data base, then selects a production rule whose condition matches the data base.

Rule-based systems are typically used as goal-directed systems or data-driven systems. As the name implies in both cases, the system begins with a goal to be achieved or a set of data that triggers a specific rule to alter the data base and achieve a goal. Hence, rule-based production systems are efficiently applied to represent knowledge domains that exist in the form of independent parts which once combined constitute a knowledge domain.

(2) Structured systems.

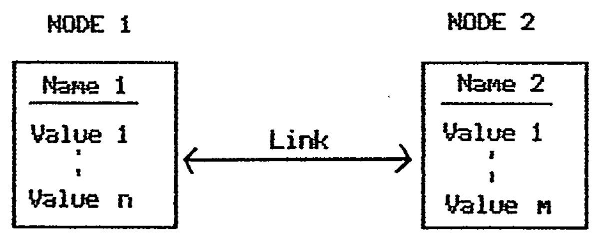
If the knowledge domain admits taxonomic relations, rule-based production systems are inefficient in expressing the domain. Structured systems use knowledge structures to represent knowledge. They are ideal for model-based systems and can express knowledge patterns such as set or classification memberships. A structured system can be conceived as a tree with branches and nodes. The root node of the tree represents a whole class of objects, non-terminal nodes stand for important sub-classes, and terminal nodes represent states of these objects. The branches form the links between the objects of the same class and can express their relationships, distinctions etc. Frames, semantic nets and frame-based conceptual graphs are typical examples of structured systems.

Frame-based conceptual graphs are of particular relevance to this work as they are the technique adopted for knowledge representation. This structured system is composed of nodes and links. A node represents a concept or an object and a link defines the relationship between two nodes [12]. A node is represented using a frame. A frame is a structure combining the knowledge about an object or a concept. The frame may be named, hence the node has the same name. Each frame consists of several values, where each value can be a certain property of an object or an identification of a concept. A frame-based conceptual graph is actually a combination of two techniques used in representing knowledge: frames and semantic nets [11,13], and is shown in Fig. 2.4.

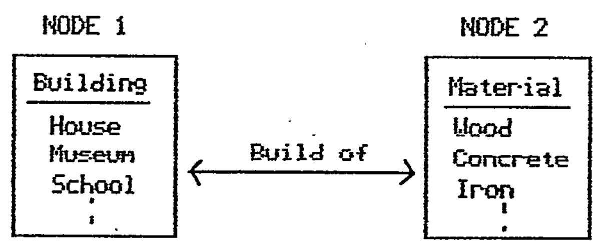
(3) Logic-based systems.

These are systems that utilize predicate logic languages to represent knowledge in the form of logic database. PROLOG and LISP are typical samples of such languages. A logic database is usually a set of clauses that represents special cases first, e.g. particular facts and exceptions; then general cases that include the greater bulk of the knowledge in the form of inference rules and then axioms.

Structured systems are widely used for knowledge representation in expert systems, whereas logic-based systems are the least used [9]. One of the few programs employing a logic-based system is MECHO.



(a) A general single frame-based conceptual graph



(b) A simple example

Figure 2.4 Frame-based conceptual graph technique.

2.5. Evaluation of Expert Systems

The available references that tackle the evaluation of expert systems agree that it is a difficult issue to address and discuss, but disagree whether a criteria for evaluation exists or not. The disagreement stems from the wide diversity of applications of expert systems which has a direct impact on the characteristics of expert systems. Yet, the initially declared objectives for building an expert system hint at criteria that can be used to evaluate the system. The assessment of the criteria relies on empirical data obtained from successive trials of the expert system. The number of situations that the system is expected to face once operational is very large especially for complex applications. Hence, this assessment tends to be inaccurate, and is reliable only to a certain extent. The criteria are as follows.

(1) Correctness and validity of the knowledge base.

Since the knowledge base yields the advice as an inferred combination of the facts and rules, this base must include correct and updated facts and rules. The advice or decision that the system outputs must be sound and must match an expert's advice for the same situation of the knowledge domain. If the knowledge base is correct, valid and outputs sound advice, then the system should be able to justify its advice by correct reasoning, i.e. it can display the rational premises that led to the advice. The degree of correctness and validity can be measured by actually executing expert system sessions.

(2) Efficiency of the expert system.

The performance of the system has to be reliable and fast. Speed is of major concern, for in certain knowledge domains like trouble shooting on production lines, it becomes crucial to obtain sound advice in a minimal period of time. The ability of the system to respond even when interrupted contributes to how efficient it is once operational. The efficiency of the system can be inspected on higher levels by measuring the degree to which the knowledge representation scheme has abided to the criteria of expressibility, accessibility and naturalness discussed earlier. This is a troublesome and difficult task especially if the knowledge domain is complex with a wide knowledge base that involves fuzzy knowledge. Fuzzy knowledge is facts with a high degree of uncertainty.

(3) User-friendliness of the expert system.

The user must be able to execute any session with ease, speed and without knowledge of programming, but with familiarity of the knowledge domain. Even in the case of abrupt interruption of an expert system session, the user should be able to obtain a reliable system output.

(4) Credibility of the expert system.

One of the major problems that expert systems face is credibility. Many users are reluctant to accept and utilize the expert system advice due to psychological reasons. MYCIN, the system used for diagnoses of rare blood diseases, could output advice that was 73% correct, outperforming many human experts, but many doctors who have used it preferred not to take the advice as it was critical pieces of information given by a "machine". This problem can be overcome by the ability of the system to explain its decision in a convincing manner. The expert system is more successful if it is credible.

(5) Feasibility of the expert system.

Cost-effectiveness must be justified for commercial expert systems as it is considered to be unjustifiable to invest in building and using an expert system if it is cheaper and easier to consult the human expert. An expert system is not feasible if it runs on special and unaccessible hardware. The question of feasibility takes different trends for expert systems built exclusively for research purposes. An expert system is said to be feasible if the research goals are beneficial and progressive in comparison to the existing recent achievements.

2.6. Expert Systems: An Example

As an illustrative example, consider the hypothetical problem of emulating an expert whose expertise domain is the diagnosis of the diseases that may infect pine trees growing in semiarid regions and the methods of treatment. This example is simplified to a minimal degree to avoid redundancy by assuming that only two diseases exist, each with a certain set of symptoms and a unique treatment method. Any unknown sympton will be beyond the expertise of the plant pathologist, and if the group of symptoms include known elements, then the plant pathologist can give an opinion with a specific degree of certainity. This degree of certainty is called the certainty factor (CF). The two diseases are code named $d1$ and $d2$. The methods of treatment are $r1$ and $r2$ respectively. The complete set of symptoms for $d1$ includes $s11$, $s12$ and $s13$, and for $d2$ includes $s21$ and $s22$.

The purpose is to build an expert system that can diagnose which one of the two diseases a pine tree may be infected with, and to give the treatment as its expert advice. The components of the expert system are as follows.

1. The user, who may be a forest inspector, a farmer or any person interested in this field.
2. The knowledge base, which includes facts about the different pine tree species existing in the climatic area mentioned above, the two diseases and their

symptoms, the recommended remedies and rules that combine facts leading to an advice which is typically a treatment method or a remedy.

3. The inference engine which uses a forward chaining inference strategy. This engine can use a backward chaining inference strategy or a combination of both strategies, but only one strategy is used due to the simplicity of the knowledge base.

An expert shell will be the expert system build-up tool which provides the inference engine as well. External data files and programs are not used since the advice does not include graphic displays or images. The design begins by identifying the expert system which was done above. The knowledge sources are references on plant diseases [14]. The expert shell is assessed to determine the inference strategy employed by the inference engine and the knowledge representation scheme that the shell accommodates. The initial system model is shown in Fig. 2.5.

The build-up process starts with acquiring the relevant knowledge from the source. No acquisition module is needed, and the facts are as follows:

Pine tree species:

$P\ 1$ = Aleppo pine, $P\ 2$ = Canary island pine,

$P\ 3$ = Monterey pine, $P\ 4$ = Japanese black pine.

Pine tree diseases:

$d\ 1$ = Poor condition, $d\ 2$ = Aleppo pine disease.

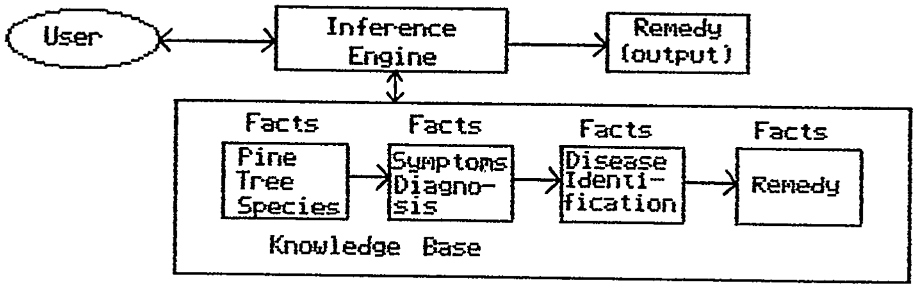


Figure 2.5 Initial system model of pine tree disease diagnosis expert system

Symptoms of d2:

d 2 = S11 drying of needles, *S* 22 = drying of twigs.

Remedy of d1 (output advice):

r1 = A mulch of two inches of corral manure applied to the root-area to the end of branches.

Remedy of d2 (output advice):

r2 = A mulch of manure, two or three months before the usual time of appearance of the disease, spread over twigs.

There is only one obvious problem group in this knowledge domain and it is diagnosing the symptoms. Knowledge base refinement shapes the facts and rules to accommodate all possible combinations of the symptoms. If only one symptom is present, then there is a 50% certainty of knowing the disease, and the remedy should be advised but with a warning that its application is at the user's risk. Hence, *r* 3 = Warning: apply the advice at own risk.

As mentioned earlier, there are many techniques to represent knowledge. Rule-based production systems may be used, but this means that every detail must be accounted for, and if the data base proves to be of considerable size (not in this example), then searching for the right rule consumes longer durations and may prove to be inefficient. Logic-based systems can be used as well, but because an expert shell is available, structured systems representation is the most feasible. The

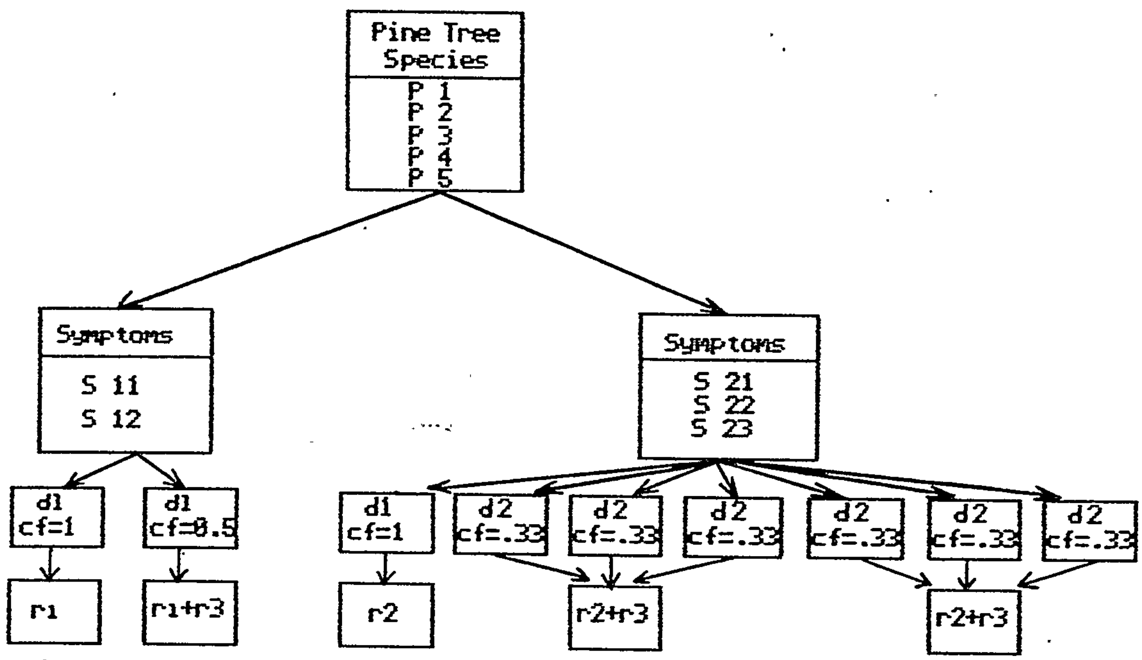


Figure 2.6 Frame-based conceptual graph representation of pine tree disease diagnosis expert system

frame-based conceptual graph technique will yield the hierarchical representation in Fig. 2.6 above.

With the knowledge base components available, the expert system developer builds the base using the knowledge base build-up facility of the expert shell. The knowledge base is then coupled to the inference engine and the system is tested and evaluated. As for the inference strategy, the inference engine tries to establish all the facts of a rule in a hierarchical manner until all the elements are satisfied, and at that stage, the system is able to assert a result and outputs it. This is forward chaining inference strategy.

CHAPTER 3

THE EXPERT SHELL

3.0. Introduction

The 1st-Class Expert System shell (1st-Class shell) developed by Programs in Motion Inc. [15] was used to build the knowledge bases for the expert systems constructed in executing this work, and was used as an inference engine. The shell is composed of two parts.

(a) A knowledge representation scheme and a knowledge base build-up facility. The shell takes the knowledge engineer through a series of 6 menu-based screens to build a graphic decision tree and the knowledge base.

(b) The inference engine. Once the knowledge scheme is designed and the knowledge base is built, the inference engine provided by the shell establishes the interactive mode between the user, knowledge base, external programs and files.

This chapter examines the expert system tools that were actually used to develop four different expert systems. The inference engine and build-up facilities system structure will not be discussed as it will not contribute to the outcome of this work. The discussion will focus on the characteristics and special features of the components of the expert shell.

3.1. The Knowledge Representation and Base Build-up Facility

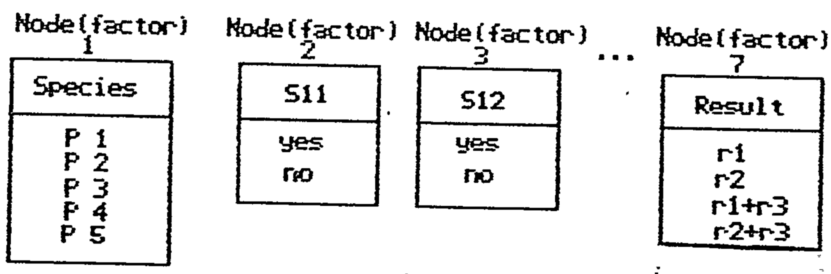
Knowledge is represented in this shell using the frame-based conceptual graph technique. As mentioned earlier, this technique declares a concept or an object in the form of a node. A node is represented using a frame, which is a structure that has several values, each representing a characteristic or property of the declared concept or object. The frames and their values are interconnected by links that define the relationships between the frames and values. This structure comprises the knowledge representation scheme, and is referred to as a conceptual graph or a decision tree. The knowledge embodied within this structure as facts in the frames and rules in the links comprises the knowledge base.

The 6 screens, menu-based build-up facility assigns the knowledge base and captures in a step-by-step procedure the main features of the technique after acquiring and partitioning the knowledge. The screens in order are the files, definitions, examples, methods, rule and advisor screens. This step-by-step procedure is as follows.

1. Node Identification.

This step is accomplished on the second screen, which is called the definitions screen. The nodes are defined and each is given corresponding values. Nodes are referred to as "factors" in the 1st-Class Expert System shell terminology. Two factors are already declared and reserved by the shell. The first is named "MEMO" and it serves to display preface text. This node is inactive, meaning that it is not a component of the decision tree. The second reserved node is named "RESULT".

The values assigned for this node are the end results of the decision tree, or the advice that the expert system outputs. Each of these values is considered a node which cannot be given any value by its nature. The shell allows a maximum of 32 nodes including the reserved nodes with 32 values for each node per knowledge base. Referring to the illustrative example in Chapter 2, the nodes, their names and values can be as follows.



The names of the nodes and the meaning of the values are a symbolic representation of the knowledge domain which is familiar only to the knowledge engineer. The shell allows these names to be edited on the definitions screen and this edited text is presented to the user during an advisory session. For instance, the edited text for the S 11 node can be: "Does your tree have thin foliage?"

2. Link Declaration.

The relationships between the nodes and values are declared on the third screen, called the examples screen, in the form of examples. These examples define the relations between different nodes, values or any combination of nodes and values. A maximum of 255 examples is allowed per knowledge base. Each example is entered by assigning a value for each node that corresponds to its relationship with the other facts in the same example. Once the last assignment is made in each example, the user is asked to enter a weight, or a certainty factor for the example in the 0 to 1 range. This factor is processed by the shell which will give the probability or degree of certainty that the expert system outputs if the example is involved. In the plant pathologist expert system discussed earlier, a typical example can be as follows.

	node							
	1	2	3	4	5	6	7	
example 5	$P3$	yes	yes	no	no	no	$r1 + r3$	weight = 0.66

This example can be interpreted as a production rule of the form if (A) then (B). This rule natural language interpretation is as follows.

If (tree is $P3$) and (symptoms are $S11$) and ($S12$) and (not $S13$) and (not $S21$) and (not($S22$)) then (remedy is $r1 + r3$) . C.F. = 0.66, which is equivalent to,

If (A1) and (A2) and (A3) and (A4) and (A5) and (A6) then ($r1 + r3$).

C.F. = 0.66.

A complete set of examples can be formed by determining the total number of examples possible with the different combinations of all the nodes and their values. Mathematically, there can be 5 * 2 * 2 * 2 * 2 * 2 = 160 distinct combination of the nodes (or factors). Hence, a total of 160 examples are possible, many of which have the same result or are redundant. To minimize this number, the shell provides a reserved result that can be shared by all the possible combinations of the values that were not included in the set of examples. This result is code named "no-data" because most of the redundant combinations share a result reflecting the inavailability of information in an area where the knowledge base is not prepared to handle.

3. Conceptual Graph Formation.

The conceptual graph, or decision tree, is formed using one of four methods provided on the fourth screen which is called the methods screen. The basic components of the knowledge base were formed in the last two steps, and what remains is to combine these facts and examples in an inferrable form. The 1st-Class shell automatically converts the knowledge base into a concise rule which can be utilized by the inferrence engine. This conversion is executed by any one of four methods. Hence, the result of the conversion is a knowledge base expressed in one rule which is actually a group of all the examples declared earlier. The resulting decision tree, which can be seen and edited if needed on the fifth screen or the rule screen, is the combination of two methods for creating knowledge bases: (1) the example-based, and (2) the rule-based methods. This tree is the basic rule that generates the required advice for a corresponding input. The four methods are as follows.

(a) The optimization method. For large knowledge bases, it is common to have factors which are redundant in certain examples. Instead of designing an elaborate decision tree, the shell can create an optimized or compacted decision tree by choosing the right factors in the right sequence. Optimizing the rule eliminates redundant factors and organizes the remaining factors into a sequence that asks the fewest questions. 1st-Class utilizes the ID3 algorithm [14] which inspects all the factors or nodes in the knowledge base and selects the factors with values leading to a certain result in order of importance and ignores irrelevant factors. In a simpler form, the optimization method transforms the knowledge base into a rule that would generate an advice or make a decision without using any redundant data. Hence, the route to the advice is the shortest possible, or is said to be optimized. If the optimization method is chosen to build the rule of the plant pathologist expert system any combination of the symptoms $S11$, $S12$ or $S13$ with $S21$ and $S22$ will be answered with "no-data" without asking about all the symptoms.

(b) The left-right method. This method is similar to the optimization method in yielding an advice without using any redundant or irrelevant factors, but it processes the factors in the order in which they appear on the definitions screen, from left to right, and not in the order of their importance or relevance to the advice. A compromise between those two methods can be achieved if the optimization method is run first, then the generated rule is modified on the rule screen (fifth screen) by editing it to suit the desired sequence of factors or nodes.

(c) The customization method. In case of a clearly-defined knowledge base structure such as a procedure manual, logic diagram, or an already established decision tree, the customization method is used. This method allows manual construction of the decision tree and consequently no examples are needed.

(d) The match method. There are two situations in which the methods presented above will not be able to generate a usable rule.

(i) When too many examples are needed to generate the rule such as in complex and larger knowledge domains.

(ii) When one or more results are obtained without a complete set of examples or with no examples.

The first situation will result in a large decision tree which becomes impractical to be viewed and examined on the fifth screen. The second situation will result in an advice being given after what seems to be an illogical sequence of questions. Applying the match method does not generate a rule but will allow asking all the possible questions, or invoking all the factors on the definition screen, in order from left to right. Thus, the match method is slower in operation but can handle the situations mentioned above.

The knowledge base is given a name which is automatically given a ".KMB" file extension name on the first screen. This format is used to save the knowledge base under the pre-assigned file name on screen four. Once the knowledge base is completed and saved, it can be tested on the sixth screen where the inference engine is used to handle a test advice session.

3.1.1. The knowledge base build-up facility features

The build-up facility provides a set of features distributed over the six screens. These features are as follows.

1. The ability to load three different file types on the first screen. These file types are the ".KBM" files, the ".KXT" knowledge base test files and the ".PRN" Lotus 1-2-3 print text files.

2. The shell provides a natural language editing facility through which nodes (factors), end nodes (results) and values can be edited. The editing follows the proposed knowledge representation in the decision tree to capture the desired expertise, and is done on the definitions (second) screen.

3. Numerical factors or nodes that include numerical factors can be entered, and the expert shell processes these numerical values by either sending them as answers to other knowledge bases or external files or receiving them as answers from other bases, programs or the end user.

4. Any knowledge base can be linked to another by forward and backward chaining techniques. This is the most important feature as it allows an expert system developer to build complex systems with a large knowledge base decomposed into a chain of knowledge bases.

5. A knowledge base can use other programming environments which perform functions that the shell cannot perform such as simulation or graphic displays.

6. The shell furnishes a series of special keys and words that enables functions such as global assignments for factors or text, command execution for external programs, backward and forward chaining and usage of the internal DOS system ERRORLEVEL, etc.

7. Statistical variables and weights can be embedded in the text to display useful statistics and text such as the total number of active examples, the number of examples that have led to a result, the result frequency which is the ratio of the number of times a result occurs in the rule to the total number of appearances of all the result values in the rule, the result probability which is the sum of the weights or certainty factors that caused this result divided by the sum of the weights of all the examples, the result relative probability which gives the probability of obtaining a result in case of multiple advices for the same set of facts, and several other functions.

8. Ability to activate and inactivate any factor or example to test their effect on the knowledge base.

9. To control the number of questions that the advisor asks during an advisory session, 1st-Class shell defines an Inference Cutoff number (ICO), which can be set to values from 0 to 9, where each value corresponds to a certain function, such as enabling the user to answer"?" (I do not know) to any question or when the match method is used.

10. The testing and debugging techniques are carried out by performing a "what-if" analysis, which allows the expert system developer to vary the facts (or values) of any example without affecting the knowledge base and observe the result. The advisor asks only the questions for the factors that were selected, but use the other facts of the example as is. Another technique is tracing the data back to its source is by highlighting on the rule or fifth screen the result that was obtained during an advisory session. Statistics and weights are used as debugging techniques as well.

11. The shell can be used to generate the ".ANS" answer file, and to automatically generate the ".RPT" session report file. The shell can be activated to build or generate these files after the completion of a session. The answer file serves as an input to another knowledge base that uses this file to give further advice which is actually a backward chaining process. The report file serves to give a summary of the advisory session which shows the values or facts of the examples that led to the advice during the session and the advice as well. This report file will not show the text associated with the data that it displays, hence, this file serves only to a certain degree in explaining the reasoning

behind a result or advice.

12. The shell can generate a ".KXT" knowledge base complete text file, a ".PNT" file especially formatted to generate readable documents and reports for printing, and a ".RUL" file used to print the rule (fifth) screen.

3.2. The Inference Engine

The inference engine, or the advisor as referred to in the expert shell, is a program that enables the user to interact with the knowledge bases and external programs by applying inference strategies. The 1st-Class shell inference engine has flexible inference strategies as the system developer can choose forward or backward chaining in any combination which increases the possibilities of applications, and which helps to build systems that can capture the expert's knowledge and reasoning to a wider extent. Forward chaining is realized by building the rule in such a way as to connect or chain to other knowledge bases in the results, factors or their values using reserved functions and words. Backward chaining is realized in the same way and also through the generation of answers files (.ANS) as explained above. The inference engine can also connect to external programs written in any language and can be called from other programming environments or MS-DOS batch files.

3.3. External Programs and Files

The 1st-Class shell does not provide a built-in graphics or simulation package. The expert system application chosen for this work required extensive graphic displays and simulations. Hence, in order to enhance the expressive capabilities of this shell, external programming languages were used. Enhanced desire [16], a continuous simulation programming language, was used to simulate the control system response. Turbo-Pascal was used to prepare tables and perform data analysis, whereas, Superkey [17] was employed to make the developed expert systems more user-friendly by reducing the amount of commands that the user has to type. The inference engine can connect to these programs within the environment of the 1st-Class shell, and once the function of the program is exhausted, the inference engine re-establishes the expert system original mode of operation.

3.4. 1st-Class Explanation Facilities for Expert System Reasoning

One of the fundamental characteristics of an expert system is its ability to explain the reasoning behind its advice to a user upon request. 1st-Class provides several explanation facilities that perform this function, making this feature one of its strong points. The nature of the advice in particular, and the knowledge domain in general, imposes the need for an explanation. Hence, the variety of these facilities enhances the flexibility of the expert system construction process. The explanation of reasoning is achieved as follows.

(1) Using the result text. This is a simple form of explanation, as it accompanies the result, or end node, edited text providing the user with the advice and the line of reasoning behind it directly on the same screen.

(2) Using statistics. As explained earlier, many statistical variables are available and can be added to an advice where probabilistic explanations are required. The values show directly on the screen with the advice.

(3) Using the example MEMO text. Specific explanatory text can be attached to any example using the reserved MEMO factor. The advice will be displayed with the example explanation text for all the examples that led to the advice, or for only the last example. This explanation facility is efficient in the case of chained knowledge bases.

(4) Using the review screen. A report file of the expert system advisory session has to be generated in order to use this screen. The review screen displays a summary of the questions asked, the answers obtained and the sequence of factors and their values that led to the advice. The display is carried out using the symbolic knowledge representation of the knowledge domain that the system developer has formulted. Hence, this explanation facility is suitable for the knowledge engineer only, and not for an ordinary user.

(5) Using controlled external programs. The shell provides a command statement that controls the execution of an external program by pressing the <F2> function key. An external program can be written to contain the appropriate advice explanatory text. Once the advice is displayed, a choice can be included on the same screen to display the explanation of the expert system reasoning process. To display the explanation, the user has to press the <F2> key which will execute the external program displaying the accompanying

text. This explanation facility is powerful as it reflects a total degree of freedom for the system developer as well as the user to obtain an explanatory text upon request.

(6) Building an explanation value into the question factor. This facility provides an explanation upon request, but is less powerful than the controlled external programs, as the user has to go back to the questions or choices menu and choose the explanation value to obtain the explanation.

3.5. 1st-Class Expert Shell Advantages and Disadvantages

Various expert systems were built using this expert shell in addition to the four systems developed and presented in this work. With an established reference of what can actually be achieved versus the originally planned objectives of building these systems, a set of advantages and disadvantages for using the 1st-Class expert shell as an expert system build-up tool can be formulated.

3.5.1. 1st-Class Advantages

(1) Ease of use. The 6 screens menu-based expert system build-up procedure is easily understood and used without any previous programming background. This furnishes a quick start in developing simple expert systems. A strong knowledge in programming is inevitable to develop complex systems.

(2) Use of natural languages to develop the knowledge bases, as no programming language is required. The text editing feature enables the expert system developer to use natural language in building the knowledge base.

(3) Clarity of knowledge representation. The combination of the rule-based and example-based methods of knowledge representation guides the expert system developer into partitioning the acquired knowledge and reconstructing it to fit the expert shell requirements in an easy and clear manner.

(4) Availablility of extra features. An expert shell must provide adequate tools to handle various knowledge domains. 1st-Class expert shell provides a set of features ranging from the ability to accept numerical factors to setting of global values. These features were discussed earlier in this chapter.

(5) Ability to explain the knowledge base reasoning in several explanation techniques. This ability gives 1st-Class expert shell its expert system tool identity and establishes any system built using this tool as an expert system. The explanation techniques provided by this shell were discussed earlier in a separate section, and their variety is an advantage because any combination of techniques can be used to explain the system reasoning depending on the particular application of that system.

3.5.2. 1st-Class Disadvantages

(1) 1st-Class expert shell is best suitable to represent taxonomic knowledge domains. The frame-based conceptual graph knowledge representation technique used in this shell is very efficient to represent knowledge domains which admit classification or categorization of objects or concepts such as plant and animal species, diseases or identification of organic compounds. If a knowledge domain requires analysis of any type, and reaches expert

conclusions or results in a deductive manner, then this domain cannot be represented efficiently using this technique of knowledge representation. Domains such as theorem proofs in trigonometry and Euclidean geometry require a production rule form of knowledge representation. Hence, the 1st-Class expert shell restricted knowledge representation technique which hampers its adaptability and efficiency in handling different knowledge domains.

(2) Inadequate expressive ability. The system developer has to resort to external programs and packages that furnish graphic displays and various computational needs which are not provided by this shell. If no external programs or packages are used, the 1st-Class expert shell can be used to develop simple and primitive expert systems.

(3) Limited knowledge bases. The shell allows 32 factors and 255 examples per knowledge base and 32 values per factor, which limits the ability to represent larger and more complex knowledge domains in a single knowledge base. The only way to extend a knowledge base is through chaining several bases to represent the complete knowledge domain. This limitation hampers the ability of an expert system developer to handle what is refered to as the effect of the "combinational explosion". To understand this effect, assume that a researcher could isolate ten general characteristics capable of describing a certain knowledge domain, where each has only three values. To represent this domain using 1st-Class requires at least $3^{10} = 2187$ examples to exhaust all

the possible combinations. This shell provides 255 examples as a maximum limit. The knowledge domains that admit multiple descriptive characteristics with several values are common. Consequently, the limitation imposed by 1st-Class on the knowledge base size can lead to complications in knowledge representation.

(4) Inefficient backward chaining strategy implementation. Expressing large knowledge domains requires forward and backward chaining. If the backward chaining is used in a case where a "no-data" or "no-result" is the chained objective, the inference engine activates an error message indicating a "no-data" result. This message is unavoidable even if the system developer wants to replace this result with a different edited text that expresses the advice output more clearly. This limits the efficiency of backward chaining.

(5) Difficulty in modifying or correcting the knowledge base. For a large knowledge base composed of several bases, the expert system developer faces considerable time consuming obstacles if modification of the examples, definitions, or values is required. The knowledge engineer has to go through the 6 screens to implement even a slight error or adjustment.

The assessment of the 1st-Class expert shell can be based on this set of advantages and disadvantages. More elements could have been added to the set provided that a wider variety of expert systems were built using this shell. The complete assessment and evaluation of the shell is avoided at this stage, and is left for the concluding parts of this work.

CHAPTER 4

EXPERT SYSTEMS FOR LABORATORY EXPERIMENTS IN CONTROL

4.0. Introduction

ADVISOR1, ADVISOR2 and ADVISOR3 are three expert systems built to assist in executing three laboratory experiments in control systems with a minimum degree of error. These experiments are conducted by fourth year electrical engineering students at the University of Calgary as a partial requirement for the ENEL 541 course.

Expert systems have been successfully applied in the field of electrical engineering. Their applications are numerous, and seem to be promising and beneficial. Three major topics were considered as potential application fields for expert systems. The first topic was power systems with two possible applications. The first application is the optimization of cost and performance of electrical power distribution networks, by finding which underground cable loop configuration is most suitable for a certain residential site under construction, in a way that will minimize the cable length and equipment needed. Another application is to build an expert system capable of detecting and locating network faults and assisting in fixing the fault by organizing repairs. The second topic was the design of feedback control systems with the aim of building an expert system capable of analysing the design criteria then giving optimal design parameters for a certain plant. The third topic was laboratory instruction for electrical engineering courses. The main goal was to develop an expert system that is capable of assisting, or even replacing, the laboratory instructor who analyses a problem, either theoretical or technical, that a student may face while conducting a laboratory experiment, and gives an advice to solve that problem.

The three topics were carefully weighed and the choice made was influenced by the feasibility and originality of applying expert systems in one of the three topics. The application of expert systems in power systems lacks originality in one case, and feasibility in the other. The NEWMEDIUS expert system, from IAL Data Communications, locates network faults and aids in fixing them by organizing repairs [2]. As for the application in the loop optimization problem, its feasibility is limited to consulting firms that design such loops. The application of expert systems in the design of feedback control systems is a feasible research topic, but lacks originality. The application of expert systems in laboratory experiments instruction is feasible and original. With the increase in the number of students and the constraints on budgets and availability of qualified personnel, the development of an expert system for instruction will be useful and feasible. Laboratory instruction involves theory and its applications in certain experiments. Hence, this application covers the development of an expert system for the revision and explanation of theoretical aspects and the solution of experimental problems. Even though many systems have been developed for computer-aided instruction [18], this application is original, because it employs expert systems. In contrast to traditional programs for instruction, these expert systems maintain a structure that has the potential for further expansion. The application of these expert systems can be generalized to cover different laboratory and experimental environments. This contributes to the feasibility of this application of expert systems.

Prior to the analysis of the three expert systems, the knowledge domain environment, namely the laboratory, is introduced. The laboratory environment is basically controllable especially when experiments are conducted for educational purposes. Such experiments conducted in the laboratory at an undergraduate level are set, analysed and tried prior to their execution by the students, who repeat these experiments and reproduce their results. A laboratory instructor usually supervises the execution of these experiments and makes sure that they are carried out properly. If any technical problem arises, the instructor is expected to provide solutions and theoretical explanations upon request. These solutions and the accompanying theoretical explanations constitute a part of the laboratory instruction knowledge domain which tends to be practically complete and manageable after an established record of laboratory sessions in which a certain set of experiments is conducted. Consequently, most of the problems facing the students during these laboratory sessions are known and their solution has already been determined. Hence, such a knowledge domain can be characterized by being deterministic to the extent that the degree of uncertainty in negligible. This fact does not imply that all the knowledge domain of laboratory instruction is deterministic. It is illogical to assume that all the problems that may face a student while conducting a controllable experiment can be accurately predicted. A non-predicted problem implies its infrequent occurrence, and is the source of uncertainty in this knowledge domain. For experiments which are procedurally executed, the uncertainty level tends to be minimal due to the rigidity of the execution. In such a case, the sources of uncertainty stem from the external factors that may affect the execution of the experiment. To correct this situation, a set of assumptions and constraints can be formulated to regulate this level. As a final conclusion, the laboratory instruction knowledge domain is generally deterministic and admits a low uncertainty level that can be regulated and controlled if the experiment execution is procedural.

The short term feasibility prospect is to build expert systems to assist in laboratory instruction. The long term feasibility prospect is to design the system structure in a manner that generalizes its use to cover different laboratory instruction requirements. ADVISOR1, ADVISOR2 and ADVISOR3 expert systems were built following the design and build-up procedures presented in Chapter 2 with both feasibility prospects in mind. In this chapter, the three expert systems are discussed, analysed and evaluated.

4.1. The Knowledge Domain, General Assumptions, and System Requirements for ADVISOR1, ADVISOR2 and ADVISOR3

The three expert systems share the control laboratory instruction knowledge domain. The control laboratory was chosen for four reasons.

1. The availability and easy access to the expert knowledge sources,
2. the familiarity of the expert systems developer with control systems,
3. the potential benefits provided that the systems are successfully operational, and
4. the availability of an effective testing facility.

The students are obliged to conduct these experiments in control systems as a requirement to complete the ENEL 541 Control Systems II course. This course discusses various mathematical techniques used in the analysis of control systems (e.g. z-transform), basic control concepts (e.g. time domain analysis, stability), computer applications in control, different control systems, digital simulation, and several methods of control systems analysis. The student performs three experiments in control systems at a stage when this student has developed a theoretical background which provides enough tools to analyse these experiments and understand their results. The control laboratory environment - in relation to these experiments - is basically the experimental set-up with all the necessary instrumentation. The laboratory instruction is confined to problem solving and general assistance during laboratory sessions. The instructor is expected to reproduce and explain the results of these experiments successfully, and to detect the sources of problems and offer solutions.

This knowledge domain is obviously simple to describe since the major component is the familiar duty of laboratory instructors. However, detailing this knowledge domain involves cumbersome problems, since the expert systems developer must simulate in an efficient manner an actual laboratory session with the most frequent questions asked and the answers given. These experiments are procedural in nature, and as mentioned earlier, the major source of uncertainty in building an expert system for such a knowledge domain is external factors. To diminish this source, a set of general assumptions was formalized restricting the use of these systems in order to increase the efficiency of their implementation. These general assumptions are as follows.

1. The experimental set up corresponding to an experiment is prepared prior to the laboratory session by the instructors.

2. The student, or the end user of the expert system, is expected to possess a basic knowledge of different measuring devices such as digital voltmeters (DVM) ohmmeters, the experimental set-up and the oscilloscope.

3. The experimenter must possess a minimum theoretical background related to the experiments. Basic concepts of control and an understanding of a control system are required.

4. The experimenters must perform all the necessary wiring before proceeding with the experiment. They also should be familiar with the experimental procedure by reading the laboratory hand-out sheet.

The system hardware requirements were discussed in Chapter 2. The software requirements are the "ADVISOR.EXE" execution file which is the inference engine provided by 1st-Class shell, the knowledge base of the expert system, MS-DOS operating system, the Turbo-Pascal version 3.0 system implemented for the PC-DOS, MS-DOS, CP/M-80 operating systems, the "KEY.COM" SUPERKEY command file, the "DJSC.EXE" ENHANCED DESIRE continuous simulation language execution file, external programs, files, and batch files to control the operation of the system.

4.2. ADVISOR1, ADVISOR2 and ADVISOR3 Design Procedure

ADVISOR1, ADVISOR2 and ADVISOR3 have the same design procedure because they share the same knowledge domain and the same laboratory instruction environment. The design procedure outlined in Chapter 2 was accurately followed as insignificant interaction occurred between the design and build-up procedures due to the nature of the knowledge domain. This minimal interaction resulted in the set of assumptions that made the knowledge domain manageable and minimized the uncertainty level. The design procedure is as follows.

(1) The expert system

The laboratory instruction for the control systems experiments of ENEL 541 is the knowledge domain of the three expert systems. The three expert systems enable a student to conduct the experiments by allowing the student to check observations or results made using simulation and to review the theory. The student is assisted in conducting these experiments with a reduced degree of error by guiding the experimenter in circuit connections, and parameter adjustments and tuning. The objective of these expert systems is not to replace the laboratory instructors, but to decrease the instructor's working load during a laboratory session.

(2) The expert knowledge sources

The laboratory instructors* were the primary sources of information concerning the problems usually encountered during a laboratory session and their solutions. Technical manuals of the equipment [19,20] used in each experiment, laboratory hand-out sheets, reference texts [21,22], and feedback from the students were the secondary sources of information.

(3) The expert shell

The 1st-Class expert shell is efficient in representing taxonomic knowledge domains since it utilized the frame-based conceptual graph technique for representing knowledge. The three experiments are conducted procedurally. Consequently, 1st-Class shell can be used to represent the structure of these experiments since procedures are a form of taxonomic knowledge. The knowledge bases are expected to be of moderate size, and the shell is expected to accommodate all the facts and rules in one compact knowledge base. If the knowledge base exceeds the shell limits, inference strategies can be used to create several chained knowledge bases. The ability of the shell to link its environment to other programming environments must be used to simulate the control systems and execute various computations. The advices of an expert system during a session are not related,

* Dr. M.H. Hamza and Mr. B. Yeung of The University of Calgary.

and this gives a higher degree of flexibility in building these expert systems and editing their results. A prototype model of a laboratory instruction expert system was built at this stage. The nodes in the prototype knowledge base represent the experiment parts with values representing sub-parts.

(4) The prototype laboratory expert system is shown in Fig. 4.1.

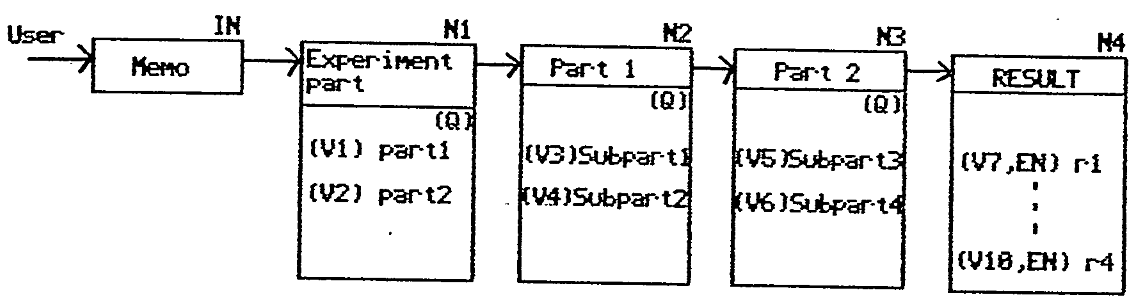


Figure 4.1 The prototype laboratory instruction expert system description initial model

4.3. ADVISOR1, ADVISOR2 and ADVISOR3 Build-up Procedure

The knowledge sources provided all the base for the process of knowledge acquisition. The experimental procedures were taken from the laboratory hand-out sheets, since they guide the students in conducting these experiments in a standard form. The laboratory instructors outlined the common problems encountered during a laboratory session and their solutions. All the experiments were conducted by the expert systems developer numerous times, and the technical manuals of the equipment used in these experiments were thoroughly referenced for solutions to the problems faced. The conducting of the experiments during actual laboratory sessions was carefully monitored. Questions asked by students and the instructors' answers were recorded. In addition, the laboratory reports written by the students to evaluate the results of an experiment were inspected to identify which parts of the theory requires more explanation and revision. The reference textbooks provided the theoretical explanations and the information required to simulate the control plant.

At this stage, enough knowledge had accumulated to be used for the laboratory instruction expert system development. The knowledge was divided into two major problem groups: circuit adjustment to achieve the required system response and display of that response in the time domain. Related problems such as analytical explanations and background theoretical comments were considered to be supportive information that complemented the knowledge and a minor problem group. The refinement of the knowledge was through integrating heuristics provided by the instructors which made a part of the acquired knowledge redundant.

To represent the partitioned and refined knowledge, the frame-based conceptual graph technique was used on all the systems without exception since it is enforced by the 1st-Class shell knowledge representation scheme. The knowledge bases were then built using the 1st-Class shell. External programs were written to simulate the control system plant response and to perform control system analysis. Each expert system was then completed in the final build-up and implementation stage by grouping the knowledge bases and external programs into one compact knowledge base. Each expert system, its knowledge base, and the testing and evaluation results are discussed in detail in separate sections. Figure 4.2 shows the different stages of the design and build-up procedures and their dynamic relationship. However, it must be noted that these dynamic relationships were not prevalent since the major part of the assumptions was foreseen, prior to the build-up process. The adjustments introduced to any of these systems after testing were mainly the sequence of factors on the definitions screen to make that sequence in the same order as the experiment execution procedure.

4.4. The Control System Analysis and Simulation

The three control experiments share the same plant which is an armature controlled dc motor. The controller is either a relay that can have several characteristics or a microcomputer. The system is loaded by magnetic discs or is unloaded. The system input is adjusted using potentiometers, and a pre-amplifier is coupled to the motor to amplify the system input gain. Figure 4.3 shows the general form of this control system .

The control system can be either a closed loop system or an open loop system ($H \rightarrow 0$). This system can be assumed to be a second order linear system, since the only sources of non-linearity are the operational amplifier and the potentiometers. The open loop system is a minor part of the first experiment only, and accordingly no analysis is performed for such a system. As for the closed loop

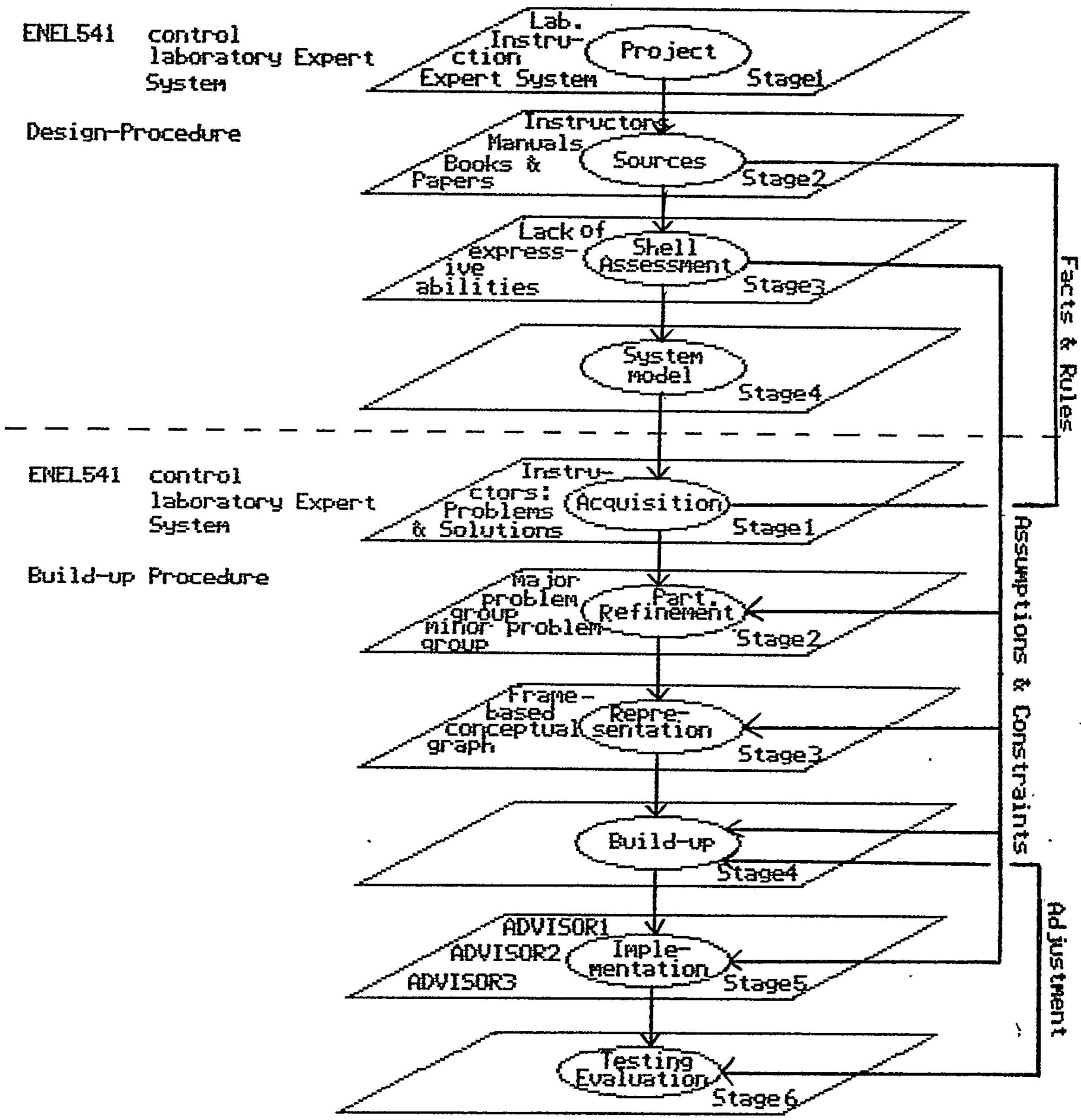


Figure 4.2 ENEL 541 Control Laboratory expert system stages of designs and build-up procedures and their dynamic relationship

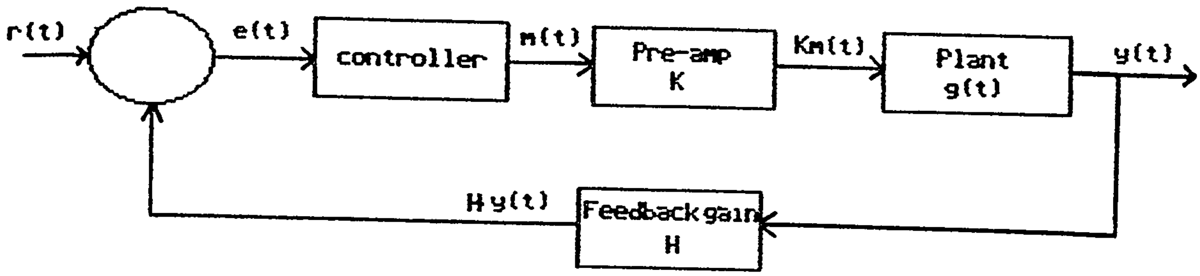


Figure 4.3 The control system

system, the three experiments mainly portray position control of a dc motor. In a position control system, the motor rotates an output shaft to the same angle as an input shaft. In the three experiments, an output potentiometer is coupled to the low speed dc motor shaft through a shaft coupler. The output potentiometer generates a voltage signal proportional to the motor shaft alignment. The closed loop is completed by achieving negative feedback through connecting this signal to the input of the system by using an adder. The reference input of the system is an input potentiometer. The error signal resulting from the reference input and the negative feedback output is used as the controller input. The student must display the response of the system in the time domain with respect to different parameters such as time and error signal. The potentiometers have track angles of 300^o and the total voltage across them is 30 V, giving 0.1 V/degree. Assuming that these potentiometers are linear, the error signal $e(t)$, is given by

$$e(t) = K_e \; [\Theta_i(t) - \Theta_o(t)] \quad , \tag{4.1}$$

where K_e is the error factor of a potentiometer, Θ_i is the angular displacement reference input in radians, and Θ_o is the output angular displacement of the dc motor shaft in radians.

For an armature controlled dc motor, the differential equation governing the motor speed is,

$$\tau_m \; \frac{d \; \omega(t)}{dt} + \omega(t) = K_m \; v_a(t) \quad , \tag{4.2}$$

where τ_m is the motor time constraint, K_m is the motor gain constraint, $\omega(t)$ is the angular velocity of the motor, and $v_a(t)$ is the motor input armature voltage.

Since the input armature voltage is proportional to the error signal, then

$$\theta_a(t) = Ke(t) = K \; K_e \; [\Theta_i(t) - \Theta_o(t)] \quad , \tag{4.3}$$

where K is the operational amplifier gain. In addition,

$$\omega(t) = N \; \frac{d \; \Theta_o}{dt} \quad , \tag{4.4}$$

where N is the output potentiometer-motor coupling gear ratio. Combining the above equations yields,

$$\tau_m \; \frac{d^2 \; \Theta_o}{dt^2} + \frac{d \; \Theta_o}{dt} + K_v \; \Theta_o(t) = K_v \; \Theta_i(t) \quad , \tag{4.5}$$

where $K_v = K \; K_e \; K_m / N$ is the velocity error constant which determines how fast the output shaft rotates for a constant error. The normalized form of a second order differential equation is given by,

$$\frac{d^2x}{dt^2} + 2\,\zeta\,\omega_n \frac{dx}{dt} + \omega_n^2\, x = \omega_n^2\, f(t) \quad . \tag{4.6}$$

Comparing (4.5) and (4.6) yields,

$$\frac{d^2\,\Theta_o}{dt} + 2\,\zeta\,\omega_n \frac{d\,\Theta_o}{dt} + \omega_n^2\,\Theta_o(t) = \omega_n^2\,\Theta_i(t) \quad , \tag{4.7}$$

where ω_n is the undamped natural frequency and ζ is the damping ratio, and are given by (4.8)

$$\omega_n = \sqrt{K_v/\tau_m} \quad , \quad \zeta = (2\sqrt{K_v\;\tau_m})^{-1} \quad . \tag{4.8}$$

A second order linear time invariant system is expressed using the Laplace transform by

$$G(s) = \frac{a_o}{b_o + b_1\,s + b_2\,s^2} \quad . \tag{4.9}$$

From Fig. 4.3, taking the Laplace transform of the elements of the control system yields,

$$Y(s) = KM(s)\,G(s) = \frac{a_o K\;M(s)}{b_o + b_1\,s + b_2\,s^2} \quad , \tag{4.10}$$

where $Y(s)$ is the system output. Taking the inverse Laplace transform of (4.10) yields,

$$b_o\;Y(t) + b_1 \frac{dy}{dt} + b_2 \frac{d^2y}{dt^2} = a_o\;K\;m(t) \quad . \tag{4.11}$$

If the output is considered to be the dc motor angular displacement $\Theta_o(t)$, then

(4.10) becomes

$$\frac{d^2\,\Theta_o}{dt^2} + \frac{b_1}{b_2}\,\frac{d\,\Theta_o}{dt} + \frac{b_o}{b_2}\,\Theta_o(t) = \frac{a_o}{b_2}\,K\;m(t) \quad , \quad b_2 \neq 0 \tag{4.12}$$

comparing equations (4.7) and (4.12) yields,

$$2\,\zeta\,\omega_n = \frac{b_1}{b_2} \quad , \quad \frac{b_o}{b_2} = \omega_n^2 \quad , \quad \frac{a_o}{b_2}\,K = \omega_n^2 \;. \tag{4.13}$$

If $b_2 = 1$ and $K = 1$, then

$$G(s) = \frac{\omega_n^2}{s^2 + 2\,\zeta\,\omega_n\,s + \omega_n^2} \quad , \tag{4.14}$$

$$d^2\,\Theta_o/dt^2 = \omega_n^2\,m(t) - 2\,\zeta\,\omega_n\,d\,\Theta_o/dt - \omega_n^2\,\Theta_o \quad , \tag{4.15}$$

where (4.14) describes the normalized Laplace transform of a linear, time invariant, second order system, and (4.15) can be used to simulate this system by taking the output of the controller as the plant input.

The controller output $m(t)$ is related to its input, the error signal $e(t)$, according to the type of controller. Hence, in simulating this system a part of the simulation reproduces the error signal, another part reproduces the controller output and the rest of the simulation program reproduces the control system to obtain a certain result corresponding to a part of an experiment. The general error signal (controller input) is described by (4.16).

$$E(s) = R(s) - H\;Y(s) \tag{4.16}$$

4.5. Analysis of ADVISOR1, ADVISOR2 and ADVISOR3 Expert Systems Structure

The three expert systems share the same structure because they are built to express information of the same domain and have the same knowledge representation scheme. Each expert system was built in a way to mimic the laboratory hand-out sheet. This proved to be consistent with the procedure of each experiment and agreed with the logical sequence of questions asked in the hand-out sheet. At the beginning of a consultation session, the end user is prompted with a menu that outlines the experiment parts as separate choices. The end user inputs data related to the part of the experiment in which a problem is faced in the form of a choice. Once the first choice is made, subsequent choices are decided upon until the source of the problem is detected, then the end user is prompted with an advice that can correct the problem. This approach has two advantages. Firstly, it saves considerable time in detecting the source of the problem which is critical in this particular application of expert systems the student must complete the experiment in an allocated period of time. Secondly, this approach preserves the naturalness of the knowledge domain by optimizing the use of the frame-based conceptual graph knowledge representation technique. This optimization was possible because the knowledge domain admits taxonomic relationships which is a logical consequence of its procedural nature. These expert systems invoke external programs and use function-key control files to complement their expressive abilities. All these programs are integrated in the expert systems structure as end nodes or or results. The knowledge domain does not involve any compound decisions. Compound decisions are reached after a sequence of deductions and conclusions that can be used as separate results. Hence, backward chaining is not required as an inference strategy to simulate the instructor's decision making process. Forward chaining was applied as an inference strategy since an instructor analyses each experiment part under questioning separately to advice on a result. The knowledge bases of the three expert systems are different where each portrays a different experiment. The general knowledge representation structure is shown in Fig. 4.4

The complete set of facts and rules including the heuristics employed in these expert systems, their decision trees, decisions or advice given, and listings of the

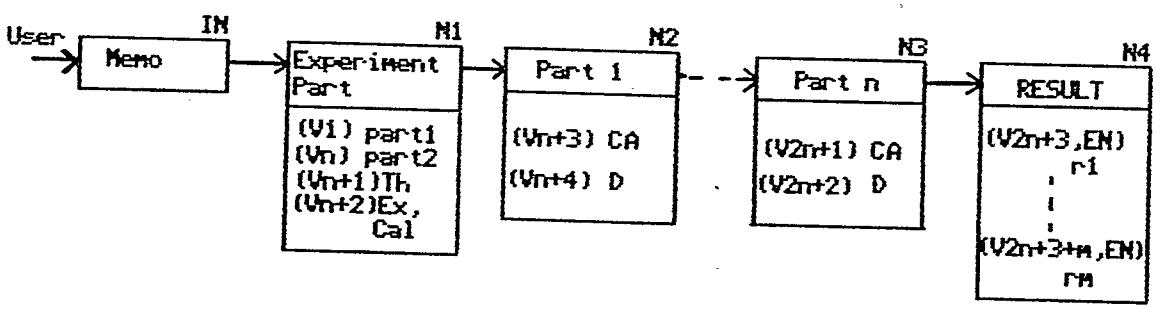


Figure 4.4 The laboratory instruction expert system general knowledge representation structure diagram

external programs and files is documented and available in [23].

4.5.1. External programs and files usage

External programs and files were used to complement the expert systems since the 1st-Class shell lacks certain expressive abilities. The ENHANCED DESIRE [16] continuous simulation language was used to write programs that determine the system response, while Turbo-Pascal was used to write the program RELAY.PAS that performs system analysis and displays the system response in the time domain and gives the phase-plane plot. To make the expert system user-friendly, SUPERKEY [17] and simple MS-DOS batch files were employed. The SUPERKEY package enables programming the keyboard by writing files that can control any key to execute several MS-DOS commands. This package was used because 1st-Class inference engine links to the ENHANCED DESIRE environment through a certain integer parameter but does not invoke the program required to display the system response. In order to run a program in the ENHANCED DESIRE environment, a series of special command lines must be typed in. The end-user is not expected to be familiar with this programming language, and consequently, the SUPERKEY package was used to write files that enable the end-user to run the simulation program with a simple sequence of pressing the <F3> and <F4> function keys without having to type in any command lines. These files are with a ".MAC" file extension name. One of the requirements of the SUPERKEY package is that it must be invoked prior to working in any other programming environment. Hence, the three batch files ADVISOR1.BAT, ADVISOR2.BAT and

ADVISOR3.BAT were written to invoke KEY.COM, the execution file of SUPER-KEY, and then to invoke the required knowledge base.

4.6. The ADVISOR1 Expert System Knowledge Base

EN541EX1.KBM is the knowledge base used by ADVISOR1. This knowledge base was built using the 1st-Class shell. It includes all the necessary knowledge for the laboratory instruction of the relay control systems experiment (first experiment) in control for the ENEL 541 course.

The objective of the first experiment in control is to study the behaviour of some common nonlinear control systems using a relay as a controller. The experiment introduces several relay characteristics which the student is asked to simulate. The student then connects the experimental set-up of the relay control system as shown in the laboratory hand-out sheet, and is asked to reproduce specific results of the system response and document any observations for three types of system configurations. The first type is the open-loop system, and the student is asked to adjust the circuit to achieve this system configuration and measure the deadband width for an ideal relay characteristic. The second type is the closed-loop system where the student is asked to connect the circuit and sketch the observations made for a 15 volt square wave test signal. The student is also asked to record the observations for the effects of inertia, damping, deadband, hysteresis, ramp input and backlash on the system response. The third and last type is the speed control system. In this part of the experiment the student is asked to connect the circuit and observe the system response for an ideal relay characteristic, hysteresis, and the effects of various load settings using magnetic loading discs. The laboratory hand-out sheet, which can be referred to in [23], describes all these parts in sequence and in detail, shows all the required connection diagrams and shows the different relay characteristics used in the control system.

EN541EX1.KBM follows the general structure of the laboratory instruction expert system analysed earlier. The first node represents the experiment parts and offers different facilities to complement the instruction process. The values of this node represent all these parts and facilities in the form of choices. Each of the rest of the nodes represents a part of the experiments with values that indicate the possible problem group that this part may involve. For reasons of clarity and to avoid the "combinational explosion" of factors and values effect, EN541EX1.KBM is forward chained to three other knowledge bases that represent the three major parts of the first experiment. The PROC3.KBM represents the open-loop system part of the experiments, the PROC4.KBM represents the closed-loop system part, and the PROC5.KBM represents the speed-control part. These three knowledge bases were built using the 1st-Class shell and have the same general structure of EN541EX1.KBM. The results of EN541EX1.KBM and the three knowledge bases are edited using the 1st-Class shell editing facility. Whenever a display of the system response is required, the knowledge base is linked to ENHANCED DESIRE environment where an external program written using this simulation language is invoked. The results of EN541EX1.KBM include the RELAY.COM* external program through which the parameters of the transfer function can be varied to obtain different system time domain responses and phase plane plots according to a chosen relay characteristic and system reference input. The source code of this program is RELAY.PAS and it was written in Turbo-Pascal. This program is invoked through the knowledge base directly upon the end user's choice.

The ADVISOR1 expert system was built to emulate expertise in a knowledge domain mode deterministic through a set of assumptions and constraints. The number of all factors and results is manageable and the procedural sequence of questions that can optimize a consultation session is known and was realized by the order of factors on the definitions screens while building a knowledge base. Hence, a complete set of examples was constructed and the optimization method was used to build the decision tree or knowledge base rule. Figures 4.5 through 4.8 show the knowledge representation and operation structure diagram of EN541EX1.KBM, PROC3.KBM, PROC4.KBM and PROC5.KBM knowledge bases respectively.

4.7. The ADVISOR2 Expert System Knowledge Base

EN541EX2.KBM is the knowledge base used by ADVISOR2. This knowledge base was built using the 1st-Class shell, and includes all the necessary knowledge for the laboratory instruction of the phase plane control experiment

*RELAY.PAS was originally written by Mr. B. Boutin as a part of a term report for ENEL 541 which was instructed by Dr. M.H. Hamza at the University of Calgary in 1986, and was modified by the author of this thesis.

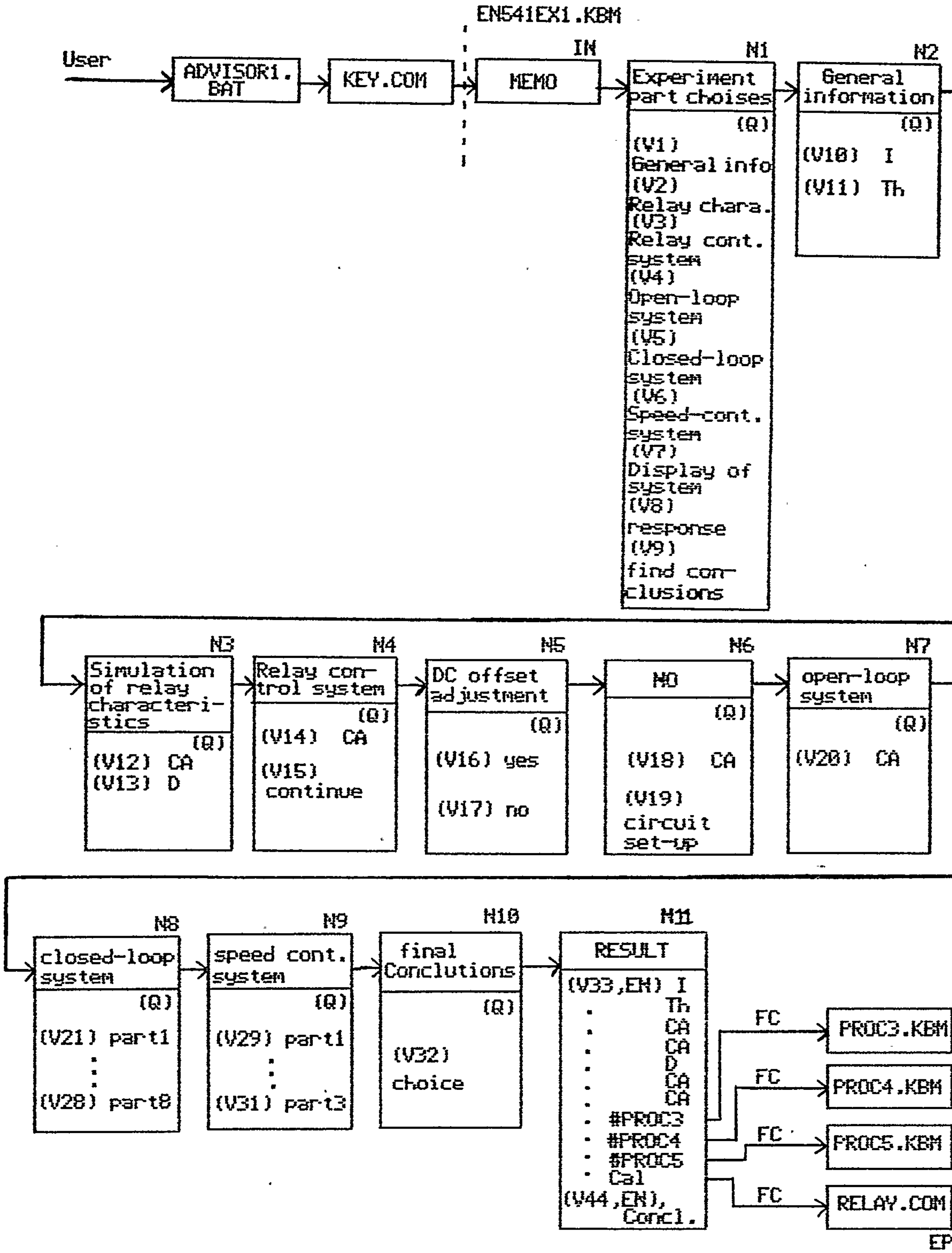


Figure 4.5 EN541EX1.KBM knowledge representation and operation structure diagram.

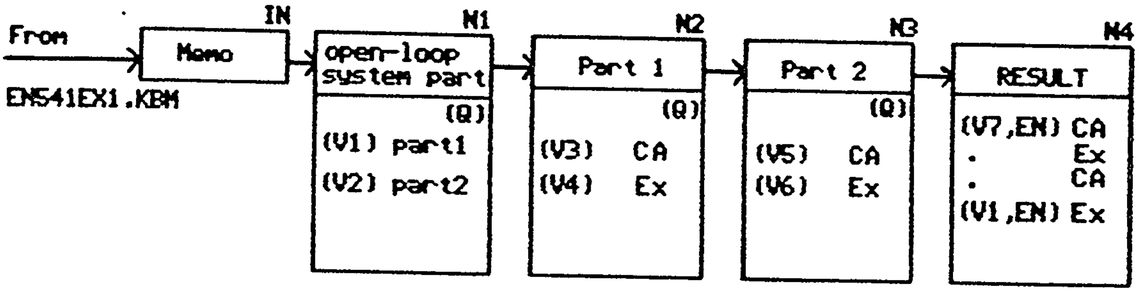


Figure 4.6 PROC3.KBM knowledge representation and operation structure diagram.

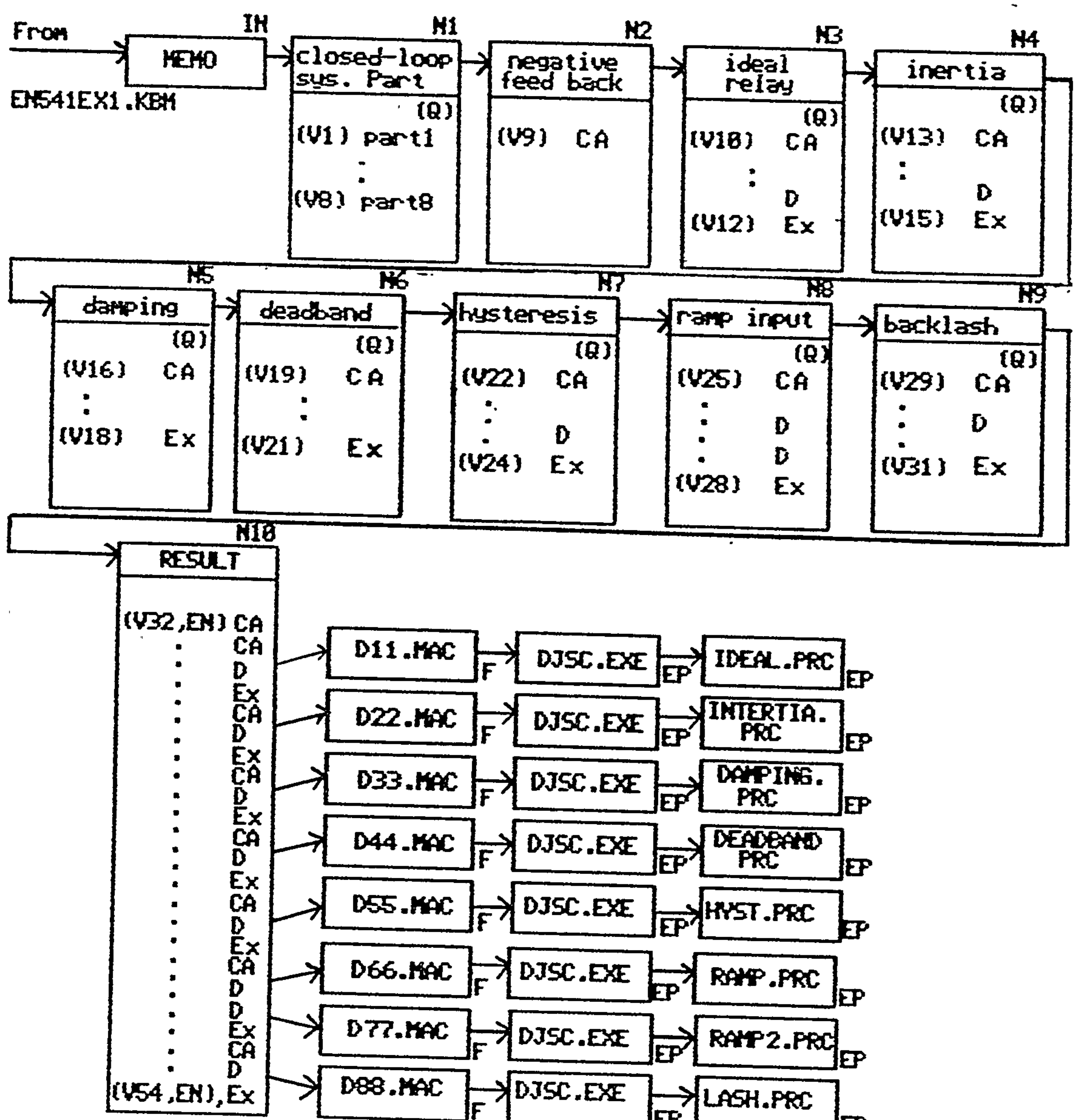


Figure 4.7 PROC4.KBM knowledge representation and operation structure diagram.

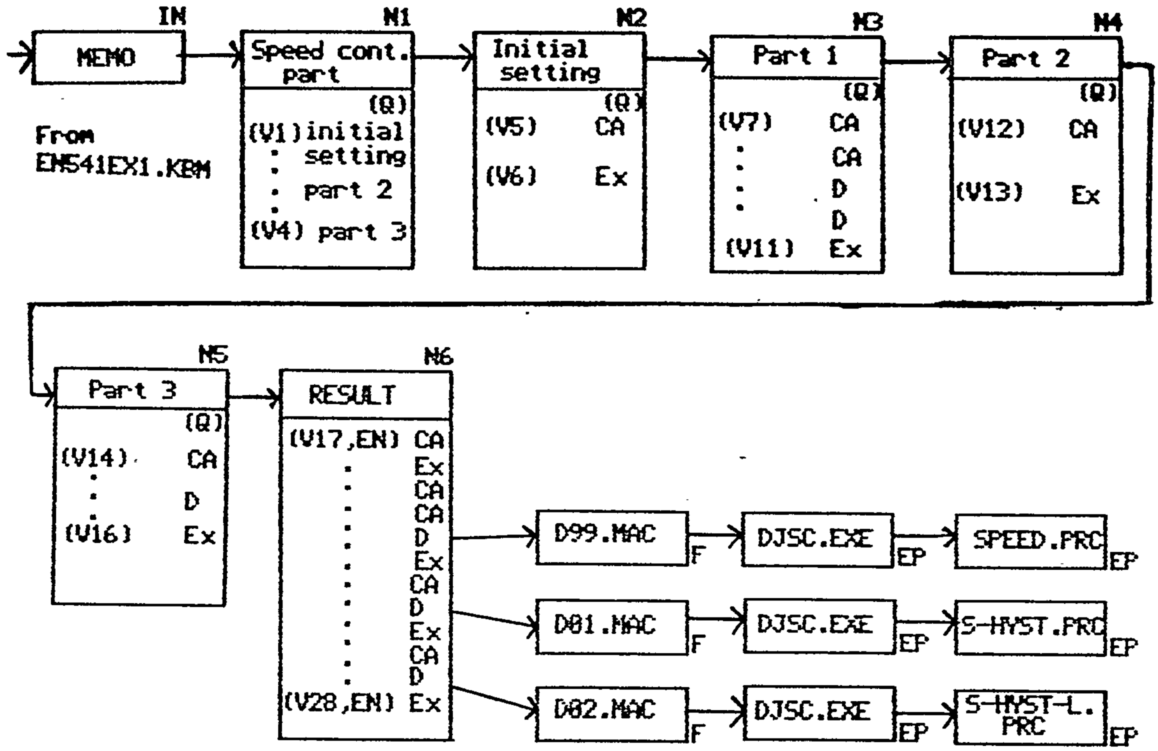


Figure 4.8 PROC5.KBM knowledge representation and operation structure diagram.

(second experiment) for the ENEL 541 course.

The objective of the second experiment is to introduce the phase-plane method of analysis and design. The experimental set-up is basically the same as that in the first experiment using a relay as a controller. The student performs all the circuit connections as shown in the laboratory hand-out sheet. The experiment has two major configurations or parts, the relay control system configuration and the velocity feedback system configuration. Each configuration has five sub-parts.

The student is asked to reproduce, observe record and plot results made in each part for different relay characteristics. The details of the experiment including the connection diagrams are described in detail in the laboratory hand-out sheet which can be referred to in [23].

EN541EX2.KBM follows the general structure of the laboratory instruction expert system analysed earlier, and has the same knowledge representation structure as the EN541EX1.KBM knowledge base. The size and type of information related to the second experiment simplified the process of building the EN541EX2.KBM knowledge base. All this information was manageable and no chaining was required. The results or advice represented in this knowledge base includes general comments, conclusions, theory review, circuit adjustments and displays of the system response using simulation programs written in ENHANCED DESIRE language.

EN541EX2.KBM was built with a complete set of examples since all possible combination of factors and values and their advices are known. Hence, the optimization method was used to build the decision tree or rule. Figure 4.9 shows the knowledge representation and operation structure diagram of EN541EX2.KBM.

4.8. The ADVISOR3 Expert System Knowledge Base

EN541EX3.KBM is the knowledge base of the ADVISOR 3 expert system. It was built using the 1st-Class shell, and includes all the necessary knowledge for the laboratory instruction of the optimum control using a microcomputer control

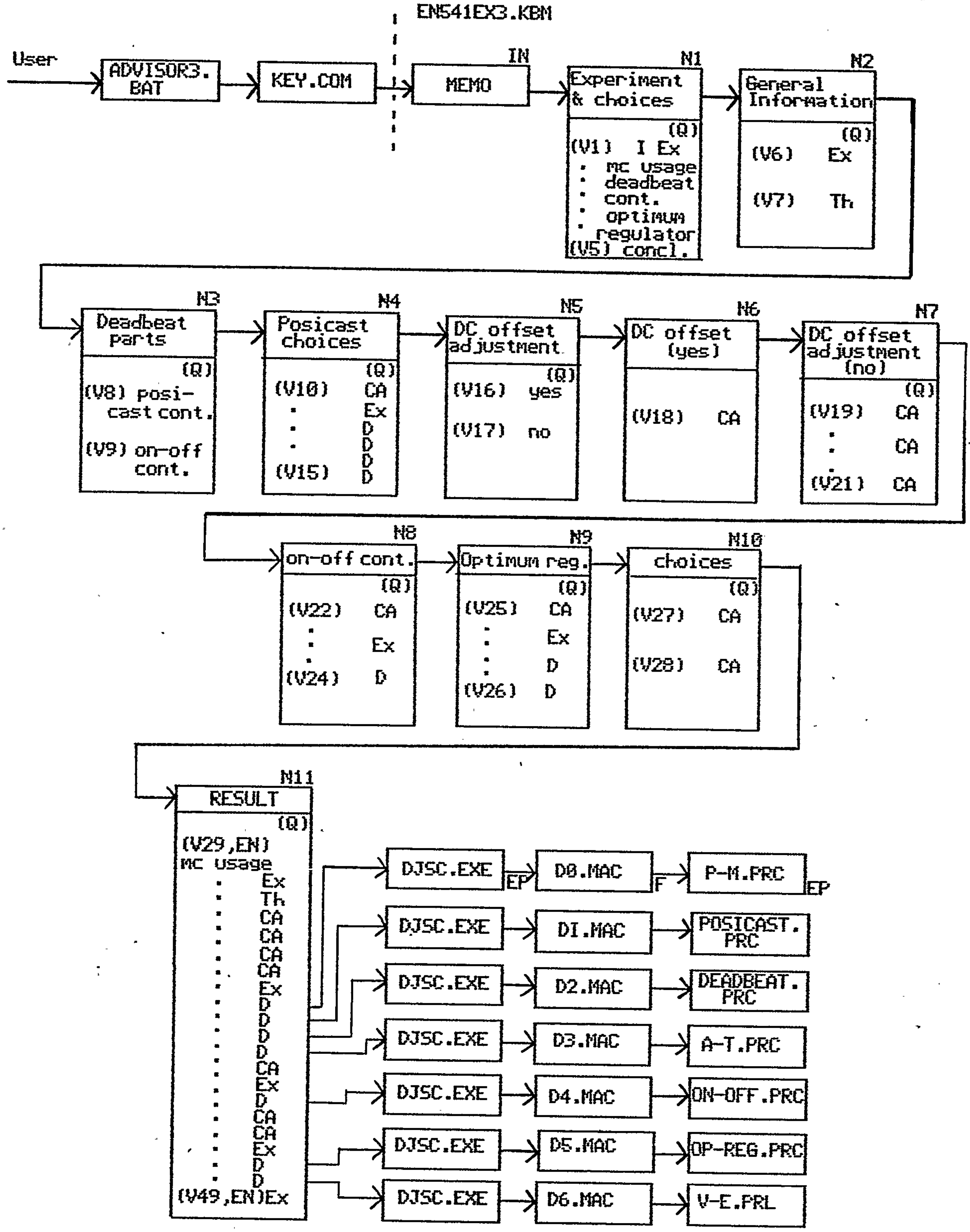


Figure 4.9 EN541EX2.KBM knowledge representation and operation structure diagram

experiment (third experiment) for the ENEL 541 course.

The objective of this experiment is to introduce optimum control using of a microcomputer as a controller. The system to be controlled is basically the same as the previous experiments, and is a position control system. The student performs all the circuit connections as shown in the laboratory hand-out sheet, and must learn how to use the microcomputer during the laboratory session. The experiment is composed of two major parts: the deadbeat control part and the optimum regulator part. In the deadbeat control part, the student is asked to achieve a deadbeat response through posicast control, to record that response and observe the effects of varying the input wave parameters on the system response. The second sub-part is to use on-off control to achieve deadbeat response. The student concludes the deadbeat control part of the experiment by describing the advantages, disadvantages and possible applications of these methods of control. In the second major part of the experiment, the microcomputer is used as a regulator. The purpose is to employ bang-bang control to bring the system to rest in an optimal minimum time for a square wave test signal considered as a disturbance using the microcomputer as a regulator. The student is asked to achieve this response and describes the possible advantages, disadvantages and possible applications of the optimal regulator method of control. The details of this experiment including the circuit diagrams are described in detail in the laboratory hand-out sheet which can be referred to in [23].

EN541EX3.KBM follows the general structure of the laboratory instruction expert system analysed earlier, and has the same knowledge representation structure as the previous knowledge bases. This knowledge base is not chained to any other knowledge base because all the information related to the laboratory instruction of this experiment could be included in one compact knowledge base. The results or the advice represented in EN541EX3.KBM includes exploration of how to use the microcomputer, example values for microcomputer settings to achieve the required response, display of the response, theoretical review, and general remarks and comments. All the simulation programs were written in the ENHANCED DESIRE simulation language.

The optimization method was used to build the EN541EX3.KBM decision tree or rule since a complete set of examples and an optimal sequence for the expert system questions are known. Figure 4.10 shows the laboratory representation and operation structure diagram of EN541EX3.KBM.

4.9. ADVISOR1, ADVISOR2 and ADVISOR3 Testing

The three expert systems were run and checked for errors and mistakes. The laboratory instructors supervised this inspection and the mistakes and errors in their knowledge bases were detected and corrected. At this stage, the expert systems were ready to be tested during actual laboratory sessions. The method of testing involved monitoring students consulting one of the expert systems and recording the time taken to complete an experiment successfully through consulting the

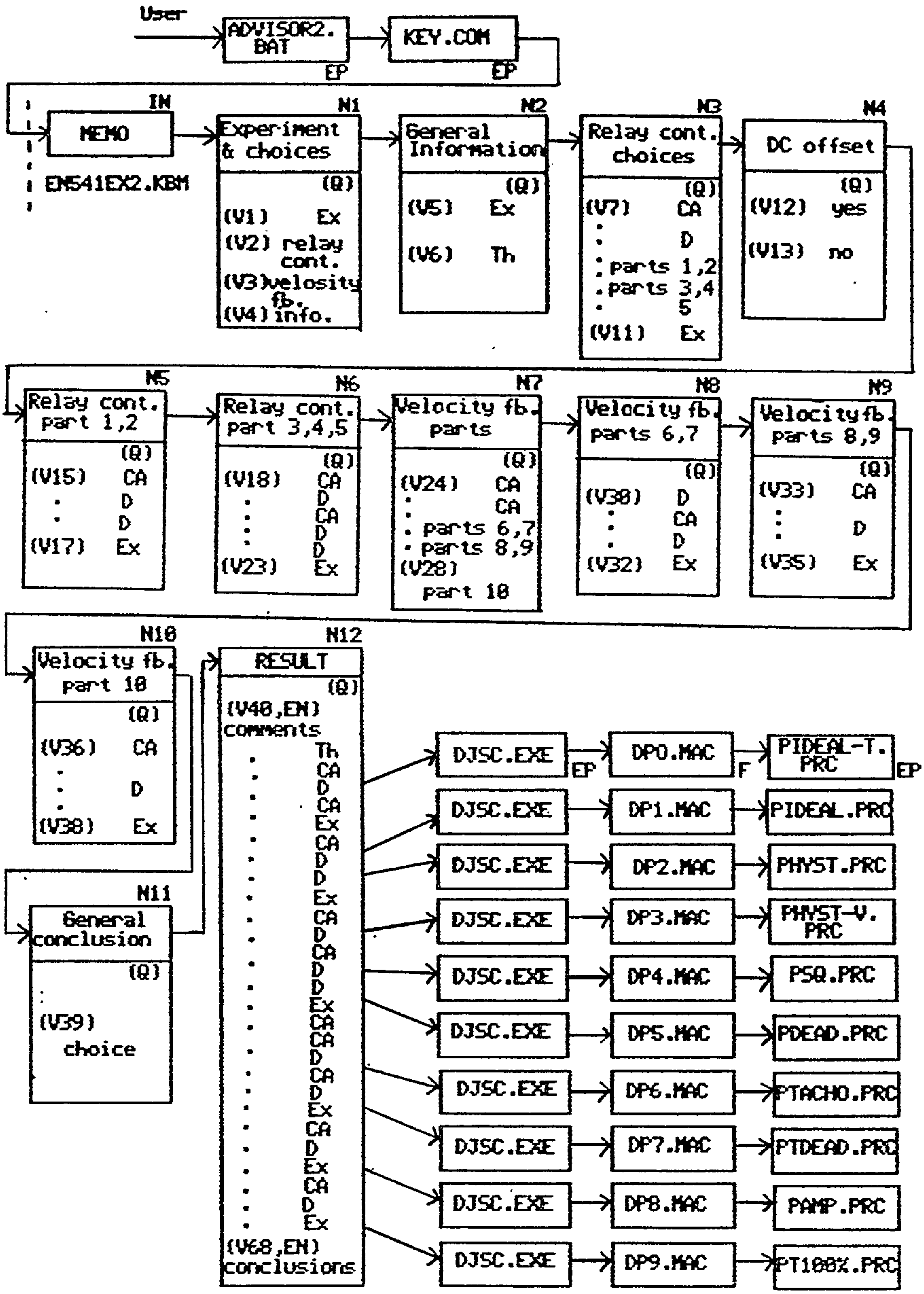


Figure 4.10 EN541EX3.KBM knowledge representation and operation structure diagram

expert system. The frequency of consultation, feedback from the students about the systems weaknesses and strengths, and the time of consultation taken to obtain an advice per student were considered. Two students with no previous experience in performing experiments in control but with the proper theoretical background conducted the first experiments in control by consulting ADVISOR1. The average time for completing the experiment successfully was 191 minutes where the average time taken to complete the experiment in an ordinary laboratory session with the instructors assistance is 150 minutes. ADVISOR2 and ADVISOR3 were each tested in an actual laboratory session. The students consulted the expert systems and could solve the problems they faced without referring to the laboratory instructor and could complete the experiments successfully. All the laboratory groups consulted the expert systems two to three times on the average during a laboratory session. The time of one consultation per student depended on the problem faced but was not less than four minutes. The testing of the expert systems was done under the instructor's supervision, and in many instances without complying to the set of assumptions presumed for the successful utilization of the three expert systems. More testing must be performed prior to the official employment of these systems, but with the test results and error free status, these systems can be considered operational.

4.10. ADVISOR1, ADVISOR2 and ADVISOR3 Evaluation

Evaluating a system that simulates realistic domains which lack the structure of formal systems such as mathematics and logic is a difficult process as it is not possible to prove that an expert system is actually an expert in a rational sense [9]. Hence, these systems were evaluated using practical criteria which involve proving that they are expert systems, assessing the degree to which the research goals were achieved, and outlining the weaknesses and strengths of these systems in relation to the evaluation criteria discussed in Chapter 2.

(1) Are ADVISOR1, ADVISOR2 and ADVISOR3 expert systems?

Each of these systems has a knowledge base, external programs and files and an inference engine that establishes the interaction with an end user. Their output is typically an advice, and are rule-based. The knowledge domain that these systems tackle is narrow and admits heuristics that can be obtained from experts only. These systems do not offer the line of reasoning behind an advice because of the nature of the knowledge domain. Consequently, ADVISOR1, ADVISOR2 and ADVISOR3 can be considered as expert systems.

(2) Degree of research goal achievement

These systems were capable of assisting the laboratory instructor by decreasing the work load during laboratory sessions. The structure of the systems has the potential generalization to be applied in different laboratory environments, but more research is required to investigate this potential.

(3) The three expert systems strong points

The knowledge embodied in these systems is correct and valid within the domain of their application, and their knowledge representation structure captures the natural expression of the knowledge domain. They are user-friendly and easy to use and offer the end user a theoretical backup that assists in comprehending any of the experiments. Expandability of any system is possible through the chaining strategies, which will not have dramatic effects on the memory requirements of the system since the inference engine is separate from the knowledge base. The three expert systems have obvious educational and practical benefits which asserts the feasibility of their application.

(4) The three expert system weaknesses

Even though these systems are easy to use, they may crash if the end user does not follow the instructions displayed exactly. The end user has to obtain a print-out of the advice in order to be able to apply it. Hence, the system seems to be ineffective without a printer.

A comprehensive evaluation of these systems can be obtained once they are tested in several laboratory sessions, and have gained the total acceptance of the students and instructors. At this stage of development, these expert systems have demonstrated a satisfactory level of performance upon employment, and do have a potential for further development.

CHAPTER 5

MARPLAN: THE MARKETING PLAN DEVELOPMENT EXPERT SYSTEM

5.0. Introduction

MARPLAN is an expert system that enables a manufacturer or an end user with certain characteristics, to develop a marketing plan to launch an imitative, durable product used in households and of the unsought product class. An imitative product is a product that faces potential competitors once launched.

Prior to the analysis of MARPLAN, four basic terms must be introduced and briefly analysed. These terms are: the market, product, marketing management, and marketing plan.

A market is considered to be all the potential customers sharing a particular need or want, who might be willing and able to engage in exchange to satisfy that need or want. A market intended for the exchange to notify a particular need or want is referred to as a target market [24].

A product is anything that can be offered to a market for attention, acquisition, use, or consumption that might satisfy a want or need. It includes physical objects, services, persons, places, organizations, and ideas. Products can be grouped into three groups: nondurable goods, durable goods and services. Nondurable goods are tangible goods that are consumed in one or a few uses such as soap, beer and salt. Durable goods are tangible goods that survive many uses such as clothing and machine tools. Services are activities, benefits or satisfaction that are offered for role such as haircuts or legal counseling. Products are classified into consumer and industrial goods. Consumer-goods are of importance to this work, and they can be convenience, shopping, speciality and unsought goods. Convenience goods are goods that are purchased frequently, immediately, and with a minimum effort in comparison and buying, such as tobacco products and newspapers. Shopping goods are those which, in the process of being selected and purchased, are characteristically compared on such bases as suitability, quality, price and style, such as furniture and household appliances. Specialty goods are unique goods that are purchased after exerting a considerable buying effort such as quality foods and cars. Unsought goods are goods that a consumer does not know about or knows about, but does not normally think of buying, such as smoke detectors and life insurance [24].

Marketing management is the analysis, planning, implementation and control of programs designed to create, build, and maintain beneficial exchanges and relationships with target markets for the purpose of achieving organizational objectives [25].

A marketing plan is a plan for selling a particular product or service in a particular industrial or geographical market [24, 25]. The contents of this plan vary according to the type of product sold and the company structure which in turn depends on the company business planning stage. A typical marketing plan usually has eight different sections [24].

- I. Executive Summary. This presents a summary of the plan for quick skimming by the management.

- II. Current Marketing Situation. This presents relevant data on the market, product, competition, distribution and marketing environment.

- III. Opportunity and Issue Analysis. This outlines the main opportunities, strengths and weaknesses of the product.

- IV. Objectives. This defines the objectives of the plan in areas of sales volume, market share and profit.

- V. Marketing Strategies. This presents the approach adopted to meet the plan objectives.

- VI. Action Programs. This details the approaches adopted to meet the objectives of the plan.

- VII. Projected Profit-and-Loss Statement. This specifies the loss or gain expected from executing the plan.

- VIII. Controls. This summarizes the methods of monitoring the plan.

These eight components vary in precedence and may mutually combine altering the structure of the plan. The basic elements of a marketing plan are either controllable or uncontrollable as mentioned earlier. These elements are extracted from the functions of each component of the typical marketing plan, and these basic elements, namely: the product, potential consumers or target market, potential competition, marketing environment, capabilities of the producer, price, promotion and distribution, comprise the fields of development of the plan proposed by MARPLAN. A typical marketing plan is among a group of plans that companies usually develop such as corporate, divisional, brand and functional plans where each has a considerable marketing component [24]. The need for a marketing plan stems from the fact that contemporary markets are sophisticated and highly competitive and in order to sell a product successfully in such markets, a marketer must adapt to modern marketing strategies [26]. Marketing strategies vary according to the product group, class and life cycle which may be in the introductory, growth, maturity and decline stages.

Decision making in marketing management involves a considerable degree of uncertainty and requires an updated level of expertise that has adapted continuously to the change in marketing trends and means. Developing a marketing plan to sell a product or a service is a field in which market research data, empirical knowledge of the market needs and expert interpretation are the basic interacting elements that lead to the expert advice, or the marketing plan. Hence, the application of expert systems in the marketing management field is promising due to the ability of these systems to handle uncertainty and to represent various complex knowledge domains. The feasibility of this application is realized in the long run if a system is developed that can output an effective marketing plan that can rival a plan achieved by an expert at a cheaper expense and easier availability. The feasibility of this application is realized in the short run by a framework for a marketing plan development expert system for a specific type and class of products. This framework can serve as a model suitable for further expansion and build-up of a system that is capable of yielding a comprehensive marketing plan for any product. The complexity and variety of the marketing management knowledge domain makes this long term feasibility prospect an ambitious expectation. The realistic objective of this work was to build a system capable of advicing on a marketing plan up to a limited degree and within a set of constraints.

As in the three control laboratory expert systems, the design and build-up procedures presented in Chapter 2 were followed, but it must be noted that the details and steps constituting these procedures were formalized after the completion of the systems, including the system which is the subject of discussion of this chapter namely the Marketing Plan Development expert system (MARPLAN).

5.1. The Knowledge Domain, Constraints, Assumptions, and System Requirements of MARPLAN

Developing a marketing plan from its basic elements is a process that involves extracting data for each element separately. This data is processed, and decisions about an element are derived in relation to decisions derived about the other elements in a manner that produces a plan which specifies ways of selling the product in order to meet predesigned objectives. The data is collected by screening the present conditions of each element and relevant characteristics. For the product, the marketer examines the class, type, packaging levels, characteristics and various aspects related to the product. To determine the target market, questions like who like who would buy the product, what are the possible segments of the market, what is the buying behaviour and habits of the consumer, etc., must be answered. The capabilities of the producer are related to the producer's strengths and weaknesses with respect to the competitors, funding, distribution channels, contacts, etc. The marketing environment is the set of laws and present situations that govern transactions and dealings in the market. The products, strengths and weaknesses, prices, marketing strategies and important information about potential competitors are investigated. Different pricing methods are weighed, and the marketer must develop offers and possible pricing strategies depending on several variables such as distribution channels and methods of selling. The product distribution specifies the selling location and the possible optimal distribution levels, or even includes alternatives such as commissioning a second party to distribute the product. To promote the product, the marketer must select an advertising mix that can deliver a message capable of achieving a promotional objective such as increasing the awareness about the product or just introducing it to the target market. The advertising mix can include radio, television, mailing brochures, etc.

Once all the data is collected, the marketer weighs the results and implements common sense rules of thumb and acknowledged marketing decision making processes to develop a marketing strategy and to set a plan. In a knowledge base, data are considered as facts, and the relationships between these facts that yield decisions are the rules. These facts and rules are discussed in detail in the section about MARPLAN knowledge base.

Expert marketers have developed strategies that correspond to the different stages in a product life cycle. MARPLAN develops a marketing plan to launch the product, and consequently, it focused on the marketing strategies in the introduction stage of a product. If the marketers consider only price and promotion as marketing variables, then one of four strategies can be pursued in the introduction stages shown in Fig. 5.1.

In a rapid-skimming strategy, the new product is launched at a high price and a high promotional level. This strategy is applied if a large part of the target market is unaware of the product, and those who are aware are eager and capable of buying the product, and there is potential competition which makes building a brand preference and important objective of the strategy. A slow-skimming strategy consists of launching the product at a high price and low promotion. This strategy is applied if the market is limited in size, aware of the product, willing to buy it at a high price and the potential competition is weak. A rapid-penetration strategy consists of launching the product at a low price and heavy promotion. This strategy is applied if the market is large, unaware of the product, price sensitive, and strong potential competition exists. A slow-penetration strategy consists of launching the new product at a low price and low level of promotion. This strategy is applied if the market is large, highly aware of the product, price sensitive and potential competition exists [24].

		Promotion: High	Promotion: Low
Price	High	Rapid-skimming Strategy	Slow-skimming Strategy
Price	Low	Rapid-peretration Strategy	Rapid-peretration Strategy

Figure 5.1 Four Introductory Marketing strategies [24].

In MARPLAN, the marketing strategy is selected after processing incoming data from all the basic elements constituting the marketing plan.

A large number of marketing variables exists which makes the knowledge domain very difficult to simulate. A set of constraints, assumptions and system requirements was formalized to limit the knowledge domain into a manageable size without sacrificing the essential reasoning requirements necessary for assisting the end user in developing a sound marketing plan. The limitations imposed by the capabilities of the 1st-Class shell, contribute to this set of constraints, assumptions and system requirements as explained earlier.

The constraints are related to the output of the system. These constraints are:

1. all the estimates projected by the advisor are valid for a short duration of one year, which is considered to be the required launching period,

2. the decisions taken by the advisor are subject to the data input provided by the user, which means that the advisor reflects the degree of reasoning in the input data, and

3. the advisor provides a reasonable evaluation of the market demand estimates for the initial launch disregarding any relationship to the following stages of the product life cycle.

MARPLAN was developed under the major assumptions that

1. the end user, who may be a manufacturer, an entrepreneur or a general user, is unexperienced with marketing plans, but has a basic understanding of its major elements,

2. the end user utilizes individual efforts to market the product and not through an organized commercial establishment,

3. the target market includes individuals and households that buy goods and services for personal consumption, and is in Canada,

4. the end user possesses data needed by the advisor about the market, and

5. the end user is developing a marketing plan for a durable product of the unsought type and used in households.

The system hardware requirements were discussed earlier. The software requirements are the "ADVISOR.EXE" execution file which is the inference engine provided by 1st-Class shell, the knowledge bases, MS-DOS operating system, the Turbo-Pascal version 3.0 system implemented for the PC-DOS, MS-DOS, CP/M-86 and CP/M-80 operating systems to handle the external programs and external text and batch files to execute the print-out process.

5.2. MARPLAN Design Procedure

The MARPLAN design presented in this section is the final outcome of the interaction process between the design and build-up procedures. The continuous adjustments imposed on the design objectives are due to the limits realized while building the MARPLAN expert system. These limits are actually the set of constraints, assumptions and system requirements forced by the nature of the knowledge domain and the capabilities of the expert shell and outlined in the preceeding section. The design procedure is as follows.

(1) The expert system.

Marketing plan development in marketing management is the general knowledge domain of this expert system. MARPLAN is an expert system that enables a manufacturer or a general user to develop a marketing plan to launch an imitative product of the durable class and used in households within a period of one year. The objective of this plan is to set a strategy that will obtain a market share in a competitive environment. This expert system provides a "go" or "no-go" decision depending on input data provided by the user about the uncontrollable factors in any general marketing plan mainly: the consumer, competition, marketing environment and the capabilities of the producer. It also investigates the controllable factors of any general marketing plan mainly: the product, price, promotion and distribution. MARPLAN outputs a marketing plan that includes demand estimates, loss-gain projections, pricing, distribution and promotion decision guidelines. The operating environment of this expert system consists of an IBM PC-AT with MS-DOS operating system or any compatible personal computer and IBM compatible graphics printer. The memory requirements to operate MARPLAN do not exceed 640K depending on the size of the files generated by the expert system during an advisory session.

(2) The expert knowledge sources.

In the field of marketing management, the possible knowledge sources include human expertise, printed material related to marketing plan development and various feedback data from marketing research. Human expertise is not easily available or accessible and is expensive in many cases, especially if the consultation or knowledge acquisition process requires time dedication. Marketing research departments in many companies are usually reluctant to provide data about the type of marketing plans implemented for different products and the degree of success of such plans. Hence, human expertise and feedback of marketing research data were not available as knowledge sources. Several text references [24,25,26,27,28] discussing marketing management and marketing plan development, and notes derived from two graduate courses in marketing (Marketing Management I and New Venture Marketing), and various marketing journals [29,30], comprise the basic knowledge sources utilized to build MARPLAN expert system.

(3) The expert shell.

Since the knowledge base is expected to be large, the shell will not be able to accommodate all its facts and rules in one compact knowledge base. The MARPLAN expert systems application is different from the previous applications in the fact that the advices for the separate sections of the marketing plan are related. In addition, computation and graphic displays are needed to express various parts of the plan. Hence, chaining to external programs must be extensively used to perform these functions. All the external programs were written using Turbo-Pascal programming languages. Any program with the ".COM" extension name is the compiled version of a source code program with ".PAS" extension name. The 1st-Class expert shell possesses all the required qualities to build the MARPLAN expert system. A prototype MARPLAN expert system was built. This prototype system had a small knowledge base composed of nodes representing the controllable and uncontrollable factors where each node had several values and only one result was provided which was the marketing plan.

(4) The initial expert system is shown in Figure 5.2.

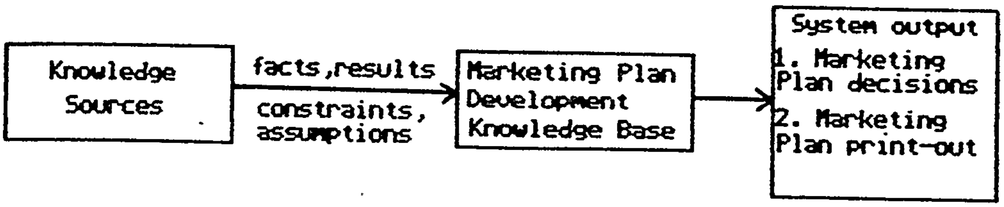


Figure 5.2 MARPLAN expert system descriptive initial model.

5.3. MARPLAN Build-up Procedure

The notes obtained from the two graduate courses in marketing management provided the outline of the marketing plan. These notes outlined the major elements constituting the marketing plan namely: the consumer, competition, marketing environment, producer or manufacturer, product, price, promotion and distribution. The marketing plan usually involves details that determine its final form, but these details vary according to the company and type of market. The emphasis was only on the basic elements constituting the marketing plan due to three major reasons.

(1) The aim of MARPLAN is to develop a specific marketing plan. This plan will consequently include particularities and specifics of a plan especially developed by an entrepreneur with restricted characteristics. Hence, the plan is specific, but also is general within the product category. Specifying the

product category and the marketer's characteristics and abilities reduces the details of the plan. For instance, no marketing controls, executive reports, marketing alternatives or other specific elements are included in the plan.

(2) The capabilities of the expert shell limit the size of the knowledge base, and a general plan that adapts to any situation cannot be built using the available tools.

(3) In order to construct an efficient system, the knowledge domain was probed carefully and a set of assumptions and constraints was adopted to minimize error and uncertainty in the system output. This set is discussed in the knowledge base section.

The rest of the knowledge sources mainly the textbook references, papers and journals, provided the relationships between these elements and the decision making network interconnecting the data obtained from certain elements and the advice or results from the other elements. As an example, the data obtained through interaction with the user about the product potential competitors is processed with other data obtained from other elements to deduce pricing, distribution and promotion decisions. The sequence of elements presented in MARPLAN plays an important role in the ability of the system to output sound advice. The appropriate sequence of these elements considering all the assumptions and constraints of the knowledge domain was obtained from the second set of sources [25,27,29].

Knowing all the elements constituting the marketing plan, their facts, rules that govern the relationships between these elements, and the logical sequence of their occurrence, gives enough leads to partition and refine the knowledge. The elements of the marketing plan developed by MARPLAN are divided into three problem groups or sets. A data input problem set that includes the product, potential consumers or target market, potential competitors and the abilities of the producer. The second set is the input data processing problem set which includes the price, promotion, marketing environment and distribution elements. The third set is the decision or advice problem set that includes pricing, promotion, distribution and "go" or "no-go" decisions, sale and cost estimates and forecasts and product positioning. The knowledge is then refined by eliminating redundant facts and rules, and by tailoring the core of the knowledge base to suit the assumptions and constraints that gradually emerge through the build-up process. Knowledge representation follows the scheme imposed by the expert shell. The frame based conceptual graph technique was used for knowledge representation in all the knowledge bases without any exceptions. The MARPLAN knowledge base is actually composed of several knowledge bases which embody the facts and rules of each element of the marketing plan separately. All these bases are combined to form one single knowledge base in one expert system (MARPLAN) by employing forward and backward chaining strategies.

The knowledge acquired was used to build a knowledge base for each element of the marketing plan. External programs were written to process input data and perform specific decision making functions such as data transfer or graphic product partitioning displays. These programs compensated for the 1st-Class expert shell lack of expressive abilities and ensured the communication between the different knowledge bases to form the reasoning network of MARPLAN.

The final build-up and implementation of MARPLAN was realized by grouping all the knowledge bases and external program in one knowledge base that has a single factor with values leading to results that forward or backward chained to a knowledge base or an external program depending on their sequence in the marketing plan. The system was then tested and evaluated. MARPLAN expert system, its knowledge base, and the testing and evaluation results are discussed in detail in separate sections. Figure 5.3 demonstrates the various levels of the design and build-up procedures and their dynamic relationships.

5.4. Analysis of MARPLAN Expert System Structure

MARPLAN expert system is a control center that organizes the interaction among the knowledge bases that constitute the system as a whole. It invokes a knowledge base or an external program through forward or backward chaining, and introduces the system to the end user. The control mechanism of MARPLAN is simple. Each element of the marketing plan is treated separately. The end user is prompted with a menu that includes all these elements. Once the end user chooses an item on the menu, the system invokes a corresponding knowledge base or external program. The user must answer a series of questions after which a decision is given by MARPLAN, or information is passed to other knowledge bases for processing in order to give further decisions. When the choice of items on the menu

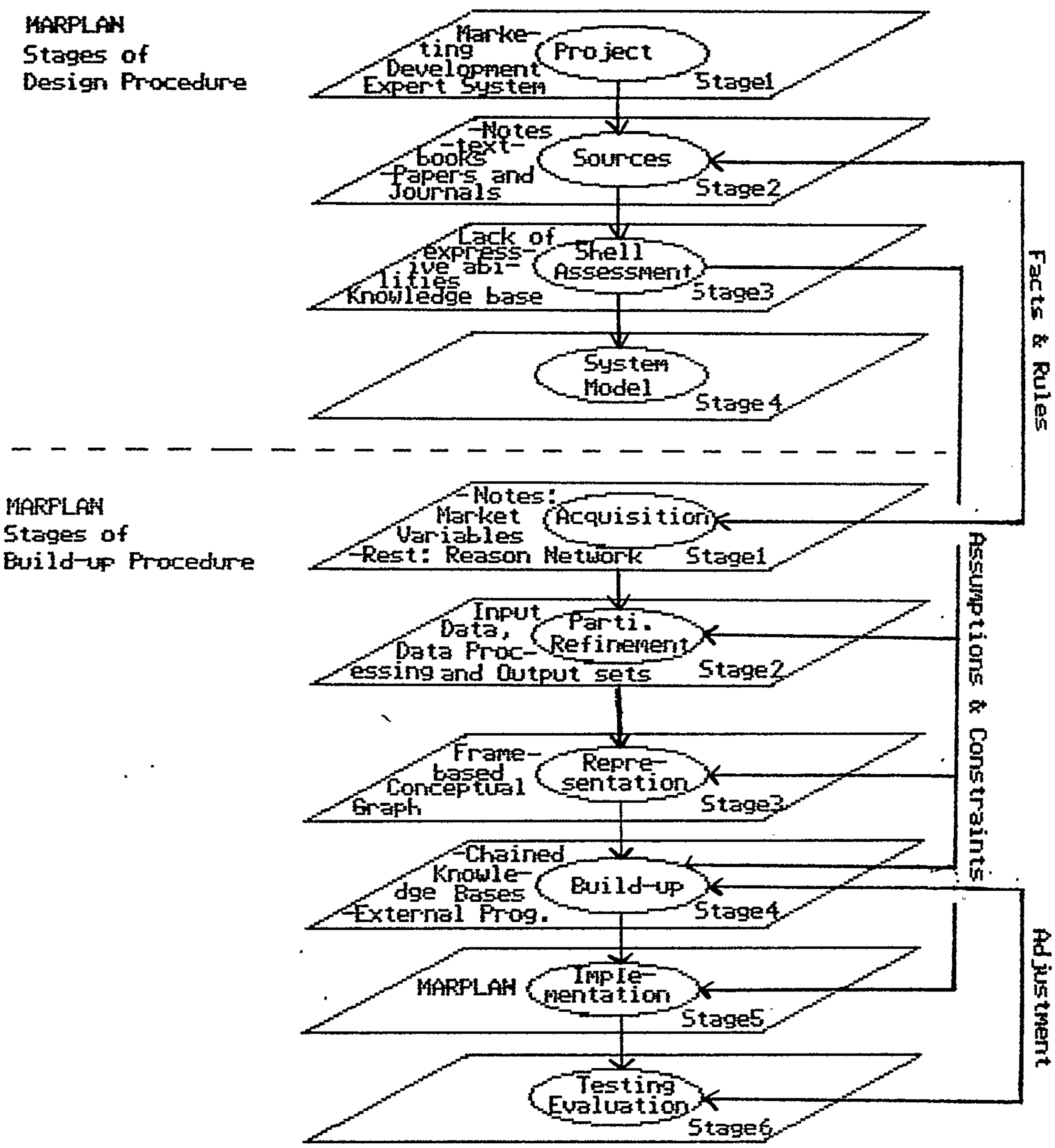


Figure 5.3 MARPLAN stages of design and build-up procedures and their dynamic relationship

is exhausted, all the knowledge bases and external programs are consequently invoked, and the marketing plan is said to be complete. The end user is advised of the best sequence of the item choices on the menu required to achieve best results, but if this sequence is not followed, the system can still output a non-optimal marketing plan depending on the input data. A print-out of the marketing plan can be obtained upon request.

5.4.1. MARPLAN knowledge base structure

The memo, plan part, and RESULT factors constitute the knowledge base. The MEMO factor includes preface text that explains the system and how to operate it to the end user. The plan part factor has 12 values, ten of which represent the different elements of the marketing plan, one for the marketing plan report print-out, and one to exit from the advisor to the operating system. The RESULT factor has 12 values or end nodes each forward or backward chained to a knowledge base or external program that tackles an element of the marketing plan. Since the MEMO factor is inactive, the part plan and RESULT factors are related by 12 examples describing a simple if (value) then (result) production rule. The optimization method was used to create MARPLAN decision tree or rule. This method creates a decision tree that provides the user with complete freedom in choosing any item on a menu as described earlier. The optimal sequence of choices can be enforced on the user by building MARPLAN knowledge base with 12 factors each with a single value representing an element of the marketing plan including the print-out and exit to operating system in addition to the MEMO and

RESULT factors. Then, the matching method should be used to build the decision tree without the need of examples or rules. The advisor will then take the user through the optimal sequence realized by the order of factors on the definitions screen. Enforcing a sequence was avoided to comply with the basic assumption that the end user is not familiar with marketing plans. Giving this user the freedom to choose from the elements has the advantage of educating the user about the plan in order to aid him/her in developing a concept about marketing plans and their development procedure. MARPLAN utilizes eight different marketing bases and two external Turbo-Pascal version 3.0 programs directly through inference. The PRCHARES.KBM and PROD.KBM knowledge bases deal with product characteristics and packaging and labeling decisions. The DEMO-SEG.KBM and GEO-SEG.KBM deal with the target market demographic, socio-economic and geographic segmentation. The PRICE.KBM knowledge base gives pricing decisions, while the DIST.KBM knowledge bases give product distribution and promotion decisions. The CLIM.KBM knowledge base deals with the market legal environments. The POS.COM external program does the market analysis, product positioning and the demand and cost analysis and estimates. The LOSSGAIN.COM external program gives the projected profit or loss and final advisor statement. Figure 5.4 shows the decision tree structure diagram of MARPLAN and the corresponding inference strategies used per end node.

The interaction between the knowledge bases and external programs is complex and constitutes the decision making and reasoning network of MARPLAN and

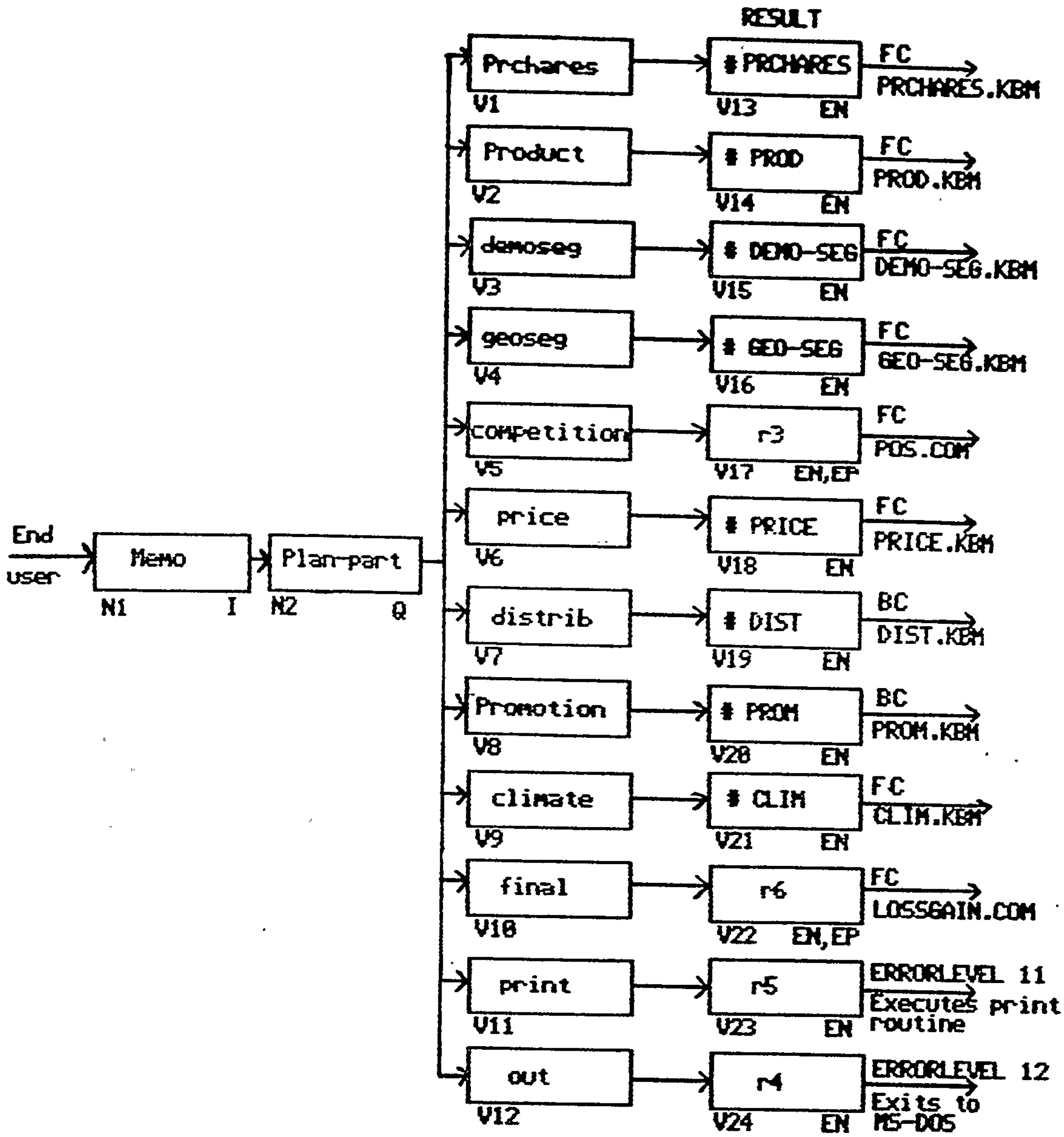


Figure 5.4 MARPLAN Decision Tree Structured Diagram

this is discussed in detail in the following section. In addition, the total number of knowledge bases, external programs and files constituting MARPLAN is 182, and this number becomes 215 with all the files generated after a session to develop an optimal marketing plan. A complete documentation of MARPLAN including the decision tree and listings of external programs and files is available in [23].

5.5. MARPLAN Knowledge Base Components

There are ten knowledge bases that constitute MARPLAN, eight of which were built using 1st-Class shell and two using Turbo-Pascal. In this section, the structure and interaction between the MARPLAN components is investigated. The components to be analyzed are in the same order required for an optimal marketing plan. A complete documentation of all these knowledge bases and listings of external programs and decision trees is available in [23]. It is important to note that the information provided in this section orients in understanding and analysing the documentation available in [23] and complements it.

5.5.1. The PRCHARES.KBM knowledge base

PRCHARES.KBM is a knowledge base built by 1st-Class shell to define simple general characteristics of the product. The first step in developing a marketing plan is to know the product to be marketed and sold. It is important to know its features and attributes. This assists in setting a price for the product, knowing where it stands among its competitors, developing a successful promotional message, realizing who may buy it and why, and in distributing this product.

Contemporary life styles are sophisticated and require many needs, therefore the products available to satisfy these needs are of immense variety. A system developer faces considerable difficulties in building a knowledge base that includes information about the general characteristics of even one class and type of products, using 1st-Class shell. The reason is that each characteristic is dependent on the features of a particular product as well as the buying and usage behaviour of the target market [26]. Consequently, each characteristic becomes a marketing variable with a separate set of descriptions. To represent such a structure using 1st-Class, each characteristic will be a factor with several values. Taking into account the number of these characteristics and values, the system developer will be left with a "combination explosion" of factors and values. The realization of this difficulty led to the constraints and assumptions about the product class and type. Therefore, the initial general characteristics of the product is that it must be durable, of the unsought class and used in households.

To avoid the combinational explosion effect and still give the end user the initial step in developing a marketing plan, PRCHARES.KBM was built in such a way to introduce the importance of the general characteristics of a product contributed by the functional and buying behaviour elements. PRCHARES.KBM is a knowledge base that deals with the product usage. These results orient the end user to make decisions about the other elements of the marketing plan. Hence, this knowledge base is not interactive with the other components of MARPLAN with the exception of the marketing plan print-out. The knowledge base does not give decisions, but transforms a combination of general characteristics into a descriptive comment about the product using the end user's choice. The optimization method was used to build the decision tree of PRCHARES.KBM because a complete set of examples could be obtained from its manageable knowledge base. Figure 5.5 shows PRCHARES.KBM knowledge representation and operation structure diagram.

5.5.2. The PROD.KBM knowledge base

PROD.KBM is a knowledge base built by 1st-Class shell. It complements PRCHARES.KBM by orienting the end user towards deciding on the product brand name, packaging and labeling. The brand name, packaging and labeling stand in the same category as features and characteristics in describing the product. This knowledge base outlines the advantages and disadvantages of giving a brand name, packaging and labeling the product to the end user. Once this information is read and understood, the end user is asked if a brand name is ought to be given for the product. If the product is given a brand name, the end user is asked to decide on the packaging level and the labeling objective within PROD.KBM by choosing from a set that includes all options available for that purpose. If the product is not given a brand name, the knowledge base is forward chained to PRXNAME.KBM.

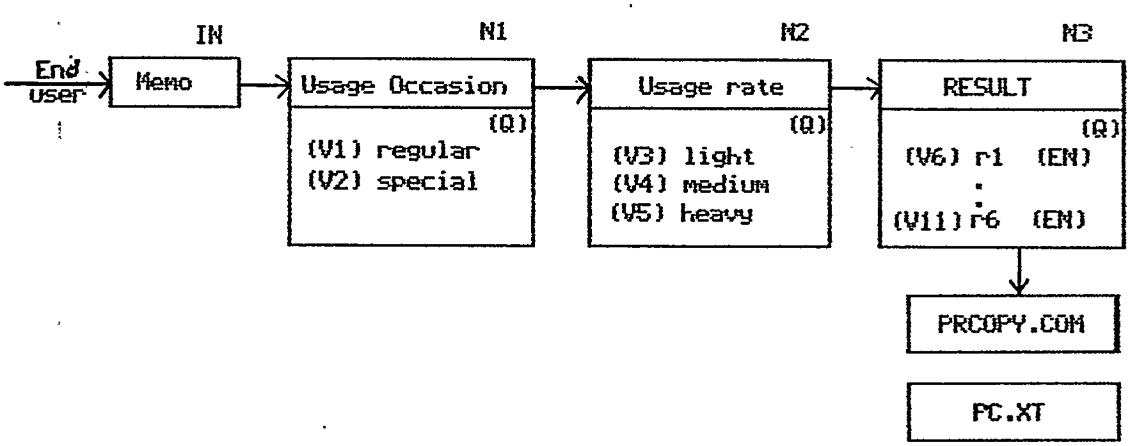


Figure 5.5 PRCHARES.KBM knowledge representation and operation structure diagram

PRXNAME.KBM is a knowledge base built by 1st-Class shell to enable the end user to decide on packaging levels and labeling objectives if the product is not given a brand name. The results or end nodes of the PROD.KBM and its sub-component PRXNAME.KBM contain the brand name, packaging and labeling decisions that the knowledge presented by the advisor has led the end user into taking.

This knowledge base interacts with the other parts of the marketing plan by invoking the external program NAME.COM, that accepts a brand name for the product and sends it POS.COM where it is used as a reference name throughout the marketing plan. If no brand name is assigned to the product, the knowledge base invokes another external program, NONAME.COM, which cancels any existing brand name assigned in previous sessions. A reference name must be assigned to be used throughout the marketing plan and the advisor requests this assignment in POS.COM is case NONAME.COM has been invoked. PROD.KBM interacts also with the print-out component of MARPLAN.

The knowledge embodied in PROD.KBM and PRXNAME.KBM complements the knowledge in PRCHARES.KBM to conclude introducing and investigating the basic marketing variable in a marketing plan which is the product. The information included in these knowledge bases complies with the constraints and assumptions assigned to complete this work. The results obtained defining the product aid in deciding for the other marketing plan elements. This information was included as a part of the plan due to the important roles that the brand name, packaging and labeling decisions play in selling a product [24]. To avoid redundant information, a complete set of examples was included to detect redundancy. The matching method was used in both knowledge bases. Figure 5.6 shows PROD.KBM and PRXNAME.KBM knowledge representation and operation diagram.

5.5.3. The DEMO-SEG.KBM and GEO-SEG.KBM knowledge bases

The second step in developing a marketing plan is to determine the target method. The total market is very large, and cannot be served as a whole especially by smaller producers. In order to determine the target market, the total market is broken into segments that share common properties. This segmentation can be done in many ways [24]. A marketer selects segmentation variables that are most relevant to selling the product and then tries to determine which group of the

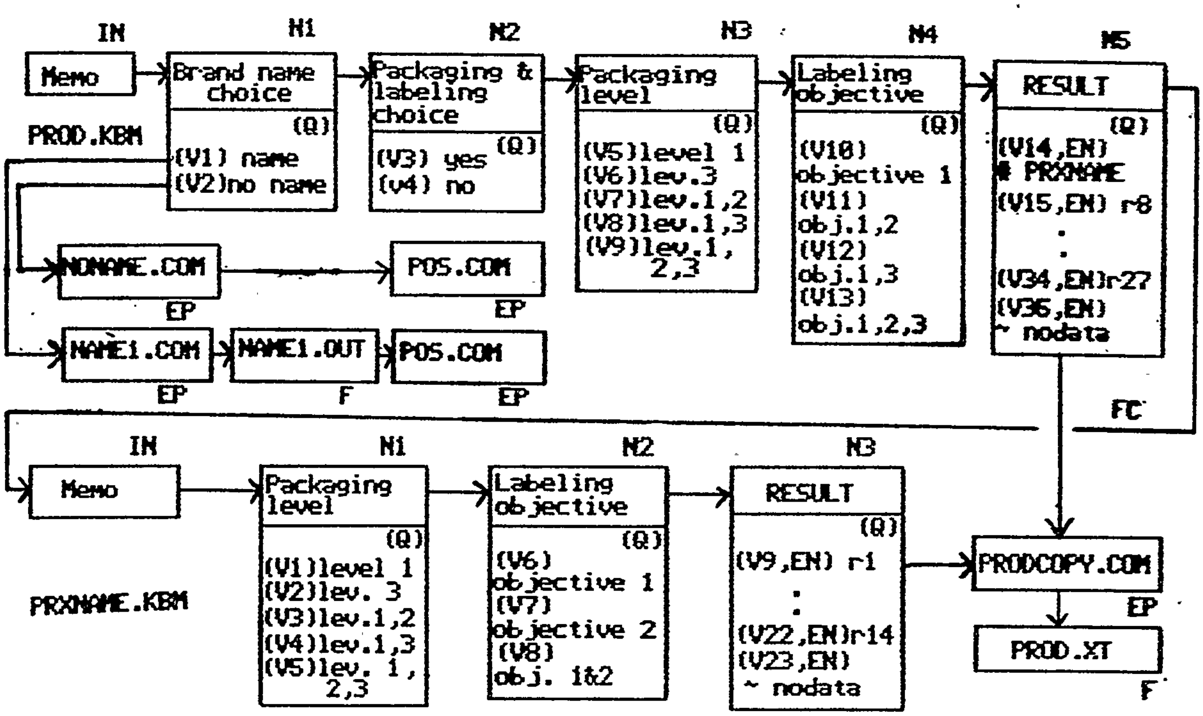


Figure 5.6 PROD.KBM and PRXNAME.KBM knowledge representation and operation structure diagram.

market satisfies the segmentation variables criteria. A car target market and the baby toy market will have different segmentation criteria. These variables can be geographic such as region and country size, demographic such as family size and life-cycle, socioeconomic such as income and education level, psychographic such as social class and lifestyle, and behavioural such as brand loyalty status and attitude toward the product. The number and type of segmentation variables under each major category changes according to the product and the total market environment. Each segmentation variable can be further broken into its actual components. For instance the income segmentation variable can be broken into several income levels, depending on specific or general occupations.

Due to the combinational explosion effect, 1st-Class shell is capable of representing the market segmentation knowledge domain by chaining many knowledge bases. To comply with the constraints and assumptions and to build a practical knowledge base that segments the market in order to determine the target market, the market segmentation knowledge domain was skimmed to include general segmentation variables of the consumer market. These variables portray information that helps the end user to develop segmentation criteria that can be expanded to give a comprehensive description of the target market, and introduces the concept and methods of market segmentation.

DEMO-SEG.KBM and GEO-SEG.KBM are the basic knowledge bases for market segmentation. Demographic and socioeconomic segmentation are determined in the DEMO-SEG.KBM knowledge base. This segmentation includes the family size, life-cycle and income. DEMO-SEG.KBM explains the expected occupation for each income level. The breakdown of these variables is very general. This breakdown yields a maximum of 3 * 9 * 4 = 108 different results part of which is redundant. There are 3 values for the family size, 9 values for the family life-cycle, and 4 values for the income level. Redundant results are all the illogical combinations of these values such as family size of one and a life-cycle described as young and married with no children, and these results are obtained as the value '~ non data' in the RESULT factor. This knowledge base is forward chained to DRES.KBM in order to accommodate all the possible results. The geographic segmentation of the market is portrayed by GEO-SEG.KBM which includes density type and the community size of the target market. The '~ impossible' value in the RESULT factor represents all the redundant combinations in GEO-SEG.KBM.

These knowledge bases provide essential data for the development of the marketing plan. The DEMO-SEG.KBM invokes DEMOSEG.COM, that generates PRICE.ANS answer file used to backward chain the PRICE.KBM knowledge base to MARPLAN. It also interacts with the print-out component of MARPLAN by invoking DEMOCOPY.COM. The GEO-SEG.KBM invokes three external programs. The BUYERS.COM program which asks the end user to give an estimate of the expected number of buyers in the target market after the community size of that market is chosen. This program generates the file BUYERS.OUT which provides POS.COM with the estimated number of buyers. The GEO-SEG.COM generates DIST.ANS answer file used by DIST.KBM to backward chain to

MARPLAN. The GEOCOPY.COM program interacts with the print-out and component of MARPLAN.

The matching method was used to build the decision tree of DEMO-SEG.KBM, DRES.KBM and GEO-SEG.KBM to ensure that the sequence of questions to be answered coincides with the order of factors on the definitions screen. All knowledge bases include a complete set of examples to enable the advisor to handle redundancy. Figures 5.7 and 5.8 show the knowledge representation structure diagrams of the three knowledge bases.

5.5.4. The POS.COM external program

POS.COM is an external program that performs market analysis, product positioning and gives cost and sale estimates. This program is the major data processing component of MARPLAN where most of the input data required for the decision making process is obtained from the end user. MARPLAN invokes POS.COM upon the end user's request. POS.COM was developed to process input data since 1st-Class shell does not furnish any computational nor graphic facilities. The operations executed by this program are equivalent to investigating the third basic marking variable of the market plan, namely the competition, and providing essential data that complements the marketing plan. POS.COM is composed of three major parts: the market analysis and product positioning part, the demand and roles estimate part, and the cost estimate part.

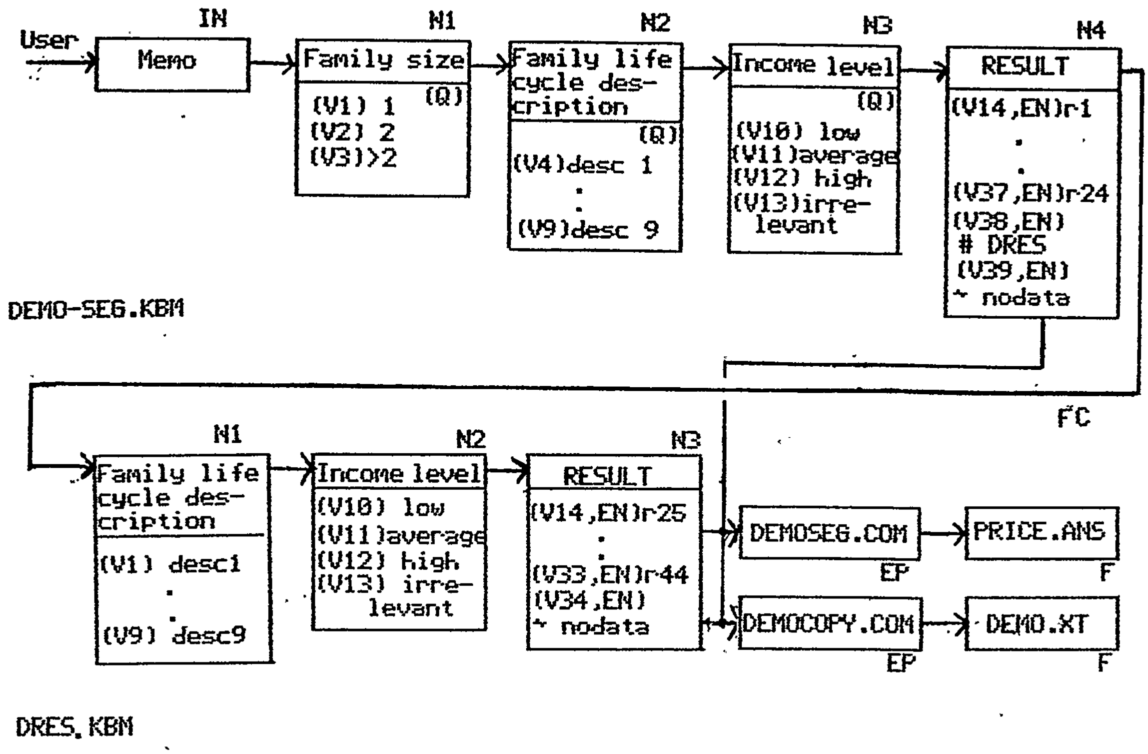


Figure 5.7 DEMO-SEG.KBM and DRES.KBM knowledge representation structure diagram

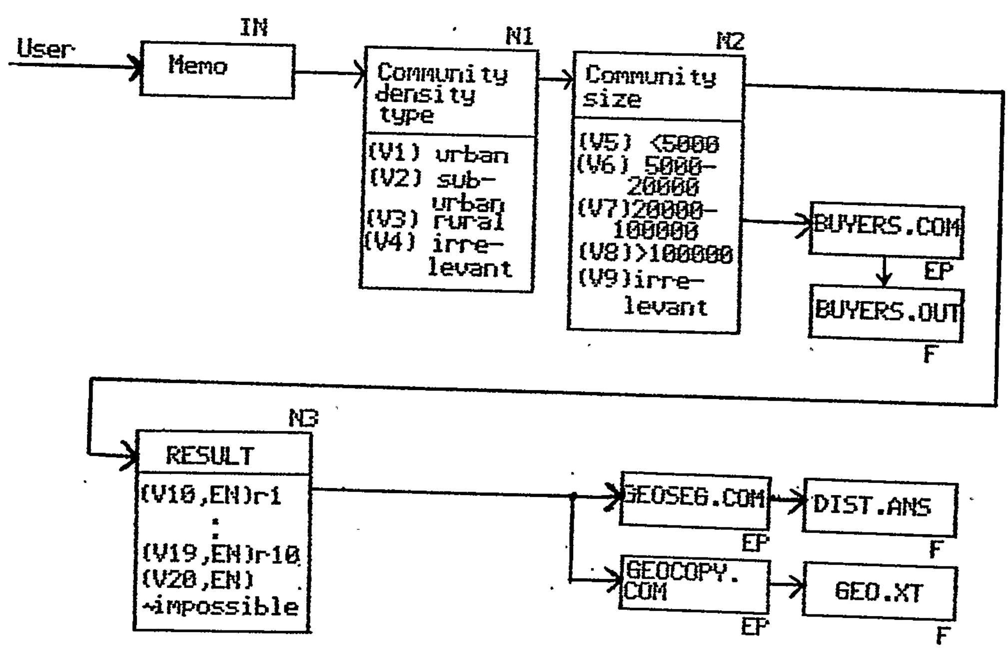


Figure 5.8 GEO-SEG.KBM knowledge representation structure diagram

(1). Market analysis and product positioning.

Before proceeding with market analysis and product positioning, the end user at that stage has examined the product and determined the target market. These realizations aid the end user in investigating competition. There are several factors that contribute to a comprehensive analysis of the potential competition. These factors include strengths and weaknesses of the competitors, their distribution channels, promotional mix, general capabilities. Other factors are determined by the particularity of the competitive environment. The traditional approach taken to analyse these factors focuses on scanning each factor separately, then collecting all the information and deducing a set of conclusions that hint at ways of achieving the marketing objectives successfully despite competition. Such an approach requires natural language interaction between the end user and the expert systems, or a complete knowledge base that embodies all possible and non-redundant information about the competition analysis factors. Requirements of this magnitude are beyond the capabilities of the 1st-Class shell, and do not confine to the constraints of the marketing plan. Hence, a method had to be developed that can analyse competition and be a basis for further decisions. Marketing management offers two techniques which when combined can form a powerful method for competition analysis. These techniques are based on the product attributes and positioning concepts. Product attributes are the main values offered by a product that influence a customer's decision in the process of buying that product. This concept tries to answer the question of why does a customer decide to buy a certain product from a set of products that satisfy the customer's need. For a product of the class and type that MARPLAN is built to handle, four main attributes are used to describe the customer's buying behaviour. These attributes are the product price, quality, features and style. Product positioning is the process of understanding the target market's perception of the product relative to the major competing products in the same market [24]. The product attributes can be used as measurement parameters to evaluate the customer's perception of the product and how it stands among other competing products. Once an evaluation is determined, an estimate of which product is expected to sell most can be obtained. At this stage, the end user knows where the product stands, and has an idea of how the product should be improved to increase its chances of success in the market.

The market analysis and product positioning part of POS.COM translates the concepts and ideas mentioned above into a series of five screens complete with instructions and explanations. The program begins by prompting the end user with the name of the product that was chosen in the PROD.KBM knowledge base, or is asked to provide a refence name. The end user must specify up to five potential competitors with the price of their products in dollars per unit. POS.COM then displays the attribute table on which the end user is asked to assign an importance weight for each of the four attributes, then to assign an attribution value for each attribute of all the products. The total attribution is then automatically calculated using the attribution formula:

$$A = \sum_{i=1}^{n} a_i b_i \quad , \tag{5.1}$$

where A is the total attribution, a_i is the importance weight of attribute i, b_i is the attribute level of the product for attribute i, and n is the maximum of attributes. With the total attribution of all the products available, the product with the highest attribution is the product expected to sell most in the market. POS.COM displays this result by arranging all the products in a descending order and highlighting the product with the highest attribution level among the competing products. The program then shows the products positioning on two consecutive plots. The first plot gives the quality versus price as attribution parameters, and the second plot shows features versus style as attribution parameters.

(2). Demand and sales estimate.

The purpose of this part of POS.COM is to give an estimate of the actual number of buyers per year and an estimate of the number of units of the product to be manufactured per year. A simple procedure was developed to give these estimates. The total annual potential market sales in dollars, Q, is calculated using,

$$Q = n\ q\ p \quad , \tag{5.2}$$

where n is the estimated total number of buyers provided by the end user in GEO-SEG.KBM, q is the product quantity purchased by an average customer per year, and p is the average price per unit in dollars. The values q and p are obtained from the end user, and p is calculated using

$$p = \sum_{i=1}^{j} \frac{p_i}{j} \quad , \quad \text{maximum} \quad j = 5 \tag{5.3}$$

where p_i is the price per unit in dollars of competitor i product and j is the number of competitors. The next step is to determine the available market share in dollars. The end user is asked to provide statistics about the annual market share in dollars of the competitors within the target marketing area. Hence, the available market share will be

$$m = Q - T \tag{5.4}$$

where T is the total annual market share in dollars of the potential competition. At this point, enough data is available to determine the actual market share in dollars. The end user is asked to specify in percent an estimate of the number of customers who are accessible, aware, and intend to buy the product. The actual annual market share will be

$$a = m \ (a_1 \, a_2 \, a_3) \times 10^{-6} \tag{5.5}$$

where a_1, a_2 and a_3 are the estimated percentages. The estimated actual number of buyers will then be,

$$b = n(a_1 \, a_2 \, a_3) \times 10^{-6} \tag{5.6}$$

and the estimated demand, or number of units to be manufactured per year is,

$$u = b \ n \tag{5.7}$$

This part of POS.COM concludes by a screen displaying the sale estimate results.

(3). Cost estimate.

The estimated annual quantity of the product, needed to satisfy the market demand is an essential piece of information in estimating the cost of production and marketing that quantity. Having estimated the number of units to be manufactured for one year in the second part of POS.COM, the third part provides a general cost statement that enables the end user to estimate that cost. The general cost statement is divided into product costs and period costs. The total product costs are the sum of the manufacturing overhead, indirect labor, and direct materials. The total period cost is the sum of the administration and selling costs [28]. The total estimated cost is the sum of both costs. The breakeven price of one unit in products can be easily determined. Two values are calculated, one taking the product costs only, and another taking the total costs. This part of POS.COM is concluded by a screen that displays all these important results.

POS.COM interacts with the other components of MARPLAN by reading or adjusting the data in NAME1.OUT, BUYERS.OUT, DIST.ANS, PROM.ANS files, and generating 16 data files that are used by PRICE.KBM, LOSSGAIN.COM and the print-out component of MARPLAN. A detailed analysis of the information embodied in POS.COM and the rules binding this information, in addition to flowchart diagram of POS.COM can be obtained from [23]. This program is complete with explanations that help an end user to carry out an advisory session efficiently and with ease.

5.5.5. The PRICE.KBM knowledge base

PRICE.KBM is a knowledge base built by 1st-Class shell that invokes PRICE.COM or PMETHODS.COM external programs depending on the end users choice. This knowledge base is backward chained to MARPLAN through the answer file PRICE.ANS. Both external programs were written in Turbo-Pascal, and each gives a different pricing decision. At this stage of development of the marketing plan, the breakeven price per unit of the product, the prices per unit of the competing products, and information about the target market, are all available and can be processed to give the product a competitive selling price. PRICE.KBM provides the end user with two options. The first option is to display MARPLAN pricing decision through invoking PRICE.COM. The second option enables the end user to set a price per unit for the product according to any of four acknowledged pricing methods by invoking PMETHODS.COM.

Price is a marketing variable which is a basic component of the marketing plan. The pricing decision is of vital importance in achieving the success of a product. The product, being imitative in nature, must be priced following a certain pricing strategy that minimizes risks and accounts for the objective of this plan which is to achieve a market share and, if possible, some profit. With these guidelines in mind, the options offered by PRICE.KBM is discussed in what follows.

(1) MARPLAN pricing decision

Pricing an imitative new product is a process that involves a product positioning problem [24,25]. In PRICE.KBM, the pricing strategy was set according to the target's market income level and the market leaders pricing strategy, taking the product price and quality as product positioning parameters. The choice of the pricing strategy that suits the product most was determined by common sense and pricing guidelines indicated in reference texts [24,26]. The basic guideline states that if the market leader followed a certain pricing strategy then the newcomer might prefer to use one of nine other possible strategies. These strategies are shown on the Price/Quality product positioning plot on Fig. 5.9. In addition, it is important to note that PRICE.COM pricing decisions for a target market with an "irrelevant" income level are the same for a target market with a low income level. This program is also set to output an error message in case of insufficient input

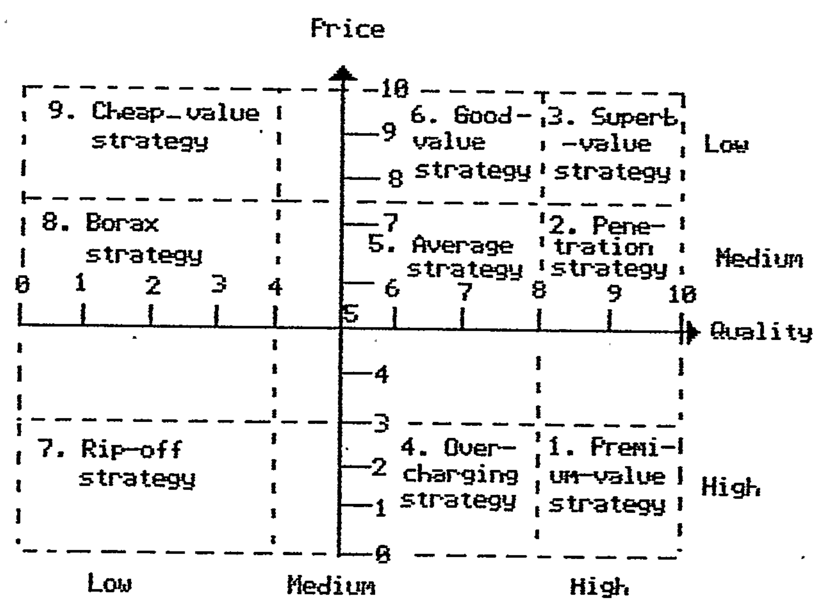


Figure 5.9 Nine possible strategies on Price/Quality

data.

The basic guideline was translated into pricing decisions given by MARPLAN. These decisions and their conditions of occurrence are outlined in Tables 5.1, 5.2 and 5.3 where all the values of ranges are in relation to Fig. 5.9. The advisor, or MARPLAN, pricing decision includes advice on the appropriate attributions range values for better product performance, and instructs the end user to set a price for the product within an optimal price range in dollars. PRICE.COM interacts with MARPLAN by reading from data files from POS.COM and generating PRICE.OUT which is read by LOSSGAIN.COM, and other files required by the print-out component of MARPLAN.

(2) The pricing methods

The second pricing decision option of PRICE.KBM enables the end user to choose one of four acknowledged pricing methods. This option was included to assert the flexibility of a pricing decision in MARPLAN since this decision is limited with constraints imposed by the amount of data it can process. The pricing methods option is initiated upon the end users request by invoking the PMETHODS.COM external program. This first pricing method is the cost-plus pricing. It is a simple method in which a standard markup is added to the cost of the product. The second method is the break-even pricing method. This method is cost-oriented where the marketing objective served is establishing a market share disregarding profit and avoiding losses. The perceived-value pricing method in which the key pricing is the buyers' perception of the product value. The fourth

Price Range (P)	Quanlity Range (Q)	Market Leader Pricing Strategy	Product Pricing Strategy
$0 < P < 3$	$Q \geq 8$	1 Premium	2
$3 \leq P < 7.5$	$Q \geq 8$	2 Penetration	3
$P \geq 7.5$	$Q \geq 8$	3 Superb-value	6
$0 \leq P < 3$	$4 \leq Q < 8$	4 Overcharging	5
$3 \leq P < 7.5$	$4 \leq Q < 8$	5 Average	6
$P \geq 7.5$	$4 \leq Q < 8$	6 Good-value	9
$0 \leq P < 3$	$0 \leq Q < 4$	7 Rip-off	8
$3 \leq P < 7.5$	$0 \leq Q < 4$	8 Borax	5
$P \geq 7.5$	$0 \leq Q < 4$	9 Cheap-value	6

Table 5.1a Pricing Decisions for long term marketing objectives - Low income group (Price Sensitive)

Price Range (P)	Quanlity Range (Q)	Market Leader Pricing Strategy	Product Pricing Strategy	Comments (must improve on features)
$0 < P < 3$	$Q \geq 8$	1 Premium	2	MIOF*
$3 \leq P < 7.5$	$Q \geq 8$	2 Penetration	3	MIOF*
$P \geq 7.5$	$Q \geq 8$	3 Superb-value	3	MIOF & P less than leader
$0 \leq P < 3$	$4 \leq Q < 8$	4 Overcharging	5	MIOF*
$3 \leq P < 7.5$	$4 \leq Q < 8$	5 Average	6	MIOF*
$P \geq 7.5$	$4 \leq Q < 8$	6 Good-value	6	MIOF & P less than leader
$0 \leq P < 3$	$0 \leq Q < 4$	7 Rip-off	4	MIOF & P less than leader
$3 \leq P < 7.5$	$0 \leq Q < 4$	8 Borax	5	MIOF*
$P \geq 7.5$	$0 \leq Q < 4$	9 Cheap-value	6	MIOF*

* Must improve on features

Table 5.1b Pricing Decisions for long term marketing objectives - Average income group (Price, Quality and Features Sensitive).

Price Range (P)	Quanlity Range (Q)	Market Leader Pricing Strategy	Product Pricing Strategy	Comments (better style)
$0 < P < 3$	$Q \geq 8$	1 Premium	1	better style
$3 \leq P < 7.5$	$Q \geq 8$	2 Penetration	1	better style + min Q as ML*
$P \geq 7.5$	$Q \geq 8$	3 Superb-value	2	better style + min Q as ML*
$0 \leq P < 3$	$4 \leq Q < 8$	4 Overcharging	1	better style + max P as ML*
$3 \leq P < 7.5$	$4 \leq Q < 8$	5 Average	2	better style + max P as ML*
$P \geq 7.5$	$4 \leq Q < 8$	6 Good-value	5	better style + min Q as ML*
$0 \leq P < 3$	$0 \leq Q < 4$	7 Rip-off	4	better style + max P as ML*
$3 \leq P < 7.5$	$0 \leq Q < 4$	8 Borax	5	better style + min Q and max P as ML*
$P \geq 7.5$	$0 \leq Q < 4$	9 Cheap-value	6	better style + min Q and min P as ML*

*ML - Market Leader

Table 5.1c Pricing Decisions for long term marketing objectives - High income group (Quality and Features Sensitive) pricing method is the going-rate pricing in which the price is set largely on competitors prices, with less attention paid to cost or demand. The price given is the average price of the competitors products. PMETHODS.COM interacts with MARPLAN by reading two input files and generating PRICE.OUT and other files required by the print-out component of MARPLAN.

Since the decisions are known and manageable in number, a complete set of examples was built, and consequently, the optimization method was used to construct the knowledge representation decision tree. PRICE.KBM is complete with instructions and explanations. Figure 5.10 shows the PRICE.KBM knowledge representation structure diagram.

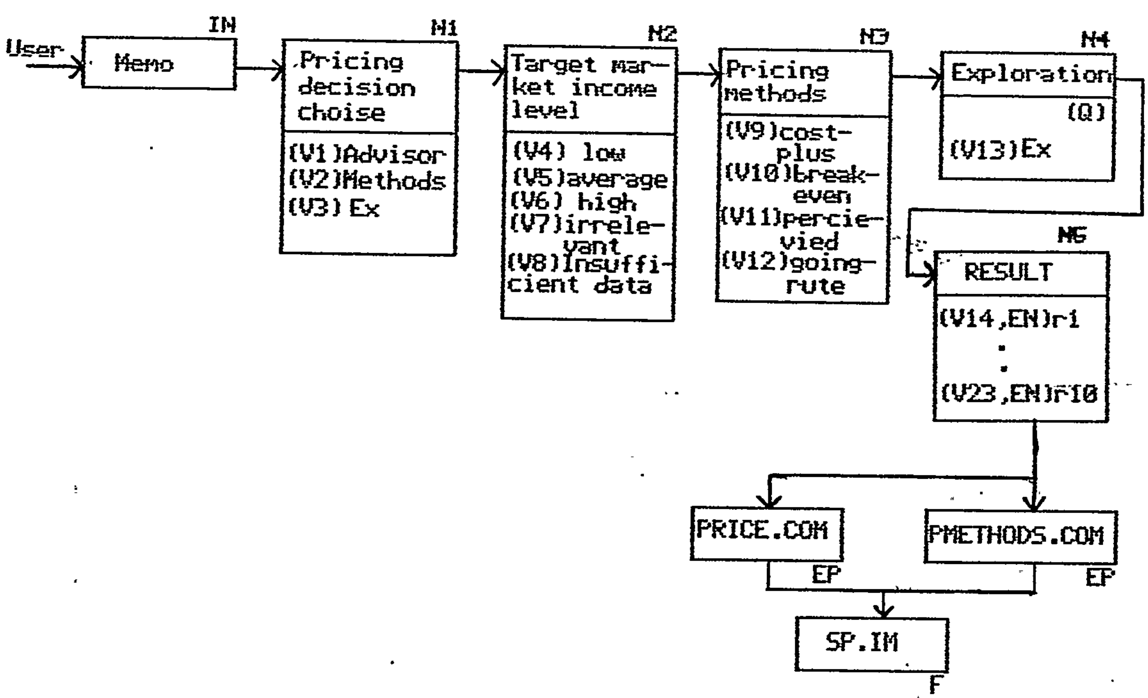


Figure 5.10 PRICE.KBM knowledge representation structure diagram

5.5.6. The DIST.KBM knowledge Base

Product distribution can be defined as the various activities undertaken to make a product accessible and available to the target market [24]. DIST.KBM is a knowledge base built by 1st-Class shell that provides the distribution decisions of the advisor. The knowledge base is backward chained to MARPLAN through the DIST.ANS answer file. Product distribution is a wide knowledge domain where major decisions that detail the distribution process are taken after the evaluation of a considerable number of product distribution variables. The approach to making decisions in DIST.KBM does not deviate from the guidelines of MARPLAN, and its decisions complement the components of the marketing plan by being general in nature and capable of achieving the productability and educational objectives for the same reasons outlined in the analysis of other knowledge bases. DIST.KBM decisions advice on the optimal distribution level out of five possible distribution levels. The decision is reached after weighing the target markets community density type, the financial capabilities of the party interested in marketing the product, and the contacts available for that party within the distribution channels. These are the essential elements involved in deciding for a distribution level. DIST.KBM interacts with MARPLAN by reading an input data file (PROM.OUT) from POS.COM and generating the PROM.ANS answer file and other files required by the print-out component of MARPLAN. All the distribution decisions are non-redundant and of a manageable number. Hence, a complete set of examples describing the knowledge domain was built and the optimization method was consequently used to construct the knowledge representation decision tree. Figure 5.11 shows the DIST.KBM knowledge representation structure diagram.

5.5.7. The PROM.KBM knowledge base

Product promotion can be defined as the various activities undertaken to communicate a certain message to the target market showing the product merits and trying to persuade the target customers to buy that product [24]. PROM.KBM is a knowledge base built by 1st-Class shell that provides a promotion decision through weighing the distribution decisions, promotional funds availability, and the target markets degree of awareness of the product. This knowledge base is backward chained to MARPLAN by the answer file PROM.ANS. Product promotion is a large knowledge domain that has evolved to become an art and a science. This knowledge base confines to the same guidelines of generality that have

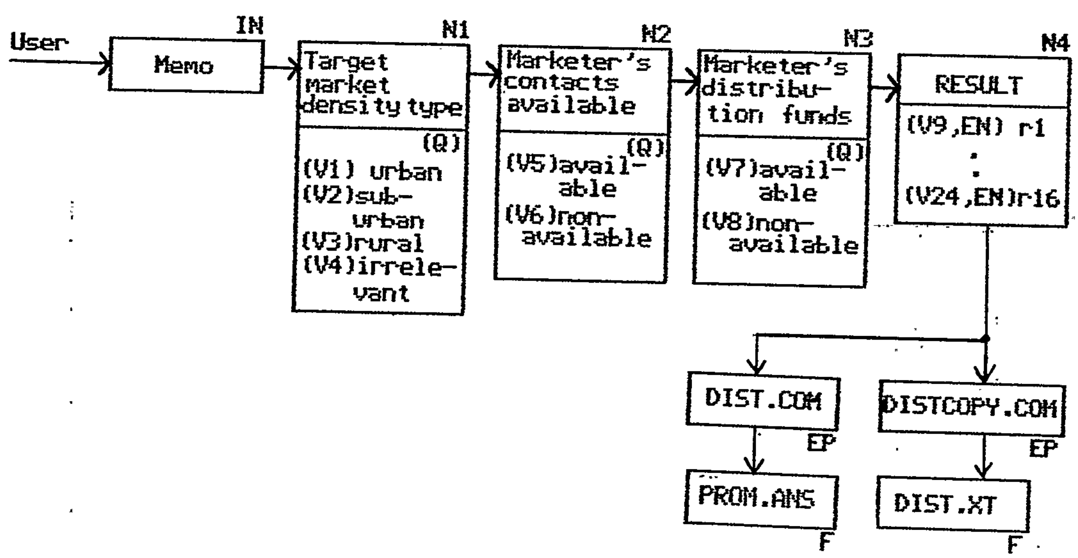


Figure 5.11 DIST.KBM knowledge representation structure diagram

architectured the decision making process of MARPLAN. Hence, PROM.KBM complements MARPLAN by providing its promotional component, giving decisions that outline the best promotional strategy with the given input data and suggest some effective promotional tools. This knowledge base interacts with MARPLAN by generating files required by its print-out component.

A complete set of examples was built, since all the results of the knowledge domain were known. Hence, the optimization method was used to build the knowledge representation decision tree. Figure 5.12 shows the PROM.KBM knowledge representation structure diagram.

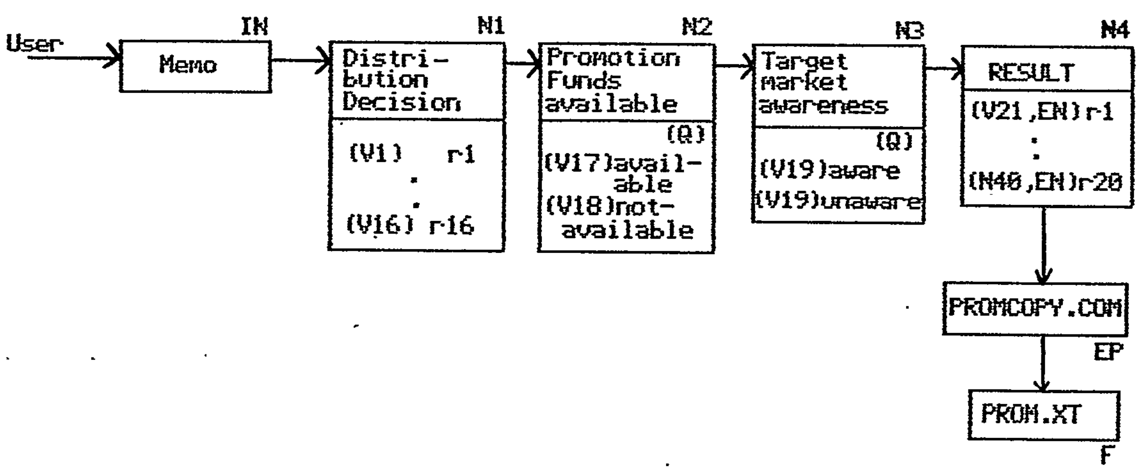


Figure 5.12 PROM.KBM knowledge representation structure diagram

5.5.8. The CLIM.KBM knowledge base

The market environment is the status of the microenvironment and macroenvironment of the market at a given time. The microenvironment is the set of factors that affect the production and sale of the production, and sale of the product, such as raw material supplies, marketing intermediaries, and public laws. These factors are mainly related to the conscientiousness of the marketer of the producer's status and are not covered in MARPLAN due to their particular knowledge domain. The macroenvironment are the broad trends in demographic, economic, physical, technological, political/legal, and social/cultural developments. The nature of the knowledge domain that MARPLAN tackles is of a limited size and does not permit on elaboration on the market macroenvironment in which the product is manufactured. Hence, for reasons of practicality, and due to the nature of the problem, the macroenvironment aspects investigated in MARPLAN are narrowed down to the legal aspects. CLIM.KBM is a knowledge base built using the 1st-Class shell in which the legal aspects of the market macroenvironment are handled. It is a simple knowledge base that was included as a component of MARPLAN to focus on the existence of this part in a marketing plan. This knowledge base includes introductory information aimed at pointing out major elements of the legal aspects. These elements have a greater impact on the production process, and the general advice given by MARPLAN through CLIM.KBM increases the end user's degree of awareness of the legal aspects in the production process. CLIM.KBM interacts with MARPLAN by generating files required by its print-out component. The results of CLIM.KBM are all non-redundant and of a manageable number. Hence, a complete set of examples was built and the optimization method was used to construct the knowledge representation decision tree. Figure 5.13 shows the CLIM.KBM knowledge representation structure diagram.

5.5.9. The LOSSGAIN.COM external program

The final component of the marketing plan simulated in MARPLAN is the loss-gain statement. LOSSGAIN.COM is an external program that calculates the expected market share in percent for the estimated rules, gives the projected annual profit or loss in dollars taking the product costs only and the total costs, and issues a "go" or "no-go" decision based on this data. These values are obtained by processing the sales, cost and price results. The methods followed to calculate these values are simple and can be obtained with the program flow charts from [23].

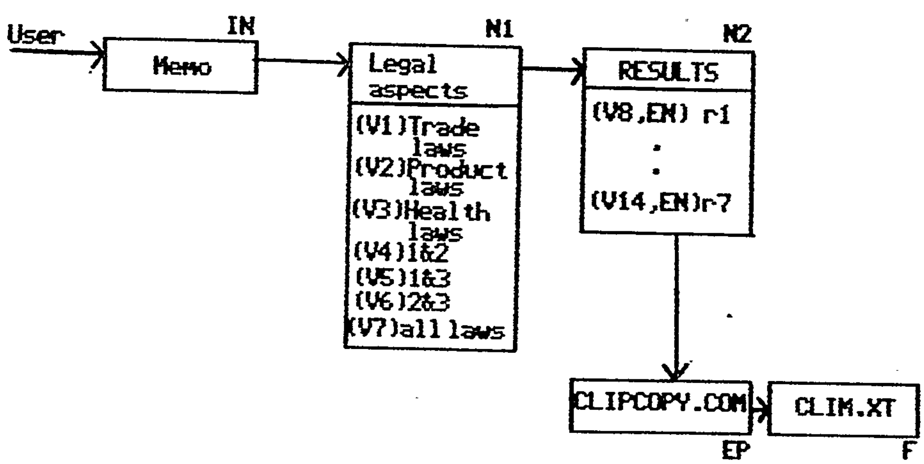


Figure 5.13 CLIM.KBM knowledge representation structure diagram

LOSSGAIN.COM interacts with the other components of MARPLAN by reading and processing two data files from POS.COM and one data file from PRICE.KBM, and by generating files required by the print-out component of MARPLAN.

5.5.10. The print-out component of MARPLAN

A comprehensive understanding of how a marketing plan is developed, within the objectives and the set of assumptions and constraints limiting the scope of this work, can be obtained when a compact output of the marketing plan that includes all its components is available. The 1st-Class shell does not provide one printable output for the results obtained during a consultation. To overcome this difficulty, each possible result for all the components of MARPLAN was saved in a separate text file with the ".IN" file extension name. If the advice or result is from a external program such as POS.COM or PRICE.COM, then a special subroutine written and included in these programs to save the screen in a file with ".IM" extension, is invoked. Whenever an advice or a result is reached during an advisory session with MARPLAN, an external program is invoked with the "*COPY.COM" file name that reads the ".IN" file associated with this result or advice and stores it in a file with the ".XT" extension. A copy program was written for each knowledge base. At the end of an advisory session, MARPLAN would have generated all the text and image files that include the results or advices given during that session. To obtain a standard marketing plan print-out, all these files must be organized and prepared to be printed on a printer. A special batch file was written to perform this task. MARKET.BAT is a batch file that, among other functions, organizes all these files and calls the PRTSC.COM external program that retrieves the image files then sends all the results to the printer.

5.6. MARPLAN Control Mechanism

To control MARPLAN execution and facilitate its marketing plan print-out component, the DOS system MARKET.BAT batch file was written. This batch file makes use of the MS-DOS system ERRORLEVEL. This ERRORLEVEL is an internal DOS number often used as a debugging tool to indicate when a program has had an error [15]. It can also be used to communicate between programs and between programs and batch files. This number can take any value from 0 to 255. 1st-Class shell inference engine, the advisor, automatically sets the DOS ERRORLEVEL to a number corresponding to the position of the advice or result it gave last in the RESULT column. For instance, if the result given was the fifth value on the RESULT column, then the ERRORLEVEL is equal to 5. An MS-DOS system batch file can read this ERRORLEVEL by the simple conditional statement, IF ERRORLEVEL *nn* statement, where *nn* is the ERRORLEVEL number and statement is any MS-DOS command useful for execution purposes. The end user activates MARKET.BAT by typing MARKET, which invokes the CAN.COM external program that deletes all data files generated by MARPLAN during a previous advisory session, then deletes all the ".XT", ".IM" and ".ANS" files already present, then calls the advisor MARPLAN. IF the user ends an advisory session abruptly, MARKET.BAT would halt the execution of MARPLAN and exit to the MS-DOS by conditional statements that read the ERRORLEVEL and enforce the exit command. If the ERRORLEVEL read by MARKET.BAT is 11, which corresponds to the print-out component of MARPLAN (eleventh value on MARPLAN RESULT column), then the batch file automatically saves all the text files into two files, M1.XT and M2.XT, then invokes PRTSC.COM, and prints out the marketing plan in a standard form.

5.7. MARPLAN Testing Process

The purpose of testing MARPLAN was to determine if the system is functioning according to the design criteria and to obtain feedback of any existing programming and knowledge domain errors and correct such mistakes. The system was run and inspected for existing errors. All these errors were detected and corrected. The result was an enhancement in the user-friendliness of the system. All the input data was carefully set within reasonable names and illogical or contradictory decisions were avoided and identified through appropriate error messages. The knowledge embodied in MARPLAN was checked thoroughly. Utmost care was taken to ensure an accurate and exact simulation of the marketing plan development knowledge domain. The checking process was executed by comparing the sources of the knowledge to MARPLAN knowledge base with all its components. As for evaluation of the correctness of this knowledge and the decisions by marketing management experts, only one expert was available for such an assessment. Dr. J. Graham of the Faculty of Management at The University of Calgary revised the knowledge embodied in MARPLAN and agreed on the correctness of the approach, but criticized the validity of certain decisions and concluded that this system can be used by a marketer as a checking list for an actual marketing plan. The main reason for this conclusion was based on the fact that the factors which determine any conclusion in all the components of MARPLAN are not enough to justify that conclusion. A demonstration of MARPLAN was held at SHELL Tower in Calgary in the presence of Mr. John McClurg, a marketing manager at SHELL, and two other engineers working in the field of Expert Systems. This group indicated that this system can be beneficial as a successful introduction to marketing plan development. The ideal testing process is in trying MARPLAN on several products and obtaining opinions of entrepreneurs and marketers about the strengths and weaknesses of the system. Such a process is difficult to achieve knowing that the development of MARPLAN was for academic research and due to scarceness of data sources of this type. MARPLAN in its present form is operational and up to a good estimate, is error free. Figure 5.14 shows the MARPLAN system structure diagram.

5.8. MARPLAN Evaluation

As in evaluating the expert systems developed for three laboratory experiments in control, MARPLAN is evaluated in relation to whether it is an expert system, to the degree of its goal achievement, and to the evaluation criteria outlined in Chapter 2 which is actually the set of strengths and weaknesses of the system.

(1) Is MARPLAN an expert system?

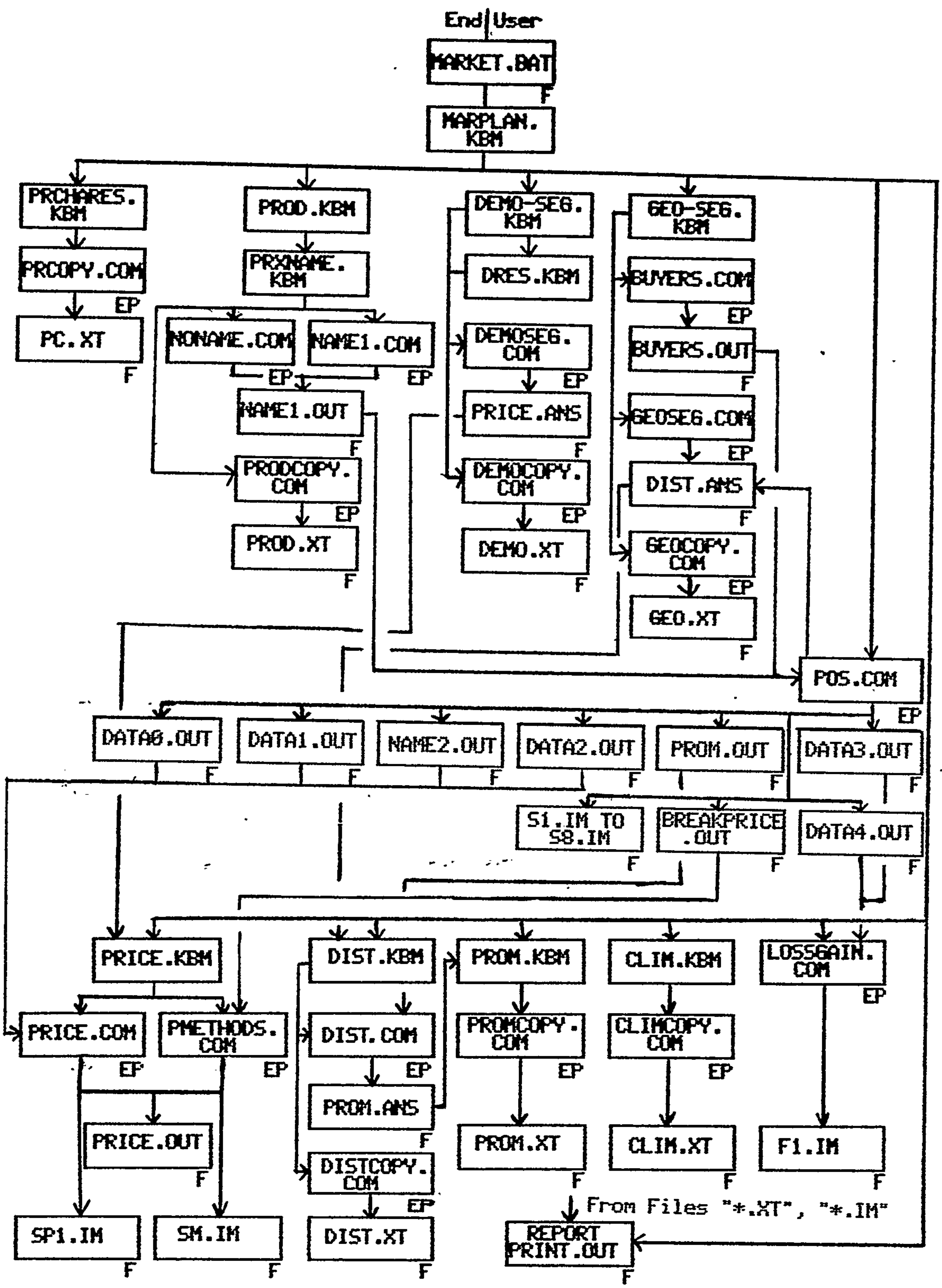


Figure 5.14 MARPLAN system structure diagram

The architecture of MARPLAN replicates that of an expert system. It has a knowledge base and a group of external programs and files interacting with each other using the inference engine provided by 1st-Class. MARPLAN tackles a specialized domain, but not narrow in nature since it involves a large number of variables. This system is capable of reasoning with uncertainty by giving the end user the choice to express lack of information and still provide conclusions. Although not all the results are advices, but the output as a whole is an advice, and the system is typically rule-based. In addition, the MARPLAN inference mechanism is separate from the knowledge base, and is capable of explaining its advice or line of reasoning upon request. Hence, MARPLAN is an expert system disregarding its level of performance.

(2) MARPLAN degree of goal achievement

As an expert system, MARPLAN is supposed to output sound and expert advice in the form of a marketing plan for a certain product. The advice it outputs is a marketing plan, but is rather limited due to the binding set of constraints and assumptions imposed on the scope of this application. Hence, the knowledge embodied in MARPLAN is not adequate to output an applicable advice in a realistic environment. To overcome this drawback, the MARPLAN knowledge base should be expanded by including more variables to its components and by chaining to additional component knowledge bases. MARPLAN did achieve its academic goals, since it can be considered as a framework which can be used to build a system applicable to real situations.

(3) MARPLAN strong points

This expert system embodies knowledge related to developing an expert system which is correct and valid within the domain of its application. As mentioned earlier, the knowledge with its facts, rules and data processing was thoroughly checked to assert its correctness. MARPLAN with all its components is user-friendly, flexible and easy to use. This was accomplished by instructions, error messages, and options. The system provides definite educational benefits and can be improved for actual application. The fact that the inference engine is a separate entity from the system itself enables considerable future development by the addition of more knowledge bases. The feasibility of building MARPLAN is demonstrated in the research field and is outlined in Chapter 6.

(4) MARPLAN weaknesses

The efficiency of this expert system is not as high as expected. The system is slow to start and in invoking external programs. This drawback is due to the fact that 1st-Class does not provide a collective printable output which can be formatted to fit a standard requirement for a session that involves chained knowledge bases. Hence, the batch file built to overcome this weakness is slow to operate. Eventhough the system is accessible and responds when a session is interrupted, it lacks the ability of natural expression. Naturalness of expression was sacrificed for feasibility and generalization. This is most evident in the promotion and distribution decisions which fail to give an explicit account on means of distribution and promotion. As for the credibility of this expert system, its decisions seem to orient the end user towards developing a marketing plan through iteration. This means that several sessions must be executed before a satisfactory plan can be obtained. The credibility of MARPLAN cannot be assessed unless it was actually tried by several end users. In addition, building this system without the supervision of a marketing management expert deprived the knowledge base from heuristics which could have minimized the decision making process.

The real evaluation of MARPLAN would be more accurate if the system was actually tried after further development and refinement, but as in every research oriented application in any field, the assessment stems from the degree of contribution that such an application yields to improve the knowledge in and about that field.

CHAPTER 6

CONCLUSIONS AND SUGGESTIONS FOR FURTHER RESEARCH

6.0. Conclusions

The possibility of applying expert systems in the electrical engineering field have been explored through the development of three laboratory instruction expert systems. ADVISOR1, ADVISOR2 and ADVISOR3 can be consulted by students performing three experiments in control to conduct these experiments with a minimum degree of error. These systems were tested and can be considered operational. The expert systems developed exposed the great benefits that can be hoped for by applying these systems for educational purposes which simultaneously involved theoretical and analytical aspects, proving that such an application is highly feasible and promising.

The application of expert systems in marketing management was investigated through developing MARPLAN, an expert system that assists an entrepreneur in developing a marketing plan for a durable imitative product used in households.

An expert system design and build-up procedures were developed in the process of constructing the four expert systems. These procedures can be utilized for further development of any expert system using an expert shell as a build-up tool, especially the 1st-Class expert system shell. The design and build-up procedures can be generalized due to their practicality, as they stem from a realistic approach undertaken to develop feasible expert systems in two different knowledge domains. This generalization has to be assessed through further investigations by building more expert systems using these procedures.

The credibility of an expert system is enhanced for procedural and deterministic knowledge domains. This leads to a main conclusion that deleting uncertainty enhances the credibility level of an expert system.

An ideal expert shell used for applications in electrical engineering must furnish effective expressive facilities such as simulation, editing, computational and graphic display capabilities in addition to all the basic facilities that a shell such as the 1st-Class expert shell has to offer.

Using an expert shell as a build-up tool for expert systems applications in electrical engineering hinted at the limited application programming nature of expert systems and that they cannot be dealt with as a distinct scientific discipline such as control theory or the study of power systems.

6.1. Suggestions for Further Research

Research in the field of expert systems is active and is progressing at an exhilarating rate. Expert systems are an outcome of the field of artificial intelligence, but is now treated as a separate discipline in computer sciences. The research is centered on developing better build-up tools and more efficient expert shells. Hence, as an independent research field, more investigation has be be focused on knowledge acquisition and representation for general use and for particular applications such as electrical engineering. Developing knowledge acquisition modules for elecitating expert knowledge from a particular electrical engineering knowledge domain such as telecommunications can save time and have considerable benefits for applying expert systems in telecommunications.

Further research can be conducted on applying expert systems in the electrical engineering field for educational purposes by building expert systems for laboratory instruction using the laboratory instruction expert system general structure developed in this thesis for different laboratory environments and requirements. In the marketing management field, an ambitious objective is to build an expert system that can develop a comprehensive plan for a certain class of products using the framework developed in this thesis.

Possible applications of expert systems in electrical engineering seem to be numerous. A good application is to build an expert system for VLSI testing diagnostics and design. Another application is building an expert system that can advise on designing control systems. An expert system that can detect faults in power plants and advice on solutions can be a feasible application. If an expert shell is used to build these expert systems or any other systems, it is of interest to follow the design and build-up procedures and measure the extent of generalization that these procedures offer.

REFERENCES

[1] R. Keller, "*Expert System Technology,*" Yourdon Press: Englewood Cliffs, N.J., 1987.

[2] G. L. Simmons, "*Expert Systems and Micros,*" NCC Publications: Manchester, England, 1985.

[3] F. Jay (Editor), *IEEE Standard Dictionary of Electrical and Electronics Terms,*" IEEE: New York, 1984.

[4] T. Johnson, "*The Commercial Application of Expert System Technology,*" OVUM Ltd., 1984.

[5] Hayes-Roth, F., D. A. Wateman and D. B. Lenat, "*Building Expert Systems,*" Addison Wesley: Reading, Mass., 1983.

[6] R. Forsyth, "*Expert Systems,*" Chapman and Hall: London, 1984.

[7] T. R. Addis, "*Designing Knowledge-based Systems,*" Kogan Page: London, 1985.

[8] A. Walker, "*Knowledge Systems and Prolog*", Addison-Wesley: Reading, Mass. 1987.

[9] P. Jackson, "*Introduction to Expert Systems,*" Addison-Wesley: Wokingham, Berkshire, 1986.

[10] A. Hart, "*Knowledge Acquistion for Expert Systems,*" McGraw-Hill: N.Y., 1986.

[11] L. Kramer, "*Knowledge Representation in Expert Systems,*" National Library of Canada: Ottawa, 1985.

[12] J. Sowa, "*Conceptual Structures,*" Addison-Wesley Inc.: N.Y., 1984.

[13] J. R. James, "*Considerations Concerning the Construction of an Expert System for Control System Design,*" Ph.D. Thesis, Renssalear Polytechnic Institute: Troy, N.Y., 1986.

[14] R. B. Streets, Sr., "*Diseases of the Cultivated Plants of the Southwest,*" The University of Arizona Press: Tucson, Arizona, 1969.

[15] W. Thomas "*1st-Class Instruction Manual,*" Programs in Motion Inc.: Wayland, Mass., 1986.

[16] G. A. Korn and T. M. Korn, "*ENHANCED DESIRE,*" Tucson, Arizona, 1986.

[17] SUPERKEY, Borland International: Scotts Valley, California, 1985.

[18] K. Duncan and D. Harris (Editors), "Computers in Education", *Proc. IFIP World Conference:* Norfolk, Va., Elsevier, 1985.

[19] "*Modular Servo Type MS150, Parts I & II*", Feedback Instruments Limited: Crowborough, Sussex, England, 1982.

[20] "*Modular Servo System MS150, Book 4, Simulated Relay System SR 150G*", Feedback Instruments Limited: Crowborough, Sussex, England, 1982.

[21] K. Ogata, "*Modern Control Engineering*", Prentic-Hall Inc.: Englewood, Cliffs, N.J., 1970.

[22] G. H. Hostetter, F. J. Savant and R. T. Stefani, "*Design of Feedback Control Systems*", CBS College Publishing: N.Y., 1982.

[23] A. Soukaria, "*ADVISOR1, ADVISOR2, ADVISOR3 and MARPLAN Expert Documentation*", The University of Calgary: Calgary, 1987.

[24] P. Kotler and R.E. Turner, "*Marketing Management Analysis, Planning and Control*", Prentice-Hall Canada Inc: Scarborough, Ontario, 1985.

[25] M. H. B. McDonald, "*Marketing Plans*", William Heineman Ltd.: London, U.K., 1984.

[26] D. L. Flesher, T. K. Flesher and G. U. Sbelly, "*The New-Product Decision*", National Association of Accountants: N.Y., 1984.

[27] G. Wills, R. Hayhurst and D. Midgley, "*Creating and Marketing New Products*", Granada Publishing Limited: London, U.K., 1973.

[28] R. Carrison, "*Managerial Accounting*", Business Publications, Inc.: Plano, Texas, 1982.

[29] American Marketing Association, "*American Marketing Journal*", American Marketing Association: Chicago.

[30] American Marketing Association, "*Marketing News*", American Marketing Association: Chicago.

BIBLIOGRAPHY

[1] M. J. Combs, "*Developments in Expert Systems*", Academic Press: New York, N.Y., 1984.

[2] G. S. Kahn, S. Nowlan and J. McDermott, "Strategies for Knowledge Acquisition", *IEEE Pattern Recognition and Machine Intelligence,* September, 1985.

[3] J. H. Boose and B. R. Gains (Editors), *Proc. Knowledge Acquisition for Knowledge-Based Systems Workshop,* The American Association for Artificial Intelligence, Banff, 1986.

[4] A. Soukaria and M. H. Hamza, An Expert System for Laboratory Experiments in Control", *IASTED International Symposium on Expert Systems:* Geneva, Switzerland, 1987, pp. 84-90.

[5] P. S. Licker, "*How is an Expert System like consulting with an Expert*", University of Calgary, Faculty of Management: Calgary, 1984.

[6] P. H. Winston and K. A. Prendergart, "*The AI Business*", The Massachusets Institute of Technology: Mass., 1984.

[7] O. J. M. Smith, "*Feedback Control Systems*", McGraw-Hill Book Company Inc., York, P.A., 1958.