Tikrit Journal of Pure Science 23 (10) 2018 *ISSN: 1813 – 1662 (Print*
E-ISSN: 2415 – 1726 (On Line)

TJPS

TIKRIT JOURNAL OF PURE SCIENCE

Journal Homepage: http://main.tu-jo.com/ojs/index.php/TJPS/index

Construction and Implementation of an Expert System for Medical Diagnosis Based on Blood Test

Ibraheem Nadher Ibraheem

Computer science Department , Collage of Basic Education , University of Mustansiriyah ,

ARTICLE INFO.

Article history:

-Received: 12 / 9 / 2013

-Accepted: 25 / 11 / 2013

-Available online: / / 2018

Keywords: ES, blood test, medical test, HEM, SRM, HEM, DPM .

Corresponding Author:

Name: Ibraheem Nadher

E-mail:

Ibraheemnadher@yahoo.com

Tel:

Abstract

The recent developments in the field of artificial intelligence to the emergence of expert systems, and computational tools designed to capture and access to knowledge from experts in the medical field. We have built an expert system for the purpose of giving the diagnosis of some diseases by linking this system devise testing and analysis of blood type (CELL-DYN Ruby) and which has a private entrance to the computer. The system is divided into several modules, History Entering Module (HEM) systematic Review Module (SRM), a Laboratory Test List Module, Diagnosis Process Module (DPM), Data Base (DB), and finally the graphical interface to the user.

All these modules work together to give a high degree of accuracy in the diagnosis of disease through blood samples and we will describe the work of each unit in detail.

1.Introduction

Expert systems are defined as a set of programs that manipulate encoded knowledge to solve problems in a specialized domain that normally requires human expertise. In medical field, expert system knowledge is obtained from expert human sources and coded in a form suitable for the system to use in its inference or reasoning process [1].

Although the idea is quite simple, the mechanism of making an expert system functioning is difficult. To make the system function, there must be a knowledge base that is supplied by an expert. Every step of the process must be programmed meticulously, including all the options to decisions made throughout the process. This computerized knowledge base is designed to be updated periodically to include new rules and facts [2]. Several illnesses may produce similar symptoms, in the earlier stages at least. And some illnesses do not produce all the symptoms in the early stages .Therefore, when working backwards from all the symptoms, a doctor must consider several possibilities. It may be, of course, that some of these possibilities are more likely than others, such that the result of diagnosis could be a list of possible illness, with some indication of how likely each is[3].

If thought about it, all the people would like to find out how they stand on important health issues such as diabetes, cholesterol, thyroid problems, arthritic factors or other health related conditions which could adversely affect our lives. All of these can be defined by a blood test. After testing, when you get your blood test results back, you will definitely have questions. The first question that comes to mind is; "What does this mean to me?". Interpretation of your blood test results should be carefully reviewed with your physician, or a licensed nutritionist aware of the specific blood tests and familiar with your condition, medication and needs [4].

The expert system get the blood test results from the CELL-DYN Ruby, which is a new automated hematology analyzer used in AL-Yarmuk hospital laboratory. The CELL-DYN Ruby connected with computer system working under windows operating system and the results printed as text file by C++. The system reads the text file as input. The diagnostic system will give the results with probability depending on the medical history and the computerized blood test results.

The programming language used to build medical expert system is Visual Basic (VB) version 6.0, which is a suitable environment to design applications and user interfaces for this research.

There is no doubt that any developed method or system that tries to solve any kind of problem, must be tested and evaluated in a real life to measure its suitability for use with problems.

Thus the research has been done in two stages. First, diagnosis system in its earlier development was tested on a number of problems taken from literature. Second, the system was applied at Al-Yarmuk Hospital Laboratory.

2. Architecture of Typical Expert System

A typical expert system has four major components (see figure 1) [5]:

1. knowledge acquisition
2. The knowledge base
3. The inference engine
4. The explanatory interface

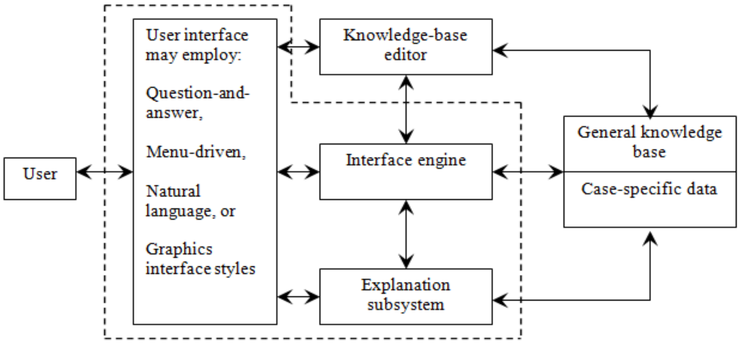

Figure (1) Architecture of a typical expert system.

2.1. Knowledge Acquisition

Knowledge acquisition is the process of adding new knowledge to a knowledge base and refining or otherwise improving knowledge that was previously acquired. Acquisition is usually associated with some purpose such as expanding the capabilities of a system or improving its performance at some specified task.

Acquired knowledge may consist of facts, rules, concepts, procedures, heuristics, formulas, relationships, statistics, or other useful information. Sources of this knowledge may include one or more of the following [1]: -

- Experts in domain of interest
- Textbooks
- Technical papers
- Databases
- Reports the environment

2.2. Database

An important part of an expert system is the database. It is sometimes called a global database because it contains a broad range of information about the current status of the problem being solved. In practice, the database is really a portion of working memory where the current status of the problem-solving process is stored.

The database is also referred to as the fact base because it records facts about the problem. Known facts are stored there initially. Then new facts, as they are gleaned from the inference process, are added. The fact base keeps track of all that is known during the inferencing operations.

Among the important things stored in the database are the initial conditions of the problem to be solved. Usually, the expert system asks the user for some beginning input. It may ask questions to which answers must be typed in, or it may present a menu of options from which the user must choose. This information gives the expert system a starting point to begin the search process.

The inference engine begins its search, matching the rules in the knowledge base against the information in the database. As each rule is examined, actions taken when a rule fires and may change the content of the database, thereby updating the status of the problem. New facts become available to use in the decision-making process. In addition, special functions such as a request for additional information from the user may be triggered.

The database also stores a list of rules that have been examined, fired, and in what sequence. This helps to keep track of the process. The rule sequence can be given later if the user requires an explanation of the reasoning process [6].

2.3. Knowledge Base

The heart of any expert system is its knowledge base[6]. The real power of expert system comes from the knowledge it possesses rather than the particular inference schemes and other formalisms it employs [1].

The knowledge base consists of information structures for coding expertise. Usually this is elicited from a human specialist and reformulated as a collection of rules, a network of facts or frame –based structure. A knowledge base differs from a database in several ways: in particular, it is more active. That is, it contains rules for deducing facts that are not stored explicitly[5].

The knowledge is stored in a knowledge base separate from the control and inferencing components. This makes it possible to add new knowledge or refine existing knowledge without recompiling the control and inferencing programs [1].

Domain knowledge typically has many forms, including descriptive definitions of the domain-specific terms, descriptions of individual objects, classes of objects and their interrelationships, and criteria for making decisions [8].

There are many different methods for representing knowledge in AI software: -

A. Logical Representation Schemes

This class of representations uses expressions in formal logic to represent a knowledge base. Inference rules and proof procedures apply this knowledge to problem instances. Although propositional logic is a knowledge representation alternative, it is not very useful in artificial intelligence. Since propositional logic deals primarily with complete statements and whether they are true or false, its ability to represent real world knowledge is limited. Consequently, AI uses predicate calculus instead. First-order predicate calculus is the most widely used logical representation scheme, but it is only one of a number of logical representations. PROLOG is an ideal programming language for implementing logical representation schemes.

In predicate calculus, a proposition or premise is divided into two parts, the arguments (or objects) and the predicate (and assertion). The arguments are the individuals or objects an assertion is made about. The predicate is the assertion made about them. In a common English language sentence, objects and individuals are nouns that serve as subject and object of the sentence. In a sentence the predicate would be the verb or part of a verb. The two are combined to create a proposition [6,7]:

PREDICATE (individual [object] 1, individual [object] 2)

For example, the proposition:

The car is in the garage

Would be stated as follows:

IN (car, garage)

In = products (assertion)

Car = argument (object)

Garage = argument (object)

B. Procedural Representation Schemes

Procedural schemes represent knowledge as a set of instructions for solving a problem. This contrasts with the *d*eclarative representations provided by logic and semantic network. In a rule-based system rules can be considered a subset of predicate logic. They have become a popular representation scheme for expert systems (also called rule-based systems).

Rules have two component parts: a left-hand side (LHS) referred to as antecedent, premise, condition, or situation, and a right-hand side (RHS) known as the consequent, conclusion, action, or response. The LHS is also known as the IF part and the RHS as the then part of the rule. Some rules also include another part (Else part). Examples of rules, which might be used in expert systems, are given below [1, 7].

IF: the temperature is greater than 95 degrees C,

THEN: open the relief valve.

IF: the lights do not come on,

And the engine does not turn over.

THEN: the battery is dead or the cable is loose.

C. Network Representation Schemes

Network representations capture knowledge as a graph in which the nodes represent objects or concepts in the problem domain and the arcs represent relations or associations between them. Networks also provide a more natural way to map to and from natural language than do other representation schemes. Network representations give a pictorial presentation of objects, their attributes and the relationships that exist between them and other entities. Examples of a network representation include semantic networks, and conceptual graphs.

Semantic networks are directed graphs with labeled nodes and arcs or arrows. In Figure 2, a class of objects known as Bird is depicted. The class has some properties and a specific member of the class named Tweety is shown. The color of Tweety is seen to be yellow.

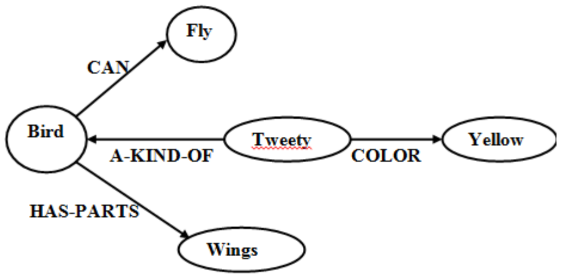

Figure (2) semantic networks example

A conceptual graph is a graphical portrayal of a mental perception,

Which consists of basic or primitive concepts and the relationships that exist between the concepts. A single conceptual graph is roughly equivalent to a graphical diagram of a natural language sentence where the words are depicted as concepts and relationships. An example of such a graph, which represents the sentence “Joe is eating soup with a spoon”, is depicted in Figure 3.

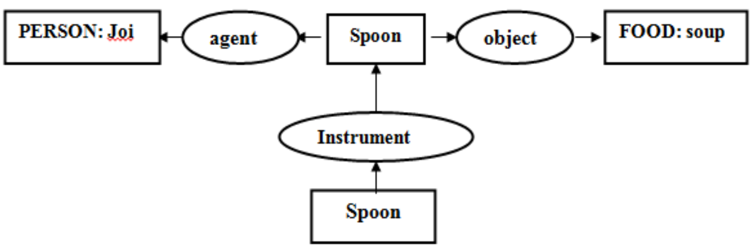

Figure (3) A conceptual graph.

In Figure 3, concepts are enclosed in boxes and relations between the concepts are enclosed in ovals. The direction of the arrow corresponds to the order of the arguments in the relation they connect. The last or nth arc (argument) points away from the circle relation and all other arcs point toward the relation [1,7].

D. Structured Representation Schemes

Structured representation languages extend networks by allowing each node to be a complex data structure consisting of named slots with attached values. These values may be simple numeric or symbolic data, pointers to other frames, or even procedures for performing a particular task. Examples of structured representations include scripts and frames.

A frame is relatively large block or chunk of knowledge about a particular object, event, location, situation, or other element. The frame describes that object in great detail. The detail is given in the form of slots which describe the various attributes and characteristics of the object or situation.

Figure 4 shows how a hotel bed may be represented using frame approaches. A hotel bed is a specialization of the general bed object. Many of the values are default assumptions about hotel beds. These include the size and firmness relation, specific instances of hotel beds may or may not inherit these values. The slot values may be values: pointers to other frames as in the case of the mattress, or even attached procedures for performing some function, such as getting the bed made [7].

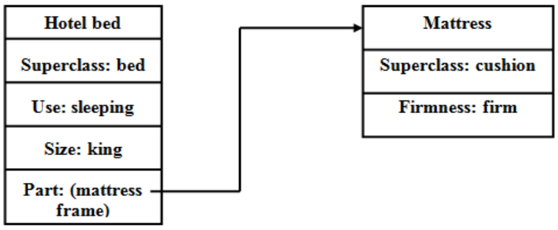

Figure (4) frame descriptions of a hotel bed

A script is a knowledge representation scheme similar to a frame, but instead of describing an object, the script describes a sequence of events. Like the frame, the script portrays a stereotyped situation. Unlike the frame, it is usually presented in a particular context. To describe a sequence of events, the script uses a series of slots containing information about the people, objects, and actions that are involved in the events [6].

E. Selection of Knowledge Representation Scheme

The key to the success of any AI program is the selection of a knowledge representation scheme that best fits the domain knowledge and the problem to be solved. That choice is best left to the knowledge engineer with considerable AI software design experience. In fact an expert system is a production system and the term can be used interchangeably. An expert system uses rules to encode the knowledge of an expert in a particular domain. This knowledge is then converted into a computer program. The resulting expert system software allows anyone with access to the computer to solve problems and make decisions comparable to those achievable by an expert system [6].

We might want to ask what are the pros and cons of using rules versus something else to set up the expert system [9].

There are at least four reasons why rule-based architectures are used. Of one sort or another, are so widely used [10]:

1. The individual rules frequently mirror the way experts' frame their own heuristics to solve problems. This is the way the expert system will talk to you in describing a problem.

2. Rules are natural modules. Like any modular structure, they promote easy modification and/or additions to the system.

3. It is particularly easy to do automatic reasoning (i.e.. to make the computer reason) if the underling structure is one of rules. In fact, at the present practitioners of artificial intelligence technologies do not know any other way to do this.

4. If a program is rule based, it is easy to make the computer give at coherent account of its actions. As noted previously, when the computer asks for information, the user should be able to ask, "Why do you want it?" The computer can construct an answer by showing, in order, the rules with which it was most recently working. When the computer states a conclusion. The user should be able to ask, "How do you know that?" again, in a rule-based system it is relatively easy to construct a pertinent answer by tracing the rules used to support that conclusion.

Rule- base systems, in practice, do not only use production rules to represent knowledge. There is in addition something called the 'working memory', which uses some other formalism entirely. There may be other kinds of structure too. For example, the Prospector system (Gashnig, 1982) is a rule-based expert system which has semantic network in the background to supply geological knowledge. So most real expert systems that employ production rules also employ alternative representation in the background very often in the form of semantic nets and/or frames [5].

3.The System Architecture:

The architecture of diagnosis system consists of the following modules and stages, as shown in figure (5):

- History entering module (HEM): This module is for entering information about the patient history, family history, surgical history and drug history. Part of this data must be entered by the patient, name, age and gender and the other part of data is entered by choosing yes or no from a combo box.
- Systematic review module (SRM): This module consists of seven tracts, each one contains a number of symptoms .The patient must choose yes if he has the symptom or no if he does not have it.

- Laboratory test list module (LTLM): In this module the system must give all the laboratory results from the CELL-DYN Ruby is a new automated hematology analyzer suitable for routine use in laboratories. This equipment used in Yarmuk Hospital Laboratory.
- Diagnosis process module (DPM): This module represents the inference engine part in the expert system.
- Data base (DB): This is a data base which contains information about the symptoms of each illness and the laboratory test which (affects) the diagnosis of this illness.
- User interface (UI): This is an interactive interface between the user (patient) and the system to have the best utilization of the system.

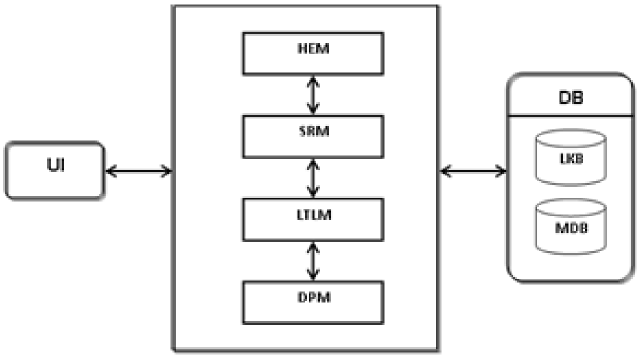

Figure (5) system architecture

The knowledge is represented as rules, which are called rule-based system. It consists of rules in the form of condition-action pairs. The program flowchart is shown in Figure (6).

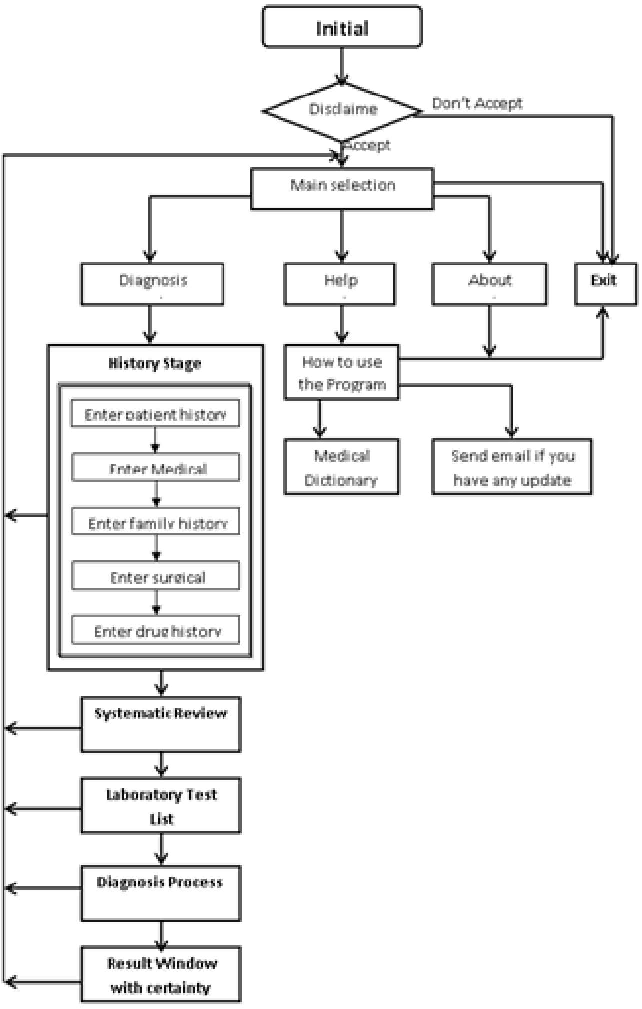

Figure (6) The Top down of medical diagnosis system

3.1. History entering module(HEM):
This module consists of five lists each one contains a number of queries; each query affects in some way the final diagnostic result. Also it may affect the type of information and the laboratory test results. Figure (7) show history entering window.

Figure (7) history entering window (HEM)

3.2. Systematic review module(SRM):
This module contains seven tracts each tract consists of list of symptoms the patient may have, if he has any illness in this tract. The patient must choose Yes from the combo box if he has this symptom or No if he does not. In our program the No is set as the default choice. Each symptom has an effect on the final result of the diagnosis system because the program gives us the illness which the patient has with certainty rate, this rate is calculated from a number of related symptoms and laboratory test results. figure (8) show systematic review window.

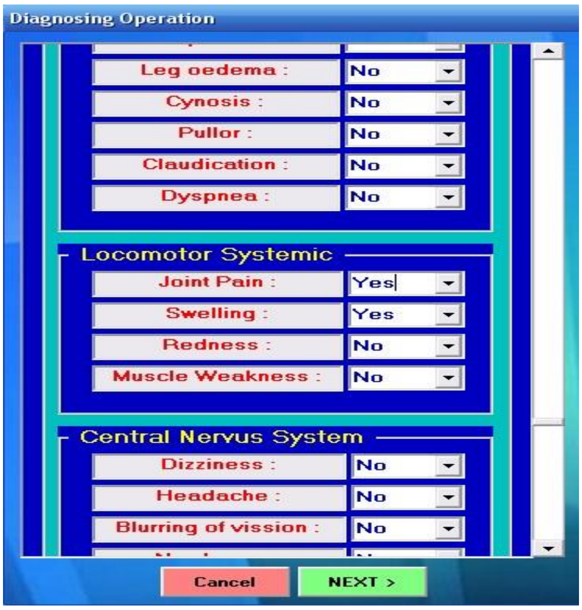

Figure (8) systematic review window (SRW)

3.3. Laboratory test list module:
Laboratory test list represents a set of tests the patient made in the laboratory and then the program will gave the results from the CELL – DYN ruby system during the diagnosis process.
This stage contains twenty eight items each one is related to one of the tracts which was checked in the previous stage. If the patient test results are abnormal the box color will change and the question box will be enabled. If the patient wants to know why his results are abnormal he can click on the questions box to see the reasons. Figure (9) show laboratory test list window

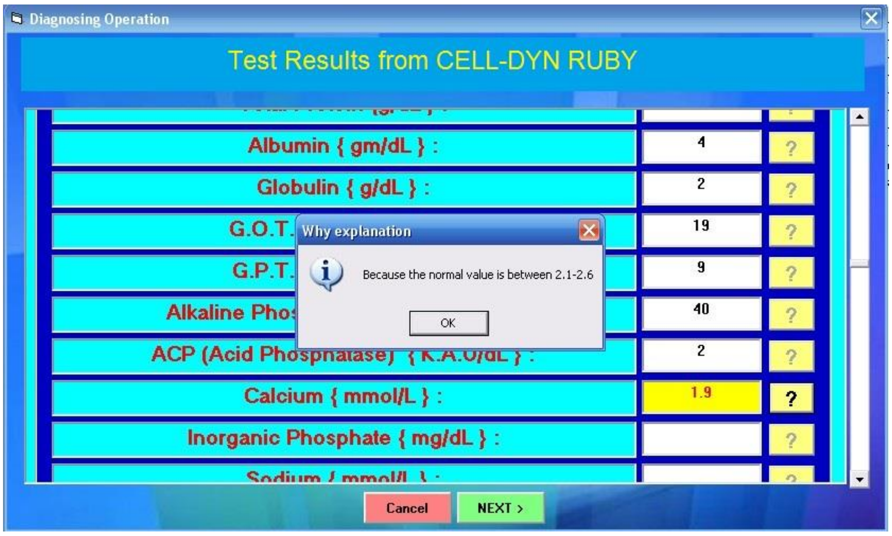

Figure (9) laboratory test list window

3.4. Diagnosis process module:

This part in our program represents the inference engine part in the expert system; the inference engine implements a search and pattern-matching operation. The rules are examined in a particular sequence looking for matching between the initial and current conditions given in the data base. As rules matching of these conditions are fired. (the specified actions are initiated).

The inference engine can take two basic approaches to search for an answer. These are forward and backward chaining. The first one begins with some initial information and works forward, attempting to match that information with a rule. This is called forward chaining. In forward chaining, the rule interpreter attempts to match a fact or a statement in the data base to the situation stated in the left-hand or IF part of the rule.

We use the forward chaining in our program by examining the IF part of the rules which represent a set of symptoms and test results related to each illness if they match the facts or statements in the data base. Then the patient may have this illness with a certainty calculated in the previous stages. Figure (10) show results display window.

Figure (10) results display window

Thus the practical part of the research has been done in two stages. Firstly, our diagnosis system IN its earlier development was tested on a number of problems taken from literature. Secondly, the system has been applied at Al-Zahra'a Hospital Laboratory. When we use the system and compare the system results with the doctor's diagnosis, we found 80% matching and these results will be more accurate depending on the data base size (amount of information).

4. Conclusions and Future Work

4.1. Conclusions:

Through the research work, the following conclusions can be drawn.

1- Expert system for disease diagnosis is very useful, because any one can use it easily and can get good result.

2- Medical diagnosis that depends on blood tests is very complex because the disease costs that are related to blood or else depends on it are too much.

3- Because some of cases in medical diagnosis have ambiguous states, we use the uncertainty factors in the system to preview probability results.

4- Because, the results of the system are probability numbers, it's a good way to display these results as bar graph and percentage of probabilities.

5- Visual Basic 6.0 is a good developer tool for software; it's easy to use and has a rich command and interface tool to build good and neat interface.

In this paper, the results of the system are probability numbers; it's a good way to display these results as bar graph and percentage of probabilities.

Some diseases need an image or x-ray. It is useful to make expert system include movie and sound and image.

4.2. Future Works:

For future it's good way to build general expert system shell. To use any knowledge that makes this system work in different domains. To improve system, we can add printed report option that makes system print a result as a report on printer.

References

[1] Patterson, D.W.(1998). "introduction to Artificial intelligence & expert system", New Delhi: prentice-Hall.

[2] Heizer, (1996) "*Production and Operations Management*", New Jersey: prentice-Hall .

[3] Charles C.C. and Okafor E.C.(2010), "Framework for eliciting knowledge for a medical laboratory diagnostic expert system", *ELSEVIER, Expert Systems with Applications* 37 (2010) 5009–5016.

[4] Jonathan J. Ross, Mouloud A. Denaı, Mahdi Mahfouf, (2009), "A hybrid hierarchical decision support system for cardiac surgical intensive care patients. Part II. Clinical implementation and evaluation", *ELSEVIER, Artificial Intelligence in Medicine* 45(1):53—62

[5] Forsyth R.(1989) "Expert systems, principles and case studies", London: Chapman.

[6] Frenzel, L.E.(1987) "Crash Course in Artificial intelligence & expert systems", United States of America: First edition.

[7] Luger G.F., (2009) "Artificial intelligence", London: Addison Wesely .

[8] Kusick, (1990). "Intelligent Manufacturing System".

[9] Bratko I. (2011), "Prolog Programming For Artificial Intelligence", Tech.LIB, 1986.

[10] Marcellus (1989), "Expert system and advanced data processing", Prentice-Hall.

بناء وتنفيذ نظام خبير للتحليل الطبي بالاعتماد على اختبار الدم

ابراهيم نضير ابراهيم

قسم الحاسبات ، كلية التربية الاساسية ، الجامعة المستنصرية ، بغداد ، العراق

الملخص

أدت التطورات الحديثة في مجال الذكاء الاصطناعي الى ظهور النظم الخبيرة, وادوات حسابية مصممة لالتقاط واتاحة المعرفة من الخبراء في الحقل الطبي. قمنا ببناء خبير لغرض اعطاء تشخيص لبعض الامراض عن طريق ربط هذا النظام بجهاز فحص وتحليل الدم نوع (CELL-DYN Ruby) والذي يحتوي على مدخل خاص للكومبيوتر. يقسم النظام الى عدة وحدات اهمها وحدة ادخال التاريخ الطبي للمريض, وحدة المراجعة المنهجية, وحدة قائمة الاختبار, وحدة عملية التشخيص, قاعدة بيانات واخيرا الواجهة الرسومية للمستخدم.

كل هذه الوحدات تعمل سوية لاعطاء درجة عالية من الدقة في تشخيص المرض عن طريق عينات الدم وسنقوم بشرح عمل كل وحدة بالتفصيل