1

Micro-D

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, tests publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30.

**The Design
of a
Medical Knowledge Base System
for
Ophthalmologists**

by

Geetha Raghu Ram, B.Eng.

A thesis submitted to the
Faculty of Graduate Studies and Research
in partial fulfillment of the requirements
for the degree of
Master of Engineering.

Ottawa-Carleton Institute for Electrical Engineering
Faculty of Engineering
Department of Systems and Computer Engineering
Carleton University
Ottawa, Ontario, Canada

June, 1988

© Geetha Raghu Ram 1988

Permission has been granted to the National Library of Canada to microfilm this thesis and to lend or sell copies of the film.

The author (copyright owner) has reserved other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without his/her written permission.

L'autorisation a été accordée à la Bibliothèque nationale du Canada de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur (titulaire du droit d'auteur) se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation écrite.

ISBN 0-315-46267-1

The undersigned hereby recommend to the Faculty of Graduate Studies and Research acceptance of the thesis

*The Design of a Medical Knowledge Base System
for Ophthalmologists.*

submitted by Geetha Raghu Ram, B.Eng., in partial fulfilment of the requirements for the degree of Master of Engineering.

Thesis Supervisor

External Examiner

Chairman,

Department of Systems and
Computer Engineering

Department of Systems and Computer Engineering

Faculty of Engineering

Carleton University

August, 1988

Abstract

Knowledge Base (KB) systems are emerging as a technology that can provide significant help in solving complex problems. In this thesis, a KB system in the Familial Retinal Degeneration (FRD) domain has been developed for the diagnosis and the research purposes in the FRD domain. Furthermore, the feasibility of application of a layered model of expertise to the design of the KB system is examined.

This research presents a review of a number of existing medical KB systems. The FRD KB system has been designed based on a layered model of expertise. A layered model for the expertise in the KB system is given. Next, an initial prototype for the FRD KB system was implemented using the expert system building tool, KEE. Future research directions are also discussed.

Acknowledgements

I would like to express my sincere gratitude to my thesis supervisor, Dr. M. Niktash for his constant help and able guidance throughout the process of this research work.

I would also like to express my special thanks to Dr. H.T.J. Mount, Department of Ophthalmology, Ottawa Civic Hospital, for having spent his valuable time in imparting his expertise for this research.

Special appreciations go to my friends Debbie Alan and Vivek Kapil for their assistance in preparing this report.

I would also like to thank my mother for her constant everlasting support.

Contents

1	Introduction	1
2	FRD Knowledge Base System Overview	3
2.1	Introduction	3
2.2	Problem Area Selection Criteria	4
2.3	FRD Disease Classification	7
2.4	Tests Employed in the FRD Diagnosis	13
2.5	Objective of the FRD KB System	24
3	Design and Development of KB Systems	26
3.1	Introduction	26
3.2	Problem Solving Expertise	27
3.3	A Model for Problem Solving Expertise	28
3.4	Knowledge Representation	36
3.5	KB User Interface	44
3.6	KB System Development Phases	47
4	Some of the Existing Medical Knowledge Base Systems	52

4.1	Introduction	52
4.2	A Framework to Identify KBS Features	53
4.3	MYCIN	54
4.4	NEOMYCIN	59
4.5	KB System in Dermatology	65
4.6	PIP	69
4.7	BREASTCAN	75
4.8	CASNET	81
4.9	CADIAG-2	85
4.10	INTERNIST	90
4.11	MDX	94
5	Overview of the KEE	99
5.1	Introduction	99
5.2	Knowledge Representation in KEE	100
5.3	Reasoning in KEE	103
5.4	KEE Methods	107
5.5	Active Values and Active Images	109
5.6	Other Extended Features of KEE	113
6	Design and Implementation of the FRD KB System	116
6.1	Introduction	116
6.2	Knowledge Acquisition	117
6.3	Architecture of the FRD KB System	119
6.3.1	Domain Level	120

6.3.2	Inference Level	133
6.3.3	Task Level	141
6.4	User Interface	145
6.5	KB Implementation Environment	149
6.6	Testing	149
7	Conclusions	150
7.1	Contributions of this Research	150
7.2	Future Developments	152
8	Bibliography	156
9	Test Disease Relationships	161
10	LISP Methods	174

List of Figures

2.1	<i>Internal structures of the human eye.</i>	9
2.2	<i>The Family Tree of a patient with CSNB.</i>	15
2.3	<i>The Visual Fields chart.</i>	19
2.4	<i>The color defective patterns of patients.</i>	21
2.5	<i>The Electroretinogram responses.</i>	23
3.1	<i>Metaclasses in a medical domain.</i>	31
3.2	<i>Examples of a few Knowledge Sources.</i>	32
3.3	<i>The Inference structure in NEOMYCIN.</i>	33
3.4	<i>A part of the task hierarchy in PDP KB system.</i>	34
3.5	<i>The structure of a Strategic Reasoning process.</i>	35
3.6	<i>Knowledge represented in predicate calculus.</i>	37
3.7	<i>A hypothetical causal model of the coronary heart disease.</i>	39
3.8	<i>A network of frames.</i>	41
3.9	<i>A few general production rules.</i>	43
4.1	<i>A few rules from the MYCIN knowledge base system.</i>	55
4.2	<i>The structure of NEOMYCIN's domain knowledge.</i>	61

4.3	<i>A portion of the etiological taxonomy in NEOMYCIN.</i>	62
4.4	<i>The static representation of NEOMYCIN's diagnostic task.</i>	64
4.5	<i>A rule and a fact from the KB system in Dermatology.</i>	66
4.6	<i>The architecture of the PIP knowledge base system.</i>	70
4.7	<i>The "nephrotic syndrome" frame in PIP.</i>	72
4.8	<i>A complex network of frames in PIP.</i>	73
4.9	<i>The domain knowledge in BREASTCAN.</i>	76
4.10	<i>The primary treatment frame in BREASTCAN.</i>	78
4.11	<i>The flowcharts for the task structure in BREASTCAN.</i>	79
4.12	<i>The three layer domain model in CASNET.</i>	82
4.13	<i>An example of a rule from CADIAG-2.</i>	86
4.14	<i>The task structure in CADIAG-2.</i>	88
4.15	<i>A sample manifestations' list from INTERNIST.</i>	92
4.16	<i>A conceptual structure of "cholestasis" in MDX.</i>	96
5.1	<i>A hierarchy of units in KEE.</i>	101
5.2	<i>An example of a unit in KEE.</i>	102
5.3	<i>A few rules in KEE.</i>	104
5.4	<i>An intermediate result of rule tracing during the process of backward chaining.</i>	106
5.5	<i>The display of a method in KEE.</i>	108
5.6	<i>An active value unit in KEE.</i>	110
5.7	<i>A few active images in KEE.</i>	112
5.8	<i>Large Scale Expert System Tool Evaluation.</i>	115

6.1	<i>The domain level in the FRD KB system.</i>	121
6.2	<i>The KEE output graph of the FRD knowledge base.</i>	123
6.3	<i>The general slots of a disease unit.</i>	124
6.4	<i>The "CSNB" unit.</i>	125
6.5	<i>The general slots of a test unit.</i>	127
6.6	<i>The "Central-progression" unit</i>	130
6.7	<i>The general slots of a patient unit.</i>	131
6.8	<i>Metaclasses in the FRD KB system.</i>	134
6.9	<i>The knowledge sources in the inference level.</i>	135
6.10	<i>The inference structure in the FRD KB system.</i>	137
6.11	<i>A task structure for the FRD KB system.</i>	142
6.12	<i>The active images for user interface in FRD KB system.</i>	146
6.13	<i>The active images which display the current status of a patient.</i>	147

Chapter 1

Introduction

Artificial Intelligence is the part of computer science concerned with designing intelligent computer systems; that is systems that exhibit characteristics associated with intelligence in human behavior[37]. An example of such a system is a knowledge base system.

The knowledge base system stores data, facts, knowledge found in literature, knowhow elicited from experts etc. It is often employed to solve problems about the world in general, such as in diagnosis, fault finding etc. Various groups involved in an organisation may benefit from the KB systems in several ways. In order to build a KB system, it is important to define the problem being solved by the system. Next, a model for the expertise is designed and finally, this model is implemented.

Chapter 2 describes the problem selection criteria and gives the problem def-

inition and the objective of the FRD (Familial Retinal Degeneration) knowledge base system built in this thesis. In Chapter 3 the modelling, representation and development phases involved in a KB system are described. A number of existing medical KB systems are reviewed in chapter 4. The expert system tool KEE, which was used to build the FRD KB system prototype is described in chapter 5. Chapter 6 presents the design and the implementation of the FRD KB system. And finally, in chapter 7 the contributions of this research and the future developments for the system are given.

Chapter 2

FRD Knowledge Base System Overview

2.1 Introduction

There are a number of areas with large collections of knowledge (both theoretical knowledge and the experience of the experts). The knowledge can be stored in computers. Systems can be built which utilize this knowledge and solve various problems, not necessarily similar to the actual method of problem solving employed by the experts. Such systems are called knowledge base systems. Clearly, a knowledge base system can solve problems with various degrees of decision making. In order to build such a system, the very first step would be to understand and define the problem to be solved by the system.

In the next section, several issues regarding the choice of a problem area are

given. We have chosen the problem of the diagnosis of the Familial Retinal Degeneration (FRD) diseases. Section 2.3 describes the Familial Retinal Degeneration diseases and their classification. The tests employed in the diagnosis of those diseases are briefly described in section 2.4. And finally, after having described the problem, the objective of the FRD KB system is described in section 2.5.

2.2 Problem Area Selection Criteria

It is very important to select a suitable and not just any problem before trying to build a KB system to solve that problem. There are several points that one must consider in problem area selection[2,6].

- The clarity of the problem.

Some problems are well defined and can be represented by mathematical equations (for example, numerical integration). Mathematical models which provide numerical solutions exist. Therefore, numerical techniques can be used to solve such problems.

There are other problems which cannot be represented by mathematical equations, but all or parts of it are well defined. Some or all the underlying rules are well understood. A knowledge base system can be developed to solve such problems. Some problems in medicine or law have such nature

(e.g. FRD domain in ophthalmology, some aspects of law etc.).

Lastly there exist problems, where the underlying rules are not well defined. For example, the problem of handwritten character recognition. Machine learning techniques (e.g. neural net) can be employed to solve such problems.

- The degree of decision making.

There are various degrees of decision making that a KB system can enforce. Before a problem is represented one must determine the type of decision making that is required by the system. In the following, three degrees of decision making namely advisor, decision maker and tutor are explained.

A knowledge base system can be an advisory system and can advise the user regarding a particular problem (for example: MYCIN [10]). A knowledge base system can be designed as a representative system, to represent a problem and make decisions about the problem[18]. Finally the system can be designed as a tutor to teach and guide students (for example: NEOMYCIN[8]).

- Availability of source knowledge.

The availability of knowledge to develop a knowledge base system, to solve a given problem, is an important selection criterion. The knowledge is gener-

ally elicited from the experts in the area and from the published materials.

- Cost benefit.

The cost of the development of a KB must be compensated in some shape or form. The knowledge base system must provide benefit to various groups involved in an organisation:

- User of the KB system: The users benefit due to the increased availability of the system. Furthermore they enjoy gaining more knowhow working with the system.
- Experts: Experts often do not evaluate their approach to solving problems due to the fact that they figure they do not need to do so. However, this gives them an opportunity to evaluate and exercise their problem solving process. The experts can also apply data to expertise and hence use the results to discover new classes and subclasses. Another advantage is that, they can be relieved from tedious tasks which can be designated to the system.
- The management: The management will find it easier to retain and use the expertise. It can protect itself from the loss when an expert leaves or retires from the company.

After choosing a suitable problem, it is very important to understand it and state at least parts of it as clearly as possible. Clear definition of a problem, makes it easier to design a model and develop a KB system. This is because, it is easier to decide on 'how' to do things, if we know clearly 'what' it is to be done.

The problem chosen by us was the diagnosis and research pertaining to the FRD diseases. The reasons for having chosen this problem is the following. The FRD is a domain with overlapping of disease class criteria with each other and with normal such that patterns could only be separated using computer and AI techniques.

The next two sections describe the knowledge domain of the FRD KB system.

2.3 FRD Disease Classification

Familial Retinal Degeneration (FRD) diseases are hereditary retinal and choroidal degenerative diseases. It is a large class of disease having many subclasses with various disorders and all subclasses have a large number of characteristics and ranges of characteristics. FRD diseases tend to be genetically determined disorders with a biochemical basis affecting the retina. Figure 2.1 shows the internal structure of the human eye. The figure shows the retina, which is the part of the eye, that converts the images from the eye's optical system into electrical impulses sent along the optic nerve for transmission back to the brain. It consists of many

layers that include the rods and cones, bipolar cells, amacrine cells, ganglion cells, Mueller cells, horizontal cells and all the interconnecting nerve fibres. The FRD diseases each affect a different set of retinal cells. The rod-cone (see section 2.3) diseases predominantly affect the rod more than the cone system and more at the peripheral pigment epithelial and the outer segment of the receptor than the inner retina. Congenital Stationary Night Blindness (see section 2.3) is an example in which the rod system is affected but at the bipolar layer where evidence of the electrophysiologic sort suggests amacrine cell malfunctions as the prime disorder, not the receptor level itself. The family history plays a very important role in these diseases, as the diseases are known to be familial and genetically induced.

There are a number of subclasses of these diseases [24]. Most often, the subclasses have very similar manifestations and are very difficult to distinguish from each other. A number of these subclasses are described below.

1. Congenital Stationary Night Blindness (CSNB)

This group of diseases has been frequently labeled as essential night blindness [5,24]. One form of CSNB can be traced back to Jean Nougaret, who was the first patient to have been identified as having the disease, in France over 300 years ago. Since then, among 2000 members of his family, 135 are said to have had night blindness. The following describe some of the important characteristics of CSNB.

Figure 2.1: *Internal structures of the human eye.*

Patients with CSNB do not show any progression in the disease, as the name itself indicates. Night blindness is present, but in many cases, the patient may have no problem with his day vision. In one form of CSNB, that with myopia (short sightedness) and nystagmus, however the electroocular level of the disorder, the full photopic field and normal ocular fundus are as in the Nougaret case. But there is nystagmus or involuntary rhythmic movement of the eyes and frequently poor acuity (see section 2.4) in the male showing the full characteristics of sex linked disorder. Some patients may have strabismus or misalignment of the eyes, high refractive error and central blind spot at low illuminance. Some patients suffer from a slight color vision abnormality, particularly the difficulty in distinguishing the color blue. The disease is inherited in an autosomal recessive, autosomal dominant or x-linked recessive manner (see section 2.4). The electroretinographic responses (ERG) are extinguished, with the b-wave affected more than the a-wave (see section 2.4).

2. Rod Cone Dystrophy (RC Dyst)

RC Dyst is a genetically determined group of progressive disorders of vision with characteristic fundus appearance in which the rod system is affected primarily and predominantly, while the cone system is affected in the later stages [5,24]. The classic term "retinitis pigmentosa" is given to this disease subclass. The following are some of its characteristics.

Initially, the patient exhibits progression in the loss of peripheral vision and

later on, exhibits a progression in the loss of central vision. Night vision is poor, but day vision is good. In addition, high myopia (short sightedness) is generally present. The retinal vessels are attenuated. The presence of pigmented retinal spots give this disease its name. Central complicated cataract, glaucoma and vitriol degeneration may all occur. The disease announces itself with poor night vision and peripheral field loss before 30 years of age. Patients have a color vision abnormality, especially difficulty in distinguishing the color blue from the others. The rod and the cone responses of the ERG are eventually extinguished (see section 2.4). The mode of disease inheritance can be autosomal recessive, autosomal dominant or x-linked recessive (see section 2.4).

3. Cone Rod Dystrophy (CR Dyst)

CR Dystrophy is characterised primarily by severe cone abnormalities.[5,24]. The first report of a patient with cone degeneration was published in 1870, under the title of "an unusual form of retinitis pigmentosa". The following describe the characteristics of CR Dyst.

Initially, the patient exhibits a loss of central vision and later on, exhibits a loss of peripheral vision. The patient has good night vision, but poor day vision. Myopia (short sightedness) is present in both the eyes and the retinal vessels are attenuated, but to a lesser degree than RC-Dyst. There is a pigmentary degeneration with clumping, especially centrally, reflected in poor

loss of central vision. Typically a macular lesion occurs with a bull's eye appearance. ERG response suggests a profound cone system deficiency, the cone responses being more extinguished than the rod responses. However, the color vision in contrast is remarkably intact. The disease is inherited in an autosomal dominant or an autosomal recessive manner (see section 2.4).

4. Pseudo Retinitis Pigmentosa (Pseudo RP)

This class of disorders is due to infectious diseases, exhaustive disorders, drug induced retinal degeneration and exogenous causes[5]. This disease generally is a clinical presentation similar to RP in function and morphology. The following are some of its characteristics briefly.

The patient may have a history of prior good night vision, and an asymmetry between the eyes. The disease may be familial, but not genetically induced. Portions of the retina may be destroyed, but the remaining portions will tend to be completely normal. At one extreme, the disease involving most of the retina can lead to extinguished ERG responses (see section 2.4).

So far, the FRD disease subclasses and their characteristic features were described. During the diagnostic stages, each of these features is derived from the patient as a result of performing tests on him/her. The following section describes the tests in detail.

2.4 Tests Employed in the FRD Diagnosis

This section describes the tests employed in the diagnosis of Familial Retinal Degeneration diseases. The list is not necessarily an exhaustive one and the order given here is not the order in which the tests are actually performed. This order may vary from patient to patient, and not all patients may take all the tests.

Moreover, the physician has the option of choosing, ordering and repeating the tests to his satisfaction. In general, the tests can be classified as history, clinical or lab-tests. Talking to the patient and deriving results from the conversation is called history taking. The examinations done on the patient by the physician is termed clinical. And the tests done in the laboratory are called lab-tests. Following is a list of tests. In the following, each test is briefly explained. For more details, the reader may refer to appendix A.

- History

1. Progression

Progression is a test, the result of which is determined from discussions with the subject and from comparison of data, over the years and lends itself to the minimum discriminable measures. The result of this test is usually obtained after a conversation with the patient. The test answers questions regarding the progression of the patient's disease, whether the disease has remained stationary or progressed for worse over the past years. For example, a disease in a patient may be

progressing at the rate recognizable as a change every 5 years. There are two types of progression, the central progression, of the cone system disease and the peripheral progression, of the rod system disease. The central progression describes the way in which the patient's central vision has been affected (i.e. if the patient is able to see things straight ahead of him/her) and the peripheral progression describes the way in which the patient's peripheral vision has been affected (i.e. if the patient is able to see things at his/her sides).

2. Nyctalopia and Hemeralopia

Nyctalopia and Hemeralopia constitute the subjective resultant of diffuse rod and cone system diseases and provide a means for estimating the times of onset of the subjective component of the disease process.

Nyctalopia means night blindness. This test is a part of the history taking procedure. The result of this test is obtained after having a conversation with the patient. Its results tell us whether a patient has poor night vision or not. Furthermore, if he does have a poor night vision, whether the defect remained stationary or has progressed.

Hemeralopia means day blindness. This test is also a part of the history taking procedure. The result of this test is obtained after a conversation with the patient. Its results tell us whether a patient has poor day

Figure 2.2: *The Family Tree of a patient with CSNB.*

vision or not. Furthermore, if he does have a poor day vision, whether the defect remained stationary or has progressed.

3. Family tree

An important part of history taking is to get the patient's genetic background. The physician talks to the patient about the latter's family, and draws a family tree, as shown in figure 2.2 [28]. In the figure, the member of the family who is the first one to be examined by the physician is called the proband. From the family tree, he decides the type of inheritance of the disease, if any.

The type of inheritance may be autosomal recessive, autosomal dominant or x-linked recessive, or the patient may show no familial tendency[23].

The type of inheritance is autosomal recessive if the following properties exist. The disease appears only in siblings, and not in their parents, offspring or other relatives or if the disease appears sporadically or as an isolated case. There is an equal sex incidence and there is a possibility that the parents of an affected child may be related.

The characteristics of autosomal dominant type of inheritance may be summarised as follows. The disease appears in every generation and is transmitted by an affected person to 50% of offspring. There is an equal sex incidence and unaffected persons do not transfer the disease to their children.

The characteristics of x-linked recessive type of inheritance are as follows. The disease is present mostly in males. The disease is passed on from the affected male through all his daughters to half their sons, but never transmitted from the father to the son. The affected males in a family are brothers or related to one another through carrier females.

- Clinical Examinations

They include a number of tests which have specific significance in delineation of FRD group of diseases.

1. Snellen Central Acuity

Snellen Central Acuity (referred to as 'acuity' hereafter) is measured by the ability to identify objects. For example, an acuity of 20/40 means that a patient can just identify an object placed at a distance of 20 feet, what a normal person could clearly identify the same at 40 feet.

2. Ocular Media

This is a clinical examination performed by the physician. The media are those tissues and fluids, through which light passes on its way to the receptors, the outer segments of rods and cones in the outer most part of the retina. Inflammatory and degenerative debris within the vitreous body in the center of the globe and liquefaction of the body with collagen fibres into bands tend to occur in proportion in involvement of the rod system in a FRD system.

3. Visual field

The visual field is the total area of space which is visible to a person while the eye is fixed in one direction [33]. The 'hill of vision' (see figure 2.3) is a 3 dimensional plot of this field, with minimum discrim-

inables along the y axis, degrees of fixation along the x and z axes. Patterns of field loss characterise the anatomic and physiologic focus for a disease process.

The subject maintains fixation on a central target as the test target representing the isopter under consideration is brought from seeing to non seeing retina in a 'dynamic field study' or as the significance of the target is increased until it is just seen.

The optic disc is located on the fundus of the eye devoid of retinal nerve fibres, inducing a physiologically blind area of the retina. The field defect is absolute, with no vision at all in its depth. Field defects may be present to a certain degree, from faint relative undulations of the surface to deep craters. The shape, location and depth all denote the characteristics assisting in locating the cause and occurrence of the pathology.

The result of this test is drawn on a field chart, as shown in figure 2.3. In this figure the patient's area of visibility in the form of a "hill of vision" is shown above. This is mapped to a field chart as shown below. The figure shows that the smaller targets are visible only when they are closer to the point of fixation. But the larger targets are visible out to the periphery. This examination can be repeated by changing the size and intensity of the target in different back grounds.

Figure 2.3: The Visual Fields chart.

4. Fundus Oculi

This is a clinical examination performed on a patient, wherein the physician, with special instruments, observes the inner surface or the fundus of the eye. The morphological appearance of the inside of the eye is viewed through the pupil. The physician can thus identify and locate abnormalities in form including a distortion or loss of normal anatomical detail. RP derives its name from loss of retinal pigment epithelial cells and migration of these cells into the retina adjacent to retinal veins. It is noted predominantly where the rod system deficiency predominates.

- Lab-tests

1. 100-Hue

The Farnsworth-Munsell 100 hue test offers a method for testing the color discrimination capability.

The test consists of ordering 93 plastic caps of different colors. The detailed procedure involved in performing this test is given in [11]. The result of this test gives an error score for color discrimination for each of the color plastic caps. These scores are plotted on a polar diagram giving a color defective pattern as shown in figure 2.4. The final error score in conjunction with the age, and the axis of the deficit score give

(a)

(b)

Figure 2.4: The color defective patterns of patients.

the level of significance of the color deficit. Figure 2.4 demonstrate color defective patterns. A congenital color defective pattern is shown in figure 2.4(a). Figure 2.4(b) shows acquired deficit characteristics of rod-cone disease of the autosomal dominant sort in a 26 year old male with advanced field loss and profound field loss and ERG deficit, but 'normal' Farnsworth-Munsell 100 hue polar diagram without clear axis.

2. Electroretinogram (ERG)

A flash of light elicits a distinctive electrical response from the human eye to the brain. This is called the electroretinogram (ERG).

On the retina of the eye, there are two types of receptor cells, called the rods and the cones (section 2.3). The responses of these rods and cones to light of various colors, frequencies and intensities and the electrical response of associated retinal cells are measured in this test. There are two different measurements made from each response. They are the amplitude and the implicit time of the response. The measurements are made under different conditions and based on this, there are different types of ERG tests. The response of the dark adapted retina includes the responses of both the rod and the cone system. Whereas the responses of the light adapted retina and rapid stimuli includes the response of primarily the cone system.

Figure 2.5: *The Electrorretinogram responses.*

If the diagnosis is to be made for one patient, the test values are compared with the normal and for the range expected in each potential disease state. A normal ERG response is shown in figure 2.5(a). In figure 2.5(b) a very extinguished ERG response for a patient with CSNB is shown.

Having explained the FRD diseases, their classifications, characteristics and the tests employed for their diagnosis, the next step is to develop a knowledge base system to diagnose FRD diseases. The following section describes the objective of this KB system.

2.5 Objective of the FRD KB System

In this thesis, the FRD KB system was designed and implemented as an initial prototype. This section gives a general overview of the FRD KB system prototype. The objectives of the FRD KB system prototype are as follows. First, to examine the feasibility of using the layered model approach to the design of the FRD KB system. Next to develop a system which could be employed as an assistant by an expert while performing diagnosis and research in the FRD domain. Clearly the FRD KB system is an initial prototype, hence it sets the stage for the development of an expanded prototype and finally an end product. Issues such as discovery of new subclasses of FRD diseases, optimisation of number of tests to be performed on a given patient for the reduction of health care costs, etc. are not treated in this thesis, and left for future studies as explained in chapter 7. The existing system diagnoses a patient as follows.

First, a number of tests are performed on the patient, by the physician. Next, the results of these tests are given to the system. To each result of the test is assigned a confidence factor. This test confidence factor is a measure that reflects how much confidence the doctor has in the result of the test. In addition, associated with each test is a measure of the significance of the test in the diagnosis of each disease. This is called the test-disease significance factor. The system uses the result of the test, the test confidence factor and the test-disease significance factor and calculates the probability of the disease (this is explained in detail in chapter 6).

At the end of performing the required tests, the system gives a ranked output of the diagnoses. This initial list of diagnoses can be further refined either by performing some more tests or by repeating the already performed tests. The system then repeats the process all over again. The system can be made to stop when the user is satisfied with the level of probability reached.

The following chapters describe the design and implementation of the knowledge base system for the diagnosis of the FRD diseases.

Chapter 3

Design and Development of KB Systems

3.1 Introduction

A Knowledge Base System is a system which manipulates “knowledge” in order to perform tasks. The knowledge - also called expertise - consists of: knowledge from the literature, experience of the experts etc. and this expertise is used for problem solving in general.

The next section describes the problem solving expertise with examples. It is very important to be able to develop a model for the expertise; as once this is accomplished, the model could be represented in the computer and the user can benefit from the value-added capabilities offered by the computer. An overview of a model for the expertise is given in section 3.3. In section 3.4, various knowledge

representation schemes are described. The user interface capabilities for a KB system are described in section 3.5. And finally in section 3.6 the actual stages of a KB system development are described.

3.2 Problem Solving Expertise

Problem solving expertise, is the knowledge (both-knowledge of the books and the expert's expertise) in a particular domain, used to solve problems in the same domain. The expertise can be used to solve a number of problems. In the following, various classes of problems are given.

- Fault finding: There are systems which are capable of finding the problem or fault, for example: finding hardware faults, semantic errors of a program etc[14,18].
- Teaching : Some systems are used to explain problems and guide a student through its solution[8].
- Diagnosis: Systems can diagnose the causes of a problem and sometimes recommend a solution too. For example: diagnose and recommend therapy in medical problems, diagnose problems in photolithography etc[10].
- Adviscry: There are systems which can advise its user during consultation; for example: in evaluation of mineral salts, selection and use of subroutines in the large FORTRAN library etc[4].

- **Planning and Designing:** Some systems also aid in the areas of planning and designing. For example: planning an aircraft's route, planning of the manufacturing steps in a plant, designing special purpose electric motors etc[4].

In order to develop the KB system, it is important to model the expertise in such a way that the resulting model preserves the extreme flexibility of the experts in performing tasks and solving problems. In the following section, the model for the expertise presented by Wielinga et al[36] is reviewed.

3.3 A Model for Problem Solving Expertise

The expertise consists of 2 levels in general. A domain level, where mainly objects and their relations are described and inference level(s) containing the descriptions of the ways of using the domain knowledge. The models of expertise are distinguished mainly by the amount of dependency between the 2 levels.

There are systems [10,35] in which the 2 levels are strongly dependent i.e. the inference is domain dependent and therefore it is not possible to clearly identify the inference and its interaction with the domain knowledge. It is not easy to see how the knowledge interrelates and interplays to implement the system's problem solving tasks. Moreover, if at a later stage, the domain knowledge is changed, altered or updated, the inference has to be adjusted too. Thus the system's flexibility is greatly reduced as it does not permit the flexibility that an expert employs when

solving problems. These problems clearly arise due to the fact that the expertise is not structurally modelled.

There are other systems which do employ this 2 level modelling. In such systems [29,34] the domain knowledge forms the first level. The higher level is an inference layer, which is supported by the underlying domain level.

Wielinga et al[36] extend such a separation of knowledge into 2 levels a step further, for a refined distinction of the different types of knowledge at various levels. There are several layers or levels at which the knowledge is described, based on the different types of knowledge they contain. Each level has its own organisational principles as described below.

1. DOMAIN LEVEL

In this lowest level, knowledge is represented as concepts, relations between concepts and structures built from relations.

For example, in the medical domain, objects or concepts could be - various disorders, symptoms, causes, complaints, laboratory data etc. Relations may be - causal (between causes and complaints), subsumption (between disorders) and empirical (between data and disorders).

The knowledge at this level is task neutral, i.e. it is not presented in a special way to support any task. It does not contain facilities to control problem

solving. The only constraint at this level is the axiomatisation chosen for a particular domain. This limits the kind of, say inference that one may wish to apply on the domain knowledge. One should therefore identify the appropriate axiomatisation with respect to the requirements of the higher layers.

2. INFERENCE LEVEL

This level of the model contains the inference steps to be done on the domain knowledge. It is a structure made up of 2 components - **Metaclasses** and **Knowledge Sources**.

Metaclasses describe: (a) what role(s) a concept can play in an inference process; (b) how the concept is to be used and (c) what its status can be during the various stages of problem solving.

For example, a particular "infection" can play the role of an explanatory object of certain facts; a hypothesis to be verified or a solution in a problem solving process. Figure 3.1 gives a list of metaclasses, their roles and examples (Wielinga et al[36]). For example in that table, symptom is a metaclass, with examples - fever, pallor, headache and complaints. The role of symptom in the inference is in triggering hypothesis.

Knowledge Sources are descriptions of the ways in which a domain relation

meta class	examples	role
symptom	complaints fever pallor headache	in triggering hypothesis
manifestation	hypotension fever	in verification of hypothesis
factor	alcohol consumption medication age	as supporting evidence
hypothesis	pathological state causal object	in finding a diagnosis
clinical diagnosis	syndrome	solution
causal diagnosis	disorder	

Figure 3.1: *Metaclasses in a medical domain.*

Domain relations

subsumes(O(i),O(j)).
has-attribute(O(i),A(i,k)).

Meta Classes

hypothesis(O(i)).
attribute(A(i,k)).

Knowledge Sources

%refinement-by-specialisation

refines(O(j),O(i)):-
 hypothesis(O(i)),
 hypothesis(O(j)),
 subsumes(O(i),O(j)).

%generalisation

generalises(O(i),O(j)):-
 hypothesis(O(i)),
 hypothesis(O(j)),
 subsumes(O(i),O(j)).

Figure 3.2: *Examples of a few Knowledge Sources.*

can be used to make inferences. For example the relation "subsumes" can be used in a number of ways. The knowledge sources are thus, handles for the strategic knowledge on the domain knowledge.

Figure 3.2 gives a few examples of knowledge sources. In the figure there are 2 knowledge sources. For example, the first one is a knowledge source describing the process of "refinement". It has a metaclass - hypothesis and a domain relation - subsumes. The 2 together form a knowledge source - refine which describes the process of refinement through specialisation.

The inference structure relates the metaclasses and the knowledge sources.

Data - -(abstraction)- → evidence - -(association)- → hypothesis
 - -(refinement)- → diagnosis.

Figure 3.3: *The Inference structure in NEOMYCIN.*

It indicates what inferences are possible and what are not (implicitly). It describes what inferences can be made, but not in what order. In other words it is a static structure, which does not specify a strategy.

Figure 3.3 shows an inference structure. In this figure, data, evidence etc. are metaclasses; and abstraction, association etc. (those in the paranthesis) are knowledge sources. The steps of inference are - abstraction of data, formation of hypothesis and finally the refinement of hypothesis to get the final diagnosis. This is an inference structure for KB system - NEOMYCIN (see section 4.3),[8].

3. TASK LEVEL

The next step is to define a strategy which determines how and when certain inference steps will be made. A common representation is a "goal-tree". All actions at the task level refer to metaclasses and knowledge sources. Tasks are goal directed and they are organised in a hierarchy - from general to specific.

Figure 3.4: A part of the task hierarchy in PDP KB system.

Figure 3.4 gives an example of a part of the task hierarchy for the physics problem solving system PDP [36]. Here, "solve problem" is the main task. To accomplish this task, 3 subtasks have to be performed, "analyse", "solve" and "evaluate". To achieve these subtasks, other subtasks have to be done as seen in the figure and so on.

4. STRATEGIC LEVEL

Although models with fixed problem solving strategies are quite powerful, they are not as flexible as the expert. This is because the fixed strategy always pursues similar issues, asks the same questions, and fails in the same ways. Models need to be made more flexible.

Figure 3.5: *The structure of a Strategic Reasoning process.*

The solution is to have a planning component in the model, which dynamically generates a goal tree. Such a planner uses the knowledge about general problem solving methods, such as means end analysis, heuristic classification etc.

In general, the planning component must execute the plan, monitor the progress of the plan and remedy possible difficulties.

An example for flexible strategic reasoning given in [36] is shown in figure 3.5.

The figure shows the planning, monitoring, execution and diagnosis stages done using the feedback methods. The planning component generates a strategy dynamically. This strategy is executed and monitored. If during execution, a problem arises, the system tries to repair it. If it cannot, it generates another strategy and the process is repeated until the problem is solved.

After the development of a problem solving expertise model, one should look at the various ways of representing the knowledge in the model. The next section describes several knowledge representation schemes.

3.4 Knowledge Representation

When talking about knowledge representation, two things come to mind. The knowledge representation schemes and the implementation of the knowledge representation schemes.

The knowledge representation schemes are the higher level concepts of the representation of the knowledge for human understanding. These schemes can be implemented using tools that range from the more sophisticated "shells" to the simple programming languages like LISP, PROLOG etc. In the following, an overview of the various knowledge representation schemes is given.

For all diseases x and y,
 if x is the category of y, then
 for all symptoms s,
 if s is associated with x, then
 s is associated with y.

$$\forall x \forall y ((\text{DISEASE}(x) \wedge \text{DISEASE}(y) \wedge \text{CATEGORY}(x,y)) \implies$$

$$(\forall s ((\text{SYMPTOM}(s) \wedge \text{ASSOCIATED-WITH}(s,x)) \implies$$

$$\text{ASSOCIATED-WITH}(s,y))))).$$

Figure 3.6: *Knowledge represented in predicate calculus.*

1. Predicate Calculus:

The first order predicate calculus can be used as a knowledge representation scheme. Here the knowledge is represented as an unstructured sequence of independent sentences. There are no specifications to indicate relations between sentences. However, there is no need to group them explicitly, as grouping only reflects their applications in problem-solving contexts.

An example is given in figure 3.6. In the top portion of the figure, a body of knowledge in english is given and the same piece of knowledge is represented in predicate calculus at the bottom of the figure.

There are certain applications, for which predicate calculus is the most suit-

able form of knowledge representation, even though they appear a bit complicated when compared to the other schemes. But they are not suitable for representing descriptive knowledge or taxonomical classifications.

2. Associative/Semantic Network:

An associative network is a collection of nodes and arcs. Nodes represent situations, places, processes, events, concepts or objects; and arcs represent binary relationships. The relationships could be causes, effects, associations etc. Schubert et, al[30] have developed a topically oriented semantic net system in which knowledge is topically organised around concepts in the form of a tree.

Figure 3.7 gives an example of a network from the KB system - RX[3]. In this figure "heartattack" and "sedentary lifestyle" are both nodes. They are connected by an arc, which represents the fact that - "heartattack is one of the causes of sedentary lifestyle and that it increases sedentary lifestyle" The strengths of the relationships are shown on the arcs by numbers and the "+" or "-" sign indicate whether the cause increases or decreases the effect.

The associative network scheme captures the given conceptual knowledge adequately. Taxonomic classifications, hierarchies, inheritances etc. can be represented easily. But this type of representation is possible only if true causal/associational relations are known. The networks form the highest

Figure 3.7: A hypothetical causal model of the coronary heart disease.

level of abstraction among schemes.

3. Frames:

A Frame is a structure consisting of a network of nodes and relations, used to represent a situation or a topic stereotype. Frames have Slots in them and slots contain information such as: findings, symptoms, causes, effects, exclusionary and sufficiency conditions etc. of that frame. In short, this scheme permits the co-existence of both the factual and the reasoning knowledge. Moreover, frames provide a mechanism of inheritance. Inheritance is a property where information is automatically inherited from one frame to a number of other related member frames. This prevents the need to repeatedly describe this information in each of the member frames.

In figure 3.8, a network of frames is shown. A number of frames (with slots) linked to each other are shown. Slots contain information about the frame in which they are present and also the links between various frames as seen in the figure.

Frames adequately capture the given structure. They facilitate easy retrieval of closely related information. They reflect the expert's complex set of associations. Taxonomical classifications can be represented through specialisation slots. Frames are thus a nice and compact method of knowledge representation.

Figure 3.8: A network of frames.

4. Production Rules:

A Production Rule is of the form - "IF... THEN... ". The "if" part is called the antecedent or the premise and the "then" part is termed the consequent or the action.

Examples of a few general rules are shown in figure 3.9.

Rules are simple and easy to write. But they do have some disadvantages. They do not capture the taxonomical classifications explicitly, but implicitly by repetition. This can be seen in the example, as in almost every rule, certain clauses are repeated to capture the knowledge clearly. Rules moreover do not adequately capture the given conceptual knowledge. In order to represent complex situations, rules often have nested clauses, which make them very complex to process. Moreover, rules with more than 6 premises become extremely complex to process and in some domains 6 or more variables interact at a time. Due to the following reasons, rules are seldom used as the only knowledge representation scheme.

Any one (or more) of the above 4 representation schemes can be used to represent the knowledge at the various levels of the model. And this depends upon the type of the knowledge involved at the different levels. So schemes are chosen for each of these levels, based upon the knowledge to be represented in that level

- Rule 1: **i f** the patient suffers from A **and**
 the patient does not exhibit symptom Z
 then the patient is not likely to be suffering from disease B
- Rule 2: **i f** the patient suffers from A **and**
 the patient does not suffer from B
 then the patient could be suffering from disease C
- Rule 3: **i f** the patient suffers from A **and**
 the patient does not suffer from B
 then the patient could be suffering from disease D
- Rule 4: **i f** the patient suffers from A **and**
 the patient does not suffer from C **and**
 the patient does not suffer from D
 then the patient could be suffering from disease B

Figure 3.9: *A few general production rules.*

and which scheme can best represent that knowledge. For example, Takenouchi et al[32] have come up with an integrated knowledge representation language - MRL (Modular Representation Language) - which has used the advantages of each of the schemes above.

The development of a model for the problem solving expertise and the representation of the knowledge in that model does not make a KB system complete. A vital aspect of any knowledge base system is how the users of that system interact with it. In order to facilitate this interaction, several user interface facilities can be employed. In the following section, some of these facilities are enumerated.

3.5 KB User Interface

A KB system can be designed to have a number of user interface utilities. They help the users, during both - the general usage of the KB and the KB development phases. The following are some of the interface capabilities.

1. Explanation Facility:

Explanation facilities are the utilities of a knowledge base system which can explain the problem solving strategies of a knowledge base system, and also explain, how and why the system follows those strategies. KB systems with simple explanation facilities are able to answer questions like:

- why they arrived at certain conclusions

- why they asked certain kinds of information from the user
- how they deduced a particular goal
- what the current status of the KB is

and the more complicated systems can answer questions like:

- why didn't they take a particular step
- what would happen if a certain hypothesis turned out to be true/false and so on.

These facilities are the main reason of acceptance of the KB system by the users. They not only help the users in general, but aid in KB development too, i.e. in updating or modifying the existing knowledge .

2. Consistency Checking:

The system can be designed to give advice regarding errors. Such validity checks are of several types, and they help keep the knowledge base consistent. The following describes some examples of how a system with consistency checking programs maintains consistency in the KB.

The system might point out that the finding is in error (for example: a weight gain of 50lbs in 48 hrs [29]). It might note that the new information is inconsistent with other known facts (for example: the presence of red cell casts in the absence of hematuria [29]). The system might also indicate that a new finding contradicts a conclusion already drawn about the problem.

Finally, the system may check against the misinterpretation of data by the user (for example: "dark urine" - may be blood in the urine; or due to the presence of bile or due to anthocyanins from beets [29]).

Consistency checking programs help the user check and correct his response as well as help the programmer during the development of the KB, i.e. in updating and modifying the KB.

3. Graphic Capability:

This is the capability of the system to display its reasoning and explaining strategies in the graphics form.

There are systems which display their reasoning and explaining strategies as AND/OR logical trees. They also display their domain knowledge in the form of a tree or a network.

Graphic capability makes it a whole lot easier to interact with the system. Instead of having a lengthy text of information, they are much easier to understand and grasp; as a picture is worth a thousand words.

The model for the expertise, knowledge representation schemes and user interface utilities for a knowledge base system have been described in the above

sections. The next step is to develop and implement a knowledge base system based on the model above. The following section describes the different phases involved in the development of a knowledge base system and how it is actually implemented.

3.6 KB System Development Phases

In general, Citrenbaum et al[7] have identified four phases in the development of a KB system as described below.

1. Problem Identification:

The very first stage of the KB development, would be to select and identify the problem clearly. This has been discussed broadly in Chapter 2.

2. Initial Prototype:

The major objective of the initial prototype stage is to quickly demonstrate the technical and the economical feasibility of the desired system. Expert system shells are generally used to build such prototypes.

Expert system shells are tools that support a minimum set of features - knowledge representation scheme(s), inference mechanism, a means of describing a problem and a way to determine the status of a problem when it is being solved. Rapid prototypers with a minimum but essential set of

features are suitable to build the initial prototype. The following are some of the essential features of a shell:

- To support the domain level - The shell must provide at least one clear knowledge representation scheme. If possible, it must provide translation to other schemes for portability. It must be able to represent the domain knowledge as clearly as possible, able to be understood even by a novice.
- To support inference levels - The shell must support one or more of the following inference types:
 - Deductive - this is a type of inference, where an evidence for a hypothesis deduces a conclusion. For example, a hypothesis ($P \rightarrow Q$) and an evidence (P is true) generate a conclusion (Q is true).
 - Inductive - this is a method of learning or a technique of knowledge generation from existing knowledge and/or data.
 - Abductive - this refers to the formation of a totally new hypothesis from observations.

The inference techniques can be implemented using either the forward or the backward chaining method (in rule based systems) or simple reasoning cycles (in other systems).

Besides these, a more elaborate shell will support uncertain reasoning, truth maintenance, dynamic hypothesis generation etc.

- To support the user interface utilities - The shell must support the standard default features and explanation facilities. It should aid in prototyping. A more sophisticated shell might provide graphics, colors, fonts, slogans etc.

Following the demonstration of an initial prototype, and its approval, the expanded prototype is built.

3. Expanded Prototype:

The major objective of the expanded prototype stage is to develop the full set of the KB system functions required to deal with the complete problem.

The initial prototype is expanded in the following areas.

- Knowledge acquisition - The system must be provided with knowledge acquisition tools, to acquire more knowledge and expand the knowledge base (both domain and inference), as time goes by.

There are systems [10,25] which acquire knowledge by having a dialogue with the user. Some other systems [26], present the user with a number of situations, ask the user to comment on them and thus acquire knowledge from it. Finally there are systems [3], which automatically acquire knowledge using the existing knowledge and data.

But none of these systems are complete. They do have limitations, such as incomplete data and knowledge, inability to account for special cases etc. And knowledge acquisition is still a topic of research, at its infancy.

- Systems integration - The KBs must provide links to software products such as databases and spreadsheets[31].

Adlassnig et al[1] have developed a medical KB system, which uses a large hospital database.

- Other features - The system can be designed to support other sophisticated features, like - real time performance, elaborate user interface facilities etc.

After the approval of the expanded prototype, the final stage of the KB development is to come up with a product ready for delivery and everyday use.

4. Delivery System:

A Delivery system differs from the expanded prototype in that, it must run on inexpensive hardware (such as PC), satisfy more stringent performances and robustness requirements. The system must be made faster, smaller and portable at the time of delivery. The software engineering issues of performance become much more important at this stage. The major problem

encountered at this stage is the design and function tradeoff.

The above sections have provided a framework for the modelling, representation and development of a KB system. Several existing medical knowledge base systems are reviewed in the following chapter.

Chapter 4

Some of the Existing Medical Knowledge Base Systems

4.1 Introduction

Medicine is a branch of science, with a well organised knowledge acquiring when compared with other cultural activities of humans. The daily accrue of medical knowledge is enormous and is steadily increasing. This is due to the development of progressive technologies and the priority extended to health care. This is one of the reasons, why the first knowledge base systems originated in medicine and were exported from here to other branches.

In the subsequent sections of this chapter a number of medical KB systems have been reviewed. This list is not necessarily an exhaustive one. These systems are reviewed using the framework established in the following section.

4.2 A Framework to Identify KBS Features

A number of existing medical knowledge base systems (KBS) have been reviewed in the following sections. The following framework has been developed to identify their features.

In each case the system architecture is described first. In this, the system components are described as a model with several levels. The different levels identified in the model are the domain, inference and the task levels. The design, representation and the implementation of each level is explained. Not all the systems have all the levels in their model, and only those that are present are described.

Next, the user interface utilities of the KB system are given. The system may have one or more utilities (such as explanation facilities, consistency checking) or none at all. Lastly, the KB implementation environment is given. This includes the general implementation details of the system and its future extensions, if any.

The following sections have used the framework given above, to review some of the existing medical KB systems.

4.3 MYCIN

MYCIN [10,17] is a medical KB system which diagnoses certain antimicrobial infections and recommends drug treatment.

Initially MYCIN was developed as a thesis project by Edward H. Shortliffe at Stanford University. Later on it spawned an active research group. In 1977, Randall Davis and Bruce Buchanan joined Shortliffe in this research. Although it was never implemented for routine clinical use, its capabilities served as a basis for several new experiments at Stanford University. The name 'MYCIN' was taken from the common suffix shared by several of the drugs used in treatments - kanamycin, erythromycin, clindamycin etc.

The structure and working of MYCIN are given below.

- Architecture of MYCIN

In MYCIN, a clear model for the expertise does not exist. The separation between the different types of knowledge into several layers is not so clear. But, the knowledge in MYCIN can be classified into the following levels.

1. Domain level:

At the domain level, the knowledge base consists of domain specific rules. Examples of such rules are shown in figure 4.1. The first rule states that - "if the site of the culture is throat and the identity of the

- Rule 1:** **IF** 1) the site of the culture is the throat
 2) the identity of the organism is streptococcus
 THEN there is suggestive evidence (0.8) that the
 subtype of the organism is not group-D
- Rule 2:** **IF** 1) the infection is meningitis,
 2) the subtype of meningitis is bacterial,
 3) only circumstantial evidence is available,
 4) the patient is atleast 17 years old, and
 5) the patient is an alcoholic
 THEN there is suggestive evidence that diplococcus-
 pneumoniae is an organism causing the
 meningitis.
- Rule 3:** **IF** 1) the site of the culture is one of the nonsterile
 sites, and
 2) there are rules which mention in their premise
 a previous organism which may be the same as
 the current organism
 THEN it is definite (1.0) that each of them is not going
 to be useful.

Figure 4.1: A few rules from the MYCIN knowledge base system.

organism is streptococcus then there is a suggestive evidence (with a certainty of 0.8) that the subtype of the organism is not group-D".

The rules however, are not purely domain specific but they do have some inference embedded in them. For example consider the second rule in figure 4.1 Ordering of the clauses 1 and 2 gives inference steps describing the top down process of refinement. Four major subgoals are

established here - Is there an infection ? Is it meningitis ? Is it bacterial?

Is it *Diplococcus Pneumoniae* ? - which form a specialisation hierarchy. Thus the inference and the domain knowledge are interrelated and embedded, making them very unclear.

There are some domain independent rules - called the meta knowledge (rule 3 in figure 4.1) in MYCIN. The rules are used to prune the list of domain rules and this prevents the exhaustive invocation of all the domain rules during inference. Researchers have called the domain and the meta knowledge the 2 levels of the model. But this is not so, as the meta knowledge does not describe a task or a strategy for diagnosis. So the meta knowledge cannot be considered equivalent to an inference, task or a strategic level, with respect to the four level model described in the previous chapter.

The scheme used for knowledge representation is "production rules". Every antecedent of a rule is associated with a certainty factor (CF). This describes the measure of how certain the physician is that the antecedent is true. Every rule is also associated with a certainty factor. This gives a measure of how certain the physician is that the conclusions of the rule are true, if all the premises of the rule are true. This helps the representation of judgemental knowledge and assists inexact reasoning.

All rules are implemented in INTERLISP.

2. Inference level:

An explicit task or a strategic level is absent in MYCIN. The way in which the rules are ordered determine their order of invocation. All rules are invoked exhaustively.

The type of inference is deductive, and it is called inexact reasoning in MYCIN. Deduction is a process wherein a hypothesis and an evidence produce a conclusion. The domain rules are used to deductively infer conclusions. During the process of inference, if the required information is not available, it is derived from the patient. When all rules are applied and the inference has come to an end, the final diagnosis and a list of treatments for the same are output. The final diagnosis is made with a particular certainty. The processes involved in calculating the certainty factors for the diagnoses are as follows. Every clause in a rule in MYCIN is associated with a certainty factor. During the process of reasoning, certainty factors are propagated from one rule to the other and a final certainty factor (CF) is assigned to the final diagnosis. Minimisation and maximisation procedures are used to propagate CFs and calculate the final CF. If a rule has more than 1 clause in its premise, the total CF of the premise is the minimum of them all. This is called the process of minimisation. And if a conclusion is supported by more than one rule, the CF of the conclusion is the maximum of the CFs of

all those rules. This is called the process of maximisation.

The inference is implemented as a backward chaining program. The rules are invoked in a goal directed manner. The backward chainer is also written in INTERLISP.

- User Interface

MYCIN's user interface has the following capabilities.

- a question answering program:

This program displays information during consultation. It explains the occurrence of an event. It also searches the knowledge base for some requested information.

- an explanation program:

This program explains "how" a goal was deduced and "why" certain information was requested from the user.

The system does not have any graphic capabilities or consistency checking programs.

- KB Implementation Environment

MYCIN runs on Tenex Operating system on DEC KI-10. It has approximately 450 rules, about 130K words of non shared code.

Extended versions of MYCIN have a knowledge acquisition tool - TERESIAS [25] - to assist in acquiring knowledge. Moreover, the domain independent portions of MYCIN were separated to form an expert system tool EMYCIN, which was used to develop other KB systems at Stanford.

4.4 NEOMYCIN

The NEOMYCIN [8,17] system is an attempt to rearrange the knowledge structure and the diagnostic strategy of MYCIN (see section 4.3) to make it suitable for teaching purposes. Its domain and area of application are the same as that of MYCIN [10].

William J. Clancey and Reed Letsinger were NEOMYCIN's principle researchers. It was developed at Stanford University in 1981. Its main goal was to provide an explicit basis for teaching with an explicit representation of strategic knowledge.

The structure and working of NEOMYCIN are described below.

- Architecture of NEOMYCIN

NEOMYCIN's model of expertise is the closest to Wielinga et al's[36] model of expertise; as the latter is just an extension of the former. The layers of the model are described below.

1. Domain level:

The domain level consists of the domain knowledge. The domain knowledge consists of: etiological taxonomy of disease classifications, state categories and observations. Figure 4.2 shows the domain knowledge. In this figure, observations are linked to etiologies via state categories. A portion of the disease classification is shown in figure 4.3. This figure explains the patient's illnesses (etiologies) in terms of the classifications of the diseases. For example, infection is one of the disorders and meningitis is one of its types. Further, meningitis could be acute or chronic; and acute meningitis could be bacterial or viral; and so on. State categories are intermediate diagnostic states. And the observations are the various patient findings (such as signs, symptoms, history etc.). The disease process knowledge describes the diseases in the classification (such as the extent, course, location of the disease). This is shown in figure 4.2, as an extension of the disease classification.

At the domain level there are 2 types of knowledge representation

Figure 4.2: *The structure of NEOMYCIN's domain knowledge.*

Figure 4.3: A portion of the etiological taxonomy in NEOMYCIN.

schemes involved. The diseases in the classification tree (figure 4.3) are represented as frames. Slots in these frames carry the disease process knowledge. Observations are linked to etiologies via state categories and these links are represented using the production rules. For example, there are screening rules to represent the commonsense knowledge; trigger and hypothesis rules to link findings to etiologies[8].

All rules and frames are implemented in INTERLISP.

2. Inference level:

NEOMYCIN has an inference structure composed of metaclasses and knowledge sources. It has 2 metaclasses - findings and hypothesis; and the knowledge sources using the following domain relations - causal, trigger, subsumption, factual. The inference structure is shown in figure 3.3. As shown in the figure, the process of inference is as follows. Data is abstracted to get the evidence. The evidence is applied to the rules and various associations are triggered to deduce the hypothesis. The hypotheses are refined to get the final diagnosis.

At the inference level, pieces of inferential knowledge are in the form of rules - called meta rules. The type of inference is deductive.

The inference technique is implemented using the backward and the forward rule chaining programs; in INTERLISP.

Figure 4.4: *The static representation of NEOMYCIN's diagnostic task.*

3. Task level:

The task level consists of the overall diagnostic reasoning knowledge. It is in the form of a hierarchy of tasks. A static representation of the task level is shown in figure 4.4. In this figure, "make diagnosis" is the chief task. In order to achieve this task, 3 other subtasks have to be performed. They are, "identify-problem", "establish-hypothesis-space" and "process-hard-data". In order to complete these tasks, there are

other subtasks and so on.

Each task is represented as a collection of meta rules. Meta rules refer to metaclasses rather than the actual domain values, making them quite domain independent.

All the meta rules are implemented in INTERLISP.

- User Interface

NEOMYCIN has 2 explanation facilities. It can explain "how" a particular task was performed. It can also explain "why" a certain information was requested from the user. Besides these, the status of the KB, its current hypothesis, evidence for a hypothesis etc. can be viewed at any time.

- KB Implementation Environment

NEOMYCIN was built using the shell EMYCIN (Essential MYCIN, the domain independent portions of MYCIN). The implementation language is INTERLISP. Computational resources were provided by the SUMEX-AIM facility.

4.5 KB System in Dermatology

The principle task of this program [12] is the differential diagnosis of 9 facial skin tumours.

Fact: pearly-edge-is-a-key-symptom-for-basal-cell-carcinoma.

Rule: <patient> may-have <disease> if <patient> has-key-symptom-for <disease> and <patient> may-suffer-from <symptom> for-all <symptom> such-that <symptom> suggests <disease>.

Figure 4.5: *A rule and a fact from the KB system in Dermatology.*

The main aim of the researchers was to illustrate the principles and methods involved in setting up a knowledge base system for use in dermatological diagnosis and exploiting its advantages. This area was primarily chosen because it was small and well defined.

The structure and working of the system is given below.

- Architecture of the KB system

Though a layered model was not designed for this system, a model with the following layers can be identified and described in the KB system.

1. Domain level:

At the domain level, the system consists of domain specific rules and facts which describe relations between patients and symptoms, symptoms and diseases etc. Examples of such rules and facts are shown in

figure 4.5. In the figure, the first part illustrates a fact, which states that "pearly edge is a key symptom for one of the facial tumours basal cell carcinoma". The second part shows a rule, which states that "A patient may have a disease if the patient has the key symptom for that disease and if the patient suffers from all the symptoms suggesting that disease".

The rules and facts are implemented in micro-PROLOG. The resulting program is used in the expert system shell APES.

2. Inference level:

An inference structure with explicit metaclasses and knowledge sources is not described. Steps of inference are not stated explicitly. The system uses a deductive type of inference. The hypotheses are deductively inferred from the domain knowledge, by triggering the appropriate rules.

The inference is provided by the APES expert system tool in micro-PROLOG. It is implemented as backward chaining program.

3. Task level:

The system has a simple task for diagnosis. The procedure involved in the diagnosis is as follows. The system first checks for the presence of any of the 9 key symptoms corresponding to the 9 facial skin tumours. If any one of them is present, the system pursues the hypothesis (i.e.

the facial skin tumour) corresponding to that symptom. More information pertaining to the hypothesis is derived from the patient. The related domain rules are applied and the hypothesis is refined to account for all the patient manifestations. If the system is successful in proving the presence of the hypothesis, it stops. Otherwise, it pursues another hypothesis, based on another highly likely key symptom and repeats all of the above.

This layer is represented using rules. An example is the rule shown in figure 4.5. This rule has a key symptom embedded in it, and it is used to differentiate one of the 9 facial tumours.

The rules and facts are implemented in micro-PROLOG in APES.

- User Interface

APES provides 3 types of explanation facilities for the system and they are as follows. The system is able to explain how it deduced a goal. It is capable of explaining why a particular information was requested from the user and also its inability to confirm a particular diagnosis.

- KB Implementation Environment

The system is built using the expert system tool APES (A Prolog Expert system Shell). The program is written in micro-PROLOG. In this study, a

NorthStar computer was used.

4.6 PIP

Cognitive insights gained from the study of clinicians' behavior were used to develop PIP (Present Illness Program) [29]. This is a computer program designed to take the present illness of a patient with edema and come up with diagnoses and treatments for them.

The research challenges that grew out of PIP have to a large extent defined research directions in AI at Tufts, New England Medical Center and M.I.T.

The system architecture and working are described below.

- Architecture of PIP

Though a layered model was not designed for PIP, a model with the following levels can be identified and described.

1. Domain level:

The domain knowledge is found in the following components of PIP - the patient specific data, the short term memory and the long term memory. They are shown in figure 4.6. The patient specific data contains the facts and data related to the patient. The long term memory contains a rich collection of general medical knowledge. And the short

Figure 4.6: *The architecture of the PIP knowledge base system.*

term memory is the working memory where the patient data interacts with the general medical knowledge.

The scheme of knowledge representation is frames. The general medical knowledge in the long term memory is represented as frames, one of which is shown in figure 4.7. This is an example of a "nephrotic syndrome" frame. Its slots contain information regarding the disease. For example 'low serum albumin concentration' is a finding or a symp-

tom in its favour; 'acute glomerulonephritis' is a cause of this disease; 'sodium retention' is one of its effects and so on. The slots in general, contain information about the disease represented by the frame such as the findings, causes, effects, exclusionary and sufficiency conditions etc. The exclusionary and sufficiency conditions are in the form of rules in the frame. The frames are connected to form a complex causal network in the long term memory. A part of such a network is shown in figure 4.8. The patient specific data are just the findings, similar to the findings in the disease frames.

2. Inference level:

This level is not explicitly stated in PIP, but the steps of inference are as follows. Data and facts are gathered from the patient, to form the evidence. The evidence is applied to the domain knowledge to deduce hypotheses. The hypotheses are refined to get the final diagnosis.

The type of inference is deductive. And it is implemented as a pattern matching program. This program deduces hypotheses as follows. Frames from the long term memory extend their daemons (findings etc.) into the short term memory, searching for a matching fact. The patient facts and data are brought into the short term memory, from the patient specific data. These facts and data match the daemons, and the corresponding frame is brought into the short term memory, as a hypothesis.

NAME: NEPHROTIC SYNDROME
IS-A-TYPE-OF: CLINICAL STATE
FINDING: LOW SERUM ALBUMIN CONCENTRATION
FINDING: HEAVY PROTEINURIA
FINDING: >5GRAMS/24HRS PROTEINURIA
FINDING: MASSIVE SYMMETRICAL EDEMA
FINDING: EITHER FACIAL OR PERI-ORBITAL SYMMETRICAL EDEMA
FINDING: HIGH SERUM CHOLESTEROL CONCENTRATION
FINDING: URINE LIPIDS PRESENT
MUST-NOT-HAVE: PROTEINURIA ABSENT
IS-SUFFICIENT: BOTH MASSIVE PEDAL EDEMA AND >5GRAMS/24HRS PROTEINURIA
MAJOR SCORING:
 SERUM ALBUMIN CONCENTRATIONS
 LO: 1.0
 HIGH: -1.0
 PROTEINURIA
 >5GRAMS/24HRS: 1.0
 HEAVY: 0.5
 EITHER ABSENT OR LIGHT: -1.0
 EDEMA
 MASSIVE AND SYMMETRICAL: 1.0
 NOT MASSIVE BUT SYMMETRICAL: 0.5
 ERYTHEMATOUS: -0.2
 ASSYMETRICAL: -0.5
 ABSENT: -1.0
MINOR SCORING:
 SERUM CHOLESTEROL CONCENTRATION
 HIGH: 1.0
 NOT HIGH: -1.0
 URINE LIPIDS
 PRESENT: 1.0
 ABSENT: -0.5
MAY-BE-CAUSED-BY:
 ACUTE GLOMERULONEPHRITIS
 CHRONIC GLOMERULONEPHRITIS
 NEPHROTOXIC DRUGS
 INSECT BITE
 IDIOPATHIC NEPHROTIC SYNDROME
 SYSTEMATIC LUPUS ERYTHEMATOSUS, OR
 DIABETES MELLITUS
MAY-BE-CAUSE-OF: SODIUM RETENTION
MAY-BE-COMPLICATED-BY:
 HYPOVOLEMIA
 CELLULITIS
DIFFERENTIAL DIAGNOSIS:
 IF NECK VEINS ELEVATED, CONSIDER: CONSTRICTIVE PERICARDITIS
 IF ASCITES PRESENT, CONSIDER: CIRRHOSIS
 IF PULMONARY EMBOLI PRESENT, CONSIDER: RENAL VEIN THROMBOSIS

Figure 4.7: The "nephrotic syndrome" frame in PIP.

Figure 4.8: A complex network of frames in PIP.

3. Task level:

At this level, PIP consists of a Supervisory Program (figure 4.6). The task of diagnosis is as follows. Patient data and facts are presented to the supervisory program. The program places them in the patient specific data. The program then consults the patient specific data, the short term and the long term memories to generate hypotheses (as explained above). Then the program pursues the highest ranked hypothesis. It gathers additional information from the patient and repeats the whole process of diagnosis again. This process is repeated several times until the system is satisfied with the results or can no longer proceed. The program then outputs the hypotheses after evaluating and scoring each.

- User Interface

PIP does not have any explanation facilities. The system does have an advisory program, which advises the supervisory program about the efficient methods for the exploration and organisation of the case at hand. This program is like a consistency checker. For example it lets the program know if the finding itself is clearly in error (a weight gain of 50lbs in 48 hours), or if the new information is inconsistent with the already existing ones. The system might also indicate that a new finding contradicts a conclusion already drawn about the problem. Finally, it may also check against the misinter-

pretation of the data by the user (for example: "dark urine" may be due to the presence of blood in the urine, the presence of bile or the presence of anthcyanins from beets).

4.7 BREASTCAN

BREASTCAN [13] is a KB system designed to assist physicians giving post operative adjuvant chemotherapy for breast cancer.

BREASTCAN has been proven capable to closely reproduce the actual decision taken over the period of 6 months by medical oncologists at the National Cancer Institute of Milan, Italy, in the treatment of 14 patients.

The components and working of the system are given below.

• Architecture of BREASTCAN

Though a layered model was not designed for BREASTCAN, a model with the following levels can be identified and described.

1. Domain level:

The domain level objects are arranged in the form of a tree. A part of this tree is shown in figure 4.9. The objects become more and more

Figure 4.9: *The domain knowledge in BREASTCAN.*

specific as one goes down the tree. In the figure 'primary treatment' is an object and the specific treatments are 'T1', 'T2' and 'T3'.

The objects of the tree are represented as frames. The frames contain information in the form of rules and steps. Steps are procedural knowledge or statements executed sequentially. An example of a frame (written in PROLOG) for the 'primary treatment' is shown in figure 4.10. The frame has 4 steps. It has a few frame conditions like asking the date, checking the date, updating the time etc. The frame contains four subframes as shown in the figure. And these subframes contain rules that relate to more specific conditions than frames. A step is next shown, which calculates the delay in treatments. Lastly the frame has a rule, which states that "the suggested dose of CTX (cyclophosphamide) is 300mg/m if the platelet count is < 100000 , leukocyte count is < 3800 and 2 consecutive delays have already occurred in drug administration".

All rules and frames are implemented in PROLOG.

2. Inference level:

An explicit inference structure with metaclasses and knowledge sources is not stated in this system. The steps of inference are as follows. Data and facts are collected from the patient. They are applied to the domain knowledge to deduce hypotheses. The hypotheses are refined to get the final treatment.

The type of inference is deductive. The hypotheses are deductively inferred from the domain knowledge, by triggering the appropriate rules and associations. There is no special program to deduce hypotheses. This is because PROLOG the language used provides a built in deductive inference capability.

3. Task level:

At the task level, BREASTCAN has a program for the process of diagnosis. The program collects the patient data. It applies the data to the domain knowledge to deduce hypotheses. The program then refines the initial list of hypotheses, by gathering more information from the patient. The system stops when it can no longer proceed, and outputs the treatment recommendations.

```

/**** Frame Descriptor ****/

frame (3,primary_treatment).
n_steps (3,4).
possibility (3,10).

frame_condition (3) :-
    ask_date (3, Date),
    date_test (Date, Date1),
    f(days (Date1, N_days)),
    (start_frame (3, Date) -> update_time (3, N_days)).

/**** Items ****/

n_days(3,time(_,_),N_days,[unknown,0,0]).
n_cycle(3,time(_,N_days),N_cycle,[unknown,0,0]).
n_cycle-cmf(3,time(N_cycle,N_days),N_cmf,[unknown,0,0]).
n_cycle-adm(3,time(N_cycle,N_days),N_adm,[unknown,0,0]).

/**** Subframes ****/

subframe(3,4).
subframe(3,5).
subframe(3,6).
subframe(3,7).

/**** Steps ****/

step(3,1,examination,uncompleted).
step(3,1):-
    heading(3).
    find_time(3,time(N_cycle,N_days)).
    fill_in(examination_date,2,time(N_cycle,N_days),Flag).
    ask_patient_status(3,time(N_cycle,N_days)),
    fill_in(delay,2,time(N_cycle,N_days),Flag),
    (last_cycle(3) ->> step_complete(3,1,completed)).

/**** Rules ****/

rule(1,3,delay,2,time(N_cycle,N_days),7,Truth,10) :-
    f(look_for(platelets,2,time(N_cycle,N_days),Platelets,Truth1)),
    f(look_for(leukocytes,2,time(N_cycle,N_days),Leukocytes,Truth2)),
    (Platelets < 100000; Leukocytes < 3800),
    f(look_for(n_delays,2,time(N_cycle,_),N_delays,10)),
    N_delays < 2,
    min(Truth1,Truth2,Truth).

```

Figure 4.10: *The primary treatment frame in BREASTCAN.*

Figure 4.11: The flowcharts for the task structure in BREASTCAN.

The above program is implemented in PROLOG. The program flowcharts are shown in figure 4.11(a) and (b). Figure 4.11(a) describes the main process of diagnosis. This shows how a frame with the highest priority number is made the current frame and executed, until no more frames remain for execution. Figure 4.11(b) describes the control procedure to execute frames. As shown in the figure, steps are executed sequentially in a frame, until no more steps exist when the frame becomes deactivated. If steps cannot be executed, due to the lack of data, frames are put back in the queue for future execution when the data becomes available.

- User Interface

The system user interface is quite simple. Like most other systems, it can explain how it deduced a goal, give reasons for its conclusions and also give advice about a particular case.

- KB Implementation Environment

BREASTCAN is written in PROLOG. It runs on VAX 11/750. In the long run, researchers have decided to set up a KB system (NEWCHEM), with a knowledge acquisition tool, which studies all possible drug interactions and includes its results into BREASTCAN, to help achieve better diagnostic results.

4.8 CASNET

CASNET [17,34] is a medical knowledge base system, based on the causal associational network models of a disease. It is used in the diagnosis and treatment of glaucoma.

The system was developed by a group of computer scientists at the Rutgers University, led by professor Casimir Kulikowski. A detailed description of its structure and working is given below.

- Architecture of CASNET

Though a layered model was not designed for CASNET, a model with the following levels of knowledge can be identified and described.

1. Domain level:

At the domain level, the disease process is described as a 3 layer model as shown figure 4.12. At the lowest layer there are observations such as signs, symptoms, tests etc. For example, 'visual acuity' is a test in the figure. At the next layer, pathophysiological states or the intermediate diagnoses are described. 'Elevated intraocular' pressure is an example. And finally at the last layer, there are disease categories. 'Open angle glaucoma' and 'secondary glaucoma' are the examples in figure. The most important thing to be noted is that, this type of modelling at the domain level is possible for well established disease processes only.

Figure 4.12: The three layer domain model in CASNET.

The knowledge representation scheme at this level is as follows. Observations are connected to the pathophysiological states by associational links, and the states are connected to the disease categories through classification links. States are connected to one another via causal links. Each link has a strength associated with it.

The various links of the network are implemented as rules.

2. Inference level:

An inference structure with metaclasses and knowledge sources is not explicitly stated in the system. But the steps of inference are as follows. Tests are performed on the patient, and the results are abstracted to get the patient data. The data is applied to the domain knowledge to deduce an initial list of hypotheses. This initial list of hypotheses is pursued to get a complete list of hypotheses. The complete list of hypotheses is classified as a disease category and treatments are recommended.

The type of inference is deductive. The inference is implemented as a pattern matching and a rule chaining program. A list of hypotheses is deduced by the programs as follows. Patient data are matched to the observations at the domain level. The matched observations are associated with states by invoking rules. Both these programs are implemented in FORTRAN.

3. Task level:

At this level CASNET has a consultation program. The task of diagnosis is as follows. The program gets the test results from the patient. The results are used to deduce an initial list of hypotheses (or states) by the program. The list of hypotheses is refined to get a complete list of hypotheses. The process of refinement includes prescribing some more tests and deducing more hypotheses. This process is repeated until a list of hypotheses which can account for all the patient manifestations is deduced. This final list of hypotheses is classified as a disease category. And the disease category and its treatments are output.

The program is implemented in FORTRAN.

- User Interface

CASNET is able to explain its diagnostic reasoning. It can explain why a given state was confirmed or denied (evidence for states). It can explain why a given disease category was selected. It can also explain how conflicts of evidence were resolved.

- KB Implementation Environment

CASNET/glaucoma is running in 35K words of memory on a DEC 10 or 20 computer under either the TOPS-20 or the TENEX operating system.

Because of the speed and the efficiency considerations, it is written in FORTRAN. The editing program is written in SNOBOL.

The knowledge base of CASNET is automatically updated by a group of ophthalmological researchers who can access the system through the Ophthalmological NETWORK (ONET).

4.9 CADIAG-2

CADIAG-2[1] is a data driven fuzzy medical KB system, with application in four different areas: rheumatic diseases, pancreatic diseases, gall bladder and bile duct diseases and colon diseases.

CADIAG-2 is a successor of CADIAG-1, whose development started in 1968. The KB system operates in conjunction with a database WAMIS (Vienna General Medical Information System). It was developed at the Vienna General Hospital, Vienna, Austria.

The system architecture and working are described below.

- Architecture of CADIAG-2.

Though a layered model for the expertise was not designed by its researchers, a model with the following levels of knowledge can be identified and described

2

of/de

2

MicroD

IF Δ suspicion of liver metastases by liver palpitation
 THEN MAY BE pancreatic cancer
 WITH ($\lambda_o = \text{medium} [\mu_o = 0.55]$, $\lambda_c = \text{weak} [\mu_c = 0.30]$).

Figure 4.13: An example of a rule from CADIAG-2.

in CADIAG-2.

1. Domain level:

At the domain level, the knowledge is in the form of domain specific rules. An example of a rule is shown in figure 4.13. The rule states that "if there is a suspicion of liver metastases by liver palpitation then it may be pancreatic cancer, with a medium frequency of occurrence and a weak strength of confirmation".

The rule scheme used for the knowledge representation in CADIAG-2 has a few limitations. All rules have a frequency of occurrence and a strength of confirmation, with linguistic (λ) and numerical (μ) values. But these values are seldom point values in real life, as they change over time and are influenced by external events. Most of the rules are single antecedent/single consequent rules, which prevents the incorporation of complex medical information.

2. Inference level:

An explicit inference structure with metaclasses and knowledge sources is not described. But, for reasons of efficiency, CADIAG-2 processes a fixed number of inference steps, as follows. Patient data is collected from the WAMIS database. The data is abstracted to get evidence. The evidence is applied to the domain knowledge to deduce hypotheses. The hypotheses are pursued and refined. And a final diagnostic report is output.

The type of inference is deductive. The hypotheses are deductively inferred from the domain knowledge by triggering the appropriate associations and rules. The inference is implemented as a backward rule chaining program.

3. Task level:

The task level consists of the following diagnostic programs, which control the entire process of diagnosis. They are shown in figure 4.14. Initially, the automatic screening procedure gets the patient's personal and medical information from the WAMIS database. The data is abstracted by the patient fuzzy data interpreter and is transferred to CADIAG-2. An example for an abstracted symptom is "elevated glucose serum level", which is set according to the quantitative result of the glucose

Figure 4.14: *The task structure in CADIAG-2.*

test and the definition of "elevated". The system applies the abstracted data to the relevant rules, and deduces hypotheses. The system then outputs an initial diagnostic report list, with further examination proposals. After this, the on line consultation program, refines the initial list of hypotheses. This program is a natural language man-machine interface. It allows more patient information to be entered into the system and the system deduces more hypotheses and refines the existing hypotheses. A diagnostic report with further examination proposals are again output. The above procedure can be repeated several times, until the user is satisfied.

The programs at the task level are written in CICS/VS command level language and PL/1.

- User Interface

Following are some of the user interface capabilities offered by CADIAG-2. The system presents unexplained abnormal findings at the end of the consultation. In an explanation, it presents every item of evidence that contributed to the respective diagnostic result. During the on line consultation, the system generates further useful examinations at the end of each iteration cycle.

- KB Implementation Environment

CADIAG-2 is embedded in the time sharing environment of the medical

information system WAMIS, which runs on an IBM 4381 Model P03 under DOS/VS, controlled by VM. At present, a reimplementaion on the personal computer is in progress.

A semi automatic acquisition of medical knowledge was accomplished for CADIAG-2. The results of which were not only used to acquire new knowledge by learning from new examples, but also to verify the existing knowledge and check stored patient records for documentation errors.

4.10 INTERNIST

The KB system INTERNIST-1 [17,27] (or INTERNIST for short) is designed to make multiplex and complex diagnoses in internal medicine.

The knowledge base was developed for several years by Myers and medical student assistants. The system was constructed at the University of Pittsburgh during the 1970s. The system components and working are described below.

- Architecture of INTERNIST

Though a layered model was not designed for INTERNIST, a model with the following levels can be identified and described.

1. Domain level:

At the domain level, there are just two concepts, diseases and findings. The individual diseases in INTERNIST are a part of a disease hierarchy that is organised from the general to the specific. For example, acute viral hepatitis is classified as a hepatocellular infection, which is a subclass of diffuse hepatic parenchymal disease and so on.

Findings are historical items, symptoms, physical signs and laboratory abnormalities. Three clinical variables are associated with each of the findings. They are the evoking strength (how strong does the finding explain the disease), frequency (how often the patients with the disease have this finding) and import (global importance of the finding). The knowledge base has causal links between findings and the diseases.

A part of the list of findings are given in figure 4.15. For example, "proteinuria" is a finding with an evoking strength of 1 and a frequency of 2. The causal links between the diseases and the findings are represented as rules.

The domain knowledge is implemented in INTERLISP.

2. Inference level:

An inference structure with metaclasses and knowledge sources is not stated explicitly in this system. But following are the steps of infer-

DISPLAY WHICH MANIFESTATION LIST?
ALCOHOLIC HEPATITIS

AGE 16 TO 25 0 1
AGE 26 TO 55 0 3
AGE GTR THAN 55 0 2
ALCOHOL INGESTION RECENT HX 2 4
ALCOHOLISM CHRONIC HX 2 4
SEX FEMALE 0 2
SEX MALE 0 4
URINE DARK HX 1 3
WEIGHT LOSS GTR THAN 10 PERCENT 0 3
ABDOMEN PAIN ACUTE 1 2
ABDOMEN PAIN COLICKY 1 1
ABDOMEN PAIN EPIGASTRIUM 1 2
ABDOMEN PAIN NON-COLICKY 1 2
ABDOMEN PAIN RIGHT UPPER QUADRANT 1 3
ANOREXIA 0 4
DIARRHEA ACUTE 1 2
MYALGIA 0 3
VOMITING RECENT 0 4
ABDOMEN BRUIT CONTINUOUS RIGHT UPPER QUADRANT 1 2
ABDOMEN TENDERNESS RIGHT UPPER QUADRANT 2 4
CONJUNCTIVA AND OR MOUTH PALLOR 1 2
FECES LIGHT COLORED 1 2
FEVER 0 4
HAND(S) DUPUYTRÉNS CONTRACTURE(S) 1 2
JAUNDICE 1 3
LEG(S) EDEMA BILATERAL SLIGHT OR MODERATE 1 2
LIVER ENLARGED MASSIVE 1 2
LIVER ENLARGED MODERATE 1 3
LIVER ENLARGED SLIGHT 1 2
PAROTID GLAND(S) ENLARGED 1 2
SKIN PALLOR GENERALIZED 0 2
SKIN PALMAR ERYTHEMA 1 3
SKIN SPIDER ANGIOMATA 2 3
SKIN TELANGIECTASIA 1 1
ALKALINE PHOSPHATASE BLOOD GTR THAN 2 TIMES NORMAL 1 2
ALKALINE PHOSPHATASE BLOOD INCREASED NOT OVER 2 TIME NORMAL 1 4
BILIRUBIN BLOOD DECREASED 2 2
BILIRUBIN URINE PRESENT 2 4
CHOLESTEROL BLOOD DECREASED 2 2
CHOLESTEROL BLOOD INCREASED 1 2
HEMATOCRIT BLOOD LESS THAN 35 1 3
HEMOGLOBIN BLOOD LESS THAN 12 1 3
KETONURIA 1 2
PROTEINURIA 1 2
SGOT 120 TO 400 2 3
SGOT 40 TO 119 2 3
SGOT GTR THAN 400 1 2
UREA NITROGEN BLOOD LESS THAN 8 2 2
UROBILINOGEN URINE ABSENT 1 1
UROBILINOGEN URINE INCREASED 2 4

Figure 4.15: A sample manifestations' list from *INTERNIST*.

ence. The patient data is gathered from the patient. It is applied to the domain knowledge to deduce hypotheses. The hypotheses are pursued and refined. A final diagnosis is output.

The type of inference employed is deductive. It is implemented as a pattern matching program in INTERLISP. The program works as follows. Each of the positive and negative findings found in a patient are matched to the findings in the list of diagnoses in the domain level. The diagnoses whose findings match, are weighed, using the evoking strengths and frequencies. This gives a list of possible diseases in the patient.

3. Task level:

The task of diagnosis is as follows. The system gets the patient data. It applies this data to the domain knowledge and deduces hypotheses (as explained above). The hypotheses are pursued and refined. This is done by acquiring more information from the patient and repeating the entire procedure above. When the results are satisfactory and a final diagnosis is made, the system stops. The results are output.

The above program is implemented in INTERLISP.

- User Interface

INTERNIST's handling of explanations is very shallow. Its only source of

explanation is that it can give a list of findings supporting the final diagnosis.

- KB Implementation Environment

INTERNIST is implemented on PDP-10 under the TENEX operating system. It takes about 800 MBytes. The primary implementation language is INTERLISP, with interfaces to the knowledge base coded in assembly language. It is estimated that its current KB includes roughly 70 - 75% of the major diagnoses of internal medicine.

Some limitations in representation and control methods of INTERNIST have been the basis for the next enhanced version CADUCEUS.

4.11 MDX

MDX[16] is a knowledge base system developed for the diagnosis of the liver disease.

The system was developed at the Ohio University during the mid 1970s. Its principle researchers were Prof. B.Chandrasekaran and Fernando Gomez. The system centers its representation around concepts in a hierarchy. But the main emphasis is on the distribution of the expert's diagnostic knowledge throughout this hierarchy. The system's architecture and working are described in detail be-

low.

- Architecture of MDX

The researchers of MDX have not paid much importance to separation of knowledge into several levels. But a model with the following levels can be identified and described in MDX.

1. Domain level:

The domain knowledge consists of objects arranged in the form of a tree. The objects are called concepts (or specialists) and they form a hierarchical structure similar to a botanical or a zoological classification system. But these concepts are different from the basic idea, as they are not just what-type structures but are used as labels to organise how-type knowledge.

The figure 4.16 shows the hierarchy of concepts. The concepts are arranged from the most general at the top to the most specific at the bottom. For example, in the figure, "cholestasis" is the chief concept. "Extrahepatic cholestasis" and "intrahepatic cholestasis" are its sub-concepts and so on. The domain of MDX is not so large.

Each concept is represented as a collection of rules. There are three types of rules grouped under each concept - confirmatory, exclusionary

Figure 4.16: A conceptual structure of "cholestasis" in MDX.

and recommendation rules. Confirmatory rules are used to establish a concept or a hypothesis. Some of the competing concepts are ruled out using the exclusionary rules. And recommendation rules are pieces of knowledge that anticipate a concept's subconcept.

2. Inference level:

An inference structure with metaclasses and knowledge sources is not explicitly stated in this system. The type of inference is deductive. The hypotheses are deduced by triggering the domain rules in the concepts. The inference is implemented as a rule chainer program.

3. Task level:

The task of diagnosis is not represented as a goal tree or a hierarchical structure, but the procedure involved in the diagnosis is roughly as follows.

- The top level concept ("cholestasis") first tries to establish itself (using confirmatory rules). If it succeeds, it refines itself to account for all the patient manifestations. It then invokes its subconcepts, using a priority scheme, to account for the unexplained manifestations. These subconcepts try to establish and refine themselves and in turn invoke their subconcepts and so on. Finally, when the established concepts can explain all the patient manifestations in a nonoverlapping manner,

the case is solved.

The system was built as a small prototype, and hence it is quite simple in its design and implementation. Work is underway in extending its features, such as the introduction of blackboard, parallel as opposed to the existing serial implementation etc. [16].

- User interface

The system does not have any explanatory or consistency checking facilities.

It can only give a list of findings supporting all established hypotheses.

Some of the existing medical knowledge base systems have been reviewed so far. The following chapters describe the design of the FRD Knowledge base system and show clearly the differences between the FRD KB system and the other systems described above.

Chapter 5

Overview of the KEE

5.1 Introduction

Current trends in knowledge based computing have produced a large number of expert system building tools. The complexity of an expert system building tool can range from simple interpreters of nonprocedural programming languages (e.g. LISP, PROLOG interpreters) to very elaborate and sophisticated shells (e.g. KEE, ART). These shells provide at least, the minimum set of features, which are a knowledge representation scheme, an inference or a search mechanism, a means of describing a problem and the way to determine the status of a problem while it is being solved.

Currently a large number of expert system building tools are available in the market. They provide rapid prototyping and accelerate the development of the KB systems. Knowledge Engineering Environment(KEE) is one such expert sys-

tem building tool. KEE which was used to implement the prototype for the FRD KB system developed in this thesis is described in detail in this chapter.

The next section describes knowledge representation schemes in KEE. Section 5.3 describes reasoning in KEE. The methods in KEE are described in section 5.4. Section 5.5 describes the active values and the active images used in KEE. And finally, in section 5.6 the different versions of KEE, and the evaluation of KEE with respect to the other existing expert system building tools are given.

5.2 Knowledge Representation in KEE

The scheme of knowledge representation used in KEE is frames. Frames in KEE are called "units". Units are classified into 3 types. They are : class, subclass and member. A set of closely related units is called a "class". A subgroup of a class is called a "subclass". And a specific example of a class or a subclass is called a "member". The related units can be connected to form a hierarchy, and an example is shown in figure 5.1. In the figure, 'cars' is a class, 'smallcars' is a subclass of cars and 'my.chevette' is a member of smallcars. Note that the class/subclass - member relationships are represented by broken lines and the class - subclass relationships are represented by the solid lines in the figure.

A unit has many "slots" in it. And the slots contain the information pertaining to the unit they are in. There are two types of slots in KEE. They are - mem-

Figure 5.1: A hierarchy of units in KEE.

ber and own slots. Figure 5.2 shows the KEE display of the unit car in which 'battery', 'lights' etc. are slots.

Slots are described by "facets". Facets are attributes that describe a slot. The value facets of the slots in the unit car can be seen in figure 5.2. The value facet contains the values of a slot. For example, the value of the slot 'lights' is 'on' in the figure. Following are some of the other facets of a slot. The valueclass facet describes the set of values a slot can take. The maximum and minimum values a slot can have is specified by the maximum and minimum cardinality attributes of the slot.

KEE uses the mechanism of hierarchy to control the slot inheritance and this is accomplished as follows. The type of the slot (member or own) in the parent unit and the type of relationship between the parent and the child units (class or member) are used to control the type of slot inherited by the child unit from the parent unit. Member slots in parent units become own slots in child units across member links, but they remain member slots across class/subclass links. Own

Unit: Cars		
Member	Battery	
Slot:	Value	Unknown
Member	Fuel Tank	
Slot:	Value	Unknown
Member	Lights	
Slot:	Value	ON
Member	Condition	
Slot:	Value	Unknown
Own	Num.Bad.Batteries	
Slot:	Value	Unknown
Own	Num.Needing.Gas	
Slot:	Value	Unknown

Figure 5.2: An example of a unit in KEE.

slots cannot be inherited by the child unit from the parent unit. The actual values inherited by the child unit's slot depend upon the child's inheritance attribute and the parent's actual values. The child unit's slot may inherit values from one of its parents (single inheritance), it may inherit from all its parents (multiple inheritance) or from none of its parents[19].

KEE displays a number of menus and submenus for the various tasks, such as creating a KB, loading, saving or displaying a KB, creating units and slots, changing values and attributes, creating links between units and many more. Any procedure can be selected using mouse clicks on objects. Moreover, all the above procedures can also be performed programmatically in LISP[38].

5.3 Reasoning in KEE

There are four types of control structures in KEE, which are used for the process of reasoning. They are: rules, methods, active values and active images. Only the first is described in this section, the rest are described in the following sections.

Rules in KEE are stored in special rule units. The rule units are made members of a common ruleclass. Each rule has antecedents and consequents. Examples of a few rules in KEE are shown in figure 5.3. Every rule has one or more conditions and one or more conclusions. The first rule states that "if the car lights are on, then the battery of the car is alright".

IF Condition THEN Conclusion

IF the Car Lights are On THEN the Battery is OK

IF Condition1 AND Condition2 AND Condition3

THEN Conclusion

**IF the Car Won't Start AND the Battery is OK
AND the Fuel Tank is Not Empty**

THEN There is a Starter Problem

Figure 5.3: *A few rules in KEE.*

KEE uses rules for reasoning. The rule based reasoning can be directed in a forward (from condition to conclusion) or a backward (from conclusion to condition) manner. The chaining of rules can be directed in either forward or backward manner and most often a combination of the two.

KEE has a simple knowledge representation language called "Tell-And- Ask", with english like syntax[20]. There are three basic functions in this language, "Assert", "Retract" and "Query". Just as menus can be used to create, represent or change a KB, Tell-And-Ask can also be used for performing these functions. Besides this, Tell-And-Ask language can also be used to invoke the process of

reasoning. The function "Assert" asserts a fact and in the process starts forward chaining of rules. The function "Query" tries to check if a particular condition is true and in the process initiates the backward chaining of the rules. When the KB is changed as a result of an assert or a query, the function "Retract" can be used to remove the changes made to the KB by the last assert or query. If a certain information is not available to the system, it can be programmed (using the "Ask User" function) to ask the user the required information during the reasoning process. Rules can be created, modified or triggered by the use of menus and submenus.

Explanation facilities provide the system the capability to give explanations regarding 'how it deduced a goal', 'why it needs certain information from the user' etc. During the process of reasoning or explanation, rules can be traced. The tracing is visible either in a textual or a graphic form, for example the AND/OR tree in figure 5.4. The intermediate result of backward chaining of rules is shown in this figure. It can be seen how the battery rule and the starter rule in figure 5.3 are chained. The rule which is currently being executed is in black. The other details regarding the use of rules and the reasoning with rules are available in the manual[19, 20].

Figure 5.4: *An intermediate result of rule tracing during the process of backward chaining.*

5.4 KEE Methods

LISP programs that live in slots are called methods in KEE. They represent the procedural knowledge and describe how to do things.

Method slots are similar to the ordinary slots, except that their value attribute now contains procedural knowledge (e.g. LISP code) instead of descriptive knowledge (e.g. real values, strings etc.). Methods can be used to trigger rules, make calculations, define actions or behaviors. One of the major advantages of KEE is that the method can be used to develop object oriented programming. Object oriented programming [9] is a modular style of programming. Methods are closely associated with their related objects and tell the object what actions to take. They can send messages to other objects, and thus trigger other methods in those objects. This is definitely one of the finer points in KEE.

Methods can be activated in two different ways, either by mousing the appropriate method slot or programmatically [19]. Methods can be passed on from the parent unit to the child unit, by using the appropriate type of inheritance. A typical method as it appears in a regular display and in an edit window are shown in figure 5.5. The method can be edited in the edit window. The 'programbody' in the code in the edit window is where the LISP code is inserted in. The value attribute of the method slot in the regular display shows the location of the method (i.e. the name of the KB, unit and slot where the method is found). The figure shows a method in the 'order.parts' slot of the 'orders' unit in the 'Fixcars' KB.

Method Display

Slot: ORDER.PARTS from ORDERS
Inheritance role: METHOD
Valueclass: METHOD
Value: FIXCARS>ORDERS:ORDER.PARTS!method

Regular Display

(ORDER.PARTS (VALUE (LAMBDA (SELF) PROGRAMBODY))
(INHERITANCE METHOD)
(VALUECLASS (#Unit (METHOD KEEDATATYPES)))
(DEFAULT NIL))

Editor Display

Figure 5.5: *The display of a method in KEE.*

The inheritance and the valueclass of the method slot are both of type 'method'. The value of the slot, in the edit display shows the actual LISP code. Methods can be edited, triggered or created programmatically or by the use of menus and submenus[19,20].

5.5 Active Values and Active Images

Active values are system daemons found in units called the active value units, and are used to activate a user method found in one of their slots. The active value units are attached to a slot in one of the units in the user KB. When the value of the slot to which the active value unit is attached is referenced or changed, the user method in the active value is triggered. The user can use this facility for object oriented programming and also to write his/her own inference.

Active values are units which are made members of the unit ACTIVEVALUE in the "Activevalues" knowledge base. A method is placed in one of its slots. The type of the slot in which the user method is placed, determines when the user method is activated[19]. An active value unit is displayed in figure 5.6. It is called 'fuel.active.value'. It is made a member of the 'active value' unit in the Active Values KB and is attached to the 'fuel.tank' slot of the 'car' unit in the user KB. A method is put in its 'avput' slot. And this method triggers whenever the value of the 'fuel.tank' slot changes.

Figure 5.6: An active value unit in KEE.

Active images[21] in KEE knowledge base contain graphic images. They are extensions of active values.

Active images are used to display slot values, monitor values as they change, change a value interactively, invoke a method, show relationships between different objects and see how values change relative to each other.

A slot's valueclass determines the type of image that can be attached to it. The following are some of the different types of active images. They are shown in figure 5.7. And they are attached to the slots in '816HZK' car unit in 'Fixcars' KB. The different images are - general, numeric, alarms, text, method, state etc.[21]. The general images are attachable to any slot regardless of the valueclass. They are used to display or change the values of the slots, for example to display fuel status of a car. Numeric images are level indicators, horizontal or vertical bar graphs and thermometers which are used to display or actuate numerical values in slots. The figure shows bar graphs to display the car's fuel tank value and a thermometer to display the car's temperature. Alarms are images which start flashing when the condition shown on the image becomes true. The figure shows an alarm which starts flashing whenever the fuel tank becomes empty. Alarms can be numeric (with numbers in them), state (with text) etc. Text images are used to display the textual information. State images are used to display and input multiple values. A vertical push button type of state image is shown in the figure. It describes the color of the car. And finally, images attached to the slots that contain methods in them, can be used to display or actuate the method. A method

General Images

Numeric Images

Thermometer

Sideways.Bargraph

State Images

Vertical.Push.Buttons

State.Alarm

Text Images

Method Images

Figure 5.7: A few active images in KEE.

actuator for the method in the 'fuel.tank' slot of the car unit is shown in the figure.

Images can be modified as desired. Their title, height, width, border, range, increment, fluid shade etc. can be changed if required. Image panels are used to group related images. They can be attached to a KB or a unit.

The general features of KEE were described in the previous sections. The following section describes the different versions of KEE and compares it with the other existing expert system tools.

5.6 Other Extended Features of KEE

KEE version 2.0 was the first release of KEE system software available through Intellicorp. Certain changes and enhancements to the software resulted in the release level 2.1[22]. Both these versions were written in Zeta Lisp.

The latest version of the KEE system software is the release level 3.0[19,20,21]. It has a number of added features and is written in Common Lisp. A new powerful object oriented graphics toolkit called KEE pictures has been incorporated. The implementation of activeimages in KEE 3.0 software is more efficient. Many images can be placed within one image panel, and this requires just one window, as compared to a window for every image in level 2.1. Hypothetical worlds[20] or problem states can be modelled using the KEE worlds facility. The Truth Main-

tenance System (TMS)[20] tracks beliefs and their presence in various worlds. Multiple desktops allow the user to build his/her own screen by selecting the windows the user wants. And lastly it is possible to switch back and forth between the forward and backward chainers.

The KEE software version 3.0 has a few bugs in it. Some of these bugs are eliminated in the recent release of KEE software system level 3.1. The Carleton University has not received this release so far.

Gilmore et al[15] have listed a set of characteristics to be present in an ideal expert system tool and describe KEE as one which does not differ much from the ideal expert system tool. Figure 5.8 gives an evaluation of large scale expert system tools, which compares KEE with the other existing tools. Since KEE is totally object oriented, it tends to be slow. KEE, besides ART (another large scale expert system tool), is the most expensive tool available. KEE's activeimages and 2D bit-mapped graphics (which provide an icon editor) are its outstanding features. They help the user start or stop a process whenever or wherever he/she likes and provide application in discrete event simulation and process management.

The following chapters describe the design and implementation of the FRD • KB system using the KEE software system level 3.0.

Large Scale EST	KES	ART	KEE	LOOPS	KC	DUCK	GEST
Structured Object-Oriented	X	X	X	X	X	X	X
Non-monotonic		X	X		X		
Interaction	F	F	M	M	M	F	F
Line/Menu screen	L	M	M	M	M	L	M
Tool language	C	L	L	L	L	P,L	L
Hosts							
• Symbolics		X	X				X
• LMI		X	X				
• VAX	X	X	X			X	
• TI		X	X				
• Xerox		X	X	X			
• Other	X					X	
Rule types							
• Inference		X	X	X	X	X	X
• Production	X	X	X	X	X	X	X
• Hypothesis	X	X			X		X
• Bayesian	X						
• Constraint		X					X
• Belief	X	X					
Rule firing							
• Forward	X	X	X			X	X
• Backward	X	X	X		X	X	X
• Both		X	X			X	
Conflict Resolution							
• First-found	X	X	X	X	X	X	
• Least recent							X
• Most recent							X
• Antecedent	X						X
• Consequent		X					X
• Most complex							X
• Least complex							X
• Global							X
• User defined		X	X	X	X		X
Cost	25K	80K	60K	\$300	50K	6K	9.5K

Interaction -- F = Fast; M = Medium
 Line/Menu Screen --- L = Line; M = Menu
 Tool Language --- C = C; L = LISP; P = Prolog.

Figure 5.8: Large Scale Expert System Tool Evaluation.

Chapter 6

Design and Implementation of the FRD KB System

6.1 Introduction

The FRD KB system developed in this thesis, is an advisory system to help the physician in the diagnosis of the Familial Retinal Degeneration diseases. But most important of all, the system can be used as a research tool in medical discovery, knowledge acquisition, tutoring etc. In developing the knowledge base system, the first phase was that of acquiring knowledge. Next, a model for this knowledge was designed. And finally, this model was implemented as an initial prototype, using the KEE expert system tool.

The process of knowledge acquisition for the FRD knowledge base is given in section 6.2. In section 6.3, the FRD KB system architecture is described, which

includes the detailed description of the model designed for the expertise. The FRD model has 3 levels: the domain, inference and the task levels. And this section describes in detail, the design, and implementation of each level. Section 6.4 describes the user interface for the FRD KB system. And finally, the FRD KB implementation environment is given in section 6.5.

6.2 Knowledge Acquisition

The main goal of the FRD knowledge base system is to help diagnose and classify a patient with Familial Retinal Degeneration diseases. Two sources of knowledge were employed in order to build this system. The first one is the knowledge from the literature, data in the databases etc. The second is the knowledge elicited from the expert(s).

Initially, I read a number of published papers and books in the area of neuro-ophthalmology and FRD diseases. This helped me acquire some general knowledge about the FRD diseases, their characteristics and the tests employed to diagnose the FRD diseases. These were described in detail in chapter 2. The next step was to acquire some knowledge from the existing raw data. The Ottawa Civic Hospital has a large database containing the raw data of patients with FRD diseases. The raw data includes the test results of a patient (such as laboratory tests, history, signs, symptoms, physical examinations etc.), his/her intermediate diagnosis and the final diagnosis. This data was used to study the parameters and attributes of

a test result, and some typical patterns or characteristics of patients with FRD diseases. But the knowledge acquired from the literature and the database as explained above, does not tell us anything about the inference and procedures of diagnosis employed by the physician in the process. The source of such knowledge was the expert.

It is desirable that the knowledge should be elicited from a group of experts. However, only one expert from the Ottawa Civic Hospital was available in our case. Knowledge acquisition from the expert is more difficult than from the literature, due to the following reasons. It is a very tedious process from the point of the physician, as he/she finds it very hard to express his/her years of experience in a simple manner. Furthermore, to extract this knowledge from the expert, which is the knowledge engineer's job, is equally tough as the knowledge engineer most often is a novice in the expert's domain. Moreover, neuro-ophthalmology (and FRD diseases) is not a well defined area and the knowledge in literature on the same is not so abundant. However, through the thoughtful cooperation of the expert and his assistants, sufficient knowledge to build an initial FRD prototype was acquired. The knowledge was summarised in a three dimensional table, which is given in appendix A. This table, is a starting point and is the result of months of work put in by the expert and his assistants. Moreover, the table describes the expert's opinion and other experts in the same area may not necessarily fully agree with it. This table describes the various tests, diseases and their relationships. The significance of each test in diagnosing a disease (i.e. the 'weights' or the test-significance factor) and the interpretation of test results are given. For

example, the test 'media' can result in any one of the following values - nil, identifiable, moderate or marked - representing an increasing order of severity. The significance of the test in diagnosing CSNB is 4 on a scale from 0-5. This table describes just one level of tests and diagnoses. This can be broken down to deeper levels of abstraction. For example, some of the test values can be intermediate diagnoses. Test results which are usually numeric, can be used to determine these intermediate diagnoses. The intermediate diagnoses can cause or effect other intermediate diagnoses. And a set of intermediate diagnoses can result in a disease. I also elicited from the expert, how he performs diagnosis. Then I presented a model of reasoning which seemed satisfactory to him.

Once the knowledge is acquired, the next stage is to design a model for the knowledge. This model should reflect clearly the different types of knowledge. The following section describes the FRD KB system model in detail.

6.3 Architecture of the FRD KB System

The model for the FRD KB system is based on the layered model of expertise developed by Wielinga et al[36]. The different types of knowledge are separated into several levels in the model. The various levels of the model are: the domain, inference and task levels. The following describes the design and implementation of the levels in detail.

6.3.1 Domain Level

- Design:

The domain level consists of the domain knowledge, which is made up of domain specific concepts and relationships. The domain concepts are arranged as a 2 layer model. The higher layer consists of the disease concept and the lower layer consists of the tests concept.

The domain layer is shown in figure 6.1. As shown in the figure, the concepts at each layer of the 2 layer domain model are arranged in the form of a hierarchy. The first layer consists of the main concept "diseases". "FRD-diseases" is a subconcept of this concept. And the different types of FRD-diseases are "CSNB", "CR-Dyst", "RC-Dyst" and "Pseudo-RP", arranged as subconcepts under the "FRD-diseases" concept. The lower layer of the model consists of the concept "tests". The subconcept of tests are the three concepts "history", "clinical" and "lab-tests". Individual tests such as "acuity", "media" etc. are the subconcepts of these concepts.

The area of FRD diseases is such that, each and every test contributes in one way or another to the process of diagnosis. Depending upon the test significance factor, the test may affect the diagnosis of disease in a strong (for a large significant factor) or a weak (for a small significant factor) manner. This is because the FRD diseases have very similar characteristics and are very difficult to distinguish from each other. Hence every test is related to

It must be noted that all the tests are associated with all the diseases. These associations being too many will make the figure look complicated. Therefore, they are not shown above.

Figure 6.1: The domain level in the FRD KB system.

every disease by the domain specific associational relationships.

This type of modelling at the domain level, is not found in most existing knowledge base systems. CASNET[34] has a similar but a 3 layer domain model. But the concepts at the individual levels are not arranged hierarchically. NEOMYCIN[8] and MDX[16] have a hierarchical domain model, with all the concepts arranged in a single level. But the concepts in MDX contain pieces of inferential knowledge. The above model is suitable for the FRD KB system, because the domain knowledge permits such modelling. The model is represented as an extension of a causal net.

- Implementation:

Every concept at the domain level is implemented as a unit in KEE. An object oriented approach (as opposed to the rule based approach) has been adopted to implement the FRD KB system. This is primarily due to the fact that it offers more flexibility at the inference level. Furthermore, the object oriented approach is more suitable than the latter approach and naturally represents domain knowledge. The two layers as implemented in KEE are shown at the bottom of figure 6.2. In the figure, "diseases" is a unit which represents a class of diseases. "FRD-diseases" is a subclass of diseases and "CR-Dyst", "CSNB", "RC-Dyst" and "Pseudo-RP" are the four members or types of FRD diseases. The general slots in the disease units are shown in figure 6.3. These slots may be member or own slots in the other units in

Figure 6.2: The KEE output graph of the FRD knowledge base.

DISEASES Slots
member: NUMBER-OF-TESTS (local) PROBABILITY (local) PROBABILITY-OF-DISEASE (local)
own: DECOMPOSITION.COMPLETE DECOMPOSITION.DISJOINT MEMBERS.DATATYPE MEMBERSHIP SUBCLASSP

Figure 6.3: *The general slots of a disease unit.*

the disease hierarchy depending upon the type of link between the different units (see section 5.2). The following are the slots in the units.

- number-of-tests: This slot contains a numerical value which indicates the number of tests done on a patient.
- probability: This slot contains the total score (see section 6.3.2 for details) for the presence of a disease.
- probability-of-disease: The probability of the presence of the disease is contained in this slot. It is a number between 0 and 1.

III (Output) The CSNB Unit in the FRD Knowledge Base

Unit: **CSNB** in knowledge base **FRD**
Created by **GEETHA** on 3-24-86 12:13:25
Modified by **GEETHA** on 5-4-86 11:58:43
Member of: **FRD-DISEASES**

Own slot: **NUMBER-OF-TESTS** from **CSNB**
Inheritance: **OVERRIDE.VALUES**
Values: 2

Own slot: **PROBABILITY** from **CSNB**
Inheritance: **OVERRIDE.VALUES**
Values: 1.2

Own slot: **PROBABILITY-OF-DISEASE** from **CSNB**
Inheritance: **OVERRIDE.VALUES**
Activeimages3 **VIEWPORT-PROBABILITY-OF-DISEASE-OF-CSNB.4**
Values: 0.6

Figure 6.4: The "CSNB" unit.

For example, figure 6.4 shows the CSNB unit. The probability of the presence of CSNB in the patient after performing 2 tests is 0.6. The other slots in the diseases unit, shown in figure 6.3 are system slots[19].

At the next layer, in the domain level, "tests" is a unit which represents a class of tests employed to diagnose the FRD diseases. This is also shown in figure 6.2. "History", "clinical" and "Lab-tests" are the three subclasses of tests. "Central-progression", "nyctalopia" etc. are the members or types of history taking. "100-hue", "cone-erg" etc. are the different laboratory tests and "acuity", "media" etc. are the types of clinical examinations. The different slots of the tests unit are shown in figure 6.5. These slots in the other units of the tests hierarchy may be own or member slots depending upon the type of link between the different units of the hierarchy (see section 5.2).

The following are the slots in the units.

- actual-test-value: This slot contains the actual value or result of performing a test on a patient. The values can either be in the form of text or numbers. A value of 'unknown' indicates that the test was not performed on the patient by the physician.
- confidence-factor: The confidence in the result of a test is represented as confidence factor and is contained in this slot. It is a number between 0 and 1.

TESTS Slots

member:

ACTUAL-TEST-VALUE (local)

CONFIDENCE-FACTOR (local)

CR-WEIGHT (local)

CSNB-WEIGHT (local)

EQUIVALENT-NUMERICAL-VALUE (local)

PSEUDO-RP-WEIGHT (local)

RC-WEIGHT (local)

own:

DECOMPOSITION.COMPLETE

DECOMPOSITION.DISJOINT

DONE (local)

DONE1 (local)

HISTORY-TO-BE-DONE (local)

LAB-TEST-TO-BE-DONE (local)

MEMBERS.DATATYPE

MEMBERSHIP

PHYSICAL-TO-BE-DONE (local)

SUBCLASSP

Figure 6.5: *The general slots of a test unit.*

- cr-weight: This slot contains the significance of the test in diagnosing CR-Dyst. It is a number between 0 and 1.
- csnb-weight: This slot contains the significance of the test in diagnosing CSNB. It is a number between 0 and 1.
- rc-weight: This slot contains the significance of the test in diagnosing RC-Dyst. It is a number between 0 and 1.
- pseudo-rp-weight: This slot contains the significance of the test in diagnosing Pseudo-RP. It is a number between 0 and 1.
- equivalent-numerical-value: An equivalent numerical value that corresponds to the actual result of a test is contained in this slot. It is a number between 0 and 1; and the number represents the severity of the test result.
- history-to-be-done: This slot contains the last history test taken by the patient.
- physical-to-be-done: This slot contains the last physical examination done on the patient.

- lab-test-to-be-done: This slot contains the last laboratory test done on the patient.
- done and done1: These are slots which are used during the process of inference. They indicate whether certain procedures are completed (a value of "T") or not (a value of "nil").

For example, unit "central-progression" is shown in figure 6.6. It is a type of history taking. The result of performing this test is that progression is "stationary". The confidence in this result is 0.8. The significance of the test, central progression in diagnosing CR-Dyst is 0.4; CSNB is 0.6; RC-Dyst is 0.6 and Pseudo-RP is 0.8. The equivalent numerical value of the result "stationary" of the test is 0.25. The other slots shown in figure 6.5 are system slots[19].

Besides these, units are created dynamically and added to the knowledge base, as diagnosis proceeds. The units created dynamically are the units that represent a patient. These units are made members of the "patients" unit. The patient number becomes the name of the patient unit created dynamically. The general slots in a patient unit are shown in figure 6.7. The following are the slots in these units.

- name: This slot contains the name of the patient.

III (Output) The CENTRAL-PROGRESSION Unit in the FRD Knowledge base

Unit: **CENTRAL-PROGRESSION** in knowledge base **FRD**

Created by **GEETHA** on 3-24-86 12:20:45

Modified by **GEETHA** on 5-4-86 12:09:16

Member of: **HISTORY**

Own slot: **ACTUAL-TEST-VALUE** from **CENTRAL-PROGRESSION**

Inheritance: **OVERRIDE.VALUES**

ValueClass: (ONE.OF STATIONARY PROGRESSION-IN->OR=20YRS PROGRESSION-
IN-5-TO-20-YRS PROGRESSION-IN-2-TO-5-YRS UNKNOWN)

Activeimages3: **VIEWPORT-ACTUAL-TEST-VALUE-
OF-CENTRAL-PROGRESSION.5**

Avunits: (PROGRESSION.ACTIVEVALUE ALL NIL)

Values: **STATIONARY**

Own slot: **CONFIDENCE-FACTOR** from **CENTRAL-PROGRESSION**

Inheritance: **OVERRIDE.VALUES**

Values: **0.8**

Own slot: **CR-WEIGHT** from **CENTRAL-PROGRESSION**

Inheritance: **OVERRIDE.VALUES**

Values: **0.4**

Own slot: **CSNB-WEIGHT** from **CENTRAL-PROGRESSION**

Inheritance: **OVERRIDE.VALUES**

Values: **0.6**

Own slot: **EQUIVALENT-NUMERICAL-VALUE** from **CENTRAL-PROGRESSION**

Inheritance: **OVERRIDE.VALUES**

Values: **0.25**

Own slot: **PSEUDO-RP-WEIGHT** from **CENTRAL-PROGRESSION**

Inheritance: **OVERRIDE.VALUES**

Values: **0.8**

Own slot: **RC-WEIGHT** from **CENTRAL-PROGRESSION**

Inheritance: **OVERRIDE.VALUES**

Values: **0.6**

Figure 6.6: The "Central-progression" unit

PATIENTS Slots

member:

CR-PROB (local)

CR-TOTAL (local)

CSNB-PROB (local)

CSNB-TOTAL (local)

NAME (local)

NUMBER (local)

PSEUDO-RP-PROB (local)

PSEUDO-RP-TOTAL (local)

RC-PROB (local)

RC-TOTAL (local)

TESTS-DONE (local)

TOTAL-TESTS-DONE (local)

own:

DECOMPOSITION.COMPOSITE

DECOMPOSITION.DISJOINT

MEMBERS.DATATYPE

MEMBERSHIP

SUBCLASSP

Figure 6.7: *The general slots of a patient unit.*

- number: This slot contains the patient number of a patient, which also happens to be the name of that patient unit.
- cr-total: This slot contains the total score (see section 6.3.2) for the disease CR-Dyst in the patient.
- cr-prob: This slot contains the probability of the presence of the disease CR-Dyst in the patient.
- csnb-total: This slot contains the total score (see sec. 6.3.2) for the disease CSNB in the patient.
- csnb-prob: This slot contains the probability of the presence of the disease CSNB in the patient.
- rc-total: This slot contains the total score (see sec. 6.3.2) for the disease RC-Dyst in the patient.
- rc-prob: This slot contains the probability of the presence of the disease RC-Dyst in the patient.
- pseudo-rp-total: This slot contains the total score (see section 6.3.2) for the disease Pseudo-RP in the patient.
- pseudo-rp-prob: This slot contains the probability of the presence of the disease Pseudo-RP in the patient.
- total-tests-done: This slot contains the total number of tests performed on the patient so far.
- tests-done: This slot contains the names of the tests performed on the patient so far.

The other slots are system slots[19].

6.3.2 Inference Level

- Design:

At the inference level, the model of the FRD KB system, has an inference structure, which clearly states the steps of inference that can be made on the domain knowledge. The structure is made up of metaclasses and knowledge sources.

A domain concept can play one or more roles in an inference process. For example, a concept 'fever' can play the role of 'triggering hypotheses' in the inference process 'diagnosis'. A group of concepts playing the same role in an inference process is called a metaclass. Figure 6.8 gives a list of metaclasses used in the FRD KB system. For example, 'test-results' is a metaclass. The results obtained from performing tests are examples of this metaclass, e.g.: '20/20' for the test 'acuity'. The role of this metaclass is to get quantitative or numerical weightings (which represent the severity of the test result) from the qualitative result obtained by performing the test. Concepts such as test-weights, confidence factor etc. belong to the metaclass 'numeric-data' and are used to trigger hypotheses. Concepts such as CSNB, RC-Dyst, CR-Dyst and Pseudo-RP belong to more than one metaclass as they have more than one role in the inference process. They form the metaclass 'hypotheses',

Metaclass	Domain Concept	Role	Inference process
1. Test-results (associated with all tests)	stationary (test: progression) 20/20 (test: acuity)	as qualitative results to get quantitative or numeric data.	diagnosis
2. Numeric data	confidence factor, test weights, severity, total score, prev. score, # of tests	to trigger hypotheses.	diagnosis
3. Hypotheses	CSNB, RC-Dyst, CR-Dyst and Pseudo-RP.	to refine the diagnoses.	diagnosis
4. Diseases	CSNB, RC-Dyst, CR-Dyst and Pseudo-RP.	as final diagnoses	diagnosis
5. Output	CSNB, RC-Dyst, CR-Dyst and Pseudo-RP.	as final diagnoses presented to the user.	diagnosis

Figure 6.8: *Metaclasses in the FRD KB system.*

abstracts (T,Rt):-
 test-results (T),
 numeric-data (Rt),
 associated-with (T,Rt).

pursue (D,H):-
 hypotheses(H),
 diseases (D),
 refines (H,D).

weigh (D,Prob):-
 hypotheses (D),
 numeric-data (Rt),
 numeric-data (CFt),
 numeric-data (Wt),
 numeric-data (#-of-tests),
 numeric-data (prev-score),
 numeric-data (total-score),
 is-associated-with (D,Wt),
 total-score = [Rt X CFt X Wt] + prev-score

$$\text{Prob} = \frac{\text{total-score}}{\text{\#-of-tests}}$$

rank (D1,D2):-
 output (D1),
 output (D2),
 numeric-data (Prob1),
 numeric-data (Prob2),
 is-prob-of (Prob1,D1),
 is-prob-of (Prob2,D2),
 Prob1 > Prob2,
 output-first (D1),
 output-next (D2).

Figure 6.9: *The knowledge sources in the inference level.*

when they are used to refine diagnoses. They belong to the metaclass 'diseases', when they play the role of final diagnoses. And finally, they belong to the metaclass 'output', when they are presented to the user as final diagnoses.

Knowledge Sources are the descriptions of the ways in which a domain relation and concepts within one or more metaclasses can be used to make inferences. Figure 6.9 lists the knowledge sources used in the system. "Abstract" is a knowledge source which states that if T is a test result, Rt is a

numeric-data, and T is 'associated-with Rt'; then T gets abstracted to a numeric weighting Rt. In this knowledge source, 'associated-with' is a domain relation. The knowledge source "weigh" is used to calculate the probabilities for the presence of a disease. Every time a test is performed on a patient, the total score and the probability for every disease is calculated. The knowledge source "weigh" states that, if D belongs to metaclass 'hypotheses', then its probability 'Prob' is the total score for the disease divided by the total number of tests done on the patient. The total score is $\sum_n R_t C_{Ft} W_t$, where R_t is the normalised value of a numerical value associated with the result of a test (i.e. a normalised numeric weighting to indicate the severity); C_{Ft} is the confidence of the physician in the result of a test; and W_t is the significance of the test in the diagnosis of the disease and n is the number of tests done on the patient (see appendix A). The knowledge source "pursue" states that H is pursued from D, if H is in metaclass 'hypotheses', D is in metaclass 'diseases' and H is refined to D (i.e. H and D participate in the 'refines' association). The knowledge source "rank" is used to rank the diseases in the decreasing order of their probabilities. It states that if D1 and D2 belong to the metaclass 'output', if 'Prob1' and 'Prob2' are their probabilities respectively, and if 'Prob1' is greater than 'Prob2', then first output D1 and next output D2. Please note that one or more actions (e.g. in 'rank' output-first(D1) is called an action) are implanted in the knowledge source. This does not reduce the flexibility as any time that the knowledge source is employed by the task level, these actions must also be performed.

Figure 6.10: *The inference structure in the FRD KB system.*

Metaclasses and knowledge sources are used to construct an inference structure, as shown in figure 6.10. The steps of inference are as follows. 'Test-results' are "abstracted" to get the 'numeric-data'. The 'numeric-data' are "weighed" to get the various 'hypotheses'. The 'hypotheses' are "pursued" to get the 'diseases'. And the 'diseases' are "ranked" and 'output'.

In this model of inference, the steps of inference are clear. The process of inference can be updated, altered or refined, without having any consequences on the domain knowledge, as the two are quite independent. Most existing medical KBs do not have an inference level in their design. The steps of inference are embedded either in the domain or the task levels. This is clearly one of the major differences between the FRD KB system and the other systems.

- **Implementation:**

Metaclasses are implemented as variables, with their values equal to any of the associated domain concepts at a given time. The four knowledge sources in the inference structure - abstract, weigh, pursue and rank are implemented as units in KEE. Some of them are implemented using one or more units with methods in them and others are implemented using one or more active value units. These units are shown in figure 6.2 and are enumerated as follows.

- data-abstractors: This is a dummy unit, which groups together all the active value units. The active value units are used for data abstraction. There is an active value unit attached to the actual-test-value slot of every test unit. A method in the active value triggers, whenever the value of the actual-test-value slot changes. This method converts the actual test result to an equivalent numerical value and puts this value into equivalent-numerical-value slot of the corresponding test unit.

For example, "media.activevalue" is an active value unit which is attached to the actual-test-value slot of the "media" unit. If, during the process of diagnosis, the result of the test media changes to "marked". Then the method (see appendix B) in media.activevalue converts this value into an equivalent numerical value of 1. A value of 1 is put into the equivalent-numerical-value slot of the media unit.

- estimate-probability: This is a unit with a method (see appendix B) in it to weigh the results of the tests and calculate the probabilities of the presence of the diseases. The method is triggered after the program acquires the result and the confidence factor values for a test to calculate the probabilities for the different diseases, after performing this test. The confidence factor, the equivalent numerical value and the significance of the test in diagnosing the disease are all multiplied to calculate the score of the disease, of that test. This is added to the previous score

of that disease to get a total score for that disease. The total score is then divided by the total number of tests done on the patient to get the probability of that disease. This is repeated for all the diseases and the probabilities are put into the slot probability-of-disease of the corresponding disease unit.

For example, let us assume that 2 tests have been performed on a patient 'x'. Let the third test on 'x' be 'central-progression'. Let the equivalent numerical value of the central progression test be 0.25, the confidence factor be 0.5, and the significance of this test in diagnosing CSNB be 0.6. And let the previous score for CSNB at the end of performing the 2 previous tests be 1.125. Then the total score for CSNB after performing this test is $[0.25 \times 0.5 \times 0.6 = 0.075] + 1.125 = 1.2$. Since 3 tests have been performed on the patient so far, then the probability of the presence of CSNB is $1.2/3 = 0.4$.

- pursue-hypothesis: This is a unit which contains a method (appendix B) in its slot. The method is used to pursue and refine the list of hypotheses. It is like an interface procedure, which gets the test result and the confidence factor from the user and sends messages to the other methods to calculate the disease probabilities.

- rank-diagnosis: This unit contains a method (appendix B) to rank and display the output. The diseases are ranked in the decreasing order of their probabilities and then they are output. The current status of the patient are stored in the knowledge base.

The methods in all the above units are written in common LISP and they are given in appendix B.

6.3.3 Task Level

- Design:

At this level, a strategy which determines how and when certain inference steps will be made is defined. The task is represented as a 'goal tree' with corresponding task decompositions.

Figure 6.11 shows the task hierarchy for the FRD KB system. The main task is to "diagnose". This task is accomplished by performing 3 subtasks: "identify initial hypotheses", "explore and refine hypotheses" and "output final diagnosis". An initial list of hypotheses is identified by performing some initial tests and weighing the results of those tests. The initial list of hypotheses are explored and refined. This is done by performing some more tests or repeating tests already done. The results of the tests are again weighed to get a refined list of hypotheses. The task of exploring and refining the hypotheses is repeated, until the final results seem satisfactory. This is

Figure 6.11: A task structure for the FRD KB system.

when the final diagnosis is output. The final list of hypotheses is ranked in a decreasing order of importance and output.

The above task structure is one way of combining and using the steps of inference. This structure should model the actual method of diagnosis employed by the physician. The design of the task level not only eases the process of KB development, but also helps the expert evaluate his/her diagnostic strategy and if needed improve upon it. Most medical KBs employ the functionality of a task layer implicitly and they do not have a task layer in their system. NEOMYCIN[8] is an exception, i.e. it explicitly incorporates a task level in its architecture.

- Implementation:

The chief task consists of diagnosis which is implemented as "diagnose -the-patient" unit in KEE (refer to figure 6.2). This unit has a method in its "diagnose" slot (see appendix B) that runs the entire diagnostic procedure, by triggering other methods in other units.

There are two other methods which assist the process of diagnosis. They are the methods in the "initialise" and the "patient-checker" units (see appendix B). The method in the initialise unit initialises the system by putting in the default values into the various slots of the different units everytime a new diagnostic procedure starts. The method in the patient-checker unit

does the following. It checks if the patient had been diagnosed earlier by the system and if so, sets up the system to represent the state of the patient where it had last left and continues. If it is the first time the patient has ever been diagnosed by the system, then the method dynamically creates a unit to represent the state of the patient.

The method in the unit `diagnose-the-patient` initially triggers the above two methods to execute the two procedures explained above. It then calls the method in the unit `pursue-hypothesis` to identify initial hypotheses. This method acquires information from the user, such as the name of the test to be performed on the patient, test result and the confidence factor. This corresponds to the subtask 'perform initial tests'. Next, the method in `estimate-probability` unit is triggered to calculate the probability of the diseases and the method in the `rank diagnoses` unit is triggered to rank hypotheses. This corresponds to the 'weigh and rank results' subtask. The above procedure is repeated (to accomplish the 'explore and refine hypotheses' task) until the user is satisfied and indicates so to the system. And lastly the list of possible diagnoses are output by sending a message to the method in the `rank-diagnosis` unit.

The above units with the methods in them are displayed in figure 6.2. The unit "`a-frd-task`" is a dummy unit, which is used to bind all the units at the task level. The methods in the slots, which are all written in common LISP

are given in appendix B.

6.4 User Interface

A knowledge base system implemented as described above will be incomplete without any user interface facilities. For example, during the process of diagnosis, the system may ask the user certain information (such as the result of a test etc.). Without any user interface facilities, the user has to know LISP to input values as the system understands only LISP. To avoid this inconvenience to the user, some basic user interface facilities were added to the system as described below.

Active images are incorporated in the system for user interface purposes. The unit "keepicture.instances" in figure 6.2 represents all the active images used in the system.

When the knowledge base is loaded, an active image with "start the diagnosis of a patient" written inside it automatically appears. This is a method actuator (see section 5.5), which when clicked upon starts the process of diagnosis. This is shown in figure 6.12(a).

During the process of diagnosis, the system sometimes asks the user questions to be answered with a yes/no option. To prevent the user the inconvenience of typing in the values, an active image is displayed each time. The image is of the vertical polybutton type[21]. It displays "yes" and "no", and any one can be se-

Figure 6.12: The active images for user interface in FRD KB system.

PATIENT NAME

SHASHI

PATIENT NUMBER

P1000

Figure 6.13: The active images which display the current status of a patient.

lected by clicking on it. This image is shown in figure 6.12(d).

As the diagnosis proceeds, 3 active images display the list of tests employed in the diagnosis of FRD diseases. This is shown in figure 6.12(b). A mouse click on any item selects it. This image is of the vertical polybutton type[21]. An active image is attached to the actual-test-value slot of every test unit. This image displays the different results of performing a test. For example, the test media can result in any one of the following values - nil, identifiable, marked, moderate or unknown as shown in figure 6.12(c). Again mouse clicks are used to select the different values. This image is of the vertical polybutton type too[21]. To open and close the appropriate test results' image as and when required, the unit "display-values" has an appropriate method. This method is given in appendix B. Whenever the test list or the test results' list is displayed, an active image of the vertical pushbutton type[21] appears. This is shown in figure 6.12(e). When the user has completed selecting a test or a value for a test, the user is supposed to click on this image, which indicates to the system that he/she has completed choosing a test or a test result. This gives the user a chance to decide before making the final choice for a test or a test result.

Finally, there are active images that display the status of the patient currently being diagnosed. Two active images of the simple value display type (see section 5.5), show the patient name and patient number. The probabilities of the presence of the different diseases are shown by the four active images of the thermometer type (see section 5.5). These appear as shown in figure 6.13. The figure

shows a patient whose name is "shashi" with patient number "P1000". The probability of this patient having CSNB is 0.6, CR-Dyst is 0.4 etc.

6.5 KB Implementation Environment

The FRD KB ~~system~~ prototype was implemented using the expert system building tool, KEE, version 3.0; running on the TI - Explorer level 2.0 software. The system has a total of 204 units, out of which 43 units are used to implement the system model and the rest are used to implement the active images. All programs are written in Common LISP.

6.6 Testing

The FRD KB System prototype has been demonstrated to the expert to his satisfaction, and it is verified that the system can be employed as a primitive diagnostic system, as well as a research tool.

Chapter 7

Conclusions

7.1 Contributions of this Research

The major contribution of this research work is the development of a KB system prototype in the FRD domain. The reason for having chosen this domain is that it has a particular property, i.e. characteristics of the FRD diseases overlap substantially (see chapter 2). The problem of diagnosis in a domain with such property is a complex task, and a KB system is a good candidate to be used as an aid. The purposes of building the FRD KB system are as follows. The system can be employed as a diagnostic system, which advises the user regarding the diagnosis of patients with FRD diseases. The system can also be used as a research tool, as explained in the following paragraphs.

In the first stage of KB development, in order to design a model for the FRD expertise, a number of existing medical KB systems were reviewed. And a model[36]

was chosen and applied to the design of the KB system. The model is a layered one, with four levels, each with a different type of knowledge. They were described in detail in chapter 6.

The following observations regarding the layered model were made. The model made the interpretation of the data from the experts more effective. Moreover, the separation of knowledge into several layers in the model, eased the process of KB development. Implementation and modification of the knowledge are easier to accomplish as knowledge is separated into several layers in the model.

However, the model does have disadvantages. First, there is no proof that the model could be successfully applied to all applications. Moreover, there are no guidelines for the description and selection of alternative metaclasses, knowledge sources and strategies.

The next stage was the implementation of an initial prototype for the FRD KB system, using the model developed in stage one, using the expert system building tool KEE. Currently, the prototype performs a primitive diagnostic task. Given the test results of a patient, it assists the user in diagnosing the patient with FRD diseases. The user performs a number of tests on the patient, and gives the results of those tests to the system. Based on the test results, the test confidence factor, and the test-disease significant factor, the system calculates the probabilities for the presence of the diseases. The system uses a technique of statistical averaging to calculate the disease probabilities (see section 6.3.2).

This prototype can also be used as a research tool. Since the result of the diagnosis of a patient by the prototype is based on values of significance factors, it is more appropriate to use the system as a research tool according to the following procedures. Test results of patients whose diagnosis is already known can be fed to the system. The patients' disease probabilities calculated by the system can be compared to their associated already known results of diagnosis, and test-significance factors can be adjusted accordingly.

Moreover, the existing KB system can be enhanced in a number of ways as explained in the following section.

7.2 Future Developments

The FRD knowledge base system can be enhanced in the following ways to develop an expanded prototype system in the FRD domain.

1. At present the FRD KB system has only two layers at the domain level of the model. They are the tests and the diseases. This two layered domain model can be expanded to include more layers of abstraction, such as intermediate diagnosis etc. For instance, the results of a test, in some cases can be intermediate diagnoses (e.g. the results 'moderate' and 'marked' of the test 'media'). Currently, during the nonautomated diagnostic procedure, the

physician specifies an intermediate diagnosis, which is based on the results obtained from the performance of a number of tests. These intermediate diagnoses may determine other intermediate diagnoses. And finally, a set of intermediate diagnoses in a patient may determine a disease.

Currently, the KB system permits only positive numbers between 0 and 1 as values for the confidence factor and the test weight (i.e. the significance of a test in diagnosing a disease). As well, the system can be enhanced to handle negative numbers for both of them. For example, a negative test weight corresponds to the significance of the test against the diagnosis of a disease. And the negative confidence factor shows how confident the physician is that the result of a test for a patient is incorrect.

2. At the inference level, the FRD KB system handles uncertainty using simple technique of summation and averaging to calculate the disease probabilities (see sec. 6.3.2). Other methods can be used to calculate the probabilities (e.g. a method similar to the way uncertainty is handled in MYCIN[10]).
3. The FRD prototype can be enhanced to perform a number of other tasks, such as those explained in the following.

The system can be improved to show conflicting tests and test results. For example, if after performing test 'A' on a patient, he/she is diagnosed as

having disease 'X', after performing test 'B' on the same patient he/she is diagnosed as having disease 'Y', and 'X' and 'Y' cannot be present simultaneously in a patient, then the 2 tests are conflicting. The system must indicate this to the user and allow the user to change or correct weight values accordingly.

The FRD KB system can be modified to include machine learning techniques. The resulting system can recognise new patterns of diseases, by clustering several patients' data with special characteristic features and thus discover new classifications of FRD diseases.

Once the FRD KB system is enhanced to accommodate the above features, it can be improved to take the role of a tutor which can teach the user the diagnostic process and the steps involved in diagnosing a patient.

4. Explanation facilities can be added to the FRD KB system. The system can be made to explain its process of diagnosis and the inference it employs. It can also be made to explain how and why certain information was used. For example, the system can be made to explain 'how it calculates the disease probabilities', 'why did the probability of a disease change significantly after performing a particular test' etc. More sophisticated graphic capabilities can be added.

5. Finally, the FRD patients' data can be stored in a database (e.g. an ORACLE DBMS) and the FRD KB system can be integrated with it (as KEE - ORACLE connection software is provided by Intellicorp).

Bibliography

- [1] ADLASSNIG, K.P.; KOLARZ, G.; SCHEITHAUER, W.; and GRABNER, H.: *Approach to a hospital-based application of a medical expert system*; *Medecine et Informatique* (London); 11 (3); July - Sept., 1986.; pp.205-223.
- [2] BEMMEL, JAN H.VAN : *Formalisation of medical knowledge - the basis for diagnostic strategies and expert systems*; *Medical Decision Making: Diagnostic strategies and expert systems*; Elsevier Science Publishers; 1985; pp.1-11.
- [3] BLUM, ROBERT L : *Discovery and Representation of Causal Relationships from a large Time-Oriented Clinical Database : The RX Project*; Lecture notes in Medical Informatics; volume 19, 1982.
- [4] BUCHANAN, BRUCE G. : *Expert systems, Tutorial*; Tenth International Joint Conference on Artificial Intelligence; Milano, August 23 - 28, 1987.
- [5] CARR, RONALD E. and SEIGAL, IRWIN M. : *Visual Electrodiagnostic Testing - a practical guide for the clinician*; Williams and Wilkins Publications, Baltimore/London, 1982.
- [6] CHYTIL, M.K. and ZVAROVA, J. : *Medical Knowledge, Problems and Expert Systems*; *Medical Decision Making: Diagnostic Strategies and Expert Systems*; Elsevier Science Publishers; 1985; pp.38-41.
- [7] CITRENBAUM, RONALD; GEISSMAN J.R. and SCHULTZ, R. : *Selecting a Shell*; *AI Expert*, September 1987; pp.30-39.
- [8] CLANCEY, W.J. and LETSINGER, R. : *NEOMYCIN: Reconfiguring a rule-based expert system for application to teaching*; *Readings in Medical AI*

- The first decade; Addison-Wesley Publishing Company; 1984; pp. 361-381.
- [9] COX, BRAD J. : *Object-oriented programming - An Evolutionary Approach*; Addison-Wesley Publishing Co., Inc.; 1986.
- [10] DAVIS, R.; BUCHANAN, B.G. and SHORTLIFFE, E.H. : *Production rules as a representation for a knowledge based consultation program*; Readings in Medical AI - The first decade; Addison-Wesley Publishing Company; 1984; pp. 98-130.
- [11] FARNSWORTH, DEAN : *The Farnsworth - Munsell 100 hue test for the examination of color discrimination - Manual*; Munsell Color Company, Inc.; Baltimore, 1957.
- [12] FINLAY, A.Y. and HAMMOND, P. : *Expert Systems in Dermatology: The Computer Potential*; *Dermatologica*, 173(2); 1986; pp. 79-84.
- [13] GAGLIO, S.; RUGGIERO, C.; SPINELLI, G.; BONADONNA, G.; VALAGUSSA, P. and NICOLINI, C. : *BREASTCAN: an expert system for postoperative breast cancer therapy*; *Computer Biomedical Research*, October 19(5); 1986; pp. 445-461.
- [14] GENESERETH, MICHAEL R. : *Diagnosis Using Hierarchical Design Models*; American Association of Artificial Intelligence; 1982; pp.278-283.
- [15] GILMORE, JOHN F. and HOWARD, CHARLES : *Expert system tool evaluation*; SPIE Applications of Artificial Intelligence; Bellingham, Wash.: SPIE; 1986; pp.437-458.
- [16] GOMEZ, FERNANDO and CHANDRASEKARAN, B. : *Knowledge Organisation and Distribution for Medical Diagnosis*; Readings in Medical AI - The first decade; Addison-Wesley Publishing Company; 1984; pp. 320-338.
- [17] JOHNSON, L. and KERAVALQU, E.T. : *Expert systems technology - a guide*; Information Technology and Systems series, Abacus Press; 1985.

- [18] JOHNSON, W. LEWIS and SOLOWAY, ELLIOT : *PROUST - An Automatic Debugger for Pascal programs*; Artificial Intelligence; Byte; April, 1985; pp.179-190.
- [19] KEE SOFTWARE DEVELOPMENT SYSTEM: LEVEL 3.0; *Release Description and Installation guide, User's manual, Core Reference manual*; Volume I; Sperry Corporation, 1986.
- [20] KEE SOFTWARE DEVELOPMENT SYSTEM: LEVEL 3.0; *KEE Worlds Reference manual, Rulesystem3 Reference manual, TellAndAsk Reference manual*; volume II; Sperry Corporation, 1986.
- [21] KEE SOFTWARE DEVELOPMENT SYSTEM: LEVEL 3.0; *Active Images Reference manual, KEE pictures reference manual, Common Windows reference manual*; volume III; Sperry Corporation, 1986.
- [22] KEE Training Workshop *Student guide*; Sperry Corporation, March 1986.
- [23] KRILL, ALEX E. : *Hereditary Retinal and Choroidal Diseases - volume I*; Harper and Row, Hagerstown; 1972.
- [24] KRILL, ALEX E. : *Krill's Hereditary Retinal and Choroidal Diseases - volume II, Clinical Characteristics*, Harper and Row, Hagerstown; 1977.
- [25] MARS, N.J.I. and MILLER, P.L. : *Knowledge Acquisition and Verification tools for medical expert systems*; Medical Decision Making; volume 7, 1987; pp. 6-11.
- [26] MARS, N.J.I.; MILLER, P.L.; BLUMENFRUCHT, S.J.; ROSE, J.R.; ROTH-SCHILD, M.; SWETT, H.A. and WELTIN, G. : *HYDRA - a knowledge acquisition tool for expert systems that critique medical workup*; Medical Decision Making; volume 7, 1987; pp. 12-21.
- [27] MILLER, R.A.; POPLE, H.E.Jr. and MYERS, J.D. : *INTERNIST-1, an experimental computer-based diagnostic consultant for general internal medicine*; Readings in Medical AI - The first decade; Addison-Wesley Publishing Company; 1984; pp.190-209.

- [28] MIYAKE, YOZO; YAGASAKI, KATSUYA; HORIGUCHI, MASAYUKI; KAWASE, YOSHIKATSU; and KANDA, TAKAKO : *Congenital Stationary Night Blindness with Negative Electroretinogram - A New Classification*; Arch Ophthalmology; Vol 104, July, 1986; pp. 1013-1020.
- [29] PAUKER, S.G.; GORRY, G.A.; KASSIRER, J.P. and SCHWARTZ, W.B. : *Towards the simulation of clinical cognition: taking a present illness by computer*; Readings in Medical AI - The first decade; Addison-Wesley Publishing Company; 1984; pp. 131-159.
- [30] SCHUBERT, L.K. and JOHANNES, DE HAAN : *Inference in a topically organised semantic net*; American Association of Artificial Intelligence; 1986; pp.334-338.
- [31] SIX, P. and CHAUVET, G. : *An efficient structure for a medical information database in a general hospital*; International Journal of Biomedical Computers (Netherlands); May 1986; pp. 213-227.
- [32] TAKENOUCHI, H. and IWASHITA, Y. : *An integrated knowledge representation scheme for expert systems*; Expert Systems; volume 4, number 1, Feb. 1987; pp.38-43.
- [33] VERSTRATEN, KARIN L. : *Visual fields - Supplement for Ophthalmic assistants home study course, Part I*; Canadian Ophthalmological Society home study course for Ophthalmic assistants, Study guide phase I; Centennial College of Applied Arts and Technology; Scarborough, Ontario; 1981.
- [34] WEISS, S.M.; KULIKOWSKI, C.A.; AMAREL, S. and SAFIR, A. : *A model based method for computer-aided medical decision making*; Readings in Medical AI - The first decade; Addison-Wesley Publishing Company; 1984; pp. 160-189.
- [35] WERNER, GERHARD : *Clinical Problem Formulation in an illstructured domain*; Proceedings of the 6th annual workshop on Expert Systems and their applications; volume I; April 28-30, 1986; pp. 323-336.
- [36] WIELINGA, B.J. and BREUKER, J.A. : *Models of Expertise*; European Conference on Artificial Intelligence; 1987; pp.306-318.

[37] WINSTON, PATRICK HENRY : *Artificial Intelligence*; Second Edition;
Addison-Wesley Publishing Co., Inc.; 1984.

[38] WINSTON, P.H. and HORN, BERTHOLD KLAUS PAUL : *LISP*, Second
Edition; Addison-Wesley Publishing Co., Inc.; 1984.

Appendix A

Test Disease Relationships

In this appendix details of various tests are explained using the following steps. First, the meaning of each test is explained. Next, the method of performing a test, results measured for the test and the representation of these results are given. Finally, the association between the test and the various diseases is described.

For each test, the spectrum of the possible results are divided into a number of categories each called a severity category. To each severity category, a number is assigned in the increasing order of severity. For example, in progression, there are four severity categories numbered 1 to 4 whereas in visual field, there are seven categories numbered 1, 2.1, 2.2, 2.3, 2.4, 3. It is this value indicated in the table besides its associated severity category. The number assigned to each severity category is rescaled to a value between 0 and 1 (e.g. 0.25, 0.5, 0.75 and 1 for progression severity categories and 0.166, 0.233,1 for visual field category). It is this normalised value which is employed in the "weigh" knowledge source given in chapter 6.

The test significance factors (which represent the significance of the test in diagnosing the diseases) are shown in the rows "WEIGHTS" of the table. They are values on a scale of 0 to 5. For example, significance of the test progression in diagnosing the disease CSNB is 3. It is important to note that these values are also normalized between 0 and 1 before using them in the knowledge source "weigh" in chapter 6. Following is the list of tests used in the FRD diagnosis.

1. Progression.

Progression is a test, the result of which is determined from discussions with the subject and from comparison of data, over the years. The different types of progressions (i.e. central and peripheral) were described earlier in chapter 2.

The rate of progression of the disease is determined as the result of this test. These results are given in the table that follows. As seen in the table, progression can be 'stationary', 'progressing for worse every 20 years or more', 'progressing every 5 to 20 years' or 'progressing every 2 to 5 years'. The severity categories of this test are numbered 1,2,3 and 4 as given in the table, where the first result is the least severe ('stationary' in this case) and the last result is the most severe ('in 2 to 5 years' here).

The test significance factors are shown in the row "WEIGHTS" of the table.

2. Family Tree

An important part of history taking is to get the patient's genetic background. The physician talks to the patient about the latter's family, and draws a family tree. From the family tree, the physician determines the manner in which the disease has been inherited.

The result of performing this test as indicated in the table are the different types of inheritances. The type of inheritance may be autosomal recessive, autosomal dominant or x-linked recessive, or the patient may show no familial tendency (and are explained in chapter 2). The severity categories of this test are numbered 1,2,3 and 4 as given in the table, where the first result is the least severe ('autosomal dominant' in this case) and the last result is the most severe ('no-familial tendency' here).

The test significance factors are shown in the row "WEIGHTS" of the table.

3. Acuity

This is a clinical examination performed by the physician. The measurements make an attempt to reflect the ability to identify objects.

The result of this test is numerical. For example, an acuity of 20/40 means that a patient can just identify an object placed at a distance of 20 feet, what a normal person could clearly identify the same at 40 feet. The results are given as ranges in the table that follows. For example, the first result is a range of the measure of acuity between 20/10 and 20/20. The severity categories of this test are numbered 1,2,3,4 and 5 as given in the table, where the first result is the least severe ('[20/10,20/20]' in this case) and the last result is the most severe ('[20/400,20/infinity' here).

The test significance factors are shown in the row "WEIGHTS" of the table.

4. Media

This is a clinical examination performed by the physician. The physician studies the fluid in the retina using a light pen.

The results of this test, as shown in the table, can be 'nil, identifiable, marked or moderate'. The severity categories of this test are numbered 1,2,3 and 4 as given in the table, where the first result is the least severe ('nil' in this case) and the last result is the most severe ('marked' here).

The test significance factors are shown in the row "WEIGHTS" of the table.

5. Fundus

This is a clinical examination performed on a patient, wherein the physician, with special instruments, observes the inner surface or the fundus of the eye. The morphological appearance of the inside of the eye is viewed through the pupil.

The different results of performing this test are indicated in the table. The severity categories of this test are numbered 1,2,3 and 4 as given in the table, where the first result is the least severe ('PVR' in this case) and the last

result is the most severe ('choroidal sclerosis to some degree' here).

The test significance factors are shown in the row "WEIGHTS" of the table.

6. Visual Field

The visual field is the total area of space which is visible to a person while the eye is fixed in one direction. This test is performed by the physician, instructing the patient to fix his/her gaze on a white "+" sign on a black screen. A flashlight is beamed at various points and the patient is instructed to say when he/she sees the light. The result of this test is drawn on the field chart (see chapter 2).

The results of this test are the various patterns of the field chart. They are enumerated in the table. For example, the pattern may be normal, may have a ring deficit (the ring may be less than half, more than half, full or it may break through the periphery) etc. The severity categories of this test are numbered 1,2.1,2.2,2.3,2.4 and 3 as given in the table, where the first result is the least severe ('normal to 2mm' in this case) and the last result is the most severe ('central field left to 10mm' here).

The test significance factors are shown in the row "WEIGHTS" of the table.

7. Electroretinogram (ERG)

A flash of light elicits a distinctive electrical response from the human eye to the brain. This is called the electroretinogram (ERG). This is a lab test. The responses of the rods and cones to light of various colors, frequencies and intensities and the electrical response of associated retinal cells are measured in this test with special sophisticated instruments. There are two different measurements made from each response. They are the amplitude and the implicit time of the response. The measurements are made under different conditions and based on this, there are different types of ERG tests, such as the cone ERG, rod ERG and dark adaptation. The test results shown in the table, are the standard deviations of the above measurements from those measurements known to be normal.

The different results of the test are: 1.5 to 2 standard deviation (sd), $>2.5sd$ to $3.5sd$, $>3.5sd$ but not extinguished and completely extinguished. The severity categories of this test are numbered 1,2,3 and 4 as given in the table, where the first result is the least severe ('1.5 to 2.5sd' in this case) and the last result is the most severe ('extinguished to routine study' here).

The test significance factors are shown in the row "WEIGHTS" of the table.

8. 100 Hue

The Farnsworth-Munsell 100 hue test offers a method for testing the color discrimination capability. This is a lab test.

The test consists of identifying 93 plastic caps with different colors. The result of this test gives an error score for color discrimination for each of the color plastic caps. These scores are plotted on a chart to get a color defective pattern (see chapter 2). The scores are employed to calculate a final error score. The standard deviations of these scores from those measurements which are known to be normal are shown in the table. The severity categories of this test are numbered 1,2,3 and 4 as given in the table, where the first result is the least severe ('normal score' in this case) and the last result is the most severe ('>?500' here).

The test significance factors are shown in the row "WEIGHTS" of the table.

ID					
NI					
TA					
EG					
RN					
MO					
DISEASE	k	1	2	3	4
DIS		CSNB	CR	RC	Pseudo
IS			Dyst	Dyst	RP
A					
T					
E					

Progression

WEIGHTS, W1k

3

2

3

4

Intermediate Diagnostic

States

1. Stationary
2. progression identified in $>$ or $=$ 20 years
3. in 5 to 20 years
4. in 2 to 5 years

Family tree

WEIGHTS, W2k

3

2

3

4

Intermediate Diagnostic

States

1. Autosomal Dominant
2. Autosomal Recessive
3. X linked Recessive
4. No Familial tendency

Acuity

WEIGHTS, W3k

3

3

3

0

Intermediate Diagnostic

States

1. [20/10, 20/20]
2. [20/20, 20/40]
3. [20/40, 20/100]
4. [20/100, 20/400]
5. [20/400, 20/inf]

INITAL EQUIPMENT MO					
DISEASE	k	1	2	3	4
DIS		CSNB	CR	RC	Pseudo
IS			Dyst	Dyst	RP
ATE					

Media

WEIGHTS, W4k 4 3 4 2

Intermediate Diagnostic
States.

1. nil
2. identifiable
3. moderate
4. marked

Fundus Pathology

WEIGHTS, W5k 3 3 3 3

Intermediate Diagnostic
States

1. Identifiable loss of
uniformity in pigmentation
at rpe or perivenous retinal
clumping of pigment
identifiable (PVRC).
2. Sector of PVRC.
3. Uniform or generalised
PVRC or sine pigmenti.
4. Choroidal sclerosis
to some degree.

I D					
N I					
T A					
E G					
R N					
M O					
DISEASE	k	1	2	3	4
D I		CSNB	CR	RC	Pseudo
I S			Dyst	Dyst	RP
A					
T					
E					

Dark adaptation

WEIGHTS, W6k	4	3	4	2
--------------	---	---	---	---

Intermediate Diagnostic
States

(Minimum discriminable after 20 minutes of dark adaptation)

1. 1.5 to 2.5 sd elevated.
(stimulus ampl.
>= -7 log units)
2. >2.5 to 3.5.
3. > 3.5 but not extinguished.
4. Extinguished to routine study
(max stimulus)

Cone electro-retinogram

WEIGHTS, W7k	.3	3	3	2
--------------	----	---	---	---

Intermediate Diagnostic
States

(-3.0 or possibly response to the -2.0 log unit down stimulus)

1. 1.5 to 2.5 sd elevated
2. > 2.5 to 3.5
3. > 3.5 but not extinguished
4. Extinguished to routine study.

IDENTIFICATION					
DISEASE	k	1	2	3	4
DIS		CSNB	CR	RC	Pseudo
AST			Dyst	Dyst	RP
E					

Rod electro-retinogram

WEIGHTS, W8k 3 3 3 1

Intermediate Diagnostic
States

(30 Hz B wave may be the easiest study)

1. 1.5 to 2.5 sd elevated
2. > 2.5 to 3.5
3. > 3.5 but not extinguished
4. Extinguished to routine study.

Visual field (island of vision, light adapted)

WEIGHTS, W9k 4 3 3 2

Intermediate Diagnostic
States

1. normal to 2 mm
(? or 10 mm white).
2. Ring deficit identifiable.
 - 2.1 <= half ring
 - 2.2 > half ring
 - 2.3 full ring.
 - 2.4 Ring breaking thro
to periphery.
3. Central field left to
10 mm white at one meter
or equivalent.

```

(PUT.VALUE NUMBER 'TESTS-DONE TEMPOR)
(PUT.VALUE NUMBER 'NUMBER NUMBER))

-- ESTIMATE-PROBABILITY::PROB' method: (THISUNIT)
(PUT.VALUE 'CSNB 'PROBABILITY
  (+ (GET.VALUE 'CSNB 'PROBABILITY)
    (+ (GET.VALUE TEST 'CONFIDENCE-FACTOR) (GET.VALUE TEST 'CSNB-WEIGHT)
      (GET.VALUE TEST 'EQUIVALENT-NUMERICAL-VALUE))))
(PUT.VALUE 'CSNB 'PROBABILITY-OF-DISEASE
  (/ (GET.VALUE 'CSNB 'PROBABILITY) (GET.VALUE 'CSNB 'NUMBER-OF-TESTS)))
(PUT.VALUE 'CR-DYST 'PROBABILITY
  (+ (GET.VALUE 'CR-DYST 'PROBABILITY)
    (+ (GET.VALUE TEST 'CONFIDENCE-FACTOR) (GET.VALUE TEST 'CR-WEIGHT)
      (GET.VALUE TEST 'EQUIVALENT-NUMERICAL-VALUE))))
(PUT.VALUE 'CR-DYST 'PROBABILITY-OF-DISEASE
  (/ (GET.VALUE 'CR-DYST 'PROBABILITY) (GET.VALUE 'CR-DYST 'NUMBER-OF-TESTS)))
(PUT.VALUE 'RC-DYST 'PROBABILITY
  (+ (GET.VALUE 'RC-DYST 'PROBABILITY)
    (+ (GET.VALUE TEST 'CONFIDENCE-FACTOR) (GET.VALUE TEST 'RC-WEIGHT)
      (GET.VALUE TEST 'EQUIVALENT-NUMERICAL-VALUE))))
(PUT.VALUE 'RC-DYST 'PROBABILITY-OF-DISEASE
  (/ (GET.VALUE 'RC-DYST 'PROBABILITY) (GET.VALUE 'RC-DYST 'NUMBER-OF-TESTS)))
(PUT.VALUE 'PSEUDO-RP 'PROBABILITY
  (+ (GET.VALUE 'PSEUDO-RP 'PROBABILITY)
    (+ (GET.VALUE TEST 'CONFIDENCE-FACTOR) (GET.VALUE TEST 'PSEUDO-RP-WEIGHT)
      (GET.VALUE TEST 'EQUIVALENT-NUMERICAL-VALUE))))
(PUT.VALUE 'PSEUDO-RP 'PROBABILITY-OF-DISEASE
  (/ (GET.VALUE 'PSEUDO-RP 'PROBABILITY) (GET.VALUE 'PSEUDO-RP 'NUMBER-OF-TESTS)))
(PUT.VALUE TEST 'DONE2 (QUOTE ())))

(DEFUN :FRD>PURSUE-HYPOTHESIS::FURTHER' method: (THISUNIT)
  (TERPRI)
  (PRINT '(WHAT IS THE NAME OF THE PATIENT YOU WANT ME TO DIAGNOSE ?))
  (SETQ NAME (READ))
  (TERPRI)
  (PRINT '(GIVE ME THE PATIENT NUMBER OF ,NAME))
  (SETQ NUMBER (READ))
  (UNIT 'PATIENT-CHECKER 'CHECK)
  (DO ((PROCEED T))
    ((NULL PROCEED) NIL)
    (TERPRI)
    (PRINT '(THE TESTS EMPLOYED IN THE DIAGNOSIS OF FRD DISEASES ARE AS SHOWN ABOVE))
    (TERPRI)
    (PRINT '(YOU CAN PERFORM OR REPEAT - ANY OF THOSE TESTS))
    (TERPRI)
    (PRINT '(PLEASE CHOOSE THE TEST TO BE DONE ON ,NAME))
    (PUT.VALUE 'TESTS 'HISTORY-TO-BE-DONE 'UNKNOWN)
    (PUT.VALUE 'TESTS 'LAB-TEST-TO-BE-DONE 'UNKNOWN)
    (PUT.VALUE 'TESTS 'PHYSICAL-TO-BE-DONE 'UNKNOWN)
    (UNITMSG 'VIEWPORT-HISTORY-TO-BE-DONE-OF-TESTS.1 'OPEN-PANEL')
    (UNITMSG 'VIEWPORT-PHYSICAL-TO-BE-DONE-OF-TESTS.3 'OPEN-PANEL')
    (UNITMSG 'VIEWPORT-LAB-TEST-TO-BE-DONE-OF-TESTS.2 'OPEN-PANEL')
    (UNITMSG 'VIEWPORT-DONE-OF-TESTS.2 'OPEN-PANEL')
    (DO ((IDLE T))
      ((NULL IDLE) (UNITMSG 'VIEWPORT-HISTORY-TO-BE-DONE-OF-TESTS.1 'CLOSE-PANEL')
        (UNITMSG 'VIEWPORT-PHYSICAL-TO-BE-DONE-OF-TESTS.3 'CLOSE-PANEL')
        (UNITMSG 'VIEWPORT-LAB-TEST-TO-BE-DONE-OF-TESTS.2 'CLOSE-PANEL'))
      (COND

```

Appendix B

LISP Methods

(ADD P:MAP.TO.KB 'FRD.THERMOMETER-PROBABILITY-OF-DISEASE-OF-RC-DYST.8.MASK.2738 'FRD)

(DEFUN (FRD>VISUAL-FIELDS.ACTIVEVALUE::AVPUT'method: (THISUNIT SLOT NEWVALUE OLDVALUE UNIT SLOTT

```
(DECLARE (IGNORE SLOT OLDVALUE THISUNIT))
(SETQ X (GET.VALUE UNIT 'ACTUAL-TEST-VALUE))
(COND
  ((EQUAL X 'NORMAL-TO-2MM) (SETQ Y 0.166))
  ((EQUAL X '<=HALF-RING) (SETQ Y 0.33))
  ((EQUAL X '>HALF-RING) (SETQ Y 0.5))
  ((EQUAL X 'FULL-RING) (SETQ Y 0.66))
  ((EQUAL X 'RING-BREAKING-THRO-TO-PERIPHERY) (SETQ Y 0.833))
  ((EQUAL X 'CENTRAL-FIELD-LEFT-TO-10MM-WHITE-AT-1METER) (SETQ Y 1))
  (T (SETQ Y 0)))
(PUT.VALUE UNIT 'EQUIVALENT-NUMERICAL-VALUE Y)
NEWVALUE)
```

(DEFUN (FRD>FUNDUS.ACTIVEVALUE::AVPUT'method: (THISUNIT SLOT NEWVALUE OLDVALUE UNIT SLOTTTYPE)

```
(DECLARE (IGNORE SLOT OLDVALUE THISUNIT))
(SETQ X (GET.VALUE UNIT 'ACTUAL-TEST-VALUE))
(COND
  ((EQUAL X 'IDENTIFIABLE-LOSS-OF-UNIFORMITY-IN-PIGMENTATION-AT-RPE-OR-PVRC) (SETQ Y 0.25))
  ((EQUAL X 'SECTOR-OF-PVRC) (SETQ Y 0.5))
  ((EQUAL X 'UNIFORM-OR-GENERALISED-PVRC) (SETQ Y 0.75))
  ((EQUAL X 'CHOROIDAL-SCLEROSIS-TO-SOME-DEGREE) (SETQ Y 1))
  (T (SETQ Y 0)))
(PUT.VALUE UNIT 'EQUIVALENT-NUMERICAL-VALUE Y)
NEWVALUE)
```

(DEFUN (FRD>ACUITY.ACTIVEVALUE::AVPUT'method: (THISUNIT SLOT NEWVALUE OLDVALUE UNIT SLOTTTYPE)

```
(DECLARE (IGNORE SLOT OLDVALUE THISUNIT))
(SETQ X (GET.VALUE UNIT 'ACTUAL-TEST-VALUE))
(COND
  ((EQUAL X '20'10-TO-20'20) (SETQ Y 0.2))
  ((EQUAL X '20'20-TO-20'40) (SETQ Y 0.4))
  ((EQUAL X '20'40-TO-20'100) (SETQ Y 0.6))
  ((EQUAL X '20'100-TO-20'400) (SETQ Y 0.9))
  ((EQUAL X '20'400-TO-20'INF) (SETQ Y 1))
  (T (SETQ Y 0)))
(PUT.VALUE UNIT 'EQUIVALENT-NUMERICAL-VALUE Y)
NEWVALUE)
```

(DEFUN (FRD>EXTINGUISHED.ACTIVEVALUE::AVPUT'method: (THISUNIT SLOT NEWVALUE OLDVALUE UNIT SLOTTTYPE)

```
(DECLARE (IGNORE SLOT OLDVALUE THISUNIT))
(SETQ X (GET.VALUE UNIT 'ACTUAL-TEST-VALUE))
(COND
  ((EQUAL X '1.5-TO-2.5SD-ELEVATED) (SETQ Y 0.25))
  ((EQUAL X '>2.5-TO-3.5) (SETQ Y 0.5))
  ((EQUAL X '3-5-BUT-NOT-EXTINGUISHED) (SETQ Y 0.75))
  ((EQUAL X 'EXTINGUISHED-TO-ROUTINE-STUDY' (SETQ Y 1))
  (T (SETQ Y 0)))
(PUT.VALUE UNIT 'EQUIVALENT-NUMERICAL-VALUE Y)
NEWVALUE)
```

(DEFUN (FRD>EXTINGUISHED.ACTIVEVALUE::AVPUT'method: (THISUNIT SLOT NEWVALUE OLDVALUE UNIT SLOTTTYPE)

```

(DECLARE (IGNORE SLOT OLDVALUE THISUNIT))
(SETQ X (GET.VALUE UNIT 'ACTUAL-TEST-VALUE))
(COND
  ((EQUAL X 'NORMAL-ACCUMALATED-TOTAL-ERROR-SCORE-FOR-AGE-OR-<1.5SD) (SETQ Y 0.25))
  ((EQUAL X '1.5-TO-2.5SD) (SETQ Y 0.5))
  ((EQUAL X '>2.5-TO-500) (SETQ Y 0.75))
  ((EQUAL X '>500) (SETQ Y 1))
  (T (SETQ Y 0)))
(PUT.VALUE UNIT 'EQUIVALENT-NUMERICAL-VALUE Y)
NEWVALUE)

(DEFUN FRD FAMILY-TREE.ACTIVEVALUE::AVPUT'method: (THISUNIT SLOT NEWVALUE OLDVALUE UNIT SLOTTYP
E)
(DECLARE (IGNORE SLOT OLDVALUE THISUNIT))
(SETQ X (GET.VALUE UNIT 'ACTUAL-TEST-VALUE))
(COND
  ((EQUAL X 'AUTOSOMAL-DOMINANT) (SETQ Y 0.25))
  ((EQUAL X 'AUTOSOMAL-RECESSIVE) (SETQ Y 0.5))
  ((EQUAL X 'X-LINKED-RECESSIVE) (SETQ Y 0.75))
  ((EQUAL X 'NO-FAMILIAL-TENDENCY) (SETQ Y 1))
  (T (SETQ Y 0)))
(PUT.VALUE UNIT 'EQUIVALENT-NUMERICAL-VALUE Y)
NEWVALUE)

(DEFUN FRD PROGRESSION.ACTIVEVALUE::AVPUT'method: (THISUNIT SLOT NEWVALUE OLDVALUE UNIT SLOTTYP
E)
(DECLARE (IGNORE SLOT OLDVALUE THISUNIT))
(SETQ X (GET.VALUE UNIT 'ACTUAL-TEST-VALUE))
(COND
  ((EQUAL X 'STATIONARY) (SETQ Y 0.25))
  ((EQUAL X 'PROGRESSION-IN->OR=20YRS) (SETQ Y 0.5))
  ((EQUAL X 'PROGRESSION-IN-5-TO-20YRS) (SETQ Y 0.75))
  ((EQUAL X 'PROGRESSION-IN-2-TO-5YRS) (SETQ Y 1))
  (T (SETQ Y 0)))
(PUT.VALUE UNIT 'EQUIVALENT-NUMERICAL-VALUE Y)
NEWVALUE)

(DEFUN FRD MEDIA.ACTIVEVALUE::AVPUT'method: (THISUNIT SLOT NEWVALUE OLDVALUE UNIT SLOTTYP
E)
(DECLARE (IGNORE SLOT OLDVALUE THISUNIT))
(SETQ X (GET.VALUE UNIT 'ACTUAL-TEST-VALUE))
(COND
  ((EQUAL X 'LOSTE ()) (SETQ Y 0.25))
  ((EQUAL X 'IDENTIFIABLE) (SETQ Y 0.5))
  ((EQUAL X 'MODERATE) (SETQ Y 0.75))
  ((EQUAL X 'MARKED) (SETQ Y 1))
  (T (SETQ Y 0)))
(PUT.VALUE UNIT 'EQUIVALENT-NUMERICAL-VALUE Y)
NEWVALUE)

(DEFUN FRD DISPLAY-VALUES::OUTPUT'method: (THISUNIT)
(COND
  ((EQUAL TEST 'CENTRAL-PROGRESSION)
   (SETQ XX 'VIEWPORT-ACTUAL-TEST-VALUE-OF-CENTRAL-PROGRESSION.5))
  ((EQUAL TEST 'HEMERALOPIA) (SETQ XX 'VIEWPORT-ACTUAL-TEST-VALUE-OF-HEMERALOPIA.7))
  ((EQUAL TEST 'FAMILY-TREE) (SETQ XX 'VIEWPORT-ACTUAL-TEST-VALUE-OF-FAMILY-TREE.6))
  ((EQUAL TEST 'NYCTALOPIA) (SETQ XX 'VIEWPORT-ACTUAL-TEST-VALUE-OF-NYCTALOPIA.8))

```

```

(EQUAL TEST 'NYSTAGMUS) (SETQ XX 'VIEWPORT-ACTUAL-TEST-VALUE-OF-NYSTAGMUS.9))
(EQUAL TEST 'PERIPHERAL-PROGRESSION)
  (SETQ XX 'VIEWPORT-ACTUAL-TEST-VALUE-OF-PERIPHERAL-PROGRESSION.10))
(EQUAL TEST '100-HUE) (SETQ XX 'VIEWPORT-ACTUAL-TEST-VALUE-OF-100-HUE.11))
(EQUAL TEST 'CONE-ERG) (SETQ XX 'VIEWPORT-ACTUAL-TEST-VALUE-OF-CONE-ERG.12))
(EQUAL TEST 'DARK-ADAPTATION) (SETQ XX 'VIEWPORT-ACTUAL-TEST-VALUE-OF-DARK-ADAPTATION.13))
(EQUAL TEST 'ROD-ERG) (SETQ XX 'VIEWPORT-ACTUAL-TEST-VALUE-OF-ROD-ERG.14))
(EQUAL TEST 'ACUITY) (SETQ XX 'VIEWPORT-ACTUAL-TEST-VALUE-OF-ACUITY.15))
(EQUAL TEST 'FUNDUS) (SETQ XX 'VIEWPORT-ACTUAL-TEST-VALUE-OF-FUNDUS.16))
(EQUAL TEST 'MEDIA) (SETQ XX 'VIEWPORT-ACTUAL-TEST-VALUE-OF-MEDIA.4))
(EQUAL TEST 'VISUAL-FIELDS) (SETQ XX 'VIEWPORT-ACTUAL-TEST-VALUE-OF-VISUAL-FIELDS.17)))
(UNITMSG XX 'OPEN-PANEL')
(UNITMSG 'VIEWPORT-DONE-OF-TESTS.2 'OPEN-PANEL')
(DO ((PRO T))
  ((NULL PRO) (UNITMSG XX 'CLOSE-PANEL'))
  (COND
    ((EQUAL (GET.VALUE 'TESTS 'DONE1) 'T) (PUT.VALUE 'TESTS 'DONE1 (QUOTE ()))
      (UNITMSG 'VIEWPORT-DONE-OF-TESTS.2 'CLOSE-PANEL!) (SETQ PRO (QUOTE ())))))
(COND
  ((EQUAL (GET.VALUE TEST 'ACTUAL-TEST-VALUE) 'UNKNOWN)
    (LOOP FOR X IN (UNIT.CHILDREN 'FRD-DISEASES 'MEMBER) DO
      (PUT.VALUE X 'NUMBER-OF-TESTS (- (GET.VALUE X 'NUMBER-OF-TESTS) 1)))
    (TERPRI) (TERPRI)
    (PRINT
      '(SINCE THE RESULT OF THE TEST ,TEST IS UNKNOWN I HAVE ASSUMED THAT YOU HAVE NOT PERFORMED
        THE TEST)))
    (T (TERPRI) (PRINT '(PLEASE INDICATE YOUR CONFIDENCE IN THE TEST VALUE)) (TERPRI)
      (PRINT '(THE CONFIDENCE FACTOR CAN BE A NUMBER BETWEEN 0 AND 1)) (SETQ Y (READ))
      (PUT.VALUE TEST 'CONFIDENCE-FACTOR Y) (UNITMSG 'ESTIMATE-PROBABILITY 'PROB))))

(DEFUN :FRD-PATIENT-CHECKER::CHECK:method: (THISUNIT)
  (SLASH EXISTS 'FALSE)
  (LOOP FOR X IN (UNIT.CHILDREN 'PATIENTS 'MEMBER) DO
    (COND
      ((EQUAL (UNIT.NAME X) NUMBER) (SETQ EXISTS 'TRUE)))
    (COND
      ((EQUAL EXISTS 'TRUE) (TERPRI)
        (PRINT '(THE PATIENT ,NAME WITH PATIENT NUMBER ,NUMBER WAS DIAGNOSED EARLIER BY ME))
        (TERPRI) (PRINT '(THE FOLLOWING TESTS WERE DONE ON THE PATIENT)) (TERPRI)
        (PRINT (GET.VALUE NUMBER 'TESTS-DONE))
        (PUT.VALUE 'CSNB 'PROBABILITY-OF-DISEASE (GET.VALUE NUMBER 'CSNB-PROB))
        (PUT.VALUE 'CSNB 'PROBABILITY (GET.VALUE NUMBER 'CSNB-TOTAL))
        (PUT.VALUE 'RC-DYST 'PROBABILITY-OF-DISEASE (GET.VALUE NUMBER 'RC-PROB))
        (PUT.VALUE 'RC-DYST 'PROBABILITY (GET.VALUE NUMBER 'RC-TOTAL))
        (PUT.VALUE 'CR-DYST 'PROBABILITY-OF-DISEASE (GET.VALUE NUMBER 'CR-PROB))
        (PUT.VALUE 'CR-DYST 'PROBABILITY (GET.VALUE NUMBER 'CR-TOTAL))
        (PUT.VALUE 'PSEUDO-RP 'PROBABILITY-OF-DISEASE (GET.VALUE NUMBER 'PSEUDO-RP-PROB))
        (PUT.VALUE 'PSEUDO-RP 'PROBABILITY (GET.VALUE NUMBER 'PSEUDO-RP-TOTAL))
        (LOOP FOR X IN (UNIT.CHILDREN 'FRD-DISEASES 'MEMBER) DO
          (PUT.VALUE X 'NUMBER-OF-TESTS (GET.VALUE NUMBER 'TOTAL-TESTS-DONE)))
        (T (CREATE.UNIT NUMBER) (ADD.PARENT NUMBER 'PATIENTS 'MEMBER)
          (PUT.VALUE NUMBER 'TESTS-DONE (QUOTE ())))
        (PUT.VALUE 'PATIENTS 'NAME NAME)
        (PUT.VALUE 'PATIENTS 'NUMBER NUMBER)
        (UNITMSG 'VIEWPORT-NAME-OF-PATIENTS.1 'OPEN-PANEL')
        (UNITMSG 'VIEWPORT-NUMBER-OF-PATIENTS.2 'OPEN-PANEL')
        (UNITMSG 'VIEWPORT-PROBABILITY-OF-DISEASE-OF-CR-DYST.3 'OPEN-PANEL')
        (UNITMSG 'VIEWPORT-PROBABILITY-OF-DISEASE-OF-CSNB.4 'OPEN-PANEL')
        (UNITMSG 'VIEWPORT-PROBABILITY-OF-DISEASE-OF-PSEUDO-RP.5 'OPEN-PANEL'))

```

```
(UNITMS6 'VIEWPORT-PROBABILITY-OF-DISEASE-OF-RC-DYST.6 'OPEN-PANEL!))
```

```
(DEFUN (FRD>INITIALISE::RESET'method: (THISUNIT) "
  (LOOP FOR X IN (UNIT.CHILDREN 'FRD-DISEASES 'MEMBER) DO
    (PUT.VALUE X 'PROBABILITY-OF-DISEASE 0) (PUT.VALUE X 'PROBABILITY 0)
    (PUT.VALUE X 'NUMBER-OF-TESTS 0))
  (PUT.VALUE 'DIAGNOSE-THE-PATIENT 'DONE1 (QUOTE ()))
  (PUT.VALUE 'TESTS 'DONE1 (QUOTE ()))
  (LOOP FOR X IN (UNIT.CHILDREN 'HISTORY 'MEMBER) DO (PUT.VALUE X 'DONE1 (QUOTE ())))
  (LOOP FOR X IN (UNIT.CHILDREN 'LAB-TESTS 'MEMBER) DO (PUT.VALUE X 'DONE1 (QUOTE ())))
  (LOOP FOR X IN (UNIT.CHILDREN 'PHYSICAL 'MEMBER) DO (PUT.VALUE X 'DONE1 (QUOTE ())))
```

```
(DEFUN (FRD>DONE1.ACTIVEVALUE::AVPUT'method: (THISUNIT SLOT OLDVALUE NEWVALUE UNIT SLOTTYPE)
  (DECLARE (IGNORE THISUNIT SLOT OLDVALUE))
  (PUT.VALUE UNIT 'DONE1 'T))
```

```
(DEFUN (FRD>RANK-DIAGNOSES::RANK-OUTPUT'method: (THISUNIT)
  (PRINT '(PATIENT NAME = ,NAME))
  (PRINT '(PATIENT NUMBER = ,NUMBER))
  (TERPRI)
  (PRINT
    "THE PATIENT WAS DIAGNOSED AS BELONGING TO THE FOLLOWING CLASSES OF DISEASES WITH THE
    FOLLOWING PROBABILITIES)).
  (SETQ A (LIST (GET.VALUE 'CSNB 'PROBABILITY-OF-DISEASE) 'CSNB))
  (SETQ B (LIST (GET.VALUE 'CR-DYST 'PROBABILITY-OF-DISEASE) 'CR-DYST))
  (SETQ C (LIST (GET.VALUE 'RC-DYST 'PROBABILITY-OF-DISEASE) 'RC-DYST))
  (SETQ D (LIST (GET.VALUE 'PSEUDO-RP 'PROBABILITY-OF-DISEASE) 'PSEUDO-RP))
  (SETQ E (LIST A B C D))
  (SETQ PROB
    (LIST (GET.VALUE 'CSNB 'PROBABILITY-OF-DISEASE)
          (GET.VALUE 'CR-DYST 'PROBABILITY-OF-DISEASE)
          (GET.VALUE 'RC-DYST 'PROBABILITY-OF-DISEASE)
          (GET.VALUE 'PSEUDO-RP 'PROBABILITY-OF-DISEASE)))
  (PRINT '(DISEASE & PROBABILITY))
  (TERPRI)
  (DO ((PRO T))
    ((NULL E) NIL)
    (SETQ HIGHEST (APPLY 'MAX PROB))
    (SETQ OUTPUT (ASSOC HIGHEST E))
    (PRINT '(, (CAR (CDR OUTPUT)) ,HIGHEST))
    (SETQ E (REMOVE OUTPUT E))
    (COND
      ((EQUAL (ASSOC HIGHEST E) (QUOTE ())) (SETQ PROB (REMOVE HIGHEST PROB))))
    (PUT.VALUE NUMBER 'CSNB-PROB (GET.VALUE 'CSNB 'PROBABILITY-OF-DISEASE))
    (PUT.VALUE NUMBER 'CSNB-TOTAL (GET.VALUE 'CSNB 'PROBABILITY))
    (PUT.VALUE NUMBER 'CR-PROB (GET.VALUE 'CR-DYST 'PROBABILITY-OF-DISEASE))
    (PUT.VALUE NUMBER 'CR-TOTAL (GET.VALUE 'CR-DYST 'PROBABILITY))
    (PUT.VALUE NUMBER 'RC-PROB (GET.VALUE 'RC-DYST 'PROBABILITY-OF-DISEASE))
    (PUT.VALUE NUMBER 'RC-TOTAL (GET.VALUE 'RC-DYST 'PROBABILITY))
    (PUT.VALUE NUMBER 'PSEUDO-RP-PROB (GET.VALUE 'PSEUDO-RP 'PROBABILITY-OF-DISEASE))
    (PUT.VALUE NUMBER 'PSEUDO-RP-TOTAL (GET.VALUE 'PSEUDO-RP 'PROBABILITY))
    (PUT.VALUE NUMBER 'NAME NAME)
    (PUT.VALUE NUMBER 'TOTAL-TESTS-DONE (GET.VALUE 'CSNB 'NUMBER-OF-TESTS))
    (SETQ TEMPOR (GET.VALUE NUMBER 'TESTS-DONE))
    (COND
      ((MEMBER TEST TEMPOR) NIL)
      (T (SETQ TEMPOR (CONS TEST TEMPOR))))
```



```
(PUT.VALUE NUMBER 'TESTS-DONE TEMPOR)
(PUT.VALUE NUMBER 'NUMBER NUMBER))
```

```
(DEFUN ESTIMATE-PROBABILITY::PROB' method: (THISUNIT)
  (PUT.VALUE 'CSNB 'PROBABILITY
    (+ (GET.VALUE 'CSNB 'PROBABILITY)
      (+ (GET.VALUE TEST 'CONFIDENCE-FACTOR) (GET.VALUE TEST 'CSNB-WEIGHT)
        (GET.VALUE TEST 'EQUIVALENT-NUMERICAL-VALUE))))
  (PUT.VALUE 'CSNB 'PROBABILITY-OF-DISEASE
    (/ (GET.VALUE 'CSNB 'PROBABILITY) (GET.VALUE 'CSNB 'NUMBER-OF-TESTS)))
  (PUT.VALUE 'CR-DYST 'PROBABILITY
    (+ (GET.VALUE 'CR-DYST 'PROBABILITY)
      (+ (GET.VALUE TEST 'CONFIDENCE-FACTOR) (GET.VALUE TEST 'CR-WEIGHT)
        (GET.VALUE TEST 'EQUIVALENT-NUMERICAL-VALUE))))
  (PUT.VALUE 'CR-DYST 'PROBABILITY-OF-DISEASE
    (/ (GET.VALUE 'CR-DYST 'PROBABILITY) (GET.VALUE 'CR-DYST 'NUMBER-OF-TESTS)))
  (PUT.VALUE 'RC-DYST 'PROBABILITY
    (+ (GET.VALUE 'RC-DYST 'PROBABILITY)
      (+ (GET.VALUE TEST 'CONFIDENCE-FACTOR) (GET.VALUE TEST 'RC-WEIGHT)
        (GET.VALUE TEST 'EQUIVALENT-NUMERICAL-VALUE))))
  (PUT.VALUE 'RC-DYST 'PROBABILITY-OF-DISEASE
    (/ (GET.VALUE 'RC-DYST 'PROBABILITY) (GET.VALUE 'RC-DYST 'NUMBER-OF-TESTS)))
  (PUT.VALUE 'PSEUDO-RP 'PROBABILITY
    (+ (GET.VALUE 'PSEUDO-RP 'PROBABILITY)
      (+ (GET.VALUE TEST 'CONFIDENCE-FACTOR) (GET.VALUE TEST 'PSEUDO-RP-WEIGHT)
        (GET.VALUE TEST 'EQUIVALENT-NUMERICAL-VALUE))))
  (PUT.VALUE 'PSEUDO-RP 'PROBABILITY-OF-DISEASE
    (/ (GET.VALUE 'PSEUDO-RP 'PROBABILITY) (GET.VALUE 'PSEUDO-RP 'NUMBER-OF-TESTS)))
  (PUT.VALUE TEST 'DONE2 (QUOTE ())))
```

```
(DEFUN FRD>PURSUE-HYPOTHESIS::FURTHER' method: (THISUNIT)
  (TERPRI)
  (PRINT '(WHAT IS THE NAME OF THE PATIENT YOU WANT ME TO DIAGNOSE ?))
  (SETQ NAME (READ))
  (TERPRI)
  (PRINT '(GIVE ME THE PATIENT NUMBER OF ,NAME))
  (SETQ NUMBER (READ))
  (UNTIL (PATIENT-CHECKER 'CHECK))
  (DO ((PROCEED T))
    ((NULL PROCEED) NIL)
    (TERPRI)
    (PRINT '(THE TESTS EMPLOYED IN THE DIAGNOSIS OF FRD DISEASES ARE AS SHOWN ABOVE))
    (TERPRI)
    (PRINT '(YOU CAN PERFORM OR REPEAT - ANY OF THOSE TESTS))
    (TERPRI)
    (PRINT '(PLEASE CHOOSE THE TEST TO BE DONE ON ,NAME))
    (PUT.VALUE 'TESTS 'HISTORY-TO-BE-DONE 'UNKNOWN)
    (PUT.VALUE 'TESTS 'LAB-TEST-TO-BE-DONE 'UNKNOWN)
    (PUT.VALUE 'TESTS 'PHYSICAL-TO-BE-DONE 'UNKNOWN)
    (UNITMSG 'VIEWPORT-HISTORY-TO-BE-DONE-OF-TESTS.1 'OPEN-PANEL')
    (UNITMSG 'VIEWPORT-PHYSICAL-TO-BE-DONE-OF-TESTS.3 'OPEN-PANEL')
    (UNITMSG 'VIEWPORT-LAB-TEST-TO-BE-DONE-OF-TESTS.2 'OPEN-PANEL')
    (UNITMSG 'VIEWPORT-DONE-OF-TESTS.2 'OPEN-PANEL')
    (DO ((IDLE T))
      ((NULL IDLE) (UNITMSG 'VIEWPORT-HISTORY-TO-BE-DONE-OF-TESTS.1 'CLOSE-PANEL')
        (UNITMSG 'VIEWPORT-PHYSICAL-TO-BE-DONE-OF-TESTS.3 'CLOSE-PANEL')
        (UNITMSG 'VIEWPORT-LAB-TEST-TO-BE-DONE-OF-TESTS.2 'CLOSE-PANEL!))
      (COND
```

```

((EQUAL (GET.VALUE 'TESTS 'DONE1) 'T) (SETQ IDLE ()))
  (UNITMSG 'VIEWPORT-DONE-OF-TESTS.2 'CLOSE-PANEL') (PUT.VALUE 'TESTS 'DONE1 (QUOTE ())))
))
(SETQ TEST (GET.VALUE 'TESTS 'HISTORY-TO-BE-DONE))
(COND
  ((EQUAL TEST 'UNKNOWN) (SETQ TEST (GET.VALUE 'TESTS 'PHYSICAL-TO-BE-DONE))))
(COND
  ((EQUAL TEST 'UNKNOWN) (SETQ TEST (GET.VALUE 'TESTS 'LAB-TEST-TO-BE-DONE))))
(LOOP FOR X IN (UNIT.CHILDREN 'FRD-DISEASES 'MEMBER) DO
  (PUT.VALUE X 'NUMBER-OF-TESTS (+ 1 (GET.VALUE X 'NUMBER-OF-TESTS))))
(TERPRI)
(PRI (THE VALUE OF THE TEST ,TEST CAN TAKE ON ANY OF THE VALUES SHOWN ABOVE))
(TERPRI)
(PRINT '(PLEASE CHOOSE A VALUE FOR THE TEST))
(TERPRI)
(UNITMSG 'DISPLAY-VALUES 'OUTPUT)
(UNITMSG 'RANK-DIAGNOSES 'RANK-OUTPUT)
(TERPRI)
(PRINT
  '(DO YOU
    WANT
    TO
    REFINER
    THE
    DIAGNOSES
    BY
    PERFORMING
    ANOTHER
    TEST
    OR
    REPEATING
    A
    TEST
    ))
(UNITMSG 'VIEWPORT-ANSWER-OF-DIAGNOSE-THE-PATIENT.22 'OPEN-PANEL')
(UNITMSG 'VIEWPORT-DONE-OF-DIAGNOSE-THE-PATIENT.1 'OPEN-PANEL')
(DO ((PRO T))
  ((NULL PRO) NIL)
  (COND
    ((EQUAL (GET.VALUE 'DIAGNOSE-THE-PATIENT 'DONE1) 'T)
      (PUT.VALUE 'DIAGNOSE-THE-PATIENT 'DONE1 (QUOTE ()))
      (UNITMSG 'VIEWPORT-ANSWER-OF-DIAGNOSE-THE-PATIENT.22 'CLOSE-PANEL')
      (UNITMSG 'VIEWPORT-DONE-OF-DIAGNOSE-THE-PATIENT.1 'CLOSE-PANEL') (SETQ PRO ())))))
(COND
  ((EQUAL (GET.VALUE 'DIAGNOSE-THE-PATIENT 'ANSWER) 'NO) (SETQ PROCEED (QUOTE ())))))

(DEFUN (FRD)DIAGNOSE-THE-PATIENT::DIAGNOSE'method: (THISUNIT)
  (PRINT '(WELCOME TO THE FRD KNOWLEDGE BASE SYSTEM))
  (UNITMSG 'INITIALISE 'RESET)
  (TERPRI)
  (DO ((PROGRESS T))
    ((NULL PROGRESS) (UNITMSG 'INITIALISE 'RESET))
    (UNITMSG 'VIEWPORT-PROBABILITY-OF-DISEASE-OF-CR-DYST.3 'CLOSE-PANEL')
    (UNITMSG 'VIEWPORT-PROBABILITY-OF-DISEASE-OF-CSNB.4 'CLOSE-PANEL')
    (UNITMSG 'VIEWPORT-PROBABILITY-OF-DISEASE-OF-PSEUDO-RP.5 'CLOSE-PANEL')
    (UNITMSG 'VIEWPORT-PROBABILITY-OF-DISEASE-OF-RC-DYST.6 'CLOSE-PANEL')
    (UNITMSG 'VIEWPORT-NAME-OF-PATIENTS.1 'CLOSE-PANEL')
    (UNITMSG 'VIEWPORT-NUMBER-OF-PATIENTS.2 'CLOSE-PANEL') (GOODBYE))
  (TERPRI)

```

```

(PRINT '(DO YOU
      WANT
      ME
      TO
      DIAGNOSE
      A
      PATIENT
      ?))
(UNITMS6 'VIEWPORT-ANSWER-OF-DIAGNOSE-THE-PATIENT.22 'OPEN-PANEL')
(UNITMS6 'VIEWPORT-DONE-OF-DIAGNOSE-THE-PATIENT.1 'OPEN-PANEL')
(DD ((PRO T))
    ((NULL PRO) NIL)
    (COND
      ((EQUAL (GET.VALUE UNIT 'DONE1) 'T) (PUT.VALUE UNIT 'DONE1 (QUOTE ()))
      (UNITMS6 'VIEWPORT-ANSWER-OF-DIAGNOSE-THE-PATIENT.22 'CLOSE-PANEL')
      (UNITMS6 'VIEWPORT-DONE-OF-DIAGNOSE-THE-PATIENT.1 'CLOSE-PANEL') (SETQ PRO ())))))
(COND
  ((EQUAL (GET.VALUE UNIT 'ANSWER) 'YES) (UNITMS6 'INITIALISE 'RESET)
  (UNITMS6 'PURSUE-HYPOTHESIS 'FURTHER))
  (T (SETQ PROGRESS ())))))

```

END

17.07.89

FIN