Addis Ababa University

(Since 1950)

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

SCHOOL OF INFORMATION SCIENCE

A CASE BASED REASONING KNOWLEDGE BASED SYSTEM FOR TYPE II DIABTES MANAGEMENT: CASE OF DESSIE REFREAL HOSPITAL

ZEWDITU SISAY

NOVEMBER 2011

ADDIS ABABAUNIVERSITY

SCHOOL OF GRADUATE STUDIES

SCHOOL OF INFORMATION SCIENCE

A CASE BASED REASONING KNOWLEDGE BASED SYSTEM FOR TYPE II DIABTES MANAGEMENT: CASE OF DESSIE REFREAL HOSPITAL

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

By

ZEWDITU SISAY

NOVEMBER 2011

ADDIS ABABAUNIVERSITY

SCHOOL OF GRADUATE STUDIES

SCHOOL OF INFORMATION SCIENCE

A CASE BASED REASONING KNOWLEDGE BASED SYSTEM FOR TYPE II DIABTES MANAGEMENT: CASE OF DESSIE REFREAL HOSPITAL

By

ZEWDITU SISAY

Name and signature of Members of the Examining Board

Name	Title	Signature	Date
Henock Lulseged	Chairperson	Henok	08/11/11
Getachew Jemaneh	Advisor(s),	Getachew Jemaneh	08/11/11
Workshet Lamenew	Examiner,	[illegible]	08/11/11

Acknowledgment

First and foremost I would like to thank God and Holy Mother who made all things possible, and granted me success in my thesis work and entire journey.

I would like to thank my advisor Ato Getachew Jemaneh who patiently advised me with scholar guidance and constructive comments on my work throughout this endeavor. My deep appreciation and thanks goes to all Dessie Hospital and Ethio General Hospital staffs, particularly to Dr. Said Mohhamed, Sister Zewd Demesse and Dr.Tewdrose Abebe for their cooperation during knowledge acquisition.

Next, my especial thanks go to my beloved husband Ato Habte Zeleke for his encouragement and patience with all my pursuit.

Table of Contents

List of Tables

List of Figures

List of Acronyms

AAU- Addis Ababa University

AI- Artificial Intelligence

AIDS- Acquired Immunodeficiency Syndrome

AIM- Artificial Intelligence in Medicine

CBR- Case-Based Reasoning

DARPA- Defense Advanced Research Projects Agency

DH- Department of Health

DM- Diabetes Mellitus

EDA- Ethiopian Diabetes Association

FBG- Fasting Blood Glucose

FPG- Fasting Plasma Glucose

HIV- Human Immunodeficiency Virus

IJCAI- International Joint Conference on Artificial Intelligence

IT- Information Technology

JCOLIBRI- Java Class Ontology Libraries Integration for Building Reasoning Infrastructure

Knowledge Base Systems

KBS- knowledge base systems

KM- knowledge management

NIDDK- National Institute of Diabetes and Digestive and Kidney Diseases

OGTT- Oral Glucose Tolerance Test

OHA- Oral Hypoglycemic Agents

OPD- Outpatient Department

PACE- Patient Care Expert System

RBR- Rule-Based Reasoning

RBS- Rule-Based System

WHO- world Health Organization

XML- Extended Markup Language

Abstract

Diabetes mellitus (DM) is a common chronic disease around the world in which the body does not produce insulin (Type I diabetes) or does not properly use insulin (Type II diabetes). The study investigated the potential of case-based reasoning (CBR) approach for type II DM treatment. CBR is an approach to Artificial Intelligence that is intended to mimic an approach that people typically use to solve problems. This is the use of past experiences to reason about new situations.

In order to acquire the knowledge, the researcher conducted unstructured interviews with domain experts selected through expert sampling and other relevant documents. Then the knowledge is modeled using tree like structure called ladders. Patient history cards from Dessie Referral Hospital in outpatient department (OPD) were the primary sources of cases. Case attribute identification and weight assignment were done with the help of domain experts. The case-based contained 42 type II DM cases and stored in plain text file (attribute value pair vector).The prototype is built using jCOLIBRI, a software artifact that promotes software reuse for building CBR systems. jCOLIBRI employed Nearest Neighbor Retrieval algorithm for retrieval and propose the most similar cases for reuse. Manual revision is done by the domain expert in order to adapt a stored case's solution for a new case. There is always incremental learning through retaining newly solved cases.

The prototype performance is evaluated through statistical analysis and user feedbacks. Recall and precision were the main statistical performance measures using leave-one-out cross validation testing proportion. The retrieval performance of the prototype showed average value of 69% recall and 46% precision. Domain experts were also evaluating the prototype using certain criteria.

The main objective the research is to design and build CBR knowledge based system that retrieves relevant previously stored cases and proposes appropriate solution. The developed prototype scores promising performance and user acceptance.

CHAPTER ONE

INTRODUCTION

1.1. Background

Diabetes is a disorder of metabolism (the way the body uses digested food for growth and energy) (NIDDK, 2008). The body is made up of millions of cells that need energy to function. The food is turned into sugar, called glucose. Sugar is turned out to the cells through the blood stream. It is one of the many substances that are needed by the cells to make energy. The pancreas automatically produces the right amount of insulin to move glucose from blood into the cells. In people with diabetes, however, the pancreas either produces little or no insulin, or the cells do not respond appropriately to the insulin that is produced. Glucose builds up in the blood, overflows into the urine, and passes out of the body in the urine. Thus, the body loses its main source of fuel even though the blood contains large amounts of glucose (The Patient Education Institute, 2010). In short, diabetes is a disease that makes difficult for the cells of the body to get the glucose to make energy.

Severe complications that can come from diabetes include heart disease, vascular (blood vessel) disease and poor circulation, blindness, kidney failure, poor healing, stroke, and other neurological (nerve) diseases. Diabetes cannot be cured but can be successfully treated. Complications from diabetes can be prevented with careful blood sugar management, control of high blood pressure and cholesterol levels (Tropy, 2009).

Even among policy makers at international and national levels, awareness about the public health and clinical importance of diabetes remains low. Diabetes is widely perceived as a condition of low importance to the poorer populations in the world. In the low- and middle-income countries, the impact of diabetes is largely unrecognized. Yet, the world is facing a dramatic rise in diabetes prevalence, most of which will occur in the low- and middle-income countries. This will have a major impact on the quality of life of hundreds of millions people and their families, constrained the capability of many national health-care systems and impact adversely upon the economy of those countries that are in most need of development (Unwin and Marlin, 2004).

According Unwin and Marlin (2004), the world is facing a growing diabetes epidemic of potentially devastating proportions. Its impact will be felt most severely in the developing countries. There are many facts about severity of diabetes:

- Worldwide, 3.2 million deaths are attributable to diabetes every year.
- One in 20 deaths is attributable to diabetes; 8700 deaths every day; six deaths every minute.
- At least one in ten deaths among adults between 35 and 64 years is attributable to diabetes.
- Three-quarters of the deaths among people with diabetes aged under 35 years are due to their condition.

1.1.1. Types of Diabetes

Sometimes diabetes is named as diabetes mellitus (DM). There are two basic types of DMs.

Type I DM: often referred to as insulin-dependent diabetes and occurs when the body's pancreas does not produce enough insulin (the hormone that processes glucose). Type I DM is usually diagnosed in childhood or people less tan 40 years old. People with this type of diabetes need daily injections of insulin. If not diagnosed and treated with insulin, the person can lapse into a life threatening coma (Armengol*et al,* 2000).

Type II DM: also called "adult-onset" diabetes, is much more common, and usually develops in adults over the age of 40 being more common among adults over 55. Usually, people with diabetes type II have overweight and sedentary lifestyle. Insulin resistance is a major issue in type II diabetes that means pancreas produces insulin but the body does not use it effectively. Some people with diabetes type II must inject insulin, but most are controlled with a combination of weight loss, exercise, and prescription of oral diabetes medication (Torpy, 2009) and (Armengol*et al*, 2000).

1.1.2. Case-based Reasoning

Information technology (IT) has been successfully integrated into many different sectors of the world economy. However, one area where IT has not had much influence until recently is the field of medicine. Very recently healthcare organizations are being driven to implement IT to improve results. One of recently emerging discipline is knowledge management (KM) that focuses on efforts leading to the rational allocation of organizational knowledge assets, which is going to be implemented as Knowledge-Base systems (KBS). KBS has been applied in numerous applications in the health science domain (Althoff and Weber, 2006).

KBSs are computer programs that achieve expert-level competence in solving problems on specific task areas. KBSs are based on a coded body of human knowledge represented through some model. The task of developing a KBS is generally known as knowledge engineering. The specific task of collecting human knowledge and representing it through model is known as elicitation (Alberto *et al*, 2009). One of the components of artificial intelligence is inference engine (reasoning and search strategy for solution).There are different ways of reasoning strategy like rule-based, case-based ,inductive, deductive and probabilistic. The two popular approaches used in KBS are rule-based reasoning (RBR) and case-based reasoning (CBR). RBR is used to model human problem-solving activity and adaptive behavior in the form of IF-THEN rules. Satisfaction of the rule antecedent gives rise to the execution of the consequent; that is one action is performed (Leondes, 2000).

CBR is an artificial intelligence (AI) approach that capitalizes on past experience to solve current problems. It may be viewed, simultaneously, as a research paradigm, as a perspective on human cognition and as a methodology for building practical intelligent systems (Bichindaritz,& Marling, 2006). Case-based reasoning is a paradigm of artificial intelligence, like neural networks, genetic algorithms, multi-agent systems, Bayesian networks, etc.

CBR represents knowledge in the form of cases. To find solutions, case-based reasoning uses analogical reasoning, which is the process of determining the outcome of a current problem by comparing input problems to similar past experiences. This analogical reasoning results in finding previous cases similar to the present problem and then adapting the previous solutions to fit the current problem (Pacharavanich*et al,* 2000).

As of (Salem , 2007), CBR means reasoning from experiences or "old cases" in an effort to solve problems, critique solutions, and explaining anomalous situations People tend to be comfortable using CBR methodology for decision making, in dynamically changing situations and other situations were much is unknown and when solutions are not clear. CBR is also an incremental and evaluative reasoning process since it acquires experience every time when a new problem is solved.

Cordier (2009) in his doctoral thesis mentioned that, one aspect that distinguishes CBR from other methodologies of the artificial intelligence domain is the fact that it is based on the reuse of experiences; solutions are reused rather than built up from theoretical knowledge. This type of reasoning is closer to the way humans sometimes think in real life

CBR systems are often seen as an alternative to rule-based expert systems because they avoid some shortcomings and problems of rule bases. The first problem of rule-based expert systems is knowledge acquisition. When we extract tacit knowledge from human experts, they cannot make list of hundreds of rules, because experts most often does not use rules in problem solving. The process of knowledge acquisition often results in incorrect rules. On the other hand, CBR tries to model human expert reasoning in a more natural way by collecting cases of problem solving experiences. The behavior of human expert is in no way matches that of a rule-based. The central feature of expertise is experiences. When confronted with a problem, an expert is reminded of previous similar problems and their respective solutions. Even rules that the expert may be using are rooted in actual experiences, which have been distilled into a general formula of action. Thus, the basic unit of knowledge for an expert is not a rule but a case. Consequently, it is easy to acquire expert knowledge if the knowledge engineer asks for cases and experiences, rather than for rules (Leondes, 2000)

The second problem of rule-based reasoning is lack of memory. For example, when a medical digenesis system is presented with a patient, it might use hundreds or even thousands of rules to reach a diagnosis. When presented exactly the same case again, it would have to reuse the same set of rules to reach the exact same conclusion. This lack of memory leads to computational inefficiency. A CBR system remembers its previous problem solving activities and can apply old solutions to the same problem without high computational costs (Leondes, 2000).

The third problem of rule-based system is alleviated by CBR is scalability. Although RBS are easy to prototype, the move from small-scale to large-scale program is not linear (Alemu, 2010).

In general, technology of CBR offers new and improved methodological approaches to the creation of intelligent systems. CBR provides the developer of intelligent systems with an ease of knowledge acquisition, learning capabilities that allow the system to evolve and improve from experiences, and robustness in its reasoning and problem-solving processes.

1.2. Statement of the Problem and its Justification

Chronic medical conditions are growing causes of death among people in developing countries. This situation is aggravated by the migration to the towns and cities of subsistence farmers and their families from the rural areas (Shtitaye and Watkins, 2004). The number of people with diabetes is increasing due to population growth, aging, urbanization, and increasing prevalence of obesity and physical inactivity (Wildet al, 2004). Diabetes, though less common than in the wealthy countries such as those in Western Europe and North America, is not rare in Ethiopia (Shtitaye and Watkins, 2004). According to Ethiopian Diabetes Association (EDA), an estimated number of 246 million people, or 5.9% of the world's population, in the age group 20-79 have diabetes worldwide in 2007. Of these, more than 70% live in the developing countries. In Ethiopia, the number of deaths attributed to diabetes reached over 21,000 in 2007.

The study conducted by Shtitaye and Watkins (2004) indicated that access to care is a major problem which contributes to the poor diagnosis for people with diabetes in Ethiopia. These people have to travel long distances to the nearest medical center in order to obtain insulin and receive medical care. Most of the rural people with diabetes had to travel more than 40 km to reach their nearest hospital, 23% travelling more than 100 km and 13% over 180 km.

The health status of the Ethiopia is extremely poor. Inadequate health-care infrastructure, affordability, lack of adequate training and retraining of health care providers, and lack of education to the people living with diabetes and their families are main barriers to quality care for diabetes patients in Ethiopia.

In developing countries, such as Ethiopia, chronic disease is a growing problem. These problems are aggravated by the shortage of trained nurses, clinicians and health facilities. In 2000, there were 103 hospitals and 338 medical centres in Ethiopia. There are only two medical schools in Ethiopia and Ethiopians share fewer than three physicians per 100,000 people (Tuso, 2009).

With this worst fact of Diabetes in Ethiopia, application of KBSs in the domain is very critical. IT can be successfully integrated into the problem domain to support specialists in providing the intended treatment and consultation for diabetes patients. Therefore, this study is designed to build KBS prototype to this problem domain using CBR approach.

One of the intuitively attractive features of CBR in medicine is that the concept of patient and disease lends itself naturally to a case representation. A biological system like the human body is difficult to describe by general models of rules. Case-based reasoning is particularly well adapted to the resolution of open problems for which the associated domain theory is weak or difficult to formalize which is open and unstructured.

Specifically, case-based reasoning seems promising for diabetes management because there is a large experience base of assisting patients with problems in blood glucose control; and CBR can integrate numeric data, such as blood glucose readings with descriptive and personal preference data such as work schedules and lifestyle choices. In addition, diabetes management guidelines are general in nature, requiring personalization and CBR has been successfully applied to managing other chronic medical conditions. Management of diabetes is complex since physical and lifestyle factors frequently combine in complex ways to impact blood glucose levels in people with diabetes. This makes the diagnosis and treatment of diabetes to vary extremely from patient to patient in terms of sensitivity to insulin, response to environmental and lifestyle factors, propensity for complications, compliance with physician's orders, and response to treatment (The Patient Education Institute, 2010) and (Marling et al , 2007). Management of diabetes typically involves a considerable element of self-care, and advice should be aligned with the perceived needs and preferences of people with diabetes (NIHCE, 2008).Therefore, to provide individualized decision support that can help each patient to maintain good blood glucose control, the researcher proposed a CBR approach.

Thus, to alleviate the above mentioned and other problems as well as to expand the diabetes treatment centers all over the country, we should come across with alternative solution to control the harshness of the disease. This can be achieved only when the country's diabetes treatment service is assisted with knowledge-based systems and utilization of recent technologies.

In Ethiopia, there are researches done to investigate application of KBS in the area of medicine but some of them uses a rule based reasoning techniques. For instance in health area, Anteneh (2004) attempt to design a prototype knowledge based system for antiretroviral therapy by using rule based reasoning technique; Rediet (2006) developed a prototype of knowledge based system for HIV pretest counseling by using rule based reasoning. The other researcher Solomon (2010) is also applied rule based reasoning for diagnosis of acute respiratory tract infections. Even though Rule Based Reasoning technique has some advantage in developing knowledge based system, it has drawback when it is applied to medical domain. One of the main limitations is that knowledge acquisition process in medical domain is challenging and it is difficult to represent all the knowledge in the form of rules.

Alemu (2010) tried to design a knowledge based system for AIDS resource center by using a case based reasoning approach. But his system tested using informal methods using one or two physician evaluations and the retrieval algorithm followed is not clear.

This study is conducted with the aim of filling the gap which is stated in the above section by investigating the case based reasoning approach in designing knowledge based system that can diagnosis and treat type II diabetes cases. Hence, this study is new and an alternative research reference for future researchers.

1.3. Objective of the Study

This research has the following general and specific objectives.

1.3.1. General Objective

The general objective of this research is to investigate the potential of CBR Knowledge-Based system that can assist experts in giving efficient treatment and consultation service for type II DMpatients.

1.3.2. Specific Objectives

To achieve the general objective, this study has the following specific objectives.

- To review literatures on areas related to Knowledge-Based systems using CBR approach and available technologies.

- To acquire domain knowledge through unstructured interviews, discussions with domain experts, physical observations, and document analysis at Dessie Referal Hospital.
- To model the required knowledge for type II DM management.
- To characterize the acquired cases, define case structure, and construct the case-base that contain relevant attributes
- To design the KBS
- To build a KBS prototype using CBR tool (jCOLIBRI)
- To evaluate the performance of the prototype using standard statistical methods and user evaluations.
- To report the result, make conclusion, and forward recommendations for further studies.

1.4. Scope and Limitation of the Study

There are a number of type II DM patient treatments. It is challenging to incorporate exhaustively all treatments and drudge response managements. Due to shortage of time and other resources this study considered only basic treatments oral hypoglycemic agents (OHA) and insulin injection. To build a CBR system there are series of steps to be followed, case retrieval, case reuse, case revise and case retain. However, this research is mainly focused on development of a CBR prototype with case retrieval.

There was no ready made collection of case within the hospital so the researcher tracked cases when the patients come for treatment during their appointment at OPD. The researcher collected cases based on their accessibility not on quality, this creates coverage problem and has impact on case retrieval performance of the prototype. Accessible cases has also completeness problem. They missed some attributes like weight, height, etc and attribute weighting during case structure construction was done manually with the help of domain experts.

1.5. Methodology of the Study

Research methodology refers to the way in which we approach problems and seek answers (Steven and Robert, 1984). Other scholars define methodology as "the theory how inquiry should proceed" that involves "analysis of principles and procedures in a particular field of inquiry". In short, methodology presents "a theory and analysis of how research does or should proceed" (AAU, 2009).

Scientific researchers must be systematic and follow series of steps and rigid standard protocols that emanate from methodologies. A methodology has guidelines as the benchmark for measuring the validity of the results obtained. There must be a clear procedure to replicate experiments and verified results (AAU, 2009). With this intuition, this research strictly followed the following methods.

1.5.1. Literature Review

Different sources of information such as books, journals, magazines, published and unpublished theses, proceedings and the Internet were referred to get depth conceptual understanding about the domain of diabetes. In addition, literature review enabled the researcher to look related research works in KBS development using CBR approach.

1.5.2. Data Collection

Cases were collected at outpatient department (OPD) from the patient history cards by consulting domain experts. The researcher conducted unstructured interviews and discussions with domain experts such as doctors, nurses, and other health care professionals for more elicitations of cases and to acquire the knowledge in the area.

1.5.3. Development Tool

There are a number of tools to the model of CBR, like jCOLIBRI, Esteem, Caspian, CasePower, ReCall,ReMind , ART*Enterprise, Kate, Eclipse and Case Advisor. For this research jCOLIBRI is used as a development tool. The motivation of choosing jCOLIBRI is because of its features of flexibility and convenience (Iqbal andAshraf, 2006). The researcher is also more familiarity with the tool. jCOLIBRI is an object-oriented framework in Java for building CBR applications. It is a software artifact that promotes software reuse for building CBR systems, integrating the application of well proven software engineering techniques with a knowledge level description that separates the problem solving method that defines the reasoning process, from the domain model, that describes the domain knowledge (Recio-Garcia and Diaz-Agudo, 2004).

1.5.4. Evaluation Mechanism

The performance of the system was evaluated by domain experts' feedback and standard system performance evaluation methods of retrieval. To figure out the retrieval performance precision and recall were used.

1.6. Significance of the Study

Even thought the study conducted for academic purpose at Dessie Referal Hospital, it can be adapted to any diabetes treatment center and would have practical value to physicians and diabetic patients. Associations which fight against diabetes, like, Ethiopian diabetes association can also use the result of this study to foster their service. The overall result of this research will contribute to the effort of improving and prolong the quality of life for people living with diabetes in general and reduce the harshness of the disease. Finally this research can serve as spring-bored for future researchers in the domain area.

CHAPTER TWO

LITERATURE REVIEW

2.1. Diabetes Overview

Diabetes is a chronic disease that occurs either when the pancreas does not produce enough insulin or when the body cannot effectively use the insulin it produces. Insulin is a hormone that regulates blood sugar. Hyperglycemia, or raised blood sugar, is a common effect of uncontrolled diabetes and over time leads to serious damage to many of the body's systems, especially the nerves and blood vessels (WHO, 2011).

2.1.1. Global Prevalence of Diabetes

According to the report of WHO (2011), globally, more than 220 million people have diabetes. In 2004, an estimated 3.4 million people died from consequences of high blood sugar. More than 80% of diabetes deaths occur in low- and middle-income countries. Diabetes was once considered a rare disease in sub- Saharan Africa. But in 2010, over 12 million people in sub-Saharan Africa are estimated to have diabetes, and 330,000 people will die from diabetes-related conditions. Over the next 20 years, it is predicted that sub-Saharan Africa will have the highest growth in the number of people with diabetes of any region in the world – the 2010 estimated number is predicted to almost double in 20 years, reaching 23.9 million by 2030 (Diabetes Leadership Forum Africa, 2010). For example, in Ethiopia, diabetes patient in 2005 was 796,000 and this number will reach to 1,820,000 in 2030 (WHO, 2011).

2.1.2. Diabetes Complication

Diabetes and its complications have significant economic impact on individuals, families, health systems and countries (WHO, 2011). Diabetes more often affects people of lower socio-economic status who carry a greater disease burden for many reasons including limited access to or utilization of health care, poor nutrition, and optimal physical activity (WHO, 2006).

Diabetes can have a major impact on the physical, psychological and material well-being of individuals and their families. As it is life-long disease diabetes can also have a profound impact on lifestyle, relationships, work, income, health, well-being, life expectancy, and health and social services (DH, 2001).

Access to appropriate diabetes care in sub-Saharan Africa is often extremely limited because of inadequate healthcare systems, shortage of doctors and nurses with adequate training in diabetes diagnosis and treatment. In Ethiopia, there is no practice in raising awareness of a healthy lifestyle in the community, among young people and diabetes diagnosis among healthcare professionals. Besides, there is limited effort in developing a national diabetes program and reaching remote communities to offer medical checks and advice (Diabetes Leadership Forum Africa, 2010).

2.1.3. Symptoms and Testing of Diabetes

Typical symptoms of diabetes include excessive thirst, fatigue, frequent illness or infections, poor circulation, wounds that do not heal; blurred vision, and weight loss. Many people with type II diabetes have no symptoms, and it is discovered after testing for other medical problems or through screening in persons at high risk of developing type II diabetes (**Tropy**, 2009).

The fasting blood glucose test is the preferred test for diagnosing diabetes. The test is most reliable when done in the morning. However, a diagnosis of diabetes can be made based on any of the following test results, confirmed by retesting on a different day (NIDDK, 2008).

- A blood glucose level of 126 mg/dL or higher after an 8-hour fast. This test is called the fasting blood glucose test.
- A blood glucose level of 200 mg/dL or higher 2 hours after drinking a beverage containing 75 grams of glucose dissolved in water. This test is called the oral glucose tolerance test (OGTT)

- A random—taken at any time of day—blood glucose level of 200 mg/dL or higher, along with the presence of diabetes symptoms.

A range for a normal fasting blood sugar is between 60 and 99 mg/dL. Levels between 100 and 125 mg/dL are considered pre-diabetic levels (The Patient Education Institute, 2010).

2.1.4. Treatment

Diabetes can not be cured; however, it is possible to keep the level of sugar within the normal level (The patient education institute, 2010). Before the discovery of insulin in 1921, everyone with diabetes died within a few years after diagnosis. Although insulin is not considered a cure, its discovery was the first major breakthrough in diabetes treatment. The treatment and management of diabetes varies from patient to patient (NIDDK, 2008) .Management of the person with diabetes requires the skills of several professionals (general practitioner, specialist physician, diabetes educator, podiatrist, dietitian, ophthalmologist or optometrist, exercise professional and dentist) and the active participation of the patient (Diabetes Australia, 2010).

A bad management of diabetes will produce micro complications (such as blindness, renal failure or polyneuropathy), and macro complications (such as gangrene and amputation, aggravated coronary heart disease or stroke). Therefore, main concern in the management of the diabetes is reducing the risk of a patient developing a new long-term complication and the risks of progression in the complications already present (Armengol, 2000).

2.1.4.1. Non-drug Treatment

Various literatures stress on healthy life style to keep the level of sugar normal within the body. As described by NIDDK (2008) and Tropy (2009), Eating right (changing what to eat, how much to eat and how often to eat), Physical exercise (walking for 30 minutes, ridding bicycles, swimming) and ensuring good hygiene are the basic non-drug therapies for type II DM.

2.1.4.2. Drug Treatment

Multiple interventions and medications are needed to control the multiple risk factors associated with type II diabetes. According to Diabetes Australia (2010), if a trial of healthy lifestyle after a few months is unsuccessful in controlling blood glucose in a person with type II diabetes; oral hypoglycemic agents (OHA) tablets can be used. Medication can be used early to decrease glucose levels and relieve symptoms. Metformin is the medication of first choice in people who is overweight or obese. Metformin reduces hepatic glucose output and insulin resistance. Metformin has been shown to significantly reduce the risk of diabetes related morbidity and mortality in overweight patients. Even if there are number of medications, if the patient is thin Glibenclamide (Daonil) is recommended. After five years most patients do not respond to tablets, so they need insulin.

2.2. Artificial Intelligence

Artificial intelligence is the science and engineering of making intelligent machines, especially intelligent computer programs. Here the basic concept is intelligence. Intelligence is the computational part of the ability to achieve goals in the world. Varying kinds and degrees of intelligence occur in people, many animals and some machines (McCarthy, 2007).

The target of Artificial Intelligence (AI) is to make computer behave intelligently. AI is a relatively new approach to some very old problems about the nature of mind and intelligence. It combines with and contributes to several other disciplines, including: psychology, philosophy, linguistics, biology, anthropology, logic, mathematics, computer science & software engineering (Sloman, 2002).

The definition of AI varies along different scholars. Some concerned with the processes and others concerned with the ideal concept *rationality (do the "right thing")*. According to Russell and Norving (2004) , AI systems are systems that think and act like humans , means "the exiting new effort to make computers think … machines with minds " or " automation of activities that associated with human thinking, activities such as decision-making , problem solving , learning… that require intelligence". AI systems also think and act rationally which means "the study of the computation that make it possible to perceive, reason and act and intelligent behavior in artifacts".

AI has two main goals, scientific and engineering goals .Studying things that already exist or might exist, explaining how they work, searching for general principles relevant to understanding which is scientific goal. using that knowledge to solve practical problems, including making new useful kinds of machines, producing new forms of entertainment, perhaps helping us manage ourselves better is engineering goal. AI uses computer science, just as physics uses mathematics, but AI is not computer science, just as physics is not mathematics. AI uses computers because they are the best available tool, not because they are the subject of study (Sloman, 2002).

The use of artificial intelligence techniques for wide problem domains has been suggested in the literature for over a decade. Well known examples are legal reasoning, medical diagnosis and management, military tactical planning, software engineering, and related areas. These systems enhance the overall decision making processes by integrating general knowledge and individual past cases (Pacharavanich *et al,* 2000).

2.2.1. Artificial Intelligence in Medicine

The modern study of artificial intelligence in medicine (AIM) is 25 years old. Throughout this period, the field has attracted many best computer scientists, and their work represents a remarkable achievement .From the very earliest moments in the modern history of the computer, scientists have dreamed of creating an 'electronic brain'. Of all the modern technological quests, this search to create artificially intelligent (AI) computer systems has been one of the most ambitious. Intelligent computers able to store and process vast stores of knowledge, the hope was that they would become perfect 'doctors in **a** box', assisting or surpassing clinicians with tasks like diagnosis. With such motivations, a small but talented community of computer scientists and healthcare professionals set about shaping a research program for a new discipline called Artificial

Intelligence in Medicine (AIM). These researchers had a bold vision of the way AIM would revolutionize medicine, and push forward the frontiers of technology (Coiera, 1996).

AIM systems are by and large intended to support healthcare workers in the normal course of their duties, assisting with tasks that rely on the manipulation of data and knowledge. An AI system could be running within an electronic medical record system, for example, and alert a clinician when it detects a contraindication to a planned treatment. It could also alert the clinician when it detected patterns in clinical data that suggested significant changes in a patient's condition.

Expert or knowledge-based systems are the commonest type of AIM system in routine clinical use. They contain medical knowledge, usually about a very specifically defined task, and are able to reason with data from individual patients to come up with reasoned conclusions.

According to *Coiera (1996)*, there are many different types of clinical task to which expert systems can be applied.

Generating alerts and reminders, real-time situations, an expert system attached to a monitor can warn of changes in a patient's condition. In less acute circumstances, it might scan laboratory test results or drug orders and send reminders or warnings through an e-mail system.

Diagnostic assistance, when a patient's case is complex, rare or the person making the diagnosis is simply inexperienced, an expert system can help come up with likely diagnoses based on patient data.

Therapy critiquing and planning, Systems can either look for inconsistencies, errors and omissions in an existing treatment plan, or can be used to formulate a treatment based upon a patient's specific condition and accepted treatment guidelines.

Image recognition and interpretation, many medical images can now be automatically interpreted, from plane X-rays through to more complex images like angiograms, CT and

MRI scans. This is of value in mass-screenings, for example, when the system can flag potentially abnormal images for detailed human attention.

2.3. Knowledge-Based Systems (KBS)

Knowledge-Based Systems are often called Expert Systems. Their goal is trying to solve the kinds of problems that normally require human experts. Pomykalski *et al* (1999) put a precise definition for expert systems as follows

"An expert/knowledge-based system is a computer program that is designed to mimic the decision-making ability of a decision-maker(s) (i.e., expert(s)) in a particular narrow domain of expertise."

In order to understand the definition fully, Pomykalski *et al* (1999) explain core points of the definition in the following manner.

An expert/knowledge-based system is a computer program. A computer program is a piece of software, written by a programmer as a solution to some particular problem.

An expert/knowledge-based system is *designed to mimic the decision-making ability.* The specific task of an expert/knowledge-based system is to be an alternative source of decision-making ability for organizations to use; instead of relying on the expertise of just one person who qualified to make a particular decision. An expert/knowledge-based system attempts to capture the reasoning of a particular person for a specific problem. Expert/knowledge-based systems are often feared to be replacements for decision makers, however, in many organizations; these systems are used to "free up" the decision-maker to address more complex and important issues facing the organization.

An expert/knowledge-based system uses a decision-maker(s) (i.e., expert(s). An expert can be defined as

"One with the special skill or mastery of a particular subject"

The target in the development of an expert/knowledge-based system is to acquire and represent the knowledge and experience of a person(s) who have been identified as possessing the special skill or mastery.

An expert/knowledge-based system is created to solve problems in a particular narrow domain of expertise. The sense of this sentence restricts the term expert to a particular subject. Specific problem domain leads to successful expert/knowledge-based systems development.

2.3.1. Structure of Knowledge-Bases System

There are two main components of KBS: the Knowledge-Base and the reasoning mechanism.

The Knowledge-Base contains domain knowledge, normally provided by human experts and very specialized for a Particular problem domain, in which domain knowledge, knowledge about knowledge, factual data, procedural rules, business heuristics are available.

The inference engine infers new knowledge and utilizes existing knowledge for decision-making and problem solving. The reasoning mechanism takes descriptions from the user about the problem to be solved, requests additional information from the user as needed, at the end interprets the Knowledge-Base to make inferences, draw conclusions, and ultimately give advice (Sajja, 2008).Abraham (2005) also mentioned, the inference engine must find the right facts, interpretations, and rules and assemble them correctly.

Pomykalski *et al* (1999) described, the above two components are contained in the kernel of the expert system. They are basic and the required components for all expert systems. These components are identified as a fact base, a rule base and an inference mechanism. The fact base and the rule base combine to be the Knowledge-Base for the kernel.

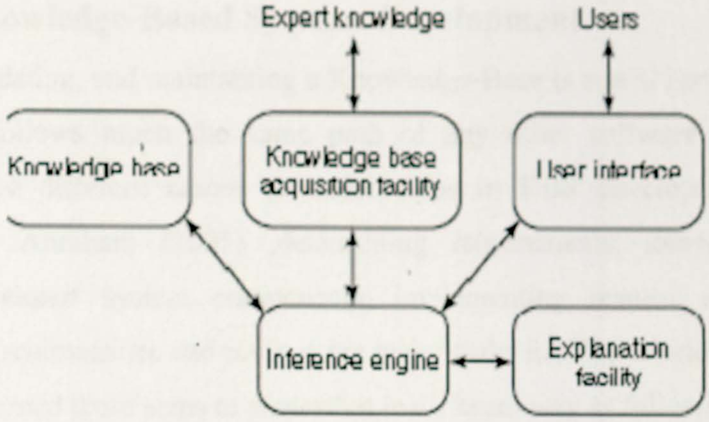


Fig. 2.1 structure of Knowledge-Bases system adopted from (Ajith, 2005)

Sajja(2008) and Pomykalski *et al* (1999) said, the "Knowledge-Base acquisition" facility process is used by the expert system to acquire new facts and rules associated with its specific domain. It is through this process that capabilities can be added to or subtracted from the expert system. Associated with this process is the concept of knowledge engineering. This is the process whereby knowledge from an expert or group of experts or other sources such as books, procedure manuals, training guides, etc. are gathered, formatted, verified and validated, and input into the Knowledge-Base of the expert system, The overall purpose of the knowledge acquisition facility is to provide a convenient and efficient means for capturing and storing all components of the Knowledge-Base.

The "user interface" is used to make communication between users and the Knowledge-Bases system in a more friendly way. Very often specialized user interface software is used for designing, updating, and using expert systems. The overall purpose of the user interface is to ease use of the expert system for developers, users, and administrators (Abraham, 2005).

The "explanation" is used by the expert system to provide to the user a reasoned history of its actions and/or recommendations. It allows a user to understand how the expert system arrived at certain results or provide justification for the decision taken.

2.3.2. Knowledge-Based Systems Development

Building, validating, and maintaining a Knowledge-Base is a skill (art) called knowledge engineering follows much the same path of any other software product. Different literatures give different names to major steps in KBS development. For example according to Abraham (2005) ,determining requirements, knowledge acquisition, constructing expert system components, implementing results, and formulating a procedure for maintenance and review are major tasks in KBS development. Pomykalski *et al* (1999) named these steps to somewhat in different way as follows:

- Problem Selection
- Knowledge Acquisition
- Knowledge Representation
- Implementation
- Testing, Verification, Validation, Evaluation
- Maintenance.

Detailed activities of each steps in both literatures is much similar, it is just a mater of naming. Majority of this section discussion took from Pomykalski *et al* (1999) and Abraham (2005).

Problem selection

In software development and scientific research, the most critical step is choosing the problem. Especially in the area of knowledge engineering, problem selection is critical. Finding a problem of the proper scope is especially imprint in KBS development, since KBS solve problems in a "particular narrow domain." If the domain is too large acquisition of the proper knowledge becomes an overwhelming task, if the domain is too small, the solution looks trivial. A good problem to solve is one that is cognitive in nature and sufficiently complex, has the support of management and users. Another critical factor in the development of a KBS is having an expert to work with the knowledge engineering team. The expert must be cooperative, articulate, and considered knowledgeable by others in the company. Involvement of domain area personnel is also critical for successful development of KBS. It is essential to have the support of the people (users and managers) for whom the system is being developed. Domain area personnel have to be involved at every step of the development process. By involving the users and managers in the process, we can significantly increase the chances of the final system being a product that is useful and potentially cost-effective for the organization.

Knowledge Acquisition

Knowledge acquisition is the most important process in the development of expert system. To get knowledge into a computer program we must acquire it from some source. There are two major sources of knowledge is KBS development: experts and documents. Both sources have advantages and disadvantages, although human experts are almost always preferred. Experts tend to be more current and have a broader range of knowledge than documents. They also can respond to questions and provide different sets of examples. However, their time is expensive and unless they support the project, they can work against the goals of the expert systems development. In some cases, expertise may have been lost, and the developer must rely on documents. Documents are generally cheaper to acquire and use. However, they typically have limited amounts of information and what they have is not always completely relevant.

Knowledge Representation

The third phase in KBS development is knowledge representation. The major objective in this phase is to take the acquired knowledge and translate it into machine-readable form. There are many different methods of knowledge representation in KBS development, the two most popular once to represent knowledge are rules and cases.

Implementation

It is, actually taking the knowledge and putting into some computer code. This can be accomplished three different ways: using a conventional programming language (BASIC, Pascal, C, C+, Java), using a programming language design for Artificial Intelligence programs (PROLOG or LISP), or using a KBS programming environment known as a shell (jCOLIBRI) that allow non-programmers to implement KBS applications.

Testing, Verification, Validation, Evaluation

An important component of any software development effort is the testing and evaluation of the software system (solution) to ensure correctness of the outputs and user satisfaction with the product in solving the given problem. Since knowledge-based systems are software solutions to problems then the importance of testing and evaluation cannot be minimized.

Maintenance

Maintenance is often a major issue for any software system. Much of the knowledge in a KBS is (or potentially can be) changing constantly and these knowledge units need to be updated. This problem of maintaining an evolving Knowledge-Base has been referred to *sustaining* the Knowledge-Base rather than maintaining. Sustenance of a KBS requires a steady upkeep of the cases (or whatever knowledge representation mode is used). Care must be taken in order to ensure that the total knowledge of the Knowledge-Base has been upgrade and not degraded by changes.

2.3.3. Knowledge-Base System Designing Approaches

Different approaches have been used to develop a KBS. Among those different approaches, RBR and CBR are the most familiar and well known approaches.

Rule-based approach

Conventional rule-based expert systems use human expert knowledge to solve real-world problems that normally would require human intelligence. Expert knowledge is often represented in the form of rules or as data within the computer (Abraham, 2005). In a rule-base Knowledge-Based system the knowledge of the domain is captured (represented) by production rules.

A rule-based system used of if-then rules to encode the knowledge into the system. These if-then rule statements are used to formulate the conditional statements that comprise the complete Knowledge-Base. A single if-then rule assumes the form 'if x is A then y is B' and the if-part of the rule 'x is A' is called the antecedent or premise, while the then-part of the rule 'y is B' is called the consequent or conclusion.

2.4. Case-Based Reasoning (CBR) Approach

2.4.1. Historical Background

The field of Case-based Reasoning (CBR) has a relatively young history and has its origin in research being done in cognitive science. The earliest contributions in this area were from Roger Shank and his colleagues at Yale University (Lützelschwab, 2007). They proposed that our general knowledge about situations is recorded as *scripts* that allow us to set up expectations and perform inferences. Scripts were proposed as a structure for conceptual memory describing information about stereotypical events such as, going to a restaurant or visiting a doctor. However, experiments on scripts showed that they were not a complete theory of memory representation - people often confused events that had similar scripts (Watson and Marir, 1994).

As mentioned by Cordier (2009), the multiplicity and wide diversity of scripts make it difficult to organize them as a structured memory. Shank thus proposes MOPs (Memory Organization Packets) which enable the organization of scripts into packets to store and find them more easily. It is then possible, via the MOPs, to generalize and specialize the scripts and thus to organize them hierarchically. The scripts existing in MOPs can be remembered and adapted to be reused in any situation. The notion of adaptation is very important, since it is rare to find a script which fits perfectly a given situation.

Other trails into the CBR field has come from the study of analogical reasoning, and - further back - from theories of concept formation, problem solving and experiential learning within philosophy and psychology. For example, Wittgenstein observed that 'natural concepts', i.e. concepts that are part of the natural world - such as bird, orange, chair, car, etc. - are polymorphic. That is, their instances may be categorized in a variety of ways, and it is not possible to come up with a useful classical definition, in terms of a set of necessary and sufficient features, for such concepts. An answer to this problem is to represent a concept extensionally, defined by its set of instances - or cases (Aamodt and Plaza, 1994).

Again Aamodt and Plaza (1994) described that; the first system that might be called a case-based reasoner was the CYRUS system, developed by Janet Kolodner, at Yale University. CYRUS was based on Shank's dynamic memory model and MOP theory of problem solving and learning. The case memory model developed for this system has later served as basis for several other case-based reasoning systems (including MEDIATOR, PERSUADER, CHEF, JULIA, CASEY).

As mentioned in Lützelschwab (2007), Various CBR workshops were organized in 1988, 1989, and 1991 by the U.S. Defense Advanced Research Projects Agency (DARPA). These events are considered to have formally created the discipline of Case-based Reasoning. In 1993, the first European workshop on Case-based Reasoning (EWCBR-93) was held in Kaiserslauten, Germany. Since then, many international workshops and conferences on CBR have been held in different parts of the world. Other major artificial intelligence conferences, such as ECAI (European Conference on Artificial Intelligence), IJCAI (International Joint Conference on Artificial Intelligence), have also had CBR workshops as part of their regular programs.

After these critical events, the fundamental principles have been more clearly defined and numerous industrial and experimental applications have demonstrated its usefulness. However, many perspectives for research are still open, with regard to implications in cognitive sciences and also possible applications in artificial intelligence (Cordier, 2009).

2.4.2. Pervious CBR Systems in Medicine

The application of IT research and development to support health and medicine is an emerging research area with significant potential. Major initiatives to improve the quality, accuracy and timeliness of healthcare data and information delivery are emerging all over the world. Since the advent of artificial intelligence in the 1970's which saw the birth of expert systems, various domains have taken advantage of this technology. The most popular application has been in the area of health and medicine. MYCIN, developed in 1970 at the Stanford University, is one of the most popular medical expert system used to assist to diagnose and treat blood diseases. MYCIN was the pioneer in demonstrating how a system can be used to successfully perform medical diagnosis. Another Early expert system is Patient Care Expert System (PACE) which was conceived in 1977 with the purpose being to make "intelligent selections" from the overwhelming and ever changing information related to health in order to facilitate patient care (Masizana-Katongo*et al*, 2009).

In the late 1980's, followed by ground laying work done by Koton, and Bareiss, CBR appeared as an interesting alternative for building medical AI applications, and has since been further established in the field. Certainly, one of the intuitively attractive features of CBR in medicine is that the concept of patient and disease lends itself naturally to a case representation (Nilsson and Sollenborn, 2004).

In medicine, CBR has mainly been applied for diagnostic and partly for therapeutic tasks. One of the earliest medical expert systems that used CBR techniques is CASEY. CASEY diagnosed heart failure patients by comparing them to earlier patients whose diagnoses were known. It incorporated and extended an earlier model-based system, which diagnosed heart failures based on a physiologic model of the human heart. A diagnosis, in CASEY, was represented as a graph, showing the causality among the different symptoms and features of a patient (Bichindaritz and Marling, 2006). CASEY stores a large amount of information in its cases. In addition to all observed features, it retains the causal explanation for the diagnosis found, as well as the list of states in the heart failure model for which there was evidence in the patient (Amadot and Plaza, 1994).

PROTOS was another early CBR system, which diagnosed audio-logical disorders. PROTOS incorporated the knowledge of an expert audiologist and approximately 200 actual patient cases. A diagnosis in PROTOS was a categorical label, assigning a new patient to one of a number of pre-specified diagnostic categories. A new patient was effectively classified as having the same diagnosis as other similar past patients (Bichindaritz and Marling, 2006).

There are other early CBR influential systems in the health sciences (Bichindaritz and Marling, 2006). MEDIC diagnoses pulmonary disease, FLORENCE assists with nursing diagnosis, prognosis and prescription; ALEXIA determines a patient's hypertension etiology, ROENTGEN helps to design radiation therapy plans, MacRad helps to interpret radiological images, HPISIS diagnoses degenerative brain diseases, MNAOMIA supports diagnosis, treatment and research for psychiatric eating disorders and ICONS advises physicians about antibiotics to prescribe for patients with bacterial infections.

2.4.3. CBR in Diabetes Management

The first research project to investigate CBR for diabetes management was the Telematic Management of Insulin-Dependent Diabetes Mellitus (T-IDDM) project. The goals of this project were to support physicians in providing appropriate treatment for maintaining blood glucose control, provide remote patients with tele- monitoring and tele-consultation services, and provide cost-effective monitoring of large numbers of patients and support patient education (Marling *et al,* 2007). Both research and the practical application of CBR in the health sciences are currently experiencing rapid growth and development.

2.4.4. CBR General Overview

Case-based reasoning (CBR) is a paradigm of problem-solving which uses past experiences to solve new problems. Case-based reasoning is a paradigm of artificial intelligence, like neural networks, rule-based reasoning, genetic algorithms, multi-agent systems, Bayesian networks, etc. (Cordier, 2009). In the early 90s, CBR appeared as an interesting alternative to the rule-based systems which were beginning to show their limits. Especially appropriate when the number of rules needed to capture an expert's knowledge is unmanageable or when the domain theory is too weak or incomplete (Lopez, 2001).

CBR solves new problem by remembering previous similar situation and also reuse information and knowledge of that situation. In CBR, new problem are often similar to previously encountered problem and the current solution mostly based on past solution. In CBR, experiences are stored in cases. The case-based on pervious experience is that it can be used for future problems solving, and can referred to past case, stored or retained case (Iqbal and Ashraf, 2006). Describe reasoning by reusing past cases is the most strong and common applied way to find or to solve problem of human.

CBR is used when generalized knowledge is lacking. The method works on a set of cases formerly processed and stored in a case base. A new case is interpreted by searching for similar cases in the case base. Among this set of similar cases the closest case with its associated result is selected and presented as the output (Perner, 2008).

As stated by Pacharavanich *et al* (2000), the quality of case-based reasoning's solutions depends on three fundamental factors. The first one is the number of well-defined cases stored in the system. The other one is the ability of the system to recall experiences by using an index and to interpret the new situation in terms of those experiences. Adaptation of an old solution to meet the demands of a new situation is another factor. We will see each of them in later sections.

As compared with RBR, CBR has several advantages. Cordier (2009), Iqbal and Ashraf (2006) and Pacharavanich *et al* (2000) found that the following are rewards of CBR.

- **Reducing the knowledge acquisition task**

  The knowledge acquisition task of CBR consists of collection of relative past cases, and their representation and storage. CBR tries to model human expert reasoning in a more natural way by collecting cases of problem solving experiences. In the rule-based systems, knowledge acquisition is necessary by extracting a set of rules.

- **Avoiding repeating mistakes made in the past**

  In CBR, system failure, system success, as well as reason to system failures are recorded. This information about cause of the failure is used in the future for reducing the failure outputs. It captures and indexes its past mistakes. Moreover, it provides a warning to the reasoner so that he/she can avoid those past failures.

Reasoning in domains that have not been fully understood, defined or modeled
CBR system can still be developed by only adding small set of causes from the domain, in situation where too insufficient knowledge exists to build a causal model.

Providing flexibility in knowledge modeling
CBR systems used past experience as domain knowledge and can provide reasonable solution, appropriate solution through adaptation. But, modeled base like rule-based systems, due to their rigidity in the problem formulation and modeling, they can not solve problems when they encounter some missed data.

Making prediction of the probable success of proffered solution
In CBR, the past solution for a case is stored in the case-base for future use. Based on that stored information case-based reasoner may be able to predict success of solution for current problem.

- **Learning over time**
  CBR is competent over time. The level of success tested and determined in the real world, these solutions can be added to case bases to solve the future problems.

- **Reasoning with incomplete or imprecise data of concept**
  The retrieved cases are not may be very similar solution to the current case. When they are within some defined potential of similarity to the present case any drought and incompleteness can be deal by Case-Based reasoner. These factors have a little impact on performance, because the increasing lack of similarity between the current and retrieved cases.

Avoiding repeating all the steps that need to be taken to arrive at a solution

Case-based reasoning is an efficient reasoner. It solves problems by adapting old solutions without any need to derive answers from scratch each time

Providing a means of explanation

CBR systems can explain solution to user by explaining how pervious case was successful in that situation, by using similarities between the cases. The case-based reasoning explanation facilities allow the system to explain why the current problem is similar to or different from a set of cases drawn from its database. These explanations are important for building users' confidence and for helping novices learn from past experiences.

Extending to a broad range of domains

CBR can be applied on various numbers of application domains. This is due to the appearing of unlimited number of ways representing, indexing, adapting and retrieving cases.

Maintaining

Maintaining CBR system is easier than rule-based system since adding new knowledge can be as simple as adding a new case.

2.4.5. The CBR Process

In CBR terminology, a *case* usually denotes a *problem situation.* A previously experienced situation, which has been captured and learned in a way that it can be reused in the solving of future problems, is referred to as a past case, previous case, stored case, or retained case. Correspondingly, a new case or unsolved case is the description of a new problem to be solved. Case-based reasoning is a cyclic and integrated process of solving a problem, learning from this experience, solving a new problem, etc (Aamodt and Plaza, 1994).

The principle of CBR, reusing a past problem-solving experience to solve a similar problem, is simple, but the implementation of this principle remains complex and raises a certain number of questions. How do we represent an experience? What is a similar problem? How do we reuse an experience and adapt it to the present situation? What can be retained from a specific problem-solving experience? In order to answer these questions there are knowledge engineering tasks and issues which are crucial in developing the CBR-based systems. (Aamodt and Plaza, 1994) have described CBR typically as a cyclical process comprising the four REs:

RETRIEVE the most similar case

REUSE the case to attempt to solve the problem

REVISE the proposed solution if necessary, and

RETAIN the new solution as a part of a new case

A new problem is matched against cases in the case-base and one or more similar cases are *retrieved.* A solution suggested by the matching cases is then *reused* and tested for success. Unless the retrieved case is a close match the solution will probably have to be *revised* producing a new case that can be *retained*

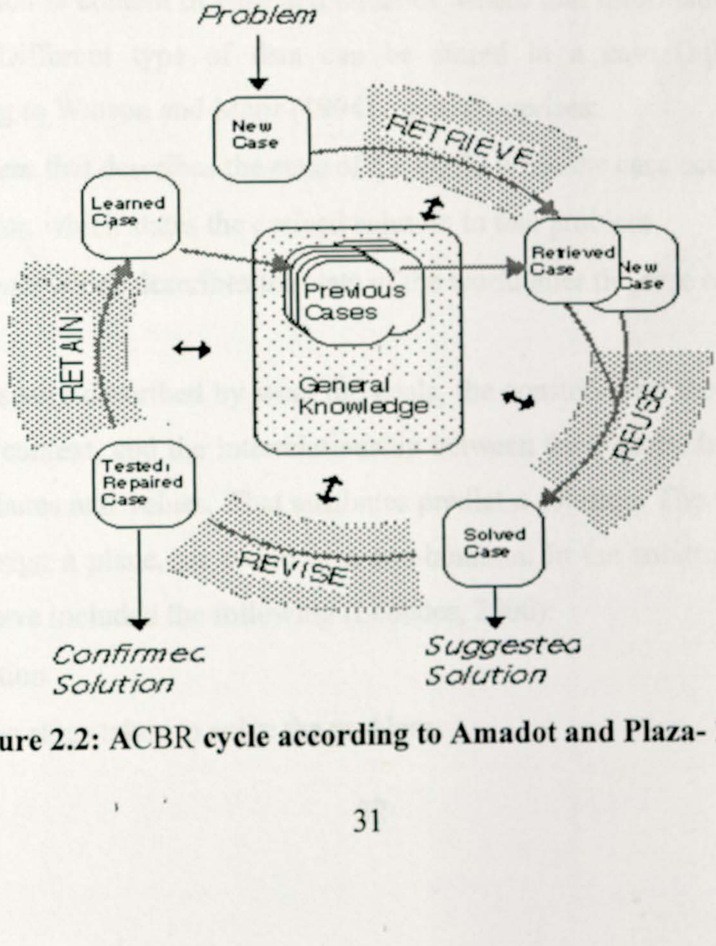


Figure 2.2: ACBR **cycle according to Amadot and Plaza- 1994**

An initial description of a problem (top of figure) defines a new case. This new case is used to RETRIEVE a case from the collection of previous cases. The retrieved case is combined with the new case - through REUSE - into a solved case, i.e. a proposed solution to the initial problem. Through the REVISE process this solution is tested for success, e.g. by being applied to the real world environment, and repaired if failed. During RETAIN, useful experience is retained for future reuse, and the case-base is updated by a new *learned case*, or by modification of some existing cases.

This cycle currently rarely occurs without human intervention. For example many CBR tools act primarily as case retrieval and reuse systems. Case revision (adaptation) is often being undertaken by managers of the case base. However, it should not be viewed as weakness of CBR that it encourages human collaboration in decision support (Watson and Marir, 1994). The following sections will outline how each process in the cycle can be handled.

2.4.5.1. Case representation

A case is a contextualized piece of knowledge representing an experience. It contains the information which is content of case and situation where that information or experience can be used. Different type of data can be stored in a case (Iqbal and Ashraf, 2006).According to Watson and Marir (1994) a case comprises:

- the ***problem*** that describes the state of the world when the case occurred
- the ***solution*** which states the derived solution to that problem
- the ***outcome*** which describes the state of the world after the case occurred

A Problem is usually described by what the goals, the constraint on the goal, the feature of the problem context, and the interrelationship between these parts. In short it consists of a set of attributes and values. That attributes predict a solution. The solution can be a diagnosis, a design a plane, an action or a combination. In the solution part of a case, CBR systems have included the following (Leondes, 2000):

- the solution
- reasoning steps taken to solve the problem

- justification for decision made
- alternative solutions that were not used
- unacceptable solutions that were rejected
- Exceptions of what would happen if the solution were to be applied.

Depending on what is included in a case, the case can be used for a variety of purposes. Cases which comprise problems and their solutions can be used to derive solutions to new problems. Cases comprising problems and outcomes can be used to evaluate new situations. If, in addition, such cases contain solutions they can be used in evaluating proposed solutions and anticipating potential problems before they occur (Salem, 2007).

There is a lack of consensus within the CBR community as to exactly what information should be in a case. But, a case should represent two things: functionality of information and the easiest way in which information is obtained in case (Watson and Marir, 1994). Iqbal and Ashraf (2006) noticed that, whenever we want to choose representation format of a case, we have to keep in mind certain factors

- Language or shell in which we have to implement CBR system. Selection of shell may reduce number of formats for case representation.
- Indexing and search mechanism. Format of case should be selected according to search mechanism. Case format should be able to interact with mechanism easily
- Type or structured associated with content. These types must be available in case representation

In addition, when, we want to choose format for case representation. Cases also have to be structured for efficient case retrieval. There are two types of structures (memory models. Structures types are Common Structure and Hierarchal Structure. The Hierarchal Structure (Amadot and Plaza- 1994) called it called 'episodic memory organization. The basic idea is to organize specific cases which share similar properties under a more general structure (a generalized episode). Memory model for chosen form of case representation depend on following factors:

- Representation used in case base.
- Number of features that are used to match cases during search.
- Number and complexity of cases.

2.4.5.2. Case Indexing

The CBR system derives its power from its ability to retrieve relevant cases quickly and accurately from its memory. Index is computational data structure can be held in memory. Figuring out when a case should be selected for retrieval in similar future situations is the goal of the *case indexing.* Case indexing involves assigning indices to cases to facilitate their retrieval (Salem, 2007).

A case description contains the features of the problem that were significant in reaching the solution and in explaining why a solution was reached. These features of the case are used for indexing, which will allow the CBR system to retrieve cases from memory. Not all features are used in indexing for retrieval but may provide contextual information useful to the user (Leondes, 2000). Hence, according to Iqbal and Ashraf (2006) information in a case can be two types:

1. Indexed information used for retrieval.
2. Unneeded information that may provide information to user but not used in retrieval.

For instance in medical systems, where patient's age, sex, height and weight can be uses as index features. That information is helpful for future retrieval. Patient's photograph can be includes as an unneeded feature. That can't be use in retrieval. Picture may be helpful for doctor to remind patient.

Indices should:

- be predictive
- address the purposes the case will be used for
- be abstract enough to allow for widening the future use of the case-base
- be concrete enough to be recognized in future

In general, the case indices should link the case with its purpose, should not be too specific or too general, should expand to include new cases, and should help predict new information. If the indices are not specific enough, too many cases will be retrieved. On the other hand, if the indices are too narrow, cases that could solve the problem may be overlooked. An example of a predictive feature is medical CBR system that predicts heart disease risk by looking at family histories. This feature may be used to predict that the patient is at a higher risk for a particular risk. It is important to note that the indices chosen for a case-base depend on the problem solving context (Leondes, 2000).

Both manual and automated methods have been used to select indices. Choosing indices manually involves deciding a case's purpose with respect to the aims of the reasoner and deciding under what circumstances the case will be useful (Iqbal and Ashraf, 2006).As described by Watson and Marir (1994) there are an ever increasing number of automated indexing methods including

Indexing cases by features and dimensions that tend to be predictive across the entire domain i.e., descriptors of the case which are responsible for solving it or which influence its outcome. In this method the domain is analyzed and the dimensions that tend to be important are computed. These are put in a checklist and all cases are indexed by their values along these dimensions. For example, MEDIATOR uses this method by indexing on type and function of disputed objects and relationship of disputants, whilst CHEF indexes on texture and taste. This kind of technique is called *checklist-based* indexing.

Difference-based indexing selects indices that differentiate a case from other cases. During this process the system discovers which features of a case differentiate it from other similar cases, choosing as indices those features that differentiate cases best.

Similarity and explanation-based generalization methods, which produce an appropriate set of indices for ***abstract cases*** created from cases that share some common set of features, whilst the unshared features are used as indices to the original cases

In inductive learning method, features are identified which latterly uses as an indexes. These techniques are widely used (e.g., in Cognitive system's ReMind) and commonly use variants of the ID3 algorithm used for rule induction.

Explanation-based techniques, which determine relevant features for each case. This method analyses each case to find which of their features predictive ones are. Cases are then indexed by those features.

Despite the success of many automated methods, people tend to do better at choosing indices than algorithms, and therefore for practical applications indices should be chosen by hand.

Good indexing is not enough when the case memory is large. Good organization of the memory is necessary because a simple linear organization, like a list, is very inefficient for retrieval purposes (Lopez, 2001).One of the representation problems in CBR is deciding how the case memory should be organized and indexed for effective retrieval and reuse. This leads us to the problem of problem of how to integrate the case memory structure into a model of general domain knowledge (case storage).

Case storage is an important aspect in designing efficient CBR systems in that, it should reflect the conceptual view of what is represented in the case and take into account the indices that characterize the case. The case-base should be organized into a manageable structure that supports efficient search and retrieval methods. Building a structure that will return the most appropriate case is the goal of the case memory organization process (Watson and Marir, 1994).

A balance has to be found between storing methods that preserve the semantic richness of cases and their indices and methods that simplify the access and retrieval of relevant cases. These methods are usually referred to as case memory models. The two most influential case memory models are the dynamic memory model of Schank and Kolodner, and the category-exemplar model of Porter and Bareiss (Amadot and Plaza, 1994).

2.4.5.3. The Dynamic Memory Model

The case memory in this model is a hierarchical structure of what is called 'episodic memory organization packets' (E-MOPs), also referred to as generalized episodes (GE) developed from Schank's more general MOP theory. The first system that may be referred to as a case-based reasoner was CYRUS, based on dynamic memory model (Amadot and Plaza, 1994).

According to (Watson and Marir, 1994) and (Amadot and Plaza, 1994), MOPs are a form of frame and are the basic unit in dynamic memory. They can be used to represent knowledge about classes of events using two kinds of MOPs:

- instances representing cases, events or objects, and
- abstractions representing generalized versions of instances or of other abstractions

The basic idea is to organize specific cases which share similar properties under a more general structure (i.e., a generalized episode). A GE contains three different types of objects: *norms*, *cases* and *indices*. Norms are features common to all cases indexed under a GE. Indices are features which discriminate between a GE's cases. An index may point to a more specific generalized episode or to a case, and is composed of an index name and an index value.

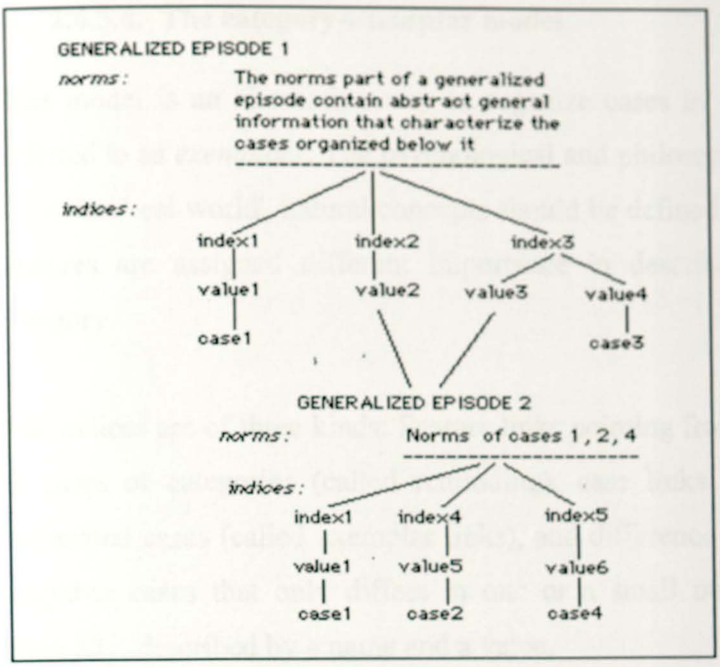


Figure 2.3: Structure of cases and generalized episodes adopted from Amadot and Plaza-1994

The case-memory is a discrimination network where nodes are either a GE, an index name, index value or a case. Index name-value pairs point from a GE to another GE or case. The primary role of a GE is as an indexing structure for storing, matching and retrieval of cases. During case storage when a feature (i.e., index name and index value) of a new case matches a feature of an existing case a new GE is created. The two cases are then discriminated by indexing them under different indices below the new GE (assuming the cases are not identical). Thus, the memory is *dynamic* in that similar parts of two cases are dynamically generalized into a new GE, the cases being indexed under the GE by their differences.

However, this process can lead to an explosive growth in the number of indices as case numbers increase. So for practical purposes most CBR systems using this method limit the number of permissible indices to a limited vocabulary.

2.4.5.4. The category-exemplar model

This model is an alternative way to organize cases in a case memory. Cases are also referred to as *exemplars*. The psychological and philosophical basis of this method is the view that 'real world', natural concepts should be defined extensionally. Further, different features are assigned different importance in describing a case's membership to a category.

The indices are of three kinds: Feature links pointing from problem descriptors (features) to cases or categories (called reminding), case links pointing from categories to its associated cases (called exemplar links), and difference links pointing from cases to the neighbor cases that only differs in one or a small number of features. A feature is, generally, described by a name and a value.

A feature is described by a name-value pair. A category's exemplars are stored according to their degree of prototypically to the category. Within this memory organization, the categories are inter-linked within a semantic network containing the features and intermediate states referred to by other terms. This network represents a background of general domain knowledge that enables explanatory support to some CBR tasks. A new case is stored by searching for a matching case and by establishing the relevant feature indices. If a case is found with only minor differences to the new case, the new case may not be retained, or the two cases may be merged.

2.4.5.5. Case Retrieval

Reminding is the most important component of reasoning though cases; to reason a person or computer must be reminded of the appropriate case at the right time. To learn, a new experience must be integrated into memory, and, possibly, old experience must be reevaluated and re-indexed. The recall of case in CBR is called retrieval (Leondes, 2000). The most basic problems in CBR are the retrieval and selection of cases since the remaining operations of adaptation and evaluation will succeed only if the past cases are the relevant (Lopez, 2001). Case retrieval is a process of finding cases which are closest to current case. Given a description of a problem, a retrieval algorithm, using the indices in the case-memory, should retrieve the most similar cases to the current problem or situation. The retrieval algorithm relies on the indices and the organization of the memory to search to a set of cases that are sufficiently similar to the new case - given a similarity threshold of some kind, and the selection task works on this set of cases and chooses the best match (Watson and Marir, 1994).

During retrial, each case must be compared with the current problem and be assigned a degree of similarity. Then the retrieving program will select the cases with the highest degree of similarity. To retrieve a case from memory, a CBR system must decide whether it is the most appropriate one for the current situation. This is called similarity assessment.

2.4.5.5.1. Similarity assessment

Structural similarity

The simplest method for computing similarity is to look at structural similarity between the current problem and a case. Symbolic values are index values that are textual and not numerical. The simplest definition of similarity between symbolic values is to demand an exact match (Leondes, 2000). Though computationally expensive because it relies on extensive use of domain knowledge, retrieval based on structural similarity has the advantage that more relevant cases may be retrieved (López*et al*, 2005).

The biggest problem here is to determine the weights of the features. The weights assigned to case attributes allow them to have varying degrees of importance and may be selected by a domain expert or user. This weight indicates the significances of the feature to retrieval. After we determine the weight of each feature, we need to compute a similarity value for the whole case. Usually this is done with nearest neighbor method.

$$\frac{\sum_{i=1}^{n} w_i \times sim\left(f_i^I, f_i^R\right)}{\sum_{i=1}^{n} w_i}$$

A Nearest Neighbor Algorithm adopted from-(Watson and Marir, 1994)

Where w is the importance weighting of a feature, *sim* is the similarity function, and *fI* and *fR* are the values for feature *i* in the input and retrieved cases respectively.

The limitations of this approach include problems in converging on the correct solution and retrieval times. In general the use of this method leads to the retrieval time increasing linearly with the number of cases. Sequentially processing all cases in memory has complexity O(n), where n is the number of cases. This may not be an acceptable overhead if n is very large. Therefore this approach is more effective when the case-base is relatively small (Watson and Marir, 1994).

Functional similarity

Leondes (2000) explained, Structural similarities and dissimilarities are not relevant in determining actual case similarity. In such cases we need to look at functional similarity, that is whether the old case can solve the current problem, rather than whether it look like the current problem

The following table shows as how functional similarity is work. This example is adapted from CASEY's functional similarity assessment function. CASEY used functional similarity, not rely on structural similarity, but needed a deep model of causal relation of heart disease.

	Old case	**Current problem**
Age	72	65
Gender	Male	Male
Pulse rate	96	90
Pulse	Normal	Slow rise
Temperature	98.7	98.4
Chest pain	Angina	Angina
Calcification	None	Mitral & aortic
cough	Ascent	Ascent

Table 2.1: functional similarity assessment taken from CASEY, adopted from (Leondes, 2000).

Looking at the diagnosis of the old case, CASEY noticed that the patient was diagnosed with aortic valve disease. CASEY also noticed that aortic valve disease may have aortic and mitral calcification and a slow rising pulse. So, although the values for the features pulse and classification do not match, they can be supported by the solution of the old case, and they qualitatively match.

2.4.5.6. Case adaptation/ reuse

The process of changing the old, retrieved solution to fit the current problem is called *adaptation* (Leondes, 2000). Once a matching case is retrieved a CBR system should adapt the solution stored in the retrieved case to the needs of the current case. Adaptation looks for prominent differences between the retrieved case and the current case and then applies formulae or rules that take those differences into account when suggesting a solution (Watson and Marir, 1994). Usually adaptation involves deleting, adding or substituting something from the old case to make it into an appropriate solution for the new input problem case.

As explained by Lopez (2001), the reuse of the retrieved case solution in the context of the new case focuses on two aspects:

1. the differences among the past and the current case
2. What part of a retrieved case can be transferred to the new case?

Watson and Marir (1994) added that, in general, there are two kinds of adaptation in CBR:

1. Structural adaptation, in which adaptation rules are applied directly to the solution stored in case.
2. Derivational adaptation, which reuses the algorithms, methods or rules that generated the original solution to produce a new solution to the current problem. In this method the planning sequence that constructed that original solution must be stored in memory along with the solution. Derivational adaptation, sometimes referred to a re-instantiation, can only be used for cases that are well understood.

Lópezet *al* (2005) added one more kind of adaptation; transformation adaptation alters the structure of the old case solution. It involves using general knowledge to guide the changing of a stored case to solve a new problem. Several techniques, ranging from simple to complex, have been used in CBR for adaptation. (Watson and Marir, 1994) explained it in detail as follows:

Null Adaptation: It uses no adaptation at all. It just applies whatever solution is retrieved to current problem without adapting it. Null adaptation is useful for problems which involve complex reasoning.

Parameter adjustment: It is a structured technique which compares specified parameters of retrieve and current case to give a solution in the right direction. This technique used in CBR called JUDGE.

Abstraction and re-specialization: general structural adaptation technique that is used in a basic way to achieve simple adaptations and in a complex way to generate novel, creative solutions.

Critic-based adaptation: *in* which a critic looks for combinations of features that can cause a problem in a solution. Importantly, the critic is aware of repairs for these problems.

2.4.6. CBR Software Tools

CBR tool is software that can be used to develop applications that follow CBR Framework. Some scholars argue that the current surge in interest in CBR is due to the intuitive nature of CBR and because it may closely resemble human reasoning. Software vendors might argue that it is because CBR tools have made the theory practically feasible. There is truth in both views but certainly the tools have made a contribution (Watson and Marir, 1994).

Several commercial companies offer software tools for building CBR systems (Amadot and Plaza, 1994). Ideal CBR tool should support the main processes of CBR. Processes are Representation, Retention, Retrieval and adaptation (Iqbal and Ashraf, 2009).

Developing tool for CBR system should perform the following function in each processes of CBR development

Representation: It should support all data types and should structure cases relevant to the application domain.

Retention: The Case-Base should be organized in a manageable way which supports search and retrieval method.

Retrieval: Cases must be properly indexed for efficient retrieval. It may be automatic but developers can influence the processes.

Adaptation: Arrange it in advance within tool for case adaptation.

With these functionalities (Watson and Marir, 1994) mentioned number of CBR tools as follows.

CBR-Express: produced by inference corporation. CBR-Express has a simple case structure and uses nearest neighbor matching to retrieve cases. CBR-Express is extremely well suited to help desk applications and has also been used successfully for intelligent task assistance, information access systems and knowledge publishing. It is very easy to use, reliable, network ready, and notable for its intelligent handling of text.

Case Point: Case Point, also from Inference Corporation, is a runtime version of CBR-Express. Case Point is a CBR delivery system as it only runs case-bases (i.e., users of Case Point cannot develop or edit cases, for this they must use CBR-Express) and does not contain the customer tracking facilities. Case Point is also extremely memory efficient, only requiring a few hundred K to run.

ART*Enterpris: is the latest tool of Inference Corporation's. That company is one of oldest AI Company. In 1980, ART was advertised as an AI-based tool.

Case Power: this tool builds its cases in matrix environment provided by Microsoft Excel. Rows and columns of spread sheet are used to define cases and their attributes.

CASUEL: a Common Case Representation tool written in C developed by the INREC (INtegratedREasoning from CAses). INREC is a case-based reasoning project funded by the European Union.

As mentioned in (Abdrabou and Salem, 2008), In addition to the above there are CBR tools based on object-oriented Framework.

CBR*Tools: is an object-oriented framework for CBR which is specified with the Unified Modeling Language (UML) notation and written in Java. It offers a set of abstract classes to model the main concepts necessary to develop applications integrating case-based reasoning techniques: case, case base, index, measurements of similarity, reasoning control.

CAT-CBR: platform uses a library of CBR components to guide the user in the development of a CBR application. CAT-CBR has an interactive tool where users choose the components that need to be included in an application. This tool is built over a CBR system that guides and gives support to users during the configuration process.

jCOLIBRI :The application framework of JColibri comprises a hierarchy of JAVA classes plus a number of XML files.

Finally (Watson and Marir, 1994) classified CBR tools can be into two broad categories.

Domain independent or domain dedicated: These CBR shells that generate applications with graphical user interface, where some parameters can be defined by the user to develop a new application. CBR shells are a kind of tools which can be used by non-programmer user and cannot extend or integrate new components in these tools.

CBR Application Programming Interfaces (APIs): provide set of functions to deal with CBR algorithm. Sometimes CBR APIs by using a programming language can also be extended.

2.4.7. CBR Framework: jCOLIBRI

A framework is a set of classes that embodies an abstract design for solutions to a family of related problems. In other words, a framework is a partial design and implementation for an application in a given problem domain. Frameworks are a well-known technology related with software reuse that leverages the prior efforts of developers in order to avoid recreating and revalidating common solutions. (Juan et al, 2010). Main theme of a framework is to occupy a set of concepts related to a domain and the way they interact. It allows the reuse of code and design for a class of problems. It also gives the ability to non-expert to write complex applications quickly. The use of framework reduces the cost of application creation because using a framework is closer to maintaining an existing application than to developing a new one from scratch.

According to Iqbal and Ashraf (2009), in 1980, the concept of object-oriented frameworks has been introduced. An object-oriented framework is a reusable design of all or part of a system that is represented by a set of abstract classes and the way their instances interact. It is very important that the framework for CBR takes care of these axes which may be reused in different situations. Researchers are tried to work out and make research work how to make a new Framework to build a CBR systems. Researchers has developed **jCOLIBRI**, an object-oriented framework in Java for building CBR systems that greatly benefits from the reuse of previously developed CBR systems (Juan *et al*, 2010).

jCOLIBRI is an evolution of the COLIBRI architecture (Juan et al , 2010). In 2002, Belen Diaz-Agudo developed domain independent architecture called COLIBRI. It stands for Cases and Ontology Libraries Integration for Building Reasoning Infrastructures. COLIBRI helps in the design of Knowledge Intensive CBR systems. COLIBRI is basically based on knowledge gained from a library of ontology which are application independent and also use CBR Ontology. COLIBRI is very useful for certain time period but, it is not helpful to non-expert users. Then, CBR community thought to develop a new application –jCOLIBRI (Iqbal and Ashraf, 2009).

jCOLIBRI is an object-oriented framework in java for developing CBR systems . It offers an easier development process that is based on the reuse of past designs and implementations (Stoyanov et al, 2005). jCOLIBRI is a software artifact that promotes software reuse for building CBR systems, integrating the application of well proven Software Engineering techniques with a knowledge level description that separates the problem solving method, that defines the reasoning process, from the domain model (Recio-Garcia and Diaz-Agudo, 2004).

The motivation of the researcher for choosing this framework is based on a comparative analysis between jCOLIBRI tool and other tools. jCOLIBRI enhances the other CBR shells in several aspects: availability (open source framework), implementation (the Java implementation implies to a great extend usability, extensibility and user acceptance), GUI (the provided graphical tools facilitate the system design).

Figure 2.4 depicts the architecture of jCOLIBRI Framework. The framework is organized around the following elements: tasks and methods, case base, cases, and problem solving methods (Abdrabou and Salem, 2008).

Tasks and Methods: The tasks supported by the framework and the methods that solve them are all stored in a set of XML files.

Case Base: Different connectors are defined to support several types of case storages, from the file system to a database.

Cases: A number of interfaces and classes are included in the framework to provide an abstract representation of cases that support any type of actual case structure. jColibri represent cases in a very simple way.

Problem Solving Methods: The actual code that supports the methods included in the framework.

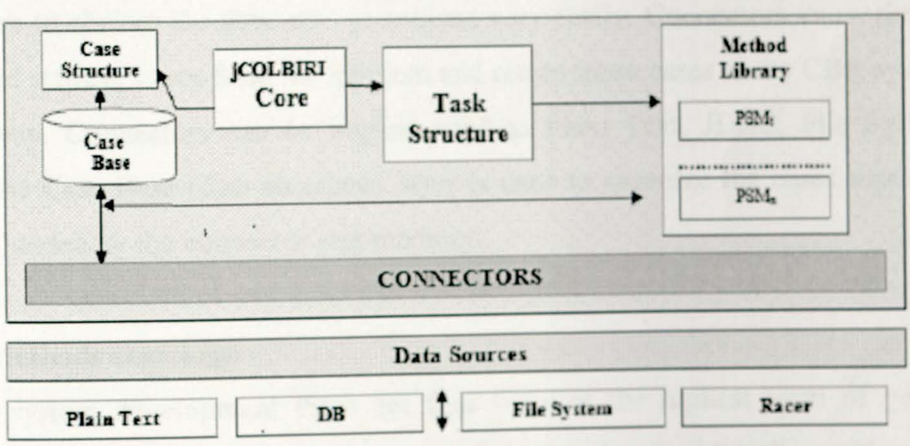


Figure 2.4: jCOLIBRI Architecture adopted from- ((Stoyanov et al, 2005).

jCOLIBRI also shows the four different data sources types (Persistence Layer), which are connected with the other framework components via objects, referred to as Connectors (Stoyanov *et al*, 2005).

Case Structure

jCOLIBRI fits to several case structures, from simple attribute value records to complex hierarchical trees with composed attribute. As we have discussed in section 2.3.4.1, a case mainly comprises problem and solution. According to (Stoyanov *et al*, 2005) problems and solution are sets of attributes, and there are two types of attributes: simple and compound. Simple Attributes have Name, Type, Weight and Local Similarity Function. Compound Attributes contained other simple attributes, allowing complex case structure. A local similarity function is used to compare two cases using simple attribute values. Global similarity functions are connected to compound attributes and are used to gather similarities of the collected attributes in a unique similarity value. Finally the similarity value of two cases is computed as the similarity of their problem and solution descriptions.

Case-base and Connectors

Case-base Management of jCOLIBRI splits into two layers: persistency mechanism and in-memory organization. Persistence layer consists of several connectors that allow developers to change the data storage sources very easily. Connectors know the way to access and retrieve cases from the medium and return those cases to the CBR system in a similar way. Connectors can be implemented as Plain Text, JDBC, File System and Racer. The Case Base (data structure) layer is used to organize the cases that are once read and loaded by the connector into memory.

Tasks /Methods Ontology

In CBR system development there are four tasks at the highest level of generality: Retrieve the most similar case/s, reuse its/their knowledge to solve the problem, Revise the proposed Solution and retain the experience. Each of the tasks has its own specific tasks.

jCOLIBRI Core

The Core is the critical component of the framework. It comprises the CBR configuration and executing the application. When a user develops a CBR application template a Java code is generated that configures the Core components with the appropriate tasks, methods, data types and case structures. The core contains elements: CBR State (maintains the tasks and method configuration, CBR Context (contains the Case-base and working cases), Packages (manage the remaining components, such as similarity functions, case structure.

CHAPTER THREE

RESEARCH DESIGN AND METHODOLOGY

3.1. Introduction

Research design provides the glue that holds the research roadmap. A research design is used to structure the research and to show how all of the major parts of the research work together.

3.2. Study Area

The focus of this research is about the treatment of type II DM at Dessie Referral Hospital. There is a separate department which took the overall responsibility of diabetes management staffed with three medical doctors and two nurses who took special training on diabetes diagnosis and treatment. Management of diabetes involves a considerable element of self-care and advice that should, therefore, be aligned with the perceived needs and preferences of people with diabetes. The opportunity is given to the patients to make informed decisions about their care and treatment. If patients do not have the capacity to make decisions, healthcare professionals follow the department health guidelines. The Hospital works in collaboration with Ethiopian diabetes association to provide free medications.

Dessie Hospital is a referral hospital in east Amahara region hence; physicians of the hospital are confronted with extremely various types of cases. Thus, the diabetes management department of the hospital was taken as a research area since of the hospital has vast amount of experience in education of patients and treatment of the disease.

3.3. Study Subjects

The study subjects to this research were type II DM patients who have follow-up at Dessie Referal Hospital.

3.4. Knowledge Sources

The knowledge that built up this research came from different sources. The primary knowledge sources were various experts including generalists, internists and nurses. In addition, the researcher directly observed dialogues between experts and DM patients. Secondary sources of knowledge including relevant literatures from all possible sources and formats including journal articles, guidelines for type II DM management, books, theses, and internet were reviewed.

3.5. Knowledge Acquisition Procedure

The knowledge was captured through unstructured interviews, discussions with individual experts, and document analysis (patient case cards). The reason for employing unstructured interview was because it helps to dig out complicated concepts from experts in a better way than the structured one. Unstructured interview enabled experts to describe their procedures and logics in a detailed manner.

Experts were usually busy; to maximize access to them and minimize interruptions, interviews and discussions were held after work hours and on weekends. Patient case cards stored at the hospital and guidelines to diabetes treatment were also used to capture the problem in detail.

3.6. Sampling Techniques

To capture domain knowledge, the researcher used expert sampling. According to Lisa (2008), expert sampling is a kind of purposive sampling in which the researcher is looking for individuals who have particular expertise that is most likely to be able to advance the researcher's interest and generate new ideas. Hence, the researcher selected one internist, three generalists as well as two nurses who have special expertise on diabetes treatment. These experts were selected based on their job position in the domain area.

On the other hand, convenient sampling was employed to select cases. Convenient sampling is non-probabilistic sampling where population elements are selected on the basis of their availability (Herek, 2009). Accordingly, the researcher collected 42 representative cases among diabetic cases treated at Dessie Referral Hospital diabetes treatment department.

3.7. Data Preparation for Analysis

Data preparation involves checking the data for accuracy, entering the data into the computer, transforming the data, and developing and documenting a database structure (Trochim, 2006).

After the concepts were collected from available sources, they were organized in a format comfortable to the development tool. Some cases were not complete and have various noises because of different reasons. For example, some physicians didn't write full history of patients, or a patient may lose his or her card number and registered as a new patient. Thus, data cleaning is performed in order to identify and eliminate errors in the course of case collection.

3.8. CBR Design

After the data is ready through data preparation the next task is modeling the problem. 42 type II DM cases were collected from Dessie Referral Hospital. The case-based was built by using these cases. All the 42 cases have been used for training, testing and validation of the prototype through leave-one-out cross validation.

The leave-one-out cross validation provides almost unbiased generalization of performance and the greatest possible amount of data is used for training in each case, which presumably increases the chance that the retrieval is an accurate one (Witten and Frank, 2005). Alemu (2010) and Henok (2011) also proved the applicability of the leave-one-out cross validation for CBR system retrieval evaluation in their respective master's thesis. So with the leave-one-out cross validation the researcher conducted 42 testing with 41 training cases.

The whole case-base is composed of 42 instance cases, this number is ideal for this study because there are other medical CBR researches conducted with number of case instances around 42. For instance, Schwartz *et al* (2008) designed a CBR research entitled "Use of Case-Based Reasoning to Enhance Intensive Management of Patients on Insulin Pump Therapy". Their study achieved the research objective with 20 type I DM case instances. The other medical CBR research conducted by Salem *et al* (2005), a case-based expert system for supporting diagnosis of heart diseases also used 42 cases instances for training and testing the system.

3.9. Implementation Tool

The development tool selected in this research is jCOLIBRI. jCOLIBRI is an object-oriented framework for developing CBR applications. It provided most of the functionalities needed to represent structured cases, methods, and similarity functions used in this study. jCOLIBRI is supported by a GUI that guided the researcher on configuration of the prototype and made things easy. The main critical reason the researcher selected this tool is to save time because jCOLIBRI is a framework suited to CBR applications. According to, Iqbal and Ashraf (2006), framework approach is very appealing. It allows the reuse of code and design for a broad class of problems. It also gives the ability to write complex applications quickly. Hence, jCOLIBRI provided the following rewards for the researcher in this study.

- Easy to accesses, install, learn, and use
- Allows to represent cases easily
- Offered an easy development process
- Provided GUI that allowed the researcher to choose methods for desired tasks.
- Easily integrated new elements with the old one.

3.10. Performance Evaluation Mechanism

The developed CBR system was tested and the results were evaluated against the research objective established. The whole evaluation process was performed with two approaches: system validation and user acceptance testing. The system validation was performed to assure weather the prototype is performed the intended task or not. On the other hand, the user acceptance tests were concerned with the test of the user with regard to certain criteria.

The validity of the system was tested using test datasets prepared by the researcher or case instances from the case-base to measure the system performance. First the system was tested for accuracy and consistency. Next to this, the system also tested for similarity. The main objective of this testing was to evaluate the system weather it retrieved the appropriate case for a given query Q. Standard statistical performance evaluation techniques, precision and recall, were used to measure system's retrieval effectiveness. Using leave-one-out cross validation the dataset was partitioned into testing and training sets. Then, for each case instance, precision and recall were calculated and the overall performance of the system was analyzed. Finally, feedbacks from the users were also gathered through user evaluation form. Users evaluate the system by taking different predefined criteria provided by the researcher.

CHAPTER FOUR
KNOWLEDGE ACQUISITION AND CONCEPT MODELING

4.1. Knowledge Acquisition

Knowledge acquisition is formally defined as "the transfer and transformation of problem solving expertise from some knowledge source to a computer program" (Miller, 2009). Knowledge acquisition cannot be thought of as a single problem; there are several dimensions to the transfer and transformation of problem-solving expertise from a human expert or other knowledge source into a program (Buchanan, 1985). Hence, according to Jones (1998), knowledge acquisition is the process of extracting, structuring and organizing knowledge from one source, usually human experts.

Buchanan (1985) conducted a research on how to systematize the tedious tasks of knowledge acquisition and he found three stages of knowledge acquisition called opening, middle game, and end game.

In the opening stage, the knowledge engineer must plan for the terminology and the problem-solving framework. This stage is the base for all other works of the process. The middle game is played based on the design in the previous stage. In a rule-based system, a specialist provides a large block of rules to cover many cases. In the end game, the Knowledge-Base is evaluated by applying it to test cases.

There are three main issues critical to all Knowledge-Based systems in knowledge acquisition task. The first and the most one is to assure the problem domain and the type of knowledge in the domain is goes with a KBS, the second one, identify where to find expertise and evaluate them weather they have or not the required expertise that the project demands. Third, if the expertise emanate from the person, identify specific knowledge acquisition techniques and participant individuals (Jones, 1989).

The knowledge acquisition process for CBR is similar with other techniques of reasoning. The difference is a matter of terminology. According to Miller (2009), the knowledge acquisition process in CBR has two basic tasks: problem analysis and development of the inference mechanism. Problem analysis involves transforming the information taken from the domain expert into the problem and solution fields in the case-based data structure. The inference mechanism is the algorithm used for similarity assessment during the case retrieval process.

Miller (2009) also described that there are three general sequences of events for knowledge acquisition for AI by citing Byrd et al (1992) as follows:

1. Knowledge Engineer elicits data and information from the domain expert
2. Knowledge Engineer interprets the data and information and draws conclusions about the expert's underlying knowledge and reasoning processes
3. Knowledge Engineer uses the conclusions to construct a model which describes the expert's knowledge and processes
4. Repeat steps 1-3 as the expert system evolves into a functional system

CBR solves problems using the already stored knowledge and captures new knowledge, making it immediately available for solving the next problem. Therefore, CBR can be seen as a method for problem solving and also as a method to capture new experience and make it immediately available for problem solving (Perner, 2008). In addition, in section 2.4.3 also described that CBR is often considered as a way to reduce the knowledge acquisition task because of the ease of storage of solved cases in the case base. Indeed, by accumulating cases, a CBR system progressively increases its experience base and its ability to solve more problems. However, case retention is not sufficient to acquire all the knowledge a system needs to reason on cases. Hence, cases must be acquired by other means.

Accordingly, in this chapter, the researcher focused mainly on obtaining knowledge from experts and relevant documents. Following the knowledge acquisition, the researcher constructed the case structure and built the model by identifying the concepts and patterns involved in managing diabetes patient treatment. In this research, the knowledge acquisition process was done through the collection of case histories, consulting related documents, direct observation, and conducting unstructured interviews and discussions with domain experts.

In CBR, the basic knowledge representation is the case, which consists of a description of the problem, solution to the problem, and the outcome. The description of the problem will includes all information explicitly taken into account in solving the problem (Maimone, 2006). Acquiring cases provides acquiring data and acquiring knowledge. Hence, the researcher collected descriptions and solutions of cases. Because of time and other constraints the outcome of the solution is not included in the case collection process of the research.

4.1.1. Knowledge Acquisition from Domain Experts

There number of knowledge sources to develop Knowledge-Bases systems such as books, research articles and manuals; however human experts are the core sources of knowledge (Jones, 1989). Experts tend to be more current and have a broader range of knowledge than other sources (Pomykalski *et al*, 1999). They can also respond to interviews actively and provide different sets of examples.

A Knowledge-Base system attempts to replicate in software the reasoning abilities of human experts who are distinctive because of their particular knowledge and specialized intelligence (Jones, 1989). The intelligence that is found in the heads of human experts is called tacit knowledge.

Tacit knowledge is accumulated through study and experience. Tacit knowledge grows through the practice of trial and error and the experience of success and failure. Tacit knowledge, therefore, is time specific as well as context-specific. It is difficult to formalize, record, or articulate. It includes subjective insights, intuitions and conjectures (Uriarte, 2008). In a process of extracting tacit knowledge, interview is at the heart. Tacit knowledge is usually extracted through a series of intense and systematic interviews

(Jones, 1998). In CBR knowledge is acquired and represented in cases. Therefore, the researcher has acquired knowledge from physicians by tracking how they interpret patient data.

4.1.2. Knowledge Acquisition from Documents

Documents are other alternatives of knowledge in KBS development. Documents are generally cheaper to acquire and use. However, they typically have limited amounts of information and what they have is not always completely relevant (Sajja, 2008). The knowledge we found in documents is called explicit knowledge. Explicit knowledge comprises anything that can be codified and documented (Uriarte, 2008). The researcher obtained these knowledge from guidelines of diabetes management, national and other diabetes association resources (bulletins, brochures, and websites), research journals, diabetes related books, and patient case cards.

4.1.3. Knowledge Acquisition through direct observation

Direct observation is a tool where the researcher directly observes the events and gathers first-hand information. This occurs when the researcher makes a site visit to gather data. In this research the researcher did number of direct observations on patient and physician interactions such as diagnosis, treatment and advice in order to collect facts that couldn't be found in other sources.

4.2. Concept Modeling

Primary elements of knowledge are concepts and relationships between concepts. Concepts are perceived as regularities in events (Cañas *et al*, 2004). Knowledge Modeling enables to present concepts' data or information in a reusable format for the purpose of preserving, improving, sharing, aggregating, and processing knowledge to simulate intelligence (Makhfi, 2011).

A thorough understanding of different knowledge representations is a vital element of KBS, since the ease of solving a problem is almost completely determined by the way the problem is conceptualized and represented. Knowledge engineers make use of a number of ways of representing knowledge when acquiring concepts from experts. These are usually referred to as knowledge models. Among these ladders are preferred modeling types. Ladders are hierarchical (tree-like) diagrams. Some important types of ladders are concept ladder, composition ladder, decision ladder, and attribute ladder. From these sub-types of ladders, decision ladder is selected by the researcher, because a decision ladder shows the alternative courses of action for a particular decision and is a useful way of representing detailed process knowledge.

4.2.1 Identifying Case Features

According to (Marling *et al* ., 2007), in representing a problem, it is necessary to include all information that is typically used to describe such a problem as well as all information that is explicitly taken into account by a human problem solver in solving the problem. Typical information used to describe diabetes management problems includes: Age, Sex, duration of treatment, pervious blood glucose level, current blood glucose levels, current medication signs and symptoms, and complication. These diabetes problem characterization concepts are identified by conducting intense interviews with the domain experts, through direct observation and analyzing patient case story cards with the help of experts.

With this respect, concepts related to type IIDM management are modeled as follows. This research focuses on the treatment of DM type II, not included the diagnosis. So the problem is modeled after the patient is declared as full blown type II DM

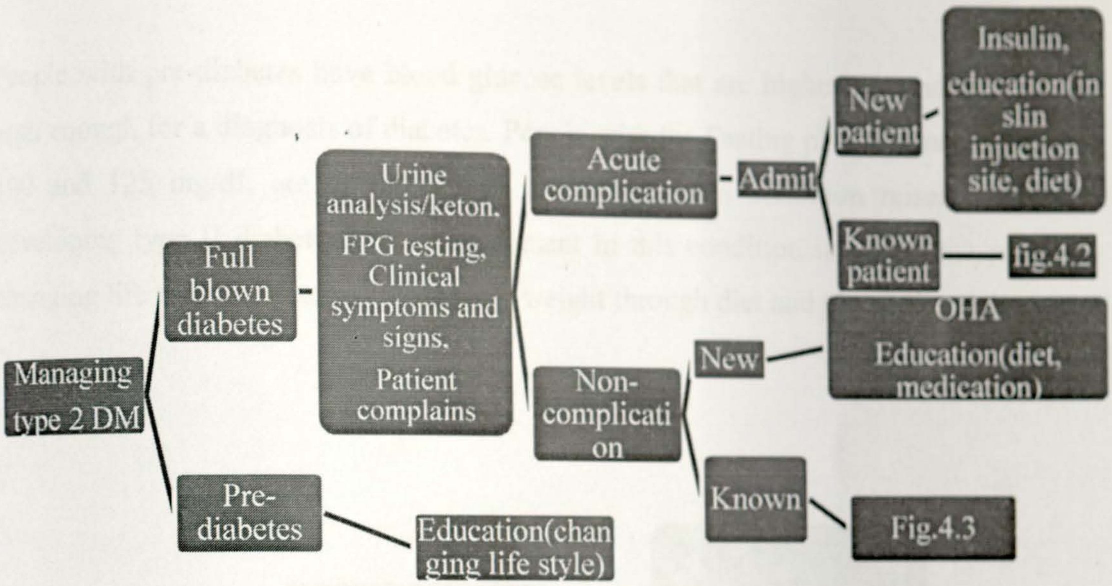


Figure 4.1: Management of Type II DM

The management started by looking the patient seriously whether complications are developed or not. Acute complications of diabetes occurred when acidosis and ketones exist in the urine and/or blood, and if FPG (fasting plasma glucose) $\geq$ 600 mg/dL. Clinical signs and symptoms , and patient complains such as dry mouth, thirst, frequent urination, blurry vision, fatigue, weight loss, sweating, shaking , hunger, dizziness, faintness, numbness of lips and tongues, headache and slowed speech are also indicate the degree complications.

If the patient developed complications, then the patient is admitted. The patient is admitted means the hospital allows the patient to take a bed for full management of the complications. If the patient is registered for the first time (new patient), the management is started according to patient complaint, and clinical signs and symptoms in addition to laboratory results (urine analysis and FPG). The treatment of complicated type II DM is insulin injection and for each other chronic illness (such as kidney failure, heart disease, stroke, blindness, gangrene), they are referred to respective specialists. Non- complicated new patients are treated with oral hyperglycemic agents (OHA) plus education (such as diet, ensuring hygiene, and medication).

People with pre-diabetes have blood glucose levels that are higher than normal but not high enough for a diagnosis of diabetes. People with the Fasting plasma glucose between 100 and 125 mg/dL are considered as pre-diabetes this condition raises the risk of developing type II diabetes. The management in this condition is advice the people to changing life style for example, loss body weight through diet and physical activity.

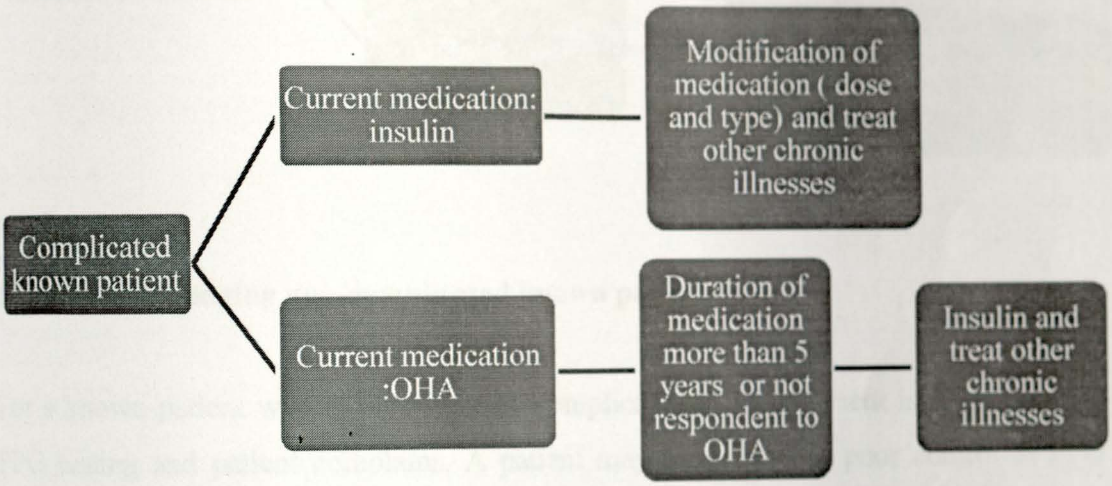


Figure 4.2: Managing complicated known patient

Before starting to treat complicated known patients, their current medication information is critical such as name, type and dose of drugs. Most of the time type II DM patients start their treatment with OHA (tablets). After some time latter (fife years), most of them do not respond to OHA or they develop other chronic diseases. At this time it is recommended to treat with insulin. There are exceptions that start medication with insulin at their first contact depending on their degree of severity as we have seen in Fig. 4.1. For complicated patients who are already treated with insulin, the management continues with insulin, just by doing modifications (type and dose). It is not recommended back to OHA drugs.

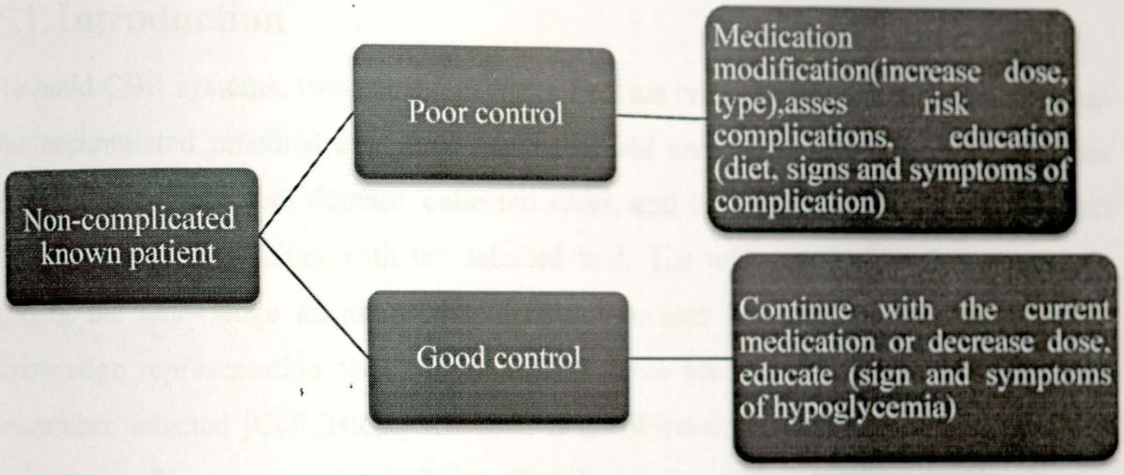


Figure 4.3: Managing non-complicated known patient

For a known patient who did not develop complications, the treatment is determined by FPG testing and patient complains. A patient may have good or poor control of FPG testing as compared to the pervious time and target glucose level (110 ml/dl). A patient who has good control is recommended to continue with the current medication or decrease the dose to prevent the patient from hypoglycemia, which is abnormal low blood sugar usually resulting from excessive insulin, physical exercise, or a poor diet. The patient is also informed about the signs and symptoms of hypoglycemia such as sweating, shaking, hunger, dizziness, faintness, numbness of lips and tongues and headache.

On the other hand, if the patient has poor control, then modification of medication (such as increase drug dose, change drug type), investigate the risk of complication, and education about diet, signs and symptoms of complications are critical parts of the treatment.

CHAPTER FIVE

IMPLEMENTATION AND PERFORMANCE EVALUATION

5.1. Introduction

To build CBR systems, two necessities have become critical: the availability of tool and the accumulated practical experience of real-world problems. After the researcher has understood the problem domain, collected cases, and the problem is modeled, the next task is prototype building with the selected tool. The main goal of implementation is coding the knowledge acquired from different sources into the computer using CBR knowledge representation tool. Even though there are a number of other tools, the researcher selected jCOLIBRI. jCOLIBRI is an object-oriented framework in Java that promotes software reuse for building Case-based Reasoning (CBR) systems (Recio-Garcia and Diaz-Agudo, 2004). In addition, it is easy to use, open source software, and the researcher is more familiar than other tools.

5.2. Retrieval Algorithm

The retrieval task starts with a partial problem description and ends when a best matching previous case has been found. The goal of retrieval task is to return a set of cases that are sufficiently similar to the query. Unlike database searches that target a specific value in a record, retrieval of cases from the case-base perform partial matches since; in general, there may be no existing case that exactly matches the new case. CBR will be ideal for a problem at hand only when retrieval algorithm is efficient in handling cases. Among the well-known algorithms for case retrieval, the simplest one is the Nearest Neighbor using local and global similarity functions (Recio-Garc´ıaet al, 2010). According to Salem *et al* (2005), in their diagnosis of heart diseases, the retrieval strategy employed by Nearest Neighbor was acceptable. In addition, jCOLIBRI's framework is very powerful which supports different types of algorithms from simple Nearest Neighbor to more complex retrieval algorithms.

Nearest neighbor algorithm involves the assessment of similarity between stored cases and the new input case-based on matching a weighted sum of features. jCOLIBRI enables the knowledge engineers to set weight to each case feature based on their relevance to the physician's decision (Fig.5.1). There was too much involvement of domain experts in assignment of weight to features in this research. The similarity of each case to the input case is represented as a real number in [0, 1]. Nearest-neighbor algorithm finds the closest matches of the cases already stored in the case-base to the new case using a distance calculation, which determines how similar two cases are by comparing their features, the pseudo code of this algorithm can be written as follows (Salem et al, 2005).

For each feature in the input case:

Find the corresponding feature in the stored case

Compare the two values to each other and compute the degree of match

Multiply by a coefficient representing the importance of the feature to the match

Add the results to derive an average match score

This number represents the degree of match of the old case to the input.

A case can be chosen by choosing the item with the largest score.

The distance between the new case and stored cases was computed using the following equation.

$$\frac{\sum_{i=1}^{n} w_i \times sim\left(f_i^I, f_i^R\right)}{\sum_{i=1}^{n} w_i}$$

Where:

W_i:- is the importance (weight) of the feature *i*

Sim ():- is the similarity function

f_i^I, f_i^R:- are the values for feature f_i in the source and target cases, respectively

n:- the number of attributes in each case

The similarity function (Sim (f_i^I, f_i^R)) defined as follows

If feature f_i is numeric, then $Sim\ (f_i^I, f_i^R) = 1 - (|f_i^I - f_i^R| / |f_{max} - f_{min}|)$

If feature f_i is symbolic and $f_i^I = f_i^R$, then $Sim\ (f_i^I, f_i^R) = 1$

If feature f_i is symbolic and $f_i^I \neq f_i^R$, then $Sim\ (f_i^I, f_i^R) = 0$

Where:

f_{max} *is the maximum value of the attribute i in the case-base*

f_{min} *is the minimum value of the attribute i in the case-base*

f_i^I *is the value of attribute i of the input case (new case)*

f_i^R *is the value of attribute i of the case in the case-base*

The similarity between cases is considered to be the weighted summation of the similarity between attributes. Then cases whose descriptions are similar to a new case are ranked higher than those whose descriptions are less similar.

5.3. Similarity

In this research, cases were compared by their surface similarity. The simplest method for computing similarity is to look at surface similarity between the current problem and a case in memory (López *et al*, 2005). Cases were compared by looking individual attributes. The researcher presented 42 cases as a simple feature vector (set of attribute-value pairs). The simplest definition of similarity between symbolic values is to demand an exact match. With this representation, a *local* similarity measure was usually defined for each attribute and a *global* similarity measure was computed as a weighted average of the local similarities. The weights, assigned by domain experts to case attribute, allowed them to have varying degrees of importance.

5.4. Building the CBR Prototype for Type II DM Patients

5.4.1. Building the Case-Base

CBR solves new problems by remembering the previous similar situations and also reuses information and knowledge of those situations. Hence, in this study, 42 previously solved DM type II cases were collected from Dessie Referral Hospital. After collecting the cases, case characterization was done with the help of domain experts such as doctors, nurses, and laboratory technicians. After identification of case attributes, the cases were stored in a plain text as columns and rows. Columns represented case attributes and rows represented individual cases. Each attribute has a sequence of possible values separated by commas associated with a case. A small section of the case-base (Plain Text File), which contained the descriptions of three type II DM patients, is given in appendix II.

A case is composed of description (describes the problem by means of several attributes), solution (contains the description of the solution of the case) and result (represents the result of applying the case to a real situation). Descriptions and solutions are sets of attributes and the researcher stored these attributes in plain text file. Management of DM type II is complex by its nature because it causes or may be caused by a number of other chronic illnesses. The treatment was also vey individualized. This made describing type II DM problems with certain number of attributes difficult. However, as it is depicted in **Table 5.1**, the following attributes were identified through deep discussions with problem domain experts. In this research, each individual case has nine attributes: eight description attributes and one solution attribute. Description attributes describe type II DM case and the solution attribute is used to store the treatment provided by the physician.

Attribute name	Description	Parameter of case
Age	Age of the patient	Description
Sex	Sex of the patient	Description
Duration	How log the treatment was started	Description
Pervious glucose	The last visit of glucose level	Description
Current glucose	The glucose level at current time	Description
Medication	The regimen the patient on	Description
Complication	Other chronic illnesses resulted by the diabetes	Description
Signs and Symptoms	Phenomena that indicate existence of type II DM	Description
Treatment	The management based on the case description (physician's decision).	Description

Table 5.1: Case Attributes

5.4.2. Defining the Case Structure with jCOLIBRI 1.0

The framework supports several case structures from plain attribute value records to hierarchical trees with composed attributes. Case structure of jCOLIBRI contains the meta-information or description of case attributes. There are two types of attributes: simple and compound. Simple attributes are described by name, type, weight and local similarity function. Compound attributes collect other simple attributes allowing complex case structure and described by name, weight, and global similarity function. When two cases are compared, the local similarity functions are used to compare simple attribute values. Global similarity functions are linked to compound attributes and are used to gather similarities of the collected attributes in a unique similarity value. Local similarity functions are liked for simple attributes. In this research, *equal local* and *interval local* similarity functions were used.

Equal local similarity: if the local similarity of an attribute is equal, then the corresponding attribute of the query matches exactly with the case otherwise matching fails.

Interval local similarity: exact value matching is not mandatory. Values will match within interval set.

Global similarity function: It is linked with compound attributes and used to get similarity of cases in unique similarity value. The global similarity function used in this research was average similarity function. It considered the average of all attribute values. To retrieve similar cases to the input case, individual attributes were compared by local similarity function. In this research, numeric attributes were compared with interval local similarity function and string type attributes were compared by equal local similarity function. Thus, input case attributes (pervious glucose, current glucose, duration, and age) were compared by local interval similarity function whereas current medication, complication, sign and symptom, and sex were compared by equal local similarity function with the respective attributes of stored cases. On the other hand, global similarity function was used to compute a weighted average of the local similarities.

Name	Data type	Weight	Similarity function
Most influential attributes			
Pervious glucose	Integer	0.9	Local (interval)
Current glucose	Integer	0.9	Local (interval)
Current medication	String	1.0	Local (equal)
Signs and symptoms	String	1.0	Local(equal)
Influential attributes			
Complication	String	0.6	Local (equal)
Duration	Integer	0.5	Local (interval)
Other attributes			
Age	Integer	0.4	Local (interval)
Sex	string	0.5	Local (equal)
Solution attribute			
Treatment	string	1.0	Global (average)

Table 5.2: Properties of Case Attributes

Table 5.2 shows the description of case attributes. The whole attributes were classified into four groups based on their impact on the decision made by the physician. The groups are most influential attributes (pervious glucose, current glucose, medication, and signs and symptoms), influential attributes (complication and duration), other attributes (age and sex), and solution attribute (treatment). Most influential attributes are attributes of type II DM case on which the decision can be drawn with the absence of other attribute values. As it is described in **Table 5.2,** the weight of most influential attributes is higher than that of others. Thus, the treatment decision cannot be drawn without the value of these attributes. Influential attributes are attributes which supplement decision making. Other attributes are used to make the decision more complete.

Defining case structure in jCOLIBRI was done by using simple case structure window. To define the case structure, the researcher used GUI of jCOLIBRI. The case structure window has two panels as shown in Fig. 5.1. The left panel enables to write and display attribute name as tree structure whereas the right panel enables to set and display property values of the selected attribute. Once defined, the case structure is stored in XML file.

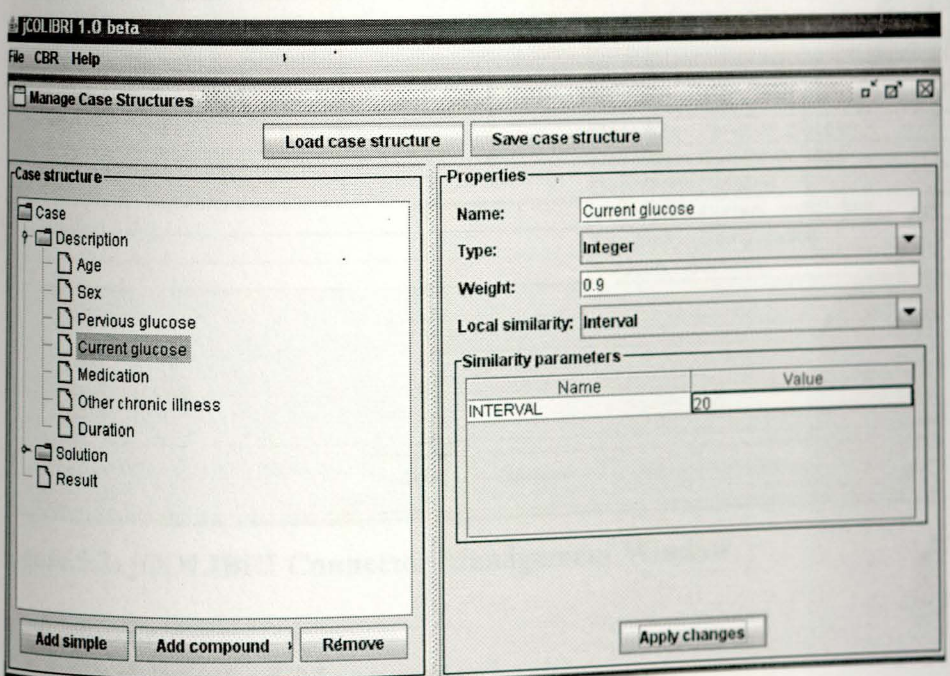


Figure.5.1: jCOLIBRI Case Structure Definition Window

5.4.3. Configuring Connectors /Load Case-base

jCOLIBRI is able to access the case-base from the physical medium through connectors. Connectors are objects that know how to access and retrieve cases from the medium and return those cases to the CBR system in a uniform way. In this research plain text type (Attribute-value cases separated by commas) connector is used. Connectors require the case structure's file and the case-base's file path in order to map the case structure attributes into columns of the plain text (case-base) as depicted below in **Fig.5.2** . The connector is also stored in XML file.

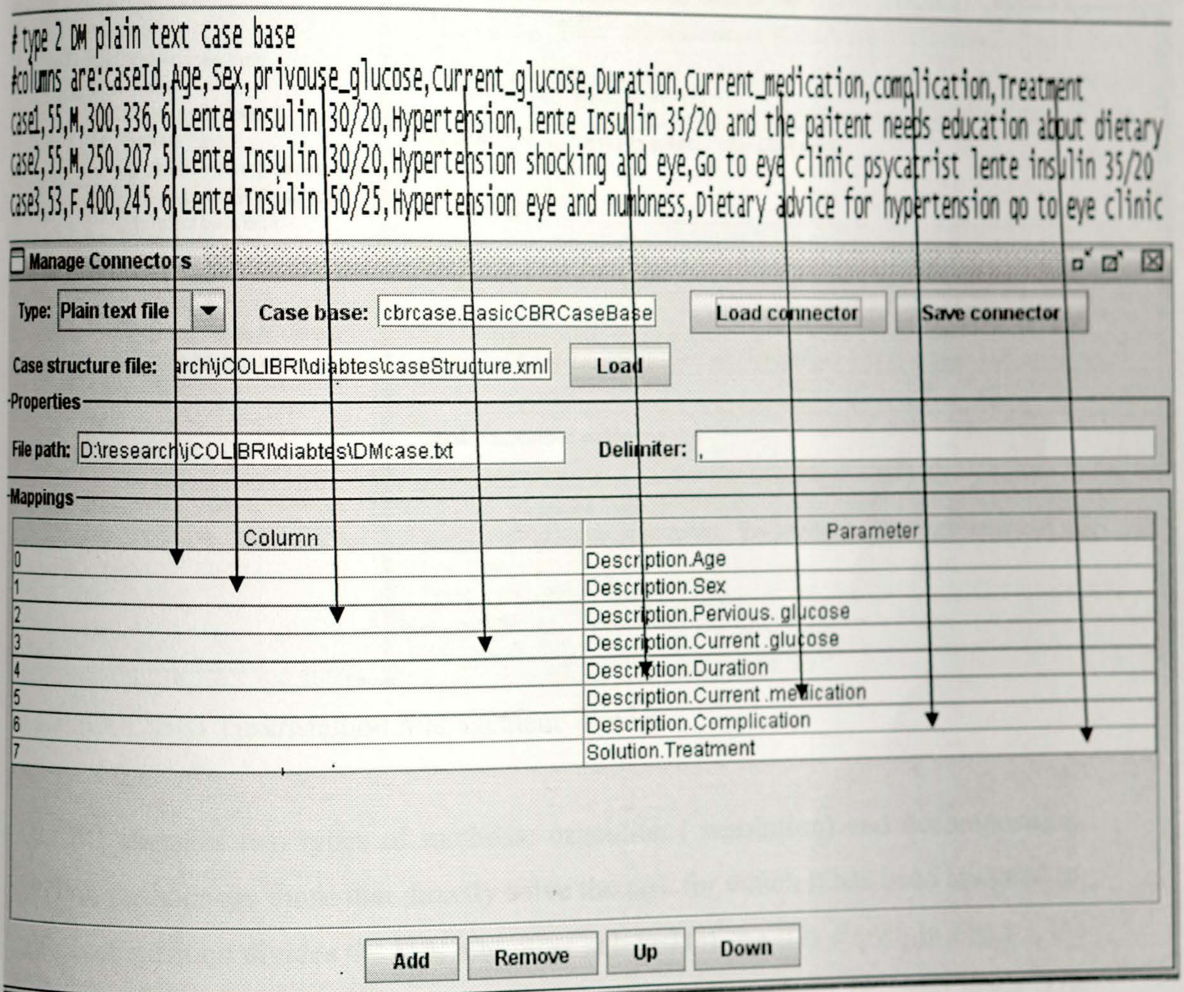


Figure.5.2: jCOLIBRI Connector Management Window

5.4.4. Task/Method Management

Tasks are represented in XML files (tasks.xml) that describe the tasks supported by the framework along with the methods (methods.xml) for solving these tasks. jCOLIBRI is organized in packages. Among these packages that perform and execute the task/method decomposition process is the kernel of jCOLIBRI (Recio-Garcia and Diaz-Agudo, 2004). jCOLIBRI has two types of task packages: core and user defined. Core package tasks are used in this research. jCOLIBRI core task package has built-in methods.

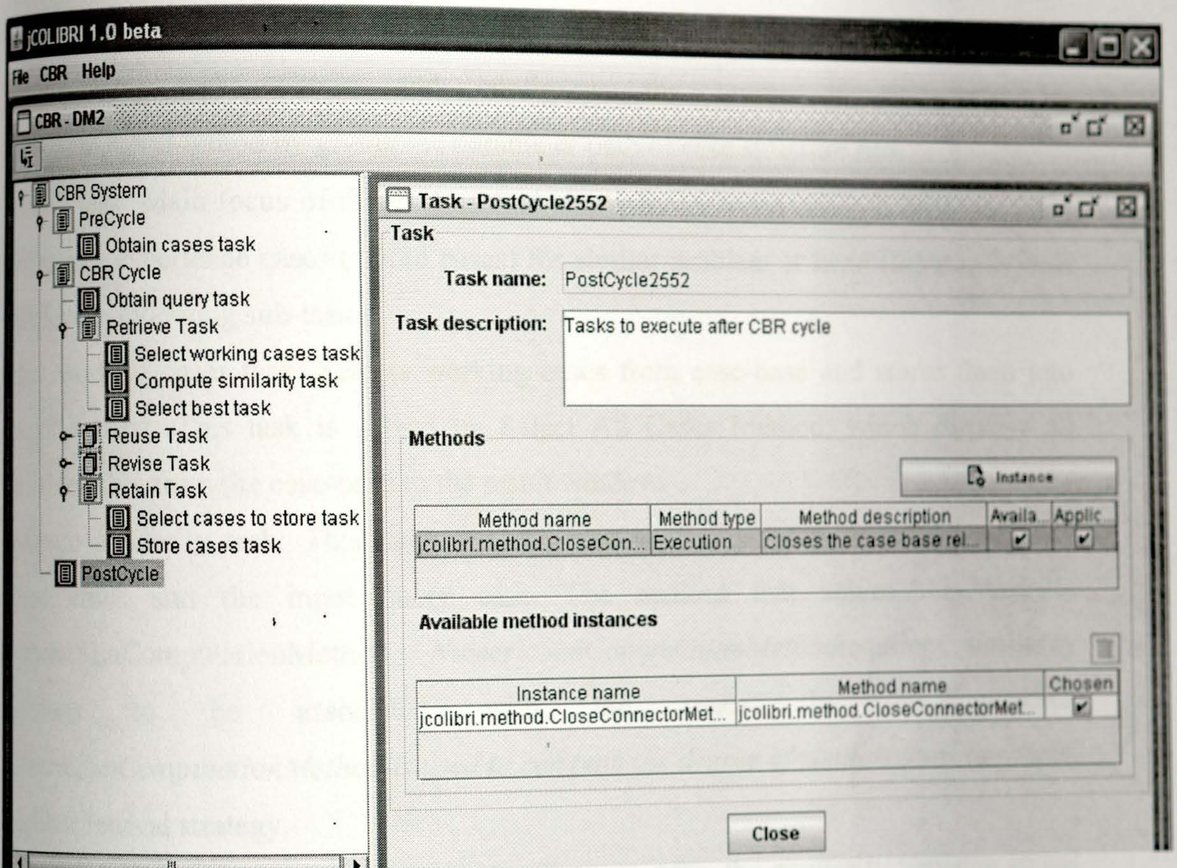


Fig.5.3: jCOLIBRI Task/Method Management Window

jCOLIBRI manages two types of methods: execution (resolution) and decomposition. Execution methods are those that directly solve the task for which it has been assigned to, while decomposition divides the task into other sub-task(s). As it is shown in Fig.5.3, the task panel (left) shows the task tree and the decomposition relates between tasks. The task configuration panel (right) shows available methods for the given task in the given situation. It is the responsibility of the knowledge engineer to select the method that resolves the task. The task decomposition structure tree contains three tasks at the base: pre-cycle, CBR cycle and post-cycle.

Pre-cycle task: This task will execute before CBR cycle. Its task is to get al l the cases in case-base through its sub-task *ObtainCasesTask*solved by *LoadCaseBaseMethod*that returns the whole Case-base.*LoadCaseBaseMethod uses connector path as a parameter.*

CBR cycle: It is main task of CBR cycle. It also has sub-tasks. The following are sub-tasks of CBR cycle along with their methods.

- Obtain query task: obtain and configures the query. It is solved by ConFigureQueryMethod by giving the parameter path of case structure file.
- Retrieve task: Main focus of this task is find similarity between cases. It retrieves most similar past experience cases (stored cases) for similar problem at hand (input query). It includes the following sub-tasks:
  - Select working cases task: Selects working cases from case-base and stores them into current context. This task is solved by Select All Cases Method, which displays all available cases from the case-base to the result window.
  - Compute similarity task: This task will compute similarity between each previously stored cases and the input query case. The method that solves this task is NumericSimComputationMethod. *NumericSimComputationMethod*requires similarity functions to be associated with the cases. In this research *NumericSimComputationMethod is used to compute the degree of similarity using* nearest neighbor retrival strategy.
  - Select best task: Selects the best of the found cases. After the similarity between stored cases and query case is calculated by NumericSimComputationMethod, N number of cases those have better similarity will be selected by the method SelectSomeCaseMethod. The number N is specified by the user. In other words, this task retrieves the N most similar cases. This approach often referred to as "k nearest neighbor" retrieval or simply k-NN.

Reuse task: the retrieve task retrieves cases that are similar to our input situation but may not be entirely appropriate to provide a completed solution. Hence, the solution of the retrieved case is transformed in order to propose a solution to the query by modifying small portions of the retrieved case that do not meet the input specification. This process is also called case-adaptation. It has two sub tasks: Prepare Cases for Adaptation Task and Automatic Reuse Task.

- Prepare Cases for Adaptation: Selects working cases from case-base and stores them into current context. CopyCasesforAdaptationMethod solves this task by taking case structure as a parameter.
- Automatic reuse task: CBR case reuse atomic task.

This task solved by NmericDirectproportionmethod (computes a direct proportion between numeric descriptions attributes and solution attributes.)

In general, the notion of adaptation is very important, since it is rare to find a case which fits perfectly to a given situation. Adaptation allows performing the necessary adjustments to the case, taking into account the specificity of the present situation.

- Revise task: The main focus of this task is to repair the problems identified in the adapted solution. At the end of the adaptation task, the candidate solution proposed by the system has to be tested to check if it is satisfactory or not. The Method that solves this task is ManualRevesionMethod which allows the user to modify cases.
- Retain task: After a case is judged as an acceptable solution to the problem by the revise task, the retain task permits the addition of the solved (adapted) case to the case-base. Subtasks of this task are following
- Select cases to store task: It will give authentication to user for storing case. RetainChoserMethod which allows the user choose cases to store solve this task.
- Store cases task: This will store cases into case-base. StoreCasesMethod solves this task.

Post cycle: It will execute after a CBR cycle. Its task is to close a connection between the case-base and GUI.

Developing a CBR application is a complex task where many decisions must be made. The system designer has to choose how the cases will be represented, the case organization structure, which methods will solve the CBR tasks, and which knowledge will be used by these methods.

5.4.5. Ranking

Ranking refers to place the most relevant cases at the front. Nearest-neighbor retrieval algorithm uses the result of global similarity function to rank retrieved cases. In jCOLIBRI there are two options of displaying relevant cases. The sub-task of Retrieve cycle, *Select best task,* implements ranking of cases. The select best task has two instance methods, SelectBestCaseMethod and SelectSomeCasesMethod. SelectBestCaseMethod ranks all cases in the case-base with their respective degree of similarity to the new case. SelectSomeCasesMethod takes the parameter N, which specifies the top N most similar cases and displays the N top most similar cases to the new case.

5.5. Performance Evaluation and Results

5.5.1. Experimental set up

After the prototype is built, the next task is to evaluate the system in accordance with the objective of the research. The system evaluation was done with two approaches: standard statistical techniques and user evaluation method. The researcher assessed the prototype using test cases and different techniques of system evaluation. Tests were conducted by removing some cases from the case- base and then using them as test cases. According to,(Schwartz, 2008), this is a standard approach used to test CBR systems, especially when it is labor intensive and time consuming to acquire cases .In addition, selected domain experts from the problem domain area were allowed to evaluate the prototype with certain criteria using a given case description.

5.5.2. Retrieval Evaluation

Retrieval evaluation analyzes how well the system performs retrieval of similar cases. The retrieval performance is measured by considering to what degree cases relevant to the input case are moved toward the front of the ordered list of cases. The main objective of this research is to retrieve the most similar previously solved case to a new case. Thus, the evaluation part mainly focused on retrieval evaluation.

Baig (2008) and Salem et al (2005) proposed accuracy and consistency as system evaluation in CBR using nearest neighbor approach. Evaluate for accuracy, which means that if the case-base is queried with one of its cases, it should give the same case with distance measure equals to 100%. Retrieval consistency refers to if exactly the same search has been performed twice, the same source cases should be retrieved with the same accuracy.

So the researcher evaluated the system's accuracy for the following sample cases:

Case 40:

A type IIDM, 60 years old man who has been treated for six years with lente insulin 30 ml in the morning and 20 ml in the evening. The patient has no any complain about diabetes complication. The pervious and current glucose level readings are 320 and 342 respectively.

Case 15:

A type II DM, 56 years old woman who has been treated for three years with glibenclamide 5 mg BID and Metformin 1500 mg daily and she doesn't have any diabetes complication. Her pervious and current glucose level readings are 192 and 190respectively.

The researcher queried the prototype using the above cases; the system returns one case for each retrieval that hits the value 1.0 degree of similarity (Appendix III).

The consistency of the system is also evaluated using two sample cases by testing twice for each case. The top ten seen cases presented as a sample to show the system consistency for each evaluation. The researcher queried the system twice using each of the following two new cases. For each hit the system retrieved the same sequences of cases with similar degree of similarity to each case as sown in **Table 5.3**.

Problem 1: *A 71 years old man who is screened as type II DM before 10 year and currently treated with lente insulin30 ml morning and 15 ml in the evening. The patient developed hypertension with the BP 135/160 and retinopathy complications. The pervious and current glucose level readings are 320 and 450 respectively.*

Problem 2: A 45 years old woman who is screened as type II DM before 15 years currently treated with human insulin 40 ml in the morning and 20 ml in the evening. The patient also has retinopathy complication. The current and the pervious glucose level readings are 170 and 186 respectively.

	Problem 1	Problem 2
First retrieval	Case23,case21,case40,case1,case16,case32,case12,case13,case15,case29	case41,case30,case36,case15,case22,case24,case5,case31,case39,case4,
Second retrieval	Case23,case21,case40,case1,case16,case32,case12,case13,case15,case29	case41,case30,case36,case15,case22,case24,case5,case31,case39,case4,

Table 5.3: system consistency

5.5.2.1. Similarity Testing

The similarity testing was performed to know the actual performance the system in retrieving appropriate case. For this testing the experimental setting was adapted from Henok (2011) with some modification within the context of this research. The researcher prepared three categories of cases. The first category was from the case-base or exactly similar to some cases within the case-base. The second category consists of cases with certain attribute value difference (one or two) from some cases of the case-base. The third group consists of cases with all attribute values are unique. The followings are sample cases against which the queries were modified.

Case 26:

A compliant is 78 years old man screened as type II DM before 3 years and currently treated with lente insulin 15 ml daily. He doesn't have any complication. The pervious and the current fasting blood glucose (FBG) readings are 113ml/dl and 111 ml/dl respectively.

Case 38:

A complainant is 50 years old women screened as type II DM before 3 years, currently treated with glibenclamide 10 mg twice a day and Metformin 500 mg daily. The patient is also hypertensive and treated with enalarpi 5 mg daily. The pervious and the current fasting blood glucose (FBG) readings are 205 mg/dl and 342 mg/dl respectively.

As shown in **table 5.4**, in order to test the similarity performance of the system, the researcher prepared the following 10 queries by changing attribute values of the above sample cases.

Query	Modification of attribute value	With respect to case	Degree of similarity [0,1]
Query1	All attribute values are similar	Case 26	1.0
Query2	The value of attribute" **previous-glucose**" is modified	case 26	0.92
Query3	The value of attribute "**Current-medication**" is modified	Case26	0.87
Query4	The value of attributes **"previous-glucose"** and **"Current-medication**" are modified.	Case 26	0.80
Query5	All attribute values are different	Case 26	Out of range
Query6	All attribute values are similar	Case38	1.0
Query7	The value of attribute "**current-glucose**" is modified	Case 38	0.95
Query8	The value of attribute " **Signs and symptoms** " is modified	Case 38	0.87
Query9	The value of attributes **"Signs and symptoms" and "current glucose"** are modified	case38	0.82
Query10	All attribute values are different	case38	Out of range

Table 5.4 Similarity Testing Performance

This experiment indicated that queries with all attributes similar to a case have 1.0 similarity (exact match). When values of more number of attributes or values of attributes which have more weight were modified, the similarity between a query and the case was degraded. In addition, queries (Query 5 and Query 10) that have unique values to all attributes didn't have similarity with any case in the case-base. From this experiment, the researcher understood that the prototype is able to retrieve the appropriate case for a given query Q.

5.5.2.2. Evaluation with Statistical Methods

To evaluate the system the critical issue is the proportion of the training and the testing data from the whole dataset. What part of the data set is used for testing and which part should be used for training requires a systematic way of partitioning. In this research, there are 42 cases collected from Dessie Referal hospital and leave-one-out cross validation testing proportion was used. Leave-one-out cross-validation is simply *n*-fold cross-validation, where *n* is the number of instances within the case-base. In this research, n refers to 42 which is the total number of type II DM cases collected from Dessie Referral Hospital. Each instance in turn is left out for testing; the remaining instances (case-base) are used for training.

The leave-one-out cross validation provides an almost unbiased generalization of performance and the greatest possible amount of data is used for training in each case, which presumably increases the chance that the retrieval is an accurate one (Witten and Frank, 2005). Alemu (2010) and Henok (2011) also have proven the applicability of the leave-one-out cross validation for CBR system retrieval evaluation in their respective masters thesis. So with the leave-one-out cross validation, the researcher conducted 42 testing with 41 training cases.

The most commonly used statistical retrieval performance measurements are recall and precision (Losee, 2000 and Junker *et al*, 1999). These two measurements evaluate the system effectiveness. For a query Q, *recall* is the percentage of relevant cases in the case-base that have been retrieved and *precision* is the percentage of cases that have been retrieved that are relevant.

Recall and precision in this research defined as follows:

$$\text{Recall}=\frac{Numer\ of\ relevant\ retrived\ cases}{Total\ number\ of\ relevant\ cases\ in\ the\ case\ base}$$

$$\text{Precision}=\frac{Number\ of\ relevant\ retrived\ cases}{total\ number\ of\ retrived\ cases}$$

In order to evaluate using *recall* and *precision,* there should exists relevant and non-relevant cases in the case-base for each case instance and a threshold degree of similarity. Domain experts judged relevancy of cases by analyzing the solution and the attribute pattern between the new case and old case manually. The test cases were selected from the case-base. Domain experts grouped cases mainly based on their solution (treatment) and using some refined specificities. For instance, if there are two cases relevant to a new case N exactly by their treatment, the similarity comparison is done by description attributes pattern analysis. According to the experience of domain experts, there are 6 categories of treatments broadly provided to patients. These classes of treatments are non-drug (life style change), only OHA, both OHA and insulin, and only insulin. Cases whose treatments fall in the same category are considered as relevant to each other.

The researcher analyzed the retrieval output using different intervals of similarity thresholds and reached at optimal results of precision and recall using the threshold interval (0.5, 1.0]. i.e., the cases that have the degree of similarity of greater than 0.5 are retrieved.

The following Table depicted some tested cases and their respective relevant cases.

Tested case	Relevant cases from the case-base
Case1	Case 10, case23, case40, case12
Case4	Case6, case7, case31
Case13	Case2, case3, case14, case17, case19, case20, case26
Case 15	Case16, case22, case27, case28, case29
Case 18	Case21, case 32,case 36,case 38
Case 24	Case37 ,case39, case42

Table 5.6: Relevant cases for sample test cases

After the relevant cases were assigned to each test case, the next step is to calculate the precision and recall values of the retrieval performance of the case-based reasoning system. After conducting 42 experiments using leave-one-out cross validation testing proportion to measure precision and recall results for the sample test queries are shown in the Table below.

Test case	*Retrieval Performance measure*	
	Recall	**Precision**
Case1	1.0	0.4
Case4	1.0	0.3
Case13	0.71	0.5
Case 15	0.4	0.2
Case 18	0.75	0.3
Case 24	0.66	0.2

Table 5.7: Recall and Precision Result for the Sample Test Case

The average precision and recall are 0.46 and 0.69 respectively. The average precision and recall of this research is degraded as compared with previously conducted researches. The first and the most reason for this is heterogeneity of the cases. By their nature, type II DM cases extremely vary and there was no previous collection of cases from the source

(Dessie Referral Hospital). Cases were acquired from patient cards and these cards were tracked when patients came for treatment at outpatient department (OPD). In addition, even for cases that have almost similar case description, physicians suggested different treatments. There was no chance to incorporate more similar cases in the case-base. Hence, the degree of similarity between cases is far apart. This resulted zero precision and recall for some tested cases. So the overall average precision and recall of the research is degraded.

The other reason is case representation. Representative features of cases were not easy to identify and did not get registered from the sources. In almost all cases, the researcher could not get the patients' daily schedules (for work, exercise, meals, and sleep), occupational information, weight and height. For instance, weight and height are important case descriptions based on which body mass index (BMI) of the patient could be determined. BMI helps to adjust the diabetes treatment accordingly and is useful to identify the risk of cardiovascular disease in a given patient. Most case were not fully described, some physicians discuss concepts orally with their clients. Cases were partially described and were not compatible to the CBR system. Thus, the researcher manually transformed cases, and the number of attributes were small. Cases were compared with less number of features, this make the degree of similarity between cases degraded.

5.5.3. Reuse Evaluation

The reuse in this research is not done automatically; rather the domain experts selected the solution of more similar cases for the new problem. In jCOLIBRI, several closest match cases retrieved from the case-based as working cases (proposed solutions) for the new case. From the working cases, the domain expert can select the solution for the queried case. Adaptation process required domain knowledge in order to make amendments to the retrieved case solution to make it suitable for the new case. It is up to the domain expert to use directly or transform the solution of the retrieved case in order to generate solution to the new case.

The researcher tested the reuse performance of the system using the leave-one-out cross testing proportion in the same way as recall and precision testing. The testing was conducted by looking working cases for each test case. From the list of working cases, the top ranked cases were more similar than the other, hence the testing was to check whether these cases were relevant for the new case or not. The underlying idea is the assumption that similar problems have similar solutions. Though this assumption is not always true, it holds for many practical domains. The reuse process was successful when more number of attributes was similar. When the similarity between attribute values far apart the ruse becomes complex and required more effort to adapt it for the new case. Among the 42 tests using leave-one-out cross validation, for 27(64.3%) of them the solution of the first ranked retrieved case was appropriate, While 10 (23.8%) and 5(11.9%) of the tastings get their solution at the seconded and the third ranked cases respectively.

5.5.4. Learning Evaluation

Following revision of the proposed solution, the problem description and its solution retained as a new case, and the system has learned to solve a new problem. The main aim of the learning mechanism of this prototype is to learn from new solved DM cases and use it for managing other DM cases in the future.

The prototype was tested its learning performance using the following sample case

A patient was 56 years old man screened as type II DM patient before 6 year with lente insulin30 ml morning and 15 ml in the evening. He had symptoms of sweating, hunger, shaking, blurred vision, and tiredness. He was free from hypertension and complications. The pervious and current FBG readings were 320 and 450 respectively.

This new case is queried to the prototype and the prototype proposed *lente insulin 35 ml in the morning and 20 ml* in the evening as a treatment (solution). The solution is verified by a domain expert and the solution is revised as lente insulin 35 ml in the morning and 20 ml in the evening and strictly advice the patient about signs and symptoms of hypoglycemia. As the domain expert confirms that the new solved case is valid, the new solved case is retained for future use by the built in retain task of jCLOBRI.

In order to assure the learning performance of the prototype , the same case is queried again and the prototype propose the solution with 1.0(100%) degree of similarity so no need to revise the solution. This shows that the system learn successfully new cases and used them to solve other DM cases.

5.5.5. User Acceptance Testing

Users' acceptance is the opinions that users have formed as a result of interacting with the system. From the users' point of view, system performance evaluation was done in this section. After all the entire end goal of KBS research is to make the task of users easy as much as possible. So the feedback from the users is used to identify problems and to take corrective measures. Inculcating their ideas, views and comments is not a formality, rather it is mandatory. Potential users of this system are experts of diabetes treatment including internists, nurses, and laboratory technicians.

Users were selected from Dessie Hospital with expert sampling. The selection is based on the academic qualification in the domain area, work experience, willingness and participation in this research. Experts participated in the research were three internists, one laboratory technician, and two nurses. The total number participants were six experts since they are the only domain experts available in department of diabetes. These experts knew the research very well in different phases of the research activities such as problem domain understanding, case acquisition, and case attribute identification.

Each expert was allowed to test the system by querying using certain type II DM case and then put their judgment about the system using predefined criteria through user evaluation form (appendix I). The user evaluated the system in terms of adequacy and clarity of advising, relevancy of the retrieved cases in the decision making process, rank of the retrieved relevant cases, fitness of the final solution to the new case at hand, ease of use, relevance of the attributes in describing the case, and speed of the system. These evaluation parameters are adapted from Buchanan and Forsythe (1991), Lee et al (2008), Alemu (2010), Yemisrach (2010) and Henok (2011) with some amendments within the context of this study.

The researcher prepared evaluation values in a ranked manner which is going to be given to each criterion by user evaluation participants. The evaluation values are excellent, very good, good, fair, and poor. For the purpose of analysis, evaluation values have some numeric values in accordance with their rank, excellent=5, very good=4, good=3, fair=2, and poor=1

The summary of all the seven users evaluation is depicted in Table 5.8.

Evaluation Criteria	Poor	Fair	Good	Very good	Excellent	Average
Ease of use		3	3			2.5
Relevancy of the retrieved cases in the decision making process		1		5		3.7
Relevance of the attributes in describing the case			2	3	1	3.8
Fitness of the final solution to the case at hand.		1	3	1	1	3.3
Adequacy and Clarity of advising		1	2	2	1	3.5
Speed of the system			1	3	2	4.2
	Average					3.5

Table 5.8: User Acceptance Performance Evaluation

As it is depicted in the above table, 50% of evaluators rate the criteria "Ease of use "and "Fitness of the final solution to the case at hand" as good. Similarly, criteria "Relevance of the attributes in describing the case" and "Speed of the system" were rated as very good, and criterion "ease of use" was rated as fair by 50 % of the evaluators.

On the other hand, around 16.7 % of evaluators rated the criteria "Relevancy of the retrieved cases in the decision making process", "Fitness of the final solution to the case at hand" and "Adequacy and Clarity of advising" as fair. The same percentage of evaluators rate criteria "Relevance of the attributes in describing the case", "Fitness of the final solution to the case at hand" and "Adequacy and Clarity of advising" rated as excellent. A very large percentage, 83.3%, of the evaluators rated the criterion "Relevancy of the retrieved cases in the decision making process" as very good. In contrary, 16.7 % of the respondents rate the criteria "Fitness of the final solution to the case at hand" and "speed of the system "as very good and good respectively. The criterion "Adequacy and Clarity of advising" was rated as very good by the same percentage of the population, 33.3%. Finally, both "Relevance of the attributes in describing the case" and "speed of the system" were also rated as good and excellent respectively by 33.3 % of the evaluators.

The overall average performance of the prototype based on the above criteria scored 3.5 out of five (70%) from the domain experts (users) point of view.

CHAPTER SIX

CONCLUSION AND RECOMMENDATIONS

6.1. Conclusion

The goal of this thesis is to apply the techniques and concepts known by CBR, for type II DM management, particularly in providing relevant cases and proposing solutions to the new DM case from already solved cases. The potential of CBR is used to enhance the management of patients in providing a flexible case representation, an efficient case-based organization and an effective retrieval algorithm.

CBR is a new technology which enables knowledge engineers to design a KBS that reasons and makes decision from the past solved cases. CBR has become a successful technique for knowledge-based systems in many domains. In medical domains the concept of patient and disease lends itself naturally to a case representation.CBR is particularly well suited to diabetes' management domain. Managing other long-term or chronic medical conditions is an attractive feature of CBR in diabetes management. In addition, CBR is used when generalized knowledge is lacking, thus, it is helpful to manage diabetes cases in which extreme individual variability among patients is found.

The advantage of CBR systems compared to rule-based systems is that they always give a solution using partial matching. Because the closed world assumption of rule-base system needs all facts should be known to the system, this implies that known facts are false. CBR systems do not expect all facts to be known, rather solutions provided to a new problem through adaptation even with some unknown facts.

After the researcher understood the overall concepts and processes of CBR through a thorough literature review, relevant documents were reviewed and domain experts' were interviewed to collect cases. Domain experts were primary sources of knowledge selected by expert sampling from Dessie Referal Hospital diabetes OPD. Unstructured interview was the technique employed to gather knowledge from human domain experts. Cases were collected from patients' history cards at OPD. Hierarchical (tree-like) diagrams, called ladders, were used to model the required knowledge. Case characterization was done with the help of domain experts which resulted case attributes with their respective weight. Each case instance has eight description attributes and one solution attribute.

The prototype is built using the development tool; jCLOBRI. The case structure was constructed using development tool with the identified case attributes and saved in XML file format. The case-base is built with 42 cases collected from Dessie Referral Hospital at diabetes OPD. Cases were stored in plain text (attribute values pair separated by comma) file. A plain text connector type is selected to map the columns of the text file case–base with the XML file case structure. jCLOBRI's core package tasks and methods were used to generate the prototype application.

The prototype is tested and evaluated in order to investigate its performance in providing relevant cases and proposing solution for the problem (new DM case). The evaluation was mainly conducted through statistical analysis recall and precision and scores 0.69 and 0.46 respectively. The feedback from users was also encouraging. In addition, the system was evaluated for its accuracy and consistency. The overall result of the test and evaluation showed that the system produced acceptable and promising results. It is possible to improve the overall performance of the system by increasing the coverage of cases and automatic reuse of solutions.

Demonstrating success in this area may motivate others to attempt a CBR based approach to the management of other chronic diseases. In addition to research value, successful development of a CBR decision support system would have practical value to physicians and diabetic patients.

6.2. Recommendations

This research was conducted for an academic purpose and it has revealed the potential applicability of CBR approach to support the management of type II DM. Moreover, it is the researcher's belief that the contribution of this research work could be a good experience for further studies in the application of CBR approach in therapy and management of other health domains. However, there are improvements and open issues still pending and hence the researcher recommends the following issues as future research directions based on the study.

- A wide range of both physical and lifestyle factors (life events) influence blood glucose levels. Life-event data are used by physicians to determine appropriate therapy. Current systems do not allow documentation of life events. Lack of life-event data impairs the ability to detect clinical problems. As a future research area, investigating how patients' life event data are available helps for a better decision making in CBR for health domain problems.

- This research handled the treatment side of type II DM management. Hence, extending the prototype's capability to predict and prevent problems presents new research challenges and opportunities to improve health outcomes.

- In this research the case characterization and the case-base are constructed manually from patients' history cards with the help of domain experts. Thus, Investigating the applicability of natural language processing (NLP) in constructing the case-base for the CBR approach is a new research area.

- This study investigated a pure case-based reasoning approach for type II DM management. Further researches can be conducted by integrating other approaches like rule-based reasoning with the aim of improving the performance of the Knowledge-Based system.

References

- **AAU (2009)**. Introduction to Research Methods. Preparatory Module for Addis Ababa University Graduate Programs, 2009
- **Abdrabou, E. and Salem,A. (2008)**. Case-Based Reasoning Tools from Shells to Object-Oriented Frameworks. xiv[th] International Conference "Knowledge-Dialogue-Solution" KDS 2008, Varna, Bulgaria, June-July 2008.
- **Abraham, A. (2005)**.Rule-Based Expert Systems. Oklahoma State University, Stillwater, Ok, USA.
- **Alemu, J. (2010)**.A Case-Based Approach for Designing Knowledge-Based System for Aids resource Center. MSc Thesis, Unpublished, Addis Ababa University.
- **Althoff, K. and Weber, R.O. (2006).** Knowledge Management in Case-Based Reasoning. The Knowledge Engineering Review, Vol. 20:3, 305–310.
- **Amadot, A. and Plaza, E. (1994)**.Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. Artificial Intelligence Communications, IOS Press, Vol. 7: 1, pp. 39-59.Available at [http: //www.cs.bham.ac.uk/~axs/], accessed on January 26, 2011.
- **Anteneh,W.(2004).** Design and Development of a Prototype Knowledge-Based System for HIV Pre-Test Counseling. MSc Thesis, Unpublished, Addis Ababa University.
- **Armengol,E.,Palaudàries, A., Plaza,E.(2000).** Individual Prognosis of Diabetes Long-Term Risks: ACBR Approach. Artificial Intelligence Research Institute (Iiia-Csic) and Unitat D' Endocrinologia. Hospital De Mataró, Technical Report Iiia-2000-04. Available at [http://herek.socialpsychology.org/], accessed on July 16, 2011.
- **Armengol,E.,Palaudàries, A., Plaza,E.(2000).**Individual Prognosis of Diabetes Long-Term Risks: ACBR Approach. Artificial Intelligence Research Institute (Iiia-Csic) and Unitat D' Endocrinologia, Hospital De Mataró, Technical Report Iiia-2000-04.

Available at [http://www.diabetesethiopia.org.et/index.html], accessed on June 05, 2011.

- **Baig, M. (2008).** Case-based reasoning – an effective paradigm for providing diagnostic support for stroke patients. A thesis submitted to the School of Computing In conformity with the requirements for the degree of Master of Science Queen's University Kingston, Ontario, Canada September, 2008.
- **Bichindaritz, I. and Marling, C. (2006).** Case-Based Reasoning in the Health Sciences: What's next?Artificial Intelligence in Medicine (2006) 36, 127—135.
- **Buchanan, B. G. (1985).** Some Approaches to Knowledge Acquisition. Stanford University, Department of Computer Science, Stanford, CA, 94305, Knowledge Systems Laboratory, Report No. KSL-85-38, October 1985.
- **Cañas,A.J.***et al*, (2004). Cmap tools: A Knowledge Modeling and Sharing Environment. Concept Maps: Theory, Methodology, and Technology. Proc. of the First Int. Conference on Concept Mapping, Pamplona, Spain 2004.
- **Coiera, W. E. (1996)**.Artificial Intelligence in Medicine: The Challenge Ahead. Journal of American Medical Informatics Association Vol.3, No. 6 Nov/Dec 19.
- ***Cordier, A. (*2009)**. Interactive and Opportunistic Knowledge Acquisition in Case-Based Reasoning. Doctorate Thesis, University Delyon.
- **DH (2001).** National Service Framework for Diabetes: Standards. 26192 1p 50k December 2001, London. Available at [http://www.doh.gov.uk/nsf/diabetes.htm],accessed on January 22, 2011.
- **Diabetes Australia (2010**). The National Diabetes Services Scheme (NDSS): Diabetes Management in General Practice. Fifteenth Edition 2009/10. Available at [www.racgp.org.au], accessed on May 4, 2011.
- **Diabetes Leadership Forum Africa (2010).**Diabetes: The Hidden Pandemic an Impact on Sub-Saharan Africa. Johannesburg, 30 September and 1st October, 2010.
- **EDA (2010)**.The World Diabetes Day 2010 Campaign.
- **Forsythe, D.E. and Buchanan, B.G. (1991).** Broadening Our Approach to Evaluating Medical Information Systems. Proceedings of the Annual Symposium on Computer Application in Medical Care, pp. 8-12

- **HenokBekele (2011).** A Case-Based Reasoning Knowledge-Based System for Hypertension Management. MSc Thesis, Addis Ababa University, Ethiopia.
- **Herek,J.M.(2009).** A Brief Introduction to Sampling.
- **Iqbal, N. and Ashraf, M. H. (2006).** Evaluation of jCOLIBRI. Malardalen University, Vasagatan 44 72123 Vasteras, Sweden
- **Jones, P.H. (1998).** Knowledge Acquisition: Knowledge Engineering in Agriculture. Monograph, ASAE, St. Joseph, MI, No. 8.
- **Lee, K.C.2008).** User Evaluations of Tax Filing Websites. Online Information Review, Vol. 32, pp. 842-859
- **Leondes, C. T. (2000),**Knowledge-Based Systems: Techniques and Applications. Academic Press: London, Vol. 3, pp 802-820.
- **Leondes, C. T. (2000).**Knowledge-Based Systems: Techniques and Applications. Academic Press: London, Vol. 3, pp. 802-820.
- **Lisa, M. (2008).**Purposive Sampling.The Sage Encyclopedia of Qualitative Research Methods: Thousand Oaks, CA, Vol.2, pp.697-698.
- **López, M.** *et al,* **(2005).**Retrieval, Reuse, Revision, and Retention in Case-Based Reasoning. The Knowledge Engineering Review, Vol. 00:0, 1–2. 2005, Cambridge University Press DOI: 10.1017/S0000000000000000, Printed in the United Kingdom.
- **Lopez,M. (2001).** Case-Based Reasoning. IIIA-Artificial Intelligence Research Institute.
- **Losee, R.M. (2000).** When Information Retrieval Measures Agree about the Relative Quality of Document Rankings. Journal of the American Society for Information Science, Vol. 51, pp. 834-840.
- **Lützelschwab, S. (2007).** Case-Based Reasoner: For Owl-S Web Services. Dynamic and Distributed Information Systems, Switzerland.
- **Maimone, A. (2006).** Data and Knowledge Acquisition in Case-basedReasoning for Diabetes Management. A thesis Presented to Faculty of The Russ College of Engineering and Technology of Ohio University, in Partial Fulfillment of the Requirement for the Degree of Master of Science, August 2006.

- **Makhfi, P. (2011).** Introduction to Knowledge Modeling: World of Knowledge Modeling. Available at [http://www.makhfi.com/kcm_intro.htm], accessed on June, 10 2011.
- **Marling, C., Shubrook, J., Schwartz, F. (2007).** Towards Case-Based Reasoning for Diabetes Management. School of Electrical Engineering and Computer Science: Russ College of Engineering and Technology And Appalachian Rural Health Institute, Diabetes And Endocrine Center: College of Osteopathic Medicine, Ohio University, Athens, Ohio 45701, USA, 2007.
- **Masizana-Katongo, A.N., Leburu-Dingalo, T.K., Mpoeleng, D. (2009).** An Expert System for HIV and AIDS Information. Proceedings of the World Congress on Engineering 2009, VolI WCE 2009, July 1 - 3, 2009, London, U.K.
- **Mccarthy, J. (2007).**What is Artificial Intelligence? Computer Science Department, Stanford University.
- **Miller, W. A. (2009).** Problem Detection for Situation Assessment in Case-Based Reasoning for Diabetes Management. A Thesis Presented to The Faculty of the Russ College of Engineering and Technology in Partial Fulfillment of The Requirements for the Degree of Master of Science, June, 2009.
- **Niddk (2008).**Diabetes Overview. Nih Publication, No. 09–3873, November, 2008.
- **Nilsson,M. and Sollenborn, M. (2004).** Advancements and Trends in Medical Case-Based Reasoning: An Overview of Systems and System Development. American Association for Artificial Intelligence, 2004.
- ***Pacharavanich, P., Wongpinunwatana, N.,* Rossini,P.(2000)**. The Development of a Case-Based Reasoning System as a Tool for Residential Valuation in Bangkok. Pacific-Rim Real Estate Society (Press) Conference 2000.
- ***Pacharavanich, P., Wongpinunwatana, N.,* Rossini,P.(2000)**. The Development of a Case-Based Reasoning System as a Tool for Residential Valuation in Bangkok. Pacific-Rim Real Estate Society (Press) Conference 2000.
- **Perner, P. (2008).** Case-Based Reasoning and the Statistical Challenges. Institute of Computer Vision and Applied Computer Sciences, Ibai, Leipzig, Germany.

- **Pomykalski, J. J., Truszkowski, W. F., Brown, D. E. (1999).** Expert Systems. Wiley Encyclopedia for Electrical and Electronics Engineering, J. Webster (Editor), February, 1999.
- **Recio-Garc'Ia,A.J et al (2010).**Lessons Learnt In The Development Of A CBR Framework. Dep. Ingenier'Ia Del Software E Inteligencia Artificial Universidad Complutense De Madrid, Spain, Expert Update, Vol.10, No.1, 2010.
- **Recio-Garcia, J. A. and Diaz-Agudo,B.(2004).** An Introductory User Guide To jCOLIBRI 0.3, Department of Information System Program, University of Complutense Madrid, Spain.
- **Russell,S. and Norving, P. (2004).** Artificial Intelligence A Modern Approach. 2nd Edition, Pearson Education: India, 2004.
- **Sajja,P.S.(2008).**Multi-Agent System for Knowledge-Based Access to Distributed Databases. Interdisciplinary Journal of Information, Knowledge, and Management Volume 3, 2008.
- **Salem, A. M. (2007).**Case-based Reasoning Technology for Medical Diagnosis. World Academy of Science, Engineering and Technology 31 2007.
- **Salem, A.M., Roushdy, M. And Hodhod, R.A. (2005).** A Case-Based Expert System for Supporting Diagnosis of Heart Diseases. International Journal on Artificial Intelligence and Machine Learning, Vol. 5, pp.33-39.
- **Salem, A.M., Roushdy, M. and HodHod, R.A. (2005).** A Case-based Expert System for Supporting Diagnosis of Heart Diseases. International Journal on Artificial Intelligence and Machine Learning, Vol. 5, pp.33-39.
- **Schwartz, F.L Et al (2008).** Use of Case-Based Reasoning to Enhance Intensive Management of Patients on Insulin Pump Therapy. Journal of Diabetes Science and Technology, 2008 July; 2(4): 603–611.
- **Shtitaye, A. and Watkins, P. (2004).** Access to Diabetes Care in Northern Ethiopia. Diabtes Voice, March, 2004, Vol.49, No.1.
- **Sloman, A. (2002).**Artificial Intelligence Development Environment. School of Computer Science, University Of Birmingham

- **Stoyanov, S., Govedarova, N., Popchev, I. (2005)**.CBR-Based Search in BULCHINO Catalogue. University of Plovdiv, Bulgaria, E-Commerce Laboratory
- **The Patient Education Institute (2010)**. Diabetes Introduction. Available At [www.x-plain.com] ,accessed on March 23, 2011
- **Trochim, W. M.K. (2006).** The Research Method in Knowledge-Base. Web Center for Social Research Methods. Avaliable at [http://www.socialresearchmethods.net/], accessed on July, 15, 2011.
- **Tropy, J. M. (2009)**.Diabetes. The Journal of the American Medical Association (JAMA), April 15, 2009, Vol.301, No. 15.
- **Tuso,P.J.(2009).**SERVE Ethiopia. The Permanente Journal/ summer 2009/ Volume 13 No. 3.

  Unwin, N. and Marlin, A. (2004). Diabetes in Society. Diabetes Voice, June 2004, Vol.49, No. 2.
- **Uriarte, F.A. (2008).** Introduction to Knowledge Management: A Brief Introduction to the Basic Elements of Knowledge Management for Non-practitioners Interested in Understanding the Subject. Published by the ASEAN Foundation, Jakarta, Indonesia, 2008.
- **Watson, I. and Marir, F. (1994)**.Case-Based Reasoning: Ai-CBR. Dept. Of Computer Science, University of Auckland, New Zealand, Published in the Knowledge Engineering Review, Vol. 9, No. 4, 1994.
- **WHO (2006)**. Managing Diabetes in Primary Care in Caribbean. Caribbean Health Research Council, 2006.
- **WHO (2011).** Diabetes. Media Centre, Fact Sheet No 312, January 2011.
- **Wild, S.*et al.* (2004).**Global Prevalence of Diabetes: Estimates for the Year 2000 and Projections for 2030.Diabetes Care 27:1047–1053, 2004.
- **Witten, I.H and Frank, E. (2005).** Data Mining: PracticalMachine Learning Tools and Techniques, Morgan Kaufmann Publishers, San Francisco.
- **YemisrachHailemariam (2010).** Application of Case-Based Reasoning in Legal Knowledge-Based System: A Prototype on Children Criminal Cases in Ethiopia. Msc Thesis, Addis Ababa University, Ethiopia

Appendices

Appendix I

Sample interview questions posed to domain experts.

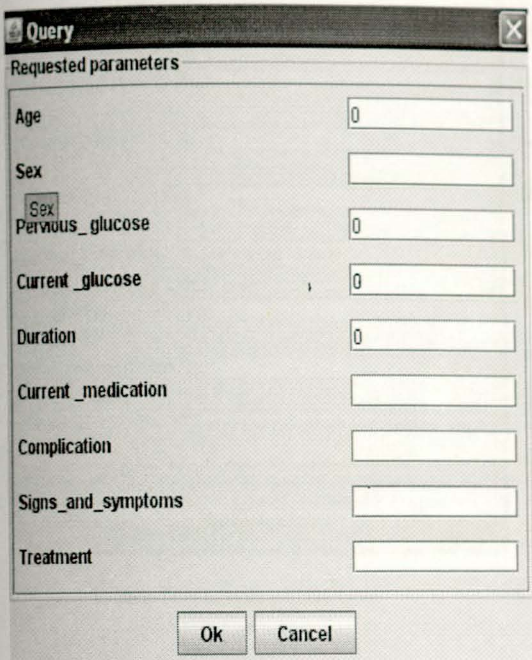
1. What is diabetes mellitus (DM)?
2. What is the basic difference between type I DM and type II DM?
3. What are symptoms and signs of DM in general and specifically type II DM
4. How type II DM is diagnosed?
5. What are treatments of diabetes and the goal of the management?
6. What are other co-illness developed together with diabetes and how they can be treated?
7. Are there stages of type II DM?
8. What are risk factors for type II DM and how to minimize them?
9. What may happen if the patient fails to follow the prescription of the physician?
10. What personal profiles' of the patient used in decision making of the management?
11. What are the other variables used in type II DM management?

Appendix II

Some Section of the plain text Case-Base

```
# type 2 DM plain text case base
#columns are:caseId,Age,Sex,privouse_glucose,Current_glucose,Duration,Current_medication,compl
case1,60,M,310,336,6,Lente Insulin 30/20 and HCT 12.5mg daily,hypertension with previous BP 15(
case2,56,M,444,360,5,Lente Insulin 30/20,hypertension with current BP 140/80 and retinopathy,we
case3,53,F,400,245,6,Lente Insulin 50/25,hypertension with previous BP 150/100 and current 160/
```

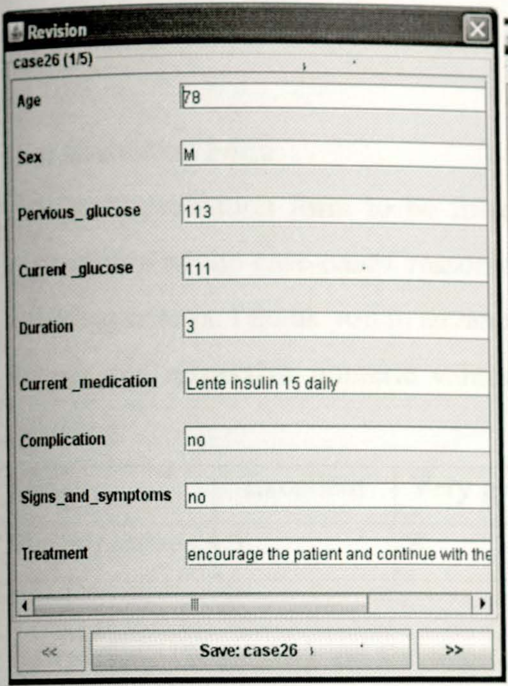
Appendix III

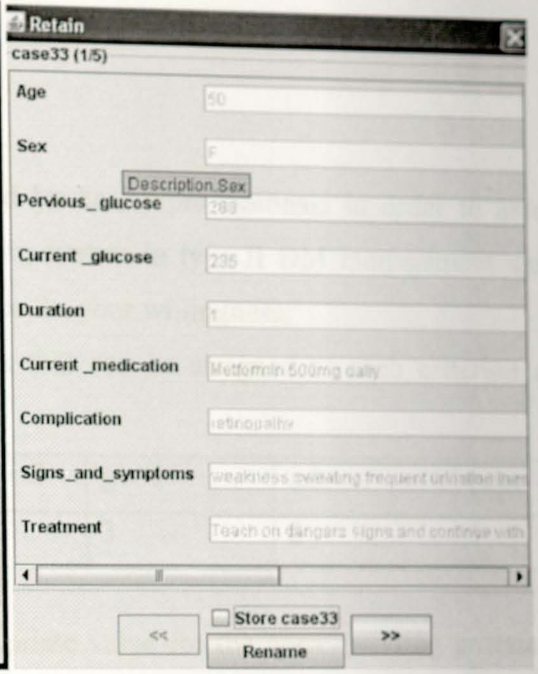


Case entry window

```
WORKING CASES:
case40
has-Description: Description
has-Solution.Treatment: Lente insulin 35/25
has-Description.Complication: no cqmplication
has-Description.Pervious_ glucose: 320
has-Description.Duration: 6
has-Description.Sex: M
has-Description.Current _glucose: 342
has-Description.Current _medication: Lente insulin 30/20
has-Description.Age: 60
has-Result: Result
has-Solution: Solution
 with value:1.0
case1
has-Description: Description
has-Solution.Treatment: lente Insulin 35/20 and the paitent needs education about dietary exercise injection site and serious follow to pressure
has-Description.Complication: Hypertension
has-Description.Pervious_ glucose: 300
has-Description.Duration: 6
has-Description.Sex: M
has-Description.Current _glucose: 336
has-Description.Current _medication: Lente Insulin 30/20
has-Description.Age: 55
has-Result: Result
has-Solution: Solution
 with value:0.3
case12
has-Description: Description
has-Solution.Treatment: Lente insulin 30/20 tell about hypoglycimia signs and symptoms all measure to be taken
has-Description.Complication: no complication
has-Description.Pervious_ glucose: 307
has-Description.Duration: 1
```

Case retrieval result window.





Case revision window

Case retansion window

Appendix IV

User Evaluation Form

This is an evaluation form to be filled by health professionals in order to assess the applicability of the case-based reasoning system in type II DM management using the following criteria. I thank you in advance for your willingness.

Ratings and respective numeric values that can be assigned to each criterion are the following

Rating	Excellent	Very good	good	fair	poor
Numeric value	5	4	3	2	1

Instruction: Please, tick on the appropriate value for the corresponding criteria of the case-based reasoning system.

Evaluation criteria	**Numeric value**				
	1	**2**	**3**	**4**	**5**
Ease of use					
Relevancy of the retrieved cases in the decision making process					
Relevance of the attributes in describing the case					
Fitness of the final solution to the case at hand.					
Adequacy and Clarity of advising					
Speed of the system					

Declaration

I declare that the thesis is my original work and has not been presented for a degree in any other university.

ZEWDITU SISAY

November 2011

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

Getachew Jemaneh (Ato)