



UNIVERSITY OF NAIROBI

KNOWLEDGE-BASED CAREER CHOICE ASSISTANT

BY

KIPROTICH KENNIE KEMBOI

A PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF
REQUIREMENTS OF MASTER OF SCIENCE IN COMPUTER SCIENCE

JULY, 2013

Declaration

Declaration by the Student

This report is my original work submitted to Nairobi University in partial fulfillment of requirements of Master of Science in Computer Science and has not been presented for a degree in any other university.

SIGN..... DATE.....

KENNIE KIPROTICH KEMBOI, UNIVERSITY OF NAIROBI

Registration Number: P58/63423/2010

Declaration by the Supervisor

This report was submitted with my approval as the Nairobi University supervisor.

SIGN DATE.....

DR. ELISHA T. OPIYO OMULO, LECTURER - UNIVERSITY OF NAIROBI

JULY, 2013

Abstract

In this report we will present the design and development of a proposed knowledge-based system prototype that aims to assist the students select the best suitable career choice program in Institutions of Higher Learning in Kenya. The current existing approach in Kenya is where the students get career advice through their teacher, motivators and career Counselor. The basic idea of our approach is designing an artificial intelligent system model using JESS plugins in eclipse IDE that accept KCSE results by enabling the students to enter grades for different subjects combinations and applying the module results to a rule-based knowledge base system to determine the appropriate or suitable career for the student(s) based on the minimum requirements for particular course with the available institutions in Kenyan. The result is shown as a list of suggested career course that is most suitable with the student KCSE results. Lastly, the prototype is evaluated, a questionnaire form is prepared and issued randomly to selected group of students and they are allowed to use the system and give their feedback, this enable us to evaluated the usability, efficiency and effectiveness of the system and most students were satisfied with the various career course the system advise them to take.

KEYWORDS: Artificial Intelligence; Knowledge-based system; Expert System; Career Choice Assistant.

Table of Contents

Declaration.....	i
Abstract.....	ii
Tables.....	vi
Figures.....	vii
Glossary of Terms.....	viii
CHAPTER ONE: INTRODUCTION.....	1
1.1 Background.....	1
1.2 The Problem statement	2
1.3 Research Questions.....	4
1.4 The Objectives of the Study.....	4
1.5 The systems objectives	4
1.6 The contribution.....	4
1.7 The Significance of the study	4
1.8 The Scope and Limitations of the Study.....	5
1.9 Definitions of Terms	5
1.10 Chapter Summary	7
CHAPTER TWO: LITERATURE RIVIEW	8
2.1 Introduction.....	8
2.2 Career choice	8
2.3 Factors in Career Choice.....	9
2.4 Education and Career Aspirations	10
2.5 The proposed application of artificial intelligence systems available for student's selection process in Kenya and their limitations	11
2.5.1 Artificial Neural Networks (ANN)	11
2.5.2 Process of Artificial Neural Networks	11
2.5.3 Limitation of Artificial Neural Network (ANN)	12
2.6 Knowledge-based systems.....	12
2.6.1 Brief history of knowledge-based.....	12
2.6.2 Definition of knowledge base system	13
2.7 Major Fields of Artificial Intelligence.....	15
2.8 Introduction of the Traditional Expert System	15
2.9 The Design of Rule base Knowledge base system	16
2.10 Structure of a Rule base knowledge base System	17
2.11 The Production System in knowledge base systems Problem Solving	18

2.12 Knowledge base.....	20
2.13 Major Components of Knowledge base	21
2.14 Characteristics of a knowledge base system.....	22
2.15 Knowledge base systems comparison	23
2.16 Knowledge and Knowledge base systems	24
2.17 Characteristics of Knowledge Acquisition	25
2.18 Knowledge and Rules	27
2.19 Rules and Knowledge Representation	27
2.20 Introduction to Knowledge Representation	28
2.21 Ontology as a Specification Mechanism	31
2.22 Knowledge-based system methodology	32
Introduction.....	32
2.22.1 Knowledge engineering methodology	32
2.22.2 Software engineering methodology	34
2.23 Turban’s account of the knowledge base system development lifecycle	41
2.24 Advantages of knowledge-based system	45
2.25 Related work	45
2.27 Architecture of Knowledge-based career choice assistant	46
2.29 Chapter Summary	48
3.1 Introduction.....	49
3.1 Differences between System Development Life Cycle (SDLC) and Expert/Knowledge-based System Development Life Cycle [E/KB SDLC]	50
3.1 Expert /Knowledge-based System Development Life Cycle (E/KB SDLC)	50
3.2 Steps in Developing Knowledge-based System	50
3.2.1 Preliminary requirements.....	51
3.2.2 Selection of expert system tools	51
3.3 Design	52
3.3.1 The User Interface.....	52
3.3.2 The Inference Engine	53
3.3.3 Knowledge representation (The Knowledge Base)	53
3.3.5 Prototype development	54
3.3.4 Choosing the right implementation.....	54
3.4 Testing (case study identification, field testing).....	54
Why use UML.	55
3.4.1 Evaluation	55
3.4.2 Documentation.....	55
3.5 Maintenance and update	55
4.1 Introduction.....	57
4.2 Functional and Non- functional requirements	57
CHAPTER FIVE: PROJECT DESIGN AND IMPLEMENTATION	62
5.1 Structure of the knowledge base career choice locator and advisory system.....	62

5.1.1 The User Interface.....	62
5.1.2 The Inference Engine	62
5.1.3 The Knowledge Base	62
5.2 Choosing the right implementation	63
5.2.1 Code from the ground up	63
5.2.2 A suitable knowledge base system language.	63
5.2.3A web-enabled Knowledge base system language	63
5.2.4A web-enabled <i>shell</i> and a pre-made user interface.	63
5.3 The knowledge base System Model	65
5.3.1 Use case diagram	65
5.3.1 Unified Modeling Language	66
5.4 The Behavioral Model	67
5.4.1 Sequence diagram for the Proposed System.....	67
5.4.2 Activity diagram for the Proposed System	68
CHAPTER SIX: EVALUATION.....	72
CHAPTER SEVEN: THE SUMMARY AND APPENDIX	75
7.1 SUMMARY	75
7.2 APPENDIX.....	76
7.2.1. Semantic Network Example	76
7.2.2. Results generated by the system when run in command window	76
7.2.3. Knowledge-based Questionnaire form	78
7.3. CONCLUSION AND FUTURE WORK	79
CHAPTER EIGHT: REFERENCES	80

Tables

Table 1: Knowledge base System comparison	23
Table 2: A qualitative & quantitative evaluation criteria of an expert system	39
Table 3: Questionnaire Results from different groups of users	73
Table 4: Questionnaire table used by the students when evaluating the knowledge-based career choice assistant.....	78

Figures

Figure 1: Conceptual model of an artificial neural network (Franklin Wabwoba & Fullgence M. Mwakondo, volume 10 2011).	12
Figure 2: Architecture of a typical Knowledge- based system for a particular problem domain, adapted from Luger (2005).	15
Figure 3: Diagram of Basic Rule base knowledge System.....	17
Figure 4: A diagram of Common Rule Base Knowledge base System	18
Figure 5: Diagram Production System Model	18
Figure 6: A production system. Control loops until working memory pattern no longer matches the conditions of any productions, adapted from Luger (2005).	20
Figure 7: Knowledge base systems Development Team	25
Figure 8: Spiral Model for Expert System Development	32
Figure 9: The steps in a typical knowledge-based systems analysis and design methodology	42
Figure 10: An architecture of Career Choice & Locator Advisory knowledge-based System.....	47
Figure 11: Knowledge-based System Development Life Cycle (Awad, Elias M., 1996, p97).	51
Figure 12: The requirements engineering process	57
Figure 13: web based knowledge base system	64
Figure 14: The servlet-based knowledge base system.....	65
Figure 15: Use Case Diagram of the KBS System	65
Figure 16: UML knowledge- based Career choice Locator & Advisory system	67
Figure 17: Sequence diagram of the knowledge-based system	68
Figure 18: Activity diagram of the KB System	68
Figure 19: Semantic Network diagram	76

Glossary of Terms

KCPE – Kenya Certificate of Secondary Education

KCSE – Kenya National Examination Council

JAB – Joint Admission Board

ANN - Artificial Neural Networks

IS – Intelligent System

XML – Extensible Mark-up Language

RDF – Resource Description Framework

Web CLIPS /JESS – Java Experts System Shell

Common KADS – Knowledge Acquisition and Documentation Structuring

CHAPTER ONE: INTRODUCTION

1.1 Background

Career choice is a complex decision for students since it determines the kind of profession that they intend to pursue in life. As students try to make career choice while in secondary school, they face problem of matching their career choices with their abilities and school performance. In Kenya every year form four secondary school students make career choices before sitting for their final examination (Kenya Certificate of Secondary Education examinations) however, studies indicate that most students enter into careers that are totally different from the ones they chose while in secondary.

Influential factors, however, state no variance was reported for persons influencing career choice by gender. Career selection is one of many important choices students will make in determining future plans. This decision will impact them throughout their lives. The essence of who the student is will revolve around what the student wants to do with their life-long work. (Basavage, 1996, p.1)

Every student carries the unique history of their past and this determines how they view the world. Career choice must be brought into a clearer focus, starting with students in elementary school and continuing beyond. Students seem to make high school a watershed for making the big decision. Career choice is an ever-evolving process. Career choice is a process that includes experimentation, trial and error, decision-making and eventually judgment. Students must be made aware of this process; it has yet to be perfected.

The 8-4-4 system of Education seems have been the bench mark in shaping the life of the young adults in terms of pursuing a certain line of career.

Every year student sit for the KCSE examination at their final year at high school and the KCSE will be used as standard and once the results are released by KNEC in collaboration with the ministry of education the JAB under the selection criteria policy to select the successful students with a cut-off of B+ to various public Universities and allocate them career course based on what the student selected and sometimes they allocate them randomly. They later find the courses that were assigned to them beyond their scope, get overwhelmed, and keep on re-sitting for exams (“Joint Board Should Review,” 2009).

It is estimated that only 20% of the qualified students eventually get admitted into local universities every year (Munavu, Ogutu, & Wasanga, 2008). The selection process for university admission has denied many capable students a chance in a public university.

Information and Communication Technology (ICT) has the potential to provide tools / technologies for building suitable systems. In as much as ICT is changing in itself, it also makes enhancements in technology that can cope with the changes from other applications. This would allow ICT applications to cope with the changing needs like those arising out of the dynamically changing course requirements for university admissions. Innovations in the field of Information Technology (IT) continue to increase at an ever spiraling rate; advances in operating systems, software, communication devices, and methodologies are, renovating the inventory of IT products on a near daily basis (Gillard, Bailey, & Nolan, 2008).

The Internet was opened to general users in 1994 and this new era of information and communication technology has played an important role in the field of knowledge-based systems. The Web technologies allowed the knowledge engineers and domain experts to build the knowledge-based systems that were having dynamic knowledgebase capabilities [Marwaha *et al.* 2002]. The domain experts could update the knowledge at the central server and the users had an access to the recent knowledgebase through a Web interface.

1.2 The Problem statement

Everyone has many special specific mental abilities, and every university faculty/major needs special abilities and capabilities. So we need a tool to measure some of the student mental abilities accurately and compare then with the university faculty/major required abilities and suggest a suitable faculty/major for the student to be academically successful.

A study done by a Ministry of Education (Kenya) official in collaboration with researchers from a Canadian University in 2000 concluded that, among all agents that influence career choice by high school students in Kenya, career guidance counselors have the least influence from the fact that they are too few and poorly trained.

Kenyan graduates even at and after University, still do not know what their career choice is: they do not know what they have trained for, what to do with their lives or where else to go.

In the second year of training at the University they are still changing the courses, others even at the end of the course are still complaining that the course they wanted to do or they would have chosen is not the actual one they trained in; that some of them were forced by their parents to take certain course.

For the next decade and half, the nation has continued to churn out graduates of its school systems people with minimal survival skills. Results from KCPE and KSCE examination testify to this, as in both, less than half of those who sit these exams, progresses to the next

level. For example 2008 graduates, only 20,000 of them got admitted to the public Universities, leaving out a total of 52,500 students who attained the minimum required grade of C+ (Siringi, 2010). Only less than 10% enroll in Universities, while 90% locked out, bearing in mind the high secondary school dropout rate, and that more than 50% score grade D in KSCE (The Stateman Group, 2009).

Private as well as the Polytechnics Universities offers various courses (which have never been exploited) attract students with mean grades as low as D+, which could ease the dropout.

Due to poor career guidance in high school or lack of it, college admission in Kenya largely places students in college courses not best suited for them. As a result, there is mismatch between occupations most professionals have and the occupations they would have felt most comfortable in.

According to (Ojenge Winston) career guidance practices in Kenyan high schools include:

1. Private Academies - Trips to university fairs; Individual attention to students; Career debates among students
2. High Cost Public Schools - Counselor discusses with interested student. No full time counselor as each counselor has teaching load too
3. Other Public Schools (nearly 90% of all Kenyan high school students). No real counseling done, only instruction on use of the careers booklet, which contains a list of college courses and their cut-off points

Therefore there is need of an appropriate College-course Admission Model to solve the above problem.

The following are the major problems facing the form four graduates:

- i. Counseling is done before the student attempt the KCSE but no counseling after result is released by KNEC
- ii. JAB: Cut-off criteria of KCSE result of B which makes 20% of form four graduates leaving out 80% also JAB make the choices for the students in the cases vacancies exist do not match with the student's choice.
- iii. Lack of intelligent system to assist the form four graduates in career choice advice & location of an institution where the course is offered after the KNEC released the KCSE results

These are the pertinent questions that the Knowledge-based career choice assistant must address these problems by providing a link between the graduate and an intelligent/expert system in terms of matching with appropriate course(s) in other Public, Private and/or Polytechnic Universities

1.3 Research Questions

- i. Can JESS be used to develop a career choice assistant?
- ii. Evaluate career choice assistant to check the potential user and satisfied with it

1.4 The Objectives of the Study

- i. Determine the systems requirements.
- ii. Develop a model of knowledge- based Career Choice Locator and Advisory System that integrates Knowledge from necessary knowledge sources and recommends suitable college courses in a specific college.
- iii. Implement a knowledge base system that incorporates various databases; subjects, courses, universities.

1.5 The systems objectives

- i. Develop a Knowledge base system that can guide and allocate appropriate course for the student.
- ii. Design a web-knowledge based system prototype to demonstrate some of the functionalities of the model.
- iii. Implement the model using JESS plug-ins on eclipse IDE
- iv. Test the model with JESS Plug- ins using eclipse IDE
- v. Analyze and Evaluate the model using the KCSE results in making the suitable career choice for the student(s)

1.6 The contribution

The main contribution of this project to form four graduates student is: a web Knowledge-based advisory system that helps the student make appropriate course in a respective institution of higher learning without necessarily having to consult or make random career course which does not march with his/her capability and ability.

1.7 The Significance of the study

Due to lack of appropriate advice system or career guidance to form four students after their KCSE results are out several students with different abilities and capabilities have been denied or lock out hence cannot get chance to exploit skill through training in the available institutions in Kenya

One of the MDG was education for all from primary secondary and University where the student put his/her talent into the mainstream of production by having the right career choice. This have called for an expert system career choice Locator and Advisory system that will march their KCSE results with the best career.

KCSE result is an accepted standard in Kenya for allocating the students courses they would like to join in various institution of higher learning and Career Choice Locator and Advisory System will use KCSE result to assist the student make informed choice in career.

JAB system have cut-off point points for those who are to join public Universities in Kenya which currently cater for 20% and the remaining 80% are left to seek advice from parents, or trial and error basis on the career they can pursue and even those selected by JAB are dissatisfied with the courses allocated hence this system will be of great importance to this group.

The study will seek to enable form four graduates access the system using their mobile phones. They will also have the option of accessing the system through the web by using laptops or Desktop Computers connected to internet.

1.8 The Scope and Limitations of the Study

- a) The system will only accept KCSE result and give the appropriate advice for the form four graduates' students who would wish to pursue their career in the institution of higher learning in Kenya and to some extend Mock results.
- b) Student who qualify to join Universities through JAB and not satisfied with the course they have been selected for can use the system for more advice on appropriate courses
- c) Students in rural places in Kenya who are not aware of which course they can pursue and are not aware of which Universities to pursue those courses this system is appropriate.
- d) Information required for this system will be contained in individual databases in a decentralized manner.
- e) Students have mobile phones, laptops or computers connected to internet.
- f) The mobile phones to be used will support java based applications.

1.9 Definitions of Terms

Domain: A possible problem space in which searching or reasoning techniques can be applied.

Knowledge-based system: A computer program that emulates the thought process of a human expert.

Expert system shell: The user interface to an expert system.

Explanation facility: An expert system component that reproduces the logic the inference engine followed to reach its conclusion.

Heuristic rule: A specific rule of thumb or common sense that can be used to restrict a search to a subset of a problem domain.

Heuristic search: A search technique that applies heuristics to reduce the size of a problem domain.

Heuristics: General rules derived from experience, common sense, inferences, and intelligent trial and error.

Inference engine: The component of an expert system that uses input parameters to access the knowledge base, reach a conclusion, and offer expert advice.

Knowledge acquisition facility: A set of software tools for capturing and encoding a human expert's expertise and creating a knowledge base.

Knowledge base: A collection of data, algorithms, and heuristic rules that forms the core of an expert system.

Knowledge engineer: A person who captures and encodes a human expert's expertise and creates a knowledge base.

Machine learning: The capacity of a machine (or an expert system) to "learn" from experience.

Prototype: A reasonably complete, working model of a system.

Reasoning: The act of using inference to lead to a conclusion based on existing knowledge and/or data.

Reasoning capability: An inference engine feature that reaches a conclusion by applying the rules in the rule base

Relation: An association or link between two objects or entities.

Rule: A formal specification or description of a unit of knowledge.

Rule base: A collection of executable rules; the rule base is accessed by the inference engine to support reasoning.

Symbolic reasoning: A technique for performing reasoning or inference with symbolic data such as graph, image, and/or picture.

Symbolic representation: A technique for representing symbolic data or knowledge

Career choice: The development of knowledge, skills and attitudes through a planned program of learning experiences that will assist students to make informed decisions about their study and/or work options and enable effective participation in working life. Career education encompasses; self-awareness in relation to interests, abilities, competencies and values

Artificial Intelligence (AI): Is a branch of *Science* which deals with helping machines finding solutions to complex problems in a more human-like fashion.

1.10 Chapter Summary

Section one highlights the information related career choice problems by identifying the challenges faced by students after KCSE results are released by the ministry of Education in Kenya and more so student from rural areas despite having mobile phone with internet enabled..

Section two states the main purpose of this project by specifying the problems that have led to the development of this system.

Section three lists the objectives that are expected to be met at the end of this research.

Section four outlines the importance of this study to students in relation to career choice problems. Section five identifies the extent to which this system cover together with the shortcomings that will be expected from the study. Section six describes the various terminologies used in the study.

CHAPTER TWO: LITERATURE RIVIEW

2.1 Introduction

This chapter reviews the literature that described the student career choice selection process, Location and the application of knowledge-based system. The body of literature available for review encompassed many volumes.

A comprehensive literature review was conducted to answer the following six questions:

1. Definition of career choice and what are the limitation form the four graduates student faces
2. Available proposed systems that exist in Kenya for selection process and their inadequacies
3. Definition of an Knowledge base system
4. Applications of Knowledge base systems
5. Characteristics of Knowledge base systems
6. Methodologies of an Knowledge base systems
7. Advantages of Knowledge base systems

2.2 Career choice

Career choice has become a complex science with the advent of information technology, the emergence of post industrial revolution and job competition. It was a common practice in the old days to find feudalism converting it into a family affair where the son of a blacksmith was destined to become a blacksmith and a feudal was born a leader. Industrialization and post industrialization has made it possible for a common person to be richer as long as she or he has due skills and knowledge (Wattles, 2009). Today, one has not only to make due career planning but also exhaustive career research before making a career choice so as to adjust with the evolving socio-economic conditions (Wattles, 2009). Most of students who are secondary schools do not have accurate information about occupational opportunities to help them make appropriate career choice. According to (Kerka 2000), career choice is influenced by multiple factors including personality, interests, self-concept, cultural identity, globalization, socialization, role model, social support and available resources such as information and financial. (Bandura et al 2001) state that each individual undertaking the process is influenced by several factors including the context in which they live in, their personal aptitudes, social contacts and educational attainment. According to (Hewitt 2010),

factors influencing career choice can either be intrinsic or extrinsic or both. Hewitt further states that most people are influenced by careers that their parents favour, others follow the careers that their educational choices have opened for them, some choose to follow their passion regardless of how much or little it will make them while others choose the careers that give high income. Students perception of being suitable for particular jobs also has been found to be influenced by a number of factors including ethnic background, year in school, level of achievement, choice of science subjects, attitudes and differences in job characteristics (McQuaid and Bond, 2003).

According to (Oyamo and Amoth 2008): *A case study of Kisumu Municipality, Kenya*, where most graduates of education and other programmes are working in careers that they did not choose before joining university.

Studies in Kenya show that rural students tend to seek help from parents more than urban students and that parents more than teachers play a major role in the career choice of students. Generally, the choice of a career is influenced by parents, friends, and counselors however variations occur from one population to the other. In Kenya, every year form four secondary school students make their career choices before sitting for their final Kenya Certificate of Secondary Examination. The result of this final examination determines who joins university since admissions into various careers are determined by grades obtained from the Kenya Certificate of Secondary Education.

Before making their career choices, students are often provided with a list of careers from which they are supposed to make choices. Most of the students lack adequate information regarding various careers hence the choices that they make are embedded in their perception of the ideal job and the subjects they study in secondary school. The only support students get within the school is from career masters or counselors as they are mostly referred to and the teachers who are expected to support students in their career choice. When the final examination results are released by the Ministry of Education, and depending on the grades, students are then admitted to the universities based on the career choices that they had made while in school. When these students graduate from the universities, some of them enter into occupations that are totally different from the ones they had chosen and trained for.

2.3 Factors in Career Choice

The first factor in career choice, **Environment** may influence the career students choose. For example, students who have lived on an island may choose a career dealing with the water, or

they may choose to leave the island behind, never to have anything to do with water again. Maybe someone in the student's life has made a significant impact or impression, leading to a definite career choice. Parents' educational background may influence student views on whether or not to continue their education. Someone they saw on television may have influenced the student, or parents may have demanded that they assume a family business. These are various environmental factors that would lead a student to a chosen career.

Opportunity is the second factor that has shaped career choices for students. Opportunity may influence how students have perceived their future in terms of the reasonable probability of a future in particular career fields.

How students have seen themselves in a role in which **personality** is a determining factor may influence a chosen career. Some careers demand that you have the personality to match the qualities of the occupation. For example, sales people have to be outgoing. (Splaver, 1977) said "personality" plays an important role in the choosing of the right career. A student's personality must be a self-motivated type, as to investigate career possibilities from early on in their lives, and not the procrastinating type that waits until they are compelled to decide. Students must take seriously the role grades play in limiting opportunities in the future. Splaver went on to say, "It is important for you to have a good understanding of yourself, your personality, if you are to make intelligent career plans" (Splaver, 1977, p.12).

2.4 Education and Career Aspirations

In Kenya's educational setting, it is significant that a majority of students are unaware of training opportunities and requirements at various levels of post-secondary schools education (Kithyo and Petrina, 2005; Kasomo, 2007; Boniface, 2009; Nthangi, 2007; Okumu, 2009). Career information and guidance need to be provided and accessed by girls in secondary schools in order for them to make well informed decisions that will help them make smooth transitions from school to higher institutions; from school to work, and from school to lifelong learning in this twenty-first century. (Gysberg, 2007) captures this 21st century scenario as a time when organizations are developing global identities; and technological change is rapid. (Okumu, 2009) states that with the complexities of unemployment and the technological shifts in today's labour market, young people need not only information but also skills they can get to ensure they make sound decisions

2.5 The proposed application of artificial intelligence systems available for student's selection process in Kenya and their limitations

2.5.1 Artificial Neural Networks (ANN)

Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. As in nature, the connections between elements largely determine the network function (Demuth, Hagan, & Beale, 2009). An artificial neural network (ANN) consists of a collection of processing elements that are highly interconnected and transform a set of inputs into a set of desired outputs. The result of the transformation is determined by the characteristics of the elements and the weights associated with the interconnections among them. By modifying the connections between the nodes the network is able to adapt to the desired outputs (Cannady, 1998).

An ANN can be trained to perform a particular function by adjusting the values of the connections (weights) between elements. A pair of actual input and desired output is required to train the artificial neural network. Neural networks have been trained to perform complex functions in various fields, including pattern recognition, identification, classification, speech, vision, and control systems (Demuth et al., 2009). The neural network gains the experience initially by training the system to correctly identify pre-selected examples of the problem. The response of the neural network is reviewed and the configuration of the system is refined until the neural network's analysis of the training data reaches a satisfactory level. Training of the network is conducted until outputs of the network satisfy the desired target or until the network reaches desired performance as indicated by some error level (difference between output and desired target of network). Usually this error is formulated as mean square error (MSE) (Haryanto, Setiawan, & Budiyo, 2007). In addition to the initial training period, the neural network also gains experience over time as it conducts analyses on data related to the students' course selection as per their preferred choice.

2.5.2 Process of Artificial Neural Networks

ANNs are mathematical entities or models based on the basic building block known as the artificial neuron and usually modeled after the biological neuron found in the brain. The artificial neurons making up the ANN are distributed processors and operate in parallel. In fact, an ANN resembles the brain in two aspects: it is able to acquire knowledge through learning and to store the knowledge through interconnection strengths known as synaptic weights (Haykin, 2004).

The ANN process basically consists of input data intended to be processed in a certain desired way, computational processors comprising several neurons that are capable of extracting important features contained in the input data, output of computational processes, expected target values, and control mechanism for adjusting the parameters values (weights and biases) not determined by a student's performance in the exam. Figure 1 below shows the conceptual model of ANN highlighting its main elements.

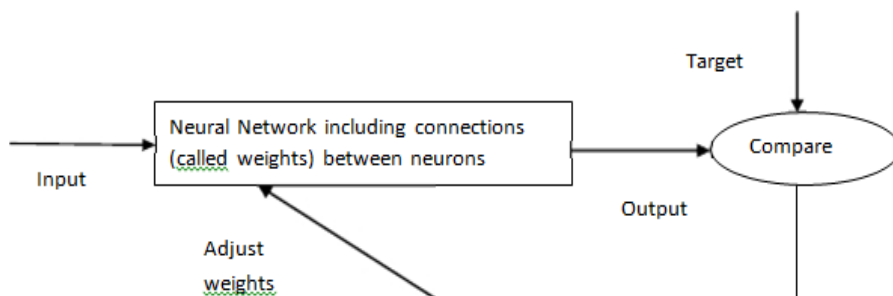


Figure 1: Conceptual model of an artificial neural network (Franklin Wabwoba & Fullgence M. Mwakondo, volume 10 2011).

2.5.3 Limitation of Artificial Neural Network (ANN)

- i. Neural network builders claim that they do not model human intelligence, do not program solutions
- ii. Do not aim to solve specific problems per se and
- iii. Instead, neural networks designers seek to put intelligence into the hardware in the form of a generalized capability to learn

2.6 Knowledge-based systems

2.6.1 Brief history of knowledge-based

Knowledgebase system is a subfield of artificial intelligence (AI), and was first conceived by Edward Feigenbaum, now considered the father of knowledge-base, who with other colleagues and associates built the first successful expert system in the late 60's at Stanford University (L. Steel, Russel & Norvig , 2003). It was called the "Dendral" system, a portmanteau of the term "Dendritic Algorithm". Dendral was meant to emulate organic chemists to help automating the process of identifying unknown organic molecules (Lindsay *et al*, 2008)

Knowledge-base was developed by artificial intelligence researchers during the late 1970s. The early applications were originally termed "knowledge-base" because they were intended to approach problem solving and analysis in a similar manner, and with similar results, as that of human experts.

2.6.2 Definition of knowledge base system

The definitions of expert system depict clearly five important aspects namely:

- i. *automated reasoning system*: that attempts to mimic the performance of the human expert
- ii. *computer software*: that emulates the decision-making ability of a human expert, or whose performance is intended to rival that of human experts while being highly domain specific
- iii. *computer applications*: which embody some non-algorithmic expertise for solving certain types of problems
- iv. *computational tools*: that mimic or at least try to mimic the human thinking process

2.6.3 Knowledge-based Systems: A definition

A system that draws upon the knowledge of human experts captured in a knowledge-base to solve

- i. Problems that normally require human expertise.
- ii. Heuristic rather than algorithmic
- iii. Heuristics in search vs. in KBS
General vs. domain-specific
- iv. Highly specific domain knowledge
- v. Knowledge is separated from how it is used

$KBS = \text{knowledge-base} + \text{inference engine}$

They have been successful in many fields. In general there are two types of problems:

- (a) Select one of several hypotheses (e.g. diagnosis and advice); and
- (b) Make a solution that meets requirements (e.g. design and planning).

An expert system is and/or sometimes referred to as knowledge-based system *computer software* that emulates the decision-making ability of a human expert (Jacksons & Peters, 2010) or whose performance is intended to rival that of human experts while being highly domain specific that are

✓ *Subjective*

- ✓ *Poorly formalized, and*
- ✓ *Require manipulating larger numbers of poorly related facts*

It can be used to record and distribute scarce expert knowledge, to apply the expert knowledge to remote locations, to ensure the quality of problem solving, and to train experts out of ordinary people.

Knowledge-base is a branch of Artificial Intelligence (AI) (Azaab S. Abu Naser S. 2000). AI is a simulation of human intelligence in a machine, so as to make the machine efficient in identifying and using the right piece of “Knowledge” at a given step of solving a problem (B.J. Wielinga *et al*, 2004).

Knowledge-based systems are *computer applications* which embody some non-algorithmic expertise for solving certain types of problems. A rule-based system is an knowledge base system that contains a general rule-base and an inference engine. The inference engine retrieves rules from the rule-base to solve new problems based on the rules for similar problems stored in the rule-based system. In this way, a rule-based system can exhibit humanlike performance in that knowledge that can seemingly be acquired through experience (B. Chadrsekaran, 2011).

Knowledge-base are *computational tools* that mimic or at least try to mimic the human thinking process and the human capability for solving problems and decision making by using knowledge bases and rules from a desired domain. They are appropriate for solving badly or poorly structured problems such as career problems, because their solution can be based on heuristic and empirical knowledge. They are capable of incorporating mathematical models, empirical knowledge and expert judgments, engineering intuition, heuristic rules and needed information, in order to provide a useful advice to the user that can help him make the best decision related to the problem.

According to (Bedi, P., Marwaha, S. 2004), there are two types of knowledge-base: knowledge based knowledge-based expert systems (KBES) and knowledge-based neural networks (NN). Often a combination of knowledge-base and a neural network, called the hybrid expert system or expert network, is used. The knowledge based expert system can be the rule based; the rule inducing or the case based expert system.

Knowledge-based system consist of six subsystems (Shu-Hsien, Liao, 2005) as shown in **figure 2**: knowledge acquisition, knowledge base, working memory, inference engine, explanation system and user interface.

The inference engine makes conclusion based on the knowledge base and applied IF-THEN rules.

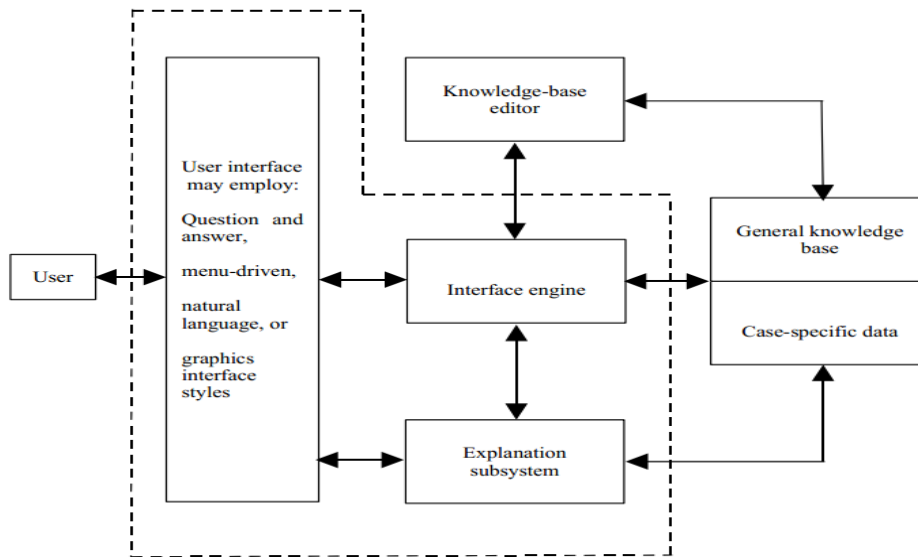


Figure 2: Architecture of a typical Knowledge- based system for a particular problem domain, adapted from Luger (2005).

2.7 Major Fields of Artificial Intelligence

AI has been growing into some fields, which requires special attention to conquer. Expert/knowledge- based System is the most well developed field of AI that many practical products are available in the market today. Other fields of AI are Natural Language Processing, Speech (Voice) Understanding, Robotics and Sensory Systems, Computer Vision and Scene Recognition, Intelligent Computer-aided Instruction, Automatic Programming, Intelligent Workstations, Summarizing News, and Translation from One Language to Another (Turban, Efraim, 1992, p21-27).

2.8 Introduction of the Traditional Expert System

The rationale behind “knowledge-based” problem solvers was that human experts knew a lot about their area of expertise. Knowledge-based systems designers acquire this knowledge from human experts and then program the system to emulate the human expert’s methodology.

These human experts also augment the system’s knowledge with tricks, shortcuts and heuristics that they have gained from experience. These heuristics can also be encoded using probabilistic methods. Knowledge-based systems are built to solve a wide range of problems

in domains such as medicine, mathematics, engineering, chemistry, geology, computer science, business, law, defense, and education. The range of problem categories can be summarized as follows: According to (Waterman 1986).

Interpretation: Forming high-level conclusions from collections of raw data

Prediction: Projecting probable consequences of given situations

Diagnosis: Determining the cause of malfunctions in complex situations based on observable symptoms.

Design: Finding a configuration of system components that meets performance goals while satisfying a set of design constraints

Planning: Devising a sequence of actions that will achieve a set of goals given certain starting conditions and run-time constraints.

Monitoring: Comparing a system's observed behavior to its expected behavior

Instruction: Assisting in the education process in technical domains

Control: Governing the behavior of a complex environment

Generally, knowledge-based systems programs tend to support the iterative development methodology. This requires that programs be easily prototyped, tested and changed. Easy modification of the knowledge base is vital for successful expert system design. Also, it is important that the expert system can display all the intermediate problem solving steps and can justify choices and decisions. These explanations are important for a human expert, such as a doctor or an engineer, if he is to accept the system's recommendations.

2.9 The Design of Rule base Knowledge base system

Figure 2 shows the modules that make up a typical expert system. The user interface, often graphical, hides much of the complexity of the system. All the knowledge of a problem domain is encoded in the knowledge-base, which is the heart of the system. Most often, this consists of *if... then... rules*. The inference engine applies the knowledge to the solution of actual problems.

This can be done using a production system (2.10). The knowledge base and inference engine are separated for several reasons:

1. If ... then... rules are a very natural way to represent human problem-solving skills.
2. System designers can focus on capturing problem-solving knowledge without worrying about implementation details.
3. Changes can be made in one part of the knowledge-based without affecting others.

4. The same inference engine can be plugged into different knowledge-base and work on different knowledge-bases.

2.10 Structure of a Rule base knowledge base System

The database includes a set of facts used to match against the IF (condition) parts of rules stored in the knowledge base.

The explanation facilities enable the user to ask the expert system *how* a particular conclusion is reached and *why* a specific fact is needed. An expert system must be able to explain its advice, and its knowledge base.

Figure 3: Diagram of Basic Rule base knowledge System

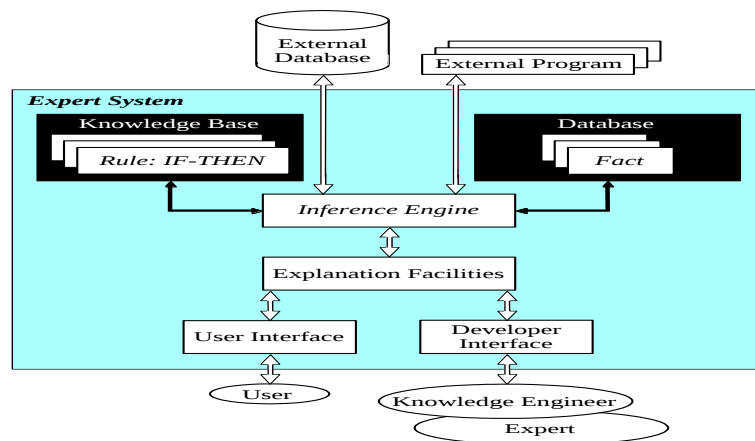


Figure 4: A diagram of Common Rule Base Knowledge base System

2.11 The Production System in knowledge base systems Problem Solving

The *production system* is a model of computation that has proved particularly important in artificial intelligence, both for implementing search algorithms and for modeling human problem solving. A production system provides pattern-directed control of a problem-solving process and consists of a set of *production rules*, a *working memory*, and a *recognize-act* control cycle.

A *production system* may be defined:

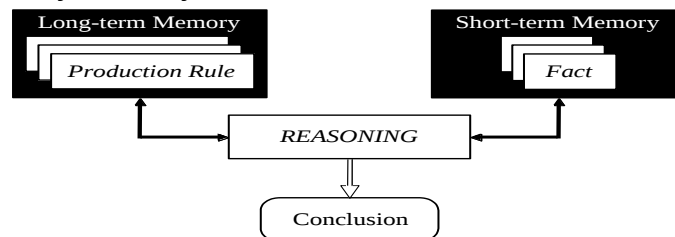


Figure 5: Diagram Production System Model

The set of production rules: These are often simply called *productions*. A production is a **Condition** **action:** pair and defines a single chunk of problem-solving knowledge. Both the condition part and the goal part of each rule is a pattern that determines when that rule may be applied to a problem instance.

Working memory: contains a description of the *current state of the world* in a reasoning process.

This description is a pattern that, in *data-driven reasoning* is matched against the condition part of the set of productions to select appropriate problem-solving actions and in *goal-driven reasoning* is matched against the current goal or sub-goal being explored.

The recognize–act cycle: The control structure for a production system is simple: *working memory* is initialized with the beginning and goal problem descriptions. The current state of the problem-solving is maintained as a set of patterns in working memory. These patterns are matched against the conditions or actions of the production rules; this produces a subset of the production rules, called the *conflict set*, whose conditions (or goals) match the patterns in working memory. After a selected production rule is fired, the control cycle repeats with the modified working memory. The process terminates when the contents of working memory do not match any rules or the problem is solved.

Conflict resolution: is a set of heuristics that is used to choose a rule from the conflict set for The architecture of rule-based knowledge-based systems may be best understood in terms of the production system model for problem solving. The parallel between the two is more than an analogy: the production system was the intellectual precursor of modern expert system architectures, where application of production rules leads to refinements of understanding of a particular problem situation. When Newell and Simon first developed the production system, their goal was to model human performance in problem solving (Newell and Simon 1976).

If we regard the expert system architecture in Figure 5 as a production system, the domain-specific knowledge base is the set of production rules. In a rule-based system, these condition action pairs are represented as *if ... then...* rules, with the premises of the rules, the if portion, corresponding to the condition, and the conclusion, the then portion, corresponding to the action. In data-driven reasoning, when the condition is satisfied, the expert system takes the action of asserting the conclusion as true. In goal-driven reasoning, the then portion of the rule is matched to see what conditions (sub-goals) must be supported for the goal to be true. Case specific data can be kept in the working memory. The inference engine implements the recognize-act cycle of the production system that may be either data-driven or goal-driven.

Many problem domains seem to lend themselves more naturally to forward search. In an interpretation problem, for example, most of the data for the problem are initially given and it is often difficult to formulate an hypotheses or goal. This suggests a forward reasoning process in which the facts are placed in working memory, conditions of rules matching those facts are matched, and the system searches for an interpretation.

In a goal-driven expert system, the goal expression is initially placed in working memory. The system matches rule conclusions with the goal, selecting one rule and placing its premises in the working memory. This corresponds to a decomposition of the problem's goal into simpler sub-goals. The process continues in the next iteration of the production system, with these sub-goals becoming the new goals to match against rule conclusions. The system

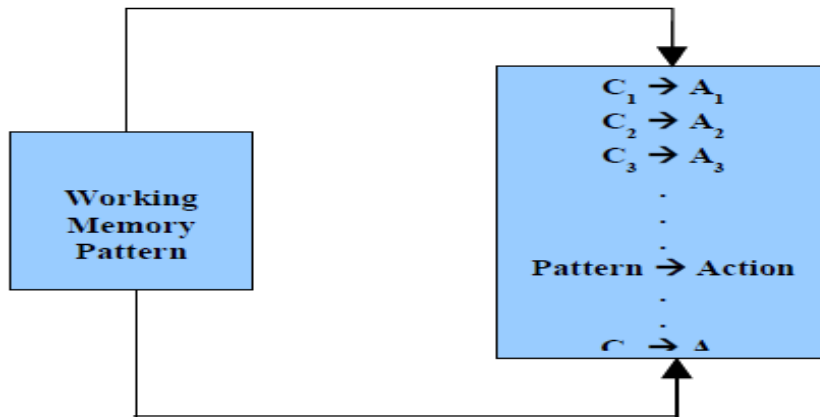


Figure 6: A production system. Control loops until working memory pattern no longer matches the conditions of any productions, adapted from Luger (2005).

Thus works back from the original goal until all the sub-goals in working memory are known to be true, indicating that the hypothesis has been verified. Thus, backward search in an expert system corresponds roughly to the process of hypothesis testing in human problem solving. In an expert system, sub-goals can be solved (when the production rules offer no matches) by asking the system's user for information. Further, some knowledge-based systems allow the designer to specify which sub-goals are to be solved by asking the user. Others simply ask the user about any sub-goals that fail to match rules in the knowledge base; i.e., if the program cannot infer the truth of a sub-goal, it asks the user.

2.12 Knowledge base

In the early seventies, Newell and Simon from Carnegie-Mellon University proposed a **production system** model, the foundation of the modern **rule base knowledge based systems**.

The production model is based on the idea that humans solve problems by applying their knowledge (expressed as production rules) to a given problem represented by problem-specific information.

The production rules are stored in the long-term memory and the problem-specific information or facts in the short-term memory.

The user interface is the means of communication between a user seeking a solution to the problem and an knowledge base system.

KBS tools - Shells

Knowledge-base can be designed using a programming language or an expert shell.

Programming languages used for the expert system design are Lisp, Prolog, C and Pascal, and nowadays C++, Java, Visual Basic and others (). For the design of knowledge-based using programming languages, it is necessary to have programming skills and to know the basic principles and rules of programming.

Expert shells are tools that enable people that do not have a good knowledge of programming to design knowledge-base. They are developed by eliminating knowledge bases from knowledge-based systems used for specific problems or fields, creating the so called “empty” knowledge-base – expert shells. Embedding new knowledge base in these expert shells using the old principles for inference and explanation, the new knowledge-based system can be designed.

Nowadays there is a wide range of different expert shells on the market that can include different types of knowledge-base (rule based, rule inducing, case based knowledge-base and/or neural networks). Some of them are free such as: BABYLON, ES, GEST, CLIPS, RT-Expert for DOS Personal Edition, while the others are for sale: Aion Development System (ADS) ART Enterprise, Doctus KBS, Expert choice, Expert edge, EXSYS, Insight 2+, KEE, Knowledge kraft, M.4, Nexpert Object, OPS83, Personal Consultant, Personal Expert, RT-Expert, XpertRule [6] and JESS (CLIPS) engine.

2.13 Major Components of Knowledge base

Components of Knowledge-based systems may vary from on expert system to another; however, all knowledge-base have four major components, which is knowledge base, inference engine, justifier/scheduler, and user interface (Awad, Elias M., 1996, p58). The knowledge base is the repository of the rules, facts, and knowledge acquired from the human expert (Awad, Elias M., 1996, p58). Inference engine is the ‘brain’ of an expert system; a cluster of computer programs that coordinate the reasoning and inferencing based on the rules of the knowledge base to come up with a solution (Awad, Elias M., 1996, p60). Justifier (Explanation Subsystem) is a component of an expert system that can explain the system’s

reasoning and justify its conclusions (Turban, Efraim, 1998, p858). Scheduler is the port of the inference engine that coordinates and controls the sequence of the rules (Awad, Elias M., 1996, p64). User interface of expert system is a language processor for friendly, problem-oriented communication between the user and the computer (Turban, Efraim, 1998, p448).

2.14 Characteristics of a knowledge base system

- i. An expert system is built to perform at a human expert level in a narrow, specialised domain. Thus, the most important characteristic of an expert system is its high-quality performance.
- ii. No matter how fast the system can solve a problem; the user will not be satisfied if the result is wrong!
- iii. On the other hand, the speed of reaching a solution is very important. Even the most accurate decision or diagnosis may not be useful if it is too late to apply, for instance, in an emergency, when a patient dies or a nuclear power plant explodes.
- iv. Knowledge-based apply **heuristics** to guide the reasoning and thus reduce the search area for a solution.
- v. A unique feature of an expert system is its **explanation capability**. It enables the expert system to review its own reasoning and explain its decisions.
- vi. Knowledge base employs **symbolic reasoning** when solving a problem. Symbols are used to represent different types of knowledge such as facts, concepts and rules
- vii. We should be aware that an expert is only a human and thus can make mistakes, and therefore, an expert system built to perform at a human expert level also should be "allowed" to make mistakes.
- v. In knowledge-base, knowledge is separated from its processing (knowledge base and inference engine are split up). A conventional program is a mixture of knowledge and the control structure to process this knowledge.
- vi. When an expert system shell is used, a knowledge engineer or an expert simply enters rules in the knowledge base. Each new rule adds some new knowledge and makes the expert system smarter.

2.15 Knowledge base systems comparison

<i>Human Experts</i>	<i>Knowledge-base</i>	<i>Conventional Programs</i>
Use knowledge in the form of rules of thumb or heuristics to solve problems in a narrow domain.	Process knowledge expressed in the form of rules and use symbolic reasoning to solve problems in a <i>narrow domain</i> .	Process data and use algorithms, a series of well-defined operations, to solve general numerical problems.
In a human brain, knowledge exists in a compiled form.	Provide a <i>clear separation of knowledge from its processing</i> .	Do not separate knowledge from the control structure to process this knowledge.
Capable of explaining a line of reasoning and providing the details.	<i>Trace the rules fired</i> during a problem-solving session and <i>explain how</i> a particular conclusion was reached and <i>why</i> specific data was needed.	Do not explain how a particular result was obtained and why input data was needed.
Use inexact reasoning and can deal with incomplete, uncertain and fuzzy information.	Permit <i>inexact reasoning</i> and can deal with incomplete, uncertain and fuzzy data.	Work only on problems where data is complete and exact.
Can make mistakes when information is incomplete or fuzzy.	<i>Can make mistakes</i> when data is incomplete or fuzzy.	Provide no solution at all, or a wrong one, when data is incomplete or fuzzy.
Enhance the quality of problem solving via years of learning and practical training. This process is slow, inefficient and expensive.	Enhance the quality of problem solving by adding new rules or adjusting old ones in the knowledge base. When new knowledge is acquired, <i>changes are easy</i> to accomplish.	Enhance the quality of problem solving by changing the program code, which affects both the knowledge and its processing, making changes difficult.

Table 1: Knowledge base System comparison

In essence, knowledge-based do not use traditional programming paradigms to solve problems; rather, they use knowledge which they reason about to draw conclusions and provide solutions. The advantage of knowledge-base 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, knowledge-base have significant applications including medical diagnosis, fault diagnosis, counseling, debugging, game playing, design in complex spaces and problem solving, question-answering, industrial process controlling, climate forecasting,

manufacturing failure analysis, decision support, monitor real time systems, underwrite insurance policies, and perform many other services which previously required human expertise (Isa, Sidek, 2000), and decision making.

2.16 Knowledge and Knowledge base systems

In this section we are going to discuss the following

- i. What is Knowledge?
- ii. What is Knowledge Acquisition?
- iii. The Knowledge base Development Team.
- iv. Rules and Knowledge Representation
- v. Rule-based Knowledge-base.
- vi. Characteristics of Knowledge-base.
- i. **Knowledge**

Knowledge is a theoretical or practical understanding of a subject or a domain. Knowledge is also the sum of what is currently known, and apparently knowledge is power.

Those who possess knowledge are called **experts**.

Anyone can be considered a domain expert if he or she has deep knowledge (of both facts and rules) and strong practical experience in a particular domain. The area of the domain may be limited.

In general, an expert is a skilful person who can do things other people cannot.

ii. Acquiring Knowledge

Knowledge acquisition can be regarded as a method by which a **knowledge engineer** gathers information mainly from **experts**, but also from text books, technical manuals, research papers and other authoritative sources for ultimate translation into a **knowledge base**, understandable by both machines and humans.

The person undertaking the knowledge acquisition, the knowledge engineer, must convert the acquired knowledge into an electronic format that a computer program can use.

iii. Acquiring Knowledge

The important characteristics of knowledge are that it is experiential, descriptive, qualitative, largely undocumented and constantly changing.

There are certain domains where all these properties are found and some where there are only a few.

The lack of documentation and the fact that experts carry a lot of information in their heads makes it difficult to gain access to their knowledge for developing information systems in general and knowledge-base in particular.

Therefore, knowledge engineers have devised specialised techniques to extract and document this information in an efficient and expedient manner.

2.17 Characteristics of Knowledge Acquisition

- Knowledge acquisition is a labour and time intensive process.
- Currently knowledge bases for knowledge based systems are crafted by hand, this is a severe limitation on the rapid deployment of such systems.
- Biggest 'bottleneck' in system development.
- Most expensive part (money, time & labour).
- Automating KA the ultimate goal

Players in the Development Team

There are five members of the expert system development team as shown in the figure below:

- The domain expert
- The knowledge engineer
- The programmer
- The project manager
- The end-user.

The success of their expert system entirely depends on how well the members work together.

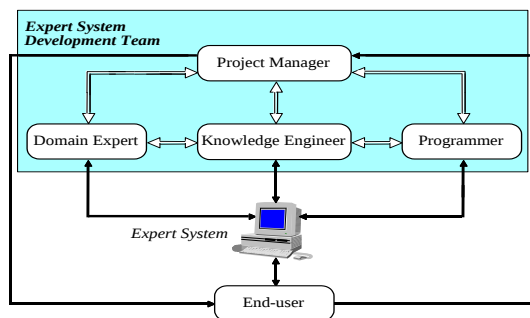


Figure 7: Knowledge base systems Development Team

a. Domain Expert

The domain expert is a knowledgeable and skilled person capable of solving problems in a specific area or domain.

This person has the greatest expertise in a given domain.

This expertise is to be captured in the expert system.

Therefore, the expert must:

- be able to communicate his or her knowledge
- be willing to participate in the expert system development
- Commit a substantial amount of time to the project.

The domain expert is the most important player in the expert system development team.

b. Knowledge Engineer

The knowledge engineer is someone who is capable of designing, building and testing an expert system.

The knowledge engineer's main tasks are:

- Interviews the domain expert to find out how a particular problem is solved.
- Establishes what reasoning methods the expert uses to handle facts and rules and decides how to represent them in the expert system.
- Chooses some development software or an expert system shell, or looks at programming languages for encoding the knowledge.
- Responsible for testing, revising and integrating the expert system into the workplace.

c. Programmer

The programmer is the person responsible for the actual programming, describing the domain knowledge in terms that a computer can understand.

The programmer needs to have skills in symbolic programming in such AI languages as LISP, Prolog and OPS5 and also some experience in the application of different types of expert system shells.

In addition, the programmer should know conventional programming languages like Java, C, Pascal, FORTRAN and Basic.

d. Project Manager

The project manager is the leader of the expert system development team, responsible for keeping the project on track.

The project manager makes sure that all deliverables and milestones are met, interacts with the expert, knowledge engineer, programmer and end-user.

e. End-User

The end-user, often called just the user, is a person who uses the expert system when it is developed.

The user must not only be confident in the expert system performance but also feel comfortable using it.

Therefore, the design of the user interface of the expert system is also vital for the project's success; the end-user's contribution here can be crucial.

2.18 Knowledge and Rules

The human mental process is internal, and it is too complex to be represented as an algorithm. However, most experts are capable of expressing their knowledge in the form of **rules** for problem solving.

IF

THEN the action is go

IF.....

THEN the action is stop

2.19 Rules and Knowledge Representation

The term **rule** in AI, which is the most commonly used type of knowledge representation, can be defined as an IF-THEN structure that relates given information or facts in the IF part to some action in the THEN part.

A rule provides some description of how to solve a problem. Rules are relatively easy to create and understand.

- Any rule consists of two parts:
 - the IF part, called the **antecedent (premise or condition)**
 - And the THEN part called the **consequent (conclusion or action)**.

A rule can have multiple antecedents joined by the keywords AND (conjunction), OR (disjunction) or a combination of both.

<i>IF</i>	< <i>antecedent 1</i> >	<i>IF</i>	< <i>antecedent 1</i> >
<i>AND</i>	< <i>antecedent 2</i> >	<i>OR</i>	< <i>antecedent 2</i> >
	.		.
<i>AND</i>	< <i>antecedent n</i> >	<i>OR</i>	< <i>antecedent n</i> >
<i>THEN</i>	< <i>consequent</i> >	<i>THEN</i>	< <i>consequent</i> >

- The antecedent of a rule incorporates two parts: an object (linguistic object) and its value. The object and its value are linked by an operator.
- The operator identifies the object and assigns the value. Operators such as *is*, *are*, *is not*, *are not* are used to assign a symbolic value to a linguistic object.
- Knowledge-base can also use mathematical operators to define an object as numerical and assign it to the numerical value for example

	OBJECT	OPERATOR	VALUE
IF	'age of the customer'	<	18
AND	'cash withdrawal'	>	1000
THEN	'signature of the parent'	is	required

Rules can represent relations, recommendations, directives, strategies and heuristics:

2.20 Introduction to Knowledge Representation

We are going to discuss the following topics

- The Defining Knowledge Representation
- Knowledge Representation Schemes
- Semantic Networks
- Taxonomy
- Ontology

i. Knowledge Representation

'A representation is a set of conventions about how to describe a class of things. A description makes use of the conventions of a representation to describe some particular thing.' (Winston 1992:16).

'Good representations make important objects and relations explicit, expose natural constraints, and bring objects and relations together' (ibid: 45)

- The representation principle:

Once a problem is described using an appropriate representation, the problem is almost solved.

ii. Knowledge Representation Schemes

A number of knowledge representation schemes (or formalisms) have been used to represent the knowledge of humans in a systematic manner. This knowledge is represented in a knowledge base such that it can be retrieved for solving problems. Amongst the well-established knowledge representation schemes are:

- a. Production Rules
- b. Semantic Networks
- c. Frames
- d. Conceptual Dependency Grammar
- e. Conceptual Graphs
- f. Ontology
- g. Predicate and Modal Logic
- h. Conceptual or Terminological Logics
- i. XML / RDF

a. Semantic Networks

Ross Quillian (1966 and 1968) was among the early AI workers to develop a computational model which represented 'concepts' as **hierarchical** networks.

This model was amended with some additional psychological assumptions to characterise the structure of [human] semantic memory.

- Collins and Quillian (1969) proposed that:
 - Concepts can be represented as hierarchies of inter connected concept nodes (e.g. animal, bird, canary)
 - Any concept has a number of associated attributes at a given level (e.g. animal --> has skin; eats etc.)
 - Some concept nodes are **super ordinates** of other nodes (e.g. animal > bird) and some are **subordinates** (canary < bird)

For reasons of **cognitive economy**, subordinates **inherit** all the attributes of their superordinate concepts.

Some instances of a concept are **excepted** from the attributes that help [humans] to define the super ordinates (e.g. ostrich is excepted from flying)

Various [psychological] processes **search** these hierarchies for information about the concepts represented **reference to Appendix figure 6.2.**

Defining Inheritance

AI researchers have refined the notion of inheritance:

- It is called a specialised inferencing technique 'for representing properties of classes, exceptions to inherited properties, multiple super classes, and

structured concepts with specific relations among the structural elements' (Touretzky 1992:690).

A semantic network is a structure for representing knowledge as a pattern of interconnected **nodes** and **arcs**. Nodes in the net represent **concepts** of entities, attributes, events, values. Arcs in the network represent **relationships** that hold between the concepts.

A semantic network is a graph theoretic data structure whose nodes represent **word senses** and whose arcs express **semantic relationships** between these word senses.

Quillian gave an account, perhaps first used by a computer scientist, of the associate features of human memory that incorporated a spreading activation model of computation.

Networks and 'Meaning' Representation

The biosystematics notions of **taxonomies**, where the concept of super ordinates and subordinates plays a major role on the knowledge representation literature

TAXONOMY OF LIFE: The taxonomic organisation of species in a hierarchical structure:

- Kingdom > Phylum (division in botany) > Class > Order > Family > Genus > Species

Carolus Linneaus (c.18th century Swedish botanist) devised the system of binomial nomenclature used for naming species.

- Each species has a two-part Latin name, formed by appending a specific epithet to the genus name.
- The latter is capitalised and both parts italicised.

Representation and Biosystematics

Work in knowledge representation has been influenced by key notions in biosystematics.

However, there are crucial differences between what a taxonomist does and a knowledge engineer does.

The key difference is that of the intended audience in the two cases:

- for the taxonomist the audience is intelligent and human
- and for the knowledge engineer the primary 'audience' is a computer system, or more accurately the representation program

iii. Ontology

Definition:

“The science or study of being; that department of metaphysics, which relates to the being or essence of things, or to being in the abstract” (OED online, <http://www.oed.com/>)

AI experts, like Tom Gruber, suggest that:

'In the context of knowledge sharing, we use the term ontology to mean a *specification of a conceptualization*. That is, ontology is a description (like a formal specification of a program) of the concepts and relationships that can exist for an agent or a community of agents. This definition is consistent with the usage of ontology as set-of-concept-definitions, but more general. And it is certainly a different sense of the word than its use in philosophy.' (Cited from www-ksl.stanford.edu/kst/what-is-an-ontology.html; site visited 12/09/05)

Also Tom Gruber, suggest that:

'Ontology is an explicit specification of a conceptualization. The term is borrowed from philosophy, where Ontology is a systematic account of Existence. For AI systems, what "exists" is that which can be represented. When the knowledge of a domain is represented in a declarative formalism, the set of objects that can be represented is called the universe of discourse. This set of objects, and the describable relationships among them, are reflected in the representational vocabulary with which a knowledge-based program represents knowledge. Thus, in the context of AI, we can describe the ontology of a program by defining a set of representational terms. In such ontology, definitions associate the names of entities in the universe of discourse (e.g., classes, relations, functions, or other objects) with human-readable text describing what the names mean, and formal axioms that constrain the interpretation and well-formed use of these terms. Formally, ontology is the statement of a logical theory.' (Cited from 'www-ksl.stanford.edu/kst/what-is-an-ontology.html; site visited 12/09/05)

2.21 Ontology as a Specification Mechanism

A body of formally represented knowledge is based on a **conceptualization**: the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them (Genesereth & Nilsson, 1987).

A conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose.

Every knowledge base, knowledge-based system, or knowledge-level agent is committed to some conceptualization, explicitly or implicitly

2.22 Knowledge-based system methodology

Introduction

Knowledge-based systems development is a complex and expensive process that needs to be applied in an organized manner. Many approaches have been introduced for this purpose e.g. CommonKADS (B. Wielinga, 2001), Components of expertise, and Generic tasks, but theoretical approaches must be supported by practical guidelines in order to apply these methodologies in real life applications. The adopted methodology presented in this paper is based on a spiral model that guides the overall life cycle of knowledge-based development (Fig.1). According to this model, the development methodology consists of two main components: *Knowledge Engineering*, and *Software Engineering*. These two components are interacting with each other. In other words, they are not sequential in nature.

Some phases of the software engineering methodology may be applied before the completion of the knowledge engineering part and vice versa.

2.22.1 Knowledge engineering methodology

The adapted methodology include three main activities, that are directed in iterations as illustrated in (Fig.1), to produce successive versions of the expert system, starting from research prototype and ending by the production version. These activities are:

- Knowledge acquisition,
- Knowledge analysis & modeling, and
- Knowledge verification.

The rest of this section is a detailed description of the applied methodology for each of these activities.

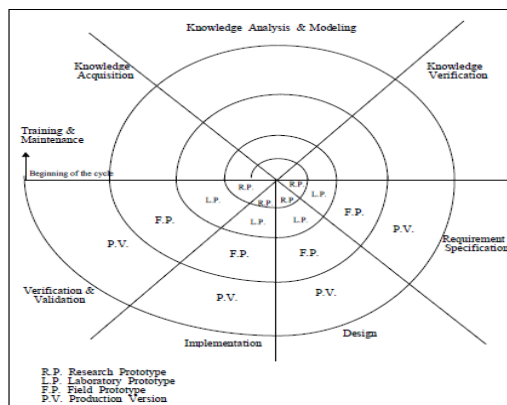


Figure 8: Spiral Model for Expert System Development

a. **Knowledge Acquisition**

Knowledge acquisition is considered the bottleneck of the expert system building process. One of the major difficulties at this stage is to explicitly identify and capture knowledge relevant to the intended application.

The developed models help in defining the set of domain models to be acquired from the domain expert, hence decrement unfruitful knowledge elicitation efforts, and direct the process in an organized manner. Accordingly, the task of a knowledge engineer is to select the appropriate model for the intended application, and determine the domain models required by each sub-task, and conduct knowledge elicitation sessions to acquire the required knowledge.

b. **Knowledge Modeling Steps**

We followed the *Select-and-Modify* approach for expertise modeling, where a complete generic model is selected from a set of predefined models, and subsequently modified to suit the needs of the intended application, giving a complete customized expertise model, after additional domain knowledge acquisition. This modeling approach distinguished KADS-I from other model driven approaches of the late 80's, like Generic tasks and Role Limiting Methods (B. Chaandrasekaran, 2010), where the generic models were hardwired into tools, and could not be modified.

The Select-and-modify approach is divided into the following four activities:

- 1- **Select-IM:** Select an interpretation model according to a set of selection criteria (task features according to CommonKADS terminology).
- 2- **Evaluate-IM:** Investigate, whether the selected IM is suitable for the application, or that it needs some modification. This activity is done by identifying the discrepancies between the required system behavior and that of the selected interpretation model. These discrepancies can be discovered either by walking through the IM or by trying hypothetical cases to evaluate the
- 3- **Modify-IM:** Modify the IM, to make it suitable for the intended application.
- 4- **Domain-KA:** Acquire the domain knowledge according to the selected and probably modified interpretation model.

c. **Knowledge Verification**

Knowledge verification is the stage whereby we make quality assurance of the acquired knowledge.

Actually there are two points of interest: review procedure, and multiple expert conflicts resolving procedure.

d. Review Procedure

Establishing a review procedure at the knowledge acquisition stage reduces the efforts to be done later in the verification and validation of the developed system. Knowledge is reviewed at the end of different phases: knowledge elicitation, knowledge analysis and modeling, and implementation.

Reviewing at the *elicitation stage* is conducted by letting the domain experts review the results of the knowledge elicitation sessions.

At the *analysis and modeling stage*, the domain experts review the filled forms describing the domain knowledge. Since task and inference knowledge are documented in KADS notation, which is hard to be understood by non-specialists, the knowledge engineer performs this activity by walking through them with the presence of the domain experts.

Reviewing at the *implementation stage* is conducted by letting the domain experts review any early prototype.

e. Multiple Experts Conflict Resolution

Multiple experts' conflict resolution is considered as a way of verifying the acquired knowledge.

Because when two experts give different knowledge for the same thing, then trying to resolve this conflict yields more reliable knowledge, hopefully, agreed upon by both of them. If no consensus is reached the expert who is recognized to be more specialized in the area of disagreement is considered.

2.22.2 Software engineering methodology

As a software, building a knowledge-based system entails doing software engineering activities that are accomplished in parallel with knowledge engineering activities that we have discussed in the previous section. These activities go through different stages, these stages are:

Requirements specification, design, implementation, and testing

As we mentioned before, these activities are done in successive iterations, each of which ends with the delivery of a new, more mature version of the expert system, until the production version is delivered.

The approach we are applying for software development is a combination of rapid prototyping, incremental, and traditional methods. Rapid prototyping is used first to reach an agreement on the initial set of the system requirements. A laboratory prototype then is incrementally developed and tested for maturity. The field prototype is then implemented by adding a better user interface, and more explanation facilities to the laboratory prototype, and then the field prototype is tested in real environment. Once the system is successfully field tested, it can be considered as the final requirements specification of the production version. As we reach a complete requirement specification, the production version can be developed using traditional software engineering methods.

As we have described the general broad lines in our software methodology, the rest of this section is a more elaborate specification of each software engineering activity.

i. Requirements Specification

As shown in **(Figure 8)**, the outcome of the early knowledge elicitation activity is an initial set of requirements specifications. This initial set is the basis for further knowledge acquisition efforts and the basis for the research prototype preliminary design. The requirements specification is revised once the research prototype is evolved. This version of the specification is the basis for developing the lab prototype. At the end of the lab prototype implementation and testing, the requirements specification document is revised again if necessary. This second revision is the basis for developing the field prototype. The final requirements specification document is then produced after the implementation and testing of the field prototype. This final document is the basis for the production version of the expert system to be developed.

ii. Design

A preliminary design is done just after the set of initial requirements specification is determined, and a preliminary model of knowledge layers is specified. This design is the basis for the research prototype which is used to produce the requirements specification for the lab prototype. Another cycle of the design is done after the lab prototype specifications are determined, and more elaboration on the knowledge model is conducted. This cycle is repeated after the implementation and testing of the lab prototype to produce the design of the field prototype. Once the field prototype is implemented and tested, the final design document of the production system is issued.

The following is a description of the main subjects to be considered in the design document.

iii. Knowledge Representation

The model of knowledge produced as an output of the knowledge engineering activities is used as a generic representation of knowledge in both the preliminary design and the lab prototype design documents. In the field prototype design document, a section is included describing how to map this generic design into the knowledge representation schemes supported by the tool to be used.

iv. Interfaces

The need for a robust and mature interfaces increase from one prototype to the next, in the research prototype, very simple interfaces are used (the interfaces provided by the shell). In the lab prototype, more enhanced user interfaces are developed. Interfaces to external packages and databases are included in the field prototype.

The same interfaces are used in the production version after adding any required enhancements.

v. Explanation Module

In the research prototype, explanation facilities depend on the capabilities provided by the shell, and are designed primarily for the developer to trace the behavior of the system. In the lab prototype, special explanation facilities concerning the why the system is asking a certain question and how the system has reached a certain conclusion are provided. Term explanation and detailed information concerning the conclusions are included in the field prototype. The production version uses the same explanation capabilities of the field prototype after making the required enhancements.

vi. Implementation

The first decision to be taken after the approval of the design is the selection of the implementation tool to be used. The criteria of selection depend on the facilities required by the system version being implemented. Expert system shells are suitable for research, and lab prototypes. Although expert system shells speed up the implementation process, customized tools provide more flexibility in implementation. For this reason, a general purpose knowledge representation object language

(KROL), has been developed, and guidelines for KADS implementation using this language were defined (J. Breuker, 2009) The research prototype implementation starts with the purpose of acquiring the user requirements specifications. The primary purpose of the lab Prototype is to test whether sufficient and appropriate expert knowledge has been obtained

and represented properly for solving the class of problems associated with the given application. The secondary purpose of the prototype is to provide a realistic test of the application's man-machine interface. The main purpose of developing the field prototype is to test the acceptance of the system by other professionals different from the experts participated in providing the domain knowledge. The prototyping process can be summarized as follows:

An initial stage, in which a research prototype is created from initial knowledge captured from experts. This prototype is a throw away prototype in which the overall skeletal frame of the system is built with one or more complete sections.

An interim stage: in which a more capable prototype is derived from the initial stage prototype by testing and reviewing the knowledge with experts and prospective users. A laboratory prototype is the output of this stage.

A final stage: in which a field prototype is produced from the laboratory prototype. The field prototype development starts once the laboratory prototype is matured. The field prototype is tested by the actual users for approval.

Once the prototyping stages are successfully terminated, the implementation of the production system starts using the field prototype as a valid specification.

vii. KNOWLEDGE BASE SYSTEM TESTING

Knowledge-based system testing, is the procedure by which we can be confident that the developed expert system is consistent, complete, correct, and satisfies the original requirements and needs of the user.

This procedure evolves through a cycle of three main steps, namely *Verification*, *Validation*, and *Evaluation*.

Verification, as defined by (Adrien et al. 2009), is the demonstration of consistency, completeness, and correctness of software et al. *Building the system right* system is functionally matching the proposed design, and free of semantic and syntactic errors.

Validation is the process whereby the system is tested to show that its performance matches the original requirements of the proposed system. It is defined by (Adrian et al. 2009) *Validation is the determination of the correctness of the final program or software produced from a development project with respect to the user needs and requirements et al., the right system*

Evaluation is the process whereby we ensure the usability, quality, and utility of the expert system (Cannaday, J. 2006). A complete testing cycle is performed in iterations through which, the expert system is updated and refined. The following is a detailed description of the different phases of testing procedure.

Vii -1 **Verification**

Verification process evolves through the following two main stages, during the development of the expert system:

1. Development stage

At this stage, the developers practice different functions of the implemented systems, looking for potential errors. This can be accomplished using different techniques. Generally, these techniques fall into two broad categories. Non Case-based techniques which include tracing, spying and other traditional debugging techniques, and Case-based verification techniques which are applied by spelled out in the requirements specification document.

2. Examination stage:

Before delivering the system to the user, the expert system is tested to make sure that it is running properly, by testing all the functions of the system trying to examine the performance of the system in different situations. The output of this stage is the **verification report** that documents the differences between system design and. This report serves two aspects. First, undocumented modifications applied to the system during development can be addressed. Second, further errors in implementation can be discovered.

As a direct consequence of this report, the design document is updated to reflect undocumented modifications in the implementation, and implementation can be revised to match design specifications.

Vii- 2 **Validation**

The validation process is performed by generating a set of test cases to be solved by both the expert system and the domain expert. Test cases are generated using the **White-box** testing method (Berners-Lee, T. Hendlers J., Lassila O. 2001) where they are selected according to different situations of the expert system.

The output of this stage is the **validation report**, which identifies the differences between domain experts' solutions, and the expert system results.

Vii - 3 Evaluations

The main goal of the evaluation step is to assess the quality, usability, and utility of the expert system from the point of view of human experts other than those domain experts who have participated in knowledge acquisition phase, and from the point of view of the target users.

The basic idea of the adopted technique is to evaluate the behavior of the Expert system, against that of human experts, by generating a collection of carefully selected test cases, and let a number of human experts in the domain - as well as the expert system - solve these test cases. Another human expert evaluates the generated solutions, and ranks them according to their grades. Later on, an open discussion is held to let human experts justify their solutions. According to this discussion, evaluation of solutions may change, and the final ranking of solutions is reached. If the expert system is far from precedence, the knowledgebase must be updated. The following is a detailed, step by step, evaluation process:

Prepare case description forms

As a primary step in the evaluation process, forms are designed for test cases. These forms vary according to the kind of knowledge to be tested.

Prepare comparison criteria

An evaluation criterion, or a formula, is designed, to enable a formal judgment on solutions generated by human experts, and the expert system. The selected criteria provides both quantitative and qualitative evaluation basis for judgment. The following is an example for qualitative and quantitative evaluation criteria:

Grade	Abbr.	Points
Excellent	E	3
Good	G	2
Acceptable	A	1
Unacceptable	U	0

Table 2: A qualitative & quantitative evaluation criteria of an expert system

The formula is given as

$$P_i = 3*NE_i + 2*NG_i + 1*NA_i + 0*NU_i$$

Where:

P_i the performance score for expert # i

NE_i number of cases evaluated as excellent

NG_i number of cases evaluated as good

NA_i number of cases evaluated as acceptable

NU_i number of cases evaluated as unacceptable

Generate test cases

Test cases are prepared manually by knowledge engineers. The most important criteria of these test cases is that it covers both normal cases, as well as the most difficult, and rare cases.

Solving test cases

A copy of the selected test cases is given to three or four domain experts. The same cases are introduced to the expert system. Each of the domain experts as well as the Expert system works out the test cases independently

Evaluation of test cases

Solutions of test cases are evaluated in a blind manner, so that distinguishing between solutions of the expert system and solutions of domain experts becomes impossible. One or two domain experts, other than those who gave the knowledge acquired by the expert system - and of course other than those who solved the test cases - are given test cases solutions for evaluation according to the previously prepared formula.

The result

A score is given to each solution, and solutions are ranked according to these scores.

Observations and remarks

A meeting is held to discuss solutions. The domain expert who gave the knowledge acquired by the expert system, domain experts who solved the test cases, evaluators, and the knowledge engineer attend this meeting to analyze solutions and reach the final conclusion about the behavior of the expert system.

Updating knowledge and implementation

According to the conclusions reached in the previous step, the knowledge-base and implementation of the Expert system must be updated, so the system becomes more robust and valid.

Documentation

A detailed evaluation report is prepared at the end of evaluation process.

KNOWLEDGE-BASE SYSTEM MAINTENANCE

System maintenance is one of the most important activities to be considered in the methodology.

There are mainly two objectives of system maintenance. The first is to discover bugs, and problems that may arise during the actual, at site running of the system. The second is to make sure that the system is up to date, and possessing the most accurate and the most recent knowledge concerning the domain of application.

To fulfill the first objective, forms are designed and distributed to all sites, where users of the system can denote their remarks, complaints, and problems. These forms are collected periodically, and discussed with domain experts if it is concerning the knowledge. Modifications to the knowledgebase are documented and attached to the design document; accordingly, these modifications may reflect some changes to the implementation. Otherwise, if the problem is concerning implementation, the required modifications are made to the development version of the system, and all modifications are documented and attached to the user manual of the next version of the system.

The second objective is achieved by the arrangement for periodical meetings with domain experts, where domain knowledge is reviewed, and the latest updates in the field are discussed, and the required knowledge are acquired and augmented into the knowledge-base.

2.23 Turban's account of the knowledge base system development lifecycle

Turban (1990) pointed out an knowledge base system may be suitable for the following types of problems that are cognitive, well understood and defined with narrow domain, with data available, intended for training purpose, aiming to improve performance or quality related issues and not too complicated or time consuming The adopted methodology presented in this paper is based on this model that guides the overall life cycle of knowledge-based systems development **Figure 9**.

He also summarized the methodology of building an expert system into the following steps:

1. Problem identification and justification
2. Appropriateness, requirement fulfillment and availability of knowledge & experts
3. Conceptual design, planning and feasibility study
4. Software and hardware selection
5. Knowledge acquisition (system design and construction)
6. Knowledge representation
7. Testing (case study identification, field testing)
8. Implementation
9. Maintenance and update

10. Evaluation

Steps 5 through 7 loops in a cycle called “Prototyping”. An important characteristic of the development of an expert system is that they can be quickly prototyped and expanded. All the steps are standard, regardless of the nature of the system built; nevertheless, the content on each step varies accordingly to it.

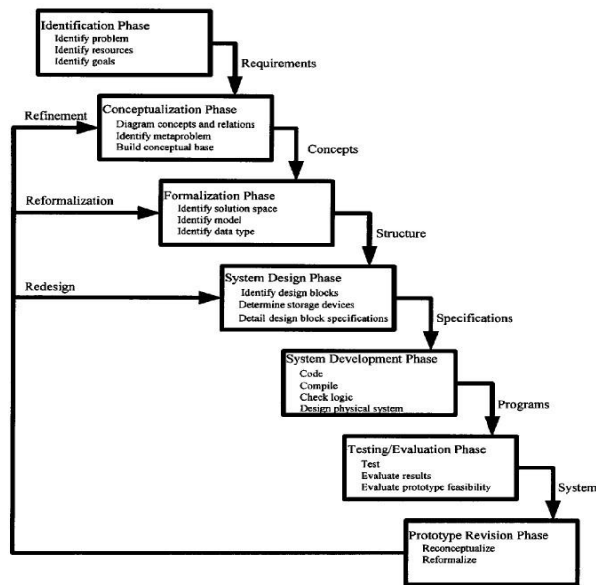


Figure 9: The steps in a typical knowledge-based systems analysis and design methodology

a. Identification phase

The first step in the identification phase, *Identify problem*, is similar to the problem definition phase in the traditional systems development life cycle. The objective is to identify, characterize, and define the problems the system will be expected to solve and then partition the problem into appropriate sub-tasks.

Once the problem is defined, the resources necessary for acquiring knowledge, implementing the system, and testing the system are identified. Typical resources include knowledge, time, computing facilities, and money. Because knowledge base systems are expensive and creating one takes considerable time, a feasibility study is often conducted before work progresses beyond this point.

In addition to identifying resources, the expert system analysts and/or designers also identify the system’s goals and objectives. It is helpful to identify and explicitly document the goals because certain design approaches, such as heuristic search, breadth search, depth search, and reasoning are goal-driven.

b. Conceptualization phase

The central task of the conceptualization phase is to diagram the system's key concepts and relations to define a conceptual base for a prototype system. Key objectives include separating the inference engine from the problem domain, factoring (analyzing) the problem into meta-problems, identifying the system's key concepts and relations, and testing those concepts and relations by challenging them (with specific examples of problem-solving activities) to ensure that they cover every general case. Many of the tools and techniques described in Part II are used in this phase.

c. Formalization Phase

The formalization phase involves mapping key concepts, sub-problems, and information flow characteristics isolated during conceptualization into more formal representations based on various knowledge engineering and problem solving tools and knowledge representation frameworks. The key objectives are to identify the solution space (a domain with a collection of all possible solutions), the hypothesis space (the hypothetical solution space), the underlying model, and the characteristics of the data.

To define the structure of the hypothesis space, the systems analysts or designers must formalize the concepts (knowledge in an abstract format that can be used to guide a searching or reasoning process) and determine how they are joined to form a hypothesis. The concepts provide clues about the nature of the space such as if it is finite, if a hierarchy must be considered, if certain levels of abstraction can be applied, and if a specific class of the concept must be generated. Such searching techniques as blind search, heuristic search, and abstracting the solution space are often used. Reasoning techniques such as assumption building, justification building, and the constraints and goal technique help to identify the underlying model of the process used to generate solutions in the domain.

d. System design phase

During the system design phase (sometimes called the logical design phase) the analyst and/or designer specifies how the system will meet the requirements identified during the previous three phases. Typically, the reports and other outputs the systems must produce are defined first. This phase is similar to the design stage in the traditional systems development life cycle. Note, however, that the representation schemes used to describe knowledge differ from traditional methodologies.

e. System development phase

A prototype knowledge-based system is created during the system development (or physical design) stage. This stage is similar to the development stage in the traditional system development life cycle.

f. Testing and evaluation phase

During this phase, the prototype system is evaluated. This phase parallels the testing stage in the traditional system development life cycle. However, in addition to the testing tools and techniques described in Part VII, knowledge-based utilize a dynamic testing technique to verify the reasoning and/or inference process.

g. Prototype revision phase

An expert system evolves over time, calling for almost constant revision, a trait knowledge-base share with most prototypes. Based on the results of the testing/evaluation phase, concepts and relations are refined, the solution space, the model, and the data characteristics are reformatized, and the system is redesigned.

h. Software selection

Different types of knowledge-base are widely used in different areas such as diagnosis, debugging and repair, interpretation, monitoring, control, design, planning, and instructions. The function of the expert system for this research is to accomplish the following:

Collect input data, arrange and pick questions according to user's answer

Store and distribute required input to other modules

Receive feedback from other modules and update database

Analyze users' preferences

Rank output based on users preferences

Give explanations of reasoning and guide users for helpful references

There are many free and/or commercially available products called expert system shell that can be used as an expert system building tool. To be more specific, a shell is a piece of software which contains the user interface, a format for declarative knowledge in the knowledge base, and an inference engine. There are a number of shell features which will be needed for the purpose of this research:

2.24 Advantages of knowledge-based system

- i. Knowledge-based systems seek to emulate or model a human expert's way of solving problems,
- ii. The expert system is highly specific to a given problem and cannot be retrained easily.

2.25 Related work

Using intelligent computer systems technology to support the academic advising process offers many advantages over the traditional student advising. The expert system is capable of advising students using prescriptive advising model and developmental advising model. Academic advising cases tested using the system showed high matching (93%) between the automated advising provided by the expert system and the advising performed by human advisors. This proves that the developed prototype expert system is successful and promising according (M. Ayman Al Ahmar).

Literature reveals many endeavors in the field of automating academic advising activities including the application of expert systems . There can never be a 'global' expert student advising system applicable to all academic institutions and departments because of the existence of academic regulations and expert advising knowledge and reasoning specific to each academic unit.

Another research was carried by (Deputy Dean, College of Information Technology Ajman University of Science and Technology (AUST) and United Arab Emirates (UAE)), is that regulations allow each student to register from three to six courses per semester. However, the accumulated advising experience and grade statistics related to the IS department show that students who can be advised to register six courses without difficulty are students with AGPA 3.00 or above (out of 4.00), whereas students with AGPA greater than 2.00 and less than 2.25 are better advised to register 4 courses only the next semester to give them a chance to increase their AGPA.

2.27 Architecture of Knowledge-based career choice assistant

The presented expert system is based on the n-tier model of the web applications. This model allows different components of the system to be built by different experts, specialized in their domain. **Figure 2** shows how the components of the system interact with each other. Each of these components can exist on the different machines or anywhere on the web. Knowledge base and inference engine are the two most important components of an expert system. The basic principal of the separation of the knowledge from its treatment is of prime importance in the building of every expert system. The building and maintenance of an expert system is greatly facilitated by trying to adhere to this principal as closely as possible.

The Knowledge Base Layer (KBL): The knowledgebase is built using facts and rules. It contains knowledge about Courses, universities, etc.

The Database Layer (DBL): This layer is implemented using MS SQL Server 2008 database. This contains the authorization information about users (students) KCSE results, courses and institutions' specific information.

The Reasoning Engine: The reasoning engine accepts user input queries and responses to questions through the I/O interface and uses this dynamic information together with the static knowledge stored in the knowledge base. The knowledge in the knowledge base is used to derive conclusions about the current case or situation as presented by the user's input. JESS is used here for this purpose.

Server Side Application Layer (SSAL): Application layer is built using Java Server Pages (JSP). The JSP provides the web developers with a framework to create dynamic content on the server using HTML, XML, Java classes, which is secure, fast and independent of server platform.

Client Side Interface Layer (CSIL): It will be implemented using Hyper Text Markup language (HTML), CSS and JavaScript. The CSIL consists of forms for accepting information from the user and validation those forms using JavaScript. It also provides the explanatory interface to the users of knowledge-based systems system.

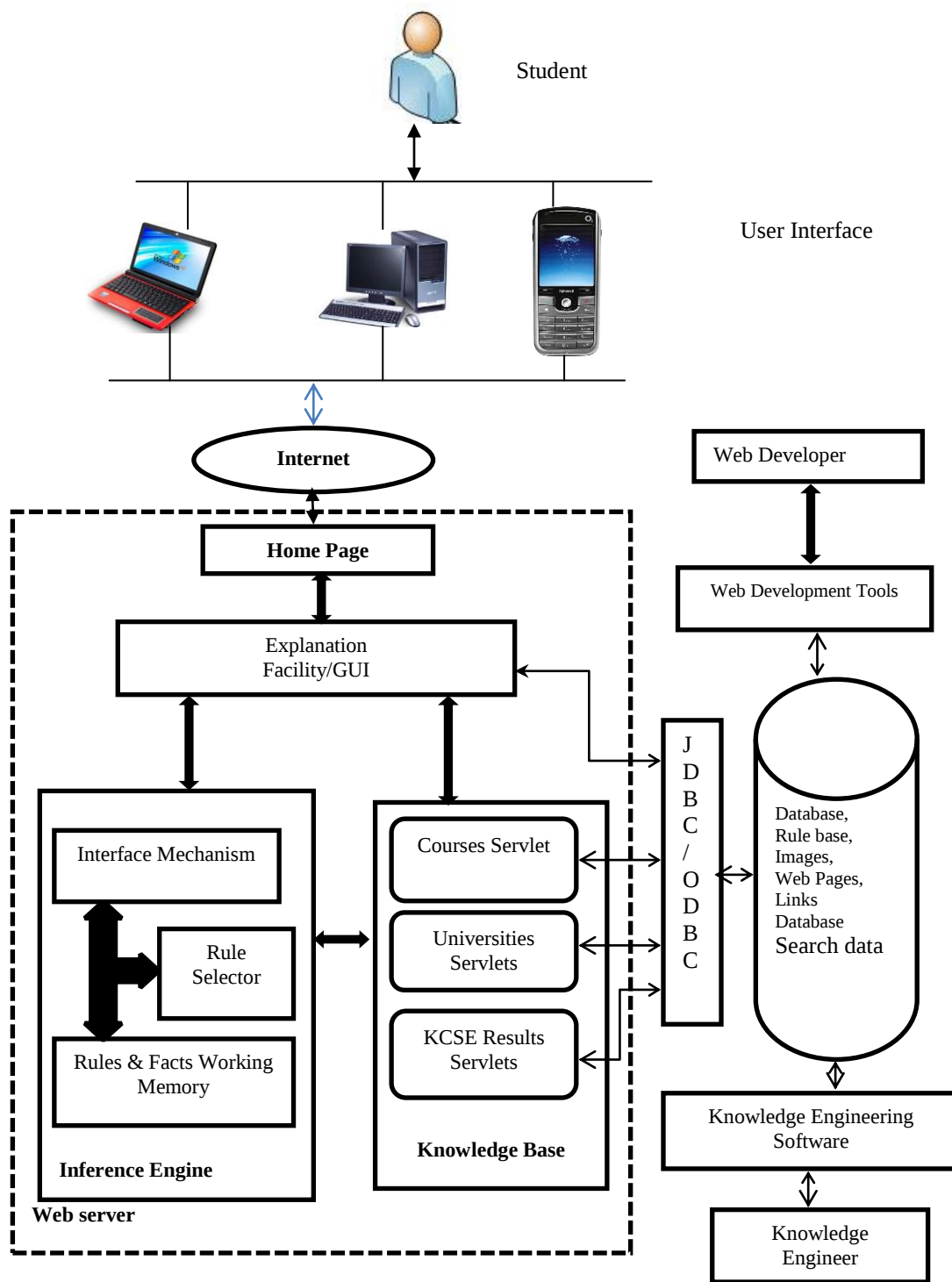


Figure 10: An architecture of Career Choice Assistant

2.28 Knowledge based career choice assistant features

There are a number of features commonly used in knowledge-based also which shall apply to Knowledge base career choice locator & advisory System features and they are:

1. Coping with uncertainty - the ability of the system to reason with rules and data which are not precisely known;
2. Data driven reasoning - an inference technique which uses IF THEN rules to deduce a problem solution from initial data; advisory system fits this model, since the aim of the system is to pick the correct diagnosis. The knowledge is structured in rules which describe how each of the possibilities might be selected. The rule breaks the problem into sub-problems. The system would try all the rules till it finds a perfect match which is then returned to the user through a user interface;
3. Data representation - the way the problem specific data is stored and accessed in the system;
4. User interface - that portion of the code which creates an easy to use system;
5. Explanations - the ability of the system to explain the reasoning process that it used to reach a recommendation.

2.29 Chapter Summary

Chapter two begins with the definitions of career choice, factors in career choice and education and career aspirations, the proposed application of AI systems available for student's selection process in Kenya and their limitation is highlighted.

Secondly Expert system is discussed based on its definitions and brief history, applications types of knowledge-base, characteristics and comparisons with human experts its methodology, advantages and lastly related research work

CHAPTER THREE: METHODOLOGY

3.1 Introduction

Research methodology is a way to systematically solve the research problem. It may be understood as a science of studying how research is done scientifically (Kothari C. R, 2004). In this chapter we discuss the methods, tools and instruments based on the previously described expert methodology that we used to achieve the objectives that were set in chapter 1. The methodologies discussed are directly matched with the major tasks that we carried out in order solve the problem. Therefore it's worthwhile briefly mentioning the tasks. The main tasks of this research are:

1. Determine the systems requirements.
2. Develop a conceptual model of knowledge-based Career Choice assistant that integrates Knowledge from necessary knowledge sources and recommends suitable college courses in a specific college -This was achieved as detailed in the previous chapter
3. Design a prototype to demonstrate some of the functionalities of the model.
4. Implement the model using JESS plug-ins on eclipse IDE
5. Test the model with JESS Plug- ins using eclipse IDE
6. Analyze and Evaluate the model using the KCSE results in making the suitable career choice for the student(s)

Task 3,4 and 5 requires selection of a software development methodology and appropriate programming tools whereas task 6 requires a suitable analysis & evaluation method in view of the nature of the system being developed- career choice locator and advisory knowledge-based system.

In this project, we largely followed the prototyping methodology to design, build and test the career choice locator and advisory knowledge-based system model. The choice of the model was dedicated by various factors. The project was bound to be completed within seven months and system development was just a part. Also, development and testing a fully work solution was impractical since it would call for acquisition of real (actual) results from (KNEC databases) and various course from different Universities with their clustering requirements. Thus by prototyping, we could easily demonstrate only those parts that were relevant to our research using available software tools.

3.1 Differences between System Development Life Cycle (SDLC) and Expert/Knowledge-based System Development Life Cycle [E/KB SDLC]

The conventional system life cycle is process-driven and documentation-oriented, with emphasis on the flow of the data and the resulting system. It fosters the “specify then build” approach. The E/KB SDLC is result-oriented. The emphasis is on a “start slow and grow” incremental process.

Conventional system development does not support rapid prototyping or advanced languages because it follows a set sequence of steps. The E/KB SDLC utilized rapid prototyping, incorporating changes on the spot, which augments and refines the system until it is ready for use (Awad, Elias M., 1996, p97).

The prototyping, knowledge acquisition, and knowledge presentation steps are unique in E/KB SDLC. Prototyping is a process that enables the developer to create a model of the software that must be built (Pressman, Roger S., 1992, p27). By nature, prototyping involves a number of iterations, through which the prototype is gradually altered and expanded to meet organizational needs (Zahedi, Fatemeh, 1992, p227). Knowledge acquisition involves elicitation, analysis, and interpretation of the knowledge that a human expert uses to solve a particular problem (Awad, Elias M., 1996, p112). Knowledge representation is formalism for representing facts or rules in the computer about a subject or a specialty (Turban, Efraim, 1998, p862).

3.1 Expert /Knowledge-based System Development Life Cycle (E/KB SDLC)

A knowledge-based system is basically computer software, so its development follows a software development process. The goal of such a process is to maximize the probability of developing viable, sustainable software within cost limitations and on schedule. The main functions of a model of this process are to determine the order of the steps (or tasks) involved in the software development and to establish the transition criteria for progressing from one stage to another (Turban, Efraim, 1998, p604).

3.2 Steps in Developing Knowledge-based System

Steps of Knowledge-based System Development Life Cycle have yet standardized, therefore there are several approaches have been suggested for building Knowledge-based System. The methodology is mainly an improvement from conventional system development life cycle. Although the conventional system development and the expert system development differ in some important ways, the two approaches are fundamentally similar. While the conventional

approach may still be used in developing knowledge-base, iterative design, prototyping, early testing, and other variations for building conventional system are replacing the waterfall steps. As a summary Elias M. Awad introduced an Expert System Development Life Cycle an integration of knowledge engineering, Software engineering and Turbans' methodologies (Awad, Elias M., 1996, p99)

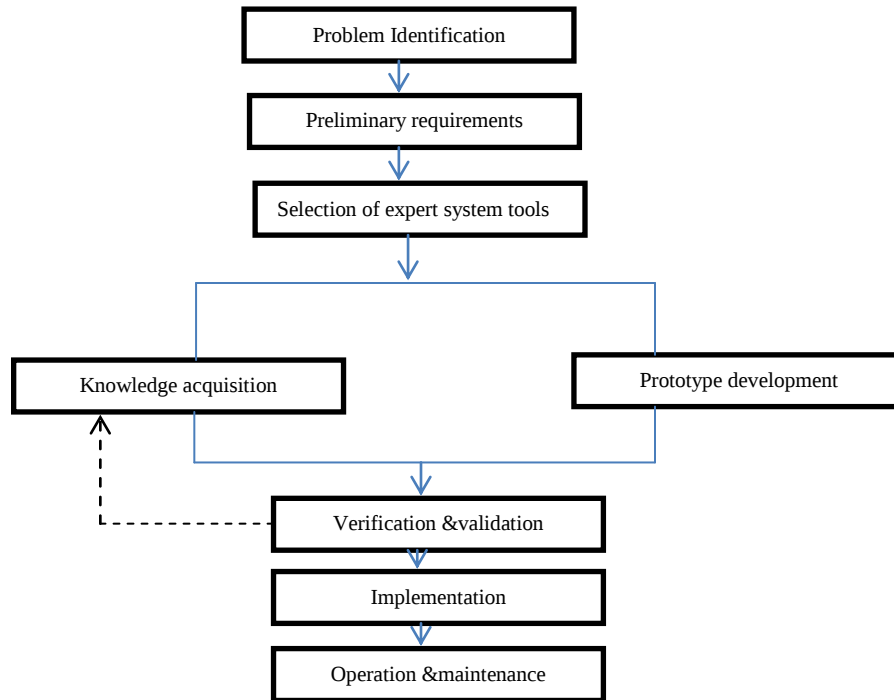


Figure 11: Knowledge-based System Development Life Cycle.

3.2.1 Preliminary requirements

The outcome of the early knowledge elicitation activity is an initial set of requirements specifications. This initial set is the basis for further knowledge acquisition efforts and the basis for the research prototype preliminary design. The requirements specification is revised once the research prototype is evolved. This version of the specification is the basis for developing the lab prototype. At the end of the lab prototype implementation and testing, the requirements specification document is revised again if necessary. This second revision is the basis for developing the field prototype. The final requirements specification document is then produced after the implementation and testing of the field prototype. This final document is the basis for the production version of the expert system to be developed.

3.2.2 Selection of expert system tools

Knowledge-base can be designed using a programming language or an expert shell.

Programming languages used for the expert system design are Lisp, Prolog, C and Pascal, and nowadays C++, Java, Visual Basic.

Expert shells are tools that enable people that do not have a good knowledge of programming to design knowledge-base. They are developed by eliminating knowledge bases from knowledge-base used for specific problems or fields, creating the so called “empty” knowledge-base – expert shells. Embedding new knowledge base in these expert shells using the old principles for inference and explanation, the new expert system can be designed.

Nowadays there is a wide range of different expert shells on the market that can include different types of knowledge-base (rule based, rule inducing, case based knowledge-base and/or neural networks). Some of them are free such as: BABYLON, ES, GEST, CLIPS, RT-Expert for DOS Personal Edition, while the others are for sale: Aion Development System (ADS) ART Enterprise, Doctus KBS, Expert choice, Expert edge, EXSYS, Insight 2+, KEE, Knowledge kraft, M.4, Nexpert Object, OPS83, Personal Consultant, Personal Expert, RT-Expert, XpertRule and JESS (CLIPS) engine.

Hence for the designing of career choice locator & advisory system we design using JESS and ECLIPSE IDE

Database

The research will require a database to store raw and processed data. The database that will be used is mySQLserver database. This is a good database for testing but during implementation Oracle or Sybase database will be recommend. Oracle and Sybase can take large number of data as compared to small databases.

3.3 Design

System Design

A formal model of the proposed system was built using Unified Modeling Language (UML).

(i) Use Case diagram of the Proposed System

3.3.1 The User Interface

This is where the user enters the query. This has been chosen to be a Java applet within a Web page. It is a lean and simple interface, very suitable for the purpose of a simple Web-based expert system.

Knowledge acquisition

When eliciting knowledge from the DE, use the KADS scheme which organises knowledge into several layers:

- Facts about entities in the domain course servlets, Universities servlets , and their relationships, are found in the bottom layer.
- Strategies for finding pieces of information (or other small-scale tasks) are found in the layer above.
- Strategies for performing larger tasks, such as doing a diagnosis, are found in the layer above

3.3.2 The Inference Engine

This is implementation using the JESS (Web CLIPS) engine.

3.3.3 Knowledge representation (The Knowledge Base)

The model of knowledge produced as an output of the knowledge engineering activities is used as a generic representation of knowledge in both the preliminary design and the lab prototype design documents. In the field prototype design document, a section is included describing how to map this generic design into the knowledge representation schemes supported by the tool to be used

This is the rules and facts repository, containing:

- The 'fixed' facts, such as the Mean grade, grades for different subjects, institutions and modules information;
- The rules - used to infer new knowledge from the fixed and asserted facts.

3.3.4 Knowledge acquisition

We are going to use KADS methodology

KADS is:

- A methodology for managing knowledge engineering projects
- A methodology for performing knowledge elicitation.

KADS scheme will be used to organise knowledge into several layers:

- Facts about entities in the domain- career servlets, university servlets , and their relationships, are found in the bottom layer.
- Strategies for finding pieces of information (or other small-scale tasks) are found in the layer above.
- Strategies for performing larger tasks, such as doing advice on career courses, institutions, are found in the layer above.

3.3.5 Prototype development

Turban leaves open the options that the prototype which features in phase 3 might be an evolutionary prototype or a throwaway prototype.

- In the 1st case, phase 4 would consist mainly of expanding this prototype, by adding more and more knowledge, until it became the knowledge base for the finished system.
- In the 2nd case, it would consist of drawing lessons from building the prototype and using these to assist in building the knowledge base from scratch, using a more appropriate tool.

3.3.4 Choosing the right implementation

The model describes the solution in terms of classes and actors, that is, methods. The knowledge-base and implementation of the Expert system must be updated, so the system becomes more robust and valid.

3.4 Testing (case study identification, field testing)

Testing is whereby the career choice locator & advisory system is verified and validated according to the requirements of the system specification. It is one of the phases that occur before and after coding. This is an ongoing process during the software development lifecycle.

During testing, UML diagrams will be so helpful in determining the validity of data that is obtained or inputted into the system. UML helps in ensuring that right processes are well tested and that there is a flow.

UML uses case diagrams to describe the functionality of a system

Use cases. A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse.

- **Actors.** An actor is a person, organization, or external system that plays a role in one or more interactions with your system. Actors are drawn as stick figures.
- **Associations.** Associations between actors and use cases are indicated in use case diagrams by solid lines. An association exists whenever an actor is involved with an interaction described by a use case. Associations are modeled as lines connecting use cases and actors to one another, with an optional arrowhead on one end of the line. The arrowhead is often used to indicating the direction of the initial invocation of the

relationship or to indicate the primary actor within the use case. The arrowheads are typically confused with data flow and as a result I avoid their use.

- **System boundary boxes (optional).** You can draw a rectangle around the use cases, called the system boundary box, to indicate the scope of your system. Anything within the box represents functionality that is in scope and anything outside the box is not.
- **Packages (optional).** Packages are UML constructs that enable you to organize model elements (such as use cases) into groups. Packages are depicted as file folders and can be used on any of the UML diagrams, including both use case diagrams and class diagrams. I use packages only when my diagrams become unwieldy, which generally implies they cannot be printed on a single page, to organize a large diagram into smaller ones (UML Use Case Diagrams, 2012).

Why use UML.

Plan

UML provides a good way of planning the system to be developed. Software without proper structures may result into software that does not meet the intended functionality or user requirements.

3.4.1 Evaluation

Solutions of test cases are evaluated in a blind manner, so that distinguishing between solutions of the expert system and solutions of domain experts becomes impossible. One or two students based on random results (domain experts), other than those who gave the knowledge acquired by the expert system - and of course other than those who solved the test cases - are given test cases solutions for evaluation according to the previously prepared formula.

3.4.2 Documentation

A detailed evaluation report is prepared at the end of evaluation process.

3.5 Maintenance and update

System maintenance is one of the most important activities to be considered in the methodology.

There are mainly two objectives of system maintenance. The first is to discover bugs, and problems that may arise during the actual, at site running of the system. The second is to

make sure that the system is up to date, and possessing the most accurate and the most recent knowledge concerning the domain of application.

To fulfill the first objective, forms are designed and distributed to all sites, where users of the system can denote their remarks, complaints, and problems. These forms are collected periodically, and discussed with domain experts if it is concerning the knowledge. Modifications to the knowledgebase are documented and attached to the design document; accordingly, these modifications may reflect some changes to the implementation. Otherwise, if the problem is concerning implementation, the required modifications are made to the development version of the system, and all modifications are documented and attached to the user manual of the next version of the system.

The second objective is achieved by the arrangement for periodical meetings with domain experts, where domain knowledge is reviewed, and the latest updates in the field are discussed, and the required knowledge are acquired and augmented into the knowledge-base.

CHAPTER FOUR: SYSTEMS REQUIREMENTS AND ANALYSIS

4.1 Introduction

What Are Requirements?

"...Requirements definition is a careful assessment of the needs that a system is to fulfill...must say **why** a system is needed, based on current and foreseen conditions, which may be internal operations or an external market...must say **what** system features will serve and satisfy this context...must also say **how** the system is to be constructed..." [Ross77]

Requirements constitute a specification for the new system. They serve as a “contract” between customers and developers.

“Requirements describe the system with respect to its environment, NOT its inner workings!”

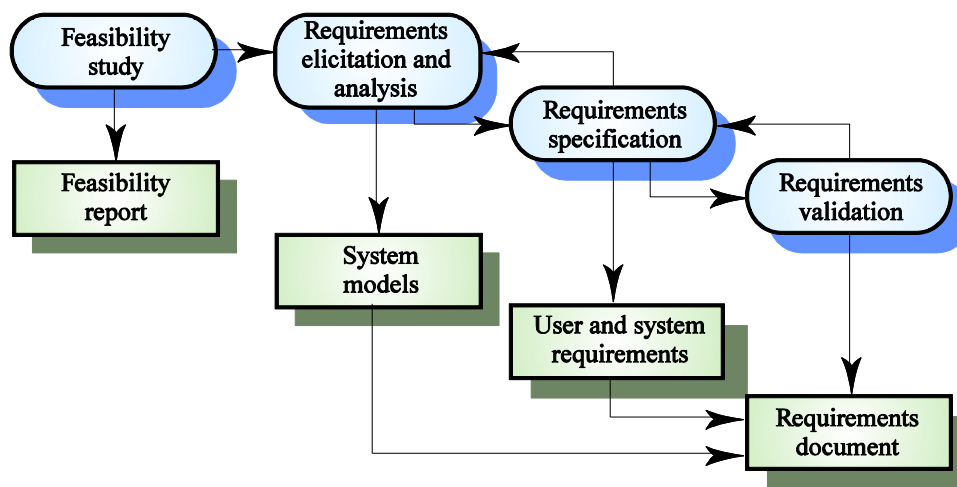


Figure 12: The requirements engineering process

4.2 Functional and Non- functional requirements

The proposed system will be based on a range of existing but enhanced concepts, which will make the course selection and education processes more flexible and effective for the form four graduate students.

In this chapter we shall describe the following

- a. Functional requirements
- b. Non-functional requirements

a. Functional requirements

1. Describe the **processing** (i.e., **functions** to be supported) by the proposed system.
2. Describe the **inputs** into the system
3. Also describe the **outputs** of the system.
4. Finally, they describe the **data that must be managed** by the system.

We are going to describe the **processing** (i.e., **functions** to be supported) by the proposed system, for these purpose we revisit The Objectives of the Study

- iv. Determine the systems requirements.
- v. Develop a model of expert based Career Choice Locator and Advisory System that integrates Knowledge from necessary knowledge sources and recommends suitable college courses in a specific college.
- vi. Design a prototype to demonstrate some of the functionalities of the model.
- vii. Implement the model using JESS plug-ins on eclipse IDE
- viii. Test the model with JESS Plug- ins using eclipse IDE
- ix. Analyze and Evaluate the model using the KCSE results in making the suitable career choice for the student(s)

The main function of the proposed (knowledge base Career Choice locator & advisory) system is to improve the method of selecting the best suitable faculty/major for student planning to be enrolled in different higher institution of learning for module II Degree programmes, Diploma and certificate courses. The basic idea of our approach is designing a model that accept the students aggregate grade and grades for different subjects, and applying the module results to a rule-based expert system to determine the compatibility of those subjects) cluster with the available faculties/majors in a particular University in Kenya. The result is shown as a list of suggested faculties/majors that are most suitable with the student capabilities and abilities.

Institution of higher learning in Kenya have a common clustering of high school subjects for a particular course and for a student to qualify or get selected to pursue that career must meet the requirements based on clustering.

Cluster analysis or **clustering** is the task of grouping a set of objects (Subjects) in such a way that objects (Subjects) in the same group (called **cluster**) are more similar (in some sense or another) to each other than to those in other groups (clusters).

Course(s) in higher institution of higher learning is based on combinations (clustering) of different groupings of subjects.

High schools subjects are categorized or placed into groups e.g.; **GROUP I, GROUP II, GROUP III, GROUP IV.**

Where by:

GROUP I: COMPULSORY

ENG: ENGLISH

KIS: KISWAHILI

MAT: MATHEMATICS

GROUP II: SCIENCES

CHE CHEMISTRY

BIO BIOLOGY

PHY PHYSICS

GROUP III

HAG HISTORY AND GOVERNMENT

GEG GEOGRAPHY

CRE CHRISTIAN RELIGIOUS EDUCATION

IRE ISLAMIC RELIGIOUS EDUCATION

HRE HINDU RELIGIOUS EDUCATION

GROUP IV

HSC HOME SCIENCE

ARD ART AND DESIGN

AGR AGRICULTURE

WWK WOODWORK

MWK METALWORK

BLG BUILDING CONSTRUCTION

PMC POWER MECHANICS

ELE ELECTRICITY

DRD DRAWING AND DESIGN

AVT AVIATION TECHNOLOGY

CMP COMPUTER STUDIES

GROUP V

FRE FRENCH

GER GERMAN

ARB ARABIC

KSL KENYA SIGN LANGUAGE

MUS MUSIC

BST BUSINESS STUDIES

Universities have three major/ faculty namely:

1. Arts
2. Science
3. Engineering

FOR EXAMPLE:

UNIVERSITY OF NAIROBI

1. SCHOOL OF COMPUTING & INFORMATICS

ADMISSION REQUIREMENTS:

Eligibility for consideration for admission into the degree of Bachelor of Science in Computer Science at the School of Computing & Informatics shall be governed by the following minimum admission requirements or an equivalent qualification recognized by Senate:

KCSE: A holder of Kenya Certificate of Secondary Education (KCSE) with a minimum aggregate performance of C+.

In addition candidates must have obtained a minimum grade of C+ in four subjects as shown below:

- a. Alternative A: Mathematics, Physics, A 2nd Group II subject or Any from Group III, Any Group II or Group III or Group IV or Group V.
- b. Alternative B: Mathematics, Physical Sciences, A 2nd Group III subject or Any from Group III, Any Group II or Group III or Group IV or Group V.

M.G: C+ AND GROUP I && GROUP II || GROUP III || GROUP IV|| GROUP V

2. SCHOOL OF ENGINEERING

Bachelor of Science Degrees in Electrical and Electronic Engineering, Civil Engineering, Mechanical Engineering, Surveying and Environmental and Bio-systems Engineering

ADMISSION REQUIREMENTS

The following will be eligible for consideration for admission into any of the above degree programmes.

- i. KCSE (Kenya Certificate of Secondary Education) Applicants (or equivalent):
Minimum Mean Grade C+ and C+ in each of the four cluster subjects (Mathematics, Physics, Chemistry and Biology OR Geography OR a Group IV subject).

M.G: C+ AND GROUP I && GROUP II || GROUP III || GROUP IV

3. SCHOOL OF EDUCATION

Bachelor of Education (Special Education)

ADMISSION REQUIREMENTS:

Mean grade of C+ (plus) at KCSE or equivalent with at least C+ (plus) in one of the following subjects, English/Kiswahili, Maths/Biology and any Arts subjects or mean grade of C (plain)

Secondly we describe the *inputs* into the system

The proposed systems will accept the following as input

- i. KCSE result; aggregate grade, subject grades
- ii. Mock results

Thirdly we describe the *output* of the proposed system

- i. Course(s) the student is suitable/ appropriate to pursue
- ii. Institution(s) the course(s) is available.

Lastly the system will manage subjects cluster as fixed facts in knowledge base and rules for different courses.

CHAPTER FIVE: PROJECT DESIGN AND IMPLEMENTATION

5.1 Structure of the knowledge base career choice locator and advisory system

This is a *knowledge-based* system. The knowledge base contains *facts* and *rules* referring to the partial / complete cluster and outcomes of module/module II course(s) offered in different University in Kenya. Other supplementary information is provided, such as which institution the course is offered, parent subject, etc. More information may also be introduced later on as necessary (i.e. module clustering).

The main elements of the proposed system are (also refer **Figure 2**)

5.1.1 The User Interface

This is where the user enters the query. This has been chosen to be a Java applet within a Web page. Is it a lean and simple interface, very suitable for the purpose of a simple Web based Knowledge base system; it has been slightly modified to match it to the specific needs of this project.

5.1.2 The Inference Engine

This is provided by a Java implementation of the CLIPS engine. The CLIPS language uses forward-chaining and the *RETE fast pattern-matching algorithm*, originally designed by Charles R. Forgy, at the Carnegie-Mellon University. CLIPS uses *lists* to process information, very similar the LISP language and it is largely used by the Artificial Intelligence community.

The Java implementation of CLIPS is called JESS - The Java Expert System Shell, currently version 7.2. This provides the user with the power of the CLIPS language and the flexibility of the Java cross- platform concept. In its basic form, JESS is a byte-code set that provides a command prompt similar to CLIPS.

However, most of the data structures provided by Java are available in JESS, including the AWT (*Abstract Window Toolkit*), which makes it possible to construct graphical user interfaces to the JESS (CLIPS) engine.

5.1.3 The Knowledge Base

This is the rules and facts repository, containing:

1. The 'fixed' facts, such as the modules information; subject cluster ;
2. The run-time asserted facts - user's grades for different subjects
3. The rules - used to infer new knowledge from the fixed and asserted facts.

Similar to CLIPS, the knowledge base is contained in a .CLP file, in plain text format. Separating the fixed facts from the may be desirable for larger knowledge bases. The listing contains information on how to achieve this.

5.2 Choosing the right implementation

In implementing the knowledge-based system, several decisions had to be made. The available choices were:

5.2.1 Code from the ground up

This simply means programming the inference engine from scratch and thus “re- inventing the wheel”. While it may be suitable for certain applications, it was totally inappropriate for the limited time and resources available in this case.

5.2.2 A suitable knowledge base system language.

This option is quite applicable to this case. The language should already contain the *inference engine* and other extensions. Valid alternatives were:

- i. **Prolog:** a backward-chaining artificial intelligence language. Prolog comes in several flavours, the latest even being *Visual*, which provides possible integration with other Visual type (event, GUI driven) programming languages
- ii. **CLIPS:** the C Language Integrated Production System. Despite its name, CLIPS is very similar to LISP (the LISt Processing language), using the concept of *lists*.

5.2.3A web-enabled Knowledge base system language;

This is an even better option. However, the available tools start getting scarce once we venture in the web-enabled artificial intelligence programming languages. Options:

- i. **Prolog** with a web enabled interface. Several Prolog Java wrappers exist. Some of them are commercial - hence not free, very big in size (Mbytes) and with lots of options useless for this particular case. Others are free or shareware, but still in various development / beta testing stages;

5.2.4A web-enabled *shell* and a pre-made user interface.

This seems to be the ideal choice for this particular project. Although a programming language *clone* is not always identical with the original, the basic functions needed for the limited scope of this project should be implemented and sufficient. There has only been one valid option:

Java Expert System Shell (JESS 7.1) and a simple Graphical User Interface.

Since it has been established that a web-based system is the best choice for this application, option 4 has been chosen. The proposed system will run either as a Java application (byte code interpreted by the Java VM) or as an applet, downloaded and run by a *Java-enabled browser* on the host machine. The (faster to load) Java application implementation will be used for developing and debugging. The proposed system will finally be deployed for

general use as an applet within a Web page.

Figure 13 shows the preferred method of deployment. It is obvious that the 'bottleneck' here is the time necessary for the applet to download - currently around 10-12 sec., maybe longer depending on the number of 'hops' on the Internet link between the client and server. This is not a big concern for this particular case.

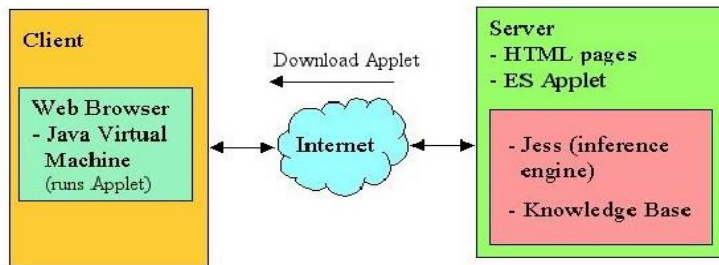


Figure 13: web based knowledge base system

Still A major bottleneck in using the existing prototype is the fact that the user's browser must download the knowledge base system applet, which then runs in its own 'sandbox' (i.e. with restricted rights) on the host machine, hence a different implementation is proposed, which will allow the expert system to run on the server and only communicate with the user for input and output. Once loaded, the applet runs at full speed though.

There is however a better way. Since the beginning of active HTML content, CGI scripts and JavaScript have made it possible to accomplish data / input processing on the server side. The trend is however shifting away from CGI (Perl) programming towards the *servlets* concept. Servlets are applets with no graphical interface, which run on the server side.

The two main possibilities with servlets are:

- i. User -program interaction occurs via HTML pages containing FORM requests and the like;
- ii. User-program interaction takes place via applet / servlet interaction: The applet runs on the host machine, and the servlet on the server. Of course, the applet is much reduced in size compared with the applet-only alternative - all it provides is a user interface to send user input and receive and display servlet output.

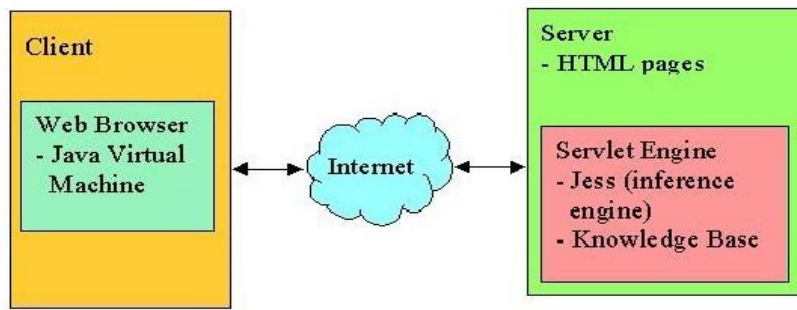


Figure 14: The servlet-based knowledge base system

5.3 The knowledge base System Model

So, what is the *structure* of the program after all? And *what* does it *do*, and *how* ? The best way to model the KBS system architecture is to use a specialized tool. One such tool is the Unified Modelling Language (UML). UML may be regarded as the successor of the Object Oriented Analysis and Design (OOAD) methods that proliferated during the *method wars* of the '80s and early '90s. UML represents the unification of the three main modelling language methods within the industry: Booch, Rumbaugh and Jacobson.

5.3.1 Use case diagram

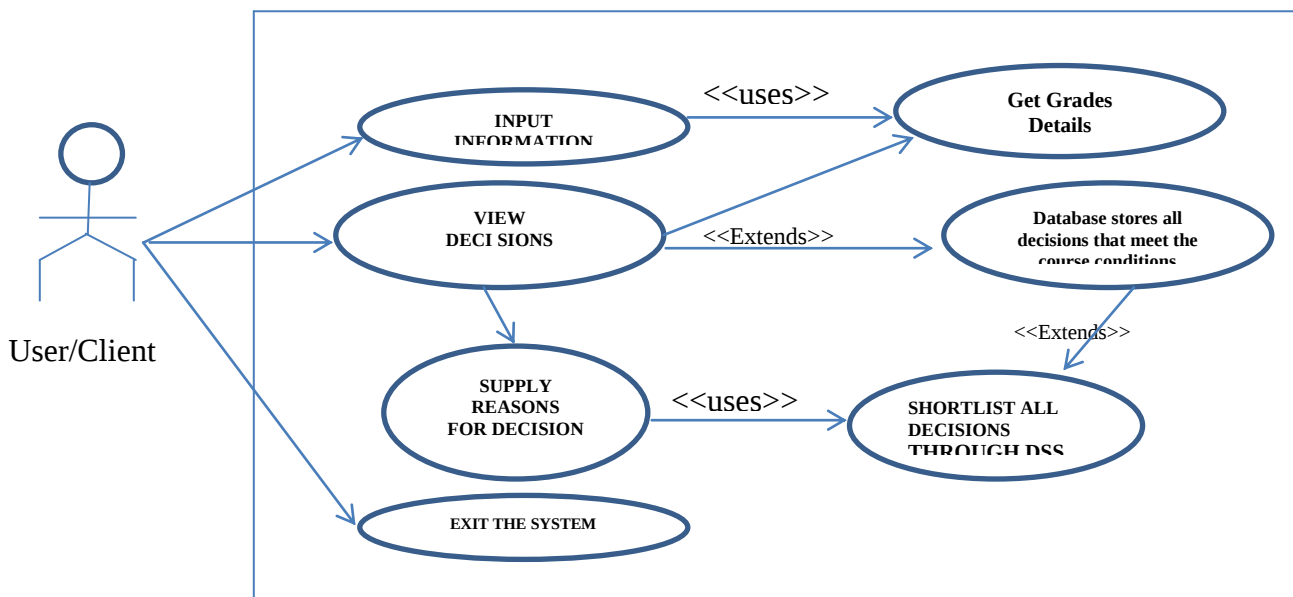


Figure 15: Use Case Diagram of the KBS System

Case diagram graphically depicts the interactions between the system, the external system (if any) and the user. Use case diagrams play a major role in system design because it acts as a roadmap in constructing the structure of the system; it also defines who will use the system and in what way the user expects to interact with the system.

The purpose of the use case diagram is to portray:

- The actor.
- A set of use cases for a system.
- The relations between the actor and the use cases.

Here, we introduce three main Use cases which extend, include or use other Use cases.

- Input Information;
- View Decisions;
- Exit System.

The User (actor): This is one of the clients that make use of the application.

Input Information: this represents the interface where the users are going to feed data into the system based on questions about their environment. The system then responds based on the correlation between user data and its foreknown intelligence. This uses another Use Case called Get Environmental Details and that is the set of questions representing the environment.

View Decisions: this is an avenue that enables the user of the system to view the system response. It's usually through an interface. All system possible decisions have been stored in a database external to the system and this is for code efficiency. It has a Use Case that is used by the decision taking Use Case.

Exit System: the user of the system can decide when to leave the application

5.3.1 Unified Modeling Language

The *class diagrams* are defined at the *model* level - meaning that they represent a generic model of the expert system (e.g. a 'Knowledge-based Expert System'), while the *object diagrams* are defined at the *user model* level, which allows them to represent a *particular* system (eg. "The Career Advisor and Locator").

The object diagram for the studied knowledgebase system as shown in **Figure below** and it has been derived from a generic class diagram. *Stereotypes* (the items enclosed in guillemots << >>) have been used to explain the classes the objects have been derived from (to avoid showing the class diagram as well). The boxes represent *objects*, and the lines connecting them represent *associations*. The filled diamonds () define a *composition*, that is, a relationship where the components may only belong to one whole at a time and where the components may not exist outside the relationship. The arrows () show the *navigation* directions, with explanation of the *associations*.

The diagram shown in **Figure below** is self-explanatory (this is the actual purpose of the

UML!). The user-expert system interaction occurs through the Applet Window which sends the problem conditions (answers to questions) to the Java Applet, which holds all the temporary data. The inference engine uses the knowledgebase for the rules and fixed facts and the Applet for the dynamic facts (asserted at run-time). The knowledge base contains the fixed facts and the rules. Finally, the solution is delivered to the applet window which forwards it to the user.

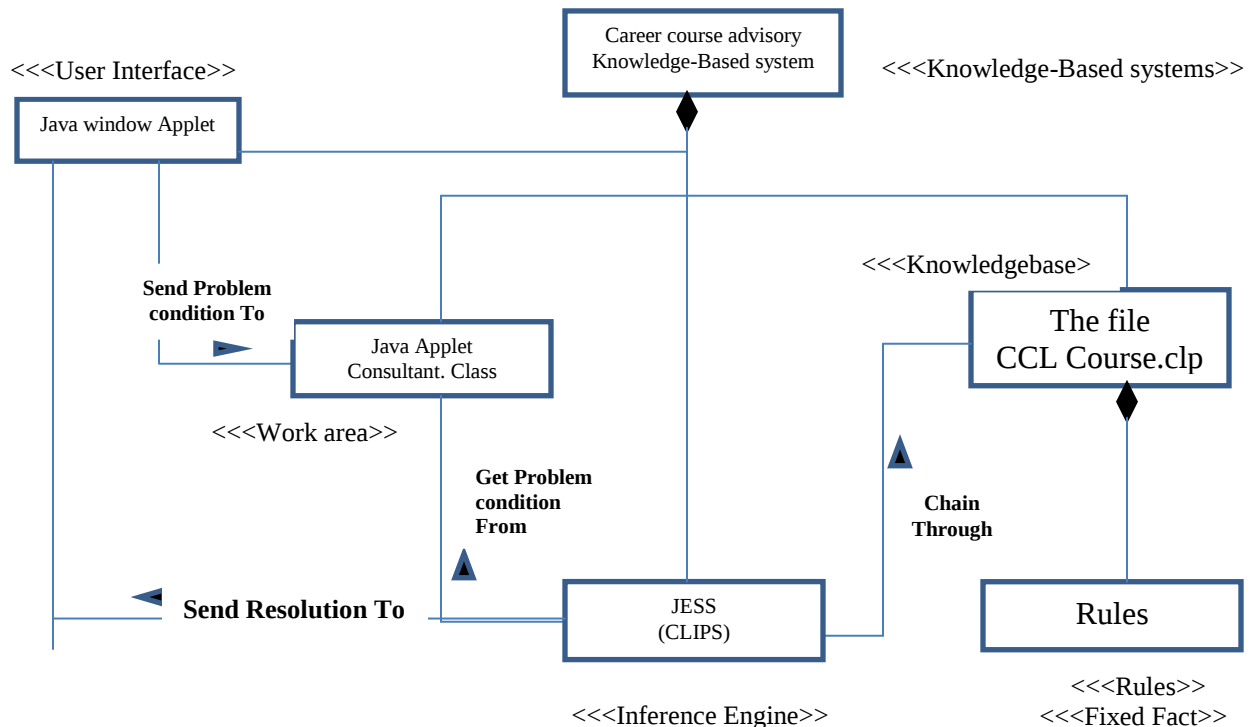


Figure 16: UML knowledge- based Career choice Locator & Advisory system

5.4 The Behavioral Model

After modeling the static aspect of the expert system, we would like to analyze the *interaction in time* of the expert system components

5.4.1 Sequence diagram for the Proposed System

A Sequence diagram is a graphical visualization of sequences of messages between objects i.e. sequence of method invocation of objects which results in accomplishing some tasks. The emphasis in a sequence diagram is on the sequence of messages. A Sequence diagram is a structured representation of behavior as a series of sequential steps over time. It is used to depict work flow, message passing and how elements in general cooperate over time to achieve a result. The sequence diagram for this system is shown in the next section.

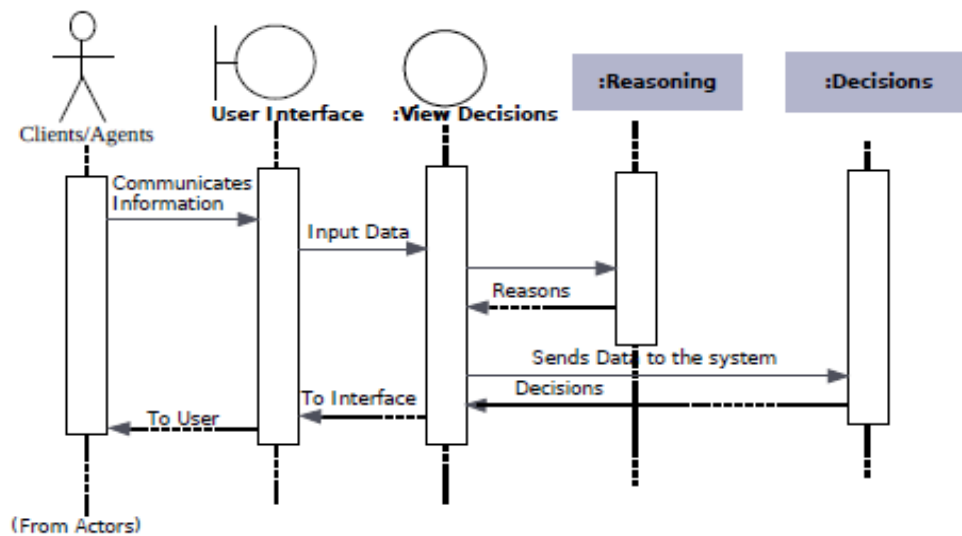


Figure 17: Sequence diagram of the knowledge-based system

5.4.2 Activity diagram for the Proposed System

Activity diagrams graphically show represent the performance of actions or sub activities and the transaction that are triggered by the completion of the actions or sub actions. It is a means of describing the workflow of activities.

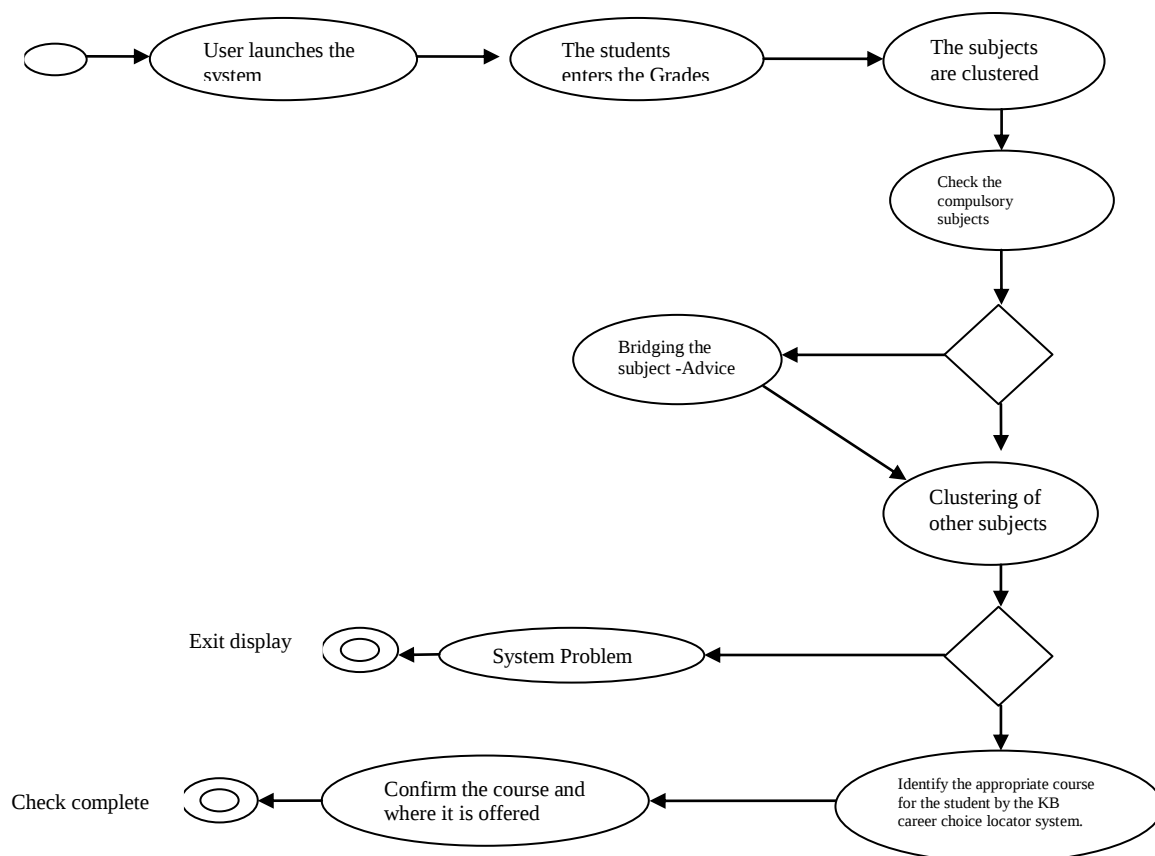


Figure 18: Activity diagram of the KB System

Our main sources for knowledge KCSE results student(s) attain at form four which include the mean grade, and subject combination which is a requirement for particular course in Kenyan institutions of higher learning for one to qualify to pursue that career. We collect the entire required criterion for course e.g. engineering, computer science and Education etc. for each school in some of the Universities in Kenya. We converted this results and course requirements into fact and rules in JESS syntax, and store them in the knowledge base of the JESS language. JESS is a suitable for forward reasoning and can be used easily to build the rules and facts, so that is why we have selected the JESS language. Furthermore, we captured all KCSE grades which include mean grade, and subjects' grades, information about the student in JESS fact templates. The following sample code shows a student structure for JESS template.

```

.....
,,,,,,,,,,,,,,,,,,,,,
(deftemplate student "Student Information"
(slot student-no)
(slot first-name)
(slot age)
(slot mean-grade)
(slot Mathematics-grade)
(slot Physics-grade )
(slot Chemistry-grade)
(slot Biology-grade)
(slot Geography-grade)
(slot Electricty-grade)
)
(deftemplate school-of-Engineering
(multislot data)
)

```

The rule is represented in the syntax of *“If conditions then actions”*. That means when the conditions are satisfied then the actions are carried out. Here is an example from our expert system using CLIPS syntax rules for the requirements of the faculty of Engineering

```

.....
,,,,,,,,,,,,,,,,,,,,,
(defrule fo-Engineering
(student (student-no ?student-no)
(mean-grade ?mean-grade)
(Mathematics-grade ?Mathematics-grade)
(Physics-grade ?Physics-grade)
(Chemistry-grade ?Chemistry-grade )
(Biology-grade ?Biology-grade )
(Geography-grade ?Geography-grade)
(Electricty-grade ?Electricty-grade)
)
)

```

```

(test (>= ?mean-grade C+))
=>
(assert (school-of-Engineering ))
(printout fdatao "[Engineering]" crlf
"Accepted=TRUE" crlf)

(if
(and (>= ? Mathematics-grade C+) (>= ? Physics-grade C+) (>=
? Chemistry-grade C+) (>= ? Biology-grade C+) (>= ? Geography-grade C+) (>= ?
Electricity-grade C+))
then
(printout fdatao "Recommended=TRUE" crlf
crlf)
else (printout fdatao "Recommended=FALSE"
crlf crlf))
)

```

This is the Engineering course function that facilitates the FIRE action once the march is found

```

.....
(defun Engineering ()
  "Career as the best suitable course"
  (printout t "Engineering as your best career" crlf))

```

By the time the student finishes the abilities test, our expert system have all information about the student capabilities and abilities captured. This information then is converted into facts in our CLIPS expert system. The following rule shows how the information is transformed into JESS facts:

```

.....
(defrule readdata
(declare (salience 1000))
(initial-fact)
?factstudent <- (initial-fact)

=>
(retract ?factstudent)
(open "data.txt" fdata "r")
(open "data-output.txt" fdatao "w")
(bind ?student-no (read fdata))
(bind ?first-name (read fdata))
(bind ?last-name (read fdata))
(bind ?mean-grade (read fdata))
(bind ?Mathematics-grade (read fdata))
(bind ?Physics-grade (read fdata))
(bind ?Chemistry-grade (read fdata))
(bind ?Biology-grade (read fdata))

```

```

(bind ?Geography-grade (read fdata))
(bind ?Electricity-grade (read fdata))
(printout fdatao "[STUINF]" crlf "No="
?student-no crlf "name= " ?first-name " " ?last-
name crlf "M.G=" ?mean-grade crlf crlf)
(assert (student
(student-no ?student-no)
(first-name ?first-name)
(mean-grade ?mean-grade)
(Mathematics-grade ?Mathematics-grade)
(Physics-grade ?Physics-grade)
(Chemistry-grade ?Chemistry-grade)
(Biology-grade ?Biology-grade)
(Geography-grade ?Geography-grade)
(Electricity-grade ?Electricity-grade)))
(close fdata)
)

```

Refer to appendix 6.2 for the results

CHAPTER SIX: EVALUATION

An evaluation criterion, or a formula, is designed, to enable a formal judgment on solutions generated by the user of the knowledge-based assistant and for this case the student(s), the designers of user interface evaluators, and the technical expert. The selected criteria provides both quantitative and qualitative evaluation basis for judgment. The table below provides a qualitative and quantitative evaluation criterion (**ref. table 2**):

The formula is given as

$$P_i = 3*NE_i + 2*NG_i + 1*NA_i + 0*NU_i$$

Where:

P_i the performance score for expert # i

NE_i number of cases evaluated as excellent

NG_i number of cases evaluated as good

NA_i number of cases evaluated as acceptable

NU_i number of cases evaluated as unacceptable

Generate test cases

Test cases are prepared manually by knowledge engineers. The most important criteria of these test cases is that it covers both normal cases, as well as the most difficult, and rare cases.

Solving test cases

A copy of the selected test cases is given to three or two domain of users; Students, Technical experts. Each of the groups works out the test cases independently

Evaluation of test cases

Solutions of test cases are evaluated. One or two domain experts, other than those who gave the knowledge acquired by the expert system - and of course other than those who solved the test cases - are given test cases solutions for evaluation according to the previously prepared formula.

The result

A score is given to each solution, and solutions are ranked according to these scores.

Category	Questions	Assessment	Weight	Value*Weight
Correctness of Answer	Does the software give the same answers as that a human expert would give?	3	2	6
	Does the software provide the right answer for the right reasons?	3	2	6
Accuracy of Answer	Is the software accurate in its answer?	2	2	4
	Is the answer complete? Does the user need to do additional work to get the results?	3	2	6
	Is the procedure of getting the right career simple and clear?	3	2	6
Correctness of Reasoning Technique	Does the career course change if new but irrelevant data is entered into the software?	3	2	6
	Can the system clearly explain its reasoning technique to the user?	1	1	1
	Does the system require a lot of irrelevant questions to reach the answer?	3	1	3
Reliability	Does the system hang-ups in its host computer?	3	1	3
	Does the system give warnings for cases involving incomplete data?	2	1	2
Cost Effectiveness	Is the cost of the system justified by its performance?	2	1	2
	Does the software still provide answers with incomplete knowledge?	1	1	1
Limitations	Can limitation of the system be detected at this point in time?	3 1	1 1	3 1
	Can the system learn from increased data or experience?			
Results	Add Weights *Value by 26			87 ≈ 2.4
		5-4-3-2-1-0 True.....False		5= Satisfied User 0= Unsatisfied User

Table 3: Questionnaire Results from different groups of users

The result is given by the formula as

$$P_i = 3*NE_i + 2*NG_i + 1*NA_i + 0*NU_i$$

$$P_i = 3*8 + 2*3 + 1*3 + 0*0$$

$$P_i = 24 + 6 + 3 + 0$$

$$P_i = 33$$

$$E = \frac{P_i}{N_i} = \frac{33}{14} = 2.357142857 \approx 2.4$$

This method gives a simple numerical indication of the likely satisfaction level of software's end users, the student. Although this method is based on subjective evaluation techniques; the results do provide a useful scale for using Knowledge career choice assistant as the alternative method of career decision making as shown in the results in the table above.

CHAPTER SEVEN: THE SUMMARY AND APPENDIX

7.1 SUMMARY

Chapter one highlights the background information related to career choice making process. The main purpose of this project which is development of an expert based career choice locator and advisory system for been given. The problems that have led to the development of this system which relates to lack of necessary intelligent advisory to students have been specified. The objectives expected to be met at the end of this research are also provided. Importance of this study to form four graduates in relation to career choice problems together with the scope of this system as well as the shortcomings that will be expected from the study are listed.

Chapter two contains the literature review. Chapter two begins with the definitions of career choice, factors in career choice and education and career aspirations, the proposed application of AI systems available for student's selection process in Kenya and their limitation is highlighted.

Secondly Expert system is discussed based on its definitions and brief history, applications types of knowledge-base, characteristics and comparisons with human experts its methodology, advantages and lastly related research work

It also highlights the various methodologies and eventually we select one methodology to be applied in the design of the proposed career choice locator and advisory expert system

The methodology presented in this paper ties three main aspects: theoretical basis, practical implementation, and workflow organization. The theoretical basis are derived from CommonKADS, the second generation expert system approaches state of the art.

A spiral model for knowledge base development is introduced, through this model, knowledge engineering activities and software engineering activities are described, also we described how the expert system evolves starting from research prototype, till the development of the production version.

7.2 APPENDIX

7.2.1. Semantic Network Example

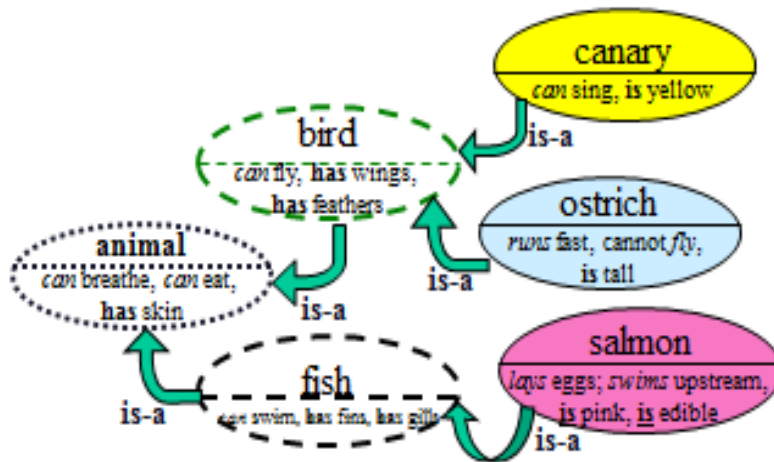


Figure 19: Semantic Network diagram

7.2.2. Results generated by the system when run in command window

Results

```
Jess> (batch "careerchoice.clp")
MAIN::fo-Engineering: +1+1+1+t
MAIN::readdata: +1+1+2+t
TRUE
Jess> (batch "careerchoice.clp")
MAIN::fo-Engineering: +1+1+1+t
MAIN::readdata: +1+1+2+t
TRUE
Jess> (batch "careerchoice.clp")
MAIN::fo-Engineering: +1+1+1+t
MAIN::readdata: +1+1+2+t
TRUE
Jess> (assert (mean-grade C+))
==> f-0 (MAIN::mean-grade C+)
<Fact-0>
Jess> (assert (Mathematics-grade C+))
==> f-1 (MAIN::Mathematics-grade C+)
<Fact-1>
Jess> (assert (Physics-grade C+))
==> f-2 (MAIN::Physics-grade C+)
<Fact-2>
Jess> (assert (Chemistry-grade B-))
==> f-3 (MAIN::Chemistry-grade B-)
<Fact-3>
```

```

Jess> (assert (Biology-grade C+))
==> f-4 (MAIN::Biology-grade C+)
<Fact-4>
Jess> (assert (Geography-grade C+))
==> f-5 (MAIN::Geography-grade C+)
<Fact-5>
Jess> (assert (Electricity-grade C+))
==> f-6 (MAIN::Electricity-grade C+)
<Fact-6>
Jess> (batch "careerchoice.clp")
MAIN::fo-Engineering: +1+1+1+t
MAIN::readdata: +1+1+2+t
TRUE
Jess> (assert (mean-grade C+))
)
FALSE
Jess> (run)
0
Jess> (reset)
==> Focus MAIN
==> f-0 (MAIN::initial-fact)
==> Activation: MAIN::readdata : f-0, f-0
==> f-1 (MAIN::school-of-Engineering (grades ) (uotput nil))
==> f-2 (MAIN::>= mean-grade C+)
==> f-3 (MAIN::>= Mathematics-grade C+)
==> f-4 (MAIN::>= Physics-grade C=)
==> f-5 (MAIN::>= Chemistry-grade C+)
==> f-6 (MAIN::>= Boilogy-grade C+)
==> f-7 (MAIN::>= Geography-grade C+)
==> f-8 (MAIN::Electricity-grade C+)
TRUE

```

This is the Engineering course function that facilitates the FIRE action once the march is found

```

[STUINF]
No=EOF
name= EOF
M.G=EOF

```

```

[Engineering]
Accepted=You can pursue Engineering course thanks for using career assistant system.
Recommended=TRUE

```

7.2.3 Knowledge-based Questionnaire form

KNOWLEDGE-BASED CAREER CHOICE ASSISTANT QUESTIONNAIRE

Part I: Personal Information

[1] In what age group are you?

- ☐ 19 and under
- ☐ 20- 24
- ☐ 24 - 28
- ☐ 28 - 32

[2] Gender

- ☐ Male
- ☐ Female

[3] In terms of your current occupation, how would you characterize yourself?

- ☐ Student
- ☐ Designer
- ☐ Technical Expert
- ☐ Other, please specify:.....

Part 2: To be completed during and/or after software use

Category	Questions	Assessment	Weight	Value*Weight
Correctness of Answer	Does the software give the same answers as that a human expert would give?		2	
	Does the software provide the right answer for the right reasons?		2	
Accuracy of Answer	Is the software accurate in its answer?		2	
	Is the answer complete? Does the user need to do additional work to get the results?		2	
	Is the procedure of getting the right career simple and clear?		2	
Correctness of Reasoning Technique	Does the career course change if new but irrelevant data is entered into the software?		2	
	Can the system clearly explain its reasoning technique to the user?		1	
	Does the system require a lot of irrelevant questions to reach the answer?		1	
Reliability	Does the system hang-ups in its host computer?		1	
	Does the system give warnings for cases involving incomplete data?		1	
Cost Effectiveness	Is the cost of the system justified by its performance?		1	
	Does the software still provide answers with incomplete knowledge?		1	
Limitations	Can limitation of the system be detected at this point in time?		1	
	Can the system learn from increased data or experience?		1	
Results	Add Weights *Value by 26			
5= Satisfied User, 0= Unsatisfied User		5-4-3-2-1-0 True.....False		

Table 4: Questionnaire table used by the students when evaluating the knowledge-based career choice assistant

7.3. CONCLUSION AND FUTURE WORK

In this report we have presented the design and development of a proposed knowledgebase system to help the student select the best suitable career based on his/her KSCE results. Our knowledgebase system is a rule based system, and we used JESS language to store our knowledge base.

Substantial research has been done on the use of expert systems for pattern matching or classification. Many research studies have been conducted on the application of expert systems such as in agriculture, diagnostics, and medicine etc. There is very little if any research that has been done in the area of student selection for University admission. There are many challenges faced by students in selecting their suitable career course. These challenges arose due to lack of suitable intelligent system.

Knowledgebase system could play a beneficial role in trying to resolve these challenges, especially given the several advantages they have. One such advantage is that universities are likely to have highly motivated students as these students will now be doing courses of their own choice. This may lead to improved chances of having a better graduate at the end of the training. This in itself will translate to improved quality of the manpower produced.

.

CHAPTER EIGHT: REFERENCES

- Adrion, W., Branstad, M., Cherniovsky, J. (1982) Validation, Verification and Testing of Computer Software, ACM Computing Surveys , Vol. 14, No. 2,.
- Akerkar, Rajendra & Sajja, Priti, (2009) *Knowledge-Based Systems*, Jones & Bartlett Publishers.
- Azaab S., Abu Naser S., and Sulisel O. (2000). A proposed expert system for selecting exploratory factor analysis procedures, Journal of the college of education, 4(2):9-26.
- B. Chandrasekaran, Generic tasks in knowledge based reasoning. High level building blocks for knowledge-base design. IEEE Expert. 1(3):23-30.
- B.J. Wielinga, *et al* (1993), Second Generation Knowledge-base, Pages 299-335, Springer-Verlag, Berlin - Heidelberg, Germany.
- B. Wielinga & J. Breuker.(May, 1991) KADS: A Model Approach to Knowledge Engineering, Esprit project 5 248 KADS II..
- Basavage, R. (1996). Gender-role stereotyping and how it relates to perceived future career choices among elementary school children. Unpublished master's thesis, University of Wisconsin-Stout, Menomonie.
- Bedi, P. and Marwaha, S. (2005). Framework for Ontology Based Knowledge-base: Disease & Pests Identification in Crops - A Case Study. In: International Conference of Artificial Intelligence (ICAI-05): 256-259.
- Bedi, P., Marwaha, S. (2004). Designing Ontologies from Traditional Taxonomies. In: International Conference on Cognitive Science, Allahabad, India
- Berners-Lee T., Hendler J., Lassila O. (2001). The Semantic Web. vol. 284, no. 5, pp. 34-43. Scientific American
- Brickley D. and Guha R.V. (1999). Resource Description Framework (RDF) Schema Specification 1.0. W3C Proposed Recommendation, <http://www.w3.org/TR/1999/PR-rdf-schema> [Accessed 4th October, 2012].
- Cannady, J. (1998). *Artificial neural networks for misuse detection*. Retrieved February 17, 2010 from <http://csrc.nist.gov/nissc/1998/proceedings/paperF13.pdf> [Accessed 17th October, 2012].
- Castillo, E., Alvarez, E. (1991). Knowledge-base: Uncertainty
- Chacha, C. (2004). Reforming higher education in Kenya: Challenges, lessons and opportunities. The Inter- university Council of East Africa at the State University of New

York Workshop with Parliamentary \Committee on Education, Science and Technology, Aug. 2004, Naivasha, Kenya. Retrieved on March 24 from <http://www.iucea.org> [Accessed 23rd October, 2012].

Chitta B.(2001). Knowledge representation reasoning and declarative problem solving. Arizona State University

Clark, K. (2008).SPARQL Protocol for RDF. W3C Recommendation, <http://www.w3.org/TR/rdf> [Accessed 23rd October, 2012].

Dean, M., Connolly, D., Harmelen, F.V., Hendler, J., Horrocks, I., McGuinness, D.L., Patel- Schneider, P.F. and Stein, L.A.(2003). OWL Web Ontology Language Reference. W3C Working Draft, <http://www.w3.org/TR/2003/WD-owl-ref-20030331> [Accessed 26th October, 2012].

Engelbrecht A. P., (2002). Computational Intelligence An Introduction. John Wiley & Sons. and learning. Computational Mechanics Publications and Elsevier Applied Science. Southampton Boston and New York.

Gillard, S., Bailey, D., & Nolan, E. (2008). Ten reasons for IT educators to be early adopters of IT innovations. *Journal of Information Technology Education*, 7, 21-33. Retrieved from <http://www.jite.org/documents/Vol7/JITEv7p021-033Gillard257.pdf> [Accessed 23rd October, 2012].

Goddard (2009). Knowledge-base Notes Chapter 7 (CpSc810)

Gysbers, N.C. (2007). Facilitating career development through comprehensive guidance and counseling programs K-12. Professional Counselling Digest ACAPCD-04.

Hewitt, J. (2010). Factors influencing career choice. Cited from www.ehow.com on 15/02/2020.

I. Benbasat and J. S. Dhaliwal, A Framework for Validation of Knowledge Acquisition, knowledge Acquisition, Vol.1 No.2, 1989, PP 215-233.

J.McDermott. (1988) Preliminary steps towards a taxonomy of problem solving methods. S.Marcus, editor, Automating knowlegde acquisition for knowledge-base. Pages 225-255.

Kluwer, Boston, Kerka, S. (2000). Career development, gender, race and class.Eric Clearing house on Adult Career and Vocational Education Columbus. ED 421641

Lindsay, Robert K., Buchanan, Bruce G., Feigenbaum, Edward A., & Lederberg, Joshua, (1980) *Applications of Artificial Intelligence for Organic Chemistry: The Dendral Project*, McGraw-Hill.

Lindsay, Robert K., Buchanan, Bruce G., Feigenbaum, Edward A., & Lederberg, Joshua, (1993) “DENDRAL: A Case Study of the First Expert System for Scientific Hypothesis Formation”, *Artificial Intelligence*, Vol. 61, No. 2, pp209-261.

Liebowitz Jay. (2006). The Handbook of APPLIED KNOWLEDGE-BASE, CRC Press LLC.

MyMajors, <http://www.mymajors.com> [Accessed 8th September, 2012].

Munavu, R., Ogutu, D., & Wasanga, P. (2008). Sustainable articulation pathways and linkages between upper secondary and higher education in Africa. *Association for the Development of Education in Af-rica*

National Center for Assessment in Higher Education in Saudi Arabia, <http://www.geyas.org>. [Accessed 12th September, 2012].

<http://www.ict.griffith.edu.au/noran/> [12th October, 2012].

Okumu, J. (2009). Foreword In: Gitonga, C. A systematic guide to career choice. Nairobi: Intergrity.

Oyamo, O. R., and Amoth, D (2008). Choice of final year options by undergraduate students at the Moi

School of Information Sciences. East African Journal of Information Science.

Technical report No. TR-88-024-31, Guide to KADS implementation using the knowledge representation object language KROL. December 1993.

Technical Report No. TR-88-024-41, Methodology for the Engineering of Knowledge-base -Version 6.0, December, 1995.

Verma, Jindal &Aggarwal, Jain, (2010) “An Approach towards designing of Car Troubleshooting Expert System”, *International Journal of Computer Applications*, Vol. 1, No. 23, pp107-114.