DEVELOPMENT OF MEDICAL EXPERT SYSTEMS WITH FUZZY CONCEPTS IN A PC ENVIRONMENT

A Thesis
presented to the Department of Computer Science
of the Chinese University of Hong Kong
in partial fulfillment of the requirements
for the Degree of Master of Philosophy

by
SO YUEN TAI
May 1990

ACKNOWLEDGEMENTS

I would like to acknowledge my supervisor Dr. K.S. Leung for his supervision and sound critiques of my work. I would also like to express my deepest gratitude to Dr. Ares K.L. Leung, a gynaecologist, for his medical knowledge support for the development of INDUCE36 and ESROM. His comments on the development of Z-III make it a success. Also, thanks are due to Dr. Felix Wong, another gynaecologist, for his medical knowledge support in developing ABVAB. Moreover, I would like to acknowledge Mr. Lam Wai, the designer of Z-II, for his advice and help. Lastly, I would like to thank Miss Joanne Leung for giving me the support throughout my M.Phil. programme.

TABLE OF CONTENTS

ABSTRACT

The diagnosis problem is becoming more difficult because the medical field is full of imprecision and uncertainty, and the amount of medical information available to physicians is also increasing. Efforts are made to apply the expert system techniques to problems in medical diagnosis which is proven to be successful. In order to facilitate better knowledge engineering in building medical expert systems, an expert system shell, code named as Z-III, which can handle both exact and inexact reasoning and includes some features such as weighting, fuzzy matching, threshold, database retrieval, has been developed in a micro-computer environment. Using Z-III, three medical expert systems, ABVAB, INDUCE36 and ESROM, in the field of obstetrics and gynaecology, incorporated with fuzzy concepts have been built successfully. The techniques used and the implementation details in knowledge engineering process are described. Checking inconsistency and incompleteness in a fuzzy environment are discussed and a new method is proposed. The implementation highlight, characteristics, and results of the three medical expert systems are summarized. It is concluded that the implementation of medical expert systems in a PC environment is not only feasible but also increases user-friendliness and availability of such expert systems.

Chapter 1. INTRODUCTION

1.1 Inexact Knowledge in Medical Expert Systems

Expert systems and expert system shells are widely studied subjects in the field of Artificial Intelligence. They have received great attention not only from computer professionals but also from various sectors of society. Medicine is the field which can employ expert system for diagnosis. Although there are a lot of developed medical expert systems, building a medical expert system, especially one which can handle inexact reasoning of human, still remains a topic for research and experiment.

Imprecision and uncertainty play an important role in the field of medicine. With the increasing volume of medical information available to physicians, the process of determining appropriate therapeutic actions becomes increasingly difficult. The knowledge concerning the symptom-disease relationship and the state of the patient constitutes the sources of imprecision and uncertainty [Klir and Folger 1988].

Classical mathematical models for medical diagnosis are very poor because of their negligence of some important information on the patients such as symptoms of past undiagnosed diseases which can only be vaguely recalled by the patient [Esogbue and Elder 1983]. Moreover, using these models, we always assume that a patient can manifest only one of the well-defined diseases, i.e. no partial or multiple disease presence. However, it is well known that patients can manifest several diseases simultaneously. In certain circumstances, the presence of several well-defined diseases is relatively common.

Expert systems which can represent inexact knowledge and perform plausible reasoning are suitable to be applied in medicine. There are two kinds of inexact knowledge. One is called fuzziness which means that the classes of objects are not sharply defined. The other is called uncertainty which means that one is not certain about a piece of information. Fuzzy logic introduced by Zadeh [Zadeh 1965] can provide a representation for the vagueness of real world objects. Certainty factor model used in MYCIN [Buchannan and Shortliffe 1984] is a good tool to model the uncertainty.

Many expert systems only address uncertainty and they use artificial categorization of concepts and relationships to avoid the fuzziness. However, the data input by a user must be represented by both fuzzy and uncertainty concepts in an event. Thus, a number of fuzzy expert system shells have been developed and many inference methods for fuzzy and uncertainty concepts have been built. The following two sections will describe several fuzzy expert system shells and medical expert systems.

1.2 Fuzzy Expert System Shells

There are many expert systems or expert system shells which incorporated with fuzzy concepts have been developed [Leung et al. 1988], [Dubois and Prade 1988], [Basu and Dutta 1986], [Fu et al. 1986], [Karwowski et al. 1987].

1.2.1 SPII-2 [Dubois and Prade 1988]

This shell is a general inference system able to deal with both imprecision and uncertainty pervading factual and expert knowledge. It works in backward chaining. Possibility theory is used for representing imprecision in terms of possibility distributions . For the rule "If X is A, then Y is B"

$$\text{let} \quad \pi_X = \mu_A \quad \text{and} \quad \pi_Y = \mu_B$$

$$\pi^*_{Y/X}(t,s) = 1 \quad \text{if} \quad \mu_A(s) \le \mu_B(t)$$

$$= 0 \quad \text{otherwise.}$$

π_X and π_Y are the possibility distributions of X and Y

μ_A and μ_B are the membership function of fuzzy sets A and B

$\pi^*_{Y/X}$ is the greatest conditional possibility of Y w.r.t. X

which is actually the Rg relation [Mizumoto et al. 1979].

Then π_Y, the possibility distribution of Y, is defined by

$$\pi_Y(t) = \sup_s \min(\pi_X(s), \pi_{Y/X}(t,s))$$

Uncertainties of a fact and a rule are represented by a pair of possibility and necessity measures. Uncertainty and imprecision in the reasoning process is propagated via deductive inferences. If more than one rule of which the condition parts are partially matched by the facts, the possibility and necessity measures are used to find out the most appropriate rule.

Weights of fact and rule are not considered in this shell and since only Rg relation has been implemented, it may not suit some problem domains.

1.2.2 Fuzzy Expert System Shell for Decision Support System [Basu and Dutta 1986]

A fuzzy expert system shell was developed by Information System Department of Business and Management, University of Maryland, College Park. All knowledge is represented in the form of well-formed formulae (wff) and objects which contain a number of relevant attributes and properties.

For each statement, there is a assigned μ-level which is a measure of its 'distance' from the corresponding precise statement. The logical operator such as AND/OR can also be fuzzy and the μ-level of AND operator is calculated as follows:

$$\mu_{\omega_x X \wedge_k \omega_y Y}(u, v) = 1 - \left[\frac{\omega_x^k (1 - \mu_x(u))^k + \omega_y^k (1 - \mu_y(v))^k}{\omega_x^k + \omega_y^k} \right]^{\frac{1}{k}}$$

where ω_x and ω_y are the relative weights of the concepts X and Y.

$\mu_X(u)$ and $\mu_Y(v)$ are the values of the membership functions of X and Y for the elements u and v respectively.

k is a constant from one to infinity.

The calculation for fuzzy OR combination is similar to that of AND operator. Each term in the above equation has a weight attached to it. This weight shows that not all the factors are equally important. The value of k acts as an important parameter. If the expert is fairly confident about the relative weights, a high k-value can be used, denoting relatively certain operators and vice versa. Thus, both fuzzy and uncertain knowledge can be represented in the same form by appropriate use of the parameter k.

Since most rules used in the system are deductive, the μ-value of the conclusion is simply calculated by the S-form or K-form functional forms. Appendix I will show the equations of the S-form and K-form.

Only goal-directed reasoning can be considered in this system. The problems (goals) are viewed as theorems in logic, that have to be proved. Partial matches are also accepted in resolution refutation. During the proving process, a reasoning tree is built with weight assigned to each node and arc. One of the features of this system is the use of lower bound value z (threshold) for each subproblem, so that candidate solutions can be evaluated based on this bound, and only those solutions with $\mu \geq z$ are accepted.

Although the formula stated above is very feasible, computation is excessive for the inference process.

1.3 Medical Expert Systems

There are many developed medical expert systems [Buchannan and Shortliffe 1984], [Koyama 1987], [Kulikowski 1983], [Esogbue 1983], [Buisson and Farreny 1985], [Buckley and Tucker 1987]. Fuzzy concept is used in the implementation of some of them.

1.3.1 EXPERT [Kulikowski 1983]

EXPERT is a shell for building medical expert system and is being used extensively in the development of several medical consultation models such as rheumatology. The consultation model consists of findings and hypotheses. Findings are facts or input data. e.g. patient's history, symptoms, signs and laboratory results. Hypotheses are conclusions inferred by the system which include diagnostic and prognostic decision categories, therapy recommendation etc. A certainty factor, from -1 to 1, is associated with each hypothesis. Multi-layer of medical expert system can be achieved by using the three types of rules provided in EXPERT. They are :

1. finding-to-finding rules (FF)

   specify truth values of findings that can be directly deduced from an already established finding.

2. finding-to-hypothesis rules (FH)

   are logical combinations of findings that indicate confidence in the confirmation or denial of hypotheses.

3. hypothesis-to-hypothesis rules (HH)

   allow the model builder to specify inferences among hypotheses and treatment selections that follow from other diagnostic and prognostic hypotheses.

Figure 1.1 illustrates the various uses of different types of EXPERT rules so that multi-layer of consultation could be resulted.

EXPERT is successfully transferred to M68000 microprocessor-based system. Thus it will greatly increase the usefulness of EXPERT. However, only uncertainty of medical data has been considered, fuzziness has not been handled.

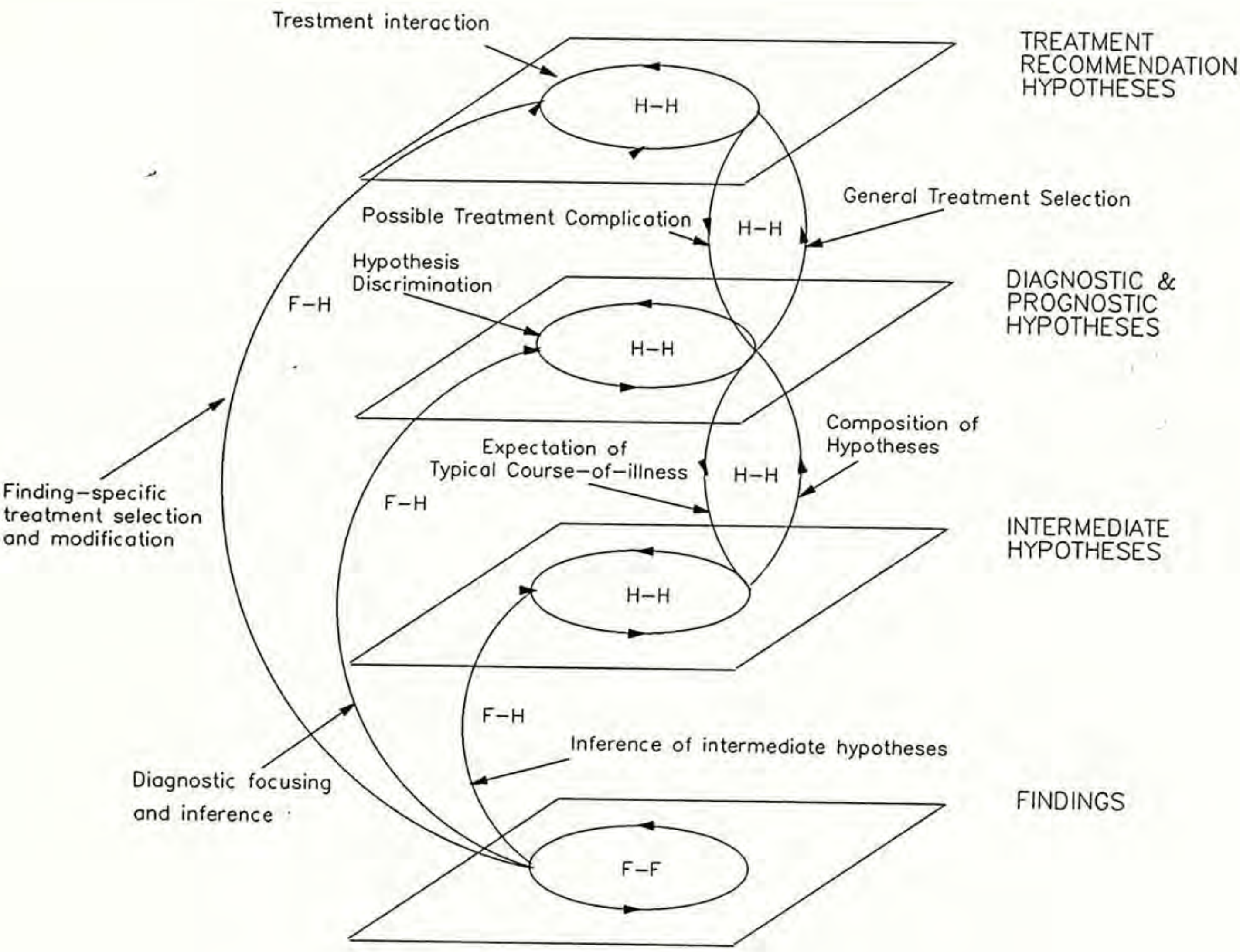

Figure 1.1 Various Uses for Different Types of Expert Rules

1.3.2 DIABETO [Buisson and Farreny 1985]

DIABETO is an expert system used as a decision-aid tool for the treatment of diabetes. The system concerns with management of imprecise or uncertain medical knowledge based on the theory of possibility.

Medical knowledge is described in terms of themes (goals), notions (facts), deduction rules and control rules. The control rules, which are used in forward chaining, organize the steps of the development of themes. Fuzzy variables are used to describe the vagueness of medical knowledge. If the value for the fuzzy variable is also uncertainty, DIABETO uses a small, non-zero degree of possibility, ϵ, to solve the case by taking the possibility measure, Π_X be :

For the fact : X is A, CF = 1 - ϵ

$$\Pi_X(u) = \max(\mu_A(u), \epsilon)$$

Figure 1.2 shows the graph of the above equation.

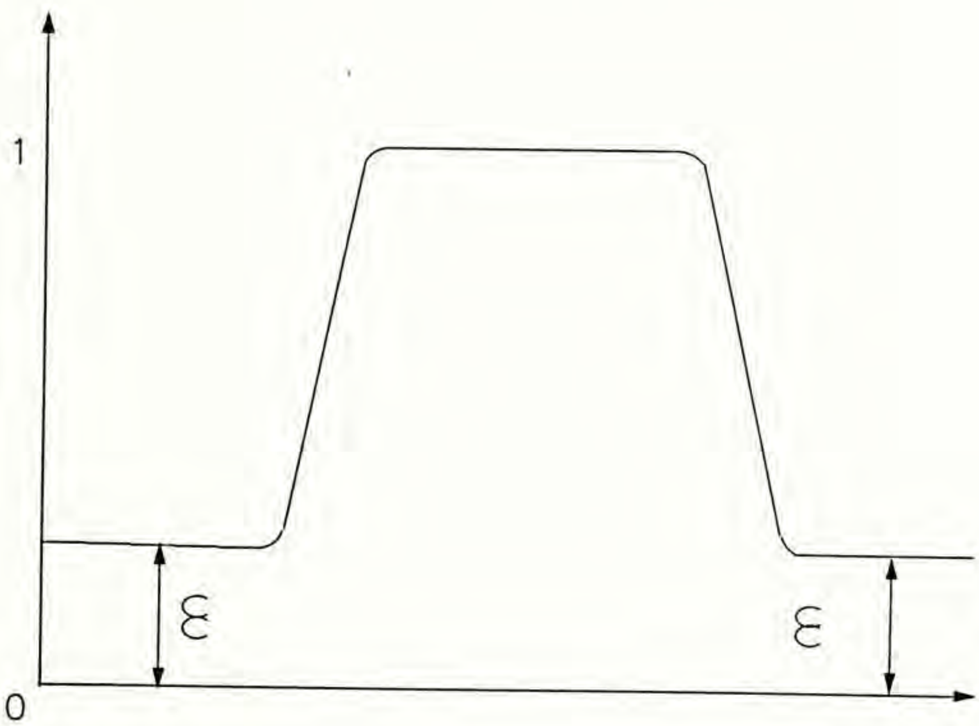

Figure 1.2

This method offers a simple and unified framework for dealing with both imprecision and uncertainty. Only fuzzy relation Rg is implemented in DIABETO for fuzzy inference (Reference to Appendix II to see the types of reasoning provided by Rg). For multiple rules having the same conclusion, the minimum μ-value will be taken as the conclusion's μ-value. This will be simple and fast in execution. No weight has been added to each notion in the rules.

Most of the above mentioned expert systems have not handled the vagueness of human knowledge and/ or are purposely built without a shell. These lead to the design and implementation of the fuzzy expert system shells of this project.

1.4 Impact from Micro-computer

Many medical expert systems are only available on mainframe or mini-computer. The successfully built medical expert systems like MYCIN (which gives the diagnosis of bacterial infections and prescribing treatments), CASNET (for the glaucomas), PIP (for renal diseases), and INTERNIST (for internal medicine) are built on big machines. In spite of the satisfactory results given by these systems, the accessibility of these system is poor.

One of the most far-reaching developments in the computer industry was the introduction of micro-computers (personal computers, PC) which made it possible to extend computer power into many unprecedented areas. Nowadays, a PC is so common, low-cost, and popular that almost everyone could possess one. As the internal and external storage capacity and computing power of PC have been rapidly increasing in recent years, a medical expert system is now possible to be built in a PC environment. Moreover, many programming languages available in PC, like C, LISP or PROLOG could be used to implement the medical expert systems.

A fuzzy expert shell in a PC environment called Z-III is developed and used to build three medical expert systems, code named as ABVAB, INDUCE36, and ESROM on obstetrics and gynaecology. Z-III is built based on the design of a novel fuzzy expert shell Z-II [Leung et al. 1988] with many new features added to enhance the knowledge acquisition and refinement processes.

1.5 Approach

In the initial phase of this research, fuzzy set theory and fuzzy logic are investigated. The characteristics of some fuzzy expert system shells and medical expert systems are examined. A survey on the methodologies used to handle inexact knowledge in Z-II is studied and is given in chapter 2. Also, the limitations of and improvements on Z-II are discussed.

Then the design and implementation of a fuzzy expert system shell Z-III in a PC environment are carried out. One of the objectives in developing Z-III is to increase the accessibility of Z-II. New features are implemented in Z-III in order to make it more suitable to handle the inexact knowledge and the large volume of data in medicine. Chapter 3 covers a detailed description of Z-III.

Building a medical expert system is an iterative process, and the size of a medical system is very large. The bottleneck of building expert system is the knowledge acquisition process. Systematic and efficient knowledge acquisition is the key to success. Thus, the techniques used in knowledge acquisition should be studied. After the study, the newly established techniques are used to construct three medical expert systems. Experience is accumulated in the development process. Moreover, the inconsistency and incompleteness easily occur in a knowledge-based system. Some of them only waste the storage or slow down the speed of consultation, but others may cause serious errors. A study of inconsistency and incompleteness of a nonfuzzy rule-based environment is carried out and a proposed method to determine the existence of inconsistency in fuzzy environment is given. Chapter 4 will describe the details in knowledge engineering process and consistency and completeness checks.

Three medical expert systems are then built using Z-III and each of them has its own characteristics respectively. They are described in details in chapter 5. The three medical expert systems are :

ABVAB diagnoses the cause of abnormal vaginal bleeding from the past history and the results of physical examinations of a patient.

INDUCE36 involves decision on whether labour should be induced for individual patients after 36 weeks of gestation.

ESROM deals with the problems of rupture of membranes.

Finally, a conclusion is given in chapter 6.

Chapter 2. SYSTEM Z-II

2.1 General Description

SYSTEM Z-II was a successfully built expert system shell developed by the Computer Science Department, the Chinese University of Hong Kong. Z-II can deal with both exact and inexact reasoning. It is a rule-based expert system shell which employs fuzzy logic, fuzzy comparison and certainty factor for its reasoning to handle the two basic inexact concepts: fuzziness and uncertainty. Any combination of fuzzy or non-fuzzy propositions in the antecedent part of a rule is allowed. It uses evidence combination for those cases with more than one rule having the same consequent proposition.

The inference engine is based on backward reasoning with forward evaluation of the values of the fuzzy terms. Explanation facilities such as WHY and HOW and review function likes WHAT-IF are supported. The programming languages chosen to implement Z-II were LISP and Pascal. LISP was used to develop the main parts of Z-II because in LISP, hash tables and property lists, both with transparent hashing functions, are provided and these features are suitable for implementing knowledge base system. Pascal was used for the fuzzy set operations in the inference module because of its execution efficiency. The programming environment of Z-II is VAX-11/780 under VMS version 4.6 which facilitates linking between languages. VAX-LISP version 2.1 and VAX-PASCAL version 3.4 are used in the implementation.

Z-II was proved to be suitable to build expert systems in many application areas such as classification, risk analysis and information retrieval [Leung et al. 1989]. ABVAB, one of the successful medical expert systems, was built on Z-II [Leung et al. 1988] [Wong et al. 1990].

2.2 Main Features

The power of Z-II is based on its ability to handle both exact and inexact reasoning. The following sections will describe the fuzzy concepts, fuzzy certainty and other theories or concepts used in Z-II. For further details, please refer to [Lam 1988], [Leung and Lam 1988], [Leung and Lam 1989], [Leung et al. 1989].

2.2.1 Fuzzy Concepts

In Z-II, fuzzy concepts are allowed in both facts and rules. Fuzzy concepts are denoted by fuzzy expressions which are in the form of natural language. Each basic fuzzy concept is associated with three basic linguistic fuzzy terms describing the upper, medium and lower level situations of this concept. For example, the fuzzy concept 'big' is associated with the basic linguistic fuzzy terms exemplified by "big", "normal" and "small". Then, fuzzy expressions like "very big" and "normal to rather small" could be constructed.

Each basic fuzzy term is represented by a fuzzy set which is effectively represented by a list of eleven numbers which are grades of membership of the points on an imaginary psychological continuum with an interval scale. The default fuzzy sets of "big", "normal" and "small" in Z-II are as follows :

"big" : [0 0 0 0 0 0 0.0017 0.0984 0.6637 1]
"normal" : [0 0 0 0.0035 0.3619 1 0.3619 0.0035 0 0 0]
"small" : [1 0.6637 0.0984 0.0017 0 0 0 0 0 0]

When hedge is used in the fuzzy expression, like "very big", the ordinary fuzzy set is modified by a function [Zadeh 1972]. For example, the square function (x^2) is used for the hedge 'very'. i.e.

very x = x^2

And the fuzzy set of "very big" is as follow :
[0 0 0 0 0 0 0 0.0097 0.4405 1]

2.2.2 Fuzzy Certainty

In Z-II, uncertainties in the facts and rules are basically handled by certainty factor model similar to the one adopted in MYCIN [Buchanan and Shortliffe 1984]. Fuzziness and imprecision are allowed in the uncertainty exemplified by "about 0.6" or "around 0.5 to 0.8".

Fuzzy numbers are employed to model the fuzzy certainty. A fuzzy number is actually a real number fuzzy set that is both convex and normal. The definition of convex and normal are given below :

A fuzzy set is convex if and only if

$$\forall x, y \in R : \mu_F[\lambda x + (1-\lambda)y] \geq \mu_F(x) \wedge \mu_F(y), \forall \lambda \in [0,1]$$

where R is the set of real number,
x, y and λ are real numbers
$\mu(x)$ is the membership function of x in the universe of discourse.

A fuzzy set is normal if and only if the highest value of the degree of membership is equal to 1.0

In Z-II, fuzzy numbers are assumed generally to be trapezoidal and they are implemented as a list of four numbers. In general, a fuzzy number is represented as (LOWER, PEAK1, PEAK2, UPPER) as in Figure 2.1.

PEAK1 and PEAK2 denote the interval in which the membership degrees are equal to 1. LOWER and UPPER denote the interval in which the membership degrees are non-zero.

The modifiers in fuzzy certainty factors are associated with numerical values specifying the degree of vagueness determined by the distance between the upper or lower bounds and the peak values. For example, the degree of vagueness of 'about' and 'around' are 0.2 and 0.15 respectively [Lam 1988], and the fuzzy number representation of 'about 0.5 to around 0.8' is [0.3 0.5 0.8 0.95].

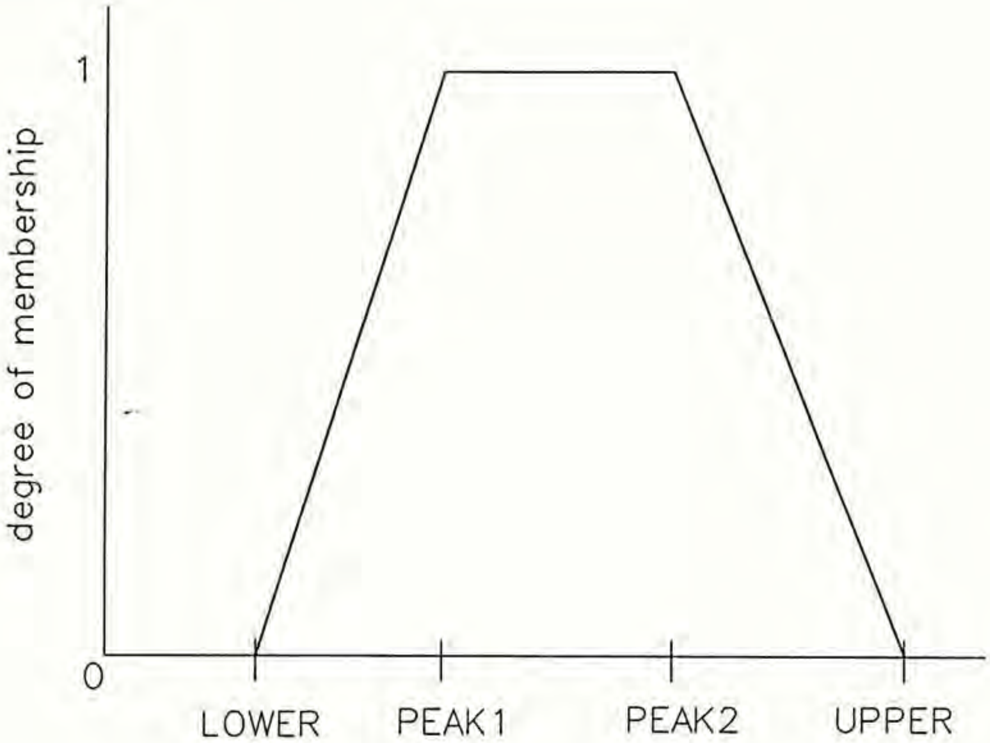

Figure 2.1 A fuzzy number in Z-II

2.2.3 Fuzzy Comparison

Fuzzy comparison can be viewed as a special case of fuzzy sets operations which uses numerical values. The general form of a fuzzy comparison is :

<Object> Operator <Value1> [, <Value2>] [ ] is optional.

E.g. Temperature > 15, 18

When fuzzy comparison is involved, a value between 0 and 1 denoting the degree of matching is returned [Leung and Lam 1989]. There are totally eight types of fuzzy comparisons.

E.g. IF gestation > = 36, 38 THEN ...

This condition will be satisfied if the value of gestation is greater than or equal to 36 and the degree of matching returned is as shown in Figure 2.2.

E.g. IF increase in white cell count < = 10, 15 THEN ...

This condition will be satisfied if the value of increase in white cell count is less than or equal to 15. The degree of matching returned is shown in Figure 2.3.

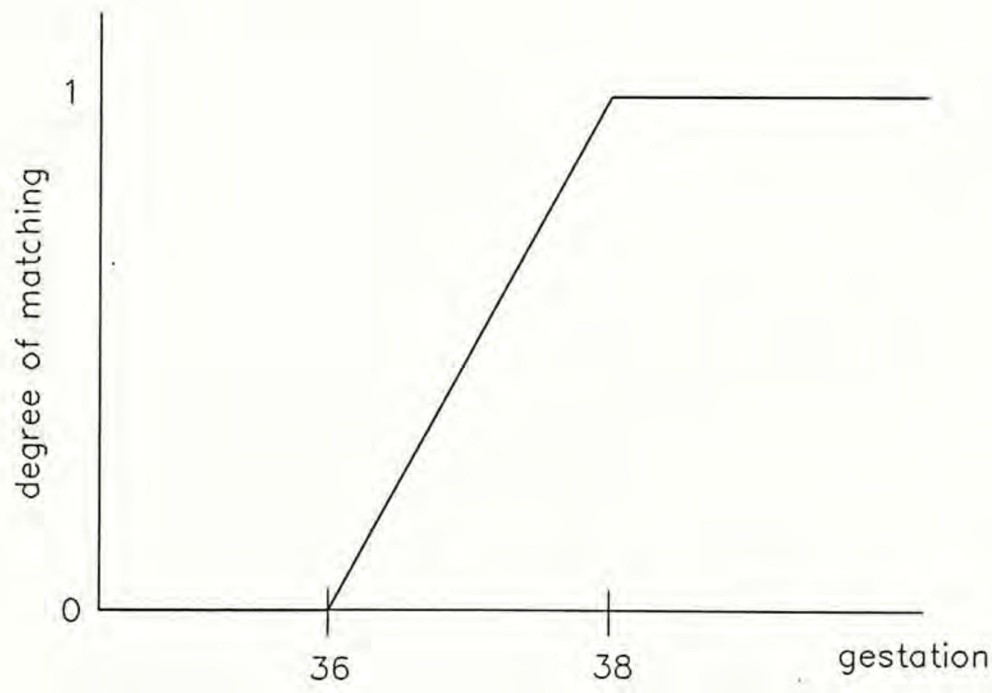

Figure 2.2 Fuzzy comparison "> ="

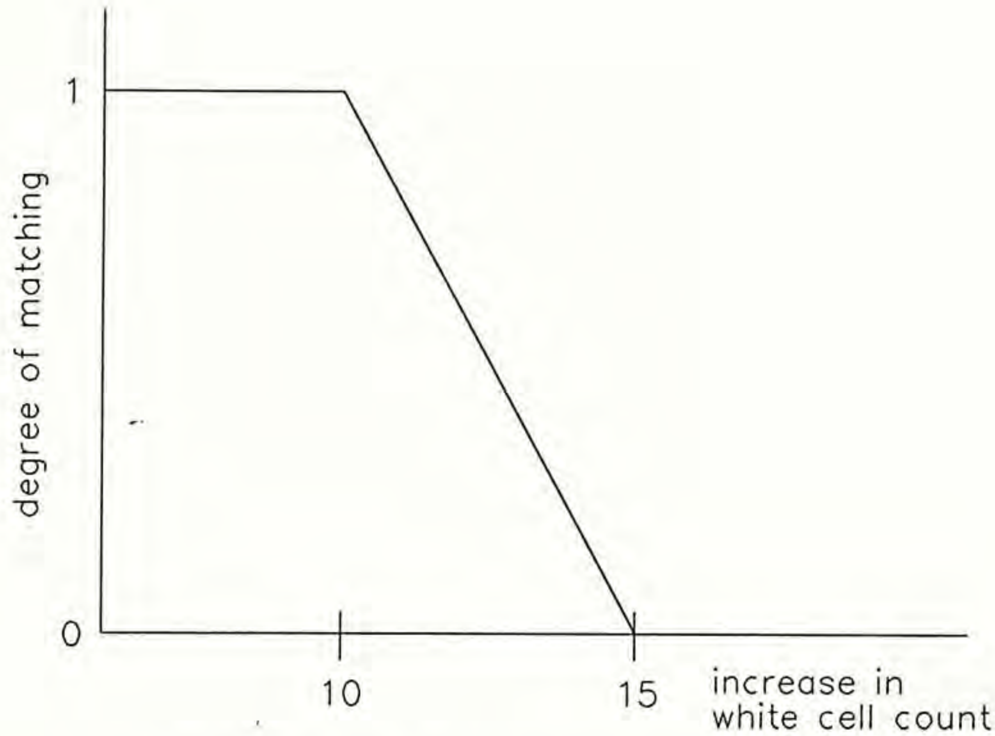

Figure 2.3 Fuzzy comparison "< ="

2.2.4 Rule Evaluation

Consider the following rule and fact:

Rule	:	IF A is V_1 then C is V_2 (certainty = FN_1)
Fact	:	A is V_1' (certainty = FN_2)
Conclusion	:	C is V_2' (certainty = FN_3)

If A is non-fuzzy, V_1 and V_1' must be the same atomic symbol in order to trigger this rule. So, V_2' must be equal to V_2. The certainty of conclusion (FN_3) could be calculated by fuzzy number multiplication of FN_1 and FN_2. i.e. $FN_3 = FN_1 * FN_2$.

If both A and C are fuzzy, a fuzzy relation R is formed by taking some fuzzy operations on the fuzzy sets of V_1 and V_2. Rs, Rg and Rsg are the fuzzy relations implemented on Z-II which can simulate the inference processes of human [Mizumoto 1979]. The choice of these three relations depend on the nature of problems and the inference process. Appendix II will give a brief description of the difference between them. The fuzzy set of V'_2 in conclusion is obtained by applying a fuzzy composition operation to the fuzzy set of V_1 on the fuzzy relation. The certainty of conclusion is the same as above.

When A is fuzzy but C is nonfuzzy, V'_2 is equal V_2. However, the certainty of the conclusion is calculated by the fuzzy multiplication of FN_1, FN_2 and similarity M between V_1 and V'_1. i.e. $FN_3 = FN_1 * FN_2 * M$. For the calculation of similarity M, please refer to section 3.4.2.1.

For multiple propositions in the antecedent part of a rule, the propositions can be connected by using the logical operators AND/OR.

Consider the rule : IF A AND B THEN ....

The fuzzy set of the antecedent part is equal to the fuzzy union of the fuzzy sets of A and B. IF OR is used instead of AND, the result is calculated by taking fuzzy intersection of the two fuzzy sets.

The calculation of the uncertainty of the conclusion in multiple proposition is similar to that in MYCIN's CF model. e.g.

Rule	:	IF A_1 and A_2 then C	(certainty = FN_R)
Fact	:	$A_1^{'}$	(certainty = FN_1)
		$A_2^{'}$	(certainty = FN_2)

Conclusion	:	$C^{'}$	(certainty = FN_C)

$$FN_C = (\min\_fn(FN_1, FN_2) * FN_R)$$

where A_1 and A_2 can be either single or multiple propositions, and min_fn is taking the minimum of two fuzzy numbers. If OR is used instead of AND, the calculation is same as above except that maximum function is used instead of minimum.

2.2.5 Certainty Factor Propagation

In many cases, there are more than one rule with the same conclusion fired in a same consultation. Z-II performs the evidence combination similar to that in MYCIN. The formula are as follows :

$$FN_R = FN_1 + FN_2 - (FN_1 * FN_2) \qquad \text{if both } CG(FN_1) \text{ and } CG(FN_2) > 0$$

$$FN_R = \frac{FN_1 + FN_2}{1 - \min\_fn(|FN_1|, |FN_2|)}$$ if only one of CG(FN_1) and CG(FN_2) < 0

$$FN_R = FN_1 + FN_2 + (FN_1 * FN_2)$$ if both CG(FN_1) and CG(FN_2) < 0

Where

FN_R	:	the combined fuzzy uncertainty
FN_1, FN_2	:	fuzzy uncertainties of the propositions
+	:	fuzzy number addition
-	:	fuzzy number subtraction
*	:	fuzzy number multiplication
min_fn	:	taking minimum of fuzzy numbers
|FN|	:	taking the absolute of a fuzzy number
CG(FN)	:	center of gravity of a fuzzy number

2.2.6 Linguistic Approximation

Linguistic approximation is a process which maps the fuzzy sets onto a set of linguistic expressions or values [Dubois and Prade 1980]. By using linguistic approximation, the output can be represented in a natural language form.

In Z-II, linguistic approximation is needed in two phases :

i, to find the corresponding verbal descriptions of fuzzy sets representing fuzzy values. E.g.

[0 0 0 0 0 0 0 0.0097 0.4405 1] transforms to 'very good' for the fuzzy concept 'good'.

ii, to get the linguistic descriptions of fuzzy numbers representing fuzzy certainty factors. E.g.

[0.3 0.5 0.8 0.95] transforms to 'about 0.5 to around 0.8'.

For both cases, the technique adopted in finding appropriate linguistic values is similar to Wenstop's approach [Wenstop 1980]. Two parameters, namely, "imprecision" and "location" are used to identify the corresponding linguistic expression. By using linguistic approximation, Z-II could have a better user-interface.

2.3 Limitations and Possible Improvements

Although Z-II is a novel fuzzy expert system building tool, it also has its limitations. Thus, it is suggested that further improvements are needed in order to enhance the power of Z-II and facilitate the knowledge acquisition and knowledge refinement processes. The limitations and the suggested enhancements are as follows :

i, Weighting of Propositions in a Rule

In Z-II, the antecedence of a rule can consist of multiple propositions and each proposition contributes the same weighting to the final conclusion. However, in real cases, it is found that human experts believe that some propositions are more or less important than the others, i.e. the propositions may have different degrees of importance. Adding relative weights to the propositions will indicate the degrees of importance to the conclusion. The inclusion of weights will modify the formulae used in the calculation of certainty factor in rule evaluation.

ii, Inference Mechanism to Trigger the Rule in Fuzzy Environment

Let us consider the followings :

r1 : IF your height is tall
THEN your dress is blue (CF = 1)

r2 : IF your height is short
THEN your dress is yellow (CF = 1)

where 'height', with values (tall average short), is a fuzzy object. The object 'dress', with values (blue yellow), is a non-fuzzy single-valued object.

For the facts : your height is tall (CF = 1)
The Conclusion is : your dress is <u>not yellow</u> (CF = 1)

Z-II will not give the conclusion 'your dress is <u>blue</u>' because 'height' is a fuzzy object and Z-II triggers r2 instead of r1 (because of the order of the rules).

As Z-II performs its fuzzy inference by using the fuzzy composition function on the fuzzy relation matrix (either Rs, Rg or Rsg), it will make Z-II fire all the propositions containing that fuzzy object. As the above case, 'dress' is a single-valued object and after Z-II triggers the rule r2, it will no longer pay attention to r1. Thus, the above anomalous conclusion is resulted.

As a result, a better mechanism should be employed to trigger rules in a fuzzy environment. Besides, a new approach has also to be used to handle the instantiation of object in order for a single-valued object having multiple values.

iii, Thresholds

In MYCIN [Buchannan and Shortliffe 1984], thresholds are added to the system. Then, the weak evidence, with certainty factor less than a threshold, will not trigger the inference mechanism. This will result in pruning the reasoning tree and increasing the consultation efficiency. Moreover, removing the weak evidence will not lower the certainty factor of the conclusion by any significant amount. Therefore, it is a good practice to include threshold in Z-II.

iv, Limited Consistency Check

Frequently, rules provided by the human experts are inconsistent. The consistency check facilities significantly help in the debugging of knowledge base. However, Z-II has not provided any facility to assist the knowledge engineer to locate the inconsistencies, especially in a fuzzy environment.

Chapter 3. A FUZZY EXPERT SYSTEM SHELL (Z-III) IN PC ENVIRONMENT

3.1 General Description

From last chapter, it can be concluded that Z-II is a robust expert system building tool. However, in order to make it become a more powerful one, further improvements and enhancements are necessary.

A new expert system shell, coded name as Z-III, is developed which consists of almost all the features in Z-II plus some new and potent features. One of the advancement of Z-III is that it is built in a PC environment. Nowadays, PC is so common, low-cost, and popular that almost everyone can possess one. As the internal and external storage capacity and computing power of PC have been rapidly increasing in recent years, and many programming languages are available in PC, it is possible to develop a powerful fuzzy expert system shell in a PC environment. The following sections will give a detailed description on the features and development of Z-III.

3.2 Programming Environment

Z-II was developed using the programming languages LISP and Pascal. LISP seems to be a very suitable language. However, it is found that the execution time of LISP is too slow. The implementation language of Z-III is Turbo C version 2.0. C is chosen because of its high efficiency and portability. Also, the data structure provided by C could handle the hash table and function quite easily. The rationales for not using LISP as compared to Z-II are given below :

a) Efficiency and Performance

Interactive LISP environment is usually memory intensive and requires more CPU power than traditional languages. Most of the micro-computers cannot comfortably support it. Although the LISP source programs can be compiled to executable programs, the running time still make one feel tiresome. Besides, LISP interpreter needs annoying garbage collection time for LISP which the user has to get accustomed. It is rather disruptive especially during an on-line consultation. In contrast, C was purposely designed as an efficient programming language, optimized for run-time performance.

b) Numeric Computation

Numerical computation power is becoming more important in expert systems. Z-III involves a lot of operations on the fuzzy sets and fuzzy numbers during inference. LISP does not possess any advantage in this aspect. On the other hand, programs in C are compiled and optimized, implying a higher computational power.

c) Portability

C is a highly portable language, i.e. a shell built in C can be ported easily to other machines with minor modifications on machine-dependent routines, such as the user-interface functions.

The development environment of Z-III is on a 80386 machine under PC-DOS version 3.3. There is no special hardware requirement. Z-III can be run on IBM PC/XT/AT/386 or compatible machines under PC/MS-DOS version 2.0 or above. However, for better performance, it is suggested that the system should be used on an 80286 or 80386 machine, with a hard disk installed. The minimum memory needed is 640 Kbytes which can allow about 400 rules and 300 objects in the knowledge base making Z-III suitable for building small to medium size applications.

3.3 Main Features and Structure

Z-III is a structured and modular rule-based expert system shell incorporated with fuzzy concepts to handle both exact and inexact reasoning. It provides an integrated and user-friendly environment for the development of expert systems. It is a menu-driven, easy-to-learn and easy-to-use system.

Fuzzy concepts, denoted by fuzzy expressions, are represented by fuzzy sets. Fuzzy numbers are used to store the fuzzy certainty factors which are between -1 and 1. Negative certainty factors reflect the degree of disconfirmation. Fuzzy comparison and linguistic approximation have also been implemented. The methods of rule evaluation and certainty factor propagation are similar to those in Z-II.

Z-III is a backward chaining system and consists of three main modules : Knowledge Acquisition Module, Consultation Module and System Properties

Management Module. There are three basic static constructs supported by Z-III : objects, rules and fuzzy terms. The structure of Z-III is shown in Figure 3.1.

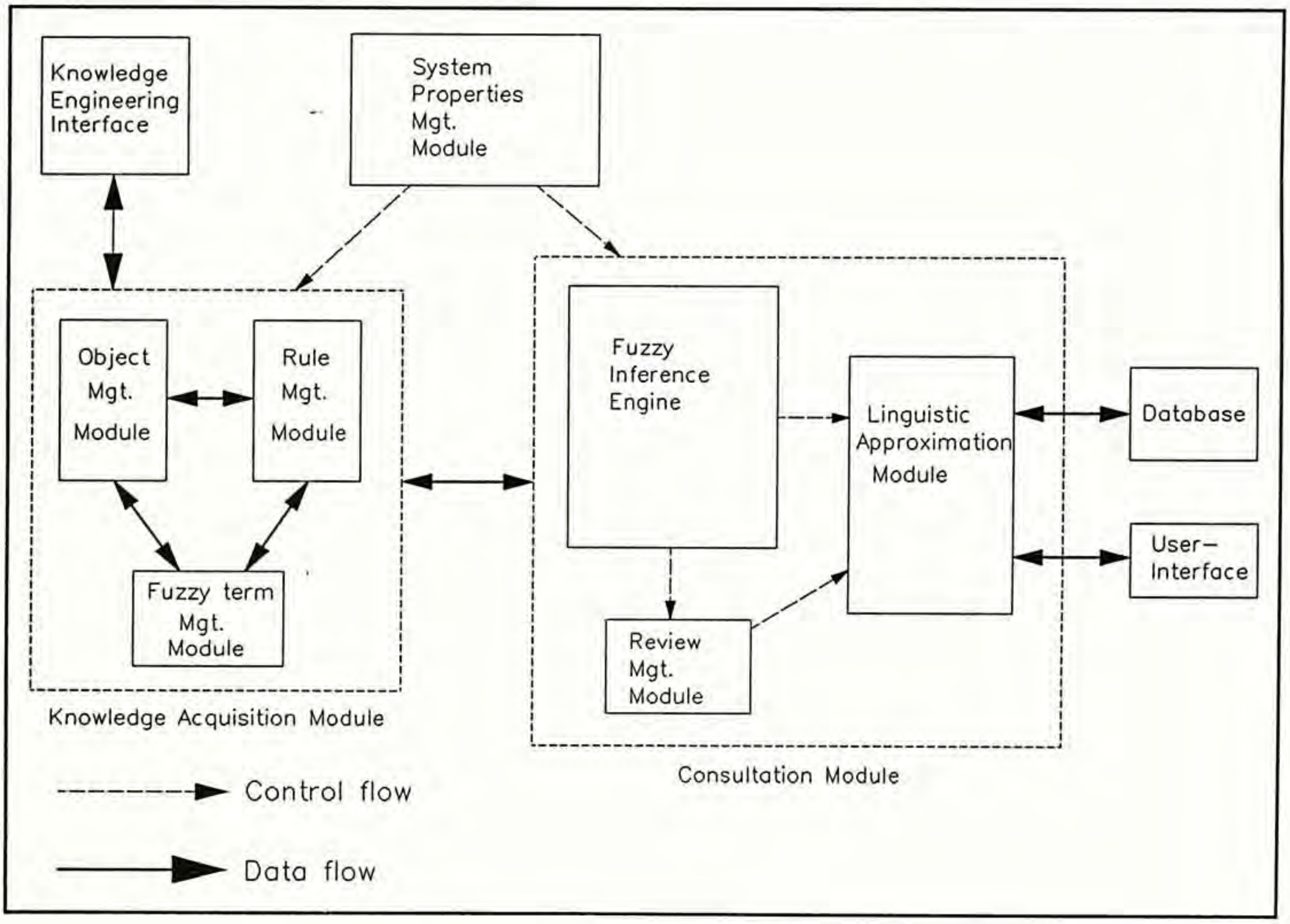

Figure 3.1 Structure of Z-III

3.3.1 Knowledge Acquisition Module

The Knowledge Acquisition Module mainly consists of three sub-modules : Object Management Module, Rule Management Module and Fuzzy Term Management Module.

3.3.1.1 Object Management Module

In Z-III, there are six type of objects :

a) Fuzzy
b) Single-valued
c) Multi-valued
d) Yes/No
e) Numeric
f) Numeric Variable

Fuzzy objects have fuzzy sets as their values (see section 3.3.1.3). The values of single- and multi-valued objects can be enumerated as a finite list of pre-defined elements. Single-valued object can take only one value at a time while multi-valued objects can have several values simultaneously. Yes/No objects are boolean object which can have the value YES or NO only. Numeric objects take the numeric values, either integer or real. Fuzzy comparison is allowed in numeric object. Numeric variable objects are used in a rule and their values are obtained from the user during consultation (see section 3.4.7).

3.3.1.2 Rule Management Module

In Z-III, rules are handled in a flexible way. They are stored in ASCII format and users can edit any rule by using a text editor. An external full-screen editor is also incorporated in Z-III for the user to edit the rules.

The format of a rule in Z-III is as follows :

```
(Rule <rule code>
IF (Ante_part)
THEN Conseq_part)
Certainty is <Certainty factor>
where

             Ante_part ::= Proposition | Proposition_list
           Conseq_part ::= Object Operator Value
      Proposition_list ::= Proposition Connective Proposition_list
           Proposition ::= Object Operator Value  Weight Threshold_expression
              Operator ::= IS | IS NOT | NumOperator
           NumOperator ::= > | >= | < | <= | = | <>
            Connective ::= AND | OR
                Weight ::= [Fuzzy Weight] | nil
  Threshold_expression ::= {CF NumOperator Num} | nil
                   Num ::= numeric value | numeric value, numeric value
                Object ::= any defined object
                 Value ::= any predefined value
```

The antecedent part of a rule may consist of multiple propositions connected by AND/OR. For each antecedent proposition, the expert could assign a weight and/or a threshold expression to its object. These two features will enhance and strengthen the representation power of Z-III in human knowledge. For a detailed discussion, please refer to sections 3.4.1 and 3.4.4.

Each rule has a certainty factor attached. The certainty factor may be fuzzy or crisp. If the certainty factor is omitted, absolute certain is assumed (i.e. Certainty is 1).

3.3.1.3 Fuzzy Term Management Module

The fuzzy construct in Z-III is called the fuzzy type. As we have seen, a certain linguistic value such as *high* can be applied to different linguistic variables such as height, employment rate, temperature, and amount of rainfall, etc. Therefore the meaning of *high* depends on the context in which it is used. In Z-II, the fuzzy concept *high* could be used in either one of them. It is because Z-II associate each linguistic term with only one fuzzy concept. To solve the problem of sharing adjective among these linguistic variables, a fuzzy type is created for each context of high. For the purpose of linguistic approximation, three fuzzy sets that correspond to the upper, medium and lower level situations of a fuzzy concept must be declared for each fuzzy type. In Table 3.1, although both the fuzzy types 'temp' and 'rate' contain the same terms high, medium and low, their representation sets and meanings can be completely different.

Linguistic variables	Fuzzy type	Fuzzy sets
temperature	temp	high medium low
unemployment rate	rate	high medium low

Table 3.1

Once an object is declared to be of a certain fuzzy type, it can take a value equivalent to any fuzzy sets under the fuzzy type. A vector of length eleven is used as the internal representation of a fuzzy set. The values of each eleven members are assigned, either default or user input, at the time when the fuzzy type is created. Therefore, although the same linguistic term is used for two fuzzy concepts, by using fuzzy type, the two linguistic terms can have different meaning.

3.3.2 Consultation Module

The fuzzy inference engine, review management module and linguistic approximation module form the Consultation Module.

3.3.2.1 Fuzzy Inference Engine

As Z-III is a production rule system, the antecedence of each rule is stored in the memory as a AND-OR tree. An example of the structure of a AND-OR tree used in Z-III is shown in Figure 3.2.

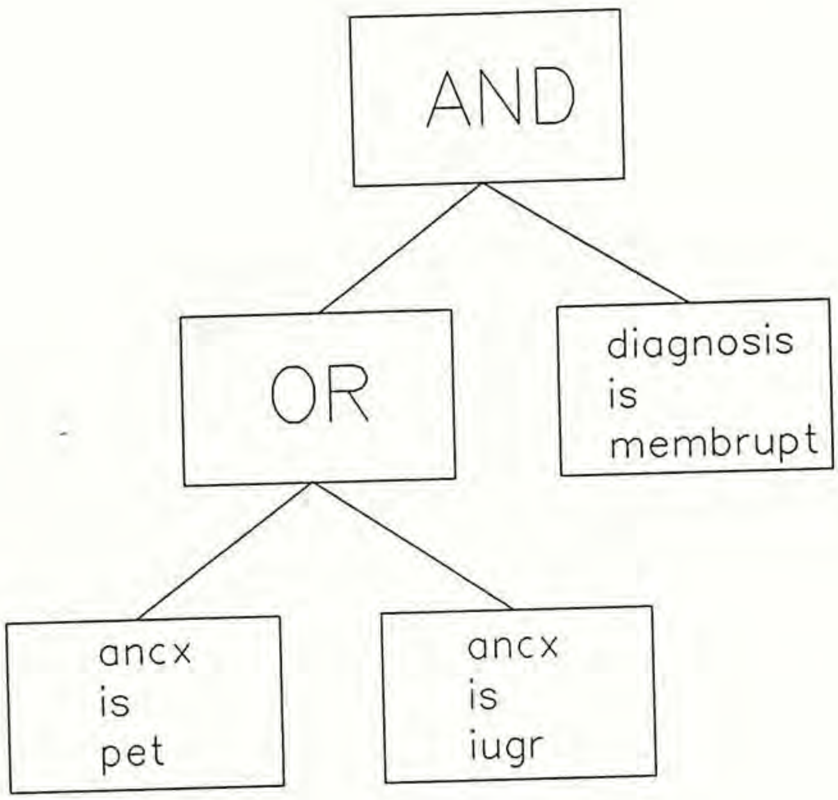

Figure 3.2 AND-OR tree

The above AND-OR tree is equivalent to the antecedence of the following rule in ESROM (see section 5.3):

```
(Rule m15
IF ( (ancx is pet) OR (ancx is iugr) )
   AND (diagnosis is membrupt) )
THEN management is delivery)
Certainty is 0.3
```

where ancx means "antenatal complication",

diagnosis means "diagnosis of rupture of membranes", and

management means "management of the patient".

Thus, consultation is just the process of traversal of all the AND-OR trees relating to the current goal and performing the evidence combination of the matched facts and rules. This approach is logically sound but places severe demands on the consistency of the rule base, since combining contradicting values will result in meaningless or even incorrect evaluations.

The backward chaining process in Z-III defines a depth first, left-to-right traversal of the tree. e.g. consider the rule :

IF A AND B AND C THEN D

The evaluation of the rule will activate the evaluation of the subgoals A, B and C, in the list order. The rule evaluation and evidence combination processes employed in Z-III is similar to those in Z-II.

Naturally, for the evaluation of a proposition, there arises a question that what constitute a successful evaluation, especially in a fuzzy environment. For matching in fuzzy environment, fuzzy matching strategy is applied (see section 3.4.2).

Moreover, Z-III also supports the explanation facilities such as WHY and HOW. The WHY option is provided for the user to question Z-III why it needs the value of a particular object during consultation, and the How option answers question as to how a certain value has been obtained for a given conclusion.

3.3.2.2 Review Management Module

There are two types of review implemented in Z-III. The first one is Fact Review by which all the currently known facts in the system can be displayed. Thus, the user can review all the facts entered and see whether the conclusion is acceptable.

Another review function is WHAT-IF Review. Sometimes, a user may wish to change some of the facts in the system to test its sensitivity to slight modifications of the problem state or to find out what might happen should changes occur in the future. WHAT-IF Review supports this type of facility for the user. The user can choose the facts to change and after editing the facts, Z-III will start the consultation using the modified fact base.

3.3.2.3 Linguistic Approximation Module

One main characteristic of Z-II is that it can communicate with the user through the use of linguistic variables and hedges. Z-III keeps this feature.

The linguistic approximation is the interface between the user and the internal representation of fuzzy concepts (stored as fuzzy sets) and fuzzy certainty factors (stored as fuzzy numbers). The approach used in linguistic approximation of fuzzy sets is similar to Z-II by following that of Wenstop [Wenstop 1980]. Firstly, a set of grammars used in expressing the linguistic values is defined. The syntax is shown in table 3.2.

STATEMENT ::=	SIMPLE_TERM | HEDGE STATEMENT | STATEMENT CONNECTIVES STATEMENT | unknown
SIMPLE_TERM ::=	high | medium | low
HEDGE ::=	above | below | around | upper | lower | rather | moreorless | very | not | neither | possibly | truly | indeed | fuzzily
CONNECTIVES ::=	and | or | but | to

Table 3.2 Linguistic Value Grammar

Note that SIMPLE_TREM refers to the three basic fuzzy sets for each fuzzy type. The HEGDES and CONNECTIVES can be viewed as operators that modify the fuzzy sets.

In Z-III, certainty factors are represented as fuzzy numbers. They can be specified according to the grammar shown in table 3.3.

C_F ::=	ADV NUM | ADV NUM TO ADV NUM
ADV ::=	around | about | roughly | close_to | very_close_to | moreorless | quite | possibly | nil
NUM ::=	real number from 0 to 1

Table 3.3 Grammar for Certainty Factor

There is another form of linguistic approximation , which is based on the imprecision-location approach. In this case, 19 linguistic values are used. After the appropriate linguistic value is found, the word "certain" will be concatenated to it so as to express a complete meaning. Table 3.4 is a comparison of the two methods :

Fuzzy number	Numeric form	Linguistic value
[0.4 0.5 0.5 0.6]	close_to 0.5	moreorless certain
[0.1 0.3 0.5 0.7]	about 0.3 to about 0.5	little certain
[0.1 0.5 0.5 0.9]	possibly 0.5	roughly certain

Table 3.4 Certainty Factor Linguistic Approximation

Both kinds of linguistic approximations have their own characteristics and practical presentations. For example, the linguistic value is used in drawing the conclusion because it is a more natural way to express certainty in such form than a numeric form.

3.3.3 System Properties Management Module

This module is used to perform the system functions like loading and saving knowledge base, system reset, OS shell, system exit etc. Moreover, users can change most of the system configuration parameters which are as follows :

a) The goal object

In Z-III, users can define more than one goal. The system will trace all the goals one by one and draw the conclusions. Thus, Z-III has the capability to construct a multi-layer expert system.

b) The inference mechanism

Users can choose one of the inference mechanism (Rs, Rg or Rsg). The choice depends on the nature of problems and the inference process.

c) The value of system threshold

By using threshold, weak evidence, with certainty factor less than the threshold, will not trigger the inference mechanism (see section 3.4.3). Users can assign different threshold values for different knowledge bases.

d) The database option

Z-III has the capability to retrieve data directly from a database during consultation. Users can choose either to input from database or from keyboard.

3.4 Additional Features

From section 2.3, it is found that there exist some limitations and weakness in Z-II. During the design of Z-III, attention has been paid to deal with these shortcomings. Besides, some new features have been added in Z-III facilitate the knowledge engineering and consultation processes. The following subsections detail the seven major additions.

3.4.1 Weights

In Z-II, all the propositions in the antecedent part of a rule have the same weighting, and also all the rules in the knowledge base have the same significance, but this may not be the real case in human reasoning. In discussions with experts who are trying to describe their decision processes, in particular diagnostic processes, we observe that they often state a condition for a specific diagnosis as a series of observations or symptoms. However, these experts may indicate that certain observations are more important than the others. Observations have ranged from very important to merely supportive or confirmatory. Therefore, a novel weighting system has been developed for propositions in the antecedence of the rules in Z-III.

On the other hand, Z-III assumes that each rule contributes to the hypothesis with identical weighting in the case of evidence combination. Although, weights are different from certainty factors used in rules, it is intuitively believed that a more certain rule gives more confidence in the conclusion drawn than a less certain one. For example, an expert gives two rules r1 and r2 and both have the same consequence. If the expert gives a larger certainty factor to r1 than that to r2, this means that, in the mind of the expert, r1 should contribute more than r2. So the certainty factors can be used as comparative indices of the weights of the rules with the same conclusion.

Thus, we can say that the certainty factors can also reflect the relative degrees of importance of the rules having the same consequence. Therefore, there is no need to have other mechanisms to handle the relative weighting of the rules. The only mechanism needed in Z-III is the handling of the weights of propositions in a rule.

3.4.1.1 Fuzzy Weight

In order to represent the degree of importance of each proposition in the antecedent part of a rule, a fuzzy or non-fuzzy weight can be attached to each proposition by putting the weight inside the []. For example :

IF (height is tall [about 0.5])
AND (education standard is high [1.0])
THEN application of a police inspector is acceptable
Certainty is 0.8

The above rule shows that in order to decide whether a man could succeed in the application of a police inspector, high education standard is more important than height. Since a weight is a relative quantity, the weight of the most important proposition should be set to 1.0. The term 'about 0.5' is a fuzzy weight.

A fuzzy weight is represented by a fuzzy number. The definitions of a fuzzy number is given in section 2.2.2. A fuzzy weight can be conveniently represented as a 4-tuple (w_1, w_2, w_3, w_4), and $\forall w_i \in [0, 1]$. w_2 and w_3 denote the interval in which the membership degrees are equal to 1. w_1 and w_4 denote the interval in which the membership degrees are non-zero. The fuzzy weight 'about 0.5' can be indicated by (0.3 0.5 0.5 0.7) as shown in Figure 3.3.

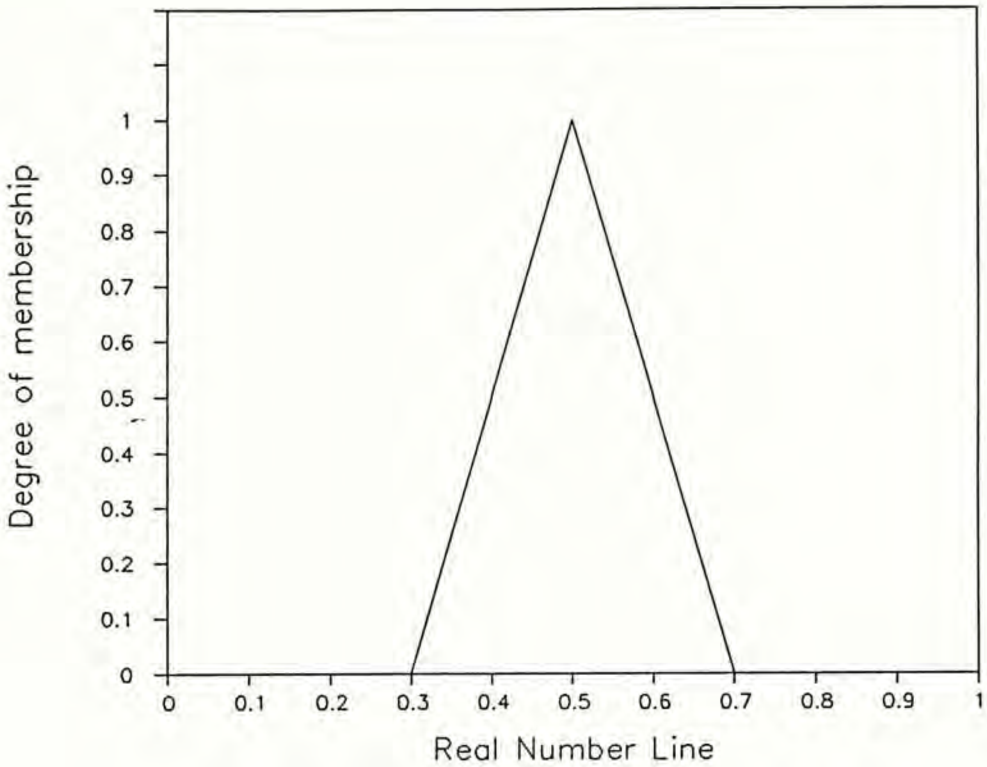

Figure 3.3 Fuzzy Weight of 'about 0.5'

3.4.1.2 Fuzzy Weight Evaluation

In order to reflect the influence of the weight attached to each proposition, the formulae used to calculate the certainty factor of a conclusion should be modified.

Consider the followings :

RULE : IF A1 [W_1] and A2 [W_2] THEN C is V (FN_R)

FACT : $A1'(FN_1)$

$A2'(FN_2)$

CONCLUSION : C is $V'(FN_c)$

where W_1, W_2 are fuzzy weights of propositions A1, A2 respectively

FN_1, FN_2 are fuzzy certainty factors of $A1'$, $A2'$ respectively

FN_R is fuzzy certainty factor of the rule

FN_c is fuzzy certainty factor of the conclusion

From section 2.2.2, the certainty factor of the conclusion is as follows :

$FN_c = \min\_fn(FN_1, FN_2) * FN_R$

where $\min\_fn$ means minimum function.

When a weight is added to each proposition, the new formula is given below :

$$FN_c = \min\_fn(FN_1^*, FN_2^*) * FN_R$$

where

$$FN_k^* = FN_k + (I - \hat{W}_k) * (I - FN_k)$$

if $W_k = (w_1, w_2, w_3, w_4), \hat{W}_k = (w_4, w_3, w_2, w_1)$ for k = 1, 2
and I = (1, 1, 1, 1)

In Z-III, all the certainty factors such as FN_1 and FN_2 are represented as fuzzy numbers , which are normal and convex, with the values of the 4-tuples in the range of 0 to 1. Therefore, in order to satisfy the definition of a fuzzy weight, FN_k^* and FN_c should be convex and normal also. Moreover, the values in the 4-tuples of FN_k^* and FN_c must be in the range between 0 and 1. The followings will prove that FN_k^* and FN_c satisfy the above requirements.

a) Proof : (the 4-tuples of FN_k^* is bounded by 0 and 1)

Let $FN_k = (f_1, f_2, f_3, f_4)$ $\forall i = 1..4, f_i \in [0, 1]$

$W_k = (w_1, w_2, w_3, w_4)$ $\forall i = 1..4, w_i \in [0, 1]$

Then

$$FN_k^* = (f_1, f_2, f_3, f_4) + (1 - w_4, 1 - w_3, 1 - w_2, 1 - w_1) * (1 - f_1, 1 - f_2, 1 - f_3, 1 - f_4)$$

The general term (ith term) =

$$f_i + (1 - w_{5-i}) * (1 - f_i)$$

$= \quad f_i + 1 - w_{5-i} - f_i + w_{5-i} f_i$

$= \quad 1 - w_{5-i}(1 - f_i)$

Since $\quad w_{5-i} \in [0, 1]$ and $(1 - f_i) \in [0, 1]$

Then $\quad 1 - w_{5-i}(1 - f_i) \in [0, 1]$

Hence, the 4-tuples of FN_k^* is bounded by 0 and 1.

b) Proof : (FN_k^* is convex)

We want to prove that if $FN_k^* = (c_1, c_2, c_3, c_4)$, then $c_i \leq c_{i+1}$, i = 1,2,3

From the above, we consider $c_{i+1} - c_i$

$= \quad 1 - w_{5-(i+1)}(1 - f_{i+1}) - [1 - w_{5-i}(1 - f_i)]$

$= \quad w_{5-i}(1 - f_i) - w_{4-i}(1 - f_{i+1})$

Since $\quad w_{5-i} \geq w_{4-i}$

$(1 - f_i) \geq (1 - f_{i+1})$

$\therefore \quad c_{i+1} - c_i \geq 0$

$c_{i+1} \geq c_i$

Hence FN_k^* is convex.

Finally, FN_k^* is obviously normal.

As FN_k^* is convex and normal, and the values in the 4-tuples are bounded by 0 and 1 , FN_c will also satisfy the above three requirements. As a conclusion, the fuzzy weight evaluation formulae are consistent with the definition of fuzzy weights.

3.4.1.3 Results of Adding Fuzzy Weights

From the fuzzy weight evaluation equations, we can see that :

If : $W_k = (1.0, 1.0, 1.0, 1.0)$, $FN_k^* = FN_k$

$W_k = (0.0, 0.0, 0.0, 0.0)$, $FN_k^* = (1, 1, 1, 1)$

$W_k = (0.5,0.5,0.5,0.5)$, $FN_k^* = (0.5,0.5,0.5,0.5) + (0.5,0.5,0.5,0.5) * FN_k$

If the weight of a proposition is 1.0, its certainty factor has influence on the conclusion as before. But if the weight of this proposition is small ($<<1$), then the effect of the uncertainty of this proposition will be weakened. The following example will illustrate the results of certainty factor evaluation when fuzzy weights are added.

Let $W_1 = (0.4, 0.5, 0.5, 0.6)$ (close to 0.5)

$W_2 = (1, 1, 1, 1)$ (1)

$FN_R = (1, 1, 1, 1)$ (absolutely certain)

Case 1

Let FN_1 = (1, 1, 1, 1) (absolutely certain)

FN_2 = (0.1, 0.3, 0.3, 0.5) (about 0.3)

Then FN_1^* = (1, 1, 1, 1)

FN_2^* = (0.1, 0.3, 0.3, 0.5)

$\therefore FN_c$ = (0.1, 0.3, 0.3, 0.5) (about 0.3)

Case 2

Let FN_1 = (0.25, 0.4, 0.4, 0.55) (around 0.4)

FN_2 = (1, 1, 1, 1) (absolutely certain)

Then FN_1^* = (0.55, 0.7, 0.7, 0.82)

FN_2^* = (1, 1, 1, 1)

$\therefore FN_c$ = (0.55, 0.7, 0.7, 0.82) (around 0.7)

In the above examples, we assume that the second proposition (A2) with weight W_2 being equal to 1.0 is more significant than the first proposition (A1) with W_1 being equal to 'close to 0.5'. In case 1, since the certainty factor of $A2'$ is 'about 0.3', hence, the certainty of conclusion is also 'about 0.3'. In case 2, $A2'$ is absolutely certain but

$A1'$ is not certain (around 0.4). As A1 is less significant, its effect on conclusion is less than that of A2. Hence, the certainty of conclusion after calculation is 'around 0.7' (between around 0.4 and 1.0).

From the above example, it seems that adding a fuzzy weight to each proposition in the antecedent part of the rule will make Z-III more suitable to model human reasoning. In order to verify the rightness and usefulness of fuzzy weight evaluation, a program has been written to test exhaustively the combination of different values of W_1, W_2, FN_1 and FN_2. The algorithm of the program is shown below :

```
W1 = 1
FOR W2 = close to 0.1 TO 1 STEP 0.1
        FOR FN1 = close to 0.1 TO 1 STEP 0.1
        FOR FN2 = close to 0.1 TO 1 STEP 0.1
                Calculate FNc
                Output the values of W1, W2, FN1, FN2 and FNc

W2 = 1
FOR W1 = close to 0.1 TO 1 STEP 0.1
        FOR FN1 = close to 0.1 TO 1 STEP 0.1
        FOR FN2 = close to 0.1 TO 1 STEP 0.1
                Calculate FNc
                Output the values of W1, W2, FN1, FN2 and FNc
```

After analyzing the outputs, it was found that the results given by fuzzy weight evaluation are applicable, consistent, and desirable. If fuzzy weights are used in Z-III, the medical experts find that the medical knowledge can be represented more accurately and effectively.

3.4.2 Fuzzy Matching

In Z-II, all the propositions containing a fuzzy object will be triggered during consultation. Thus, there must exist some mechanisms to determine whether the propositions containing a fuzzy object should be fired or not. Fuzzy matching is a good tool to manage this kind of problem.

For a non-fuzzy object, the matching process is an *all-or-none* one [Cayrol 1982]; however, it is not the case for a fuzzy object. Let's consider the following example in ESROM :

```
(Rule lp5
 IF (usgliq is decreased)
 THEN diagnosis is membrupt)
 Certainty is 0.7
```

where usgliq means 'amount of liquor apparent in USG', and

membrupt means 'membrane rupture'.

'usgliq' is a fuzzy object whose value comes from the fuzzy type 'usgliq' (excessive normal decreased). During consultation, if the user enters 'moreorless decreased' or 'not excessive', should rule lp5 be triggered ?

In order to determine the likeness of the pattern and data in fuzzy environment, the concepts of *similarity* has to be introduced first.

3.4.2.1 Similarity

The similarity M is calculated by the following algorithm: [Lam 1988]

If $N(F_1 \mid F_1') > 0.5$

then M = $P(F_1 \mid F_1')$

else M = $(N(F_1 \mid F_1') + 0.5) * P(F_1 \mid F_1')$

where

$P(F_1 \mid F_1')$ is the possibility of the fuzzy data F_1' given the fuzzy pattern F_1

$N(F_1 \mid F_1')$ is the necessity of the fuzzy data F_1' given the fuzzy pattern F_1

The formulae of the possibility and necessity measures between a fuzzy data and a fuzzy pattern are given as follows :

$$P(F_1 \mid F_1') = \max\left(\min\left(\mu_{F_1}(w), \mu_{F_1'}(w)\right)\right)$$

$$N(F_1 \mid F_1') = 1 - P(\sim F_1 \mid F_1')$$

where $\mu(w)$ is the membership function of w in the universe of discourse.

$\sim F_1$ is the complement of F_1

The possibility, necessity and similarity measures described above have been proved to be a useful tool to determine the similarity between the fuzzy data in the fact base and the fuzzy pattern in the premise part of the rule. By using the above three measures, fuzzy pattern matching can be performed in Z-III.

3.4.2.2 Evaluation of Similarity measure

In order to verify the usefulness of similarity M, the following experiment has been set up and the results are analyzed in detail.

Let height be a fuzzy object with the fuzzy concept (tall, medium, short). The three fuzzy terms are represented by the following fuzzy sets (the default fuzzy set in Z-III) :

tall (upper) :	medium (medium) :	short (lower) :
0.0000	0.0000	1.0000
0.0000	0.0000	0.6561
0.0000	0.0000	0.3277
0.0000	0.0588	0.1176
0.0000	0.3280	0.0280
0.0000	1.0000	0.0000
0.0280	0.3280	0.0000
0.1176	0.0588	0.0000
0.3277	0.0000	0.0000
0.6561	0.0000	0.0000
1.0000	0.0000	0.0000

Table 3.5 Default fuzzy sets in Z-III

We want to find out all the possible fuzzy or exact matching between the data and the given patterns. Hedges like 'very' and 'quite' and logical negation 'not' are added to the fuzzy terms in order to verify the power of the similarity measure. So the values of F_1 and F_1' could be

[not] [very | quite] {tall | medium | short}

where the choice inside the [] is optional.

The results of comparing the similarity of any two fuzzy terms F_1 and F_1' can be divided into 4 groups :

1) $P(F_1 \mid F_1') = 1.0$, $N(F_1 \mid F_1') > 0.5$ and $M(F_1 \mid F_1') = 1.0$

e.g. M(tall | tall) = 1.0

M(tall | very tall) = 1.0

M(not very medium | not quite medium] = 1.0

M(not medium | tall] = 1.0

M(not quite short | tall) = 1.0

M(not short | medium) = 1.0

2) $P(F_1 \mid F_1') = 1.0, N(F_1 \mid F_1') < 0.5$ and $M(F_1 \mid F_1') > 0.5$

e.g. M(very tall | tall) = 0.9305

M(not quite tall | not very tall) = 0.9276

M(very medium | quite medium) = 0.9272

M(very short | quite short) = 0.9276

3) $P(F_1 \mid F_1') = 1.0, N(F_1 \mid F_1') = 0.0$ and $M(F_1 \mid F_1') = 0.5$

e.g. M(tall | not medium) = 0.5

M(tall | not quite short) = 0.5

M(not very short | not quite tall) = 0.5

M(medium | not short) = 0.5

4) $P(F_1 \mid F_1') < 1.0, N(F_1 \mid F_1') = 0.0$ and $M(F_1 \mid F_1') < 0.5$

e.g. M(tall | not tall) = 0.1720

M(quite tall | medium) = 0.0837

M(medium | short) = 0.0294

M(short | tall) = 0.0

The similarities of groups 1 and 2 are both greater than 0.5. As a conclusion, if the data and pattern fall into these two groups, we say that they are exact or partially (fuzzy) matched respectively and the corresponding rules can be triggered. Similarities of group 4 are less than 0.5 and we could say that data and pattern do not match each others.

Thus, if the similarity M between the data and pattern is greater than 0.5, then we can conclude that they match with each other; otherwise, they do not. If M is greater than 0.5, it means that the similarity between the data and pattern is greater than the difference between them.

When the similarity equals to 0.5, the truth or the falsity cannot be indicated by the similarity between the data and pattern [Dubois and Prade 1988]. Thus, as the similarity M equals to 0.5 in group 3, it means that we cannot conclude whether the data and the pattern are the same or similar to what extent. However, in the implementation of Z-III, if the similarity is equal to 0.5, the corresponding proposition will be fired.

One noticeable interesting thing is that there exists some cases for $M(F_1 \mid F_1')$ = 1 (group 1) but $M(F_1' \mid F_1)$ = 0.5 (group 3)

e.g. M(not medium | tall) = 1 but M(tall | not medium) = 0.5

M(not short | medium) = 1 but M(medium | not short) = 0.5

Moreover, we could also find cases where both $M(F_1 \mid F_1')$ and $M(F_1' \mid F_1)$ are greater than 0.5, but they are not equal to each other.

e.g. M(tall | very tall) = 1 but

M(very tall | tall) = 0.9305

From the above case, it can be concluded that similarity is not commutative. That is $M(F_1 \mid F_1') \neq M(F_1' \mid F_1)$

Finally, as a conclusion, the similarity is a good measure of the degree of matching between the pattern and data. By using it, Z-III can trigger those rules in which the pattern and data are not exactly matched in a reasonable manner.

3.4.3 Use of System Threshold

In MYCIN [Buchannan and Shortliffee 1984], a threshold value 0.2 is selected empirically. A certainty factor with absolute value less than 0.2 contributes only a little evidence to the hypothesis but the consideration of this weak evidence will waste a lot of computing time.

Besides, if there are a lot of weak evidence existed, their effects may reject a strong evidence. It is because the formula for evidence combination will accumulate all the weak evidences supporting a hypothesis and finally the combined certainty factor may be very large. Table 3.6 shows the effect of the accumulation of weak evidences.

No. of weak evidences	CF = 0.200	CF = 0.150	CF = 0.100
1	0.200	0.150	0.100
2	0.360	0.278	0.190
3	0.488	0.386	0.271
4	0.590	0.498	0.344
5	0.672	0.556	0.410
6	0.738	0.623	0.469
7	0.790	0.679	0.522
8	0.832	0.728	0.577
9	0.866	0.768	0.613
10	0.893	0.803	0.651
11	0.914	0.833	0.686
12	0.931	0.858	0.718
13	0.945	0.879	0.746
14	0.956	0.897	0.711
15	0.965	0.913	0.794
16	0.972	0.926	0.815
17	0.977	0.937	0.833
18	0.982	0.946	0.850
19	0.986	0.954	0.865
20	0.988	0.961	0.878
21	0.991	0.967	0.891
22	0.993	0.972	0.902

Table 3.6 Accumulated Certainty Factor of Weak Evidence

From the above table, we can see that if 8, 10, 16 weak evidences exist with certainty factors equal to 0.2, 0.15, 0.1 respectively, they will override a strong evidence, with a certainty factor equals to 0.8. By using the system threshold, this phenomenon can be avoided.

In real cases the above phenomenon may cause no problem. In medicine, there exists some aspects that a group of weak evidences must override a strong evidence. However, there also exist another set of aspects that a group of weak evidences should not overrule a strong evidence. Therefore, in Z-III, the threshold of each knowledge base is not fixed and it is set by the knowledge engineer during knowledge acquisition. The value of the system threshold may be changed easily by using the System Properties Management Module. Therefore, different domains can have different values of thresholds.

There are two places where system thresholds can be applied. The first place is the certainty factor of the facts entered by the user during consultation. If a user inputs a fact with certainty less than the system threshold, Z-III will ignore it and so no rule will be matched. Another place is the certainty of the rule. Let consider the following rule :

(Rule x
IF (height is tall)
THEN favourite activity is basketball)
Certainty is 0.15

From rule x, we can say that the domain expert has only little confidence in the rule 'if someone is tall, he likes basketball'. Thus, rule x only contributes a weak evidence. So if the system threshold is greater than 0.15, rule x will not be triggered. The computing time will then be reduced.

By using the system threshold, the domain expert need not remove rule x from the knowledge base even a rule has a low certainty factor. Moreover, if the domain expert wants to include this rule later, the system threshold can be changed to less than 0.15. Then rule x will be active during consultation.

3.4.4 Use of Threshold Expression

As described above, the system threshold is used to eliminate all weak evidences existed in the system. However, in real cases, different objects may have different values of thresholds. Even the same object may have different values of threshold in different circumstances.

In addition, some experts think that a proposition is triggered only if the certainty of the fact is larger than a fixed value. For example, in the medical expert system INDUCE36 (see section 5.2), the expert thinks that in order to determine whether a woman is pregnant over 38 weeks, one of the below conditions should be satisfied :

a) She is certain that she is pregnancy over 38 weeks or

b) She is not certain that she is pregnancy over 38 weeks but she thinks that the minimal gestation period is 36 weeks.

The original knowledge representation used by Z-II cannot handle the above situations. Therefore, in Z-III, a threshold of individual object can be attached to each proposition in the antecedent part of a rule. For example,

```
(Rule x
 IF (height is tall {CF > = 0.7})
 THEN favourite activity is basketball)
Certainty is 0.6
```

The expression inside the {} is called threshold expression which shows the threshold of each object in the proposition. The above rule means that if the height of someone is tall with certainty greater than or equal to 0.7, his favourite activity may be basketball with certainty 0.6.

Fuzzy comparison can be used in threshold expression. The degree of matching is the same as Z-II. For example

```
(Rule x
 IF (weight is heavy {CF >= 0.7, 0.9})
 THEN favourite activity is wrestler)
Certainty is 0.7
```

If the weight is heavy with certainty 0.8, the favourite activity is wrestler with certainty 0.28. The calculation of certainty factor of the conclusion is simple and as follows :

Certainty of fact (heavy) = 0.8 and

Degree of matching of the threshold expression = 0.5

Thus Overal! certainty of proposition = 0.5*0.8 = 0.4

Then Certainty of conclusion = 0.4*0.7 = 0.28

In order to maintain the consistency of inference process in Z-III, the above calculation is also under the control of the system threshold. Therefore, for the above case, if the system threshold is greater than 0.4 (Overall certainty of proposition), the rule x will not be triggered.

With the help of threshold expression, the rule to determine whether the gestation of a pregnancy is over 38 weeks is represented in INDUCE36 as below :

```
( Rule a5
 IF ( ( gestation  > = 38  {CF > = 0.7})
   OR (    ( gestation  > = 38  {CF < = 0.7})
      AND ( mingest  > = 36 ) ) )
 THEN  gest38 is yes)
Certainty is 1
```

where gestation means the number of weeks of gestation and

mingest means the minimal weeks of gestation

Threshold expression is a very useful tool in building medical expert systems because the expert always expresses that a rule can be fired if and only if the certainty of the proposition exceeds some values. With fuzzy comparison incorporated in the threshold expression, the expert can represent his knowledge more accurately and efficiently.

3.4.5 Playback File

The playback file actually stores the consultation history for future use. At the end of a consultation, all the facts entered can be saved for future review.

A knowledge base can have several playback files. A playback file contains information for Z-III to identify its corresponding knowledge base. It has saved all the details of all the instantiated objects including their values and certainty factor.

Before each consultation, the user could load one of the playback file of the knowledge base. Figure 3.4 shows the screen layout for loading playback file in Z-III. The user can then start the consultation.

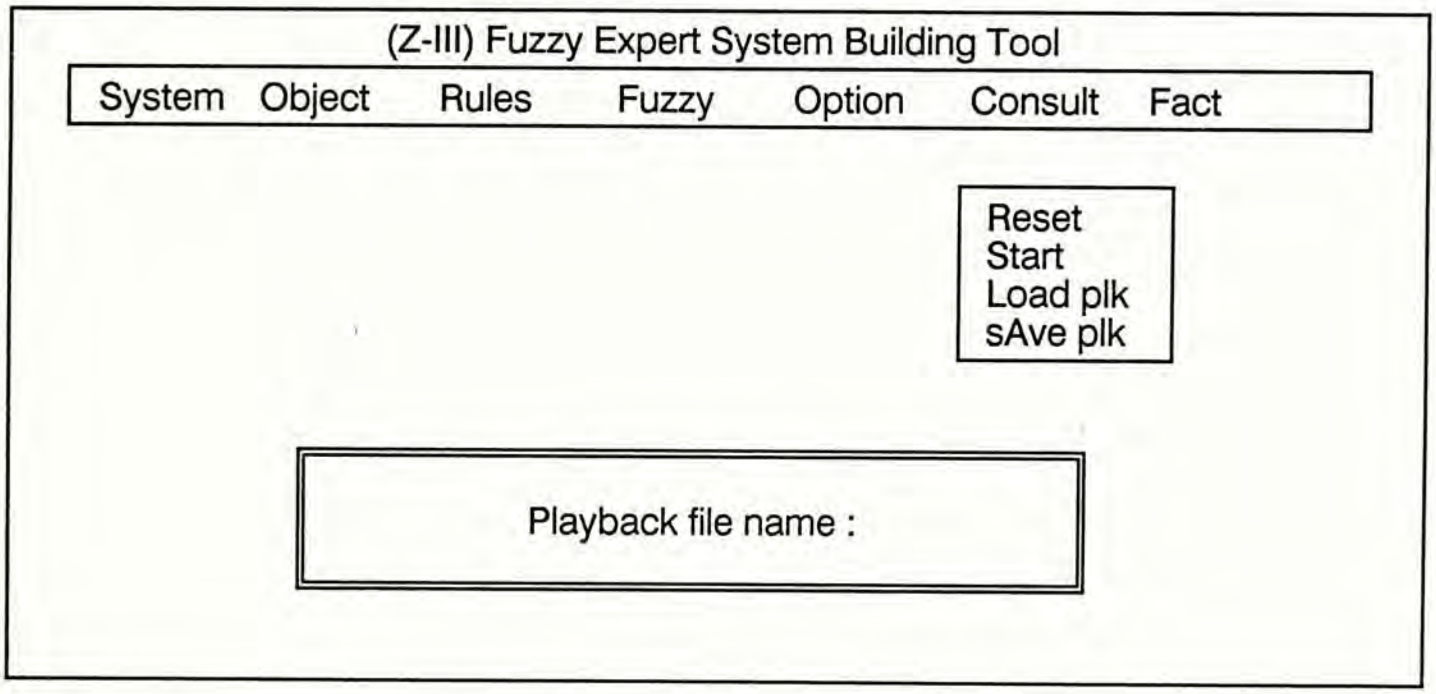

Figure 3.4 Screen Layout for Loading Playback

3.4.6 Database Retrieval

During a consultation, especially in medicine, a large amount of information is required to enter to the expert system. This process is time-consuming. Z-III is capable to retrieve the appropriate data directly from the database created by dBase III. The advantages of this approach are that existing databases can be coupled to Z-III to perform intelligent task [Schwartz et al. 1987] and the databases can be used in other medical record management applications [Barnett 1982]. Furthermore, in the verification and validation processes of an expert system, using database retrieval will save a lot of time.

If users want to use database during consultation, they should set the appropriate system parameters in System Properties Module. A format file of the database should also be created.

The communication link between Z-III and the database is through the use of a format file which is a text file containing the structure (format) of the database. The structure includes the objects' name and length. Then, according to the information stored in the format file, Z-III can find the value of an object. Figure 3.5 shows the format file of INDUCE36 (see section 5.2).

activebleed 3	lessfetomove 3
age 5	liquor 18
ancx 20	mdcs 20
ancx 20	mdcs 20
ancx 20	mdcs 20
badhx 20	membrupt 3
badhx 20	mingest 5
badhx 20	nofetus 4
bs 5	parity 5
control 18	plan 8
ctg 12	retardation 8
derft 3	socioeconomic 8
fhr 6	symptomatic 3
gestation 5	tcontrol 3
growth 3	excerbation 3
Hb 5	maturity 3

Figure 3.5 Format File of INDUCE36

'ancx', 'badhx' and 'mdcs' are multi-valued objects. Thus, three locations are reserved for each of them in order for them to have a maximum of three possible values simultaneously. The choice of the number of location for each multi-valued object depends on the nature of the object. If the domain expert thinks that a particular multi-valued object may have more than three possible values simultaneously, more than three locations may need to be reserved.

A database may contain many records. Therefore, a unique record code is assigned to each record in order for the user and the system to identify them. So during each database consultation, Z-III will ask the user to input the record code and then the inference starts automatically. Moreover, there is an auto-execute facility implemented in Z-III. By using auto-execute, Z-III will retrieve all the records in the database automatically and the results will be written to a file for later analysis and interpretation. This facility greatly enhances the efficiency in knowledge refinement stage.

3.4.7 Numeric Variable Objects

In section 3.3.1.1, six types of object have been mentioned. The last one is called numeric variable object. A numeric variable object can be used as a constant in a numeric expression but its value may be altered at different consultation if necessary.

In the development of INDUCE36, the expert has given the following rule :

```
(Rule a31
IF ( (gestation > = 45 {CF > = 0.7})
      OR ( (gestation > = 45 {CF < 0.7})
            AND (mingest > = 42)
            AND (cervix is favourable) ) )
THEN plan is delivery)
Certainty is 1
```

where gestation means the number of weeks of gestation,

mingest means the minimal weeks of gestation,

cervix means cervical inducibility, and

plan means plan of management.

The above rule is given by a gynaecologist. However, other gynaecologists may not totally agree with the above rule. They may want to replace 45 with 44 in gestation, 42 with 41 in mingest. Owing to different options given by different experts, Z-III has provided a facility for the user to adjust the numeric values in the rules during consultation. The tool Z-III used is the numeric variable object.

In Z-III, the knowledge engineer can define a numeric variable object which can be used as the second operand in a numeric expression. The value of each numeric variable object can be altered during consultations if necessary. Thus, the above rule can be written as :

```
(Rule a21
 IF ( (gestation > = x {CF > = 0.7})
        OR ( (gestation > = x {CF < 0.7})
                AND (mingest > = y)
                AND (cervix is favourable) ) )
THEN plan is delivery)
Certainty is 1
```

where x and y are numeric variable objects

Furthermore, using numeric variable object will aid the knowledge refinement process. The expert can take different values of x and y and then compare the results with the real medical cases. If the number of testing cases is sufficient, good values of x and y can be established.

Simple arithmetic operations (addition, subtraction, multiplication and division) can be performed in numeric variable objects.

For example :

IF (gestation < = x + y, z * 5) THEN ...

All x, y and z are numeric variable objects.

These simple arithmetic operations will greatly simplify the representation and formation of rules involve fuzzy comparison. For example

IF (gestation > = x - 2, x + 2) THEN ...

Only one numeric variable object (x) is needed in the fuzzy comparison.

The numeric variable object is proved to be a good tool in knowledge representation and knowledge refinement. Having implemented this kind of object, Z-III has greatly enhanced its capability.

3.5 Implementation Highlights

The implementation stage was started off after the system design was completed. A lot of the efforts have been devoted to this part because the system is large and requires a lot of coding. The total number of source code is approximately 400 Kbytes while the size of the executable program is about 180 Kbytes. The implementation process is a top-down-bottom-up manner [Shooman] and structural programming is employed. The following sections will give more specific discussions on the implementation of each major subsystems.

3.5.1 Knowledge Base

The knowledge base mainly consists of three types of data : fuzzy types, objects and rules. The data organizations of objects and rules are dynamically allocated in nature, so as to provide the largest flexibility. Although in theory, it is less efficient than array, with careful use of pointers, such as the use of hash tables, the time required will not be increased much, as can be shown by the response time of the system.

3.5.1.1 Fuzzy Type

For each fuzzy type, there are three fuzzy sets that correspond to the upper, medium and lower level under it. A vector of eleven floating point numbers is used as the internal representation of a fuzzy set. The data structure used to store fuzzy type is very simple. It contains only the fuzzy type name and the locations of the three corresponding fuzzy sets. In the Fuzzy Term Management routine, a function called SEARCH_F is implemented which returns the location of a fuzzy type given the name of the fuzzy type. Operations such as creating, listing, deleting, viewing and element-wise editing of fuzzy sets are allowed. Figure 3.6 shows the screen layout of the Fuzzy Term

Management Module.

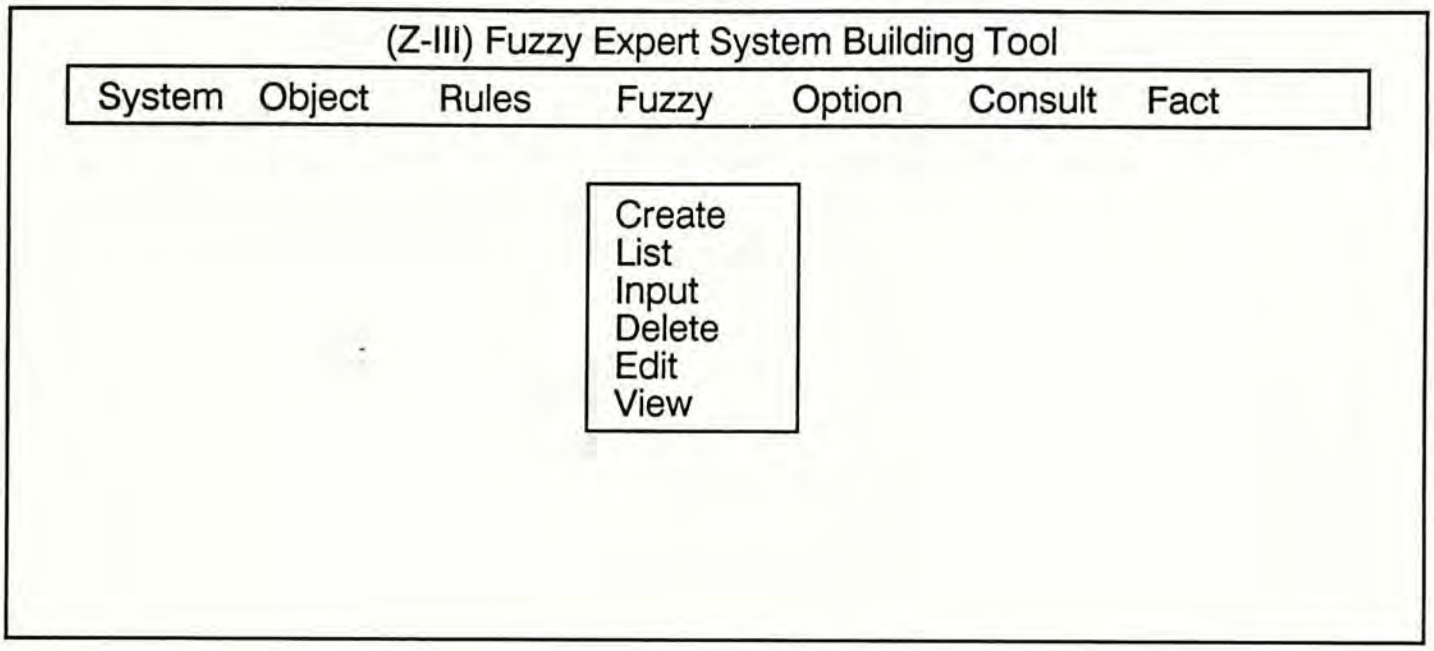

Figure 3.6 Screen Layout of Fuzzy Term Management Module

The operations provided are as follows :

Create	Create a fuzzy type and corresponding fuzzy sets
List	List all the fuzzy types
Input	Allow the users to input a particular fuzzy set
Delete	Delete a fuzzy type
Edit	Allow the users to alter a particular fuzzy set
View	Display the corresponding fuzzy sets of a fuzzy type

Users can select any option by pressing the arrow keys to move the highlight bar to the corresponding option and then pressing the <ENTER> key or pressing the corresponding hot key shown in capital character.

There are three default fuzzy sets (Table 3.5) corresponding to the three level situations of a fuzzy type. A user who does not have any particular fuzzy set in mind may use this default setting. However, the user can input or alter the appropriate fuzzy sets for each fuzzy type. Although fuzzy sets are stored as vectors, input of functional representations of fuzzy sets are allowed; an internal procedure is responsible to convert the functional representations into vector form.

3.5.1.2 Objects

Objects in Z-III is declared as a structure in C and have the following attributes :

Name	Name of the object
Type	Type of the object
Ask_first_flag	Showing whether its value is to be obtained by user during consultation
Translation	Natural language translation of the identification of the object
Prompt	Used when Z-III asks for the value of the object during consultation
Expected	Expected values specifying the legal values of the object
Obj_value	Instantiated value of the object
Rule_used	List of rules that use the object
Rule_updated	List of rules that update the object
Context	Store the context of the object and is used in the explanation and review facilities.

The field *Expected* which specifies the legal value of the object is declared as a union because there are six types of object in Z-III and each of them has different format of expected value. For both single-valued and multi-valued objects, *Expected* is a pointer pointing to a list of possible values which are in character type. For fuzzy object, *Expected* is an index which points to the corresponding fuzzy type. *Expected* is a null pointer for numeric, yes-no and numeric variable type objects, since their expected values are fixed.

Like *Expected*, the field *Obj_value* is also declared as a union. The structure of *Obj_value* is similar to that in *Expected* except for numeric object, it stores the numeric value of that object. For yes-no type object, it stores the certainty of that object. If the object's value is 'yes', positive certainty is stored; otherwise, negative certainty is stored.

One improvement has been made for storing the instantiated value of the single-valued objects. In Z-III, the *Obj_value* of a single-valued or multi-valued object is a pointer pointing to a list of instantiated values. The list is sorted according to the certainty of each instantiated values. Figure 3.7 depicts the structure of the *Obj_value* field of a single-valued object.

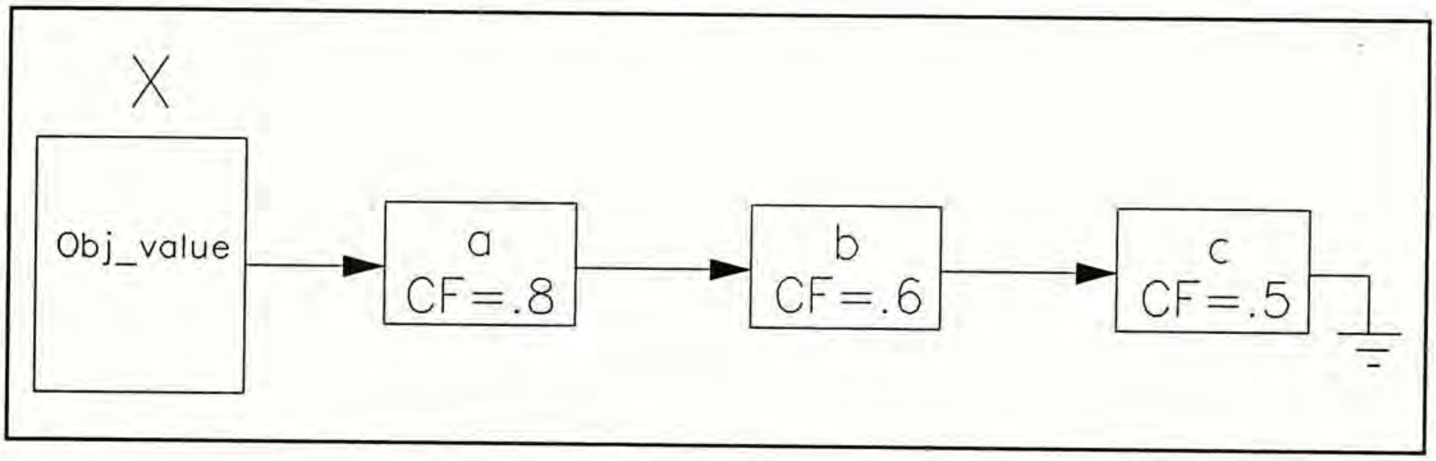

Figure 3.7 Structure of *Obj_value* of object X

As we can see, by using this approach, a single-valued object can stored more than one value. This will solve the second problem described in section 2.3. After the consultation, the output value of a single-valued object is the most certain value in the list pointed by *Obj_value*.

The objects are stored in a hash table and are accessed through the object name. The organisation of hash table of the objects are shown in figure 3.8.

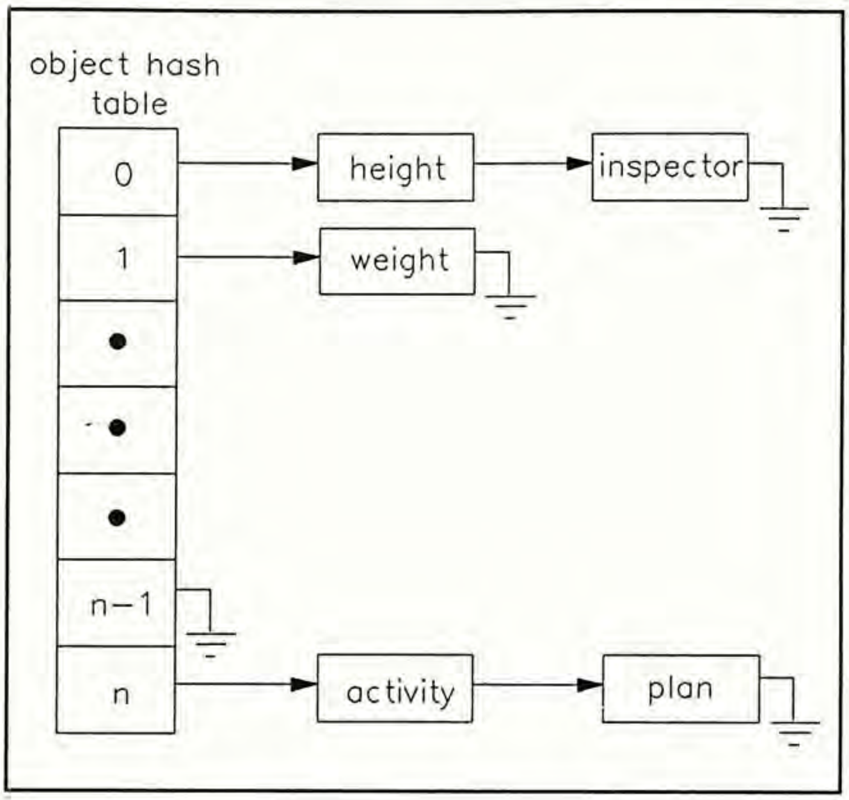

Figure 3.8 The Organization of Objects

The Object Management Module is responsible for the manipulation of all the objects in the knowledge base. The screen layout of the Object Management Module option is shown in figure 3.9.

(Z-III) Fuzzy Expert System Building Tool

System Object Rules Fuzzy Option Consult Fact

Create
List
Edit
Delete
View
Print

Figure 3.9 Screen Layout of Object Management Module

The operations provided are as follows :

Create	Create a new object
List	List out all existing objects
Edit	Edit the details of an object
Delete	Delete an object
View	View the details of an object
Print	Print all the objects in a text file

3.5.1.3 Rules

Rules are stored in rule node, which are declared as a structure in C. The rule node is the most complicated data structure in Z-III. The information contained in a rule node is as follows :

Rule_code	Rule code of a rule
Ante_part	A pointer pointing to the AND-OR tree of the antecedent proposition
Conseq_part	The structure of the consequent proposition
Cf	The certainty of that rule
Active	A flag which decides whether the rule is active in current context.

Each rule is uniquely identified by the rule code, and is organized in the knowledge base in a hash table (figure 3.10), which is equivalent to that of objects and is an array of pointers pointing to the rule node .

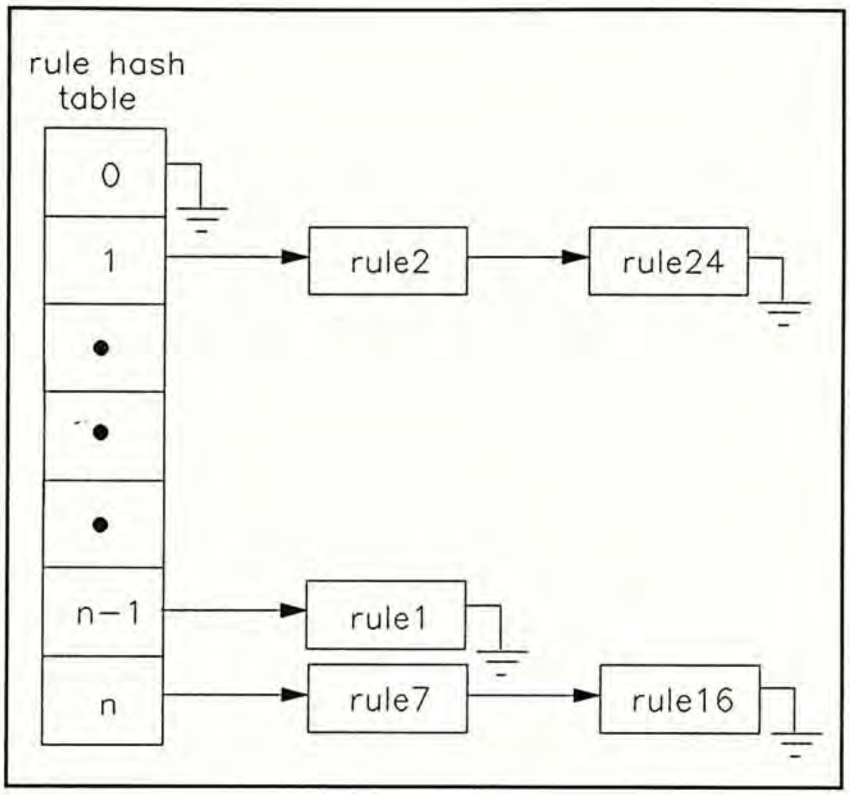

Figure 3.10 The Organization of Rules

Ante_part is a pointer to the AND-OR tree. The structure of the node of the AND-OR tree is as follows :

Obj_name	Name of the object in the node
Conn	Represents the type of the node (AND, OR, PROPOSITION)
Obj_value	The value of the object in the proposition
Left	Points to the left son of the node
right	Points to the right son of the node
Threshold1	First operand in threshold expression
Threshold2	Second operand in threshold expression
Weight	Fuzzy weight of the proposition

The field *Conn* is used to indicate the current type of the node. If its value is PROPOSITION which indicates a node containing a proposition, *Obj_name* and *Obj_value* will be filled with values. *Threshold1*, *Threshold2* and *Weight* is used optionally and their default values are 0.0, 0.0 and 1.0 respectively. However, if the value stored in *Conn* is either AND or OR, this means that the current node is a logical connective node which does not contain a proposition. The fields *Left* and *Right* will used to point to the nodes containing the left operand and right operand respectively. The structure of *Obj_value* at a rule node is similar to that of *Obj_value* at an object node and it is declared as a union.

The structure of *Conseq_part* is very simple and it contains only the *Obj_name* and *Obj_value* of the conclusion object.

The Rule Management Module is responsible for the manipulation of all the rules in the knowledge base. The screen layout of the Rule Management Module option is shown in figure 3.11.

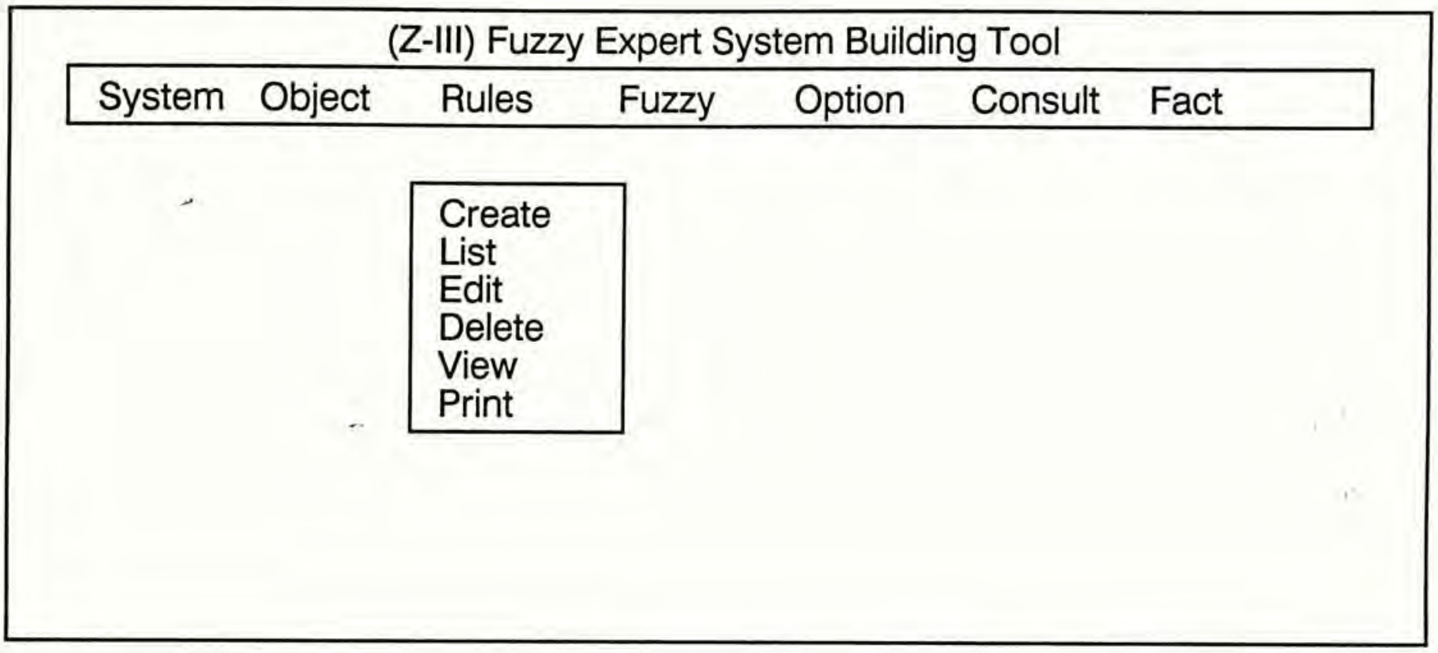

Figure 3.11 Screen Layout of Rule Management Module

The operations provided are as follows :

Create	Create a new rule
List	List out all existing rules
Edit	Edit a rule by using an external text editor
Delete	Delete a rule
View	View a rule
Print	Print all the rule in a text file

3.5.2 System Properties

There are two options in the main pull-down menus for the user to perform the necessary system properties management operations. They are *System* and *Option*.

3.5.2.1 System Menu

The System menu is shown in figure 3.12 :

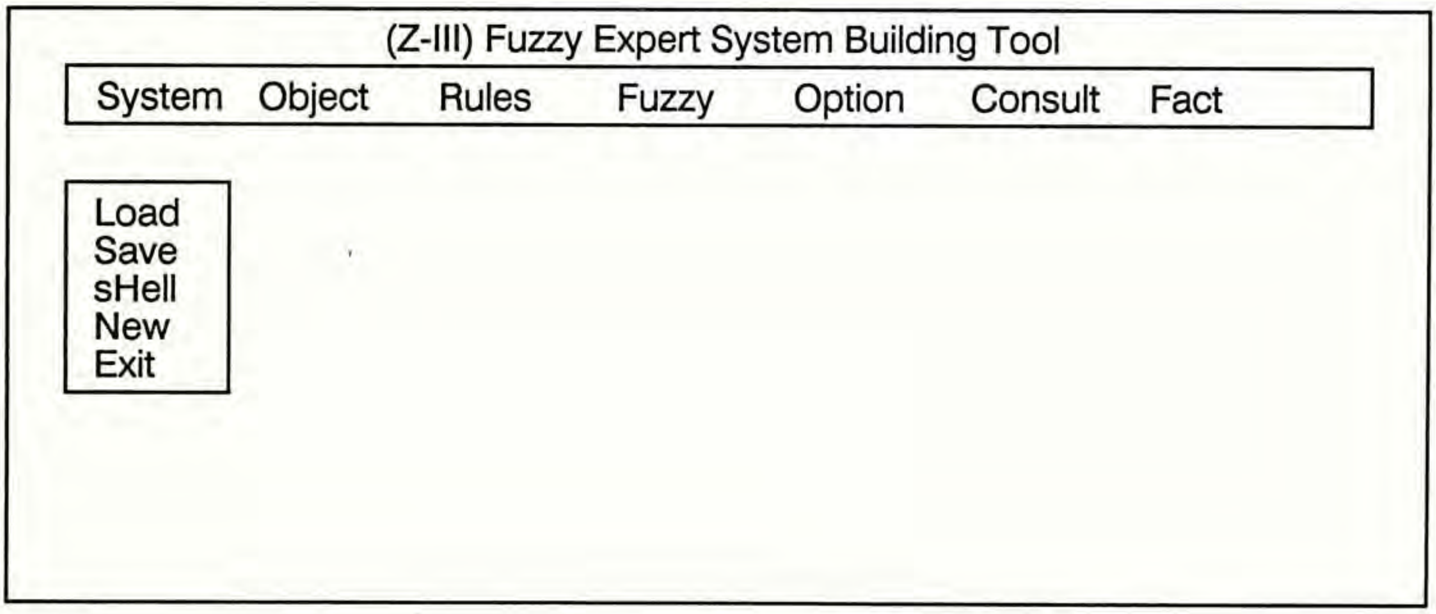

Figure 3.12 Screen Layout of System Menu

The options available include :

Load	Load knowledge base from disk
Save	Save knowledge base to disk
Shell	Go to MS/PC-DOS environment temporary
New	Clear the knowledge base in memory
Exit	Leave Z-III

3.5.2.2 Option Menu

The Option menu is shown in figure 3.13 :

(Z-III) Fuzzy Expert System Building Tool

System Object Rules Fuzzy Option Consult Fact

KB info.
Goal
Database
Variable
Formula
Threshold
Output

Figure 3.13 Screen Layout of Option Menu

The options provided are :

KB info.	List out the information of the current knowledge base
Goal	Alter the goal of the knowledge base
Database	Change the database option
Variable	Change the values of the numeric variable objects
Formula	Select the inference mechanism (Rs, Rg, Rsg)
Threshold	Alter the value of system threshold
Output	Direct the results of consultation to a file

3.5.3 Consultation System

The consultation system acts as a very important component in Z-III. From the user's point of view, consultation system is the part which the users use frequently. Thus, good control of the consultation system is necessary so as to acquire high efficiency.

The consultation system consists of three modules : inference engine, review management and linguistic approximation.

3.5.3.1 Inference Engine

The inference engine of Z-III is a recursive one, thus backward chaining can be implemented easily. In Z-III, backward chaining is used, not only because of search economy but also for the focused attention it gives to the problem, especially for diagnosis problems. Questions asked from the user are always related to the goal. This will appear more intelligible and less irritating.

In the process of consultation under backward chaining, a reasoning tree is built from the root (the goal) to the leaves (the symptoms). Figure 3.14 shows a sample reasoning tree. The reasoning tree contains the context of the current consultation.

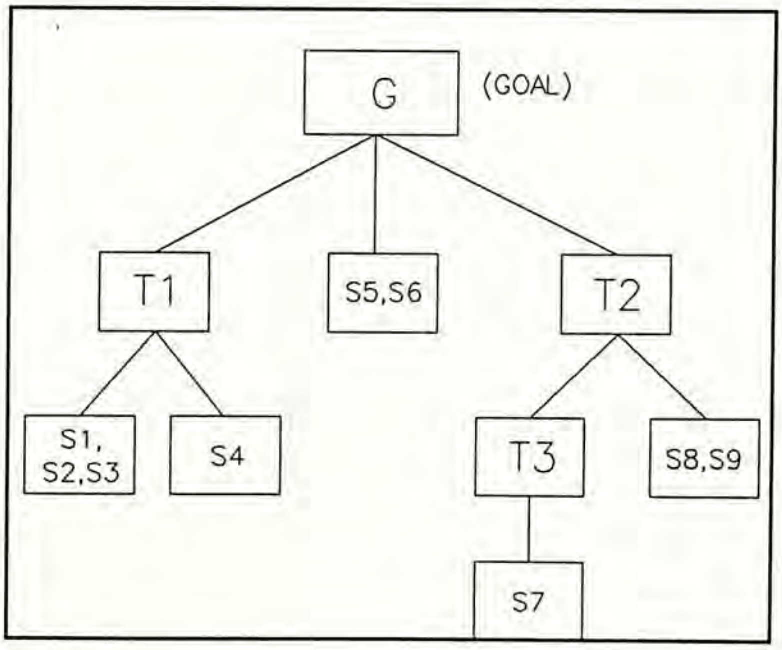

Figure 3.14 A Sample Reasoning Tree

For locating the possible rules efficiently to apply, rules are indexed in the *Rule_used* and *Rule_updated* fields of an object. Moreover, the chaining process defines a depth first, left-to-right traversal of the tree. For the above reasoning tree, the system will firstly evaluate the value of T1. This results in the system asking the values of the symptoms S1, S2 and S3 sequentially, then S4 is asked and evidence combination will perform to get the certainty of T1. This process will continue until all the possible paths to evaluate the goal (G) have been considered and then the whole reasoning tree has been built.

The screen layout of consultation is shown below :

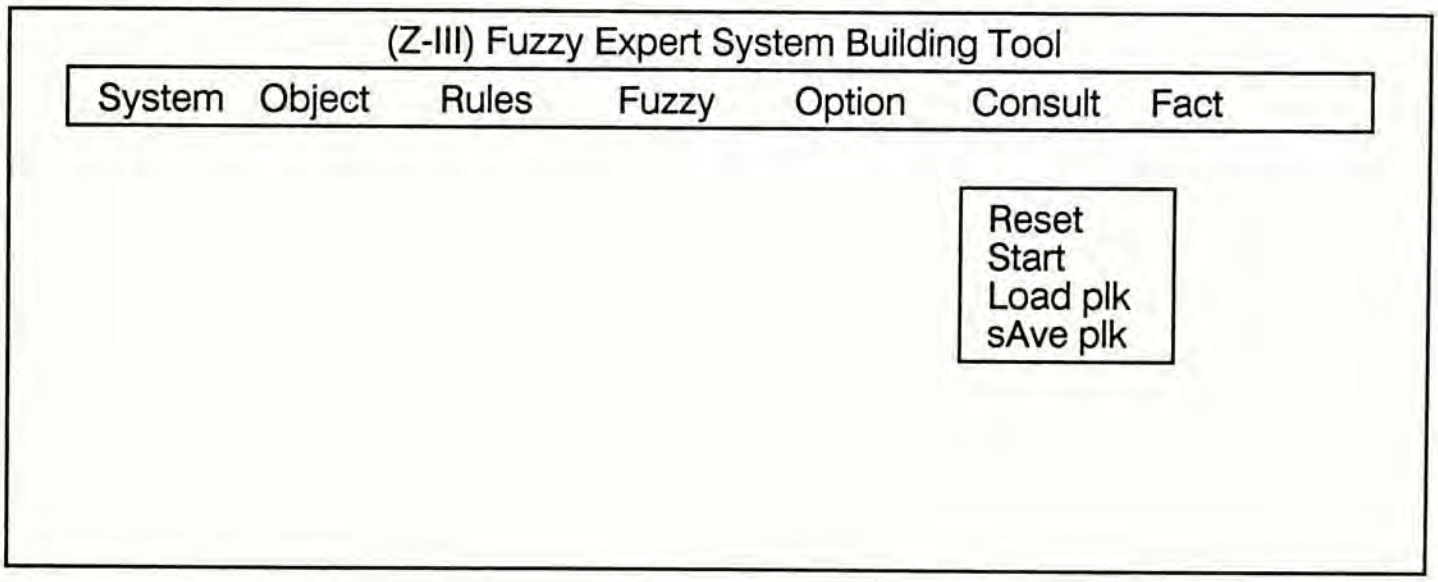

Figure 3.4 Screen Layout of Consultation Menu

The operations provided are as follows :

Reset	Reset the system to start a new consultation
Start	Start a new consultation
Load plk	Load the playback file from the disk to the fact base
Save plk	Save the fact base to a playback file

The WHY option is provided for the user to question Z-III why it needs the value of a particular object during consultation. For example, if Z-III is told to determine whether infection is found in a pregnancy according to the rule in ESROM :

```
(Rule i02
 IF (liquor discharge is msl) and (gestation < = 28, 36)
 THEN infection is yes)
Certainty is 0.8
```

At some point of consultation, Z-III decides that it needs the value of liquor discharge and checks that it is askable, then the system will ask the user to input the value. If the user wants to know why the system needs the value, the WHY option can be used. The dialogue is shown in table 3.6.

Z-III:	What is the liquor discharge ?
User:	Why
Z-III	I am trying to find the value of infection according to the rule i02 which states that : Rule i02 IF liquor discharge is msl AND gestation < = 28, 36 THEN infection is yes with certainty 0.8 (pretty certainty) with the following facts already known : [1] gestation is 26 CF is 1 (absolutely certain) so that if liquor discharge is known then the value of infection can be deduced.

Table 3.6 A Sample WHY dialogue

WHY is implemented using a goal stack, which is so managed :

1. When a subgoal G needs to be traced to evaluate a rule R because G appears in the premise of R, the pair (G,R) is pushed.
2. When a goal has been traced, all the contents in the stack are popped.

Successive answers to WHY can be obtained by looking successively deeper into the stack until a goal without a corresponding rule is found, which must be the final goal and Z-III simply answers 'Because you told me to find it!'

At the end of each consultation, Z-III will ask whether the user wants to know how the conclusion is drawn. It is the HOW facility. HOW is implemented by using a context stack which is so managed that:

1. When an object O is traced, it is pushed onto the stack with the list of contributing rules as the value
2. In WHAT-IF reviews (see section 3.5.3.2), if the user wishes to change the value of an object, the contents of the context stack are successively popped until the object is at the top of the stack.

Therefore, if the user wants to know how the goal is concluded, the system will list out all the rules that lead to that conclusion by going though the context stack.

3.5.3.2 Review Management

There are two review functions in Z-III : Fact Review and WHAT-IF Queries. The screen layout of review management is as below :

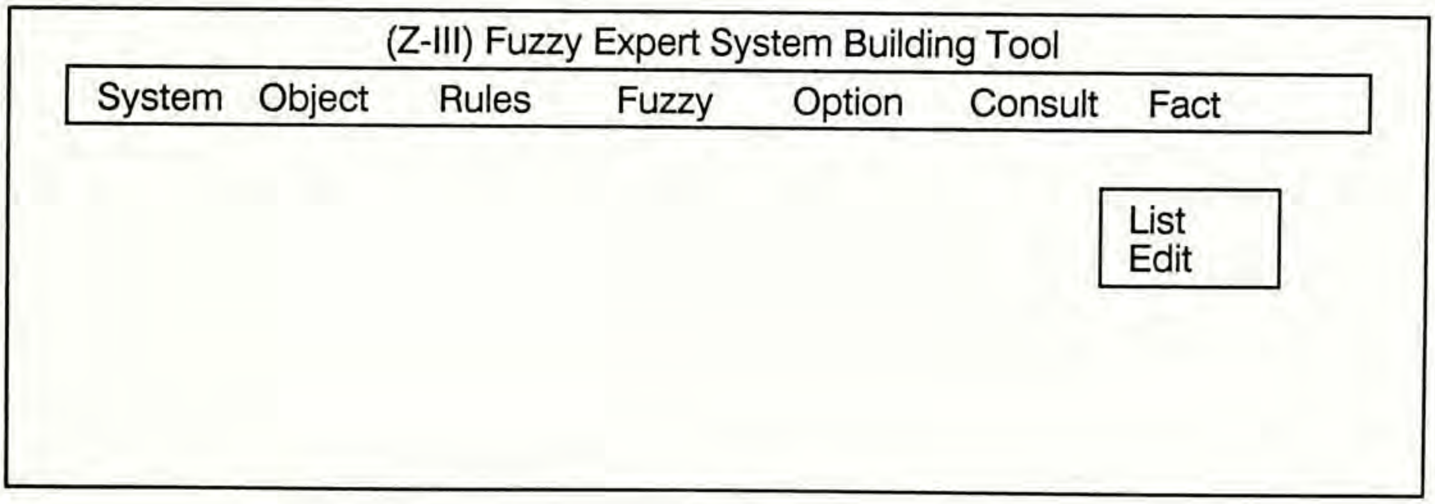

Figure 3.5 Screen Layout of Review Management

All the currently known facts in the system can be obtained from the context stack. Fact review (List) simply entails one passing of the stack to list all the facts found.

WHAT-IF queries (Edit) is useful in expert systems. As a user can slightly modify the problem state and test the sensitivity of the expert systems. It can also be used for analysis and teaching purpose.

The context stack is used for handling WHAT-IF queries. To allow a fact to be modified, we successively pop a data object from the context stack, resetting the value of the data object and activating all the rules that deduce it if the object is not askable until the interested fact has also been popped. Since an object nearer to the top of the context stack is also traced more recently, changing the value of the object can never affect the values of the objects deeper down in the stack.

3.6 Comparison on Z-III and Z-II

With the experience of using Z-II and Z-III extensively in medical expert systems like ABVAB, a comparison is drawn against certain criteria : Response Time, Accessibility, Knowledge Base size and User-friendliness. The comparing condition is that Z-II runs on VAX-11/780 and Z-III runs on a 33 MHz 80386 PC.

3.6.1 Response time

The criterion is based on the performance of the system merely from the user's point of view. Z-III gives a fast response during consultation of ABVAB (about 4.5 second); the user is prompted by questions from the consultation driver without any noticeable delay. However, Z-II needs at least 100 seconds for inference.

There is an annoying garbage collection time for the case of Z-II, because it runs on the VAX-LISP interpreter. The user has to get accustomed to that and it is rather disruptive especially during an on-line consultation under a clinical environment.

3.6.2 Accessibility

As Z-II is run on a VAX machine and only some large firms, government departments or universities can afford to buy, the distribution of Z-II is limited. However, since Z-III operates in a PC environment, almost anyone can use Z-III to build an expert system. Nowadays the cost of a PC has dropped drastically and even a small firm can afford to purchase one. As a result, owing to the popularity of PC, the accessibility of Z-III is higher than that of Z-II.

3.6.3 Accommodation of Large Knowledge Base

Since Z-II is implemented in an environment with a virtual memory management, the size of the in-core knowledge base can be very large. Thus the knowledge engineer does not need to worry about the memory requirement of the knowledge base. The system developer need not optimize code size and make decisions on trade-off of features. However, Z-III suffers from the memory limitation due to the PC-DOS environment. The operating system has only 640K of addressable main memory. Not only do the system developers need to consider the memory requirement when designing Z-III, but the knowledge engineer also has to estimate whether a new expert system can be built within the free memory limit. As a result, Z-III is only suitable in building an expert system in small or medium size.

3.6.4 User-friendliness

Concerning the I/O facilities, Z-II has the inherent difficulty of building a flexible and user-friendly interface and windowing system because of a lot of knowledge on the low-level run-time libraries and system services are required. Results from other systems using windowing on ordinary VAX terminals (VT-100 and VT-200) show a low response time because there is no bit-mapped memory buffer. Z-II uses a menu-driven approach to accept user's request. Sometimes it is tedious to go through the nested menus.

Early in the design stage, various modules of Z-III have been left traces for integration into a sophisticated pull-down menu system, providing a flexible easy-to-use environment for both knowledge acquisition and consultation.

3.7 General Comments on Z-III

Besides the specific comparison with Z-II, a self evaluation on the implemented Z-III is given below as four "A"s.

3.7.1 Adaptability

The incorporation of fuzzy logic and fuzzy certainty in the knowledge representation and inference mechanism helps to naturally model the human language. Yet it is compatible with other rule-based systems since fuzzy reasoning is a superset of exact reasoning. Fuzzy weight would be a good tool to represent the different importance of the propositions. The system threshold and threshold expression strengthen the modelling power of Z-III. Numeric variable object is useful in knowledge engineering. In addition, retrieving from database or playback file greatly enhance the usefulness of Z-III.

3.7.2 Adequacy

The capabilities of Z-III is competitive with the commercial systems available on the market. It includes a essential set of tools to assist in different stages of expert system process, while allowing a fairly large knowledge base in the core memory with acceptable performance.

3.7.3 Applicability

This aspect measures how well Z-III can fit different kinds of problem domains. As a rule-based system, Z-III is applicable in the areas where the knowledge can be modelled as rules. The generality is enhanced by the power of fuzzy conditional inference.

3.7.4 Availability

The popularity of PCs in commercial and academic areas help to spread the use of Z-III. The performance of Z-III is tolerable even running on XTs. As it is now almost a standard that PC machines comes with at least 640K memory, Z-III can be easily installed on any site.

Chapter 4. KNOWLEDGE ENGINEERING

4.1 Techniques used in Knowledge Acquisition

The process of knowledge acquisition is to acquire the problem solving expertise from some knowledge sources and then represent and transform the expertise into the computer internal format. Potential sources of knowledge include human experts, textbooks, databases etc.

In developing an expert system, the bottleneck is usually in the knowledge acquisition phase. This is because the knowledge acquisition and transfer process in human are very complex and poorly understood. Expertise may not be expressible in language or may not be available to awareness. Some experts may not be in aware of the knowledge's significance to their activities [Gaines 1987] or even they are unable to specify which information is useful in solving the problem. Besides, expertise expressed may be irrelevant, incomplete and incorrect.

There are many techniques used in knowledge acquisition process and they are summarized below [Belkin et al. 1987], [Edgar 1988], [Eliot 1987], [Johnson et al. 1987], [Olson et al. 1987], [Weisman 1987] :

i. Interviewing the Expert

- Either informally or using structured interviewing techniques.

ii. Verbal Protocol Analysis

- Analysing recordings of experts thinking aloud as they carry out a task.

iii. Observational Studies

- Observing and recording the behaviour of the experts as they work on real problems.

iv. Studying the related documents such as textbooks and so on.

The main technique used in developing the medical expert systems is interviewing the expert. It is rather difficult for a doctor to speak loudly when he is attending to a patient. Also, it is impossible for the knowledge engineer to observe how the doctor solves his clinical problem. Thus, the techniques of verbal protocol analysis and observational studies could not be used. Moreover, medical textbooks are highly specialized and professional. It is extremely difficult for a novice to read and understand. Therefore, in this project the only possible technique used is interviewing the experts. Besides, interviewing is good for eliciting procedural knowledge.

4.2 Interviewing the Expert

The most difficult aspect of knowledge acquisition is the initial helping of the expert to conceptualize and organize the domain knowledge for the use in problem solving [Davies and Hakiel 1988]. It would be better for the experts to familiar with the development process of a medical expert system and make them show interest in computing. Before talking to the domain expert, it is important to be familiar with the subject under study. Moreover, a preparation before the interview is significant to success.

The guidelines used in the interviews are as follows :

i. Establish rapport with the expert.

- Building an expert system is a time-consuming process and there is a lot of time for the knowledge engineer to work with the expert. Establishing a good rapport with the expert is an advantage.

ii. Tell the expert that building an expert system is an intellectually challenging work.

- Avoid creating an unrealistic expectation of being easy and quick about the building of an expert system.

iii. Work out a list of questions before every interview.

- Good preparation before interview is the key to success.

iv. Avoid any interruption whenever possible.

- Thus, there was only a few interviews held in the expert's office. All the other were held in a place where there is no telephone, no other person and no other disturbance.

v. Do not stop the expert if he digresses.

- It is because digressions may give some relevant vital procedural details which could not be identified at normal question-answering process. Some knowledge was acquire when the expert digressed.

- Sometimes, when the expert digresses, he may points out the deficiency of the expert system building tool (Z-III). Thus, some constructive suggestions about the design of Z-III have been got.

vi. The interviews should be neither long nor frequent.

- It is because the expert is very busy and it will make him feel annoying. Moreover, time is needed to analyse and summarize the knowledge gained.

vii. Discuss with the expert after formulating the knowledge.

- Then, he may be able to correct or extend the interpretation.

viii. Start each interview by going through the knowledge gained from the previous session.

- The expert is free to change and even contradict anything he has previously said.

4.3 Knowledge Representation

In the beginning of knowledge acquisition process, the expert has been asked to define a list of objects which are the entity population of the domain of discourse. For each element in the list, a set of possible values, either fuzzy or crisp, are specified. This set of values should be relevant to the conclusion of the diagnosis. This list of objects are the only entities that will be used in the consultations and recognized by the expert system. Without such a limitation, the expectations placed upon the computer are infinite [Regoczei and Plantinger 1987]. Usually, the defined objects are the symptoms used to determine the diagnosis. This list will also be used to remind the physicians to perform all the necessary and relevant examinations in the future.

Then the expert is asked to suggest what it will imply if a symptom is found. Usually, an object with a 'normal' value will have no effect on any diagnosis. However, if an object has an 'abnormal' value, the expert can arrive at a particular diagnosis or a set of diagnoses. Sometimes, the expert reaches his conclusion based on the finding of a number of related symptoms. He could use the relational operators AND/OR to connect them. Then, he is requested to give the certainty of belief on each case. A certainty factor, either fuzzy or crisp, is assigned to each symptom-disease relation. However, he may sometimes disconfirm a particular diagnosis because of the existence of a particular symptom. Then a negative certainty factor is assigned to this conclusion in order to indicate its disconfirmation.

Then, the knowledge elicited from the expert is analysed and represented in the form of a production rule. It is found that the experts' congitive stimulus-response sequences are equivalent to the IF-THEN rules used in Z-III.

The experts find it easy to represent their knowledge in such form. The rule formed is shown below :

```
RULE    IF the vaginal examination is done
        AND the cervix shows bleeding
        AND the tumour is benign
        THEN the bleeding is due to BENIGN GENITAL TUMOUR
        WITH CERTAINTY ----> 0.6-0.8
```

It is assumed that the rules are sufficiently independent of one another so that the expert can always give new rules without examining the rest of the knowledge base. Such modularity is desirable because the less interaction there is among the rules, the easier and safer it is to modify the rule [Buchannan and Shortliffe 1984]. Moreover, the method of propagation of certainty factor value is simply explained to the expert in order to let him know how the conclusions are reached.

Then the extracted rules will transformed into the format used by Z-III. As I have learnt more about the subject matter and have become more familiar with the terminology used, and as the experts have acquired more and more knowledge about the structure of the medical expert system, the process of knowledge acquisition speeds up. By using production rule, the experts can even express and transform their knowledge to themselves.

4.4 Development Approach

The construction of an expert system is an incremental process. The partially built expert system will not come to an abnormal end when it encounters a case of which it has not yet been given knowledge data. Because partially completed expert systems can run, the partial results will be shown to the domain expert who can offer comments, suggestions and feedbacks during the construction phase.

In building the medical expert systems, the major steps used in the iterative process of knowledge acquisition and refinement are listed in Table 4.1 [Buchannan and Shortliffe 1984].

1.	Ask the expert what rules should be added or modified.
2.	Make changes to the knowledge base.
3.	Test for one or more old cases to see whether the new system is consistency with the old one.
4.	If any problem found, discuss with the expert and goes to step 1.
5.	Run the new modifies system on some new cases until problems are discovered.
6.	If no problem exists after running a number of cases, then stop; otherwise, goes to step 1.

Table 4.1

Figure 4.1 shows this iterative process.

4.5 Knowledge Refinement

The knowledge base of an expert system needs refinement in order to produce good and accurate diagnoses. Knowledge refinement is a 'must' in developing an expert system because it is difficult for an expert to express all the rules to cover all the special cases that arise. Moreover, the expert may make erroneous assumptions which lead to incorrect conclusions. Furthermore, the knowledge engineer may overlook or incorrectly implement some of the expert's advice. Thus, after the initial design and prototype implementation of the medical expert systems, they must further grow incrementally both in breadth and depth.

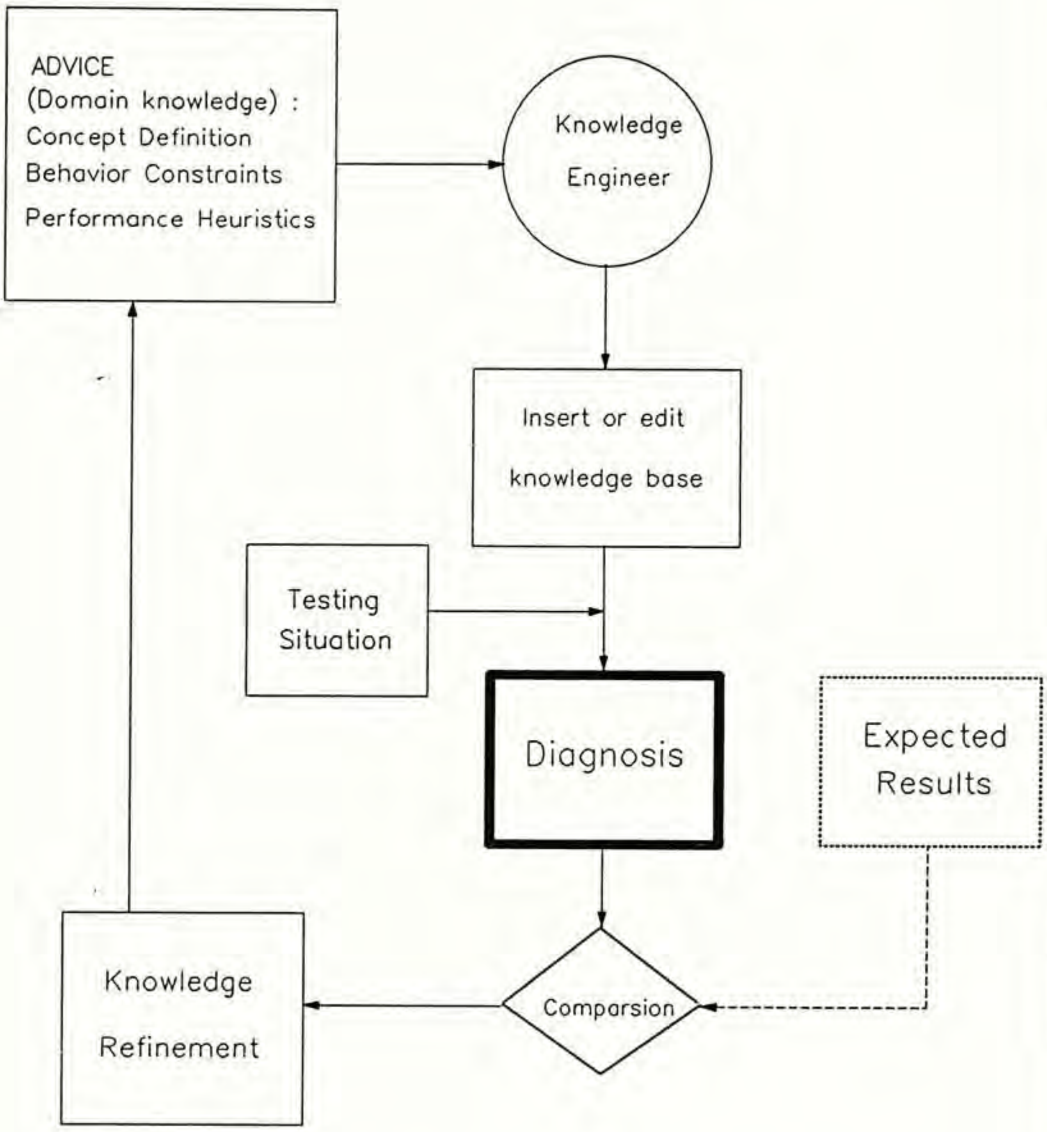

Figure 4.1. Development Approach

The following modifications on the knowledge base have been done during the development of the three medical expert systems (see section 5) :

i. Combine several related but mutually exclusive rules into one by using OR. E.g. in ABVAB

(Rule bleedcone-s7
IF (bleedcon is excessive)
THEN diagnosis is benign genital tumour)
Certainty is about 0.28

and

(Rule bleedconn-s7
IF (bleedcon is normal)
THEN diagnosis is benign genital tumour)
Certainty is about 0.28

are combined to form a new rule :

(Rule bleedconn-s7
IF ((bleedcon is excessive) OR
    (bleedcon is normal))
THEN diagnosis is benign genital tumour)
Certainty is about 0.28

The original two rules have been removed. The diagnoses are the same though the total number of rules has been decreased.

ii. Remove all the rules with certainty factor less than 0.2.

In the development of ABVAB, there are many rules with certainty factors less than 0.2. These rules will have limited impact on the value of the final certainty factors. Although, the effect of using threshold value in Z-III will ignore these rules, they will waste a lot of CPU time for calculation and a lot of primary memory to store. Thus, all of them are deleted. After the deletion, the execution time of ABVAB becomes faster and there is no change in the answer of consultation in ABVAB. Moreover, a total of around 750 rules is managed to be reduced to around 260 rules.

iii. Modify and delete the inconsistent rules

Medical consultation is a complicated process and it may involve a lot of objects and rules in the knowledge base. As the development process of an medical expert system is iterative, the expert may introduce inconsistency or incompleteness to the system. Therefore, consistency and completeness checks should be done with care to extract them. Section 4.6 will give a detailed description on consistency and completeness checks.

iv. Add new rules into the rule base.

Usually, the expert systems built are not complete. After the development of the prototype of an expert system, both real and hypothetical testing cases will be used to verify the correctness of the system. It is found that many exceptional, marginal or even common cases have not been covered. Therefore, the expert is asked to construct and add more rules to the knowledge base. However, inconsistency may occur when new rules are entered.

v, Clarify the meanings of the logical operators

In Z-III, only two logical connectives AND/OR are allowed to be used in a rule. However, the experts do not totally understand their meanings and sometimes use them incorrectly. Therefore, in order for the experts to express their knowledge more accurately, the actual meanings of the logical operators are explained to them.

For the rule : IF A AND B THEN C (CF = N1), this rule will be triggered only if both propositions A and B are satisfied and their certainty factors are greater than and equal to the system threshold. Then the result is C with certainty N1.

For the rule : IF A OR B THEN C (CF = N2), this rule will be triggered either when A is satisfied and its certainty is not less than the system threshold or when B is satisfied and its certainty is not less than the system threshold or both. But the certainty of C is N2 no matter whether only one proposition is fired or both. This means that if both A and B are satisfied, the certainty of C is still N2. The domain experts frequently misunderstand this point. They think that if both A and B are satisfied, the certainty of C should greater than N2.

The case required by the domain experts is easily handled by the evidence combination of multiple rules having the same consequence. So, the above rule can be written as :

IF A THEN C (CF = N3) and
IF B THEN C (CF = N4)

With the evidence combination of N3 and N4, the certain factor of the conclusion will be greater than N2.

4.6 Consistency Check and Completeness Check

Usually, a medical expert system is large in size. A developed medical expert system may have more than tens of objects and hundreds of rules. For example, ABVAB (see section 5.1) has 90 objects and 274 rules. In addition, the construction of a medical expert system is an incremental process and the knowledge base needs to be refined many times. In knowledge refinement, new rules may be added and existing rules may be altered. As the expert may forget the rule he has given, inconsistency and incompleteness may occur in the knowledge base.

Although Hayes-Roth had pointed out that one of the key features the rule-based systems lack is "a suitable verification methodology or a technique for testing the consistency and completeness of a rule set" [Hayes-Roth 1985], many software tools such as TERIRESIAS (for MYCIN) [Shortliffe 1976], ONCOCIN's rule checker (for ONCOCIN) [Suwa, Scott and Shortliffe 1982], INSPECTOR (for KES) [KES 1983] had been developed to identify inconsistencies and incompleteness in the knowledge base. As the consistency checking tools will help both the domain expert and knowledge engineer to build the expert system more easily, accurately and speedy, a study on different kinds of inconsistencies and incompleteness existing in nonfuzzy rule-based system will be given and then checking inconsistency in fuzzy environment will be discussed. Moreover, a algorithm of checking the consistency and completeness of a rule-based system will be proposed in the following sections.

4.6.1 The Consistency and Completeness of a nonfuzzy rule set

The following subsections will summarize the study on different kinds of inconsistencies and incompleteness on a nonfuzzy rule-based environment [Beauvieux 1988], [Buchannan and Shortliffe 1984], [Nguyen 1987], [Tsang 1988].

4.6.1.1 Inconsistency in nonfuzzy rule-based system

Let's consider the rules :

r1 : IF A1 THEN B1 (CF_1)

r2 : IF A2 THEN B2 (CF_2)

where A1 and A2 may be any combination of propositions.

i. Redundant Rules

- Two rules succeed in the same situation and have the same results.

  i.e. $A1 = A2$ & $B1 = B2$ & $\text{sign}(CF_1) = \text{sign}(CF_2)$

  or $A1 = A2$ & $B1 = \neg B2$ & $\text{sign}(CF_1) \neq \text{sign}(CF_2)$

- The two rules may cause the same information to be counted twice, leading to erroneous increases in the certainty factor of the conclusion.

ii. Conflicting Rules (Contradiction)

- Two rules succeed in the same situation but with conflicting results.

  i.e. $A1 = A2$ & $B1 = \neg B2$ & $\text{sign}(CF_1) = \text{sign}(CF_2)$

  or $A1 = A2$ & $B1 = B2$ & $\text{sign}(CF_1) \neq \text{sign}(CF_2)$

- It is a common occurrence in the rule sets. However, it may cause no problems (inconsistency) because the expert may want to conclude different values with different certainty factors. e.g. in ESROM

```
(Rule m17a
IF (((mdcs is anycd) OR
        (mdcs is gmd))
AND  (gestation >= 30)
AND  (gestation <=34) )
THEN management is delivery
Certainty is 0.7
```

```
(Rule m18
IF (((mdcs is anycd) OR
        (mdcs is gmd))
AND        (gestation >= 30)
AND        (gestation <=34) )
THEN management is observation
Certainty is 0.3
```

Management is a single-valued object with expected values (delivery, observation). The two rules listed above have the same antecedent part but with conflicting conclusions. However, no inconsistency exists.

iii. Subsumed Rules

- Two rules have the same result, but one contains additional restrictions on the situation in which it will succeed. Whenever the more restrictive rule succeeds, the less restrictive rule also succeeds, resulting in redundancy.

i.e. $(A1 \subseteq A2 \text{ or } A2 \subseteq A1)$ & $B1 = B2$ & $\text{sign}(CF_1) = \text{sign}(CF_2)$

or $(A1 \subseteq A2 \text{ or } A2 \subseteq A1)$ & $B1 = \neg B2$ & $\text{sign}(CF_1) \neq \text{sign}(CF_2)$

e.g. in ESROM

-

```
(Rule io1
IF (diagnosis is unrupt)
THEN cx is uninf)
Certainty is 0.8
```

```
(Rule ii10
IF (( diagnosis is unrupt)
AND (ctg is reactive))
THEN cx is uninf)
Certainty is 0.95
```

Whenever rule ii10 is triggered, rule io1 will also be triggered. Thus, there is redundancy.

- However, the knowledge engineer may want to write that kind of rules so that the more restrictive rules will add more weight to the conclusions. Thus, the experts should be warned and requested to clarify their meaning. In the above example, the expert wants to give more weight to 'cx is uninf' if 'ctg is reactive'. Thus, the certainty factor of rule ii10 is changed to 0.75. If both 'diagnosis is unrupt' and 'ctg is reactive' are held, the certainty of 'cx is uninf' becomes 0.95.

iv. Unnecessary IF Conditions

- Two rules have the same conclusion, an IF condition in one rule is in conflict with an IF condition in the other rule, and all other IF conditions in the two rules are equivalent.

  e.g. (A1 = $p \wedge q$), (A2 = $p \wedge \neg q$), B1 = B2 & $CF_1 = CF_2$

- The example described above actually indicates that only one rule is necessary. The second IF condition (q) is unnecessary. Although it will not cause any error in consultation, it would be better to integrate the two rules into one : IF p THEN B1(CF_1)

- e.g. (A1 = $p \wedge q$), (A2 = $\neg q$), B1 = B2 & $CF_1 = CF_2$

The second IF condition in the first rule is unnecessary, and the two rules could be combined to : IF $p \vee \neg q$ THEN B1(CF_1)

- If $CF_1 \neq CF_2$, no unnecessary IF condition occurs. It is because the expert may want to conclude a value at different certainty factor in different situations. e.g. in ESROM

```
(Rule m20
 IF ( ((wcc>10) AND (wcc<15))
  OR ( (crp>20) AND (crp<40) and
          (gestation >= 32) ))
 THEN management is delivery)
Certainty is 0.8
```

```
(Rule m21
 IF ( ((wcc>10) AND (wcc<15))
  OR ( (crp>20) AND (crp<40) and
          (gestation < 32) ))
 THEN management is delivery)
Certainty is 0.5
```

The condition (gestation >=32) in rule m20 is conflicting with the condition (gestation <32) in rule m21. However, the expert wants to assign different certainty factors to different situations.

v. Circular Rules

A set of rules is circular if the chaining of these rules in the set forms a cycle.

i.e. A1=B2 & A2=B1 & sign(CF_1)=sign(CF_2)

vi. Self-referring Rules

The condition and conclusion clause of a rule refer to the same parameter.

e.g. IF A=0 THEN A=1 (for A is not a multi-valued object)

vii. Inconsistent IF-clause

The clauses in the condition are contradict to each other.

e.g. IF A=1 and A=2 THEN B (for A is not a multi-valued object)

In vi and viii, if A is a multi-valued object, it cannot be conclude that inconsistency occurs. It is because a multi-valued object can have more than one value at anytime.

The above discussion only mentions superficial inconsistency between two rules. However, inconsistency may arise after a sequence of inferring steps. e.g.

r1	:	IF A THEN B	($CF_1 > 0$)
r2	:	IF B THEN C	($CF_2 > 0$)
r3	:	IF C THEN D	($CF_3 > 0$)
r4	:	IF A THEN D	($CF_4 > 0$)

Redundancy occurs between rule set (r1, r2, r3) and rule r4.

Moreover, the detection of circular-rule chains is affected by the threshold. The certainty factors may cause a circular chain of rules to be "broken" if the certainty factor of conclusion falls below the threshold (e.g. 0.2).

e.g.

r1	:	IF A THEN B	(CF = 0.4)
r2	:	IF B THEN C	(CF = 0.7)
r3	:	IF C THEN D	(CF = 0.7)
r4	:	IF D THEN A	(CF = 0.8)

As (0.4)(0.7)(0.7) = 0.19 < 0.2, the circular-rule chain is broken.

4.6.1.2 Incompleteness in nonfuzzy rule-based system

The development of a knowledge-based system is an iterative process. This iterative process often leaves gaps (incompleteness) in the knowledge base that both the knowledge engineer and the expert have over-looked during the knowledge acquisition process.

i. Unreferenced Attribute Values

- Some values in the set of possible values of an object's attribute are not covered by any rule's IF conditions.
- Results : rules are missing or the unreferenced attribute should be removed.

ii. Illegal Attribute Values

- A rule refers to an attribute value that is not in the set of legal values.
- As Z-III checks the object's value in the rule from the set of object's expected value, this kind of incompleteness will never occur.

iii. Unreachable Conclusions

- In a goal-driven production system, the conclusion of a rule does not match the goals or another IF condition.
- Such a rule is merely extraneous. It may affect efficiency but not the outcome because it will never be triggered.

- Finding unreachable conclusions becomes complex with certainty factors which allows the user to specify a threshold.
- Unreachable conclusion may occur if the conclusion that matches one of the IF conditions cannot be determined with a certainty factor above the threshold.
- e.g.

r1	:	IF A THEN B	(CF = 0.1)
r2	:	IF B THEN C	(CF = 1)

C is unreachable.

iv. Dead-End IF Conditions and Dead-End Goals

- The IF condition of a rule (a goal or subgoal) is not askable and it is not matched by a conclusion of one of the rules in the rule sets.
- Finding dead-end IF conditions or dead-end goals becomes complex when threshold is used.
- A dead-end IF condition or dead-end goals could occur if there is a THEN clause that concludes with a certainty factor less than the threshold (or a chain of rules that produces a certainty factor less than the threshold while they are combined).
- e.g.

r1	:	IF A THEN B	(CF = 0.4)
r2	:	IF B THEN C	(CF = 0.7)
r3	:	IF C THEN D	(CF = 0.7)

If A is known with certainty (CF = 1), D would only be known with a certainty factor of (0.4)(0.7)(0.7) = 0.19 < 0.2 (threshold).

If D is not askable, D is dead-end.

4.6.2 Consistency Checks in Fuzzy Environment

Checking inconsistency and incompleteness in fuzzy rule-based environment is not as simple as that in non-fuzzy environment.

Let height be a fuzzy object which has three fuzzy terms (tall, medium, short) as its expected values. If the user enters the following 2 rules :

r1 : IF height is tall THEN B

r2 : IF height is not short THEN B

Assume both CFs > 0, does redundancy occur ?

Consider another case :

r1 : IF A THEN height is rather tall

r2 : IF A THEN height is not very tall

Assume both CFs > 0, does conflict occur ?

Thus, it would be better if we could apply a quantitative measure on the two fuzzy terms (e.g. tall, not short) in order to determine to what extent the two fuzzy terms mean the same thing.

4.6.2.1 Affinity

Let the knowledge base contain the rule : IF A THEN height is tall.

If the user wants to create another rule : IF A THEN height is not medium. Does this additional rule introduce inconsistency (conflict, redundancy or subsumption) to the knowledge base? The three measures described in section 3.4.2 are not sufficient to conclude whether the two consequent parts are similar. It is because

M(not medium | tall) = 1 but M(tall | not medium) = 0.5

Therefore, we can see that the order of rule creation will influence the value of similarity. Owing to the above limitation, we introduce an Affinity (A) measure.

The Affinities of propositions p and q are defined as follows :

$A(p,q) = M(p \wedge q \mid p \vee q)$

where $\mu_{p \wedge q}(w) = \min(\mu_p(w), \mu_q(w))$,

$\mu_{p \vee q}(w) = \max(\mu_p(w), \mu_q(w))$ and

$\mu(w)$ is the membership function of w in universe of discount.

A(p,q) measures the similarity of $p \vee q$ given $p \wedge q$. If p and q are identical, A(p,q) will be equal to 1 (because $p \vee q = p \wedge q$). One of the important properties of A(p,q) is commutative. i.e. A(p,q) = A(q,p). Thus, the order of rule creation has no influence in the calculation of affinity. Thus, it is a good tool to check the consistency of the knowledge base.

In order to investigate the suitability of affinity in consistency check, we find out all the results of affinity with different p and q (similar to those described in section 3.4.2 with p replacing F_1 and q replacing F_1'). The results are similar to section 3.4.2 and they can be divided into 4 groups :

1) A(p,q) = 1

e.g. A(tall, tall) = 1

A(very medium, medium) = 1

2) A(p,q) > 0.5

e.g. A(tall, very tall) = 0.9305

A(rather short, short) = 0.9276

A(rather medium, very medium) = 0.9272

3) $A(p,q) = 0.5$, $P(p \wedge q \mid p \vee q) = 1$ and $N(p \wedge q \mid p \vee q) = 0$

e.g. A(tall, not medium) = 0.5

A(not short, medium) = 0.5

A(not very tall, rather short) = 0.5

4) $A(p,q) < 0.5$, $P(p \wedge q \mid p \vee q) < 1$ and $N(p \wedge q \mid p \vee q) = 0$

e.g. A(tall, not tall) = 0.1720

A(rather tall, medium) = 0.0837

A(short, tall) = 0

If the affinity of the two terms falls into group 1 or 2, we conclude that they both point to the same or similar (to some extent) concept. If the affinity is in group 4, we say that the two terms are different. If the affinity is in group 3, we cannot conclude anything.

The definition of Affinity could be extended to detect the degree of matching of the antecedent part of two rules.

Let $A1 = p_1 \wedge q_1 \wedge r_1 \wedge \ldots \wedge z_1$

$A2 = p_2 \wedge q_2 \wedge r_2 \wedge \ldots \wedge z_2$

Then $A(A1, A2) = A(p_1, p_2) * A(q_1, q_2) * A(r_1, r_2) * \ldots * A(z_1, z_2)$

If p_1 and p_2 are nonfuzzy and $p_1 = p_2$, then $A(p_1, p_2) = 1$; otherwise, $A(p_1, p_2) = 0$.

4.6.2.2 Detection of Inconsistency and Incompleteness in Fuzzy Environment

By using affinity, we could detect inconsistency and incompleteness in fuzzy environment. The detection is the same as those described in section 4.5.1 except that we use affinity to measure the likeness of two fuzzy terms.

i.e. (A1 = A2) could be replaced by A(A1, A2) > 0.5 in fuzzy environment and

(A1≠A2) could be replaced by A(A1, A2) < 0.5 in fuzzy environment.

4.6.3 Algorithm for Checking Consistency

As mentioned in section 4.5, the knowledge base created by the experts are full of inconsistency and incompleteness. Checking consistency and completeness of a knowledge base is the duty of both knowledge engineers and domain experts. *Affinity* can be used to determine the degree of matching between two propositions. An algorithm for checking the four kinds of inconsistency (redundancy, conflict, subsumption and unnecessary-if) of a rule-based expert system like Z-III is shown in table 4.2.

```
/* Determine whether the GOAL list is empty */
IF GOAL = NULL
THEN RETURN
ELSE OBJECT := GOAL /* OBJECT points to the head of GOAL list */
ENDIF

WHILE OBJECT ≠ NULL
DO
    /* P points to the RULE_UPDATED list of OBJECT */
    P := OBJECT.RULE_UPDATED

    /* Is the RULE_UPDATED list not empty ? */
    WHILE P ≠ NULL
    DO
        /* Q points to the next node for comparison */
        Q := P^.NEXT
        WHILE Q ≠ NULL
        DO
        /* COMPARE will determine the relation with the two antecedences
            in P and Q */
            RESULT := COMPARE(P^.ANTE_PART, Q^.ANTE_PART);

            CASE RESULT
                /* The two antecedences are equal or similar */
                EQUAL :
                    IF SIGN(P^.CF) = SIGN(Q^.CF)
                    THEN REDUNDANT := APPEND(REDUNDANT, P,Q)
                    ELSE CONFLICT := APPEND(CONFLICT,P,Q)
                    ENDIF

                SUBSET :
                /* The one antecedence is subset of the other */
                    APPEND(SUBSUMED,P,Q);

                CONFLICT_IN_ONE_CONDITION :
                /* Two antecedences are equal or similar except for
                    one condition */
                    APPEND(UNNECESSARY_IF,A,B)
            ENDCASE

            Q := Q^.NEXT
        ENDWHILE
        P := P^.NEXT
    ENDWHILE
    GOAL_OBJECT := GOAL_OBJECT^.NEXT
ENDWHILE
```

Table 4.2

APPEND is a routine to append the pair of rules like (P,Q) into the list indicated by the first parameter.

COMPARE is a function to determine the relation with the two antecedences

REDUNDANT, CONFLICT, SUBSUMED and UNNECESSARY_IF are linked list.

Chapter 5. FUZZY MEDICAL EXPERT SYSTEMS

Three medical expert systems built under a real life environment are described in the following sections.

5.1 ABVAB

5.1.1 General Description

ABVAB is a medical expert system which diagnoses the cause of ABnormal VAginal Bleeding from the past history and the results of physical examinations of a patient. There are three phases in the development of ABVAB. The first phase based on historical data has already been implemented [Leung et al. 1988]. The second phase of ABVAB, in which the knowledge on physical examinations are incorporated, is implemented in this project. The medical knowledge is elicited from a gynaecologist. In these two phases, ABVAB was first implemented using Z-II on a VAX11/780 machine. After the development of Z-III has been finished, ABVAB is transferred from a mini-computer environment to a PC environment.

The medical knowledge is represented as production rules. Fuzzy types such as freshness of blood are used in the histories to represent the imprecise concepts and nature of the history. Fuzzy or non-fuzzy certainty factors are attached to individual rule to indicate the degree of confirmation. More than one possible diagnosis, given in preference order, are suggested as the causes of abnormal bleeding by the expert system. The first preference diagnosis is the one with the highest certainty factor. The working features of the first phase of ABVAB can be found in [Leung et al. 1988], [Leung et al. 1989] and will not be further elaborated here. The second and third phase of ABVAB will be discussed in details below.

5.1.2 Development of ABVAB

In the first phase of development, ABVAB performs diagnosis only based on the historical data of the patient profiles and symptoms. Although, the initial results of ABVAB were quite satisfactory, it is necessary to suggest further tests to confirm the results. Diagnosis based on historical data only is not reliable because it depends on :

a. intelligence of patients,
b. stage of disease or severity of disorder,
c. amount of misleading history, and
d. cultural background of the patient.

Thus, a second stage of rules for diagnosis based on the physical examinations will be added to ABVAB in order to :

a. confirm the clinical suspicion arising from history taking,
b. detect any abnormality that may not be obvious from the history, and
c. add more weights to certain diagnoses arising from the history taking.

The construction of the second phase of ABVAB is an incremental process and the knowledge base of ABVAB has been refined for several times, including both the rules related to history and physical examination. As ABVAB is a diagnosis system, it can be effectively built by a rule-based knowledge representation and a backward-chaining inference mechanism.

Although the diagnosis of ABVAB depends on the history and physical examination, ABVAB is only a one-layer medical expert system. Both history and physical examination contribute towards the diagnoses but do not have interaction. Thus, in order to evaluate the degree of importance of history and physical examination, two analyses have been carried out and will be described in section 5.1.4. The one-layer structure of ABVAB is depicted in figure 5.1.

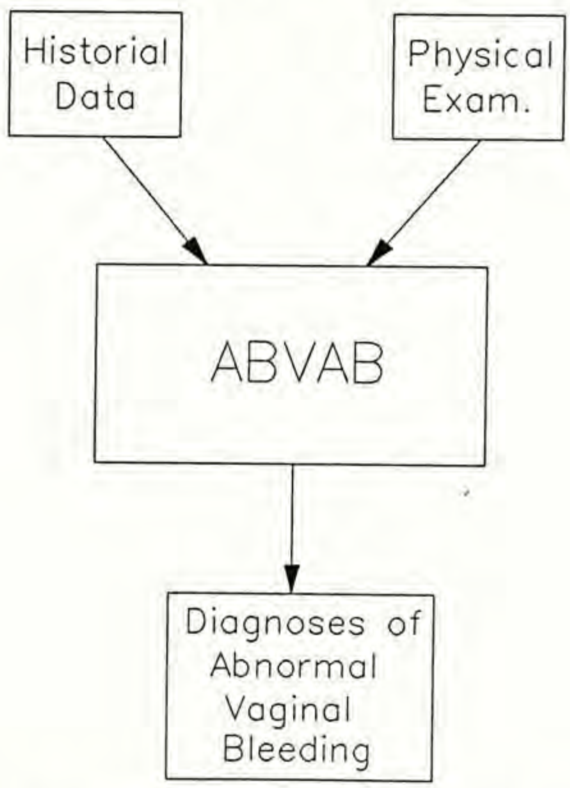

Figure 5.1 One-layer Structure of ABVAB

5.1.3 Computerisation of Database

ABVAB performs diagnosis mainly based on the historical data and physical examination of a patient. The structure of ABVAB is shown in figure 5.2.

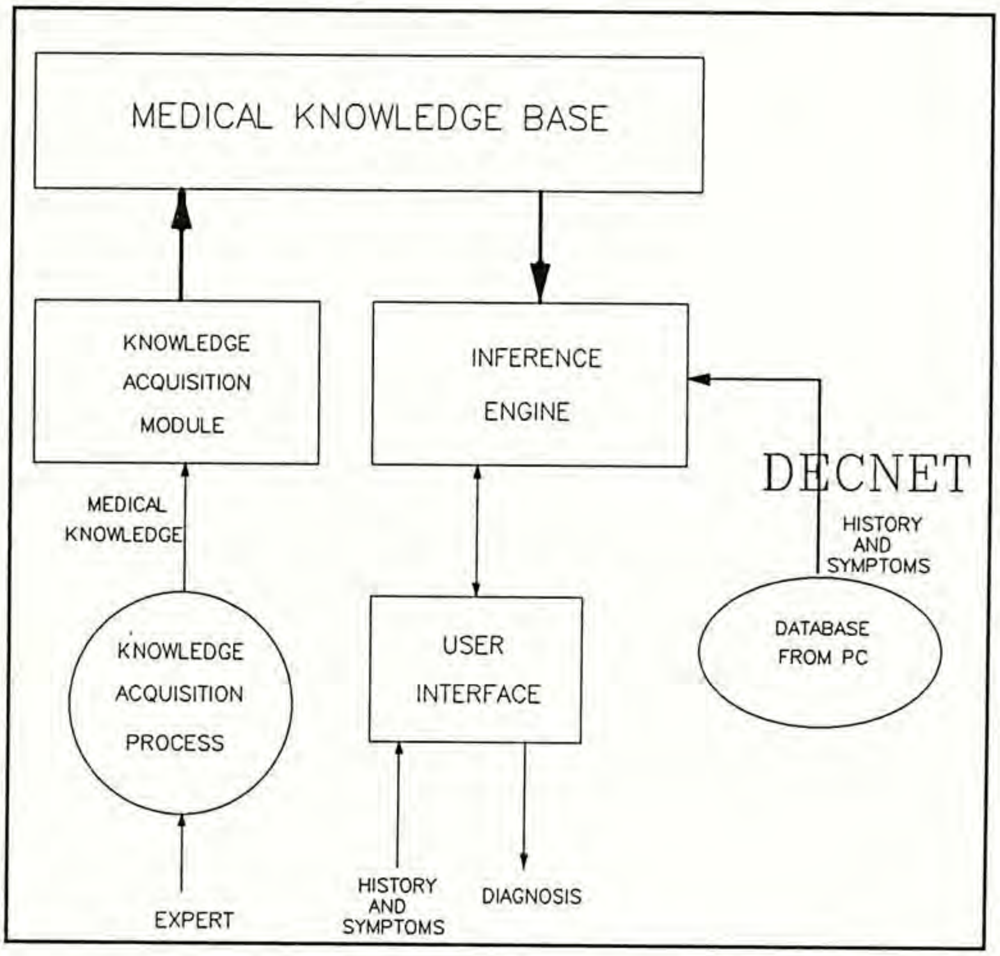

Figure 5.2 Structure of ABVAB

During a consultation, the user needs to enter both the historical data and physical examination data to ABVAB. Owing to the large quantity of data needed to be keyed in for each consultation, it is time-consuming and error-phone. A program written with dBase III Plus on a PC has been implemented in order to gather and record the patients symptom. Then, the doctor can insert or edit the data easily and ABVAB can extract the required data directly and efficiently from the database.

The program is implemented to assist the users of ABVAB to collect and store all the necessary and relevant data, and to make the data available to ABVAB. Thus, the program should :

i. be easy to use,
ii. be user friendly,
iii. provide help facilities to user when needed,
vi. be fast,
v. guarantee that no unexpected value are entered, and
vi. allow the user to input the degree of certainty for each data found.

Based on the above criteria, the design of the database program is as follows :

i. When the value of an object is being entered, a list of predetermined values will be shown in a window (Figure 5.3). The user can only select the value from this list guaranteeing that no unexpected value is entered.
ii. The user chooses the value from the list by pressing the corresponding number. The system will automatically assign the value to that object. This will eliminate all typing error.

iii. Values of the certainty factors of the facts have to be entered. The user may type in a crisp number, e.g. 0.4, or a fuzzy expression, e.g. around 0.7 or 0.4 - 0.6. Syntactic errors are checked in this stage to guarantee a correct format.

ABVAB PHYSICAL EXAMINATION DATA		
GENERAL CONDITION	:	1. good 2. fair 3. poor
2nd SEXUAL CHARACTER	:	
BREAST	:	
THYROID GLAND	:	
NECK AND BODY LYMPH GLAND	:	

Figure 5.3 Screen Layout of Database Program

iv. Easy modification of data is designed

v. For each object, a default value is assigned. Thus, if the doctor finds that the patient is normal, he just has to bypass that part of program.

vi. The program is compiled in order to have a faster speed.

vii. Help facilities have been provided for the users having troubles on running the program. When they press a key, a help message will be displayed.

The program is implemented in an IBM PC/XT/AT/386 environment. After all the data are entered, the databases will be reformatted in the manner required by ABVAB. Thus a direct link between the database system and the expert system can be established.

5.1.4 Results of ABVAB

A consultation with ABVAB involves making numerous inference using the expert rules stored in knowledge base and facts from the medical database. The data can be extracted from a database or on-line data supplies by users during a consultation. The following example is the facts used in a consultation obtained directly from a real medical database :

1. the identity no of the patient is a865932 (1.0)
2. other findings are false (1.0)
3. vaginal examination is done (1.0)
4. rectal examination is not-done (1.0)
5. uterus motility is mobile (1.0)
6. uterus is enlarged (1.0)
7. it (uterus is enlarge) is regular (1.0)
8. perineum is with tumour is false (1.0)
9. the vulva is with tumour is false (1.0)
10. abdominal tenderness is false (1.0)
11. abdomen is normal (1.0)
12. breast is pregnant breast (1.0)
13. infertility present is unknown
14. vaginal discharge is unknown
15. pelvic pain present is false (1.0)
16. dysparunia present is false (1.0)
17. dymenorrhea present is false (1.0)
18. the bleeding condition is slight (1.0)
19. the maximum cycle length (in days) is 30.0 (1.0)
20. body weight (in kg) is 54.0 (1.0)
21. the marital status is m (1.0)

22. the months for duration of abnormal bleeding is 36.0 (1.0)
23. vagina is normal (1.0)
24. cervix is bleeding (1.0)
25. it (cervix or vagina shows tumour) is unknown
26. kidneys is not-palpable (1.0)
27. spleen is not-palpable (1.0)
28. liver is not-palpable (1.0)
29. heart is normal (1.0)
30. thyroid gland is normal (1.0)
31. sexual character is developed (1.0)
32. urine sugar is nil (1.0)
33. change in libido is unknown
34. anorexia is unknown
35. anxiety is unknown
36. insomnia is unknown
37. family problem is unknown
38. occupation changed is unknown
39. tubal ligation is false (1.0)
40. taking hormones other than pills is false (1.0)
41. taking drugs for menstrual regulation is false (1.0)
42. present contraception method is condom (1.0)
43. the last menstrual period (in days) is 26.0 (1.0)
44. midcycle intermenstrual bleeding present is unknown
45. irregular cycle present is true (1.0)
46. significant weight loss is false (1.0)
47. significant weight gain present is false (1.0)
48. perineum is normal (1.0)
49. inflamed perineum is false (1.0)
50. ulcerated perineum is false (1.0)
51. vulva is normal (1.0)
52. inflamed vulva is false (1.0)
53. ulcerated vulva is false (1.0)
54. chest is normal (1.0)
55. neck and body lymph is not-palpable (1.0)
56. general condition is poor (1.0)
57. urine protein is nil (1.0)
58. pyrexia temperature present is false (1.0)
59. gyn cancer in close relative is false (1.0)
60. history of cancer present is false (1.0)
61. live birth is 3.0 (1.0)
62. abortion number is 0.0 (1.0)
63. pregnancy present is false (1.0)
64. postcoital bleeding present is false (1.0)
65. bowel symptoms present is false (1.0)
66. urinary symptoms present is false (1.0)

where the numbers inside the brackets are the certainty factors of the corresponding facts.

The conclusion for the diagnosis of abnormal bleeding in preference order is shown in Table 5.1 :

It is extremely certain (0.968583) that the diagnosis should be dub
It is very certain (0.952298) that the diagnosis should be pelvic infection
It is very certain (0.942436) that the diagnosis should be malignant genital tumour
It is indeed certain (very close to 0.917833) the diagnosis should be that benign genital tumour
It is pretty certain (0.8) that the diagnosis should be complications of pregnancy
It is quite certain (0.715365) that the diagnosis should be hormone induced
It is almost certain (0.6) that the diagnosis should be foreign body in vagina
It is somewhat certain (0.5824) that the diagnosis should be endometriosis
It is somewhat certain (0.526) that the diagnosis should be genital injury
It is little certain (0.315) that the diagnosis should be iucd

Table 5.1 Conclusion of ABVAB

The goal of ABVAB (diagnosis) is a multi-valued object so that more than one conclusions are drawn. The certainty factors shown in the conclusion in table 5.1 are used as relative possibility for comparison amongst the causes of abnormal bleeding and they should not be interpreted as the absolute certainty values.

A total of 200 patients are recruited in the feasibility study in the first phase and another 44 patients are used in the analyses in the second phase of development. The accuracy in the diagnosis based on the histories of 200 patients alone is presented in table 5.2. The proportion of various diagnoses in the second phase are listed in table 5.3.

CORRECT DIAGNOSES	Percentage
as the first choice	68
as the first three choices	90
listed	100

Table 5.2

DIAGNOSIS	Percentage
DUB	56.8
benign genital tumour	15.9
malignant genital tumour	11.4
pregnancy of complications	13.6
endometriosis	2.3

Table 5.3

Table 5.4 details the preference order of the expected diagnoses by ABVAB based on history data alone, and combined history and physical examination.

Case Number	History alone	History & P.E.
1	2nd	2nd
2	2nd	4th
3	3rd	1st
4	1st	1st
5	1st	1st
6	1st	1st
7	4th	2nd
8	1st	1st
9	1st	1st
10	1st	1st
11	1st	1st
12	6th	6th
13	1st	1st
14	1st	1st
15	1st	1st
16	1st	3rd
17	1st	3rd
18	2nd	2nd
19	1st	1st
20	4th	4th
21	1st	1st
22	1st	1st
23	1st	1st
24	5th	1st
25	1st	1st
26	1st	1st
27	1st	1st
28	5th	4th
29	1st	1st
30	5th	3rd
31	3rd	2nd
32	1st	1st
33	1st	1st
34	1st	1st
35	1st	1st
36	1st	1st
37	1st	1st
38	1st	1st
39	2nd	2nd
40	1st	1st
41	2nd	1st
42	4th	3rd
43	1st	1st
44	3rd	2nd

Table 5.4 Comparison Results of the Two Phase of ABVAB

From the results, it can be concluded that ABVAB will give more accurate diagnosis using both historical data and physical examination data (30/44 instead of 28/44 correctly give first preference). It is discovered that some expected diagnoses will never be within the first five choices. e.g. iucd, haemateria and rectal bleeding. It is because they only appear in the consequent part of a few rules and have a small certainty factor value, from 0.21 to 0.35. Thus, it is necessary to further refine the knowledge of ABVAB in order to produce a better result. In spite of these limitations, the overall testing results are quite satisfactory.

In order to find out the relative significance of historical data and physical examination data to the diagnosis problem, two analyses have been carried out. In the first one, the certainty factors of all the rules related to the physical examination data are lowered by some factors while the certainty factors of the rules related to historical data remain unchanged. This will lower the relative significance of physical examinations in reaching a diagnosis. The second analysis is similar to the first one except that the roles between historical data and physical examination data are exchanged. Figures 5.4 and 5.5 show the results of the first and second analyses respectively. The performance indices in both figures are the ratios of the number of cases which are in the first and the first three preference levels to the total number of cases respectively.

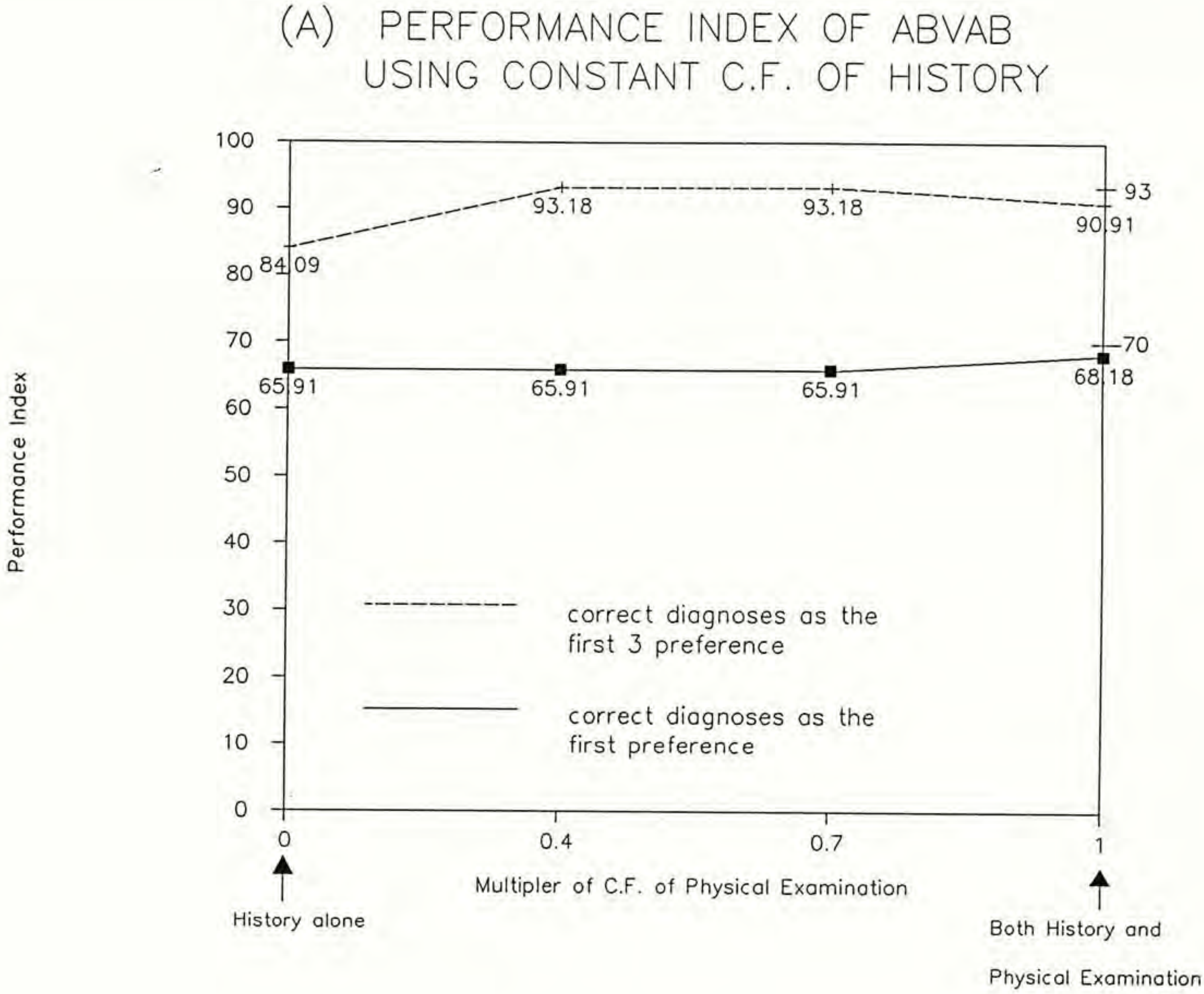
(A) PERFORMANCE INDEX OF ABVAB
USING CONSTANT C.F. OF HISTORY
Performance Index
100
90
80
70
60
50
40
30
20
10
0
84.09
93.18
93.18
90.91
93
65.91
65.91
65.91
68.18
70
correct diagnoses as the
first 3 preference
correct diagnoses as the
first preference
0
0.4
0.7
1
Multipler of C.F. of Physical Examination
History alone
Both History and
Physical Examination

Figure 5.4.

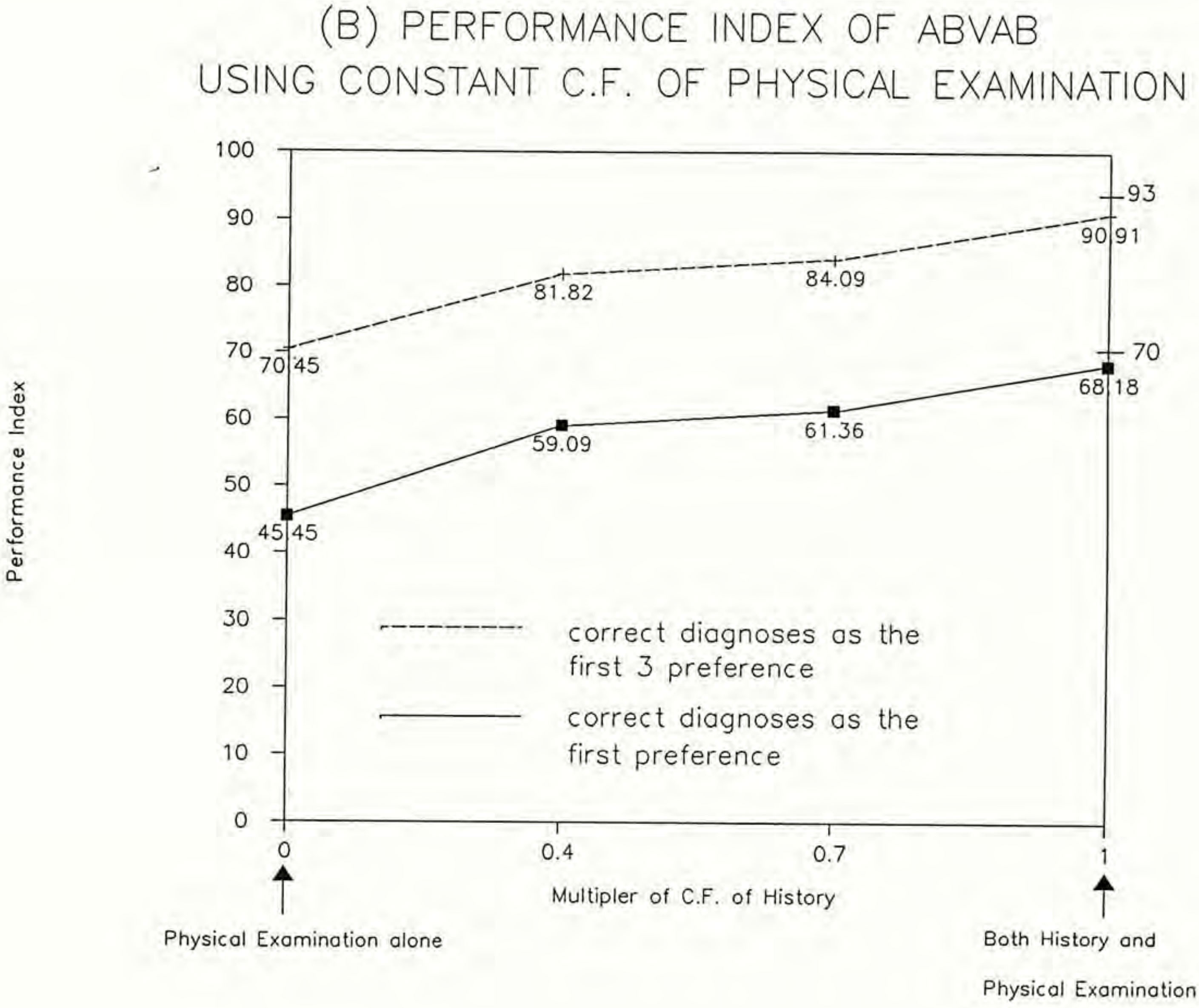

Figure 5.5.

Based on this finding it can be concluded that ABVAB will give more accurate diagnosis by using both historical data and physical examination data than by histories alone. The overall result suggested that both history and physical examination are important in making the diagnosis. Lowering the weight of any one of them will result in an unsatisfactory conclusion. Although the diagnosis with historical data bearing slightly heavier weights, the overall performance is rather good.

5.1.5 From Minicomputer to PC

In the third phase of development, ABVAB is successfully ported from Z-II to Z-III. There are 90 objects, 274 rules and 6 fuzzy types (such as the term "excessive" in the fact "The vaginal discharge condition is very excessive."). The whole knowledge base occupies about 250K of core memory, mostly occupies by the rules.

The new ABVAB on Z-III is tested with 12 real cases derived from patient records. The overall result is shown in table 5.5. Table 5.6 shows the detailed result of case 10.

PREFERENCE ORDER	Z-III	Z-II
1st preference within first 3 preference within the preference list	9 (75%) 10 (83%) 12 (100%)	9 (75%) 10 (83%) 12 (100%)

Table 5.5 Successful Rate of Z-III and Z-II in ABVAB

case #10 : patient id = G277311

	Z-III	Z-II
1st choice	pregnancy complications (0.95)	pregnancy complications (0.92)
2nd choice	malignant (0.92)	malignant (0.92)
3rd choice	benign (0.86)	benign (0.81)
Actual cause in the record	same as 1st choice	same as 1st choice

Table 5.6 Comparison results

Although the sample size is small, it can still be seen that the performance of ABVAB on Z-III is quite satisfactory and the porting from Z-II to Z-III is successful. 75% of the actual causes of the cases have been diagnosed as the first preference by Z-III, and 83% of them have been deduced within the first three preference-listed diagnoses. Although the remaining of them are not included in the first three, the important point is that none of the real causes have been left out in the list of the diagnoses.

From tables 5.5 and 5.6, it can be seen that the performance of Z-III is very similar to that of Z-II, with only some minor differences in certainty factors. The differences are probably due to the minor modifications of the inference engine during the development of Z-III. However, the response time of ABVAB in Z-III is faster and the accessibility is also higher than those in Z-II (see section 3.6).

5.2 INDUCE36

5.2.1 General Description

The domain of INDUCE36 concerns the decision on the induction of labour in pregnant patients after 36 weeks of gestation in obstetrics. Induction of labour is the action by which the delivery process of babies is started artificially. It is required because there are conditions in which the continuation of pregnancy is more risky than delivering the baby, either for the baby or the mother. On the other hand, delivery of an immature fetus carries significant risks to the baby. Unnecessary induction when conditions are unfavourable may also end up in caesarean sections. Abdominal deliveries is a relatively high risk procedure and is a major life-event for women. When induction of labour is considered, careful appraisal of the duration, risk factors of the pregnancy, and whether pointers for maternal or fetal compromise are present. After the consultation of INDUCE36, the recommended management (delivery or observation) and its certainty are resulted.

INDUCE36 is built on Z-III and evaluated by an original method using 30 hypothetical cases of varying difficulty. Its performance is compared to that of 6 doctors. 3 blinded consultants examine each set of recommendations (7 in total) and then the rank of INDUCE36 can be deduced. Next two sections will give a detailed description on the verification method used in INDUCE36 and the results.

5.2.2 Verification of INDUCE36

Frequently, medical expert systems are verified by using either clinical or hypothetical cases. As medical diagnoses are too complicate and involve too many factors, it is not easy to confirm a diagnosis. Sometimes, different doctors may give different diagnoses for the same patient. Therefore, in order to determine whether INDUCE36 is a powerful medical expert system, a new verification approach is introduced.

The medical team consists of 4 ranks of staff. They are consultants, senior registrars, registrars and interns. Consultants in medicine are very experienced doctors who are given the authority to handle patients under nobody's supervision. Their status makes them capable of employing their own staff. Senior registrars are relatively senior doctors processing specialist qualifications working as deputies of consultants and are delegated responsibility of day-to-day management of public patients. Registrars are supervised by senior registrars to make simple decisions. They are in training for their specialist qualification. Interns are newly graduated doctors from medical schools in the first year of medical training. INDUCE36 has been constructed and verified by 3 independent consultants.

The basis of the verification exercise is based on the consultants' power to employ their own staff, and that they are involved in professional examinations to judge which registrars can be prompted to senior registrars. It would be less logical to ask them to judge the performance of people of their same rank, namely consultant.

In the verification of INDUCE36, 6 doctors in different ranks (2 interns, 2 registrars and 2 senior registrars) are asked to diagnose 30 hypothetical cases. Then their answers together with the results given by INDUCE36 are given to 3 consultants who are then asked to determine the rank of the persons who complete the questions. As the consultants do not know which set of answer is given by INDUCE36, the rank of INDUCE36 can be deduced.

5.2.3 Results

The results obtained from the verification process of INDUCE36 are given in table 5.7.

	Assessed Position		
Actual Position	Senior Registrar	Registrar	Intern
Senior Registrars (2)	2	3	1
Computer	3	0	0
Registrars (2)	5	1	0
Interns (2)	2	3	1

Table 5.7 Results of Verification Process of INDUCE36

From above, senior registrars are ranked 2, 3 and 1 times as senior registrar, registrar and intern respectively. The computer has been consistently assessed by the consultants as senior registrars. The real senior registrar did not do very well. One of the 2 senior registrars made several mistakes in misreading information and is therefore ranked low. A possible explanation is that they are accustomed to the verbal reporting with clarification from the junior staff but not reading through reports about patients. Therefore, they may miss important information when they read printed or written data. Registrars and interns are ranked relatively high because they are working in a very busy unit handling more than 7000 deliveries a year, with routine work divided only between 7 registrars and 6 interns. Induction of labour is so commonly encountered that they can deal with the problem properly on their own. In addition, they are trained up in the same unit and therefore their ideas would conform to those of their consultants even more than the senior registrars.

The main concern of this study, however, is how INDUCE36 performs. It is encouraging that the consultants rank INDUCE36 high in performance. From the results, it can be concluded that medical expert systems eliminate human fatigue or bias in suggesting patient management. INDUCE36 is a successful medical expert system in obstetrics.

INDUCE36 has only one goal 'management' with (delivery, monitor) as expected value. The knowledge base of INDUCE36 contains 40 objects, 81 rules and 6 fuzzy types. The rule base of INDUCE36 is listed in Appendix III.

5.3 ESROM

5.3.1 General Description

ESROM (Expert System on Rupture Of Membranes) is a medical expert system in the domain of the diagnosis and management of rupture of membranes in obstetrics. Rupture of membranes occurs when the fetal membranes break and amniotic fluid inside the uterus leaks from the cervix. It is sometimes a sign of labour and delivery. However in other times the condition occurs before labour. It is then associated with an increased chance of infection of the fetus, which carries sinister complications. On the other hand, when the fetus is very premature, there is a high chance of mortality and morbidity when the baby is delivered immediately. The problems lie mainly in the baby's inability to maintain its respiration when the lung is very immature. Other problems of the brain, eyes and feeding difficulties may also arise. Therefore a delicate balance needs to be maintained in how long the pregnancy is prolonged and when it should be delivered.

The system has three goals as follows :

a, Diagnosis - a single-valued object with expected values :
membrupt (membrane is ruptured) and
umembrupt (membrane is not ruptured) and
is used to decide whether the membranes are ruptured or unruptured.

b, Infection - a yes-no type object and is used to indicate whether infection of the fetus is present.

c, Management - a single-valued object with expected values :
delivery and observation and is responsible to decide whether fetus should be delivered.

All these three goals are single-valued and their order in consultation is diagnosis, infection and finally management.

5.3.2 Multi-layer Medical Expert Systems

The goals in ESROM are highly related to each other. The inference of infection depends on the result of diagnosis. Similarly, the inference of management depends on the results of both diagnosis and infection. As a result, ESROM is built as a multi-layer (3-layer) medical expert system. Figure 5.6 shows the multi-layer structure of ESROM.

From figure 5.6, A subset of symptoms S_1 are input to I_1 that produces the result of diagnosis. After giving the diagnosis to the user, S_1 and diagnosis together with more symptoms S_2 are input into the second level of inference I_2. The output of I_2 is infection. Then, S_1, S_2, diagnosis and infection together with the other symptoms S_3 are put into a third level of inference I_3 to obtain the answer of management.

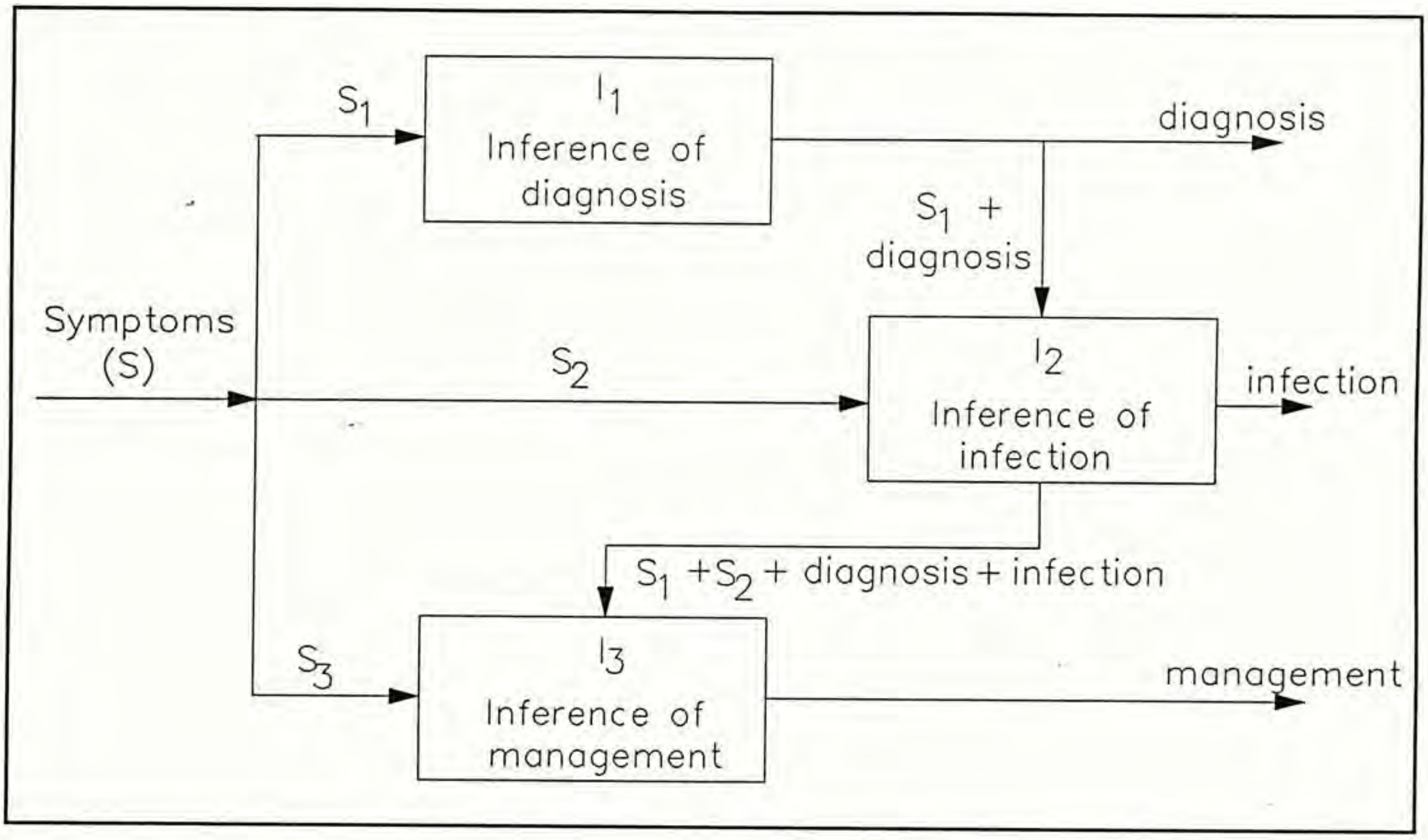

Figure 5.6 Multi-layer structure of ESROM

Table 5.8 lists out some rules which show the relation between the three goals.

(Rule i06 IF (white cell count > 10, 15) and (diagnosis is membrupt) THEN infection is yes) Certainty is 0.45	(Rule m02 IF (gestation > 36) and (diagnosis is membrupt) THEN management is delivery) Certainty is 0.9
(Rule m21 IF (diagnosis is umembrupt) and (infection is no) THEN management is observation) Certainty is 0.95	

Table 5.8

5.3.3 Results

By using Z-III, the multi-layer structure of ESROM can be implemented easily. After the knowledge engineer defines the goals in the right order, Z-III performs its inference as described above automatically.

In the current stage of development in ESROM, there are 43 objects, 61 rules and 4 fuzzy types (size, wellbeing, amount and moisture). Thirty hypothetical testing cases are used to verify the knowledge base and the results agrees with the domain experts. As ESROM is a very complicated medical expert system, future fine tuning on the knowledge base is necessary and more rules are needed to be added. The rule base of ESROM is listed in Appendix IV.

In the further development of ESROM, a fourth goal may be added which is used to decide the possible methods used to deliver the baby. As a conclusion, building ESROM is a challenging and difficult job.

Chapter 6. CONCLUSIONS

Building medical expert systems is a challenging and sophisticated process. By using an expert system shell, both the knowledge engineer and domain expert need not worry about data manipulation and knowledge representation. Only the relevant domain knowledge is required to be specified. Moreover, the algorithms, procedures or control structure will also be handled by the shell. Thus, both the knowledge engineer and domain expert can concentrate on the knowledge acquisition and knowledge refinement processes.

Although Z-III is very suitable for medical applications, it is a rule-based expert system shell which can be used in other fields such as geography and engineering. The capability of Z-III to handle mixed exact and inexact reasoning leads to its success. The reason is that fuzziness and uncertainty are frequently present in our daily natural language which is used by domain experts and end users to express the knowledge and facts. The newly introduced features of weighting, use of threshold, fuzzy matching and database retrieval have greatly enhanced the representation and inference power of Z-III. Building medical expert systems on Z-III has been proven to be efficient and simple. Furthermore, the availability of the medical expert systems has also been improved a lot because Z-III is developed in a PC environment.

In building the medical expert systems, much of the time is spent on the knowledge acquisition and refinement processes. Interview seems to be a most useful technique. An improved knowledge acquisition procedure has been introduced and used in building the three medical expert systems in this thesis. Rule base representation has been proven to be suitable for storing the knowledge on medical diagnosis. Consistency and completeness checks are important and necessary. It would be better to automate the checking process by implementing a consistency and completeness checker using the affinity measure newly developed in the thesis to determine the degree of matching of two propositions.

The results obtained by ABVAB encourage the development of other medical expert systems on Z-III. The verification method used in INDUCE36 introduces a new approach to evaluate a medical expert system. In order to obtain better statistics in the future, more doctors will be asked to perform the diagnosis and more consultants or specialists will be invited to rank the results of INDUCE36. The multi-layer structure of ESROM demonstrates a way to handle the complicated medical diagnoses and treatments. It is hoped that as the physical memory of PC is increased rapidly, more complex multi-layer expert systems can be developed by Z-III in the future and expert system applications can be more widely accepted in various fields.

REFERENCES

Barnett, G.O., (1982), "The Computer and Clinical Judgement", New England Journal of Medicine, 307, 493-494

Basu, A. and Dutta, A. (1986), "Computer Based Support of Reasoning in the Presence of Fuzziness", North-Holland Decision Support Systems, 2, 235-256

Beauvieux, A. and Dague, P., (1988), "Interactive Checking of Knowledge Base Consistency", Internation Computer Science Comference, 567-574

Berkin, N.J., Brooks, H.M. and Daniels, P.J. (1987), "Knowledge Elicitation Using Discourse Analysis", Int. J. Man-Machine Studies, 27,127-144

Buchannan, B.G. and Shortliffe, E.H. (1984), "Knowledge Engineering", Rule-Based Expert Systems, Addisley-Wesley Publishing Company.

Buckley, J.J. and Tucker, D. (1987), "Extended Fuzzy Relations : Application to Fuzzy Expert Systems", International Journal of Approximation Reasoning, 1, 177-195

Buisson, J.C. and Farreny, H. (1985), "The Development of a Medical Expert System and the Treatment of Imprecision in the Framework of Possibility Theory", Information Science, 37, 211-226

Cayrol, M., Farreny, H., and Prade H. (1982), "Fuzzy Pattern Matching", Kybernetes, Vol. 11, 103-116

Davies, M. and Hakiel, S. (1988), "Knowledge Harvesting : A Practical Guide to Interviewing", Expert System, Feb., Vol. 5, No. 1

Dubois, D. and Prade, H., (1980), Fuzzy Sets and System : Theory and Applications, Academic Press : New York

Dubois, D. and Prade, H., (1988), "The Treatment of Uncertainty in Knowledge-Based Systems Using Fuzzy Sets and Possibility Theory", Int. Journal of Intelligent Systems, Vol. 3, Summer, 141-165

Edgar D. (1988), "How to get Information out of Experts", Asian Computer Monthly, May, 34-41

Eliot, L.B. (1987), "Investigating the Nature of Expertise : Analogical Thinking, Expert System, and ThinkBack", Expert Systems, Vol. 4, No. 3, 190-195

Esogbue A.O. and Elder, R.C. (1983), "Measurement and Valuation of a Fuzzy Mathematical Model for Medical Diagnosis", Fuzzy Sets and Systems, 10, 223-242

Fu, K.S., Ishizuka M. and Yao, J.T.P. (1986), "Application of Fuzzy sets in Earthquake Engineering", Fuzzy Set and Possibility Theory - Recent Development, 504-517

Gaines, B.R. (1987), "An Overview of Knowledge-acquisition and Transfer", Int. J. Man-Machine Studies, 26, 453-472.

Hayes-Roth, F., (1985), "Rule-Based Systems", Communications of ACM, 28(9), 921-932.

Johnson, P.E., Nachtsheim, C.J. and Zualkernan I.A. (1987), "Consultant Expertise", Expert Systems, Vol. 4, No. 3, 180-188

Lam, W. (1988), "An Expert System Building Tools Incorporated with Fuzzy Concepts", MPhil. Thesis, The Chinese University of Hong Kong

Leung, K.S. and Lam, W. (1988), "Fuzzy Concepts in Expert Systems", IEEE Computer, Sept., 43-56

Leung, K.S. and Lam, W. (1989), "A Fuzzy Expert System Shell Using Both Exact and Inexact Reasoning", Journal of Automated Reasoning, 5, 207-233

Leung, K.S., Wong, F.W.S. and Lam, W. (1988), "The Development of An Expert Computer System on Medical Consultation", International Journal of Bio-Medical Computing, 23, 265-278

Leung, K.S., Wong, F.W.S., and Lam, W. (1989), "Application of a novel fuzzy expert system shell", Expert Systems, Vol. 6, No. 1, 2-10

Karwowski, W., Mulholland, Nancy O. and Ward, Thomas L. (1987), "A Fuzzy Knowledge Base of an Expert System for Analysis of Manual Lifting Tasks", Fuzzy Sets and Systems, 21, 363-374

KES General Description Manual (1983), Software Architecture and Engineering, Inc., Arlington, Va., p.33

Klir, G.J. and Folger, T.A. (1988), Fuzzy Sets, Uncertainty, and Information, Prentice Hall Englewood Cliffs, N.J. 07632

Koyama, T. (1987), "Artificial Intelligence in Clinical Support Systems", Automedia, Vol. 8, 27-33

Kulikowski, C.A. (1983), "Expert Medical Consultation Systems", Journal of Medical System, Vol. 7, 229-234

Mizumoto, M., Fukami, S., and Tanaka, K., (1979), "Some Methods of Fuzzy Reasoning", Advances in Fuzzy Set Theory and Applications, North-Holland, Amsterdam, 117-136

Nguyen, T.A., Walton, A.P., Laffey, T.J. and Pecora, D., (1987), "Knowledge Base Vertification", AI Magazine, Summer, 69-75

Olson, J.R. and Rueter, H.H. (1987), "Extracting Expertise from Experts : Methods for Knowledge Acquisition", Expert Systems, Vol. 4, No. 3, 152-168

Regoczei, S. and Plantinger, E.P.O. (1987), "Creating the domain of discourse : ontology and inventory", Int J. Man-Machine Studies, 27, 235-250

Schwartz, W.B., Patil, R.S. and Szolovits, P., (1987), "Artifical Intelligence in Medicine : Where do We Stand", New England Journal of Medicine, 316, 685-688

Shooman, M.L., Software Engineering, Design, Reliability, and Management, McGraw-Hill, 238-248

Shortliffe, E.H. (1976), Computer-Based Medical Consultation : MYCIN, New York : Elsevier

Tsang, W.W., Wan, Y.S., Lim, E.L., and Hioe, W., (1988), "A Space Searching Method for Checking the Consistency and Completeness of a Rulebase", Internation Computer Science Conference, 575-579

Wenstop, F. (1980), "Quantitative Analysis with Linguistic Values", Fuzzy Sets and Systems, 4, 99-115

Weisman, R. (1987), "Six Steps to AI-Based Functional Prototyping", Datamation, August 1, 71-72

Wong, F.W.S, Leung, K.S. and So, Y.T., "The Recent Development and Evaluation of A Medical Expert System (ABVAB)", to be published in International Journal of Bio-Medical Computing

Zadeh, L.A. (1965), Fuzzy Sets, Information and Control, 8, 338-353

Zadeh, L.A. (1972), "A Fuzzy-Set-Theoretic Interpretation of Linguistic Hedges", Journal of Cybernetics, 2,3, 4-34.

APPENDIX I

S-Form fuzzy membership function

S(x;a,b,c)

$= 0$ for $x \le a$.

$= 2\left(\dfrac{x-a}{c-a}\right)^2$ for $a \le x \le b$.

$= 1 - 2\left(\dfrac{x-c}{c-a}\right)^2$ for $b \le x \le c$.

$= 1$ for $x \ge c$.

K-form fuzzy membership function

K(x; l) = min[1,l.x].

APPENDIX II

Consider the form of inference shown below in which a fuzzy conditional proposition is contained :

Fact : x is A'
Rule : IF x is A then y is B

Conclusion : y is B'

According to human intuitions it seems that relation between A' and B' ought to be satisfied as shown below :

	Fact	Conclusion
Relation I	x is A	y is B
Relation II-1	x is <u>very</u> A	y is <u>very</u> B
Relation II-2	x is <u>very</u> A	y is B
Relation III	x is <u>more</u> <u>or</u> <u>less</u> A	y is <u>more</u> <u>or</u> <u>less</u> B
Relation IV-1	x is <u>not</u> A	y is <u>unknown</u>
Relation IV-2	x is <u>not</u> A	y is <u>not</u> B

<u>Table for the relations between fact and conclusion</u>

Fuzzy relation Rs, Rg and Rsg can satisfy different relations stated above. The following table will show the satisfaction of each relation.

	Rs	Rg	Rsg
Relation I	O	O	O
Relation II-1	O	X	O
Relation II-2	X	O	X
Relation III	O	O	O
Relation IV-1	O	O	X
Relation IV-2	X	X	O

O - satisfied

X - not satisfied

APPENDIX III : Rule base of INDUCE36

```
( rule a1
If ((( age  > = 35.000000 )
 and ( parity  = 0.000000 ))
 or (( age  >= 40.000000 )
 and ( parity  = 1.000000 )))
then  elderly is yes
) Certainty is  1

( rule a10
If (( gestation  < 35.600002 )
 or (( gestation  > = 35.600002 , 38.000000  {CF < = 0.700000}  )
 and ( mingest  < 35.000000 )))
then  gest3638 is no
) Certainty is  1

( rule a11
If (( gestation  > = 40.000000  {CF > = 0.700000}  )
 or (( gestation  > = 40.000000  {CF < = 0.700000}  )
 and (( mingest  > = 38.000000 )
 or (( mingest  > = 37.000000 )
 and ( cervix  is favourable )))))
then  gest40 is yes
) Certainty is  1

( rule a12
If (( gestation  < 40.000000 )
 or (( gestation  > = 40.000000  {CF < = 0.700000}  )
 and (( mingest  < 37.000000 )
 or ((( mingest  > = 37.000000 )
 and ( mingest  < 38.000000 ))
 and ( cervix  is unfavourable )))))
then  gest40 is no
) Certainty is  1

( rule a13
If (( gestation  > = 39.599998 , 42.000000  {CF > = 0.700000}  )
 or ((( gestation  > = 39.599998 , 42.000000  {CF < = 0.700000}  )
 and ( mingest  > 38.599998 ))
 and ( cervix  is favourable )))
then  gest4042 is yes
) Certainty is  1

( rule a14
If (( gestation  < 39.600002 )
 or (( gestation  > = 39.600002 , 42.000000  {CF < = 0.700000}  )
 and (( mingest  < 38.599998 )
 or ((( mingest  > 38.599998 )
 and ( mingest  < 39.599998 ))
 and ( cervix  is unfavourable )))))
then  gest4042 is no
) Certainty is  1
```

```
( rule a15
 If (( gestation  > = 40.599998 , 42.000000  {CF > = 0.700000}  )
  or ((( gestation  > = 40.599998 , 42.000000  {CF < = 0.700000}  )
  and ( mingest  > 39.599998 ))
  and ( cervix  is favourable )))
 then  gest4142 is yes
 ) Certainty is  1

( rule a16
 If (( gestation  < 40.600002 )
  or (( gestation  > = 40.600002 , 42.000000  {CF < = 0.700000}  )
  and (( mingest  < 39.599998 )
  or ((( mingest  > 39.599998 )
  and ( mingest  < 40.599998 ))
  and ( cervix  is unfavourable )))))
 then  gest4142 is no
 ) Certainty is  1

( rule a17
 If (( gestation  > = 41.099998 , 42.500000  {CF > = 0.700000}  )
  or ((( gestation  > = 41.099998 , 42.500000  {CF < = 0.700000}  )
  and ( mingest  > 40.099998 ))
  and ( cervix  is favourable )))
 then  gest415425 is yes
 ) Certainty is  1

( rule a18
 If (( gestation  < 41.500000 )
  or (( gestation  > = 41.500000 , 42.500000  {CF < = 0.700000}  )
  and (( mingest  < 40.099998 )
  or ((( mingest  > 40.099998 )
  and ( mingest  < 41.099998 ))
  and ( cervix  is unfavourable )))))
 then  gest415425 is no
 ) Certainty is  1

( rule a19
 If (( gestation  > = 42.000000  {CF > = 0.700000}  )
  or (( gestation  > = 42.000000  {CF < = 0.700000}  )
  and (( mingest  > = 41.000000 )
  or (( mingest  > = 39.000000 )
  and ( cervix  is favourable )))))
 then  gest42 is yes
 ) Certainty is  1

( rule a2
 If (((( age  < 35.000000 )
  and ( parity  = 0.000000 ))
  or (( age  < 40.000000 )
  and ( parity  = 1.000000 )))
  or ( parity  > 1.000000 ))
 then  elderly is no
 ) Certainty is  1
```

```
( rule a20
 If (( gestation  < 42.000000 )
  or (( gestation  > = 42.000000  {CF < = 0.700000}  )
  and (( mingest  < 39.000000 )
  or ((( mingest  > = 39.000000 )
  and ( mingest  < = 41.000000 ))
  and ( cervix  is unfavourable )))))
 then  gest42 is no
 ) Certainty is  1

( rule a21
 If (( gestation  > = x  {CF > = 0.700000}  )
  or (( gestation  > = x  {CF < = 0.700000}  )
  and (( mingest  > = 42.000000 )
  and ( cervix  is favourable ))))
 then  gestx is yes
 ) Certainty is  1

( rule a22
 If (( gestation  < x )
  or (( gestation  > = x  {CF < = 0.700000}  )
  and (( mingest  < 42.000000 )
  or ( cervix  is unfavourable ))))
 then  gestx is no
 ) Certainty is  1

( rule a3
 If ((( bs  > 3.000000 )
  and ( parity  > 0.000000 ))
  or (( bs  > 6.000000 )
  and ( parity  = 0.000000 )))
 then  cervix is favourable
 ) Certainty is  1

( rule a4
 If ((( bs  < 4.000000 )
  and ( parity  > 0.000000 ))
  or (( bs  < 6.000000 )
  and ( parity  = 0.000000 )))
 then  cervix is unfavourable
 ) Certainty is  1

( rule a5
 If (( gestation  > = 38.000000  {CF > = 0.700000}  )
  or (( gestation  > = 38.000000  {CF < = 0.700000}  )
  and ( mingest  > = 36.000000 )))
 then  gest38 is yes
 ) Certainty is  1

( rule a6
 If (( gestation  < 38.000000 )
  or (( gestation  > = 38.000000  {CF < = 0.700000}  )
  and ( mingest  < 36.000000 )))
 then  gest38 is no
 ) Certainty is  1
```

```
( rule a7
 If (( gestation  > = 37.599998 , 40.000000  {CF > = 0.700000}  )
  or ((( gestation  > = 37.599998 , 40.000000  {CF < = 0.700000}  )
  and ( mingest  > 36.599998 ))
  and ( cervix  is favourable )))
 then  gest3840 is yes
 ) Certainty is  1

( rule a8
 If (( gestation  < 37.600002 )
  or (( gestation  > = 37.600002 , 40.000000  {CF < = 0.700000}  )
  and (( mingest  < 36.599998 )
  or ((( mingest  > 36.599998 )
  and ( mingest  < 37.599998 ))
  and ( cervix  is unfavourable )))))
 then  gest3840 is no
 ) Certainty is  1

( rule a9
 If (( gestation  > = 35.900002 , 38.000000  {CF > = 0.700000}  )
  or (( gestation  > = 35.900002 , 38.000000  {CF < = 0.700000}  )
  and ( mingest  > = 35.000000 )))
 then  gest3638 is yes
 ) Certainty is  1

( rule aa
 If (( gestation  > = 36.000000  {CF > = 0.700000}  )
  or (( gestation  > = 36.000000  {CF < = 0.700000}  )
  and (( mingest  > = 34.000000 )
  or ( maturity  is yes))))
 then  gest36 is yes
 ) Certainty is  1

( rule ab
 If (( gestation  < 36.000000 )
  or (( gestation  > = 36.000000  {CF < = 0.700000}  )
  and (( mingest  < 34.000000 )
  or ( maturity  is no))))
 then  gest36 is no
 ) Certainty is  1

( rule b01
 If (( elderly  is yes)
  and ( gest42  is yes))
 then  plan is delivery
 ) Certainty is  0.75

( rule b02
 If (( elderly  is yes)
  and ( gest42  is no))
 then  plan is monitor
 ) Certainty is  0.85

( rule b03
 If (( badhx  is not nil )
  and ( gest40  is yes))
 then  plan is delivery
 ) Certainty is  0.75
```

```
( rule b04
 If (( badhx  is not nil )
  and ( gest40  is no))
 then  plan is monitor
 ) Certainty is  0.85

( rule b05
 If (( socioeconomic  is poor  )
  and ( gest42  is yes))
 then  plan is delivery
 ) Certainty is  0.75

( rule b06
 If (( socioeconomic  is poor  )
  and ( gest42  is no))
 then  plan is monitor
 ) Certainty is  0.85

( rule b07
 If (( nofetus  = 2.000000 )
  and ( gest3840  is yes))
 then  plan is delivery
 ) Certainty is  0.95

( rule b08
 If (( nofetus  = 2.000000 )
  and ( gest3840  is no))
 then  plan is monitor
 ) Certainty is  0.9

( rule b09
 If (( liquor  is excess  )
  and ((( gest415425  is yes)
  or ((( mdcs  is not nil )
  or ( ancx  is not nil ))
  and ( gest40  is yes)))
  or ( symptomatic  is yes)))
 then  plan is delivery
 ) Certainty is  0.9

( rule b10
 If ((( liquor  is excess  )
  and ( symptomatic  is no))
  and (((( gest415425  is no)
  and ( mdcs  is nil ))
  and ( ancx  is nil ))
  or ((( ancx  is not nil )
  and ( mdcs  is not nil ))
  and ( gest40  is no))))
 then  plan is monitor
 ) Certainty is  0.85

( rule b11
 If (( lessfetomove  is yes)
  and ( gest4142  is yes))
 then  plan is delivery
 ) Certainty is  0.9
```

```
( rule b12
 If (( lessfetomove  is yes)
  and ( gest4142  is no))
 then  plan is monitor
 ) Certainty is  0.8

( rule b13
 If (( mdcs  is rft )
  and (( gest3638  is yes)
  or ( derft  is yes)))
 then  plan is delivery
 ) Certainty is  0.9

( rule b14
 If ((( mdcs  is rft )
  and ( gest3638  is no))
  and ( derft  is no))
 then  plan is monitor
 ) Certainty is  0.8

( rule b15
 If ((( mdcs  is gdm )
  and ( control  is good  ))
  and ( gest3840  is yes))
 then  plan is delivery
 ) Certainty is  0.9

( rule b16
 If ((( mdcs  is gdm )
  and ( control  is good  ))
  and ( gest3840  is no))
 then  plan is monitor
 ) Certainty is  0.85

( rule b17
 If ((( mdcs  is gdm )
  and ( control  is bad  ))
  and ( gest3638  is yes))
 then  plan is delivery
 ) Certainty is  0.9

( rule b18
 If ((( mdcs  is gdm )
  and ( control  is bad  ))
  and ( gest3638  is no))
 then  plan is monitor
 ) Certainty is  0.85

( rule b19
 If ((( mdcs  is thyrotoxic )
  and ( tcontrol  is yes))
  and ( gest42  is yes))
 then  plan is delivery
 ) Certainty is  0.9
```

```
( rule b20
 If ((( mdcs  is thyrotoxic )
  and ( tcontrol  is yes))
  and ( gest42  is no))
 then  plan is monitor
 ) Certainty is  0.85

( rule b21
 If ((( mdcs  is thyrotoxic )
  and ( tcontrol  is no))
  and ( gest40  is yes))
 then  plan is delivery
 ) Certainty is  0.9

( rule b22
 If ((( mdcs  is thyrotoxic )
  and ( tcontrol  is no))
  and ( gest40  is no))
 then  plan is monitor
 ) Certainty is  0.85

( rule b23
 If ( nofetus  = 3.000000 )
 then  plan is delivery
 ) Certainty is  1

( rule b24
 If ( ctg  is not reactive )
 then  plan is delivery
 ) Certainty is  1

( rule b25
 If (( liquor  is excess  )
  and (((( symptomatic  is yes)
  or ( mdcs  is gdm ))
  or ( ancx  is lethal ))
  or ( nofetus  > 1.000000 )))
 then  plan is delivery
 ) Certainty is  0.8

( rule b26
 If ((((( liquor  is decreas  )
  or ( membrupt  is yes))
  or ( ancx  is hydrops ))
  or ( mdcs  is cancer ))
  or ( ancx  is iud ))
 then  plan is delivery
 ) Certainty is  0.95

( rule b27
 If (((( derft  is yes)
  or ( ancx  is pih ))
  or ( ancx  is pet ))
  and ( mdcs  is rft ))
 then  plan is delivery
 ) Certainty is  1
```

```
( rule b28
 If (( mdcs  is ctd )
  and ((((( ancx  is iugr )
  or ( ancx  is pih ))
  or ( excerbation  is yes))
  or ( derft  is yes))
  or ( gest42  is yes)))
 then  plan is delivery
 ) Certainty is  0.95

( rule b29
 If (( mdcs  is ctd )
  and ((( gest42  is no)
  and ( derft  is no))
  or ( ancx  is nil )))
 then  plan is monitor
 ) Certainty is  0.8

( rule b31
 If (( mdcs  is severeliver )
  and (( gest36  is yes)
  or ( derft  is yes)))
 then  plan is delivery
 ) Certainty is  0.9

( rule b31a
 If ((( mdcs  is severeliver )
  and ( gest36  is no))
  and ( derft  is no))
 then  plan is monitor
 ) Certainty is  0.8

( rule b32
 If ((( mdcs  is gdm )
  and ( control  is good  ))
  and ( gest3840  is yes))
 then  plan is delivery
 ) Certainty is  1

( rule b33
 If ((( mdcs  is gdm )
  and ( control  is good  ))
  and ( gest3840  is no))
 then  plan is monitor
 ) Certainty is  0.85

( rule b34
 If ((( mdcs  is gdm )
  and ( control  is bad  ))
  and ( gest3638  is yes))
 then  plan is delivery
 ) Certainty is  1

( rule b35
 If ((( mdcs  is gdm )
  and ( control  is bad  ))
  and ( gest3638  is no))
 then  plan is monitor
 ) Certainty is  0.85
```

```
( rule b36
If ( fhr  > = 160.000000 , 180.000000 )
then  plan is delivery
) Certainty is  1

( rule b37
If (( ancx  is aph )
 and ( gest40  is yes))
then  plan is delivery
) Certainty is  0.95

( rule b38
If (( ancx  is aph )
 and ( gest40  is no))
then  plan is monitor
) Certainty is  0.9

( rule b39
If ( ancx  is abruption )
then  plan is delivery
) Certainty is  1

( rule b40
If (( ancx  is pp )
 and (( gest3638  is yes)
 or ( activebleed  is yes)))
then  plan is delivery
) Certainty is  1

( rule b41
If ((( ancx  is pp )
 and ( gest3638n  is no))
 and ( activebleed  is no))
then  plan is monitor
) Certainty is  0.9

( rule b42
If ((( ancx  is pih )
 or ( ancx  is essentialht ))
 and ((( mdcs  is cardiac )
 and ( gest38  is yes))
 or ( gest42  is yes)))
then  plan is delivery
) Certainty is  0.9

( rule b43
If ((( ancx  is pih )
 or ( ancx  is essentialht ))
 and (( gest42  is no)
 or (( mdcs  is cardiac )
 and ( gest38  is no))))
then  plan is monitor
) Certainty is  0.9

( rule b44
If ( ancx  is pet )
then  plan is delivery
) Certainty is  0.95
```

```
( rule b45
 If (((( ancx  is iugr )
  and ( growth  is yes))
  and ( retardation  is mild  ))
  and ( gest40  is yes))
 then  plan is delivery
 ) Certainty is  0.95

( rule b46
 If (((( ancx  is iugr )
  and ( growth  is yes))
  and ( retardation  is mild  ))
  and ( gest40  is no))
 then  plan is monitor
 ) Certainty is  0.8

( rule b47
 If ((( ancx  is iugr )
  and ( retardation  is severe  ))
  and ( gest36  is yes))
 then  plan is delivery
 ) Certainty is  0.95

( rule b47a
 If ((( ancx  is iugr )
  and ( retardation  is severe  ))
  and ( gest36  is no))
 then  plan is monitor
 ) Certainty is  0.8

( rule b48
 If (( ancx  is iugr )
  and ( growth  is no))
 then  plan is delivery
 ) Certainty is  1

( rule b49
 If ( gestx  is yes)
 then  plan is delivery
 ) Certainty is  1

( rule b50
 If ((((((((((( ancx  is nil )
  and ( liquor  is normal  ))
  and ( mdcs  is nil ))
  and ( elderly  is no))
  and ( ctg  is reactive ))
  and ( fhr  > 110.000000 , 120.000000 ))
  and ( fhr  < 160.000000 , 170.000000 ))
  and ( nofetus  = 1.000000 ))
  and ( gestx  is no))
  and ( badhx  is nil ))
  and ( lessfetomove  is no))
 then  plan is monitor
 ) Certainty is  0.75
```

```
( rule b51
 If ( fhr  < 108.000000 , 120.000000 )
 then  plan is delivery
 ) Certainty is  0.4

( rule b52
 If ((( mdcs  is cardiac )
  and ( gest40  is yes))
  and ( cervix  is favourable ))
 then  plan is delivery
 ) Certainty is  0.8

( rule b53
 If (( mdcs  is cardiac )
  and (( gest40  is no)
  or ( cervix  is unfavourable )))
 then  plan is monitor
 ) Certainty is  0.7

( rule b54
 If ( mdcs  is lifethreat )
 then  plan is delivery
 ) Certainty is  0.9

( rule b55
 If ( mdcs  is lifethreat )
 then  plan is monitor
 ) Certainty is  0.8

( rule b56
 If (( mdcs  is not nil )
  and ( gest42  is yes))
 then  plan is delivery
 ) Certainty is  0.9
```

APPENDIX IV : Rule Base of ESROM

```
( rule d01
 If ( amount  is much  )
 then  diagnosis is membrupt
 ) Certainty is  0.2

( rule d05
 If ( liquor  is msl )
 then  diagnosis is membrupt
 ) Certainty is  0.95

( rule d06
 If ( cordprolapse  is yes)
 then  diagnosis is membrupt
 ) Certainty is  1.0

( rule d09
 If ( membrane  is no)
 then  diagnosis is membrupt
 ) Certainty is  0.99

( rule d10
 If ( liqorush  is yes)
 then  diagnosis is membrupt
 ) Certainty is  1.0

( rule d16
 If ((( nitrazine  is yes)
  and ( pooliq  is yes [0.7]  ))
  and ( pad  is wet   [0.4]  ))
 then  diagnosis is membrupt
 ) Certainty is  0.99

( rule d17
 If ((( nitrazine  is yes)
  and ( pooliq  is no [0.7]  ))
  and ( pad  is wet   [0.4]  ))
 then  diagnosis is membrupt
 ) Certainty is  0.9

( rule d18
 If ((( nitrazine  is yes)
  and ( pooliq  is no [0.7]  ))
  and ( pad  is dry   [0.4]  ))
 then  diagnosis is membrupt
 ) Certainty is  0.75

( rule d19
 If ((( nitrazine  is no)
  and ( pooliq  is yes [0.7]  ))
  and ( pad  is wet   [0.4]  ))
 then  diagnosis is membrupt
 ) Certainty is  0.8
```

```
( rule d20
 If ((( nitrazine  is no)
  and ( pooliq  is yes [0.7]  ))
  and ( pad  is dry   [0.4]  ))
 then  diagnosis is membrupt
 ) Certainty is  0.1

( rule d21
 If ((( nitrazine  is no)
  and ( pooliq  is no [0.7]  ))
  and ( pad  is wet   [0.4]  ))
 then  diagnosis is membrupt
 ) Certainty is  0.1

( rule d22
 If ((( nitrazine  is yes)
  and ( pooliq  is no [0.7]  ))
  and ( pad  is dry   [0.4]  ))
 then  diagnosis is umembrupt
 ) Certainty is  0.2

( rule d23
 If ((( nitrazine  is no)
  and ( pooliq  is yes [0.7]  ))
  and ( pad  is wet   [0.4]  ))
 then  diagnosis is umembrupt
 ) Certainty is  0.2

( rule d24
 If ((( nitrazine  is no)
  and ( pooliq  is yes [0.7]  ))
  and ( pad  is dry   [0.4]  ))
 then  diagnosis is umembrupt
 ) Certainty is  0.9

( rule d25
 If ((( nitrazine  is no)
  and ( pooliq  is no [0.7]  ))
  and ( pad  is wet   [0.4]  ))
 then  diagnosis is umembrupt
 ) Certainty is  0.9

( rule d26
 If ((( nitrazine  is no)
  and ( pooliq  is no [0.7]  ))
  and ( pad  is dry   [0.4]  ))
 then  diagnosis is umembrupt
 ) Certainty is  0.98

( rule i01
 If ((( temp  > 37.200001, 38.000000 )
  and ( diagnosis  is membrupt ))
  and ( otherinf  is no [0.7]  ))
 then  infection is yes
 ) Certainty is  0.45
```

```
( rule i02
 If (( liquor  is msl )
  and ( gestation  < = 28.000000, 36.000000 ))
 then  infection is yes
 ) Certainty is  0.8

( rule i03
 If ( liquor  is purulent )
 then  infection is yes
 ) Certainty is  0.85

( rule i04
 If (( ctg  is nonreactive )
  or ( ctg  is unsatisfactory ))
 then  infection is yes
 ) Certainty is  0.2

( rule i05
 If ( ctg  is decelerative )
 then  infection is yes
 ) Certainty is  0.5

( rule i06
 If (( wcc  > 10.000000, 15.000000 )
  and ( diagnosis  is membrupt ))
 then  infection is yes
 ) Certainty is  0.45

( rule i07
 If (( crp  > 20.000000, 30.000000 )
  and ( diagnosis  is membrupt ))
 then  infection is yes
 ) Certainty is  0.3

( rule i08
 If (( basalfhr  > 160.000000, 190.000000 )
  and ( diagnosis  is membrupt ))
 then  infection is yes
 ) Certainty is  0.1

( rule i10
 If ( basalfhr  > 160.000000, 190.000000 )
 then  infection is yes
 ) Certainty is  0.3

( rule i13
 If ((( inwcc  is yes)
  and ( wcc  > 10.0000, 15.0000 ))
  and ( diagnosis  is membrupt ))
 then  infection is yes
 ) Certainty is  0.4

( rule i14
 If ((( incrp  is yes)
  and ( crp  > 20.000000, 30.00000 ))
  and ( diagnosis  is membrupt ))
 then  infection is yes
 ) Certainty is  0.4
```

```
( rule i15
 If ((( liquor  is clear )
  and ( ctg  is reactive ))
  and ( basalfhr  < 165.000000 ))
 then  infection is no
 ) Certainty is  0.8

( rule i16
 If ( fbreathing  is yes)
 then  infection is no
 ) Certainty is  0.8

( rule i17
 If (( temp  < 37.200001, 37.500000 )
  and ( wcc  < 10.000000, 13.000000 ))
 then  infection is no
 ) Certainty is  0.6

( rule i18
 If (( temp  < 37.200001, 37.500000 )
  and ( crp  < 20.000000, 25.000000 ))
 then  infection is no
 ) Certainty is  0.6

( rule i19
 If ( hvs  is yes)
 then  infection is yes
 ) Certainty is  0.2

( rule i20
 If ( amniocult  is yes)
 then  infection is yes
 ) Certainty is  1.0

( rule i21
 If ( amniocult  is no)
 then  infection is no
 ) Certainty is  0.8

( rule i22
 If ((( ctg  is reactive )
  and ( temp  < 37.200001, 37.400002 ))
  and ( basalfhr  < 157.000000, 165.000000 ))
 then  infection is no
 ) Certainty is  0.95

( rule m01
 If ( infection  is yes)
 then  management is delivery
 ) Certainty is  1.0

( rule m02
 If (( gestation  > 35.990002 )
  and ( diagnosis  is membrupt ))
 then  management is delivery
 ) Certainty is  0.9
```

```
( rule m03
 If (((( gestation  > 34.000000 )
  and ( gestation  < 36.000000 ))
  and ( duration  > 48.000000 ))
  and ( diagnosis  is membrupt ))
 then  management is delivery
 ) Certainty is  0.8

( rule m03a
 If (((( gestation  > 34.000000 )
  and ( gestation  < 36.000000 ))
  and ( duration  < 48.000000 ))
  and ( diagnosis  is membrupt ))
 then  management is observation
 ) Certainty is  0.8

( rule m04
 If ( liquor  is msl )
 then  management is delivery
 ) Certainty is  0.97

( rule m05
 If ( cordprolapse  is yes)
 then  management is delivery
 ) Certainty is  1.0

( rule m06
 If (( gencon  is poor  )
  and ( diagnosis  is membrupt ))
 then  management is delivery
 ) Certainty is  0.7

( rule m07
 If ((( ctg  is nonreactive )
  or ( ctg  is unsatisfactory ))
  and ( diagnosis  is membrupt ))
 then  management is delivery
 ) Certainty is  0.7

( rule m08
 If ( ctg  is decelerative )
 then  management is delivery
 ) Certainty is  0.9

( rule m09
 If ((( ls  > 2.000000 )
  or ( pg  is yes))
  and ( diagnosis  is membrupt ))
 then  management is delivery
 ) Certainty is  0.8

( rule m10
 If ((( gestation  > 30.000000 )
  and ( noliqusg  is yes))
  and ( diagnosis  is membrupt ))
 then  management is delivery
 ) Certainty is  0.65
```

```
( rule m11
If (( mdcs  is valvular )
 and ( diagnosis  is membrupt ))
then  management is delivery
) Certainty is  0.5

( rule m11a
If (( mdcs  is gdm )
 and ( diagnosis  is membrupt ))
then  management is delivery
) Certainty is  0.5

( rule m12
If (( mdcs  is rft )
 and ( diagnosis  is membrupt ))
then  management is delivery
) Certainty is  0.5

( rule m14
If (( ancx  is pih )
 and ( diagnosis  is membrupt ))
then  management is delivery
) Certainty is  0.2

( rule m15
If ((( ancx  is pet )
 or ( ancx  is iugr ))
 and ( diagnosis  is membrupt ))
then  management is delivery
) Certainty is  0.5

( rule m16
If (( ancx  is aph )
 and ( diagnosis  is membrupt ))
then  management is delivery
) Certainty is  0.3

( rule m18
If (( mdcs  is symptomatic )
 and ( diagnosis  is membrupt ))
then  management is delivery
) Certainty is  0.3

( rule m19
If ( prevpnd  is yes)
then  management is delivery
) Certainty is  0.1

( rule m20
If ( amniocult  is yes)
then  management is delivery
) Certainty is  1.0

( rule m21
If (( diagnosis  is umembrupt )
 and ( infection  is no))
then  management is observation
) Certainty is  0.95
```

```
( rule m22
If ( gestation  < 34.000000, 35.000000 )
then  management is observation
) Certainty is  0.6

( rule m23
If ( gestation  < 32.000000, 33.990002 )
then  management is observation
) Certainty is  0.5

( rule m24
If ( gestation  < 30.000000, 31.990000 )
then  management is observation
) Certainty is  0.5

( rule m25
If ( gestation  < 26.000000, 29.990000 )
then  management is observation
) Certainty is  0.5

( rule m26
If ( gestation  < 26.000000 )
then  management is observation
) Certainty is  0.5
```