Sri Lanka Association for Artificial Intelligence (SLAAI)
Proceedings of first Annual Session
16th September 2003 - Colombo

Building your Medical Expert System- Do it yourself

D.A.I.P. Fernando
Teaching Hospital, Colombo North.
E-mail: irosh@sltnet.lk , Tel.:031-2231837, 0777-324609

A.S. Karunananda
Department of Mathematics and Computer Science
Open University, Nawala
E-mail: asoka@maths.ou.ac.lk, Tel.: 01-853777 ext 225

Abstract

Lack of appropriate expert system development tools, which are user friendly and which required less technical knowledge, have been a barrier for medical doctors themselves to develop expert systems for their own use. We have developed an expert system shell, which has a more user-friendly graphical user interface, simple inference engine and a rule based knowledge base, which represents the knowledge in a relational database. This tool was implemented using Microsoft Visual Basic and Microsoft Access. We have been able to demonstrate its usefulness by developing real world applications.

1. Introduction

Development of expert systems in medical domain has been initiated as far as thirty years back when MYCIN was designed for diagnosing and recommending the treatments for meningitis and bacterially infected blood diseases [1]. There have been programming languages too such as *Prolog* and *LISP* for developing expert systems for many years. But development and use of expert systems in the medical domain has not been grown up to an expected extent, though a promising growth was partially evident. Medical professionals are not having necessary technical knowledge on expert systems and Computer professionals are not having knowledge on medical specialties to develop knowledge bases may be the main barriers. Most of the software tools for developing expert systems are basically not very user friendly for medical professionals, as use of them require significant amount of knowledge in expert system technologies and programming.

Existing popular expert system programming environments such as *Prolog* has an inbuilt inference engine with backward chaining search process [2]. In a medical diagnosis process, as the patient complaints the symptoms, then only the doctor thinks of several possible diagnoses and start excluding them. During this process the searching space for a diagnosis becomes narrower. This process is compatible with forward-chaining searching algorithms.

We believe that if there are tools, which help to develop knowledge bases in a user-friendly manner, with built in inference engines, eliminating the need for programming, will enhance the development and use of expert systems in medical domain, by medical professionals themselves. The aim of our research was to develop such expert system shell that can be used by medical doctors without having a knowledge in expert systems.

2. How medical diagnosis works

During a medical consultation the diagnosis is made considering patient's symptoms. There must be a certain combination of these symptoms, if a particular diagnosis is to be made. When a patient tells his symptoms, in the doctors mind it arises several possible diagnoses. This is because, one or more of the symptoms the patient stated, are associated with one or more diagnoses. So the doctor needs to find out whether other symptoms, which are associated with each of those possible diagnoses, are present, by asking them from the patient. During this process when new symptoms are discovered, new possible diagnosis will arise and some of the initial possible diagnoses will be excluded when patient dose not indicate posible symptoms. This process is continued until all the possible diagnoses, which arose during this process, are tested. At the end, a diagnosis may or may not be made, and further requires physical examination of the patient and laboratory investigations, in the pursuit of a diagnosis for confirming the diagnoses, which have been already made.

3. Design and Implementation

From the description of the medical diagnosis process, we could identify some key components of an expert system required for medical diagnosis. They are basically a set of rules used for diagnosis, which represents the expert knowledge, inference engine and a human computer interface.

3.1 Diagnostic Rules

The expert knowledge is represented as rules. It is described in the literature that relational databases can be used for representing expert knowledge [4]. In fact, it has shown that a medical knowledge base can be implemented in terms of a relational database [5]. We have used a relational data base model for developing a rule base, which is required for making diagnosis. A rule consist of two components namely, a diagnosis and a set of symptoms associated with it, to confirm the diagnosis. This format is more compatible with the diagnosis criteria given for various diseases in international classification of diseases[6]. Figure-1, explains the relational data base model for rule base.

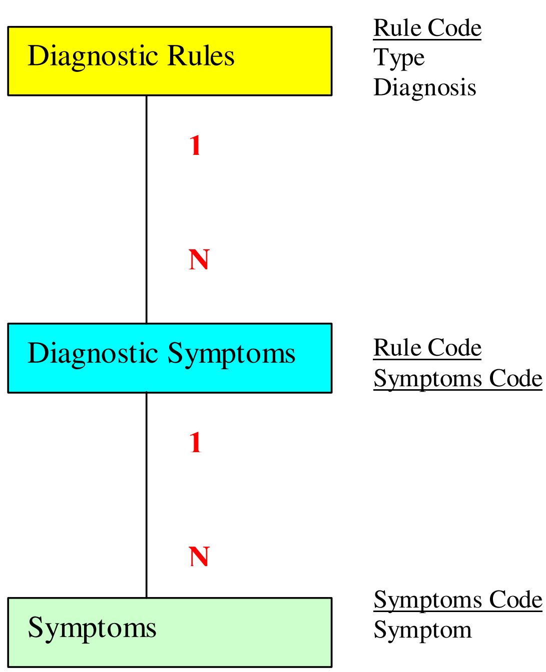

Figure-1. Relational Model of the Knowledge base

3.2 Inference Engine

We developed the inference engine considering the process of arriving at diagnoses by a doctor as described earlier. It is a binary logic inference engine, which has three main components, which are Symptoms eliciting module (SEM), Dynamic table of symptoms (DTS), and Diagnoses making module (DMM). Figure-2 shows these components.

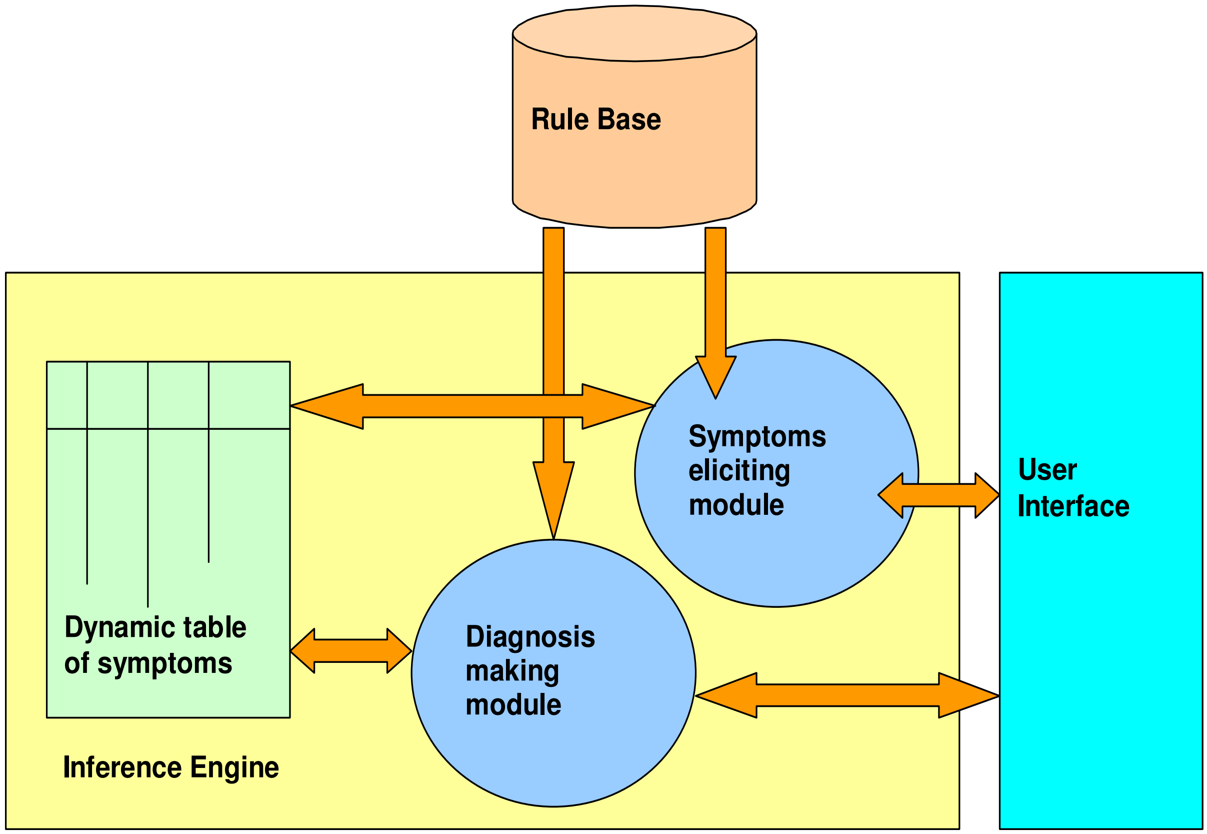

Figure -2. Components of the Inference Engine

Inference engine works as follows. DTS is a database table, which stores a set of screening symptoms to start with. They are supposed to be found common to most of the illnesses. DTF table has three fields the *Symptom code*, *Elicited* and *Has*. *Symptom code* indicates the code number of the symptom given in the diagnostic rules. *Elicited* and *Has* are Boolean fields which indicated whether the symptom is elicited or not and whether the patient has this symptom or not respectively. The symptoms which were found in the patient is marked in the *Has* field as true and otherwise false.
It then present these symptom, which has not been elicited to the user via human computer interface. These features are identified as they are indicated in the DTF, with the *Elicited* field having the value false.

Then it updates the records in the DTS, by *Elicited* filed with value true, and *Has* field as either true or false according to the user response. If a patient is having a particular feature, *Has* field is updated with the value true otherwise false.

Next it checks every feature in the DTS, with the Has field having the value true, against each and every rule. If a rule is having any symptoms that were not included in the DTS, they are added and the field *Elicited* is marked as false. This process will be repeated

Finally, DMM matches the symptoms, which have been found in a patient, against the diagnosis rules to determine the diagnosis.

3.3. Human Computer Interface

This was designed as a graphical user interface, which has several components. Symptoms to be elicited are presented as a list, which can be marked with a tick, with a mouse click, if the patient is having that symptom. Once the consultation process is finished, with a button click, diagnoses made with the reasons are presented as a report. There is a module to maintain all the symptom and to formulate the diagnosis rules. To formulate diagnosis rules, user can type a diagnosis and then select the relevant symptom that are expected to be found, in a patient with this diagnosis, from a list box.

The entire expert system shell was programmed using Microsoft Visual Basic and the database was implemented using Microsoft Access. Figure-3 and 4, show the screen shots of the system in the process of arriving at a diagnosis and creating rules in to the knowledge base respectively.

Figure 3. A diagnosis process

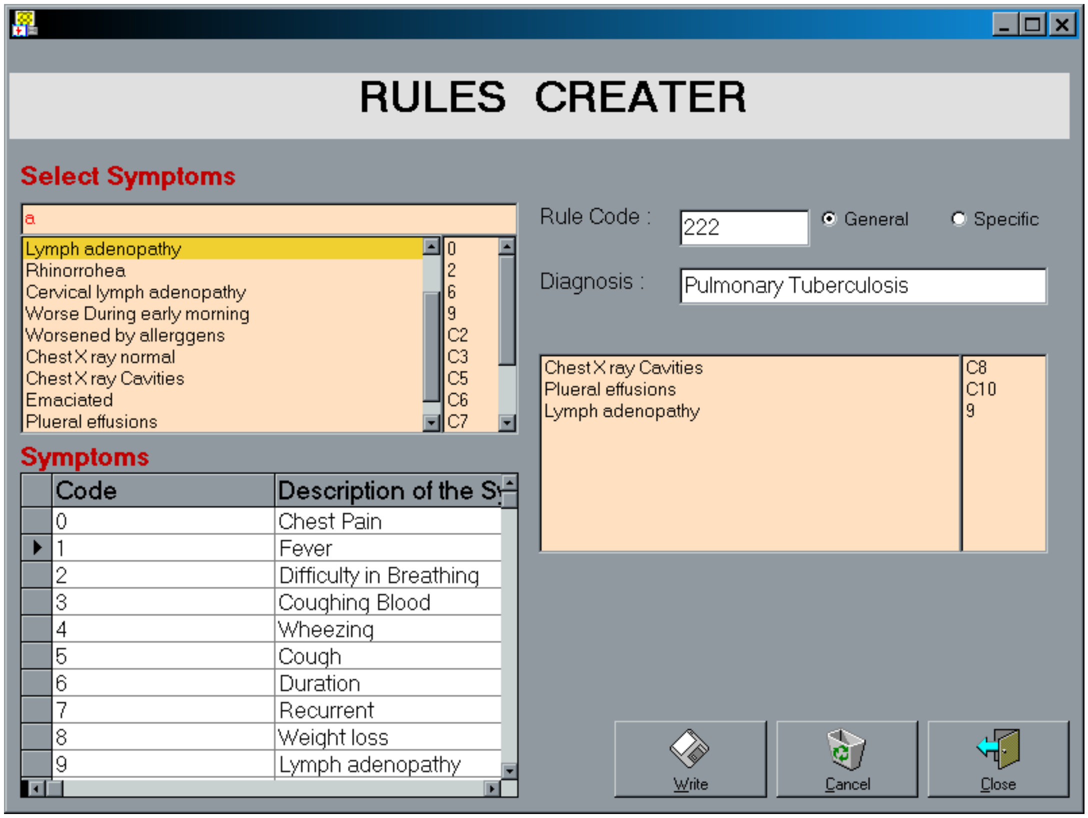

Figure -4. Process of creating rules

4. Conclusion.

We have developed an expert system shell for developing application mainly in the medical domain. It has been more user friendly compared with the existing expert system shell software. We have successfully used this tool to develop an expert system in the field of psychiatry [7]. There are certain medical sub specialties, where the process of arriving at a diagnosis is dealt with uncertainty, as the most probable diagnosis would be made. As a further improvement, an alternative inference engine can be developed using a fuzzy logic inference engine or incorporating evidential reasoning [4].

5. References

[1] Buchanan,B. G., Shortliffe, E.H, 1984, *Rule-Based Expert Systems: The MYCIN Experiments Of The Stanford Heuristic Programming Project*. Reading, MA:Addision-Wesley.

[2] Townsend, C., 1987, Introduction to Turbo Prolog, SYBEX inc.

[4]. Donna L. Hudson, Maurice E. Cohen, 2000, *Neural Networks And Artificial Intelligence For Biomedical Engineering.*, IEEE Press.

[5] Marrs,K.A.,Steib,S.A.,Abrams,C.A, and Kahn,M.G.,1993,Unifying heterogeneous distributed clinical data in a relational database. Proceedings, SCAMC17:644-648.

[6] International Classification of Diseases(ICD-10), World Health Organization.

[7] Fernando,D.A.I.P, 2003, *An Expert System For Diagnosisng Psychiatric Illnesses*. To be presented at 59th Sri Lanka Association for the Advancement of Sciences (SLAAS)annual sessions.