


ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

SCHOOL OF INFORMATION SCIENCE

# DEVELOPING AN EXPERT SYSTEM FOR COMPUTER AND NETWORK TROUBLESHOOTING

AMANUEL AYDE ERGADO

June 2014

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

SCHOOL OF INFORMATION SCIENCE

# DEVELOPING AN EXPERT SYSTEM FOR COMPUTER AND NETWORK TROUBLESHOOTING

A Thesis Submitted to the School of Graduate Studies of Addis Ababa University in Partial Fulfillment of the Requirements for the Degree of Master of Science in Information Science

BY: AMANUEL AYDE ERGADO

June 2014

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

SCHOOL OF INFORMATION SCIENCE

# DEVELOPING AN EXPERT SYSTEM FOR COMPUTER AND NETWORK TROUBLESHOOTING

BY: AMANUEL AYDE ERGADO

Name of Member of the Examiner Board

| Name | Title | Signature | Date |
|---|---|---|---|
| ---------- | Chairperson | ---------- | ---------- |
| ---------- | Advisor | ---------- | ---------- |
| ---------- | Examiner | ---------- | ---------- |
| ---------- | Examiner | ---------- | ---------- |

## DEDICATION

This work is dedicated to the Lord of Lords, King of Kings, and the merciful one Jesus Christ, for being my strength when I get weak; my inspiration when I lose passion and my joy when nothing makes sense throughout my journey, lastly, this thesis is dedicated to my mother who died when I was 2 years old.

## ACKNOWLEDGEMENT

First and foremost, I would like to give a special gratitude to the Gracious God who provided me everything to finish my courses and this thesis work.

I gratefully acknowledge my advisor, Ato Workshet Lamenew for his commitments and patience for reading every section of the thesis, his valuable comments, encouragement and guidance from initial to the final level of the research that enabled me to finish my research work.

I would like to acknowledge Jimma University ICT office for providing important information for my thesis work and allowing the ICT staff members to participate in the evaluation process after implementing the expert system in their computer.

Finally, I would like to thank all my friends for their moral support and encouragements throughout my study.

# TABLE OF CONTENTS


DEDICATION ..... i

ACKNOWLEDGEMENT ..... ii

LIST OF TABLES ..... vi

LIST OF FIGURES ..... vii

ACRONYMS ..... viii

ABSTRACT ..... ix

CHAPTER ONE ..... 1

INTRODUCTION ..... 1

1.1 BACKGROUND ..... 1

1.2 STATEMENT OF THE PROBLEM ..... 3

1.3 OBJECTIVES OF THE STUDY ..... 5

1.3.1 General Objective ..... 5

1.3.2 Specific Objectives ..... 5

1.4 SCOPE OF THE STUDY ..... 6

1.5 SIGNIFICANCE OF THE STUDY ..... 6

1.6 METHODLOGY ..... 7

1.6.1 Literature Review ..... 7

1.6.2 Data Collection ..... 7

1.6.2.1 Knowledge Acquisition and Representation ..... 8

1.6.3 Implementation Tool ..... 8

1.6.4 Evaluation ..... 9

1.7 ORGANIZATION OF THE THESIS ..... 9

CHAPTER TWO ..... 11

LITERATURE REVIEW ..... 11

2.1. ARTIFICIAL INTELLIGENCE ..... 11

2.1.2 Expert System ..... 11

2.1.3 Architecture of Knowledge Based System ..... 13

2.1.4 The Knowledge Based system Designing Process ..... 14

2.2 KNOWLEDGE BASED REASONING TECHNIQUES ..... 17

2.3.1 Case Based Reasoning ..... 17

2.3.2 Rule Based Reasoning ..... 18

2.3.2.1 Rule Based Reasoning Techniques ..... 20

2.3.3 Comparison between Rule and Case Based Reasoning ..... 23
2.4 EXPERT SYSTEM DEVELOPMENT TOOLS ..... 23
2.5.1 SWI Win-Prolog Programming Language ..... 25
2.6 MODELS OF KNOWLEDGE BASED SYSTEM ..... 26
2.7 APPROACHES TO EVALUATE THE PERFORMANCE OF EXPERT SYSTEM ..... 26
2.7.1 Methods of Evaluation ..... 27
2.8 APPLICATIONS OF EXPERT SYSTEM ..... 28
2.9 RELATED RESEARCH WORKS ..... 29
CHAPTER THREE ..... 35
KNOWLEDGE ACQUISITION AND CONCPTUAL MODELLIING ..... 35
3.1. THE PROCESS OF KNOWLEDGE ACQUISITION ..... 35
3.1.1. Interviewing Domain Experts ..... 36
3.1.2 Observation ..... 39
3.2 GENERAL STRUCTURE OF CNTES ..... 40
3.3 THE COMPONENTS OF CNTES ..... 40
3.4 CONCEPTUAL MODELING ..... 42
3.4.1 CNTES and the Decision Tree Structure ..... 42
3.5 KNOWLEDGE REPRESENTATION ..... 45
CHAPTER FOUR ..... 47
KNOWLEDGE BASED SYSTEM IMPLEMENTATION ..... 47
4.1. THE KNOWLEDGE BASE ..... 48
4.2. THE INFERENCE ENGINE ..... 49
4.3. THE USER INTERFACE ..... 52
4.4. LEARNING COMPONENTS OF THE SYSTEM ..... 57
4.4.1 Types of Learning in Rule Based Systems ..... 57
4.5. THE EXPLANATION MODULE ..... 59
CHAPTER FIVE ..... 61
TESTING AND EVALUATION ..... 61
5.1. USER ACCEPTANCE EVALUATION ..... 61
5.2. CNTES VALIDATION BY USING TEST CASES ..... 66
5.2.1. Decision Variation between System and Human Expert ..... 68
5.3. DISCUSSION ..... 69
CHAPTER SIX ..... 71
CONCLUSION AND RECOMMENDATIONS ..... 71

6.1 CONCLUSION ..... 71
6.2. RECOMMENDATION ..... 72
REFERENCE ..... 73
APPENDIXES ..... 78
APPENDIX I, II, III, IV ..... 78

# LIST OF TABLES

Table 2.1 Summary of related works……..……... 34

Table 3.1 Domain experts profile……. 36

Table 3.2 Computer and network problems …… 41

Table 5.1 Performance evaluation of CNTES …… 64

Table 5.2 Testing the accuracy of CNTES by using test cases ……..... 68

Table 5.3 Summary of system evaluators result on close ended questions…….. 70

## LIST OF FIGURES

Figure 2.1 Architecture of knowledge based system ……... 13

Figure 2.2 Development of knowledge based system ……... 14

Figure 2.3 Forward chaining inference process …….. 21

Figure 3.1 General structure of CNTES…….…….…….. 40

Figure 3.2 Decision tree for power supply problem …… 44

Figure 3.3 Decision tree for hard disc problem ……...…… 44

Figure 3.4 Decision tree for network problem …… 45

Figure 3.5 Decision tree for sound problem ……..... 45

Figure 4.1 Architecture of rule based system …….. 48

Figure 4.2 Welcoming window of CNTES user interface ……... 54

Figure 4.3 Dialog windows between the user and system ……... 55

Figure 4.4 Dialog windows between the user and system ……... 56

Figure 4.5 Dialog windows between the user and system ……... 57

Figure 4.6 Dialog windows between the user and system ……... 58

Figure 4.7 The learning component …… 59

Figure 4.8 Sample of explanation facility dialog windows for network problem ……... 60

## ACRONYMS

AI ---------- Artificial Intelligence

CBR --------- Case Based Reasoning

ES ---------- Expert System

IBM --------- IBM personal computer

KB ---------- Knowledge Base

KBS --------- knowledge Based System

KA ---------- knowledge acquisition

LAN --------- Local Area Network

PM ---------- Preventative Maintenance

PDA -------- Portable Device Application

PHP --------- Hypertext Markup language

RBR ----------rule based reasoning

# ABSTRACT

Information and communication technology become crucial in any aspects of organizations activity to perform their tasks. However, computer and network troubleshooting services were not efficient in developing countries like Ethiopia as it is required to be. In Ethiopia, computer related problems are hot issues in different organizations those were using computer and computer related devices to perform their tasks. In computer domain the professionals were limited in number but the number of institutions looking for computer professionals was high.

The major challenges in troubleshooting computer and network are lack of expertise to troubleshoot problems with hardware and software which is leading to problems such as connection and virus attaches. These and other computer related problems need maintenance from the ICT technicians. But the problems are not solved as required by the computer users due to different reasons (skill level of ICT technicians, limited number of ICT technical experts, turnover rate and motivation of ICT professionals).

As a result of this, the main aim of this study is developing knowledge based expert system which is providing troubleshooting information about problems occurred in the computer system for the ICT technicians and computer users to solve problems effectively and efficiently to utilize computer and computer related resources.

To achieve this objective, domain knowledge was acquired using semi-structured interview technique, observation and document analysis. Domain experts were selected from Jimma University ICT office using purposive sampling techniques.

The conceptual model of the knowledge based system is designed by using a decision tree structure which is easy to understand and interpret the causes involved in computer troubleshooting. Based on the conceptual model, the prototype was developed with SWI WIN prolog by using 'if – then' rules. The developed prototype uses backward chaining to infer the rules and provide appropriate recommendations.

The prototype was evaluated by using different techniques and the performance of the prototype knowledge based system had got good acceptance by the system evaluators. According to the system evaluators 83.6% of the users were satisfied with the prototype. In addition, the performance of the system was evaluated by using predictive validation techniques with twelve test cases. The results of the validation test cases indicate that the prototype registers on the average 75% accuracy. The system is learning its facts but not fully update rules as well, which left for further research.

# CHAPTER ONE

# INTRODUCTION

## 1.1 BACKGROUND

Computer systems (hardware, software and network connection) face problems from time to time and endanger the functionality of the institutions. These problems need ICT technicians to be solved. The ICT technicians follow traditional (manual) ways of solving computer problems which is taking long time and high cost to reach the solution.

An Expert system is a computer system that emulates the decision making ability of a human expert, i.e., it acts in all respect of human counterpart. An expert system is sometimes referred to as knowledge-based system and is computer software that emulates the decision-making ability of a human expert [1, 2]. Expert systems do not use traditional programming paradigms to solve problems; rather, they use knowledge which they reason about to draw conclusions and provide solutions [3]. The advantage of expert systems over conventional programs is that their core algorithm is not encapsulated in the programming code but stored as knowledge in an independent database called knowledge-base or KB. In consequence, there is no need for the expert system to be reprogrammed and recompiled every time the knowledge changes. Practically, expert systems have significant applications including medical diagnosis, fault diagnosis, question-answering, industrial process controlling, climate forecasting, manufacturing failure analysis, decision support, and decision making [4]. Today, with the prevalent use of computers, fault diagnosis is becoming crucial in the field of computer engineering and information technology, particularly in personal computer troubleshooting; however, acquiring the troubleshooting knowledge from expert computer technicians is limited as it requires continuous learning, training, and practice in maintenance skills which on the long run can dramatically increase organization operating costs, decrease their net productivity, and proliferate their revenue leakage and losses. Basically, PC (Personal Computer) troubleshooting covers a wide spectrum of problems including hardware problems, software problems, network problems, server problems, operating system problems, and application software problems. This would impose a maintenance nightmare for large-scale enterprises and IT

infrastructures as the number of possible technical problems and finding their solutions can become very huge and complex [5].

The most common cause of delay in solving a problem occurring in the computer system is trial and error. The problem can be solved earlier if the person diagnosing it already knows the cause of the problem rather than resorting to trial and error. There are instances that because of this trial and error, the problem gets worse rather than being solved. Some problems can be solved quickly; there are situations when it only takes a few minutes to solve a problem but because the person diagnosing it does not know the cause of the problem, troubleshooting takes days or months causing much inconvenience [1].

Knowledge Based CNTES help user to search solution with specific answer for the specific problem. This system was determined to find troubleshoot appropriate for storing, maintaining, and retrieving the Knowledge Base [6]. The computer troubleshooting expert system presents the technician with a list of guiding questions, when answered; provide the system with information about what problem sets the issue may belong to, and which ones it does not belong to. It then generates another list of questions that apply to subsets of those identified. When all possible sub-sets have been narrowed down to individual possible causes the system returns a list of possible solutions. This process can be saved midway through by saving the current list of questions and possible solutions [7].

The system is meant to automate the maintenance, repair, and operations process, and free-up human technicians from manually performing routine, laborious, and time-consuming maintenance tasks. Computer systems typically include a combination of hardware and software components, application programs, System programs, processors, buses, memory, input/output devices etc. As advances in semiconductor processing and computer architecture push the performance of the computer higher and higher. More sophisticated computer software has evolved to take advantage of the higher performance of the hardware, resulting in computer systems that are much more powerful than just a few years ago. In a computer system, in various parts different faults may arise. Expert system is able to detect those faults and also to suggest for the probable problems and solutions. Generally, when user consults an expert system, the system interviews (ask questions) the user and gets the facts about the problem at hand. During the interviews, questions appear on the computer display screen and the user responds by keeping in answers or important data. Then, utilizing the user's response the system searches its knowledge base. It systematically searches through the various paths for a solution without becoming lost in the vast numbers of possibilities. Eventually the program comes up with the advice and communicates it to the user. The system also explains its

reasoning process and remedy. Before demonstration of the proposed model it is required to understand the knowledge domain based on Artificial Intelligence and its application, i.e., Expert System [3].

The Expert CNTES aids in decision-making and allows automated fault detection and deep problem diagnosis by emulating the human reasoning activity. From a business perspective, the system would reduce maintenance costs, provide quicker problem resolution, reduce employee training time, and deliver higher job quality. This expert system has self learning ability to handle new problems occurring in the computer system [5].

## 1.2 STATEMENT OF THE PROBLEM

A typical, large-scale computer installation is composed of numerous subsystems including CPUs, primary and secondary storage, peripherals, and supervisory software. Each of these subsystems, in turn, consists of a richly connected set of both hardware and software components such as disk drives, controllers, CPUs, memory modules, and access methods. Generally, each individual component has an associated set of diagnostic aids designed to test its own specific integrity. However, very few maintenance tools and established diagnostic strategies are aimed at identifying faults on the system or subsystem level. As a result, identification of a single or multiple faults from systemic manifestations remains a difficult task. The non-specialist ICT technician is trained to use the existing component-specific tools and, as a result, is often unable to attack the failure at the systemic level. Expert assistance is then required, increasing both the time and cost required determining and repairing the fault. The design of knowledge based PC troubleshooting expert system reflects the expert's ability to take a systemic viewpoint on problems and to use that viewpoint to indicate specific components, thus making more effective use of the existing maintenance capabilities [8].

Lack of expertise to troubleshoot problems with hardware and software is leading to problems such as connection and virus attaches. These and other computer related problems need maintenance from the ICT technicians but are not solved as required. Diagnosing a problem is very difficult because of the complexity of the systems and the constant change. The systems are complex functionally as a whole and additional complexity is introduced through the interactions of heterogeneous components. The change comes about both internally through hardware and software updates and new components, and externally through environment. The behavior of the system is changing at different rates due to load, traffic mix and configuration [9].

According to Kunesegeran [6], the demand for higher availability of production systems to have a cost-effective distribution of fixed assets and to minimize manufacturing delays due to machine failures caused by unplanned downtimes. Unplanned downtime increases the maintenance costs and reduces productivity. The manufacturing units may cancel customers or delay, as it happens regularly, the preventative maintenance project days in not advance. Some emails from PC user based on their problem are not replayed by technician [6]. The ICT technicians are full of activity and it is hard to find them on time as needed. This knowledge based CNTES free up the ICT technician because the computer users can solve problems in their own. In addition, this system play enormous amount of roles for the organizations achievement because it reduces the wastage of time by waiting ICT technicians, reduce cost and saves computers being stored in salvage room.

The most common cause of delay in solving a problem occurring in the computer system is trial and error. The problem can be solved earlier if the person diagnosing it already knows the cause of the problem rather than resorting to trial and error [1]. This knowledge based CNTES provides enough information for the ICT technicians to solve the problem by feeding causes of the problem to the expert system. Not only the ICT technicians, but the computer user him/herself can solve problems by following advice from the expert system. "Good troubleshooters are systematic in eliminating various possible explanations as users focus on the source of the problem. With proper preventive maintenance (PM), many of these PC hardware and software can be made to last longer. When computer users were faced any PC problem they often can be repaired by themselves" [6].

In the modern world most of the activities are related to the computer technology. And computers play a crucial role for the success of various institutions. The employers working with computer face serious and minor problems which causes big impact on the operation of the organization. The employees call or directly going to the ICT technicians for maintenance. But the problems are not solved as required by the computer users due to different reasons (skill level of ICT technicians, limited number of ICT experts, turnover rate and motivation of ICT technicians). As Ian C. Cameron said, in reality, the computer troubleshooting industry had a fundamental problem. The industry has a great need for expert technicians, but few are available. The current software used to assist technicians assumes that they are already skilled at narrowing down the possible causes of a computer problem, and so they are only useful when a cause to the issue at hand is evident [8].

As the researcher's knowledge, there are no local researches in troubleshooting computer and network problems but there are international works in the area. Those international works advice the computer users and experts how to solve problems occurred in the computer system.

In general, there are various problems which lead the computer maintenance process endanger. The main problems are limited number of ICT technicians, lack of experienced ICT technicians to solve the problems, turnover rate of ICT technicians, constant change and complexity in the computer technology. The problems occurred in the computer system take long time to be solved (backlog), needs high cost, and makes third party to be benefited.

From the related works reviewed, the researcher identified the **gap** in the existing works. All the works reviewed have the ability to advice the user or computer expert to diagnose computer and network problems by inferring the recorded knowledge in the knowledge base. But they are not learning the new problems occurring in the computer system due to technological changes and system complexities and do not update the knowledge base. This expert system "computer and network troubleshooting expert system" have the ability to learn new problems occurring in the computer system to cope up with the technological changes and complexities. When the new causes were found this expert system update its knowledge base to be used in future troubleshooting process.

To solve problems identified, this study attempts to explore and answer the following research questions:

- What types of knowledge the human expert implement to troubleshoot computer and network problems?
- What models, techniques, representation and implementation tools are required for the application of the proposed knowledge based system?
- What is the ICT technicians' interest about the proposed knowledge based computer and network troubleshooting expert system?

## 1.3 OBJECTIVES OF THE STUDY

### 1.3.1 General Objective

The main objective of this study is to develop knowledge based expert system which enables users to perform computer and network troubleshooting so as to utilize computer and computer related resources effectively and efficiently.

### 1.3.2 Specific Objectives

To achieve the general objective of this study, the following specific objectives are formulated:

- To identify the domain knowledge required to design the system
- To acquire the knowledge from the human expert
- To model and represent the tacit knowledge in a suitable way for knowledge based expert system implementation
- To develop a prototype knowledge based expert system that learns through experience
- To measure performance of the knowledge based expert system

## 1.4 SCOPE OF THE STUDY

There are a number of different approaches for designing knowledge based system but this system only focuses on rule based approach. The focus area of this study was Jimma University ICT office for the purpose of data collection. This study is limited to maintaining problems related to computer systems (computer and network) and providing possible suggestions for cases and learns new cases of the problem and support for decision making in computer diagnosing. There are different problems related to computer system but including all maintenance issues influence the system performance. Due to this reason this study focused on basic hardware problems, software (power problem, storage, compatibility and fixing) problems, and network configuration problems (cable connection, IP address and driver installation). From the problems included in the system only network problem section is updating its knowledge base.

Limitation: some of the experts were not voluntary to provide the information, lack of previous works which have the ability to self learning, lack of standardized measurement to compare with the findings and the implementation tool user interface was not simple to interact.

## 1.5 SIGNIFICANCE OF THE STUDY

The computer users or computer troubleshooting ICT technicians were beneficiary from this knowledge based CNTES because computer problems can be solved easily. By the help of this system the computer users solve problems occurring in the computer system by their own instead of waiting or calling to the ICT technicians by referring to the recommended solutions from the expert system. The system fills the skill gap of the ICT technicians and helps them to cope up with the constant change and complexity of the computer system. Economically, this system reduces the cost of maintenance, increases the efficiency of the work and increased utilization of the computer related resources. The developed prototype knowledge based CNTES is used to give advising services for both computer users and ICT technicians. It gives better advice services where human expert are not

found and limited in number. Generally, the organizations success is increased because the PC downtime is decreased.

## 1.6 METHODLOGY

To achieve the main objective of this study, different methods and techniques are used.

### 1.6.1 Literature Review

For clear understanding of the problem related to PC troubleshooting, to collect necessary data related to expert system and for successful completion of this study the relevant literatures such as books, journals articles, conference proceedings, manuals, and resources from the internet are reviewed for achieving the research objective [10].

### 1.6.2 Data Collection

Data is required for any study. For the purpose of this study, both primary and secondary data collection methods were employed to collect the required domain knowledge. The knowledge based CNTES was developed by using the knowledge of different ICT experts and documentary sources which is used as organizational memory. As primary sources, ICT professionals from Jimma University ICT office were interviewed and observation technique was used when the expert troubleshoot the problems. Additionally, relevant literatures from all possible source including journal articles, computer manuals, and recorded guidelines were reviewed as secondary source.

Domain experts were selected by using purposive sampling techniques. *Burns and Grove [10]* have defined purposive sampling as "judgmental sampling that makes the conscious selection of certain subject or element to include in the study." Purposive sampling techniques were employed to select sample which can help to acquire the required knowledge from the domain experts. Domain experts were selected based on their education and responsibility level related to their domain area, year of experience, and willingness [11, 12]. Primary sources of knowledge were collected from human experts in the domain area at Jimma University ICT office. Since one of the main focus of this research is eliciting relevant knowledge from the domain experts. Four (4) domain experts were selected using purposive sampling technique. These experts were selected due to their managerial position and experience in the ICT office

A semi-structured interview technique was employed to acquire the required knowledge from the selected ICT professionals. This way of collecting data allows the interviewer to change the order of questions and add new questions depending on the response. Therefore, this interview question focuses on manuals, documentations, guidelines and experiences, which the ICT professionals use to solve problems occurring in the computer system [10].

#### 1.6.2.1 Knowledge Acquisition and Representation

Constructing knowledge based systems was viewed as a modeling activity for developing structured knowledge. To ensure well-formed models selecting appropriate knowledge modeling methodology would be a critical task. Additionally, reusing models can significantly reduce the time and costs of building a new application [13].

The acquired knowledge from the domain experts was represented by using decision tree modeling in a formal language logic which is suitable for the selected algorithm. Rule based reasoning mechanism were employed for the inference engine. In knowledge based system there are many reasoning mechanisms; among that the most commonly used are rule based approach, case based approach or the combination of the two. Case based approaches are designed to work in the way that the basic idea of similar problems having similar solutions [10]. It is a rule based System that solves problems by remembering past situations and reusing its solution and lesson learned from it. Case based approach represents situations or domain knowledge in the form of cases and it uses case based reasoning techniques to solve new problems or to handle new situations [14]. Rule based reasoning, on the other hand reason from domain knowledge represented in a set of rules. The basic format of a rule is

IF <condition> THEN <conclusion>, where <condition> represents premises and <conclusion> represents associated action for the given premises [15].

### 1.6.3 Implementation Tool

The knowledge based expert system was implemented by using prolog programming language. The reason is that, prolog is open source software and it is the preferred programming language for developing a knowledge based system. The advantage of expert systems over conventional program is that their core algorithm is not encapsulated in the programming code but stored as knowledge in an independent database called knowledge-base or KB. In consequence, there is no need for the expert system to be reprogrammed and recompiled every time the knowledge changes. Prolog

makes the representation of the solution much easier, allowing one to concentrate on solving the problems rather than coding the solution.

### 1.6.4 Evaluation

The developed rule based system was tested and evaluated to ensure the performance of the system. The evaluation processes focus on system's user acceptance. User acceptance measurements are concerned with issues how well the system addresses the needs of the user, whereas validation measurement determine if the system perform the required task successfully.

The system evaluators use visual interaction methods together with questionnaire. Based on that, they evaluate the performance of the system by using both close ended and open ended questions.

Similarly validation process of the prototype knowledge based system was tested using twelve test cases. Evaluators from the domain area interact with the system by taking a sample of test cases. Each case was selected purposively and used to test the performance of the prototype. The testing procedure was carried out by identifying correct and incorrect categorization of cases into their respective category. The comparison was the decision made by system against domain expert. Decisions without significance difference between the domain experts and the knowledge based system will be accepted as good performance.

There are a number of well-known knowledge based systems which use such approach to validate the accuracy of knowledge based system such as MYCIN and An Expert System for Computer Fault Diagnosis called DART [8].

The knowledge based system incorporated learning component to update the existing facts in working memory. Finally user evaluation was conducted by group of domain experts. The domain expert used different cases (based on problems occurred in the computer system) to interact with a knowledge based system. After assessing the results concluded system against human expert decision, then system evaluator measure its performance.

## 1.7 ORGANIZATION OF THE THESIS

The study is organized into six chapters. Chapter one is the introduction part, which contains the background of the study, problem statement, objectives, scope of the study, the significance of the study and methodology to carry-out the research.

Review of conceptual and related literature on the knowledge based systems, about its background, architecture, development phases, and knowledge based system overview and application areas in computer domain are presented in chapter two.

Chapter three discusses the knowledge acquisition, representation and conceptual modeling procedures; Chapter four deals with implementation of knowledge based system. Chapter five presents the results found in the evaluation and testing process of the prototype knowledge based system. Finally, chapter six focuses on the conclusion and recommendation based on the results of the research finding for further research work in the domain area.

# CHAPTER TWO

# LITERATURE REVIEW

The notion of knowledge based system is derived from the field of AI. The representation of knowledge and the reasoning processes that brings knowledge to life is the central point to the entire field of artificial intelligence. Knowledge and reasoning mechanism are important aspect for artificial intelligence because they enable successful behaviors that will be very hard to achieve otherwise [16].

## 2.1. ARTIFICIAL INTELLIGENCE

Artificial Intelligence (AI) is one of the up-to-date sciences; the name artificial intelligence itself was coined in 1956. Majority definitions of artificial intelligence proffered over decades have relied on comparisons to human behavior. Thus according to Fogel [17] different authors define artificial intelligence in different ways such as: -"the sciences of making machines do things that would require intelligence if done by men"—and suggested that some people define AI as the "mechanization or duplication of the human thought process." "Artificial intelligence is the study of mental process through the use of computational models," "An intelligent program is one that exhibits behavior similar to that of a human when challenged with a similar problem.

It is not necessary that the program actually solve or attempt to solve the problem in the same way that a human would." [18].

Intelligent program (AI) has interdisciplinary approach of various disciplines like computer science, cognitive science, hardware field, etc. In the medical applications areas such as diagnostic techniques in ultra-sonography, x-ray, computerized tomography scans, nuclear magnetic resonance imaging etc. The other areas can be clinical laboratories, pathological investigations and computer assisted decision-making. Therefore, the field is potentially relevant to any sphere of human intellectual activity. AI started with a goal to replicate human level of intelligence in machine learning [19].

### 2.1.2 Expert System

Expert system (ES) and knowledge based systems are synonym terms used in machine learning [20]. Knowledge-Based System (KBS) is one of the major family members of the AI group. KBS/ES can act as an expert on demand without wasting time, anytime and anywhere. KBS can save money by leveraging human expert, allowing users to function at higher level and promote consistency of work. One may consider the KBS as productive tool that have knowledge of more than one expert for long period of time. In fact a KBS is a computer based system which uses and generates knowledge from domain expert [21].

Human experts use their knowledge in particular field of expertise to solve day today activities. In the same way, knowledge based system handles problems; the computer needs an internal model of the world using the stored knowledge. All information is stored in such a way that it is readily accessible. To design knowledge based system, the expert knowledge was represented in a way that it supported for reasoning mechanism in computer languages. Representing knowledge into the expert system could offer potential advantages over human expertise.

Because, knowledge based system can use the acquired knowledge permanently, consistently, easy to transfer and document expert knowledge [18].

## The Need for Expert Systems

Expert systems are necessitated by the limitations associated with conventional human decision-making processes, including [22]:

- Human expertise is very scarce.
- Humans get tired from physical or mental workload.
- Humans forget crucial details of a problem.
- Humans are inconsistent in their day-to-day decisions.
- Humans have limited working memory.
- Humans are unable to comprehend large amounts of data quickly.
- Humans are unable to retain large amounts of data in memory.
- Humans are slow in recalling information stored in memory.
- Humans are subject to deliberate or inadvertent bias in their actions.
- Humans can deliberately avoid decision responsibilities.
- Humans lie, hide, and die.

Coupled with these human limitations are the weaknesses inherent in conventional programming and traditional decision-support tools. Despite the mechanistic power of computers, they have certain limitations that impair their effectiveness in implementing human-like decision processes. Conventional programs [22]:

- Are algorithmic in nature and depend only on raw machine power
- Depend on facts that may be difficult to obtain
- Do not make use of the effective heuristic approaches used by human experts
- Are not easily adaptable to changing problem environments
- Seek explicit and factual solutions that may not be possible

### 2.1.3 Architecture of Knowledge Based System

Figure 2.1 below shows the building blocks of knowledge based system architecture adopted from [23].

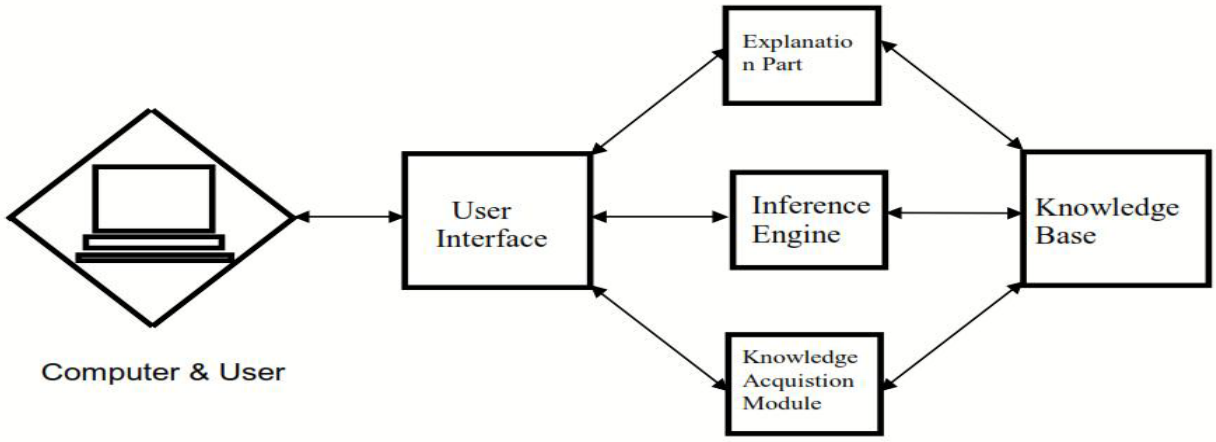



**Figure 2.1 Architecture of knowledge based system (**adopted from [23]**)**

The architecture of knowledge based system consists of different components such as Knowledge Base, Knowledge acquisition module, inference engine, user interface and explanation module. The Knowledge Base contains all relevant knowledge acquired from domain experts. Knowledge also acquired from the user during their interaction with the system.

The knowledge acquisition module helps in the collection process of knowledge from the set of human experts as shown in Figure above. The inference engines formulate questions and assert the answers provided by the user in a natural language form. It provides a mechanism for conveying recommendations to the end user. The explanation module provides a brief description to the user why the system arrived at a certain conclusion [24].

### 2.1.4 The Knowledge Based system Designing Process

The development of knowledge based system is the integration of many components. Figure 2.2 below shows the overview of knowledge based system development process [20].

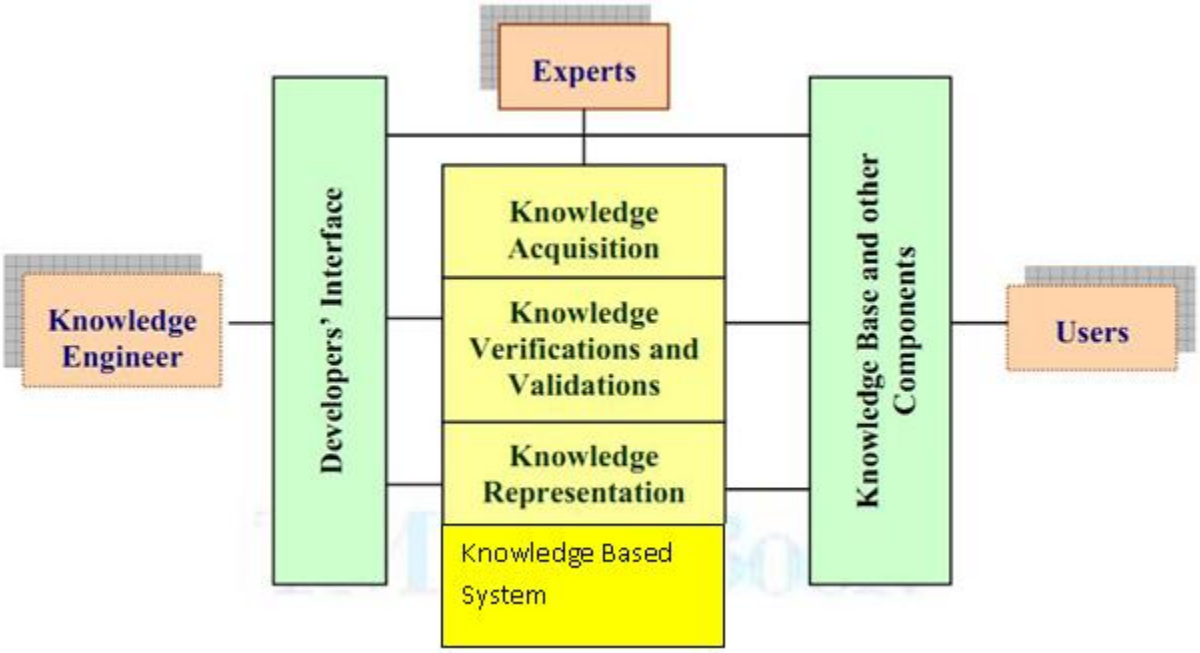



**Figure 2.2 Development of a Knowledge-Based System**

2.1.4.1 **Knowledge acquisition**- Knowledge acquisition is the process of acquiring relevant knowledge from human experts, books, documents, sensors, or computer files. The knowledge can be specific to the problem domain or to the problem-solving procedures, it can be general knowledge (e.g., knowledge about business) or it can be meta-knowledge (knowledge about knowledge). Knowledge acquisition is the bottleneck in knowledge based system development today. Because, the trustworthiness and the performance of the knowledge based system mainly depends upon the acquired knowledge [25].

The knowledge acquisition process incorporates different methods such as interviews, questionnaires, record reviews and observation to acquire factual and explicit knowledge [26].The performance of the expert systems depends upon the reliability, validity and accuracy of the elicited knowledge. The process of knowledge elicitation is affected by different contributing factors such as communication between the expert and ability of knowledge engineer [27].

Therefore, effective acquisition techniques facilitate to acquire relevant knowledge from domain experts. The commonly used knowledge acquisition techniques are discussed as follows [28].

## A. Interview

An interview technique is the process of interacting with domain expert on how they perform their tasks based on their expertise. Knowledge acquired through direct elicitation methods are procedural knowledge. Based on its structure, interview can be classified into structure, semi-structured and unstructured interview [15].

- **Structured Interviews**- A structured interview method is questioning the domain expert directly. It is goal-oriented process. It forces organized communication between the knowledge engineer and the domain expert. The structure reduces the interpretation problems inherent in unstructured interviews and allows the knowledge engineer to prevent the bias caused by the subjectivity of the domain expert [29].
- **A semi-structured interview** is an interview which has a guide that usually includes both closed-ended and open-ended questions. It is more flexible than structured one. In these kinds of interview the interviewer has a chance to change the order of questions and expand the dimension of questions based on the participants' responses [29].
- **Unstructured Interviews**- sessions are conducted informally, usually as a starting point. Unstructured interview techniques provide complete or well-organized descriptions of cognitive processes. There are many reasons that enforced to applying unstructured interview. Domain the experts usually find it very difficult to express some of the most important elements of their knowledge. Through structured interview it is difficult to acquire the required knowledge. With good training and personal experience knowledge engineers can use unstructured interview to acquire relevant knowledge from domain expert [29].

Therefore, efficient and effective interview techniques largely depend on the ability of knowledge engineer to articulate their implicit knowledge. Because every interview is different in very specific ways and it is difficult to provide comprehensive guidelines for the entire interview process. Therefore, interpersonal communication and analytic skills of knowledge engineer is very important. On the other hand eliciting knowledge using indirect methods requires human intervention such as observation, document analysis, etc. [28].

## B. Observation

In many ways, this is the most obvious and straightforward approach to knowledge acquisition.

Through this technique knowledge engineer directly observe how problems are solved by domain expert. Observation was particularly fruitful in seeing what the expert physically does when solving diagnostic problems. It was useful in determining what types of knowledge the human expert used to solve problems and the forms in which the knowledge were stored.

Observations are used primarily as a way of supporting verbal protocols. Generally, acquiring knowledge through observation is expensive and time-consuming task [30].

## C. Document Analysis

The final form of knowledge acquisition method is concerned with a detailed analysis of the existing document. This technique is used to collect relevant knowledge from the existed documents of different format. These documents include professional literature, brochures, manuals, guidelines, employee handbooks, reports, glossaries, course texts, and other relevant materials [30].

Knowledge elicitation methods can be classified into different types. Direct and indirect is the commonly known methods of knowledge elicitation. The way of classification depends upon how knowledge engineer directly obtains information from the domain expert [31].

A direct method involves directly questioning a domain expert on how they do their job. In order to implement direct methods successfully, the domain expert has to reasonably articulate and willing to share his/her knowledge. However, in case of indirect methods the required knowledge is not requested directly. Instead, the result of the knowledge elicitation session must be analyzed in order to extract the required knowledge. Indirect methods are thought to be more suitable when knowledge is not easily expressed by the domain expert [31].

**Knowledge representation**- Acquired knowledge is structured so that it was ready for use in the process of knowledge representation. This activity involves preparation of a knowledge map and encoding of knowledge in the knowledge base.

**Knowledge validation**- Knowledge validation (or verification) involves validating and verifying the content of knowledge (e.g., by using test cases and confusion matrix) and user acceptance. The testing result of knowledge based system was validated by domain expert.

**Inference**- This activity involves the design of software to enable the computer to make inferences based on the stored knowledge for the specific domain problem. In other word inference engine is a programs that reason over extensive knowledge bases.

**Explanation**- This step involves the design and programming of an explanation facility.

Explanation module is program that answered how the knowledge based system arrived at certain conclusion. Explanation module addresses the issues of system user interactivity.

## 2.2 KNOWLEDGE BASED REASONING TECHNIQUES

There are a number of knowledge based reasoning methods. The well-known reasoning approaches are ontology based reasoning, case based reasoning and rule based reasoning. For the purpose of this research work, case-based and rule-based reasoning approaches were discussed as follows.

### 2.3.1 Case Based Reasoning

Case-based reasoning (CBR) means "adapting old solutions to meet new demands, using old cases to account for new situations, using old cases to evaluate new solutions, or reasoning from precedents to interpret a new situation"[32].

CBR is more comfortable to make better decision in dynamically changing environment. People learn from their success and wrong activities to handle similar situations in the right manner and not to repeat their mistake of the past. CBR approach is more compatible to reuse previously solved problems and learning from experiences for future decision [14]. Similarly, CBR is an approach to incremental learning. Once a problem has been solved, CBR approaches use the solution to solve for future problems [33].

**Cases**

In CBR terminology cases are usually denotes a problem situation. A case can be defined as previously experienced situation which has been captured and learnt, is referred to as a past case, previous case, stored case or retained case. Note that the term problem solving is used with common practice in the area of knowledge-based systems. This means that problem solving is not necessarily the finding of a concrete solution to an application problem, it may be any problem put forth by the user [33].Therefore, and cases have three different aspects this can be described as follows.

- Situation/ problem description: describes specific circumstances, the state of a situation and state of the environment when this case is recorded.
- Solution: provide how the problem described was solved or treated in a particular instance.
- Outcome: describe the final result, consequence and feedback gained from the proposed solution.

Case-based reasoning mechanism requires the following primary activities.

- Index assignment: Characterizes the given problem by assigning the appropriate attribute that describe the features of the case.
- Retrieval: Retrieves the relevant case from the case library.
- Explanation: Explains the deficiencies of the retrieved case by making a comparison of the differences between this case and the input problem. The explanation involves two aspects, i.e., which features are unsatisfactory and require modification and how to modify these features so as to satisfy new conditions.
- Modification: Modifies the retrieved case to conform to new situations according to the result of explanation.
- Store/adapt: Saves the modified case as a new case into the case library. The case libraries are incrementally expanded as the numbers of cases increase.

**Advantage of case based reasoning**

A case based reasoning approach has tremendous advantages in the development of knowledge based system. The following are main the advantages of case-based reasoning [34].

- Ability to express specialized knowledge.
- Naturalness of representation
- Modularity
- Easy to knowledge acquisition
- Self-updatability.
- Handling unexpected or missing values.
- Inference efficiency.

**Disadvantage of case based reasoning**

Even though case based reasoning approaches have a numbers of advantages. But, due to lack of sufficient cases, the construction and inference mechanism of a case-based system loss the required objective. Some of the limitation issues in case-based reasoning are [35]:

- Inability to express general knowledge
- Knowledge acquisition problems
- Inference efficiency problems and Provision of explanations

### 2.3.2 Rule Based Reasoning

Rule based reasoning (RBR) is a system whose knowledge representation contains a set of rules and facts [35].

Symbolic rules are one of the most popular knowledge representation and reasoning methods.

This popularity is mainly due their naturalness, which facilitates comprehension of the represented knowledge. The basic forms of a rule, If<condition> then <conclusion> where

<condition> represents premises and <conclusion> represents associated action for the premises.

The condition of rules are connected between each other with logical connectives such as, AND, OR, NOT, etc., thus forming a logical function. When sufficient conditions of a rule are satisfied, then the conclusion is derived and the rule is said to be fired. Rule based reasoning was dominantly applied to represent general knowledge. Rule based expert systems have a significant role in many different domain areas such as computer maintenance, medical diagnosis, electronic troubleshooting and data interpretations. A typical rule based system consists of a list of rules, a cluster of facts and an interpreter [35].

**Rules**

The term rules represent what to do or not to do while certain conditions are satisfied. Similarly, domain knowledge is represented by a set of rules [36]. The general form of rules based system can be illustrated as follows [37].

IF

First premise, and

Second premise, and

...

THEN

Conclusion

The IF side of the rule is referred to as the left hand side (LHS) and the THEN side of the rules referred to as the right hand side (RHS). This is semantically the same as a Prolog rule:

```
Conclusion:-
    first_premise,
    second_premise ...
```

Note that this is confusing since the syntax of Prolog rely on THEN IF, and the normal RHS and LHS appear on opposite sides.

Antecedents are evaluated based on what is currently known about the problem being solved.

This mean if all antecedents (premise) of the rule evaluated is true, the actions in the consequents part is executed. Each antecedent of a rule typically checks if the particular problem instance satisfies certain conditions. When the consequents of the rules are executed, the rule is said to be fired [38].

#### 2.3.2.1 Rule Based Reasoning Techniques

There are two main inference methods in rule based reasoning mechanism. These are backward chaining and forward chaining. The former is guided by the goals (conclusions), whereas the latter one is guided by the given facts [35].

***A. Forward chaining***

During forward chaining, the inference engines first predetermine the criterion and the next steps are to add the criterion one at a time, until the entire chain has been trained. With data driven control, facts in the system are represented in a working memory which is continually updated.

Rules in the system represent possible actions to take when specified conditions hold items in the working memory. The conditions are usually patterns that must match with the items in the working memory. In forward chaining, actions are usually involves adding or deleting items from the working memory. Interpreter of the inference engine controls the application of the rules, given the working memory. The system will first checks to find all the rules whose condition holds true [38, 39]. Both data driven and goal driven chaining method follows the same procedures. However, the difference lies on the inference process. The system keeps track of the current state of problem solution and looks for rules. This cycle will be repeated until no rules fire or the specified goal state is satisfied [37].

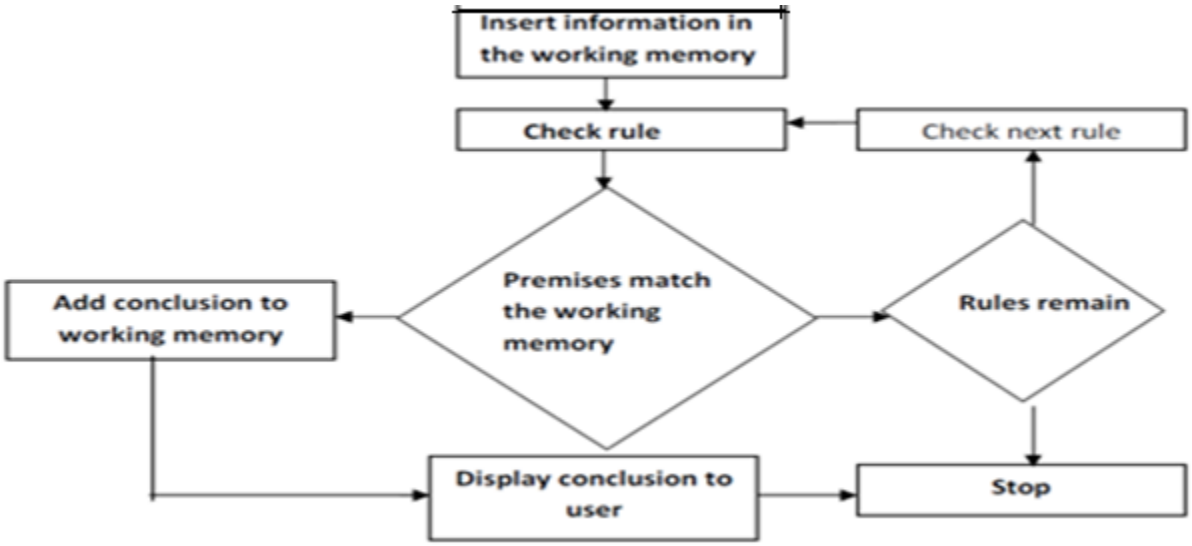



Figure 2.3 Forward Chaining Inference Processes, adopted from [37].

### *B. Backward chaining*

This strategy focuses its effort by only considering rules that are applicable to the particular goal.

It is similar with forward chaining the difference is it receives the problem description as a set of conclusions instead of conditions and tries to find the premises that cause the conclusion. Given a goal state and then the system try to prove if the goal matches with the initial facts. When a match is found goal is succeeded. But, if it doesn't then the inference engine start to check the next rules whose conclusions (previously referred to as actions) match with the given fact. Note that a backward chaining system does not need to update a working memory instead it keeps track of what goal is needed to prove its main hypothesis. Goal driven control is commonly known as top-down or backward chaining [37, 40].

**Forward Vs Backward Reasoning**

According to Freeman-Hargis [38], both forward chaining and backward chaining have similar function. But, the difference occurs due to the data structure of knowledge based system. The following point give us a clear ideas how and when to apply each reasoning mechanisms.

- Whether you use forward or backwards reasoning to solve a problem, it depends on the properties of your rule set and initial facts.
- Sometimes, if you have particular goal (to test some hypothesis), then backward chaining is more efficient, as you avoid drawing a conclusions from irrelevant facts.

- Sometimes backward chaining can be very wasteful - there may be many possible ways of proving the hypothesis, and it may require checking almost all of the rules before you find one that works.
- When you have a small set of initial facts; and when there is lots of different rules which allow you to draw the same conclusion it is better to use forward chaining.
- Backward chaining may be better if you are trying to prove a single fact, given a large set of initial facts. Because if you used forward chaining lots of rules would be eligible to fire in any cycle.

**Advantage of rule based reasoning**

Rule based reasoning approach have a numbers of good features. According to Jim Prentzas

[35], the major advantages of rule based reasoning in the development of knowledge based system are:

- Compact representation of general knowledge. Rules can easily represent general knowledge about a problem domain.
- Homogeneity. Rule based representation has uniform syntax. Hence, the meaning and interpretation of each rule can be easily analyzed.
- Independent. In rule based knowledge representation a new rule can be added without affecting the existing rules. Each rule is an independent piece of knowledge about the problem domain.
- Naturalness of representation. Rules are a very natural knowledge representation method with a high level of comprehensibility. Rules can emulate the expert's way of thinking in natural expression.
- Modularity. Each rule is a discrete knowledge unit that can be inserted into or removed from the knowledge base without taking care of any other technical detail. This characteristic grants flexibility of rule-based reasoning. Because it enables incremental development of the knowledge base.
- Provision of explanations. The ability to provide explanations for the derived conclusions is a straightforward manner. This feature of symbolic rules is a direct consequence of their naturalness and modularity.

**Disadvantage of rule based reasoning**

As rule based reasoning of prototype knowledge based system has many advantages. But, it has the following limitations [36].

- Knowledge acquisition bottleneck- The standard way of acquiring knowledge through interviews with domain experts is bulky and time-consuming.
- Brittleness/fragility of rules- It is not possible to draw conclusions from rules when there are missing values in the input data.
- Inference efficiency problems- In certain cases the performance of the inference engine is not the desired one especially when the rules are too large.
- Difficulty in maintenance of large rules- The maintenance of rule bases is getting a difficult process as the size of the rules increases.
- Interpretation problems- The general nature of rules may create problems in the interpretation of their scope during reasoning process.

### 2.3.3 Comparison between Rule and Case Based Reasoning

Cased based reasoning uses partial matching to draw a conclusion. If some of the given problem descriptions match with a given case, then the case is applicable to the proposed solution. It also tries to handle novel problems by referring previously solved cases. Rule based reasoning uses perfect matching to apply a rule for a given problem. It doesn't handle missing information and unexpected data values [36].

Rules are suitable to represent general knowledge, whereas cases are suitable for representing specific situations. Rules in a rule based system have the abilities to represent experiential knowledge acquired from experts in a direct fashion. Cases are capable of representing specific historical knowledge. The problem here is that it is difficult to acquire complete and perfect knowledge in a complex domain. Cases are natural and easy to obtain. They can be collected from the historical record, repair logs or other sources [42].

Therefore, the rule based reasoning technique will be used to develop knowledge based computer and network troubleshooting expert system.

## 2.4 EXPERT SYSTEM DEVELOPMENT TOOLS

In the 1980s and early 1990s, when commercial interest in knowledge based system was reach at its peak, approximately there are more than 200commercially available KBS tools. Many are still available but no longer described as KBS tools for marketing reasons. A knowledge based system

tool is a set of computer software that manipulates programs and other information in order to design and assist the development of knowledge based systems [41].

The actual implementation of KBS was based on high level programming languages. However, modern knowledge based system development tools highly depend on their purposes, functionality and some additional features. Based on their purposes, KBS tools are classified as general purpose programming tools such as Java, and framework 1.NET. On the other hand, there are also specific purpose programming languages such as JRULES, CLIPS, JESS (java expert shell system) [45]. In addition programming Language such as C++ provides objects as a mechanism for programmer to control the layout and data structures [4].

Finally, the popular expert system development tool was CLIPS. The programs are represented by list structures. Lisp is the foundation of many expert systems and shells such as CLIPS [44].

There are many knowledge based system tools. According to King [4] different author classified KBS development tools based on their functionality. The simplistic nature and additional feature it provides is used as parameters to select KBS development tools. Expert systems are typically written in special programming languages. The use of languages like LISP and PROLOG in the development of an expert system simplifies the coding process. The major advantage of these languages, as compared to conventional programming languages, is the simplicity of the addition, elimination, or substitution of new rules and memory management capabilities. Some of the distinguishing characteristics of programming languages needed for expert systems work are [4]:

- Efficient mix of integer and real variables
- Good memory-management procedures
- Extensive data-manipulation routines
- Incremental compilation
- Tagged memory architecture
- Optimization of the systems environment
- Efficient search procedures

Prolog (programming in logic) is one of the most widely used programming languages in artificial intelligence research. As opposed to imperative languages such as C or Java (which also happens to be object-oriented) prolog is a declarative programming language. When implementing the solution to a problem we specify what the situation (rules and facts) and the goal (query) then let Prolog interpreter derive the solution. Prolog is very useful in some problem areas, such as artificial

intelligence, natural language processing, databases . . . but it is pretty useless in others domain such as graphics or numerical algorithms[45].

Prolog derives its power from a procedural interpretation of logic. It represents knowledge in terms of procedure and reasoning becomes a simple process of calling the right procedures

[46]. Therefore, there are many widely used prolog programming languages such as Quintus Prolog, Arity Prolog, ALS Prolog, LPA Prolog, and a number of other commercially available as well as freeware Prologs from ESL and SWI. For the purpose of this research study, SWI Prolog programming language is used.

### 2.5.1 SWI Win-Prolog Programming Language

**WIN-**PROLOG provides a complete development environment including easy-to-use pull-down menus, support for multiple edit windows and rich edit (e.g. automatic syntax coloring, multiple fonts in a window, etc.).**WIN-**PROLOG has incremental and optimized compilation, together with a hashed compilation mode which allows a matching clause to be found almost instantly[46].

**WIN-**PROLOG allows you to create polished Windows applications; it provides an extensive range of graphics predicates allowing convenient access to a large number of Windows Graphical User Interface (GUI) functions [61]. WIN Prolog is a programming language which is completely different from other languages. Programming language such as BASIC, C, C++ and PASCAL uses their main techniques or methods to splitting a problem into discrete steps and following those steps in sequential order [40].

Prolog is attractive logic programming language for professionals in the area of knowledge Engineering and Artificial Intelligence. Indeed it has been used in a various real world applications. Some of the most common domain areas are Environmental Modeling, Sales Modeling, medical domain, Fungus Identification, image Recognition, Management Consultancy and Shift Allocation [17].

In Prolog the data and the program are the two different ways of looking at the Prolog objects.

Prolog has the ability to change its program whilst that program is running. Another important aspect of Prolog is that it relates to a logic called predicate calculus. This gives Prolog a number of distinct properties it inherits from logic and gives better foundations than other conventional programming languages. One of the properties is that prolog is a declarative language. This allows one to develop

a program by concentrating on "what" needs to be done instead of "how" it is done. It can also allow a programmer to understand without the need of following through its dynamic execution [46].

## 2.6 MODELS OF KNOWLEDGE BASED SYSTEM

The required system responses are sometimes difficult to describe. A finite state machine might not be able to easily show all responses. What are we then to do? Decision trees and tables have been around for decades. They allow you to visually construct the system. To construct a decision table draw a row for each stimulus that will be used in the process of making a decision. Next draw a column for every possible combination of outcomes of those conditions. If there are n conditions and each has a binary result, then there will be 2n columns. Last, add rows on the bottom of the table for each response you want the system to perform. And Decision Trees are a decision support tools that contains tree like graph of decisions and the possible consequences. They are commonly used in different real world scenarios ranging from operations research to classifying specie in a phylum given its features [12].

A decision tree is graphical rather than tabular. It is a flow chart without loops and fan-in. The decision tree takes up more room than the table but shows the order of evaluating the conditions. Due to these reasons decision tree was used to model the acquired knowledge from primary and secondary sources.

## 2.7 APPROACHES TO EVALUATE THE PERFORMANCE OF EXPERT SYSTEM

Evaluation can be defined as an iterative process of systematic assessment of knowledge based system. The evaluation process carried out at different stage of system development life cycle.

The performance of the system will be assessed or measured through quantitative and qualitative techniques to achieve the expected objective. We can evaluate the KB structure, inference engine, user interface, etc. For this reason, evaluation must follow an order, it has to be planned and it must be controlled to reduce the cost of the final system [48].

Basically, system performance evaluation procedure tends to assess not only the technical aspects of knowledge based system but also user's satisfaction. The evaluation criteria depend up on the purposes of designed knowledge based system in the domain area. Therefore, system's evaluation process tries to evaluate whether a set of rule achieved the expected goal or not [49].

Knowledge based system evaluation process involves to determine the suitability and desirability of the prototype [50]. Effective knowledge based system evaluation process incorporates both technical and non-technical aspects. The technical aspects include exploring of the code, examining the correctness of reasoning techniques, checking the efficiency and performance of the system and debugging errors in the early age of a system development. The non-technical aspect includes system compatible with users' satisfaction, the easiness of the system, the quality of the user interface and the acceptability of the system in the real world environments [39].

### 2.7.1 Methods of Evaluation

Knowledge based systems evaluation method can be split into Verification, validation, assessment of human factors and assessment of correctness. These evaluation methods are discussed as follow [51]:

**Verification** is an evaluation process that should be implemented during system design and development to answer the question 'Did we build the system correctly'. Verification can be defined as the process that involves checking for compliance with the system specifications, checking for syntactic and semantic errors in the knowledge based system. Specification assessment includes user interface, explanation facility, real time performance and security provisions specified in the system design. To verify a knowledge based system, it is possible to use either a program proof or a test strategy. The program proof confirms total correctness of the program logic with mathematical methods and the test proof strategy confirms partial correctness of the program with given test cases (s).

**Validation:** The concept of validation refers to determining the correctness of the system with respect to users' needs. Validation criteria include comparisons with known results (e.g. past cases or solved problem), comparison against expert performance, and comparison against theoretical possibilities. Empirical validation checks whether the results of content remain stable when the system is under full workload. The system test examines the complete system performance in its working environment. Validation tests include user acceptance surveys, direct comparison on random test cases between human expert and that of the system.

**Evaluation of human factors** is the process of determining the acceptability and usability of the knowledge based system. Usefulness of a system is often measured by examining user satisfaction. User satisfaction measured from different point of views such as content satisfaction, interface

satisfaction and institutional objective. Personal aspect such as individuals' dislike of computer takes into consideration [52].

**Evaluation of explanations** is used to evaluate the explanation ability of knowledge based system. According to Stranieri [53], explanation facility was measured based on the following criteria which used to judge the user interaction with the system:

- Naturalness- Explanations should be natural to the end user. Explanations that are not structured according to standard patterns of human dialogue create ambiguous information.
- Responsiveness- An explanation facility must have the ability to accept feedback from the user and provide response for the given feedback.
- Flexibility- An explanation facility must be able to offer brief description in more than one way.
- Sufficiency- An explanation module should be able to answer a range of questions that a users' wishes to ask and not limited to those questions predicted by developers.
- Sensitivity-An explication module should take into account the user's goals, the problem domain and the previous explanatory dialogue.
- Fidelity/accuracy- An explication system must accurately reflect the system's knowledge and reasoning.

## 2.8 APPLICATIONS OF EXPERT SYSTEM

Expert systems address areas where combinatory is enormous. The following are some of the application areas of expert system [54]:

- Highly interactive or conversational applications, voice server, chatterbox
- Fault diagnosis, medical diagnosis
- Computer and electronic devices diagnosis
- Decision support in complex systems, process control, interactive user guide
- Educational and tutorial software
- Logic simulation of machines or systems
- Knowledge management
- Constantly changing software.

They can also be used in software engineering for rapid prototyping applications (RAD). Indeed, the expert system quickly developed in front of the expert shows him if the future application should be programmed. Indeed, any program contains expert knowledge and classic programming always begins with an expert interview. A program written in the form of expert system receives all the specific benefits of expert system, among others things it can be developed by anyone without computer training and without programming languages. But this solution has a defect: expert system runs slower than a traditional program because it consistently "thinks" when in fact classic software just follows paths traced by the programmer. Expert systems are designed to facilitate tasks in the fields of accounting, medicine, process control, financial service, production, human resources, among others[54].

In 1970s and 1980s application of expert systems, which we today would simply call AI, was in computer games [54]. For example, the computer baseball games Earl Weaver Baseball and Tony La Russa Baseball each had highly detailed simulations of the game strategies of those two baseball managers. When a human played the game against the computer, the computer queried the Earl Weaver or Tony La Russa Expert System for a decision on what strategy to follow. Even those choices where some randomness was part of the natural system (such as when to throw a surprise pitch-out to try to trick a runner trying to steal a base) were decided based on probabilities supplied by Weaver or La Russa. Today we would simply say that "the game's AI provided the opposing manager's strategy" [54].

## 2.9 RELATED RESEARCH WORKS

For the purpose of conducting research on computer and network troubleshooting expert system different technical and research works will be reviewed. Some of them are reviewed as follows:

Manda, Chatterjee and Neogi [3] introduced a system for diagnosis and troubleshooting of computer faults based on expert system and artificial intelligence. The system is developed to detect hardware fault detection in any computer system. Different hardware system or electronics devices usually face application fault as well as hardware fault. This application oriented formal reasoning applies to determine the problem in computer system. This paper emphasizes an automated system that accepts the defects of any system and then after consulting with an intelligent database, diagnoses and advises for probable rectification. This is the extensions of the already published work by the authors implemented by Turbo Prolog programming language. This paper extends that

concept exploring the boundary of Expert System targeting the implementation using Object Oriented web based programming approach [3].

**DART**: An Expert System for Computer Fault Diagnosis; is a system which advises IBM field service personnel on the diagnosis of faults occurring in computer installations. The consultant identifies specific system components (both hardware and software) likely to be responsible for an observed fault and offers a brief explanation of the major factors and evidence supporting these indictments. The consultant, called DART, was constructed using HMYCIN, and is part of a larger research effort investigating automated diagnosis of machine faults. The DART knowledge base consists of 300 EMYCIN parameters and 190 production rules and was constructed over a period of 8 months. During this period 5 specialists were interviewed about different aspects of the diagnostic process and the knowledge base reflects their composite expertise. 50% of diagnostic processes were correctly responded from online exploration [8].

Expert systems use human knowledge often stored as rules within the computer to solve problems that generally would entail human intelligence. Today, with information systems turning out to be more pervasive and with the myriad advances in information technologies, automating computer fault diagnosis is becoming so fundamental that soon every enterprise has to endorse it. This expert system called Expert PC Troubleshooter for diagnosing computer problems. The system is composed of a user interface, a rule-base, an inference engine, and an expert interface. Additionally, the system features a fuzzy-logic module to troubleshoot POST beep errors, and an intelligent agent that assists in the knowledge acquisition process. For implementation ASP.NET 4.0 and C#.NET 4.0 under the MS .NET Framework 4.0 and MS Visual Studio 2010 is used. The system is meant to automate the maintenance, repair, and operations (MRO) process, and free-up human technicians from manually performing routine, laborious, and time consuming maintenance tasks. PC (IBM Personal Computer) troubleshooting covers a wide spectrum of problems including hardware problems, software problems, network problems, server problems, operating system problems, and application software problems. This would impose a maintenance nightmare for large-scale enterprises and IT infrastructures as the number of possible technical problems and finding their solutions can become very huge and complex [9].

The usage of the computer is widely spread today but the knowledge in the fault diagnosis is very limited. Thus, the fault diagnosis of the computer components is very important. Obviously, the risk of hardware failure is the most commonly talked-about reason to perform backups. Indeed, nothing will jolt someone into realizing the importance of backups more than an unrecoverable hard disk

failure. Since the hard disk stores your main programs and data, it is the hardware whose failure hurts the most. It is also what gets the most attention, and rightly so. However, there are other hardware problems that can cause permanent data loss and some of these can be rather hard to figure out, since they don't seem like they should be responsible for the problem. For instance, memory error with so many systems today running without error detection or correction on their system memory, there is a chance of a memory error corrupting the data on the hard disk. It is rare for it to happen, but it does happen. Accordingly, the need of creating an expert system to diagnose these faults is increasing. Here in this paper the design of an expert system that aims to help the user in diagnosing the failure in motherboard, RAM and CPU is represented. "Empirical Model using Expert System Techniques in Hardware Failure of a System during Backup of Data" was programmed with the CLIPS programming language. [54].

The widespread use of network requires specialized professionals to deal with issues related to network management, hardware failure, system administration as well as maintenance. Yet, professionals with the required expertise are difficult to find, leading to problems such as increasing maintenance cost, poor network performance as well as users' work performance. This paper presents the first phase of an on-going expert system project that aimed to support the network administrator in troubleshooting the network hardware problems. Knowledge engineering method is used to guide the development process; knowledge is acquired from one network experts using traditional interviewing method. The knowledge acquired from the source is presented in a graphical form such as cognitive map and flowcharts. The system is expected can be executed in a PDA device to ease the administrator during troubleshooting the network problems. To develop this expert system active server page (ASP) was used [9].

This paper describes a tool to diagnose the cause of failure of a HP computer server. We do this by analyzing the dump of Processor Internal Memory (PIM) and maximizing the leverage of expert learning from one hardware failure situation to another. The tool is a rule-based expert system, with some nested rules which translate to decision trees. The rules were implemented using a meta-language which was customized for the hardware failure analysis problem domain. Pimtool has been deployed to 25 users as of December 1996. We plan to expand usage to over 400 users by end of 1997 by using stratified sampling. Using Pimtool, we expect to save over 15 minutes in Mean-Time-to-Repair (MTTR) per call. We have recognized that knowledge management will be a key issue in the future and are developing tools and strategies to address it [56].

In troubleshooting Computer Systems the two most common causes of delay are Trial and Error and having Incomplete Information. The problems in Computer Systems will be fixed faster if the Possible Cause of the Problem is already known. A solution to this is to use an Expert System. This system can reproduce the ability of an expert to diagnose by giving an accurate recommendation on the possible cause of the problem for effective troubleshooting. This research solved the two most causes of delay which are Trial and Error and having Incomplete Information. This is done by developing an Expert System Algorithm that creates the rules of an Expert System by using tool called hypertext preprocessor(PHP). The rules created from the algorithm are nominal in terms that only the necessary information needs to be inputted. By using test_case evaluation from the imputed 60 causes 56 were correctly responded which is 93.3%. In instances that the data gathered are incomplete the correct Possible Cause can still be suggested. A theorem is also presented in this research about and the Information Dependency of Data which can be used with Incomplete Information Systems and unknown data. Formal Proof of the theorem is provided and its correctness was verified with actual data [1]. In this related work the method is not specified.

The present era is the information age, where faster transfer and retrieval of information is the most vital need. The chief requirement of this era is that all communicating devices should be up and running. In the current world, the most popular communicating device is the personal computer and its downtime will prove fatal for the corporate competence. Apparently most of the computer users are amateurs when it comes to the area of troubleshooting, a predicament that arose in computer hardware. One such computer hardware that is essential part of every office is printer. If the printer in the office or at home is malfunctioning, then the situation requires a hardware specialist. However, using an expert system in this scenario will be an economic and rapid solution and will overcome the need of a hardware expert. This will help to increase individual's efficiency by diminishing his/ her downtime. This paper presents a new expert system named "Printer Troubleshooting Expert System" for troubleshooting printer issues in MS-Window based Personal Computers. For the purpose of data collection 3 experts are included by using purposive sampling and ASP.NET with SQL is used for implementation. If else statement is used for representation in coding and user acceptance and test case validation is used [57]. The result is not clearly defined in this work.

On a daily basis, users connected to the Internet confront network communication problems. Most of the time these problems are common and easy to solve for experienced system administrators, On the other hand, they tend to be a real nuisance to inexperienced ones. As a consequence, there is the need

for expert knowledge that could assist inexperienced system administrators and end users in approaching and solving such problems [58].

Internet users who encounter network communication problems sometimes do not have the experience or expertise available on site to approach and solve those problems. NeXPert is a WWW-based expert system advisor developed with the intention of providing access to expert knowledge to a broad user community. It provides advisory knowledge to help users solve these problems. It was implemented with freely available tools such as CLIPS, JESS and Java. The WWW-based design proves the feasibility of developing expert systems in a distributed environment and gives access to the knowledge base to the broad Internet community [58].

Computers have simplified our working life to be more efficient and productive; for example, communication can be done through mail and instant messaging, presentation can be shared through Net-meetings and meeting minutes can be stored in workspace or shared drives. Computers, especially notebooks and laptops are very synonymous with mobile lifestyles as it is very convenient. LAN and wireless facilities which are allow users to access borderless information at anywhere and at any time, turn the computer into a necessity rather than a luxury. However these gadgets sometimes malfunction and not every computer owner is a troubleshooter or able to find out the root cause of the malfunction. This is where the expert system is needed for help to any user as a first aid approach to identifying and solving the problem. This paper presents one of such expert system designs and is limited to PC hardware maintenance and troubleshooting covering to the Monitor, Hard drive, Keyboard, Mouse, Audio device, and Missing/Lost Documents via a rule base diagnostic system using PHP [59]

Finally, from the related works reviewed, the researcher found important points such as methodologies used to develop the knowledge based system, tools used for the implementation purpose, modeling and knowledge representation, performance evaluation techniques and discussion of the findings. The above related works in the computer domain are providing computer troubleshooting information for the user and computer expert from the expert's knowledge stored in the knowledge base of the knowledge based system. The gap identified by the researcher was, all of the works advice the user and human expert to troubleshoot the problems but they do not have the ability to learn new problems occurring in the computer and network system and do not update the database of the expert system knowledge base.

| Author | Title | Tool | Method | Techniques | Evaluation and result |
|---|---|---|---|---|---|
| S. Manda, S. Chatterjee, B. Neogi. | Diagnosis And Troubleshooting Of Computer Faults Based On Expert System And Artificial | Turbo Prolog | Not specified | Object Oriented | User acceptance |
| J. S. Bennet. | **DART**: An Expert System for Computer Fault Diagnosis | prolog | Purposive (interview) | Rule based | Performance (50%) |
| Y. Bassil | Expert PC Troubleshooter for diagnosing computer problems. | ASP.NET, C#, V. studio 2010 | Interview(sampling tech. not specified) | Rule based | Not specified |
| D. M. Africa. | An Expert System Algorithm for Computer System Diagnostics. | PHP | Interview | Not specified | Performance (93.3%). |
| D. Narendra and A. Bart. | Pimtool: an expert system to Troubleshoot Computer Hardware failures. | prolog | Stratified sampling | Rule based | Performance (50%) |
| A.Kaushik and M. Barnela | Printer Troubleshooting Expert System | ASP.NET with SQL | Interview (purposive sampling) | Not specified | Not specified |

**Table 2.1 Summary of Related Works**

As indicated in table 2.1, the highest result from the user validation is 93.3 % for An Expert System Algorithm for Computer System Diagnostics. The two related works performance is 50% and the remaining works result is not clearly specified. The techniques used to design the system were rule based reasoning, Object Oriented and the remaining related works technique is not clearly specified. Most of the works used interview question to acquire knowledge from the domain experts and different implementation tools were used.

# CHAPTER THREE

# KNOWLEDGE ACQUISITION AND CONCPTUAL MODELLIING

Knowledge acquisition (KA) is the process of acquiring relevant knowledge from domain experts and other sources of information such as books, databases, guidelines, manuals, journal articles and computer files. KA is the process of eliciting, structuring and representing (formalizing) domain knowledge acquired from different sources. The term knowledge acquisition and knowledge elicitation have been used interchangeably in the field of artificial intelligence (AI) literature. The acquired knowledge can be specific to the problem domain, it can be general or it is meta-knowledge (knowledge about knowledge). Knowledge acquisition is the first step and time consuming task in the development of knowledge based system [5].

There are certain important steps that the knowledge engineer need to carry out during knowledge acquisition process [10]:

i. Eliciting data and information from the domain experts
ii. Interpreting the acquired information to understand human expert reasoning processes
iii. Constructing model to structure the expert's knowledge
iv. Represent knowledge and build knowledge base
v. Repeating step I-IV as the knowledge base system involves into a functional system

## 3.1. THE PROCESS OF KNOWLEDGE ACQUISITION

For the purpose of this research, the process of knowledge acquisition includes some basic activities such as interviewing of domain expert's, review of relevant sources of information and observing when the expert is troubleshooting.

The objective of knowledge acquisition is to gather the required knowledge, interpreting the acquired knowledge, analyzing and validating the knowledge content. Based on the acquired knowledge, the KBS was designed using decision tree model.

Therefore, knowledge acquisition process of this thesis was based on domain expert interviewing, observing and reviewing of related documents.

### 3.1.1. Interviewing Domain Experts

Primary sources of knowledge were collected from human experts in the domain area at Jimma University ICT office. To gather the required knowledge semi-structured interview technique was used. Since one of the main focuses of this research is eliciting relevant knowledge from the domain experts. Four (4) domain experts were selected using purposive sampling technique. These experts were selected due to their managerial position and experience in the ICT office. This sample was taken due to limited number of experts in the managerial position of the ICT office Jimma University. During the interview phase, the main challenge was willingness of domain expert to share their expertise and experience, which was important to acquire the relevant knowledge. The domains of interview with expert covered issues such as how the experts diagnose computer and network problems, what are techniques used to identify the pillar symptoms of the problem, the procedures of diagnosing and what are the possible solutions recommended to the problems.

During the extensive discussion, the researcher acquired the relevant knowledge which was significant to generate the rules. In addition, the domain experts were actively participated throughout the research work and they were consulted to confirm the correctness of the acquired knowledge. During face to face communication, the acquired knowledge from domain experts was recorded manually by using pen and paper sheet. Profiles of domain experts participated in the interview process were presented in the Table 3.1 below:

| No | Educational level | Specialization | Area interviewed | Position |
|---|---|---|---|---|
| 1 | Msc. | ICT | Computer and network | ICT director |
| 2 | Bsc. | ICT | Network | Network administrator |
| 3 | Bsc. | ICT | Computer | ICT technician |
| 4 | Bsc. | ICT | Computer | ICT technician |

**Table3.1. domain experts' profiles**

As indicated in the Table 3.1 above, the computer professionals were devoted to providing computer diagnosing services in their institution. In addition, they participate in conducting research and training computer users in the University. The semi structured interview questions result was discussed below:

The first question is “how computer problems are defined from the aspect of ICT technicians’?” the computer problems are not known for the user but there are some changes in screen, startup sound, computer speed, sudden restarting and overheating. So there is no common definition for computer problems.

The second question is “What are the methods and techniques ICT experts use to identify the causes of computer problem?” According to the domain experts, the investigation of computer problem starts by collecting some relevant information such as software and hardware installed recently, any configuration made in BIOS, any change made in system file, new symptoms which are not seen before, beep sound of the computer, LED light, and screen display.
The third question is “How the ICT technicians identify the unique characteristics of each problem?” Symptoms were important attributes to determine the types of network and computer problems; because some problems have unique associations with beep sounds, screen displays and LED lights. The most common problems occurring in the computer system were discussed below:

Although most complex computer issues at work can often be solved by the IT support team, there are many other small, but common, issues that occur on a regular basis on a personal computer. The good news is that many problems with computers have simple solutions, and learning to recognize a problem and fix it yourself will save you a lot of time and money. The following are the most common computer problems that you shouldn’t panic over.

The fourth question is “What are the common types of problems that occur in the computer?”
**The Computer Won’t Start:** A computer that suddenly shuts off or has difficulty starting up could have a failing power supply. Check that the computer is plugged into the power point properly and, if that doesn’t work, test the power point with another working device to confirm whether or not there is adequate power.

**The Screen is Blank:** If the computer is on but the screen is blank, there may be an issue with the connection between the computer and the screen. First, check to see if the monitor is plugged into a power point and that the connection between the monitor and computer hard drive is secure. If the problem is on a laptop, then you may need to get a professional to fix it as some of the internal wires may be worn.

**Abnormally Functioning Operating System or Software:** If the operating system or other software is either unresponsive or is acting up, then try restarting your computer and run a virus scan. To avoid having this happen, install reliable anti-virus software.

**Windows Won't Boot:** If you are having troubles booting Windows, then you may have to reinstall it with the Windows recovery disk.

**The Screen is Frozen**: When you computer freezes, you may have no other option than to reboot and risk losing any unsaved work. Freezes can be a sign of insufficient RAM, registry conflicts, corrupt or missing files, or spyware. Press and hold the power button until the computer turns off, then restart it and get to work cleaning up the system so that it doesn't freeze again.

**Computer is Slow**: If your computer is slower than normal, you can often fix the problem simply by cleaning the hard disk of unwanted files. You can also install a firewall, anti-virus and anti-spyware tools, and schedule regular registry scans. External hard drives are great storage solutions for overtaxed CPU's, and will help your computer run faster.

**Strange Noises:** A lot of noise coming from your computer is generally a sign of either hardware malfunction or a noisy fan. Hard drives often make noise just before they fail, so you may want to back up information just in case, and fans are very easy to replace.

**Slow Internet**: To improve your Internet browser performance, you need to clear cookies and Internet temporary files frequently. In the Windows search bar, type '%temp%' and hit enter to open the temporary files folder.

**Overheating**: If a computer case lacks a sufficient cooling system, then the computer's components may start to generate excess heat during operation. To avoid your computer burning itself out, turn it off and let it rest if it's getting hot. Additionally, you can check the fan to make sure it's working properly.

**Dropped Internet Connections**: Dropped Internet connections can be very frustrating. Often the problem is simple and may be caused by a bad cable or phone line, which is easy to fix. More serious problems include viruses, a bad network card or modem, or a problem with the driver.

**Network problems:** Commonly occurring problems in the LAN are:

**Cable connection problem**: it is mandatory to check the computer is on and the network cable is connected to the computer.

**IP address problem**: when configuring the computer to access the network IP address should be assigned to the computer unless the computer cannot access the network. Before assigning the IP

address the IP address used in that company should be identified first. It may be static or dynamic. To avoid IP address conflict (is static) the range of IP address used in that institution should be identified.

**Network driver problem**: the availability of network driver should be checked first before trying to access the network because without the driver accessing network is impossible. This problem will be solved by installing the network driver from CDs which are coming with the computer.

The fifth question is "Does the problems have common behaviors and their complication due to different factors?" as the computer system is following the standard most of the problems are occurring in the computer system are common. But limited number of problems may be different due to complexity and constant change in the technology.

The sixth question is "What are the decisions of the ICT technician in troubleshooting computer and network problems?" if the problems are identified the ICT technicians' tell the user about the problems and risks before troubleshooting the problems. Then the technicians' decide to maintain the problems after holding back up of the data in the computer system and after ensuring the availability of troubleshooting tools. These troubleshooting tools are hardware or software diagnostic tools which aid in maintaining the computer problem.

Question seven "What are the most important areas focused by the ICT technician to diagnose the computer problems?" all areas in the computer system are important and need focus. So in troubleshooting sometimes attention may be given to sensitive information and more focus is given to the storage device that handles sensitive information.

The last question is about recommended techniques used to manage specific problems. To handle the occurrence of the problems it is specific or general, preventative maintenance is required which protects the computer from serious attacks. If the computer is not followed preventative maintains schedule the failure of the computer is high.

The actual diagnosing task of the experts start by using a diagnosing instruments (software or hardware) which are called diagnosing tools. These tools are used to investigate problems of the computer affected by different reasons.

### 3.1.2 Observation

To find additional information in addition to document analysis and interview questions, observation was used when the human expert is troubleshooting the problems occurred in the computer system. From this activity the researcher identified, the prerequisite questions asked before starting

troubleshooting process, the procedures followed by the expert in troubleshooting process, and so on were observed.

## 3.2 GENERAL STRUCTURE OF CNTES

The computer and network troubleshooting expert system is complex and dynamic, for the purpose of this study the problems are categorized into PC and Network problems.

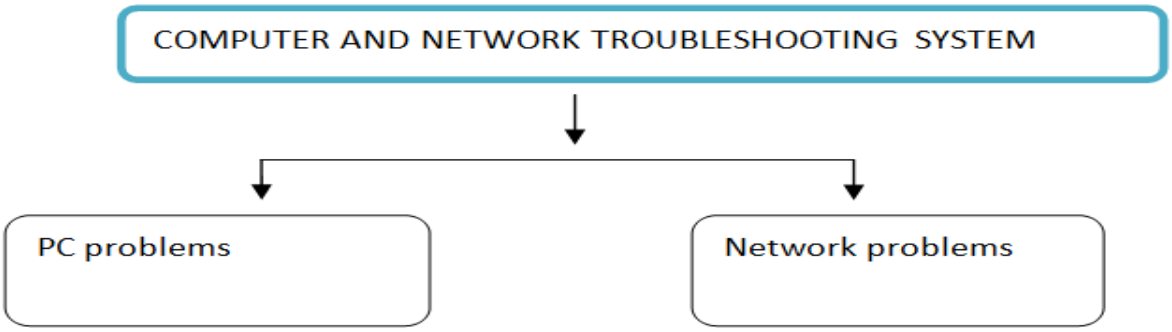



**Figure3.1. General Structure of CNTES**

The above figure 3.1 structure indicates the general classification of problems occurring in the computer system. In PC category there are different problems and in the similar way network category.

## 3.3 THE COMPONENTS OF CNTES

Figure 3.1 above indicates, the expert system is categorized into PC and network problems. For the purpose of this study sample of problems (causes) were selected from each category. For example, from PC problems (Hard disk, sound and power problems) were selected. From the network problems (cable, configuration and driver problems) were selected. The problems were verified by using the evaluation method called verification to remove irrelevant information from the data collected from domain experts by using interview technique, observation and document analysis. The verified problems, causes and solutions depicted in table 3.2 below. The framework is adopted from Expert PC Troubleshooter with Fuzzy-logic by Bassil [5].

| Problem category | Problem type | Cause | Solution |
|---|---|---|---|
| Hard disk | Invalid drive specification error | Drive has not been formatted | Run FDISK to create partition |
| | Invalid drive type error | Drive not yet formatted | Format your drive |
| | SMART warning displayed | Serious mechanical problems are detected | Backup and replace your drive |
| | No drive | Data cable is disconnected | Connect the data cable |
| | No drive | Power cable is disconnected | Connect power cable |
| | | | |
| **Sound problem** | Sound Card can't be Detected | Damaged or Sound Card Not Installed | Replace Sound Card |
| | Driver Warning | Driver Conflict or Incompatible Driver | Install The Appropriate Driver |
| | Scratchy Sound | Signal Interference | Stay Away from Radio Frequency Sources |
| | Speaker or Microphone won't Work | Incorrect Jacks | Use Proper Jacks |
| | | | |
| Network problem | No connection | Cable is not connected | Connect network cable to your computer |
| | No IP address | IP address is not assigned | Assign IP address to your computer |
| | No connection | Network driver is not installed | Install network driver |
| | | | |
| Power problem | No power | Source power is off<br>Computer power button is off | Switch ON source power<br>Switch ON PC power button<br>Check power cable |
| | No LED light | Internal power cable is disconnected | Connect internal power cable |
| | No sound from power supply | Short in power cables<br>Damaged power supply | Check each power supply cables Change power supply |

Table 3.2 computer and network problems

## 3.4 CONCEPTUAL MODELING

Conceptual Modeling of domain knowledge implies capturing the static structure of information and knowledge types [60]. Decision trees (DTs) are modeling tools that are used in a variety of different settings to organize and break down clusters of data [61].

Similarly, decision tree have been widely used in practical applications area, due to its interpretability and ease of use [62]. Currently, decision trees are used in many disciplines such as medical diagnosis, cognitive science and computer diagnosis [63]. The decision tree was used in computer domain to understand the dimension of the problem. Each tree starts with a set of computer problems and ends with solutions.

### 3.4.1 CNTES and the Decision Tree Structure

CNTES is the systematic representation of computer problems using decision tree structure.

As a result, representing computer and network problem based on decision tree structure helps the ICT technician or computer user to identify each problem with little effort. Therefore, the system improves the computer troubleshooting service; minimize the burdens of technicians, fill the gaps of limited manpower, delay of time and backlog.

The ICT technicians acquire required information from the computer user. To make this acquired knowledge functional in knowledge representation, the knowledge based system is modeled by using decision tree structures.

Decision tree structures are the bases for the development of prototype knowledge based system. The prototype follows the same procedures as presented in the decision tree when diagnosing computer problems.

The common causes of each problem is used to determine the presence of specific cases that has been illustrated from figure 3.2 to figure 3.5 using decision tree structures. Based on the decision tree structures, some problems which have similar characteristic in nature were incorporated in the knowledge base to show that some causes can also be causes for other problems. Therefore, similar procedures have been followed for all problems occurring in the computer system. Some of the causes were modeled by using decision tree as follows:

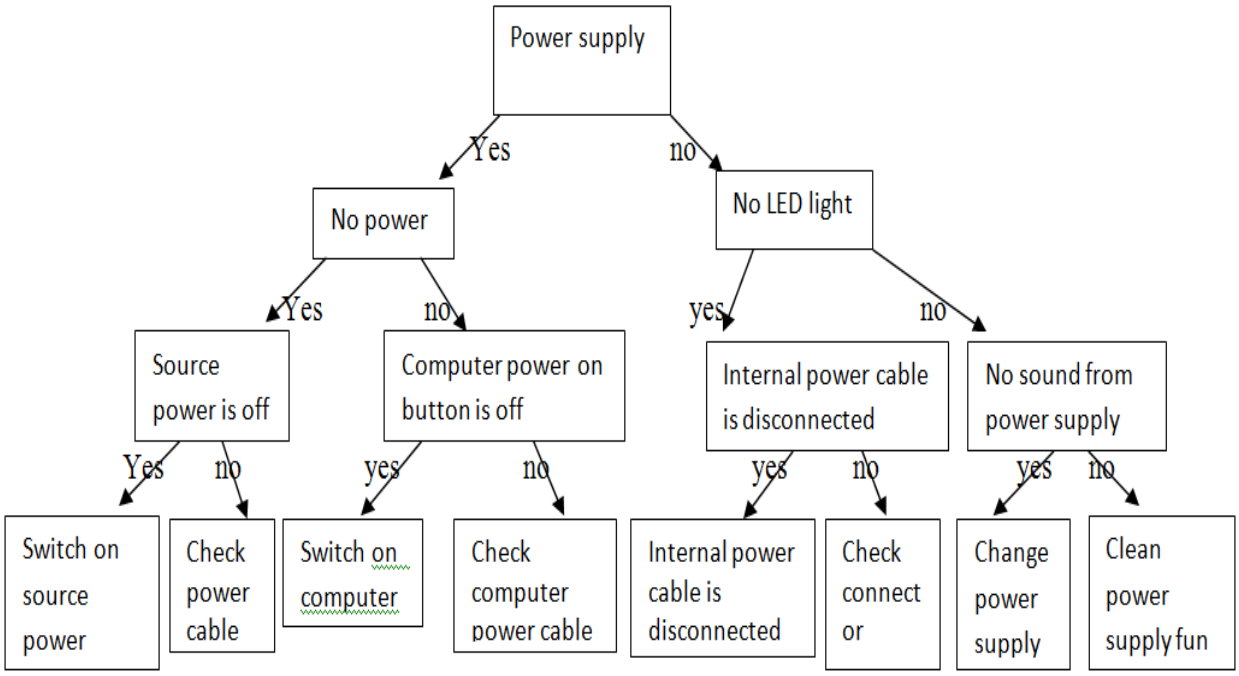



**Figure3.2. Decision tree for power supply problem**

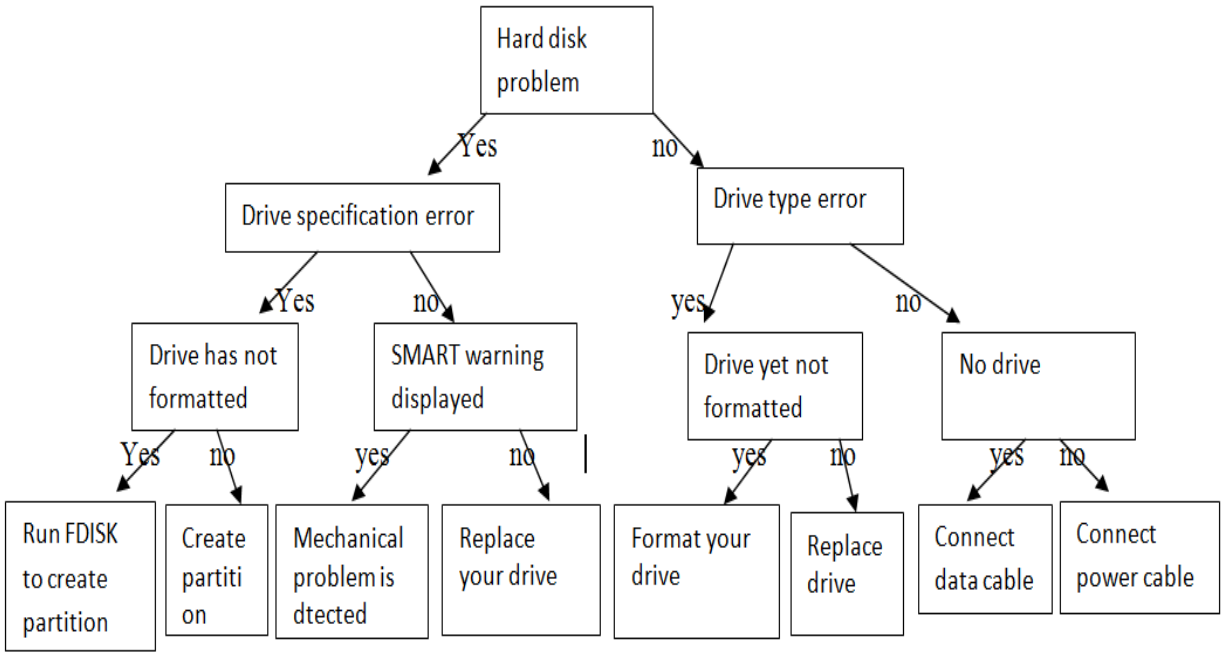



**Figure3.3. Decision tree for Hard Disk problem**

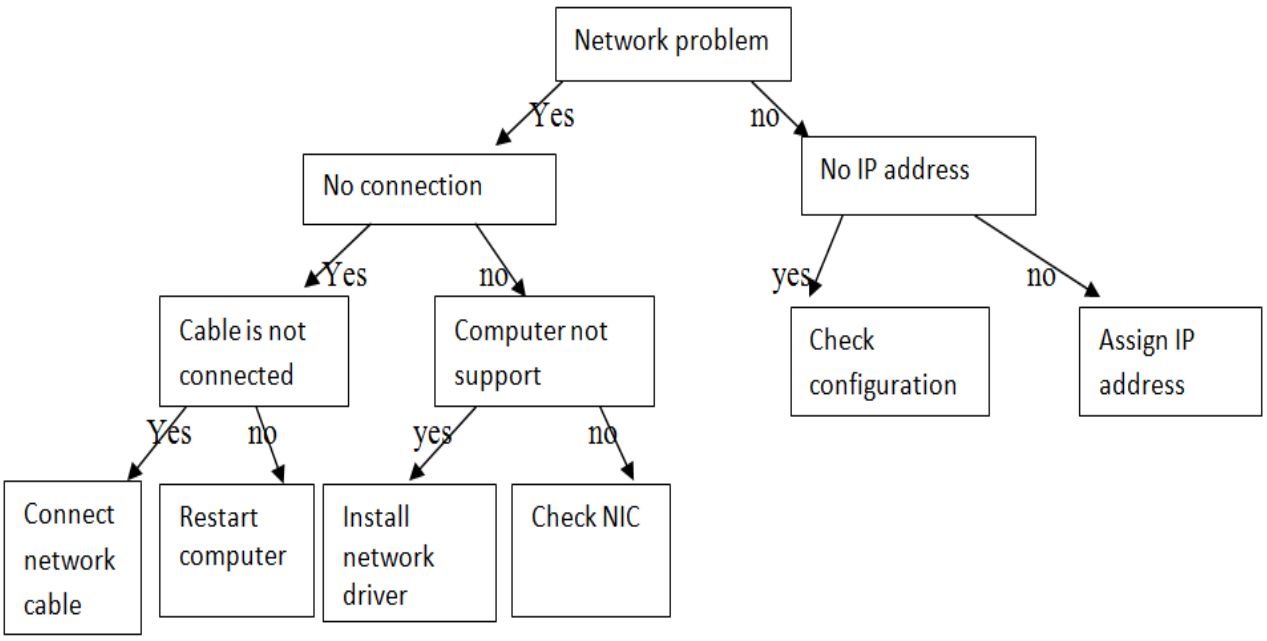



**Figure3.4. Decision tree for Network problem**

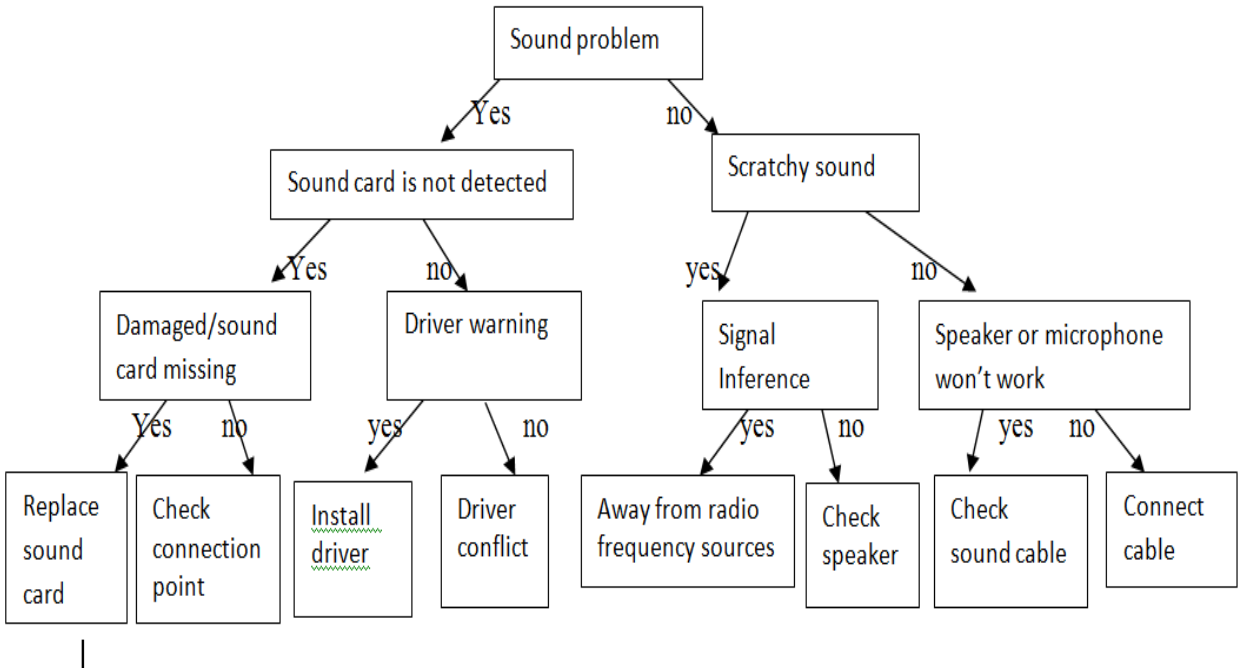



**Figure3.5. Decision tree for sound problem**

## 3.5 KNOWLEDGE REPRESENTATION

Knowledge representation is one of the basic steps in the process of knowledge based system development. Knowledge representation is the process of interpreting domain knowledge into computer understandable form by using knowledge representation methods.

The acquired domain knowledge is represented as a set of "IF – THEN" rules in the prototype.

The "if" side (also known as the left hand side) of the equation which states the condition(s) that must be true in order to fire the rule and the "then" side (right hand side) of the equation specifies the appropriate action to be taken. The inference engine evaluates the "if" portion of a statement and concludes whether a goal is satisfied or not. If the goal is not satisfied, the inference engine proceeds to the next rule until the conditions are satisfied.

A rule is a conditional statement that links the given conditions to actions. Rules in the knowledge base were constructed based on the decision tree structure on conceptual model discussed above.

To make easy and understandable prolog rules, the acquired knowledge from the domain experts represented using the "IF-THEN" form. The rules are the base for the construction of KBS. The following are sample rules which are incorporated in the knowledge base.

If the computer has no power,

The power source And/Or the computer power On button is off,

Then switch On source power And/Or switch On your computer.

If the computer has no LED light,

The internal power cable is disconnected,

Then connect the internal power cable.

If there is no sound from the power supply,

Damaged fuse And/Or damaged power supply,

Then change the power supply.

If the hard disk drive specification error,

Drive has not been formatted,

Then run FDISK to create partition.

If SMART warning displayed,

Serious mechanical problems are detected,

Backup and replace your drive.

If hard disk drive type error,

The drive not formatted,

Then format the drive.

If no drive,

Power cable is disconnected,

And/Or data cable is disconnected,

Then connect,

Data cable And/Or power cable.

If your computer has no connection,

Network cable is not connected And/Or

Network driver is not installed,

Then connect network cable And/Or

Install the network driver.

If the computer has no IP address,

The IP address is not configured,

Then configure the IP address.

Similar procedures have been used for all rules incorporated in the knowledge base.

To summarize the chapter, the transformation process of the acquired knowledge from human expert, document analysis and observation is helpful to investigate the specific attributes which are significant in the process of computer diagnosis. Once the relevant knowledge is acquired then it is represented in human understandable manner to arrive at the expected goals. Therefore, the knowledge based system was implemented by articulating the rules.

# CHAPTER FOUR

# KNOWLEDGE BASED SYSTEM IMPLEMENTATION

The most important criterion for any KBS is the accuracy of their inference engines. Also important are, robustness of the system when some information is missing, redundant and inconsistent. Human readability helps the domain experts to understand knowledge content and even to modify its content. The system's dynamic nature to adopt a new knowledge from the user response is critically important to measure system performance.

The knowledge based system implementation depends on the decision tree structure. Because of their simplicity, decision trees or generic rule-based models are commonly used to describe the knowledge acquired from domain expert. The if-then rules of KBS are generated based on the decision tree structure.

However, the main challenge of implementing a KBS was choosing the appropriate representation method. Too large rules may reduce the performance of the system because, as the numbers of rules increases the inference engine fails to infer from the complex rules. The KBS designed for computer and network troubleshooting incorporates knowledge base, inference engine, learning component, user interface, and explanation facility components as shown in figure 4.1.

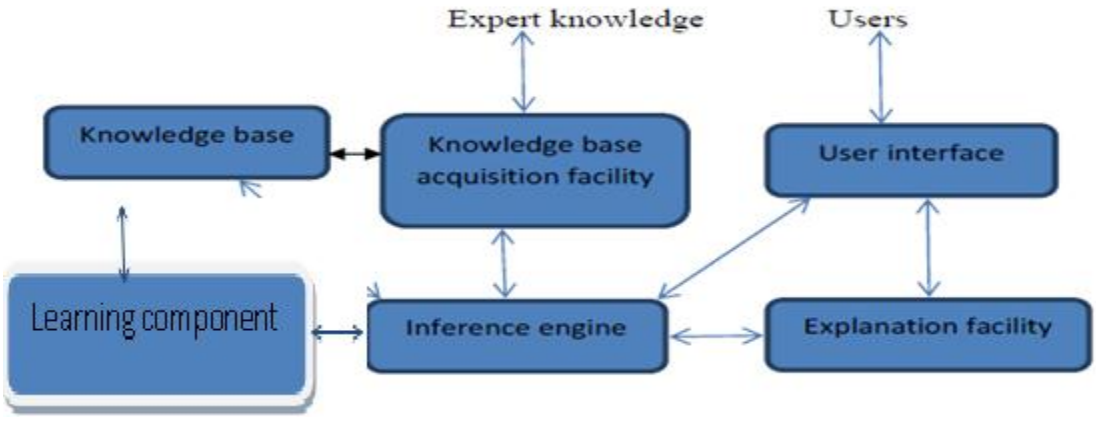



**Figure4.1.Architecture of rule based system**

The goals of any systems are to insure that the final result of the system was usable and practically applicable for the end user. To achieve this goal, the KBS was divided into different subtasks. To

accomplish this task, the inference engine starts from a goal by selecting a rule; searches back to find the facts that satisfy the given condition in the knowledge base. If the conditions were satisfied then inference engine fire the rules accordingly. If the rules satisfy all the conditions then the system draw a conclusion for the given cause. The same procedures are applied for all rules incorporated in the knowledge base. Therefore, the different components of KBS were discussed as follows:

## 4.1. THE KNOWLEDGE BASE

The knowledge base stores all relevant knowledge, fact, rules, and relationships used by the KBS. The knowledge base incorporates the relevant knowledge that was acquired from the domain experts. The knowledge base of the prototype contains the domain knowledge which is used to identify the types of problems, symptoms and solutions in PC troubleshooting services. Based on this, the knowledge base was constructed by using SWI-WIN prolog implementation tool in the form of 'if then' rule. The reason is that, prolog is open source software and it is the preferred programming language for developing a KBS. The advantage of expert systems over conventional programs is that their core algorithm is not encapsulated in the programming code but stored as knowledge in an independent database called knowledge-base or KB. In consequence, there is no need for the expert system to be reprogrammed and recompiled every time the knowledge changes. Prolog makes the representation of the solution much easier, allowing one to concentrate on solving the problems rather than coding the solution.

Prolog contains a number of features which are not found in the normal programming languages, which make it very powerful in the logical programming [6]. These features include:

- A powerful search and backtracking inference mechanism,
- A powerful built-in 'pattern matching' mechanism,
- A simple but sophisticated data structure with the ability to manipulate the data structure

In addition, the researcher is familiar with prolog software.

The fact base component of knowledge based system includes basic facts of different cases that are handled during problem solving. The number of facts depends on the number of rules incorporated into the knowledge base. Functionally, the facts in the fact base are used to compare against the "if" (condition) part of rules stored in the knowledge base.

In other words, if the given facts satisfy all the conditions which proved to be true, then the inference engine draw a conclusion. This is based on the pattern matching between the facts in fact base and their respective rules in the knowledge base.

## 4.2. THE INFERENCE ENGINE

The inference engine simulates the domain expert reasoning process. It works from the facts in the working memory (fact base) and stored knowledge in the knowledge base to fire the rule. It achieves the goal by searching through knowledge base to find rules whose premises match with the given facts in working memory. The searching process continues until the inference engine unable to match any premise with the facts in the working memory.

As the result, the prototype system uses backward chaining reasoning mechanism. During the reasoning process, the inference engine start from the consequence (from the problem occurred) and checks the reasons of the occurrence of this problem to provide suggestions for the problem. If certain antecedents (cases) are evaluated as true, then it logically follows the consequent are proved, and then the problem type, cases and solutions for the problems are provided.

As the model indicated in the decision tree (chapter three) of figure 3.2, during the diagnosis of Power supply problem the system first asks the cases of the problem (no power, no LED light, source power is off, power on button is off, internal cable is disconnected). Next the general practitioner tries to prove whether these causes are match with the causes in the knowledge base or not. Then the system provides suggestion to the ICT technician or computer user depending on the actual causes feed to the system. The inference engine of the rule based system follows similar procedures like the general practitioner(s). The inference engine sequentially searches each rules, if match is found, i.e., if computer has power supply problem, hard disk problem, network problem, and/or sound problem, then the inference engine draw the conclusion for the problems.

The rules that used to diagnosing network problems in **SWI WIN** prolog programming language are represented as indicated below.

Rule 1: network_problem (Cable):-

Your computer has no connection ==(yes/no);

Is the network cable connected == (yes/no);

From Rule1 above, if the given reason satisfies the given conditions, i.e., if the computer has no connection and network cable is not connected the expert system draws a conclusion. The conclusion is to recommend the User or ICT technician to connect the network cable and check the connection. But, if the network cable is connected and still the computer is not connected to the Internet the system requests farther question to solve the problem. Then the inference engine start sequentially searching until the user response satisfied the condition.

Rule 2: ip_problem (Ipaddress):-

Your computer has no connection ==(yes/no);

Is the network cable connected == (yes/no);

IP address is not assigned == (yes/no);

In this rule 2 the computer is not connected to the network but the network cable is connected to the computer. The inference engine looks for farther reason to reach to the conclusion. The IP address is assigned to the computer or not. If the IP address is assigned and the user response is yes the inference engine go to the conclusion. In the actual situation the network cable is connected to the computer and the computer has IP address there should be connection. If not, the problem is not configuration problem. So, the system suggests the user/ICT technician to communicate the network administrator to check the availability of the network.

Rule 3: driver_problem (NIC):-

Your computer has no connection ==(yes/no);

Is the network cable connected == (yes/no);

The computer has IP addresses == (yes/no);

The network driver is not installed == (yes/no);

From the above rule 3, the computer is not connected to the network but the network cable is connected to the computer. The response to the next case is no (the computer has no IP address), in actual situation the problem will be IP address configuration or the network driver is not installed. So, the inference engine looks the next reason from the knowledge base to reach to the conclusion which is “the network deriver is installed or not and IP address configuration”. If the network driver is not installed the expert system suggest that “install the network deriver to your computer then assign IP address which is used in your institution.” But, the network deriver is installed still the

computer is not connected to the network the system suggest the user to go to the network administrator to recognize the institutional network is functioning or not.

The main challenge of inference engine in rule based approach is, it follow a pattern matching during the reasoning process. But, it is difficult to infer from only few satisfied conditions in this approach. Therefore, to draw the conclusion based on the satisfied conditions it is better to apply case based approach which uses similarity measurement to solve the new problem.

On the other hand, if the user response does not match with any conditions in the knowledge base, then the system will be confused to infer the solution. To handle this issue there are two possible conditions. The first condition is for unknown problems, the system provide possible recommendation which helps the user to solve the problem. Second condition is for specific problems inference engine incorporate partial learning component. This component works when the solution of the problem is out of the knowledge content for inference engine to draw a conclusion. The learning module requests the user to add new problems, and then add the new reasons permanently into the working memory. After validating the problem and its cause the system draw a conclusion and when the user starts a new session the inference engine consider the new problem to execute the rule.

Finally, the system displays the following message after executing one rule.

```
Continue:-
write ('Do you want to continue? (Yes. or no.)'), nl,
read (ANSW),
determine (ANSW),
change (ANSW).
```

If the system user responds ‘yes’ the inference engine starts to execute a new rule or repeat the previous one. Else, if the system user responds ‘no’ the system display the following message and stop running the program.

```
Continue:-
Write ('Thank you very much for using this system!').
```

Cut (!) Command is used to control the Prolog program when the backward searching takes an action. It means the system does not have to do the backward search again even though the operation fails.

'Assert' function is used so that the questions will not be asked again even if the operation fails.

When the instructions fail, the 'retract' is used to perform a backward search in the previous information to see whether those queries could be satisfied or not.

Therefore, the inference engine starts searching from the top level goal and proceeds to the sub_goal by using backward searching mechanism. To fire the rule, the inference engine searches facts from fact base and try to match those fact with a given goal in the knowledge base. If a match is found, the inference engine executes the rule. But, if condition is not satisfied then the inference engine starts to search the next rule until the conditions were satisfied. Similar procedures have been used for each rules incorporated in the knowledge base.

## 4.3. THE USER INTERFACE

The acceptability of a KBS depends on the quality of the user interface. The user interface is used as the means of interaction between a user and the KBS. For this KBS, users interact with the system through "yes" and "no" response only. Based on the user's response the system draws a conclusion for each rule in the knowledge base. The systems conclusion displayed in the user interface window. Figure 4.2 below shows the welcome window and it describes the functions of KBS.

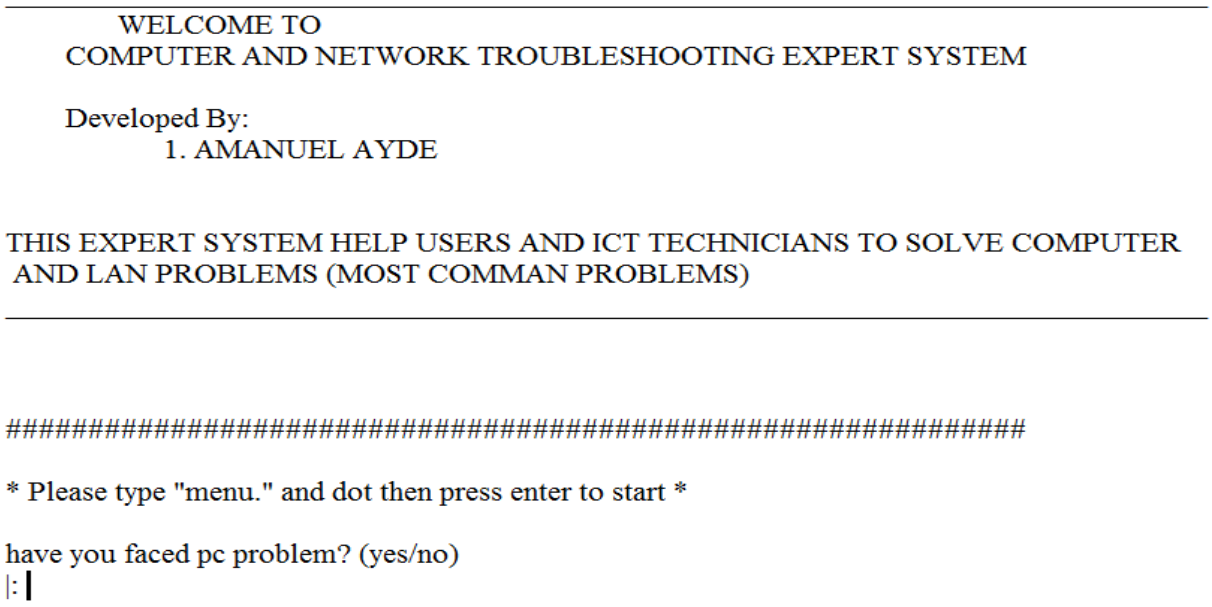



**Figure 4.2 Welcoming Window of CNTES User interface**

Once the welcome window of CNTES user interface displayed, the user can interact with the system by typing “MENU” followed by full stop next to the welcoming screen of the SWI WIN prolog Window. After the user types “menu” followed by full stop, to start the actual task system requests the user to enter yes or no. The expert system requests the user faced PC problem or not. If the response for this request is “no” the system does not allow the user to proceed because the user is faced not PC problem. The system says “sorry the system cannot help you.” If the response is ‘yes’ the expert system allows the user to the next request because the user is faced PC problem. The user response depends on the given guidelines and proceeds for all questions.

“Askable” function is used to keep track of the questions that can be asked and askable is retracted when the question is answered. Ask is responsible for getting information from the user and remembering the user’s response. If the program does not already know the answer to a question it will ask the user. Then it will assert the answer. "Ask" only deals with simple ‘yes’ or ‘no’ answers.

The user interaction with the system is carried out by responding some department oriented questions about specific cases. Every time ask is called, first it checks to see if the answer is already **known** to be “Yes or No” If it is not already **known**, then **ask** will **assert** the user response.

Furthermore, inference engine uses menuask to expand the dimension of user interaction with the system. "Menuask" has the same function as “ask”. The only difference is that it gives the users’ list rather than a single question about the causes and problem(s). As discussed in the above section of this chapter, during the computer and network troubleshooting process the basic function of the system is identifying problems, causes and suggesting solutions for the identified problems. If the system is certain that the computer has problem of X, then the inference engine draws the conclusion. The following run information in the figure4.3 shows the rule1 where all the conditions are satisfied.

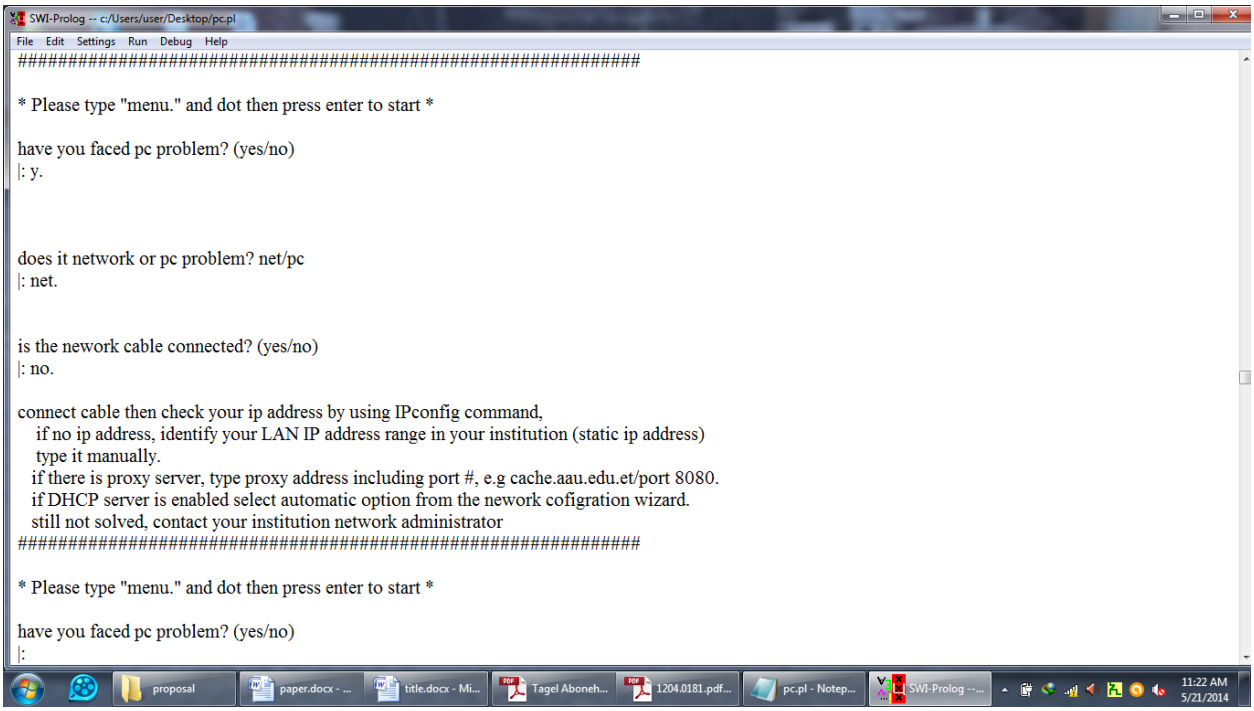



**Figure4.3 Dialog Window between the user and system on Network troubleshooting (cable).**

From figure 4.3 above, the response from the user is no. This indicates that, if the network cable is not connected to the computer there is no network at all. So, the system suggests the user to connect the network cable to the computer before proceeding to any action because the problem is network cable disconnection. The network cable is connected but there is no network the user is requested to feed additional information about the problem. The proceeding request and suggestion is shown in the following figure 4.4.

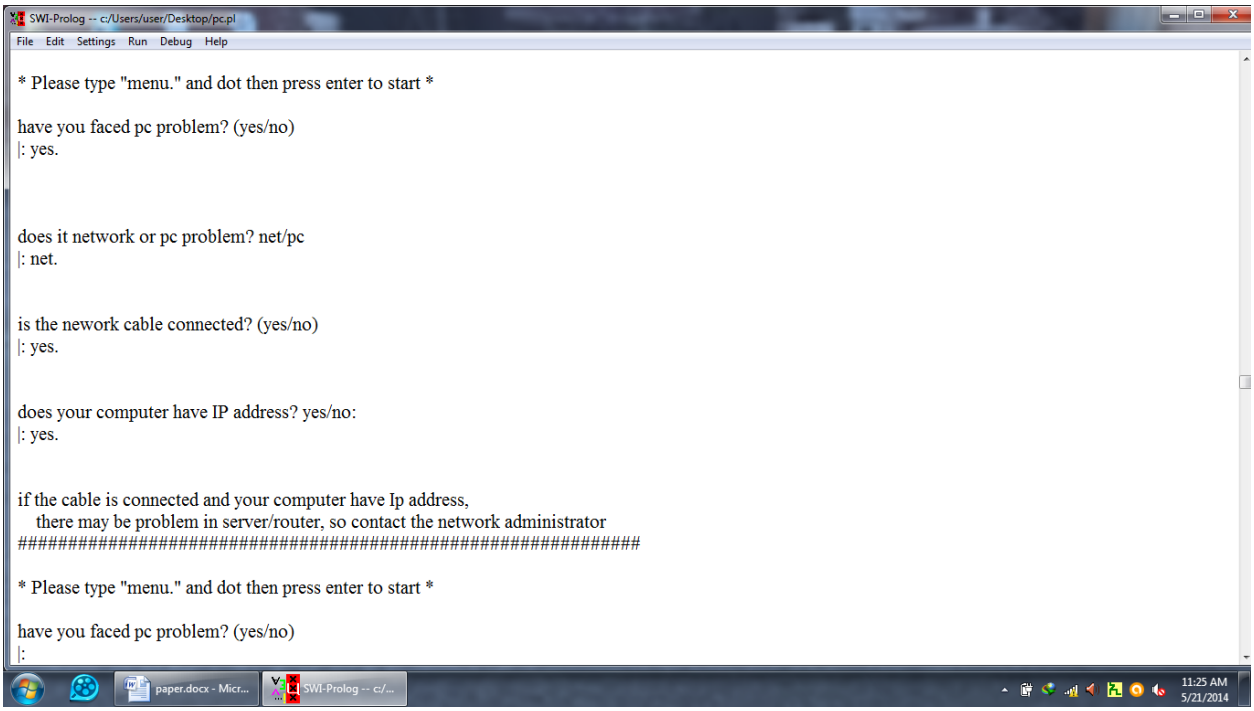



**Figure4.4 dialog Window between the user and system on Network troubleshooting (IP address).**

In figure 4.4 above, the inference engine requests the user the network cable is connected or disconnected. The response of the user is 'yes', so, the inference engine proceed to the next case which is IP address problem. In actual case, the network cable is connected to the computer and the computer IP address is configured there should be network. For this problem the system suggests the user to communicate to the network administrator because the problem is not the configuration problem. If computer has no IP address, the inference engine looks for further conditions to reach the conclusion. The following Figure 4.5 below shows when user's responses are partially satisfied with their respective recommendation of the systems. When we say partially satisfied condition, it mean that the combination of "yes" and "no" user response. But, the internal structures of the algorithms consider all the "yes" response. In other word, when the user respond "no" the inference engine will not stop rather it check the next rules until all the possible conditions are satisfied.

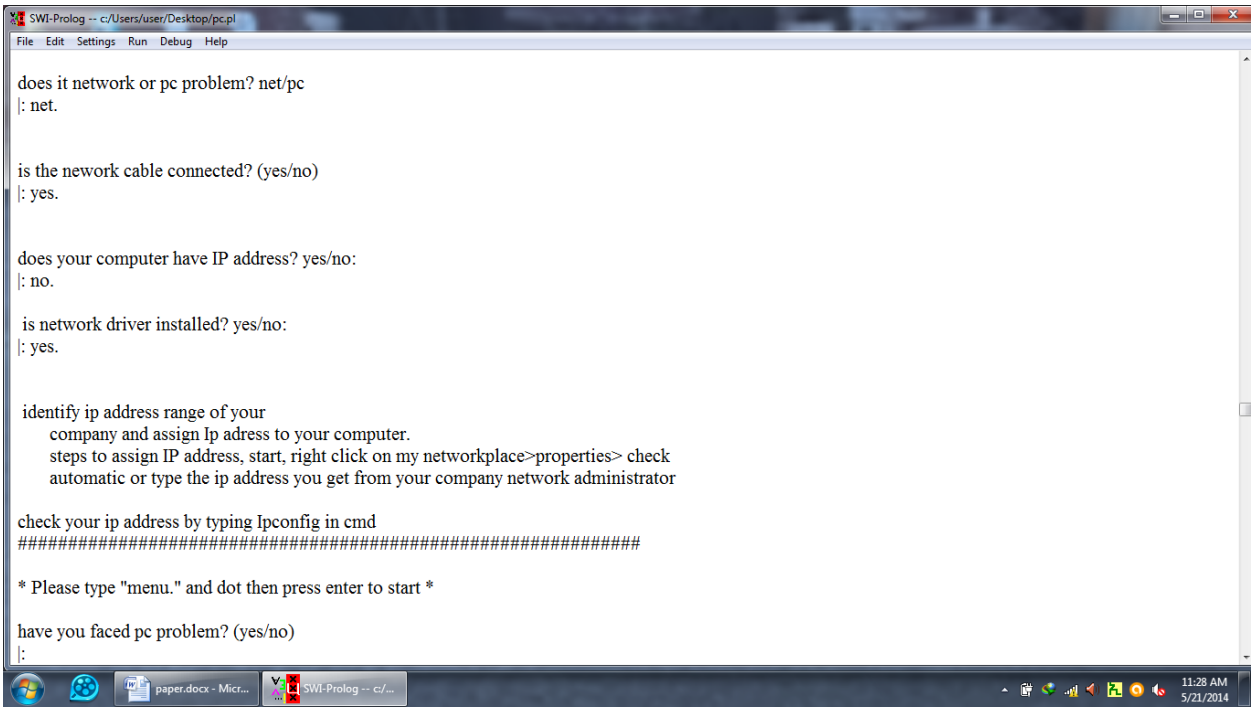



**Figure4.5 dialog Window between the user and system on Network troubleshooting (Network deriver).**

The above figure 4.5 indicates the network cable is connected but the IP address is not configured; in this case the inference engine proceeds to the next condition to reach to the conclusion. The next case is requesting the user, the network driver is installed or not. For this case, if the user response is yes the inference engine reach to the above conclusion and suggest user to configure IP address. Unless, the inference engine proceed to the next condition depending on the user response which is "no network deriver is installed". If the network driver is not installed the following dialog window is appeared.

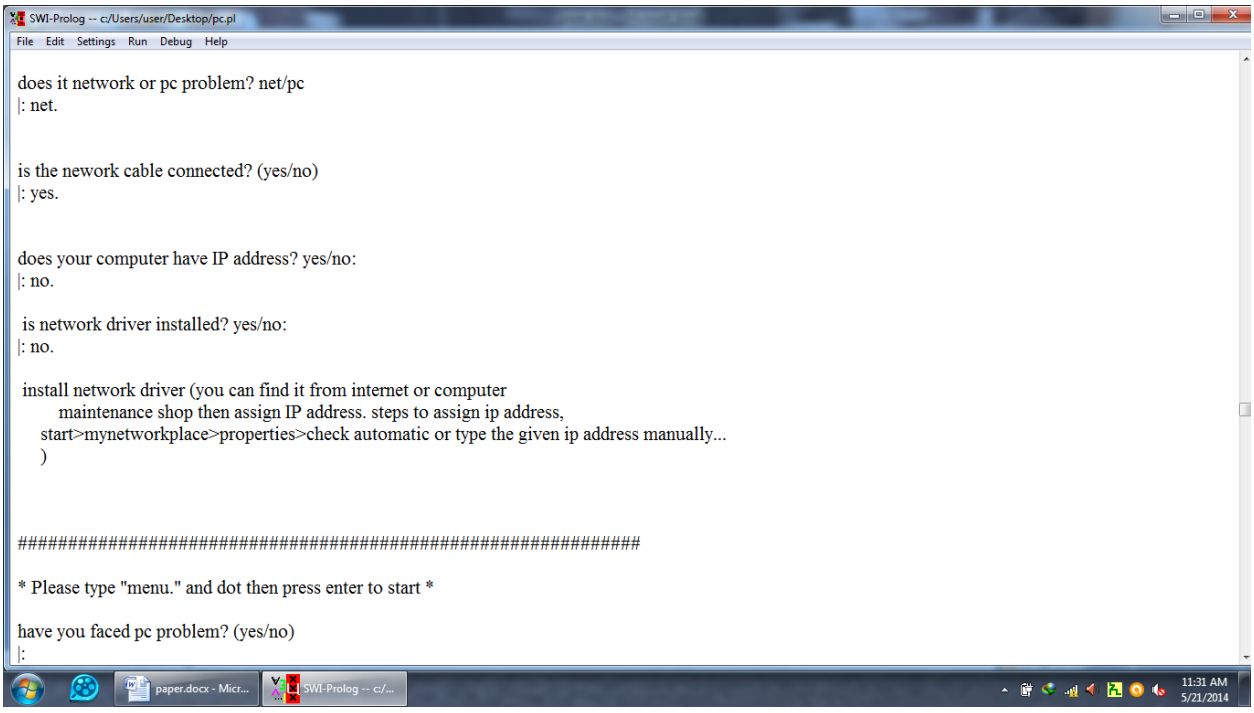



**Figure4.6 dialog Window between the user and system on Network troubleshooting (Network deriver).**

From the above figure 4.6, the responses ‘yes’, ‘no’ are checked and satisfied the conditions so, the inference engine conclude by providing suggestion what to do to solve the problems. The suggestion is that “how to install the network driver and how to configure the IP address.”

## 4.4. LEARNING COMPONENTS OF THE SYSTEM

### 4.4.1 Types of Learning in Rule Based Systems

The principal kinds of learning appropriate for rule based systems are [64]:

1. The invention of new **conditions and/or actions** for rules.
2. The invention of new **conflict resolution strategies** (i.e. meta-rules).
3. The discovery and **correction of errors** in the existing system.

According to Joshua, Thomas and Charles [63], for learning new rules (including meta-rules) there are two basic approaches: inductive and deductive learning.

1. **Inductive rule learning** methods create new rules about a domain that are not derivable from any previous rules. We take some ‘training data’, e.g. examples of an expert performing the given task, and work out corresponding rules that also **generalize** to new situations.

2. **Deductive rule learning** enhances the efficiency of a system's performance by deducing new rules from previously known domain rules and facts. Having the new rules should not change the outputs of the system, but should make it perform more efficiently.

As discussed in chapter three, the knowledge based system incorporated rules for all problems. But, for any problems that do not have rules in the knowledge base, the system provides recommendation which help the user/ICT technician to feed causes and solutions to the problems.

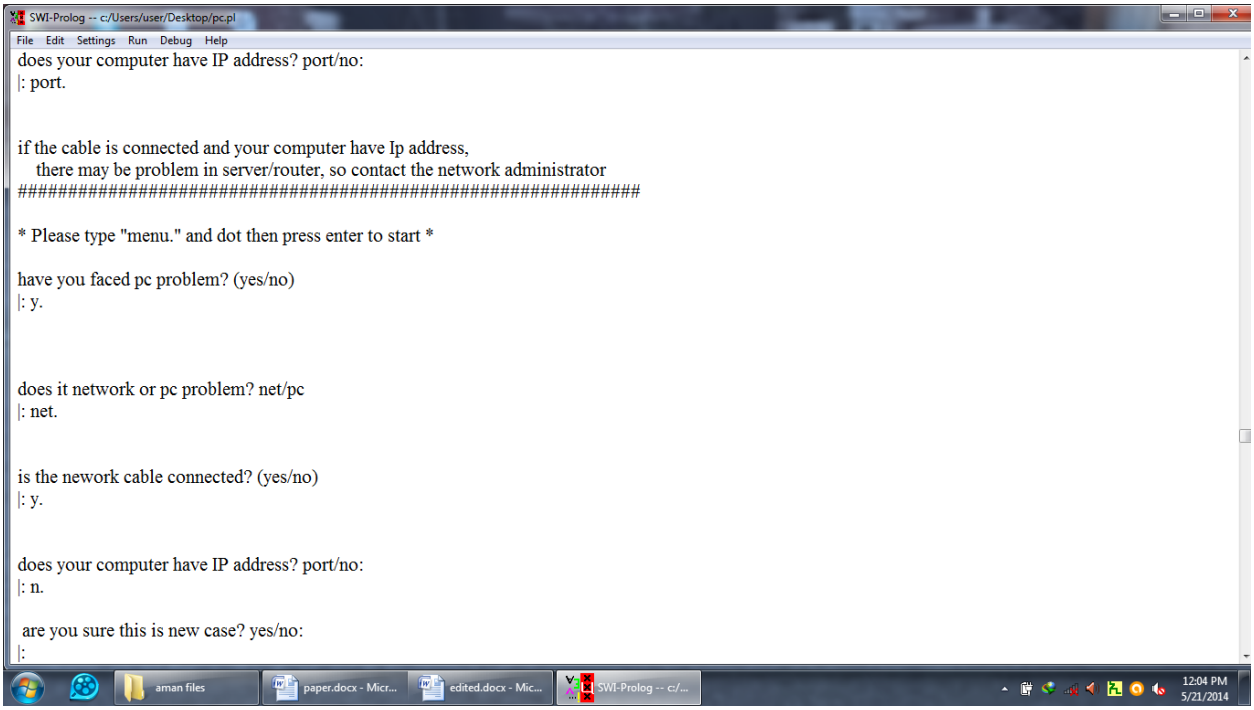



**Figure 4.7. The Learning Component**

To show the possibility of integrating learning component in the knowledge based system, the researcher integrated simple feature to dynamically update the fact base in the working memory for network problems. The inferences dynamically update the new causes in the fact base. The learning function works when the inference engine is unable to find any matching rule from the knowledge base, then it requests the user to add any new problems that the computer faced. Then confirms user response whether it is true or false by saying "are you sure this is new cause?" If user response is "yes" then the system add the new rule permanently into the working memory.

The suggestion of problem based on the identified causes follow the same procedures for all rules in the knowledge base. To execute the rule, the inference engine follows the decision tree structures as it is shown in the conceptual modeling of chapter three.

Therefore, the learning component of the CNTES follows Inductive rule learning because it creates new rules about a domain that are not derivable from any previous rules existing in the knowledge base. When a user starts a new session the inference engine consider the new rule to draw a conclusion from the knowledge base.

## 4.5. THE EXPLANATION MODULE

One of the interesting features of knowledge based systems is the ability to explain itself. The knowledge based system developer used the explanation module to have more realistic dialogs with the system user. The 'how' feature of the prototype is incorporated so as to give explanations how the system reach to certain conclusion.

As indicated in the following figure 4.8, once the inference engine identified the solution of network problems and its causes, then the inference engine requests the user whether he/she needs further explanation 'how' the system arrived at certain conclusion. Explanation module provides further information with a simple "yes or no" user response.

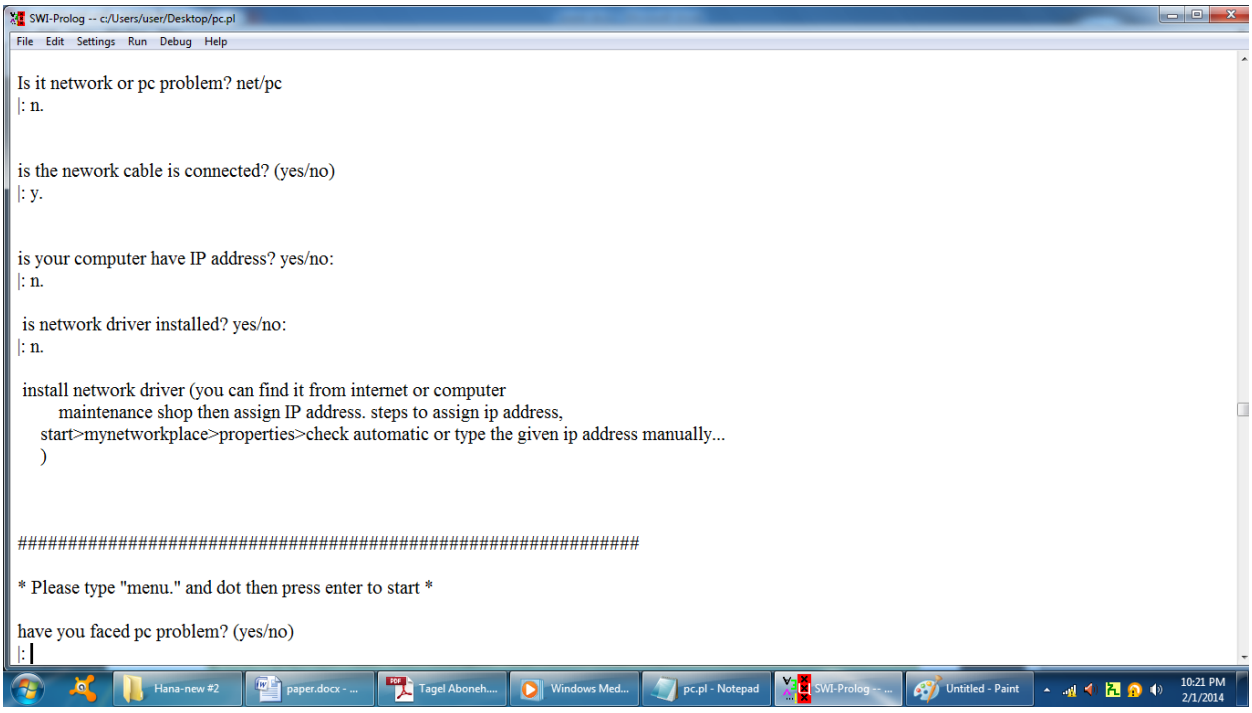



**Figure 4.8 sample of explanation facility dialog windows for network problem**

To summarize the chapter, the acquired knowledge from the domain expert was represented using a set of rules but all the problems are not included in the knowledge base because the researcher is required to design the prototype. The rules were constructed in the prototype with 'if' 'then' format.

“If - then” format of the prototype mainly depending on the decision tree model, which was discussed in chapter three. To achieve the goals of system, the inference engine using backward chaining reasoning mechanism. Furthermore, the knowledge based systems have different components as indicated in the knowledge based system architecture. The knowledge based system components are knowledge base, inference engine, learning component, user interface, and explanation module.

# CHAPTER FIVE

# TESTING AND EVALUATION

After implementing knowledge based system the final step is measuring the performance of the system to know the extent to which it has achieved the objective or not. For the purpose of this study, CNTES is tested and evaluated based on the objective of the system. This is to measure the accuracy of the system during the problem solving processes. Measurement is from the point of whether the system achieved its objective or not. In this study the performance of the system was measured against human expert decision using predictive validation method. The user acceptance of the system was carried out during system user interaction.

The KBS user acceptance was measured by using open and close ended questions. The system evaluators directly interact with the system to measure its performance from the points of its correctness in providing solutions for different problems. In addition, the validation test was done by comparing solved cases against the system conclusions on the similar issues. By comparing the result obtained from the system conclusion, the evaluators determine the performance of the system. Therefore, knowledge based system performance is measured first by using open and close ended question, and second by using the test case validation method. The details of each evaluation method were discussed in the following section.

## 5.1. USER ACCEPTANCE EVALUATION

Knowledge based system Visual interaction evaluation method allows the domain expert to directly interact with the system. Direct interaction intend system user to evaluate the performance of the knowledge based system from the users' point of view. In addition, this method helps to ensure the performance of the prototype by assessing the feedback acquired from the domain expert towards the developed system. Semi-Structured questionnaires were developed to assess and evaluate the appropriateness and applicability of CNTES in the domain area. Different local researchers adapted evaluation questionnaires that used to evaluate the model called ResQue (Recommender systems quality of user experience) from user's point of view [31]. Therefore, for the purpose of this research the evaluation questionnaires were converted and adapted into the context of computer and network troubleshooting system. *Refer appendixes* II

To achieve the goals of evaluation ten domain experts were selected purposively including the individuals participated in the interview question from Jimma University ICT office. The domain experts were selected purposively from ICT office from Computer and network technicians section because these individuals were technical workers in the ICT office of Jimma University. During the KBS development these domain experts were actively involved in the different stages of pilot study, knowledge acquisition, prototype development and consulting on the content of knowledge. The informal discussion with domain experts had significant role to understand the dimension of the problem. Before the actual evaluation process conducted, some guideline and brief explanation was given to the system evaluators on how the system works.

The system performance evaluation was carried out by using questionnaires. The questionnaire includes both close and open ended questions. Using these questions system evaluators directly interact with knowledge based system. Totally fifteen evaluation questions were prepared for system evaluators. The questions are divided into two parts, the first seven questions are close ended questions which helps system evaluators to check on the user interface design aspects, easiness of the system to use, attractiveness, correctness of the decision, adequacy of knowledge content, the problem solving ability and significance of knowledge based system in computer and network troubleshooting services. On the other hand, the remaining eight questions are open ended questions which used to collect constructive feedback from the system evaluator's based on the system's conclusion.

The evaluators were allowed to rate the options as excellent, very good, good, fair, and poor for these closed ended questions. Therefore, for easiness of analyzing the relative performance of the prototype based on the user evaluation after the interaction with the system, the researcher assigned numeric value for each of the options given in words. The values are given as **Excellent** = 5, **Very good** = 4, **Good** = 3, **Fair** = 2, and **Poor** = 1. Based on the given scale, system evaluators provide a value for each close ended question. Thus, this method helps the researcher to manually examine the user acceptance based on evaluator's response. The user acceptance of the system is measured manually as follows [10]:

$$\mathbf{AVP} = \sum_{k=1}^{n} \mathbf{SVi}.\frac{\mathbf{NRi}}{\mathbf{TNR}}$$

Where, SV scale value, TNR total number of respondent and NR is number of respondent. To get the result of user acceptance average performance is calculated out 100%.

$$\left(\mathbf{AVP} = \sum_{k=1}^{n} \mathbf{SVi}.\frac{\mathbf{NRi}}{\mathbf{TNR}}\right) \mathbf{X\ 100/NS}$$

Where NS is number of scale and Avp is average performance**.** The following table summarizes the results obtained from the respondents'.

| **No** | **Questions** | **1** | **2** | **3** | **4** | **5** | **Avg** | **Total Av/Perf. %** |
|---|---|---|---|---|---|---|---|---|
| 1 | Is the CNTES system is efficient in time? | 0 | 0 | 1 | 5 | 4 | 4.3 | 86 |
| 2 | What is the accuracy of the system in troubleshooting computer and network problems? | 0 | 0 | 2 | 4 | 4 | 4.2 | 84 |
| 3 | Does the system incorporate sufficient knowledge to solve problems? | 0 | 0 | 0 | 6 | 4 | 4.4 | 88 |
| 4 | Does the system provide appropriate suggestion for the problems? | 0 | 0 | 1 | 5 | 4 | 4.3 | 86 |
| 5 | What is the contribution of the system in the domain area? | 0 | 0 | 3 | 7 | 0 | 3.7 | 74 |
| | | Total average | | | | | 4.18 | 83.6 |

**Table 5.1. Performance evaluation of CNTES**

As indicated in table 5.1 above, for question 'efficiency in time' 40% of the respondents rated the criterion as excellent, 50% respondent evaluated as very good and the remaining 10% of them respond as good. At the same time for criteria 'the accuracy of the prototype to make correct decision' 40% respondent rated as excellent, 40% of the respondent respond as very good and the remaining 20% evaluated as good. Likewise, for the criteria of; 'does the prototype incorporate adequate knowledge' 40% as excellent and the rest 60% as very good. Again 50% of the respondent rate as excellent for the criteria of 'Does the system provide appropriate suggestion for the problems'

50% responded as excellent and the remaining 10% responded good. Finally, for the question related to 'significance of the knowledge base system in the domain area' 70% of them evaluated as excellent and the rest 30% of them responded as very good.

When summarizing table 5.1 above, based on the responses of ten system evaluator, the average performance obtained is 4.18 on a scale of 5. This value is the result obtained from the values assigned for each close ended question. The result indicates that about 83.6% of users are satisfied by the performance of the knowledge based system. It means that the knowledge based system gained about 83.6% of user acceptance.

When compared with the results from the previous works in the summary of related works in table 2.1, "DART: An Expert System for Computer Fault Diagnosis" the evaluation result is 50% and "Pimtool: an expert system to Troubleshoot Computer Hardware failures." Also 50%. So this CNTES is better than those works depending on the evaluation results from the users' response.

The system evaluators were also provided open ended questions to collect expert's feedback, suggestion and opinions. These questions focus on how the CNTES is different from human expert in diagnosing computer and network problems. Furthermore the open ended questions help evaluators to provide their feedbacks on the contributions of the system, the uncovered knowledge issues, knowledge content of the system, the limitations and strength of the knowledge based system.

The first open ended question concern with the differences of decision between human experts and knowledge based system in diagnosing the computer problems. The system evaluators responded that CNTES solve problems based on the stored knowledge in the knowledge base with in time and cost wise. But, the human expert uses their expertise, vital signs, experience, manuals and guidelines which is time consuming and difficult to remember. In addition human expert further require physical and laboratory inspection of problems to make better decision. Domain experts use physical appearance as main tool to understand the status of computer and network problems during the troubleshooting processes. They observe the hardware appearance, LED light, beep sound, screen display and etc. However, the prototype depends on the contents of knowledge it incorporated in the knowledge base. Therefore, it lacks robustness such as sensing and responding ability.

Regarding the question of knowledge coverage issues by CNTES, the respondents replied that the knowledge based system is efficient to respond from the stored knowledge in the knowledge base. The knowledge base contains knowledge of about ten problems to represent the computer and network problems. But, knowledge based system lack the incorporation of physical and laboratory

inspection results. In similar way, some respondents suggested that at problem level identifying the causes is enough to solve problems occurring in the computer system. Therefore, the knowledge base contains relevant knowledge for the selected problems to make the decision. But, some respondent argue that knowledge base system will not replace the function of human expert.

Similarly, the respondents were asked to express their opinion for the question of "Does knowledge based systems replace the tasks of ICT expert in troubleshooting the problems?" Most of the respondents confirmed positive feedback. As their response indicated, as far as the knowledge base contain all the relevant knowledge acquired from the domain expert, the knowledge based system provide similar decision as human expert. Even the system can be considered more flexible due to its reusability of the existing knowledge in a better way than the human experts.

The truth is that, human expert consider different conditions during the troubleshooting processes such as the physical condition of the device, sound emitting, and light factors. We can design knowledge based system to support or assist the domain expert but, the main challenge is incorporating knowledge of physical inspection and sensing ability into the system.

At the same time, most of the respondents reply that the knowledge based system contains adequate knowledge for all rules incorporated in the knowledge base. As their suggestions indicated, the system must update the existing knowledge and learn from the environment. To handle such issues the researcher incorporated single case to dynamically update the existing fact in working memory. This component helps knowledge engineers to design a knowledge based system that update the fact base dynamically. But, it is clear that incorporating learning component for rule based system is difficult and complex as the numbers of rule expanded.

Therefore, for specific domain area incorporating such component will improve the efficiency of the system. For the question that concern about the significance of the system in the domain area, all the system evaluator's responded that the system add value in the domain area. Therefore, the application of new technology simplifies the working environment in the domain area. In addition the system also can reduce the burden of human expert by saving their time and energy spent while referring books and other manuals to solve the problems. Not only the system recommends solutions, but it can also contribute for the reusability and sharing of the same knowledge for different purposes in the institution. Finally, the system can be used as teaching and learning instrument for academic purpose.

Lastly, the system evaluators also provided their suggestions and comments on the weakness and strength of the knowledge based system. According to the system evaluators the following are some of the basic limitations of the knowledge based system:

- Cannot make creative and intelligent responses as human expert
- Lacks robustness such as sense and response ability
- The system lack the incorporation of some basic task in troubleshooting processes such as physical inspection and screen display which is significant to make conclusion.
- The system interacts with the user using only 'yes' or 'no' replies. Therefore, it lacks some flexibility.

The system evaluators also forwarded their constructive feedback on the strength of the knowledge based system. The strong sides of the knowledge based system are:

- Helps to diagnose many problems in short time frame and solve shortage of manpower in the ICT sector.
- Consistent answers for repetitive decisions, processes and tasks
- More accurate and Easy to use
- Applicable anywhere, even at home we can advise ourselves by having the software
- Make the diagnosing processes more clear, by showing the steps and procedures carried out by domain expert.
- Encourages organizations to clarify the logic of their decision -making
- Never "forget" to ask a question even if the operation fail

User acceptance evaluation (close and open) ended questions are attached in the appendix II.

## 5.2. CNTES VALIDATION BY USING TEST CASES

Section 5.1 discussed about the evaluation of system performance using both closed and open ended questions. System evaluators directly interacted with system using these questions in order to forward their feedback and suggestion on the performance of the system.

In this section the performance of the system was tested and validated using test cases. The test cases were used to measure the accuracy of the system. For the purpose of validation process a total of twelve cases were selected. To achieve the goal of the system evaluators were purposively selected according to their willingness and specialty. Therefore, the system evaluators were from different computer and network sections in Jimma University ICT office.

As discussed in section 5.1 above system evaluators were selected using purposive sampling techniques from the domain area. From selected ten domain expert who have measured user acceptance of knowledge based system, eight of them were selected purposively to evaluate the test case. Then, to perform the test procedure, the selected system evaluators are grouped into three classes of Network and computer to represent computer and network troubleshooting expert system. For each class two domain experts are assigned to evaluate four cases.

The KBS testing procedure was carried out by system evaluator to evaluate the solutions suggested by the system were correct or incorrect. System evaluators compared the decisions made by the system against human expert. Then system evaluators validated the number of correct decisions made by the system. The result of the comparison shows that the rule based system has made close decision in the troubleshooting process of problems as human experts did. As indicated in table 5.2 below, the result provided by system evaluators showed that the knowledge based system is about 75% accurate in troubleshooting computer and network problems.

| Problems | Total number of problems selected | Correct suggestion | Incorrect suggestion | The accuracy of prototype in % |
|---|---|---|---|---|
| Network cable connection problem | 3 | 3 | 0 | 100 |
| IP address configuration problem | 3 | 1 | 2 | 33.4 |
| Computer power problem | 3 | 3 | 0 | 100 |
| Sound problem | 3 | 2 | 1 | 66.67 |
| Total | 12 | 8 | 4 | 75% |

**Table 5.2 Testing the Accuracy of CNTES by using test cases**

From table 5.2 above twelve cases are selected purposively to validate the accuracy of the system and three cases are evaluated for each problem category. For any problems stored in the knowledge base, the system can suggest solutions for a numbers of times. Purposively selected problems are

used to challenge the system performance. As a result, for network cable connection problem in the above table 5.2 from the given three cases three of them are correctly suggested by the system (100%). Similarly, from the given three cases only one of them is classified correctly in the IP address configuration problem which is (33.4%). This result shows that there are overlapping solutions for this problem. This means the more similarity in solutions the better challenge for the system in identifying the correct solution for the problem. For the power problem of the computer the system correctly provided suggestion (solutions) for all the problems (100%) . Finally, from the three causes of the sound problems the system suggested 2 correct solutions in similar way as human expert troubleshoot the problems which is (66.67%). The result indicated that all the cases are directly similar with knowledge incorporated in the knowledge base and average performance of the CNTES is 75%. Finally, sample of problems and cases are attached at the appendix III.

When comparing this result with the other expert system "An Expert System Algorithm for Computer System Diagnostics." Whose result is 93.3%. The CNTES result is lower than that of "An Expert System Algorithm for Computer System Diagnostics." Due to the number of causes used to evaluate the system are only twelve and the troubleshooting issues are complex. So, based on the given causes the result is enough; because 75% of the causes are correctly responded by the system.

### 5.2.1. Decision Variation between System and Human Expert

As discussed in section 5.2 above the decision made by the system have shown slight difference with decision of human expert during test case validation. There are different contributing factors for the variation of decision made by the knowledge based system. First, the KBS is limited to the knowledge incorporated in the knowledge base. The main problem is contextual understanding of computer system problems. This mean sometimes the problem may be a network driver problem but the expert system suggests other solutions such as, IP address configuration, network interface card installation and to go to network administrator. For example, for the problem the computer is not starting; there are different cases. Some of them are the power source is off, the computer power on button off, internal power cable is disconnected, damaged power supply and power cable shortage. Due to inability to understand the context of the different cases the system suggests different solutions this may complicate the way of solving problems. Therefore, if the problem and cases are not clearly specified, it will expose the computer to serious damage. As a result, the case and solution can be considered differently by different domain experts. Such issues affect the accuracy of the system during the test case validation.

On the other hand, to show the possibility of updating the existing knowledge, the researcher incorporated single case to update the facts from working memory based on the user response.

Once the new problem is acquired from the user the system stores it permanently in the fact base.

When user starts a new session, the system remembers previously stored causes to make a decision. In addition, the knowledge based system provides recommendation which helps the ICT technician to solve the problems by considering causes. Therefore, cause based reasoning technique need to incorporate with rule based reasoning, which enables the system to learn from past practices. Then it will facilitate the full-fledge learning capability of the system by integrating the hybrid reasoning mechanism.

## 5.3. DISCUSSION

The prototype knowledge based system is promising and applicable in the domain area. The feedback and suggestion of domain expert reveals that the knowledge based system satisfactorily gained user acceptance. The system acceptance evaluations used open and close ended questions to directly interact with system.

The table below shows the summary of system performance evaluation result obtained from close ended questions.

| Domain experts | 1(poor) | 2(fair | 3(good) | 4(v. good) | 5(Excellent) | Total Avg/ Prf.% |
|---|---|---|---|---|---|---|
| Total number | 0 | 0 | 7 | 27 | 16 | 418 |
| Average out of 100% | 0 | 0 | 14% | 54% | 32% | 83.6 |

**Table 5.3 summary of system evaluators Result on Close Ended Questions**

As the table 5.3 above indicated, based on system performance gained from user's visual interaction with the prototype expert system using the closed ended questions; There is no evaluators respond as poor and fair, evaluators responded good fourteen times (14.2 %), very good fifty four times (54%) and excellent thirty two times (32%). The total average user acceptance evaluation result of knowledge based system is about 83.6%. This result shows that more than 83% users accepted the correct decision, knowledge content, learning ability, contribution and significance of the knowledge based system. Therefore, the prototype knowledge based system satisfactorily makes the right

decisions and providing appropriate recommendation during computer and network troubleshooting process.

When comparing the performance of CNTES result 79.25% (which is the average of user acceptance and predictive validation result) with other expert systems result (An Expert System for Computer Fault Diagnosis" the evaluation result is 50%, "Pimtool: an expert system to Troubleshoot Computer Hardware failures" the result is 50% and "An Expert System Algorithm for Computer System Diagnostics." the result is 93.3%). The average of these three expert systems result is 64.43% which less than the CNTES average result. So, the performance of CNTES is better than the previous expert systems.

Similarly, the response of system evaluators confirm that developing such prototype knowledge based system can reduce the existing knowledge gap in areas where ICT technicians were scarce. As discussed in section 5.2, the system validation was carried out using solved problems by human expert. As the result indicated, the prototype has made close decision with domain experts in the processes of troubleshooting the problems. The causes for the difference between human expert and the knowledge based system in decision making processes identified. The system also contributes valuable roles in the process of sharing experiences and skill in the computer and network troubleshooting sector.

Finally, the performance of the prototype knowledge based system has got good user acceptance by the system evaluators. As a result, the knowledge based system would assist human expert to provide a better computer troubleshooting service. Therefore, this system helps the general users to solve problems occurring in the computer system in their own instead of going to ICT technicians. In addition, the system provides first level problem solving technique in order to increase the quality of computer troubleshooting services. However, the system needs to be intelligent and learn automatically during users' request. The knowledge based system should also further strengthen to include the knowledge acquired from physical inspection of the computer condition for better decision.

# CHAPTER SIX

# CONCLUSION AND RECOMMENDATIONS

## 6.1 CONCLUSION

In Developing countries like Ethiopia, the availability of ICT technician is scarce. Different factors are identified such as the skill level of the technician, shortage of skilled manpower in the area, the distribution of ICT technicians per institutions, complexity of computer and network related problems, and so on.

To address the above problems KBS was designed, the system assists the ICT technician and computer users to solve problems effectively and efficiently. The relevant knowledge was acquired from domain through interview questions, observation and document analysis. Different challenges were identified during acquisition phase such as willingness and confidence of domain expert to share their knowledge. The KBS is conceptually modeled using decision tree structure to the logical relationships between causes and solutions of the problems for representation rule based reasoning technique was used. The prototype of CNTES is developed by using SWI WIN prolog programming language.

Finally, the following conclusions were drawn from the finding with regard to the research questions:

- Applicability of KBS for computer and network troubleshooting haven been proved. But, to fully provide troubleshooting service other problems such as physical appearance of the device, LED light, beep sound, etc need to be incorporated into the current system.
- The system provides advices on the bases of problems feed by the ICT technician or computer user.
- Knowledge was represented in the form of if then rules generated from the decision tree. But, some challenges were identified during inferring from the given conditions, since, rules need pattern matching. Therefore, it is possible to conclude that other approaches such as case based system which infer from previous solved similar cases to draw a better solution.
- The prototype KBS used user acceptance and test case validation techniques to measure the performance of the system. The result of system performance indicated that users were satisfied with prototype and the test case validation results showed the system correctly provide recommendation for the identified causes. Therefore, the system registered encouraging result, and then it is possible to say the system addressed its objective.

- In the prototype KBS an attempt was made for specific case to dynamically update the fact base in working memory. So that it is possible to conclude that, it is possible to make system to learn from the user response and permanently remember the new knowledge if the system works for specific domain.

## 6.2. RECOMMENDATION

The study achieved its main objective by proving the applicability of the rule based system in computer and network troubleshooting. Based on the findings of the study the following recommendations were suggested for further study in the applicability of knowledge based system in the domain area. The recommendation would initiate interested researcher to investigate further implementation of prototype knowledge based system in related computer troubleshooting domain.

- The scope of the knowledge based system should be extended to incorporate other causes occurring in the computer system such as physical appearance of the device, sound emitting, LED light, BIOS messages, and POST.
- Rule based system can handle any domain specific problem, if there is perfect knowledge. But, most of the time the causes from the computer and the user may not satisfy the conditions of the given rule. Therefore, if we want to conclude better solution based on the few respondents response, then it is better to apply case based techniques.
- Computer troubleshooting is often complex and subject to conditions of uncertainty. This is due to continuous change in technology and inconsistency of problems affecting computer system. Therefore, further investigation should be done to integrate an intelligent agent that has a capability to self learning and update its knowledge base.
- In this study an attempt is made to apply rule based systems. But, there are different solved cases available in computer troubleshooting. Rule based systems solve problems from scratch, while case based systems use pre-stored situations to deal with similar new instances. Therefore, the integration of rule based reasoning with case based reasoning would solve the limitation when representing knowledge in the form of if then rules unable to draw a conclusion.
- The challenge in developing knowledge based expert is the user interface which is not simple and easy to interact. Because the tool used to develop CNTES using SWI Win prolog which uses yes and no response to communicate. To handle this challenge the tool with better user interaction is required.

# REFERENCE


[1] A. D. M. Africa. "An Expert System Algorithm for Computer System Diagnostics." *International Journal of Engineering (IJE),* Vol. 5, No. 5, PP. 435 -467, 2011.

[2] Wikipedia. "Expert System." http://en.wikipedia.org/wiki/Expert_system, Nov. 22, 2010 [Dec. 05, 2013]

[3] S. Manda, S. Chatterjee, B. Neogi. "Diagnosis and Troubleshooting of Computer Faults Based On Expert System and Artificial Intelligence." *International Journal of Pure and Applied Mathematics,* Vol. 83 No. 5, pp. 717-729, 2012.

[4] J.K.C Kingston "Knowledge based system development tools." *Artificial Intelligence, University of Edinburgh, Scotland,* 2000.

[5] Y. Bassil. "Expert PC Troubleshooter With Fuzzy –Logic." *International Journal of Artificial Intelligence & Applications (IJAIA*), Vol.3, No.2, PP. 11-21, 2012

[6] G. A. P Kunesegeran. " Knowledge Based System For Troubleshooting PC." Master thesis, University Teknikal, Malaysia, 2007.

[7] I. C. Cameron. "A Computer Troubleshooting Expert System To Aid Technical Support Representatives." *Master Theses*, Algoma University College, Canada, 2005.

[8] J. S. Bennet. "DART:An Expert System for Computer Fault Diagnosis." *Heuristic Programming Projec*, vol. 2. pp. 843-8454, 1980.

[9] M. H. A. Wahab, N. A. Bakar and M. F. M. Mohsin. "Designing a Web-based Network Troubleshooting Expert Systems." *FKEE Compilation of papers*, pp. 69-82, 2009.

[10] T. Aboneh. "Knowledge based system for pre-medical triage treatment at Adama University Asella Hospital." *Masters Thesis*, Addis Ababa University, Ethiopia, 2013. University Teknikal, Malaysia Melaka, 2007.

[11] M. Tongco. "Purposive sampling as a tool for information selection." *Ethnobotany Research and Application*, pp. 147-158, 2007.

[12] T. Palys. “Purposive sampling.” The sage Encyclopedia of Qualitative research Methods, Vol. 2., pp. 697-698, 2008.

[13] M. S. Abdulah, R. Paige, C. Thpmpson, I. Benes and C. Kimbe. “Conceptual modeling of knowledge Based Systems by Using UML” , pp. 24-33, Sawed Arabia, 2005.

[14] A. Salem. “Case Based Reasoning Technology for Medical Diagnosis.” World Academy of Science, Engineering and Technology, 2007.

[15] B. Henok. “A case based reasoning knowledge based system for hypertension management.” *Msc. Thesis,* Addis Ababa University, Ethiopia, 2011.

[16] J. Norving. “Artificial Intelligence Modern Approach.” Newjersy: printice hall, 2003.

[17] F. Pfenning. “Logic Programming.” Carnegie Mellon University, 2007.

[18] D. B. Fogel. “Defining Artificial Intlligence: Evolutionary Computation.” *The Institute of Electrical and Electronics Engineers,* Inc, pp.1, 2006.

[19] D. Shah. “Knowledge based Diagnosis of Abdomen Pain using Fuzzy Prolog Rules.” *Journal of Emerging Trends in Computing and Information Sciences,* 2010.

[20] Heijst. “Conceptual Modelling for Knowledge-Based Systems.” Encyclopedia of Computer Science and Technology,, Marce Dekker Inc., New York. 2006.

[21] P. Akerkar. “Knowledge-Based Systems for Development.” Pp. 1 – 11, TMRF, 2010.

[22] G. R. Yost and A. Newell. “A Problem Space Approach to Expert System Specification.” Defense Advanced Research Projects Agency (DOD), Carnegie Mellon University, USA.

[23] P. Saxena. “Architecture for Medical Diagnosis Using Rule-Based Technique.” *The First International Conference on Interdisciplinary Research and Development*. Thailand: Dayalbagh Educational Institute, 2011.

[24] R. Aref. “A Knowledge based system for comfort analysis of internal environment of hotels.”, USA, 2000.

[25] N. Alechina. “Knowledge Acquisition, Representation, and Reasoning.” United Kingdom, 2012.

[26] B. Gil. “Flexible Knowledge Acquisition Through Explicit Representation of Knowledge Roles.” *Marina del Rey*: USC/Information Sciences Institute, 1999.

[27] S. Tehrani. “A Conceptual Model of Knowledge Elicitation.” College of Business,

Technology and Communication, pp. 2., 2009.

[28] S. Wang. "Knowledge elicitation approach in enhancing tacit knowledge sharing." *Emerald*, vol.111, No. 7, pp. 1039-1064, Hong Kong, 2011.

[29] J. Ranjan. " Knowledge Acquisition, Representation, and Reasoning." pp. 69–79, California State University, USA, 2006.

[30] M. Gau. "Knowledge Acquisition for a Diagnosis-Based Task." *NATO ASI Series*." 1990.

[31] E. Osuagwu. "The Underlying Issues in Knowledge Elicitation." *Interdisciplinary Journal of Information, Knowledge, and Management*, 2006.

[32] S. Chen. "Knowledge Representation and Reasoning Methodology based on CBR Algorithm for Modular Fixture Design." *Journal of the Chinese Society of Mechanical Engineers*, pp.593-604, 2007.

[33] A. Plaza. "Case-Based Reasoning: Foundational Issues,Methodological Variations, and System Approaches." AI Communications, pp. 39-59, 1994.

[34] S. Simon. "Case-Based Reasoning: Concepts, Features and Soft Computing." Kluwer Academic Publishers. Manufactured in The United States, pp. 233–238. 2004.

[35] J. Prentzas."Categorizing Approaches Combining Rule-Based and Case-Based Reasoning." Vol.24, Greece, (2007).

[36] J. Hatzilygeroudis. "Integrations of Rule-Based and Case-Based Reasoning." Research Academic Computer Technology Institute, 2006.

[37] D. Merritt, "Building Expert Systems in Prolog." U.S.A. : Amzi! inc., 2000.

[38] J. Freeman-Hargis. "Methods of Rule-Based Systems." http://ai-depot.com/Contest/Rule-Based Systems and Identification Trees:"Writing", "http://ai-depot.com/Contest" ); include('../Include/Header.php'); include('/usr/home/alexjc/public_www/Include/Utils.php'); Menu(); ?>, sep. 2012[Jan. 2014]

[39] E. Seblewongel. "Prototype knowledge based system for anxiety mental disorder diagnosis." *Msc. Thesis*, Addis Ababa: Addis Ababa University, Ethiopia, 2011.

[40] K. Ghan. “Human judgment in diagnosing problem behavior in horses Using knowledge-based system application.” B.A.(Hons), University of Nottingham, United Kingdom: 2004.

[41] N. Kaiser. “Scaling up rule based software development environments.” *International Journal on Software Engineering & Knowledge Engineering,* pp. 59-78, 1992.

[42] J. Winterton. “Typology of knowledge, skills and competences: clarification of the concept and prototype.” *Centre for European Research on Employment and Human Resources Groupe ESC Toulouse*, 2005.

[43] E. Lamma. “Rule-based Programming for Building Expert Systems: a Comparison in the Microbiological Data Validation and Surveillance Domain.” Electronic Notes in Theoretical Computer Science, pp. 15, 2001.

[44] G. Nalepa. “Methodologies and Technologies for Rule-Based Systems Design and Implementation.” *Towards Hybrid Knowledge Engineering*. AGH University of Science and Technology, Poland, 1998.

[45] U. Endriss. “An Introduction to Prolog Programming.*”* Amsterdam : Universiteit, van Amsterdam, 2007.

[46] M. Covington. “Prolog Programming in Depth.*”* U.S.A: Artificial Intelligence Programs, The University of Georgia, 1995.

[47] R. Shalfield. “LPA WIN prolog user guide. England: *Logic Programming Associates.*” Chanaphon Changkaew, 2002.

[48] A. Gomez. “From Knowledge Based Systems to Knowledge Sharing Technology: Evaluation and Assessment.” Stanford University, USA, 1994.

[49] M. Lech.“Validation of rule based system generated by classification algorithms.” ML Owoc, Poland, 2000.

[50] M. Mak. “Applications of Knowledge-Based Expert Systems to Feng Shui Knowledge.” *Proceedings of International MultiConference of Engineers and Computer Scientists*, Vol. 1, pp. 35-46, Hong Kong, 2010.

[51] B.Thomas. “Evaluation of clinical information systems. What can be evaluated and what cannot?” *Journal of Evaluation in Clinical Practice,* Vol. 4, pp. 373–385, US, 2001.

[52] C. Ohmann. “A systematic approach to the assessment of user satisfaction with health

care systems." constructs, models and instruments Studies in Health Technology and Informatics. US National Library of Medicine, National Institutes of Health Search database, pp.781–785, US, 1997.

[53] A. Stranieri. "The evaluation of legal knowledge based systems." School of Information Technology and Mathematical Sciences, University of Ballarat, Australia, 2000.

[54] A. S. A. Al Ghamdi, S. M. Hussaini and A. K. Jilani. "Empirical Model using Expert System Techniques in Hardware Failure of a System during Backup of Data." IJCSNS International Journal of Computer Science and Network Security, Vol..13 No.2, 2013.

[55] D. Narendra and A. Bart. "Pimtool: an expert system to Troubleshoot Computer Hardware failures." *IAAI-97 Proceedings*, Mountain View, California, 1997.

[56] A. Kaushik and M. Barnela. "Printer Troubleshooting Expert System." *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 2, Issue 10, India, 2012.

[57] M. Anibal, C. Antonio and C. J. Pablo. "A WWW-Based Expert System Advisor for the Diagnostic of Network Communication Problems." University of Puerto Rico, Mayag.

[58] B. R. Japheth and G. M. Bubou. "Expert System Design for Pc Troubleshooting." *Journal of Physical Sciences and Innovation*, Delta University, Vol. 4, Niger, 2012.

[69] H. Schreiber. "Conceptual Modelling for Knowledge-Based Systems:" *Tp appear in: Encyclopedia of Computer Science and Technology*, Marce Dekker Inc., NewYork, 1999.

[60] D. Lidtke. "Fast based NP chunking with decision trees:" experiment on different POS tag.in:international conference on computational linguistic and intelligent text processing. *kyto association for computing machinery Inc.*, pp. 31-58, 2003.

[61] C. Scott. "Dynamic decision tree." *published phd thesis*, Huston, Texas:, 2004.

[62] J. Quintana. "Decision tree for indication of total hip replacement on patients with ostearthritis." *Oxford journal*, pp.1402-1409, England, 2009.

[63] B. T. Joshua, L. G. Thomas and K.Charles. "Theory-based Bayesian models of inductive learning and reasoning." *TRENDS in Cognitive Sciences*, Vol.10 No.7, 2006

# APPENDIXES

# APPENDIX I, II, III, IV

## APPENDIX I

### Interview Questions

After introducing the objective of the study, respondents were requested to participate by responding the interview questions. The answers of the respondents were recorded by using paper and pen for the following interview questions. The following interview questions are the main area to cover how ICT experts diagnose the computer problems:

1. How computer problems are defined from the aspect of ICT experts?
2. What are the method and techniques ICT experts used to identify the causes of computer problem?
3. How the ICT technicians identify the unique characteristics of each problem?
4. What are the common types of problems that occur in the computer?
5. Do problems have common behaviors and their complication due to different factors?
6. What are the decisions of the ICT technician in troubleshooting computer and network problems?
7. What are the most important areas focused by the ICT technician to diagnose the computer problems?
8. What are recommended techniques used to manage specific problems?

# APPENDIX II

## EVALUATION QUESTIONNAIRES

Questionnaire to test and validate the performance of the Computer and Network troubleshooting expert system:

1. Is the system more efficient in running time?
   1. Poor 2. Fair 3. Good 4. Very good 5. Excellent
2. Does the system incorporate sufficient knowledge to solve a given problems?
   1. Poor 2. Fair 3. Good 4. Very good 5. Excellent
3. Is the system provides the right description and suggestion to be followed while troubleshooting by human expert.
   1. Poor 2. Fair 3. Good 4. Very good 5. Excellent
4. How do you rate the significance of the system in the domain area?
   1. Poor 2. Fair 3. Good 4. Very good 5. Excellent
5. Does the system update its knowledge base?
   1. Poor 2. Fair 3. Good 4. Very good 5. Excellent
6. How is CNTES different from a diagnosis conducted by human expert?
7. What issues are covered by the advisor service of the system?
8. Do you feel any uncovered areas by the prototype about troubleshooting? If you feel please state them.
9. In your opinion, can knowledge based system (such as the rule based presented by CNTES) handle the troubleshooting task of computer system?
10. Do you think that the system incorporate sufficient knowledge to provide troubleshooting service?
11. Does the system have any significance in the domain area?
12. What is the strength of CNTES?
13. What are the limitations of CNTES?

# APPENDIX III

## Sample of Test Case Used to Validate the Accuracy of CNTES System

Case 1, network cable connection problem

| Domain expert decision | System decision |
|---|---|
| Problem: Network cable is not connected | Problem: Network cable is not connected |
| Suggestion: connect the network cable, check any cut on the cable, check the light is blinking | Suggestion: connect the network cable |
| Similar decision is made, but the system does not check, any cut on the cable and connection light indication. | |

Case 2: IP address

| Domain expert decision | System decision |
|---|---|
| Problem: IP address not configured | Problem: IP address not configured |
| Suggestion: assign IP address, identify IP address range, static or dynamic | Suggestion: assign IP address |
| Similar decision is made | |

Case 3: network driver

| Domain expert decision | System decision |
|---|---|
| Problem: network driver is not installed | Problem: network driver is not installed |
| Suggestion: install network driver, | Suggestion: install network driver, assign IP address |
| Similar decision is made, but the expert system provides extra suggestion after installing network driver | |

Case 4: power problem

| Domain expert decision | System decision |
|---|---|
| Problem: no power | Problem: no power |
| Suggestion: check source power, switch on divider, computer power on button is off, internal power cable is disconnected, damaged power supply, check physical condition of the power cables. | Suggestion: check source power, switch on divider, computer power on button is off, internal power cable is disconnected, damaged power supply |
| Similar decision is made, but the expert system does not check the physical condition of the power supply. | |

A problems and suggestions for the system side is taken as it described in the decision tree structure in the conceptual modeling part of the thesis.

# APPENDIX IV

## Sample codes of the knowledge based system

/*----------rules to identifying the case----------*/

*Problem:*- have you faced PC problem (yes/no)

*Problempcnet*:- Is it network or pc problem? net/pc') (pc/net)

% if it is network problem and net is entered

*Network*:- is the network cable is connected? (yes/no)

Is there IP address for your computer? (yes/no)

Is the network driver is installed? (yes/no)

*Netsolution:*- write('connect cable then check your ip address by using IPconfig command,

if no ip address, identify your LAN IP address range in your institution (static ip address) type it manually.

if there is proxy server, type proxy address including port #, e.g cache.aau.edu.et/port 8080.

if DHCP server is enabled select automatic option from the nework cofigration wizard.

still not solved, contact your institution network administrator

% if it is computer problem and pc is selected

Pcproblem:- is your computer faced problem?(yes/no)

Is the source power is switched on? (yes/no)

Is the power on button is on/off? (yes/no)

Powersolution:- the source power is switched on and the computer power on button is on there may be damaged power supply so consider replacing the power supply.

## DECLARATION

*This thesis is my original work and has not been submitted as a partial requirement for a degree of master in any other university*

---

Amanuel Ayde
June 2014

*This thesis has been submitted for examination with my approval as a university advisor.*

---

Advisor: WORKSHET LAMENEW