Comparative and Integration Study of Some Reasoning Methods

By:

Emdalala Saleh Mohamed Saleh

Supervisor: *Dr.Abdulhamed Mohamed Abdulkafi*

A thesis submitted to the Department of Computer Science in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

Al-Tahadi University, Faculty of Science
Department of Computer Science
Sirte, G.S.P.L.A.J.

Academic year 2006/2007

التاريخ : ....
الموافق : 27/6/
الرقم الاشاري : ع.ن.ك/ن/25/1

Faculty of Science

Department of Computer Science

Title of Thesis

((Comparative and Integration Study of some Reasoning Methods.))

By

Emdalala Saleh Mohamed

Approved by:

Dr. Abdulhamed Mohamed Abdulkafi ....
(Supervisor)

Dr. Idris S. El-Feghi ....
(External examiner)

Dr. Ali Ahmed Abdulaziz ....
(Internal examiner)

Countersigned by:

Dr. Ahmed Farag Mhgoup
(Dean of Faculty of Science)

Acknowledgements

I wish to express my gratitude to Dr. Abdulhameed.M. Abdulkafi for his academic guidance, suggestions, and support.

My special thanks are extended to Dr.Zakaria.S.Zubi for his information, comments, corrections and criticisms which are really enriched my work.

I do sincerely thank Mr.Marwan Algeblawi for his help and support.

I would like to thank Dr.Jaya Raju and Dr.Kunjan.T.Pillai for correcting the language of this thesis.

I greatly indebted to all my family and friends, who have supported me in all possible ways as needed.

Above all, I thank God for making all this work possible.

Abstract

Artificial intelligence (A.I) is defined as a branch of Computer Science that is concerned with the automation of intelligent behavior. A number of important features emerge that seem to be common to all divisions of this field including Knowledge Representation and Reasoning.

The main aim of this thesis is to explore all features of different reasoning schemes, with emphasis on benefits, drawbacks, applicability, and criteria of choosing methods for a specific application.

This thesis describes also a comparative study and evaluation of six different reasoning methods. These methods are: *Rule Based Reasoning (RBR), Logic Based Reasoning (LBR), Frame Based Reasoning (FBR), Case Based Reasoning (CBR), Model Based Reasoning (MBR), and Fuzzy Reasoning (FR).* Primary factors considered in the evaluation were Knowledge unit, Knowledge acquisition issues, Explanation mechanism, and Knowledge transfer across problems and Domain requirements.

In addition to the description of evaluation criteria in this thesis work we implement two expert systems for an appropriate methods so called Computer Diagnostic Expert System (*CDES*) which deal with RBR, and the other is called Decision Support Expert System (*DSES*) which uses FR. These frameworks were implemented to represent knowledge reasoning views.

Table of contents

List of figures

Figure *Page*

Appendix A

Appendix B

List of tables

Chapter 1

Introduction

1.1 Introduction

Artificial intelligence (AI) is a study of computations that makes perceive reason and act [44], comprises methods, tools, and systems for solving problems that normally require the possible intelligence of humans. The term intelligence is always defined as the ability to learn effectively, react adaptively, make proper decisions, and communicate in a language or images in a sophisticated way.

The main objectives of AI are to develop methods and systems for solving problems, usually solved by the intellectual activity of humans. for example; image recognition, language and speech processing, planning, and prediction to Enhance computer information systems, to develop models which can simulate living organisms and the human brain.

The main AI directions of development are to develop methods and systems for solving AI problems without following the human's way and to provide similar results [33].

One of the aims of the researchers in the AI area is the development of techniques which allow modeling of information at higher levels of abstraction. These techniques are embodied in languages or tools which allow programmers to build the logic closely resembles human logic in their implementation and easier to develop and maintain.

A number of important features emerge that seems to be common to all divisions of the field AI which include the following:

- Using computers to do reasoning, pattern recognition, learning, or some other form of differencing.
- A focus on the problems that do not respond to the algorithmic solutions. This underlies the reliance of heuristic search as an AI problem-solving technique.
- A concern with problem solving, using inexact, missing, or poorly defined information and the use of representational formalisms that enables the programmer to compensate for these problems.
- Reasoning about the significant qualitative features of the situation [19].

1.2 Reasoning in Artificial Intelligence

Reasoning is a process which we use in the knowledge, and we have to draw conclusions or infer something new about a domain of interest. It is a necessary part of what we call "Intelligence": without the ability to reason we are doing little more than a lookup when we use information. In fact this is the difference between a standard database system and knowledge base or expert system. Both have information that can be accessed in various ways but the database unlike the expert system has no reasoning facilities and therefore, answers are only limited to specific questions [4].

Experience shows that the performance of tasks that seem to involve intelligence also seem to require a huge store of knowledge. In order to use knowledge and reason with the knowledge, you need what we call a representation and reasoning system (RRS). A representation and reasoning system is composed of a language to communicate with a computer, a way to assign meaning to the language, and some procedures to compute answers for a given input in the language.

We want RRS's where the distance from a natural specification of problems is not very far from the representation of the problem. We also want RRS's where the appropriate computation is given some input that can be effectively determined.

An important and fundamental prerequisite for using an RRS is to decide how a task domain was described. This requires us to decide what kinds of things the domain consists of, and how they are related in order to express task domain problems. Experience in developing and refining representations for particular problems depend on what we know about the problem domains.

Theories about representation and reasoning are only useful so far as they provide tools for the automation of problem solving tasks. Diverse applications include medical diagnosis, scheduling factory processes, robots for hazardous environments, chess playing, autonomous vehicles, natural language translation systems, and cooperative systems. Rather than treating each application separately, we abstract essential features of such applications to allow us to study principles behind intelligent reasoning and actions [15].

1.3 Reasoning and Inference in Problem Solving and Learning

The terms problem solving, learning, reasoning, and inference will be used several times in this thesis, but basically in an intuitive sense. However, before we describe this framework we need to define these terms. The following figure 1.1, illustrates the terms implied the process structure.

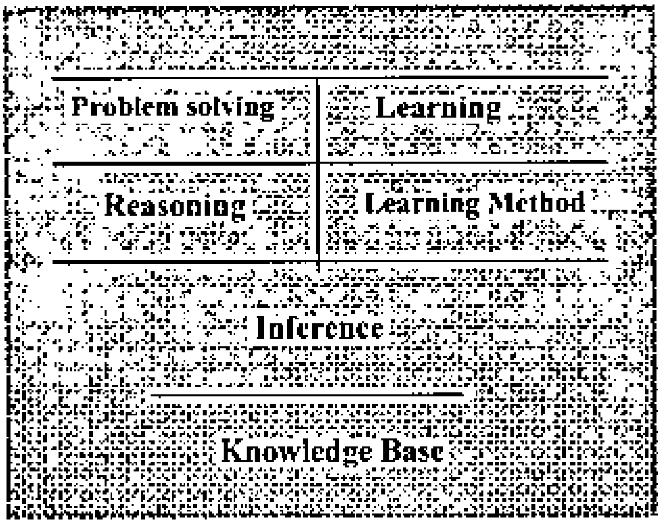


Figure 1.1 Structure of Knowledge Base Processes

Problem solving is a process that takes the problem description, the goal, and the Knowledge Base input, and derives a solution to satisfy the goal. The goal contains a specification of requirements that must be fulfilled in order to end up with a result which can be a solution to the problem. A problem may be structured as sub-problems, in which case the problem solving process may be correspondingly split into sub-processes.

Learning is a process that takes an existing problem solving system and new information input, and derives a system where the quality and/or efficiency of problem solving is improved.

Learning may, of course, be regarded as a problem in its own right, and problem solving could be said to subsume learning. But to avoid confusion in problem solving we will solely refer to solving application problems.

Reasoning is a process that takes some facts and goals, and input. and then derives a result by applying one or more inference methods to the body of the knowledge. Reasoning may be regarded as a more general term than problem solving, in the sense of input and output that may be any kind of information, which is not necessarily to be in the form of problem description and solution. Reasoning is also a subprocess of problem solving, since it is used to derive results that contribute to solving the problem. The term reasoning is most often used in connection with problem solving, and characterizes as a part of problem solving process, as in rule-based reasoning, model-based reasoning, case-based reasoning, logic based reasoning ,fuzzy reasoning, and in frame based reasoning. Being a general process of deriving result through inference, reasoning is also involved in some learning processes - and particularly in knowledge-intensive learning. In this thesis work, it should generally be assumed that reasoning refers to the problem solving process.

Reasoning methods are based on the reasoning types described above. We well put these types together in a more complex form for deriving particular types of results, i.e. for performing a task.

A reasoning model is a type related to their roles for achieving tasks.

Inference denotes the lowest level of processes in the hierarchy, i.e. is the building blocks for reasoning and learning. The inference methods define the primitive operations on the knowledge (the basis for its semantic interpretation) [1].

1.4 Reasoning as a part of Knowledge Modelling

One of the current trends in knowledge modelling is to extend the traditional notion of the knowledge to incorporate reasoning methods and problem solving strategies. Different subtypes of problems in the problem domain often need different strategies and reasoning methods to be successfully and efficiently solved. Strategies for how to solve different types of problems are characteristic of different reasoning methods, which will become part of the explicitly represented model.

Figure 1.2[1] shows the three main components of the knowledge model: the knowledge domain, the control level model of problem solving strategies, reasoning model containing two set of reasoning methods include exact and inexact where (exact set: rule, logic, frame based reasoning and inexact such as case, model, fuzzy based reasoning), and Reasoning structure (which describes the sequencing of tasks). The definitional knowledge model contains object-level as well as control level concepts and how the different types of definitional knowledge are used to specify in the control levels of problem solving and learning. The definitional knowledge is associated with several inference and reasoning methods for interpretation of its semantical content[1].

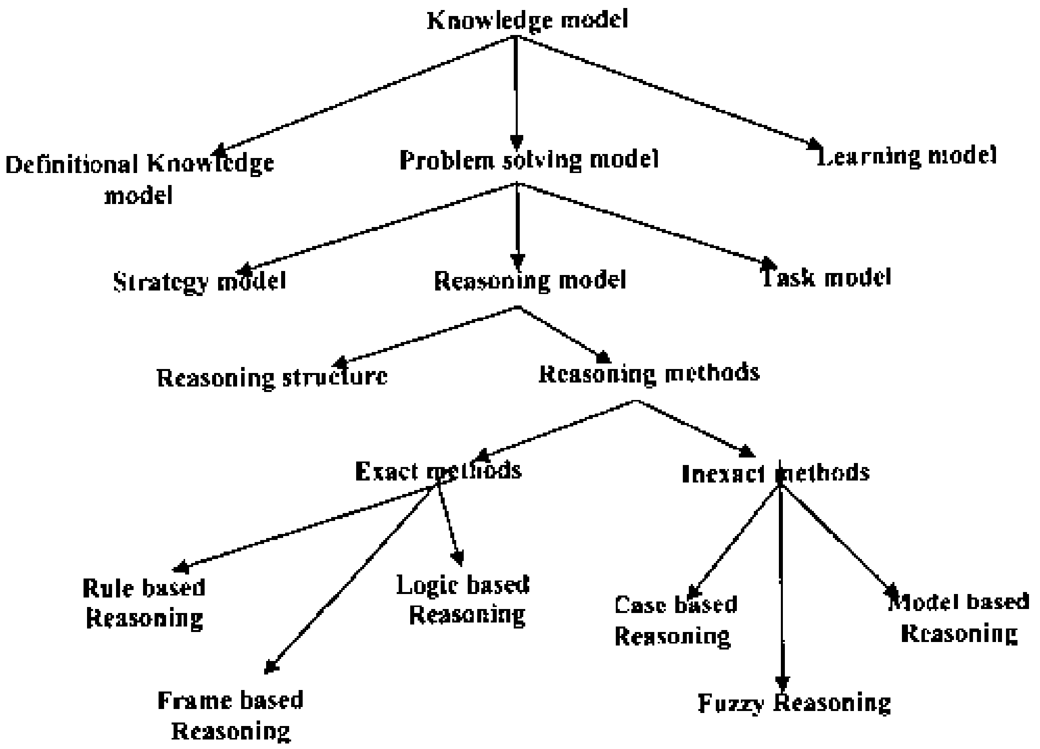


Figure 1.2 A component model of knowledge.

A reasoning model is a further specification of some parts of the problem solving process. Within the perspective of the problem solving model just described, a reasoning process as a successive combination of various inference methods (such as matching, property inheritance, constraint propagation, and rule deduction),is guided

[1] This figure is borrowed from [1].

and supported by explanation methods that focus on the current goal and context of the process.

The model of reasoning within our framework is at the higher level, which means it is in a more general level than the classical models. The purpose of this work is to serve as a top level framework that describes different approaches to the knowledge-intensive reasoning by combining general and case-specific knowledge. The model emphasizes on the generating and evaluating explanations to support both abstract and concrete associations, as well as performing elaborative deep model reasoning.

A reasoning process, given in a general level, could be described by dividing the reasoning process into three sub-processes as mentioned below[1]:

1. Activating knowledge structures,

2. Explaining candidate facts, and

3. Focusing on a conclusion.

A reasoning process is viewed as a process of activating a certain part of the existing knowledge - including triggering of hypotheses and goals, explaining the activated parts that form a coherent knowledge-structure, focusing within the explained structure and returning an explicit answer or just the final state of the system, as illustrated in figure 1.3.

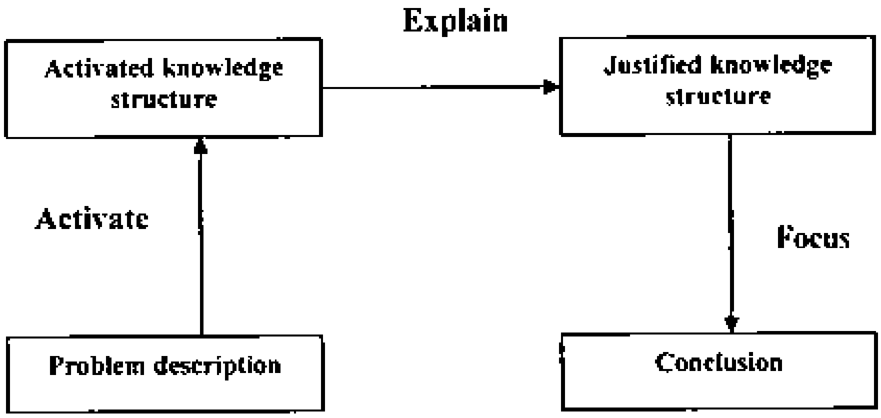


Figure 1.3 The explanation based Reasoning Model.

1. Activating knowledge structures

Activating is an initial set in the knowledge structure which requires marking network concepts that match terms of the input description of data that describes problem, and goal as active. Other activating concepts are via using mechanisms that trigger activation of associated concepts.

2. Explaining candidate facts

Explanation starts to work on the activated concepts and its job is to use the knowledge (and the user, if necessary) to justify, and to confirm or reject candidate facts by producing supporting explanations. For example, if two activated facts contradict, the one with the strongest explanation will be chosen first. The strength of explanatory support depends on the strength of single explanation chains as well as the number of alternative explanations that support the fact hold.

The goal of the reasoning process is to solve particular subproblems, and to focus on the generation of the explanations. The outcome of the explanation phase is a coherent knowledge structure and it is set to support hypotheses that are good candidates for achieving the goal of the reasoning process.

3. Focusing on a conclusion

Focus is the final step of reasoning; that uses constraints and pragmatic criteria on the conclusion to check a candidate conclusion, or to select a conclusion if more than one candidate came out of the explanation phase. While Explanation generates, support and evaluates its hypotheses according to whether a conclusion makes sense and may be useful, **Focus** refines and adapts the candidate set in order to pick the best (i.e. most plausible) conclusion.

The three phase model may be applied to each reasoning method separately, in order to specify particular characteristics of each method. It may also describe an integrated reasoning process, by describing the top level process within the same three-phase model [1].

1.5 Objectives

The primary purpose of the present work is a comparative study of some reasoning methods which include (Rule Based Reasoning (RBR), Logic Based Reasoning (LBR), Frame Based Reasoning (FBR), Case Based Reasoning (CBR), Model Based Reasoning (MBR), and Fuzzy Reasoning (FR)). The sub goals of the work are specified as follow:

1. Providing a framework for describing and exploring all features of different reasoning schemes, with emphasis on benefits, drawbacks, applicability, criteria of choosing methods for a specific application.
2. Describing some criteria of valuation and comparing different reasoning schemes used as well to guide designers to select the appropriate method for a particular domain.
3. Providing a case studies that applied to two expert systems for problem solving that can meet the framework requirement. Both the expert systems should contain an expressive, extendible representation system for one or more method of reasoning.

1.6 Related works

The author in [13] presents the comparison between three reasoning methods, RBR, CBR and MBR .In RBR, knowledge was extracted from experts and encoded in rules. That was often difficulty to achieve. In CBR most (but not all) knowledge was in the form of cases. CBR needs adaptation rules and similarity metrics and more types of knowledge, but knowledge was easier to acquire.

Both MBR and CBR were developed as methods for avoiding reasoning from scratch. And both compose knowledge into large chunks and reason using the same chunks. The differences are ·mostly the content of the knowledge used and the conditions of applicability for each knowledge.

Davis &Hamscher [37] shows that the advantage of logic as a representation and reasoning mechanism is the potential for demonstrating the completeness of the inference procedure .While this can be useful; it does not imply the resulting process in a complete form.

Negnevitsky [30] discusses the conventional programs the process data using algorithms, or in other words, a series of well defined step-by-step operations. An algorithm always performs the same operations in the same order, and it always provides an exact solution. Unlike conventional programs, expert systems do not follow a prescribed sequence of steps. They permit inexact reasoning and can deal with incomplete, uncertain and fuzzy data.

Benjamin Kuipers [9] argues that Qualitative reasoning is one of the most vigorous areas in AI. Qualitative models are more able than traditional models to express states of incomplete knowledge about continuous mechanisms. Qualitative simulation guarantees to find all possible behaviors consistent with the knowledge in the model. This expressive power and coverage are important in problem-solving for diagnosis, design, monitoring, and explanation. In [18] qualitative simulation enables computers to simulate dynamical systems and to yield useful predictions even in those cases where only very rough and incomplete descriptions of systems exist.

According to **Luger** [19]: "One of the most subtle and critical issue raised by case based reasoning (CBR) which was the concern of defining similarity. Although the notion that similarity is a function of the number of features that two cases have in common is quite reasonable, it masks a number of profound subtleties. For example, most objects and situations have an infinite number of potential descriptive properties; case based reasoners typically select cases on the basis of a tiny retrieval vocabulary. Typically, case based reasoners require that the knowledge engineer define an appropriate vocabulary of highly relevant features. Although there has been work on enabling a reasoner to determine relevant features from its own experience, determining relevance remains a difficult problem".

Leake [14] identifies five main problems in AI that can be improved by CBR: knowledge acquisition, knowledge maintenance, increasing problem-solving efficiency, increasing quality of solutions, and user acceptance. Leake explains how CBR attempts to avoid such knowledge-related problems by assuming that there are few domain rules.

D'Ambrosio [8] extends qualitative perturbation analysis with fuzzy linguistic variables, applies the fuzzy number concept to dynamical system simulation and suggests that not only state variable values, parameter values, inputs and outputs, but also model and algorithmic structure can be made fuzzy.

According to **W.Xijun** [49], Case-Based Reasoning is different from other Artificial Intelligence approaches in the following ways:

- Traditional AI approaches rely on general knowledge of a problem domain and tend to solve problems on first-principles while CBR systems solve new problems by utilizing specific knowledge of past experiences.
- CBR supports incremental, sustained learning. After CBR solves a problem, it will make the problem available for future problems.

Karamouzis & Feyock [6] show that the integration of CBR and MBR enhances CBR by the addition of a model that aids the processes of matching, adaptation; and enhances MBR by the CBR capacity to contribute new links into the causality model.

1.7 Thesis outline

The topic of this thesis work is to define inference, reasoning methods and the identification of the role played by Artificial Intelligence technologies. We intend to explore all features related to reasoning schemes which process the previous features.

Therefore, the main theme of this thesis is to explore all features of different reasoning schemes with emphasis on benefits, drawbacks, applicability, and criteria of choosing methods for a specific application.

The work is divided into six chapters:

Chapter 1: Introduction: This chapter presents a preliminary study related to the thesis topic.

Chapter 2: Exact Reasoning Methods: This part of the thesis discusses the different quantitative reasoning methods namely by rule based reasoning, logic based reasoning, and frame based reasoning.

Chapter 3: Inexact Reasoning Methods: In this chapter gives other reasoning methods such as model based reasoning, case based reasoning, and fuzzy reasoning.

Chapter 4: Comparison of Reasoning Methods: This chapter is considered as a main contribution of my study. In particular, we will describe some criteria of valuation and comparison between different reasoning schemes which are used to guide designers to select the appropriate method for a particular domain.

Chapter 5: Design and Implementations of the Case Studies: This chapter intends to explore all the mentioned reasoning schemes by showing their applicability in different domains through a case study applied on selected methods.

Chapter 6: Conclusion: This chapter presents the results of the study, presents the scope for future research in this area.

Chapter 2

Exact Reasoning Methods

Reasoning methods, problem solving, is a very important field in artificial intelligence. The methods are used to give a complete representation and to support much of our knowledge-based reasoning or to provide information about the state of its problem solving and the explanations of the choice and decisions that the programs made.

In this chapter, a number of basic approaches will be used to represent knowledge and reasoning methods will be also considered such as: rule based reasoning, logic based reasoning, and frame based reasoning. These methods can be considered depending on the availability of data. Moreover, we will summarize the strengths and weaknesses of each approach to the problem solving domain.

2.1 Rule-Based Reasoning (RBR)

2.1.1 Introduction

A particular type of reasoning uses "if-then-else" rule statements, as mentioned above. Rules are simply patterns. These patterns could be searched via inference engine in the rules that match patterns in the data. The "if" means "when the condition is true", and "then" means "take action A" and the "else" means "when the condition is not true take action B with the rule".

A rule-based reasoner (or production system) typically contains three main components ordered as follows [34]:

1. A set of rules,

2. A control structure to guide the reasoning process, and
3. Working memory.

Rules have the general form IF {conditions} THEN {actions}. Working memory contains information about the current state of the problem solving process. Inputs may come from either the user or the reasoning that the system has already undertaken and stored as object-attribute-value triples. A rule is fired when the contents of working memory match either the condition set for that rule, or the action set for that rule. The control structure has the task of determining which rule to fire during a machine cycle. If there is more than one candidate rule, a set is formed and then the task of control structure will resolve the conflicting which determine which rule to fire first.

The strength of a rule-based reasoning is in a high abstraction level. Knowledge can be declared in a very comprehensive manner, making it possible and easily to verify the knowledge base (rule) with (human) domain experts; it also gives explanations for the given answers in the form of inference traces [34].

2.1.2 Rules as a reasoning technique

Any rule consists of two main parts: the IF subpart called the antecedent (primes or condition) and the THEN subpart called the consequent (conclusion or action).

The basic structure of the rule is:

IF < antecedent>
THEN < consequent >

A rule can have multiple antecedents joined by the keywords AND (conjunction), OR (disjunction) or a combination of both joins. It is a good habit to avoid mixing both conjunctions and disjunctions in the same rule. The consequent of the rule has multiple clauses:

```
IF     < antecedent 1>
AND    < antecedent 2>
OR     < antecedent 3>
            .

            .
AND    < antecedent n >
THEN < consequent 1 >
       < Consequent 2>
            .
            .
       < Consequent n>
```

Rules can represent relations, recommendations, directives, strategies and heuristics [30].

Relations

IF the 'fuel tank' is empty
THEN the car is dead

Recommendations

IF the season is autumn
AND the sky is cloudy
AND the forecast is drizzle
THEN the advice is 'take an umbrella'

Directives

IF the spill is liquid
AND the 'spill PH'< 6
AND the 'spill smell' is vinegar
THEN the 'spill material' is 'acetic acid'.

Strategies

IF the car is dead
THEN the action is 'check the fuel tank';
Step 1 is complete
IF Step 1 is complete
AND the 'fuel tank' is full
THEN the action is 'check the battery';
Step 2 is complete

Heuristics

IF the car is dead
AND the 'fuel tank' is empty
THEN the action is 'refuel the car'

2.1.3 Reasoning strategies in Rules

There are two basic reasoning methods where Rules can be forward-chaining, also known as data-driven reasoning, because they start with data or facts and look up for rules which can be applied to the facts until the goal is reached. Rules can also be backward-chaining, known as goal-driven reasoning; it started with the rules applied in a particular goal until a conclusion is reached [34].

2.1.3.1 Forward chaining (Data Driven Reasoning)

The first method starts with the known facts and applies rules in order to eventually reach the goal of conclusion [34]. The data driven approach, or forward chaining, uses rules similar to those used for backward chaining, in case of different inference processes. The system keeps track on the current state of problem solution and looks for rules which will move that state closer to the final solution [22].

For many problems it is not possible to enumerate all of the possible answers before the system selects the correct one. For example, configuration problems fall in this category. These systems might put components in a computer design circuit boards, or lay out office space. Since the inputs vary and can be combined in an almost infinite number of ways, the goal driven approach will not be working [34].

Suppose the database initially includes facts such as A, B, C, D and E, and the knowledge base contains only three rules:

Rule 1:	IF	Y is true
	AND	D is true
	THEN	Z is true
Rule 2:	IF	X is true
	AND	B is true
	AND	E is true
	THEN	Y is true
Rule 3:	IF	A is true
	THEN	X is true

Here we will discuss the above example via using forward chaining. Let us first rewrite our rules in the following form. Figure 2.1 shows how forward chaining works for this set of rules.

Rule 1: Y & D → Z
Rule 2: X & B & E → Y
Rule 3: A → X
Let us also add two more rules:
Rule 4: C → L
Rule 5: L & M → N

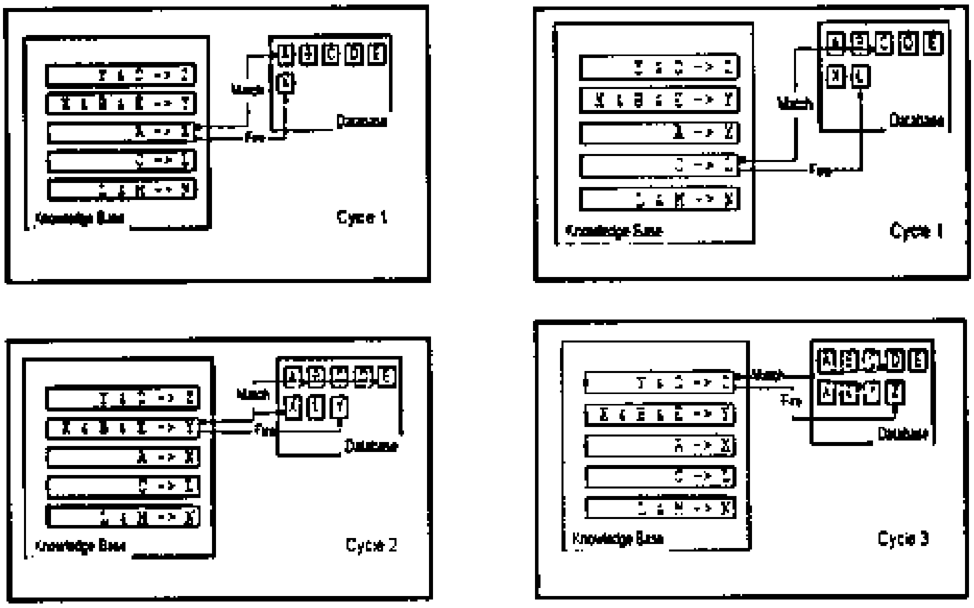


Figure 2.1[1] Forward chaining strategy.

The reasoning starts from the known data and proceeds with that data. When fired, the rule adds a new fact in the database. Any rule can be executed only once .The match fire cycle stops when no further rules can be fired.

- In the first cycle, only two rules, Rule 3: A → X and Rule 4: C → L, matches the facts in the database. Rule 3: A → X is fired first as the topmost one .The IF part of this rule matches the fact A in the database, then the THEN part will be executed and new fact such as X will be added to the database .Therefore, Rule 4: C → L is fired and the fact L is also placed in the database.
- In the second cycle , Rule 2: X & B & E → Y is fired because of the facts B,E and X are already in the database ,also the consequence fact Y is inferred and added to the database as well. This in turn causes Rule 1: Y & D → Z to execute, which will place the fact Z to the database. Finally the match fire cycles stop because of the IF part of Rule 5: L & M → N does not match all facts in the database and that causes Rule 5 not to be fired [30].

[1] This figure is borrowed from [30].

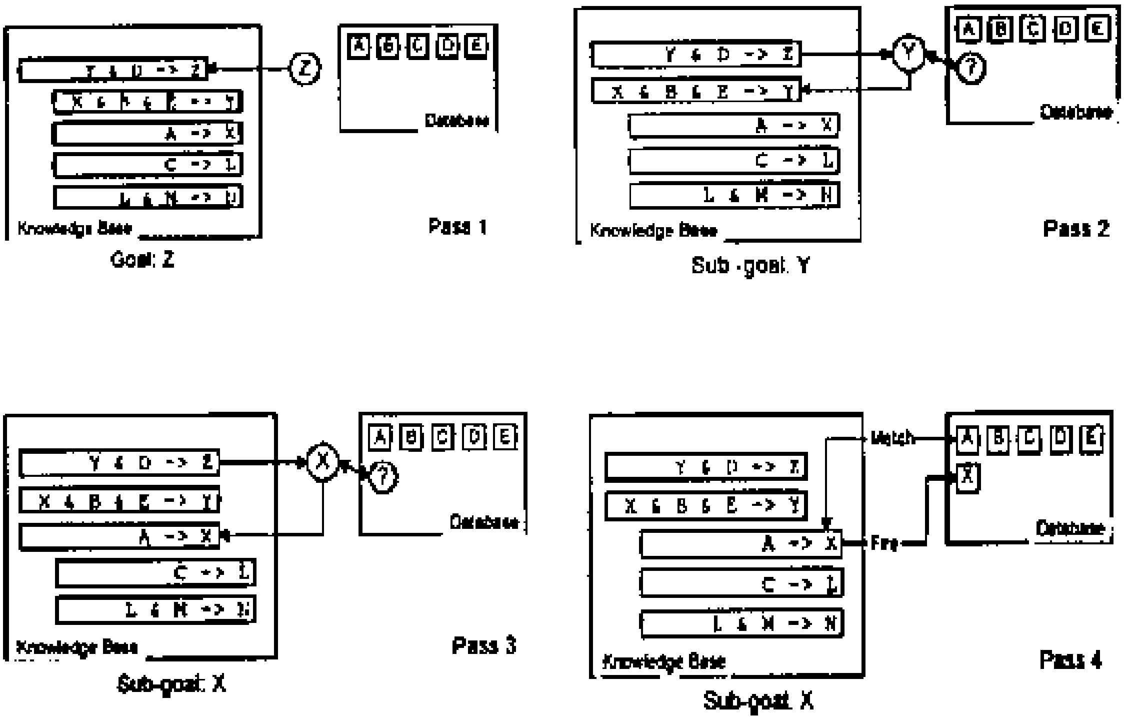
2.1.3.2 Backward chaining (Goal Driven Reasoning)

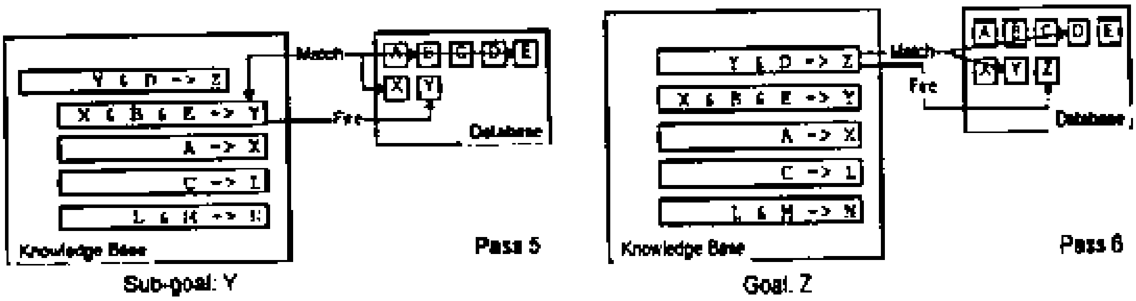
Backward chaining, or goal-driven reasoning, is an efficient way to solve problems that can be modelled as "structured selection" problems. Structured selection of the problems plays a main aim in the system by picking the best choice from many enumerated possibilities [22].

This method starts when the goal recursively selects rules that would deduce a (sub) goal until the set of goals is completely resolved by a given facts. Of course a bi-directional progress may be also possible [34].

Figure 2.2 shows how backward chaining works, using the rules for forward chaining example.

Rule 1: Y & D → Z
Rule 2: X & B & E → Y
Rule 3: A → X
Rule 4: C → L
Rule 5: L & M → N





Sub-goal: Y

Goal. Z

Figure 2.2[1] Backward chaining strategy .

- In phase 1, it attempts to infer the fact Z, by searching in the knowledge base to find the rule which holds the goal. In our case of the fact Z, its THEN part, find and stacks the Rule 1: Y & D → Z.
- In phase 2, set up the sub-goal of the fact Y and tries to determine it at the first glance in the database, but the fact Y won't be there. The knowledge base will be searched again for the fact Y in its THEN part, then locates and stacks Rule 2: X & B & E → Y. The IF part of Rule 2 consists of facts X, B and E and these facts have to be established.
- In phase 3, set up a new sub-goal, the fact X, and check the database for the fact X whether it fails or not, and if it fails it will search for the rule that infers X, and stacks Rule 3: A → X .It must eventually determine the fact A.
- In phase 4, find the fact A in the database, and Rule 3: A → X is fired and a new fact of X is inferred.
- In phase 5, return to the sub-goal fact Y and once again tries to execute Rule 2: X & B & E → Y. Facts such as X, B and E are in the database and thus Rule 2 is fired and a new fact of Y is added to the database.

[1] This figure is borrowed from [30].

- In phase 6, the system returns to Rule 1: Y & D → Z and tries to establish the original goal of the fact Z, the IF part of Rule 1 matches all facts in the database, Rule 1 will also be executed and thus the original goal is finally established as well [30].

2.1.4 The advantages of Rule Based Reasoning

1. The ability to use RBR, in a very direct fashion, requires experiential knowledge from human experts. Particularly it is important in the domains of relying heavily on heuristics to manage complexity and/or missing information.
2. Rules map the state space search with explanation facilities and support debugging.
3. The separation between knowledge and control simplifies development of the expert systems enabling an iterative development process where the engineer acquires, implements, and tests individual rules.
4. Good performance is possible with limited domains in case of applying intelligent problem solving on large amounts of knowledge. Moreover, expert systems are limited to narrow domains. However, there are many domains where a design of an appropriate system has proven extremely useful.
5. Good explanation facilitates, although the basic rule-based frame work supports flexibility, problem specific explanations. It must be mentioned that ultimate quality of these explanations depends upon the structure and content of the rules. Explanation facilities also differ widely between data- and goal-driven systems [19].

6. Natural knowledge representation: Expert systems usually explain the problem solving procedure with such expressions an this: 'in such-and-such situation, I do so-and –so'. These expressions can be represented quite naturally as IF-THEN rules.
7. Uniform structure: Rules structured via have a uniform of IF–THEN structure. Each rule is an independent piece of knowledge .The very syntax of production rules enables them to be self documented.
8. Dealing with incomplete and uncertain knowledge. Rules can be capable of representing and reasoning with incomplete and uncertain knowledge [30].

2.1.5 Disadvantages of Rule Based Reasoning

1. Often the rules obtained from human experts are highly heuristic in nature, and do not capture functional or knowledge based model of the domain.
2. Heuristic rules tend to be "brittle" and cannot handle missing information or unexpected data values.
3. Another aspect of brittleness of rules is tendency to degrade rapidly near the "edges" of the domain knowledge .Unlike humans, rule-based systems are usually unable to fall back on the first principles of reasoning.
4. Explanation function at the descriptive level only omitted theoretical expiations. These expiations follow from the fact that heuristic rules gaining much of their power by a directly associating problem symptoms with in the solutions without requiring deeper reasoning.
5. The knowledge tends to be very task based. The formalized domain knowledge tends to be very specific in its applicability [19].

6. Opaque relations between rules: Although the individual rules tend to be relatively simple and self-documented their logical interactions within the large set of rules may be opaque.
7. Ineffective search strategy .The inference engine applies an exhaustive search through all the rules during each cycle.
8. Inabilities to learn. Rule reasoning cannot automatically modify its knowledge base, or adjust existing rules or adding new rules instead [30].

2.2 Logic Based Reasoning (LBR)

2.2.1 Introduction

The approaches to the problem solving in our logic based presentation will be predominant into logic. These pieces of knowledge are explicitly used in reasoning to define a very differently depending on the context of the understanding of reason as a form of knowledge. The Logical definition is the act of using reason to derive a conclusion from certain premises, using a given methodology. We will use the two common explicit methods to reach a conclusion, which are deductive and inductive reasoning [47].

More over, we will discuss some nature representation languages, and some logical languages used in particular. We will also explain in detail the connection between the logical language and the reasoning mechanism that goes with it. Representation and reasoning support the operations of knowledge based system where the knowledge representation language is defined by two aspects described as follows:

- The syntax of the language describes the possible configurations that can constitute sentences.
- The semantics determines the facts in the word to which the sentences refer: each sentence that makes a claim about the word.

We will call the language as a Logical language if the syntax and semantics of the language are defined precisely. Also the syntax and semantics can derive reasoning mechanism for a system that uses the logical language that explains how these syntax and semantics stand for.

We recall the semantics from the language. The language determines the facts to which a given sentence refers (see Figure 2.3). It is important to distinguish between facts and their representations. Facts are parts of the word .We cannot put them all inside a particular computer, where all reasoning mechanisms must operate on representing of facts, rather than on the facts themselves. In general definition, reasoning must be a process of constructing new configurations from old ones. But proper reasoning should ensure that the new configurations represent facts that are actually followed from the facts that the old configurations represent.

The connection between sentences and facts is provided by the semantics of the language. The property of a single fact followed from some other facts is mirrored by the property of one sentence being entailed by some other sentences. Logical reasoning generates new sentences that are entailed by existing sentences.

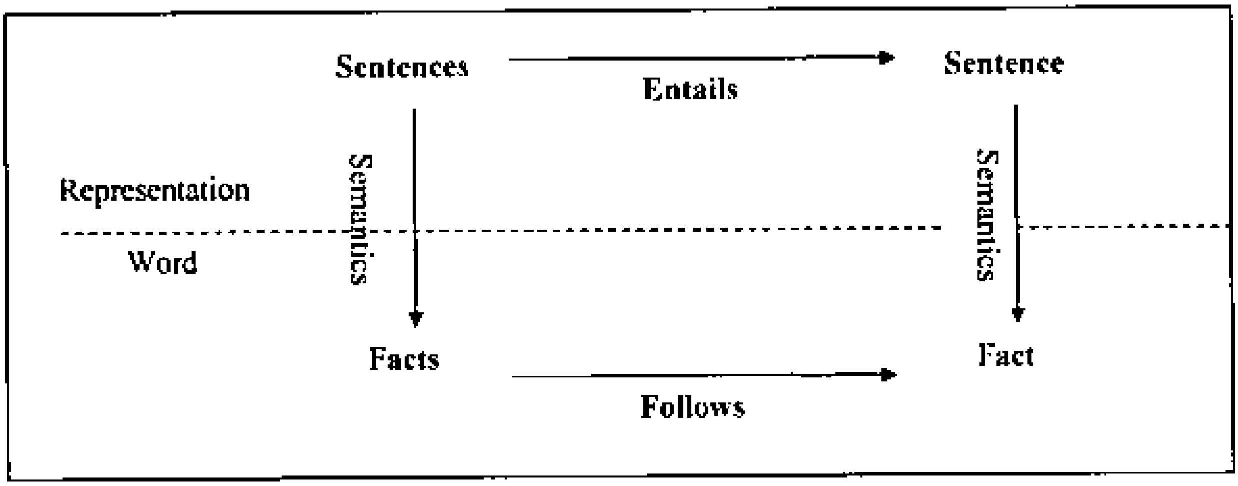


Figure 2.3[1]The connection between sentences and facts.

Logical reasoning is a process that implements the entailment relation between sentences [44].

We began our exploration of logical reasoning with the statement that reasoning diagram is used to process for connecting *experience, data, axioms and premises on the one hand side, to conclusions, judgments, estimates and inferences on the other* hand side, which schematized reasoning in terms of the following diagram:

[1] This figure is borrowed from [44].

This is an important lesson that we have learnt from our examination of reasoning systems when such premise-conclusion pairings have to be sanctioned by a set of rules of inference that we consider acceptable. Therefore, one of the central ingredients of reasoning is a set of acceptable rules of inference [28].

A logic based inference procedure begins with a set of axioms and a theorem that proved (or refuted). If a logical representation is completed, then all the logical consequences that follow from the axioms are derivable (as theorems) [40].

The discipline of mathematical logic, in particular symbolic logic, is a subset of discipline in discrete mathematical form. A systematic study of symbolic logic dates back to Aristotle. Therefore, subject of symbolic logic is only one component of AI and has some limitations. It is presented for several reasons first of all. In the first place, logic is familiar to most engineers and computer scientists. The second reason is, more importantly, symbolic logic facilitates knowledge- manipulation strategies with solid mathematical underpinnings.

The arguments believed somewhat true, are not sufficient reason to dismiss logic as useless for AI system development. It seems to be a little reason to develop AI systems that formulate illogical conclusions.

The broad objective is to consider mathematical formalisms of logic that could develop mechanisms to represent and manipulate entities known as statements, facts or stretching things in knowledge. Since statements have deeper meaning than groups of words or logic values in this stage the representations are considered as "shallow".

which do not allow the explicit integration of "common sense," nor do they invoke broad concepts or well-known scenarios that is to the statements [40].

In traditional propositional logic we combine unrelated propositions into an implication, which do not assume any cause or effect relation to exist. Logical reasoning is the process of combining given propositions into other propositions, which is doing this over and over again [35].

2.2.2 Reasoning strategies in logic

Typically, logic based inference procedure begins with a set of axioms in theorem to be proved (or refuted). If a logical representation is completed then all logical consequences that follow from the axioms are derivable (as theorems) [40].

The fundamental axioms of traditional propositional logic are [35]:

1) Every proposition is either true or false, (laws of contradiction and excluded middle). 2) Expressions given to defined terms are propositions. 3) The truth tables for conjunction, disjunction, implication, equivalence, and negation are used to derive many interpretations of preceding operations which can prove relationships between them.

To summarize the reasoning strategies in logic, we can assume that logic consists of the following [44]:

1. A formal system for describing state of affairs, consisting of:

- The syntax of the language, which describes how to make sentences.
- The semantics of the language, which states the systematic constraints on how sentences relate to state of affairs.

2. The proof theory which is a set of rules for deducing the entailments of a set of sentences.

2.2.2.1 Inference rules

Inference is a typical step in logical derivation to proof a theorem that is a logical derivation of a theorem from a theory that is assumed to be consistent and completed [1]. Inference is an act or process of deriving conclusions based solely on what one already knows. Especially in mathematical logic a rule of inference is a sub scheme for constructing valid inferences. These schemes establish syntactic relations between a set of formulas called premises and an assertion called a conclusion.

In the setting of formal logic (and many related areas), rules of inference are usually given by a standard form as follows [47]:

Premise # 1
Premise # 2
...
Premise # n
Conclusion

Prominent examples of rules via inference in logical reasoning depend on types of reasoning that can be used. Descriptions of these types of reasoning are discussed in the next section.

2.2.3 Types of Logic Based Reasoning

2.2.3.1 Deductive Reasoning

In deductive reasoning, true premises, must follow it true conclusion and it cannot be false. This type of reasoning is non-ampliative , it does not increase one's knowledge base , since the conclusion is inherent to the premises . Deduction is based on universal inference rules which are like modus ponens that is a very common form of reasoning, often abbreviated to MP, and takes the following form [47].

If P, then Q.
P.
Therefore, Q.

The argument form has two premises. The first premise is the "if-then" or conditional claim, namely that P implies Q. The second premise is that P, the antecedent of the conditional claim, is true. From these two premises it can be logically concluded that Q, the consequent of the conditional claim, must be true as well.

Here is an example of an argument that fits the form modus ponens:

If today is Tuesday, then I will go to work.
Today is Tuesday.
Therefore, I will go to work.

Therefore, it seems natural to apply logic-based reasoning mechanisms in knowledge-based systems in forward-chaining manner, i.e., to start a given knowledge base and/or with an expression it must determine the logical consequences [47].

Deductive reasoning is a strong method and its applicability for reasoning is limited to assumptions that do not generally hold for real world situations: A world - typically assumed to be closed - where all phenomena hold (for all properties of concepts) are either true or false, and a consistent domain theory by which any proposed fact (i.e. proposition) may be proved to be true or false. However, even if knowledge in general does not fit into the requirements of deductive reasoning, but some parts or types of knowledge may do so, in which case much is gained if deductive methods can be applied to that subset of knowledge [1].

2.2.3.2 Inductive Reasoning

In inductive reasoning, when the premises are true, then the conclusion follows with some degree of probability. This method of reasoning is implicative as it gives more information about the contained of the premises themselves. A classical example for inductive reasoning is [47]:-

If P,
Then Q.

If the sun rose in the east every morning untill now,
Therefore the sun will rise in the east also tomorrow.

An inductive conclusion should therefore be supported by a justification. An inductive inference step may be expressed as:

Example:

All observed crows are black.
therefore
All crows are black.

Inductive reasoning particularly is related for learning of general knowledge from observing examples (instances) and counter-examples. On the other hand, as we know deductive Reasoning is the opposite of Inductive Reasoning which means that deductive reasoning starts with a general statement. Moreover in Deductive Reasoning we can go from the general statement to a particular fact or conclusion [1].

2.2.3.3 Abductive Reasoning

A third method of reasoning is called abductive reasoning, or inference to the best explanation. This method is more complex in its structure and can involve both inductive and deductive arguments. The main characteristic of abduction is that it is an attempt to favor one conclusion above others by either attempting to falsify alternative explanations, or showing the likelihood of the favored conclusion given a set of more or less disputable assumptions [47]. A simplified way to express abductive reasoning is in the follows example:-

If all cats die.
Socrates is dead.
Therefore, Socrates is a cat.

An abductive reasoning does not guarantee the truth of the outcome based on the truth of the inputs. There may be a lot of reasons for abdominal pain, appendicitis is only one possibility. An abductive conclusion therefore has to be supported by an explanation that leaves the particular conclusion as a best choice. Abduction is also called plausible inference that is closely tied to generation and evaluation of explanations [1].

Abduction is considered to be the kind of inference typically made in diagnosis (at a top level, at least) where a set of symptoms infer a fault by being explained by the fault, i.e. by generating a plausible argument that supports a particular fault and weakens of competitive fault hypotheses.

Abduction and induction are fundamentally different from deduction, since they do not guarantee that the inferred results when it is true. Induction is similar to abduction in the form of the truth of premisses where does not guarantee the truth of the conclusion.

The advantage of inductive and abductive reasoning methods is that they are not necessarily subject to the limitations of first order logic where the word may be viewed as a more realistically as an open, dynamically changing environment. Concepts may be described in terms of typical features or default values which may be inherited to more specific concepts, and replaced by local values if these later become available. A disadvantage is that the particular inference methods needed have to be defined by the system developer, leading to a complex and often unclear definition of knowledge semantics.

Based upon the three inference types described, a range of reasoning methods that has been developed in AI. An inference method specifies a way to derive new information from existing knowledge in an actual representation system. Inference methods may be put together to formalize inference structures in a goal-oriented, complex inference structure [1].

2.2.4 Advantages of Logic Based Reasoning

1. High representational adequacy. The formalism is very general. It is independent of the type of knowledge represented as a modular knowledge. The Knowledge is represented in distinct units, where each fact being true and independent of other represented facts.
2. High inferential adequacy. Many general inferential procedures exist in knowledge.
3. Very low inferential efficiency. It ignores the problem of relevance of certain facts in certain given situations. It tries to use all the applicable facts. It uses only general heuristics as, for instance, preferred usage of simpler facts. Another cause of inferential inefficiency is the generality of the inference procedures that do not take into account that specific knowledge to the represented domain.

4. The acquisitional efficiency is high. New knowledge could be easily added to the system because of the modularity of the representation which simplifies very much the problem of integration of the new knowledge into the existing knowledge [45].
5. Logic-based reasoning formalism is that it provides backward deductive reasoning procedures; as a logic-based user modeling system that naturally offer the option of employing backward inferences for query answering [47].
6. Logic allows us to get across all the important points about what logic and how it can be used to perform inference that eventually results from action. The main problem is that there are just too many propositions to handle.
7. Different logics make different commitments about what the world is made of and what kinds of beliefs we can have regarding facts.
8. Logics are useful for the commitments. They make the lack of commitment that gives the knowledge base writer more freedom.
9. Defining logic that fits for the use of logic which derives false or contingent conclusions from premises that are not logically necessary. A definition in terms of logical truth may seem to exclude that possibility. However, when we ask what was the proof conclusively demonstrates that are not simply able to tie the conclusion. Assured conclusion follows from the premises and hence would be true if they were be true The deductive content is expressible as a single proposition (usually an entailment) of that we can expect to be logically true [17].

2.2.5 Disadvantages of Logic Based Reasoning

1. Humans do not always reason by making logical inferences.
2. Logic is too rigorous, and inflexible, to be use in all AI problem domains.

3. In logic-based representations, facts are stored as axioms .An axiom is a statement that is always true.
4. A disadvantage of identifying logic with methods is a possible difficulty in encompassing areas where there are no methods. If a definition of "logical truth" is desired (possibly in connection with establishing the correctness of the methods) some difficulty may be expected if starting from this point [17].
5. Indivisible propositions: Propositional logic assigns truth values to entire propositions. Propositional logic is not capable of analyzing the components of that proposition.
6. Incapable of dealing with quantifiers: Another major limitation of propositional logic is that it is incapable of dealing with quantifiers such as "all" or "some."
7. The combinatorial explosion is one of the critical problems in logic based reasoning, however, if there are many antecedent conditions required, for example, to classify a phenomenon into its appropriate sociological category. This may lead to a combinatorial explosion of possible combinations of conditions [45].

2.3 Frame Based Reasoning

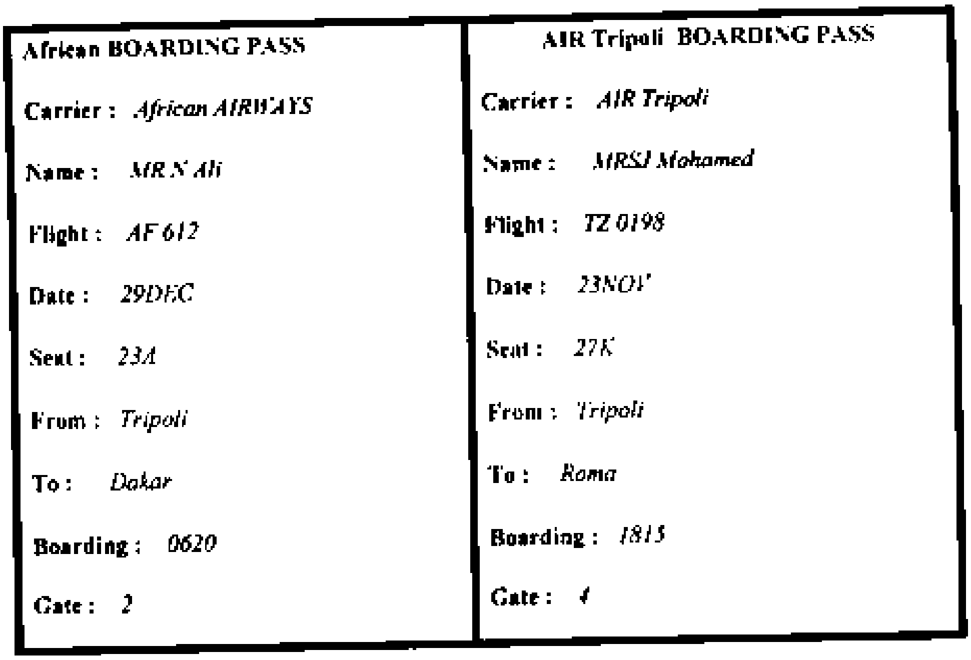
2.3.1 Introduction

In which we introduce frames as one of the common methods used for representing knowledge and reasoning. Frames are a common way of representing information, first proposed by Marvin Minsky in the 1970s.

A frame-based representation facility contributes to a knowledge base system's ability to reason and can assist the system designer in determining strategies for controlling the system's reasoning [39]. A frame is a data structure with typical knowledge about a particular object or concept [30].

A frame is an abstract data type represented in a schema, or structure that governs where they exist in hierarchy to one another. Boarding passes shown in figure 2.4 represent frames with knowledge about airline passengers. Where both frames and knowledge have the same structure.

Frames can contain properties and methods that describe the frame. A frame that has information filled in it is called an instance of that frame. The development of object-oriented programming practices has helped to foster new functionality in frame-based expert systems. Frames are a natural way of representing real-life objects, while object-oriented programming was developed for the same purpose. It does this by having a number of attributes or slots, each of which can be given as a particular value. Obviously, it is not necessary to use frame-based reasoning or any other tool.



African BOARDING PASS	AIR Tripoli BOARDING PASS
Carrier : *African AIRWAYS*	Carrier : *AIR Tripoli*
Name : *MR N Ali*	Name : *MRSJ Mohamed*
Flight : *AF 612*	Flight : *TZ 0198*
Date : *29DEC*	Date : *23NOV*
Seat : *23A*	Seat : *27K*
From : *Tripoli*	From : *Tripoli*
To : *Dakar*	To : *Roma*
Boarding : *0620*	Boarding : *1815*
Gate : *2*	Gate : *4*

Figure 2.4 Boarding-pass frames.

Each slot in a frame contains a value, either directly or indirectly. Specifically, the contents of a slot can be one of the following:

1) A pure value, normally either a string, Boolean, or number. Numbers can include real numbers, integers, or enumerated values.

2) A pointer to another frame that contains a value to be used. This might be a method for providing inheritance within the frame structure.

3) A facet, which is a means of providing extended knowledge about an attribute of frame. There are three types of facets:

- Value facets, prompt facets, and inference facets.
- Value facets can be used to compute values that should be placed in the slots.
- In graphic applications, the value facet might be a procedure for drawing a figure associated with the slot.
- Prompt facets can be used to allow the user to input values.

- Inference facets apparently provide a break mechanism for debugging rule-based systems.

In addition, slots can contain some additional information.

- A default value, which is taken to be true when no evidence is present to indicate another specific value.
- A range for the slot value, which is used to constrain the values that can be placed in the slot.
- One of several types of procedures, including procedures to be activated when the value of the slot is changed and procedures to be activated when the value is needed, called WHEN CHANGED and WHEN NEEDED procedures respectively. A WHEN NEEDED procedure allows for evaluation – the value is not computed when the frame is created, but only when the value is used by a rule. These procedures are often called demons [30].

There are different kinds of slots:

- Attribute-value pair: can have different kinds of filler, like a primitive of the language or a pointer to another frame in the knowledge base.
- Restrictions on the values of a slot, such as logical connectives, functions or predicates, an instance of another class-frame.
- A procedure, which can be a servant (if it is possible to compute the value when needed) or a demon (which is called whenever the slot value changes). This type of representation is called procedural attachment.
- A superclass slot (for class frames) or a member-of slot (for instance frames) [21].

Facet is a means of providing extended knowledge about an attribute of the frame. Some situations can be seen from more than one perspectives. This can be represented by allowing super links to connect to more than one other frame. In Minsky's original conception, some slots cannot be overridden by other values when inheriting. This property is mostly lost in other frame systems [21].

Frame-based representation is a development of semantic nets and allows us to express the idea of inheritance.

As with semantic nets, a frame system consists of a set of frames (or nodes), which are connected together by relations. Each frame describes either an instance (an instance frame) or a class (a class frame).

Thus far, we have said that instances are "objects" without really saying what an object is. In this context, an object can be a physical object, such as a color, a shape, and a place, a situation, and a feeling. Frames are thus an object-oriented representation that can be used to build expert systems.

Each frame has one or more slots, which are assigned slot values. This is the way in which the frame system network is built up. Rather than simply having links between frames, each relationship is expressed by a value being placed in a slot [7].

2.3.2 Reasoning with frames

In order to retrieve information, the type of reasoning in frame systems is often recognition.(the comparison of new objects by known ones). There are three common forms of reasoning used by frames: matching, inheritance, and the use of procedural rules embedded in frames.

2.3.2.1 Matching

Matching is the first step in the reasoning process to discover which frames can be applied to the current situation. Matching in frame systems is more difficult than that in production rule systems, because of the following reasons:-

(1) Frames are complicated structures,

(2) Slots can have default properties, and

(3) Perfect matches are rare (whereas in the other systems, it is clear-cut when one has a match or not).

Once a frame is proposed to represent a situation, the matching process tries to assign values to each frame's terminals, consistent with the markers at each place. The matching process is partly controlled by information associated with the frame (which includes information about how to deal with surprises) and partly by knowledge about the system's current goals. There are importance uses for the information, obtained when a matching process fails; it can be used to select an alternative frame that better suits the situation.

The reason matching is so powerful because it permits frames to represent a great deal of information very efficiently. For example, a frame representing specific piece of equipment in warehouse knowledge cloud by default if we assume that the equipment is assembled in a proper working order. Default knowledge corresponds to the most common state in the population of items represented, in a most effective manner [21].

2.3.2.2 Inheritance

One of the most powerful forms of reasoning in frames is the inheritance. Specialized frames can "inherit" properties of more general frames. This permits great economies of representation since many specialized frames can inherit the properties of a few more general frames by default without requiring those properties to be defined in each instance. Only when the default properties are not inherited there is a need for an explicit statement in the knowledge base.

For instance: computer system will try to return a value for a slot even when it is not provided. The super classes of the frame will searched. This can happen with breadth-first search or depth-first search or with other searching methods. Depth-first search is more efficient than breadth first search; breadth-first search always return the most specific value. Another advantage of using depth-first search strategy is that depth-first end up to join [21].

Inheritance is the process of a sub frame taking on the characteristics of the parent frame in the tree. Inheritance does not always work in the proposed manner. In the child frames, there is a possibility of a drastic difference from the parent in the tree. In the canine in the tree frame, there may be a property of number of legs. This should be a set to four legs. However, there is the possibility of having an instance node of dog that only has three legs. Child frames can overwrite inherited properties explicitly [46].

2.3.2.3 Using Procedural Rules in Frames

Frames can be used to represent both declarative knowledge (e.g., facts such as who, what, where, when, etc.) and procedural knowledge. Procedural knowledge is often represented by production rules where procedures are attached to specific slots or entire frames. A common method is to use demons--procedural rules executed only if certain conditions are found to be true. Demons include rules for determining the value of a frame is unknown and needed rules to be fired if the value in a common slot changes [21].

In general, a demon has an IF-THEN structure it is executed whenever an attribute in the demon's IF statement changes its value. Rules often use pattern matching clauses that containing variables that are used for finding matching conditions among all instance-frames.

Most frame based reasoning uses two types of methods: WHEN CHANGED and WHEN NEEDED .A WHEN CHANGED is executed when the value of its attribute change. Where as the WHEN NEEDED is executed when information associated with a particular attribute is needed for solving the problem, but the attribute value is undetermined.

Demons and methods are very similar, and the two terms are often used as synonyms. However, methods are more appropriate if we need to write complex procedures. Demons on the other hand are usually limited to IF-THEN statements [30].

2.3.3 Advantages of Frame-Based Reasoning

1. A frame based representation is a conceptual object related to a frame that can be easily accessed by looking in a slot of frames (where there is no need, for example, to search the entire knowledge-base) [24].
2. Frames are arranged in a hierarchical manner such that they can inherit relationships from other frames, and allow the representation of different knowledge types.
3. Frames facilitate faster searches of the knowledge base through the concise and compact representation of information.
4. Frames permit the representation of inheritance relationships among objects. Inference can be efficiently supported in the frame based reasoning, where inheritance is quite natural [3].
5. Frame-based reasoning captures the way in which experts think about their domain. Frames are simple and easy for humans to understand.

6. Domain experts use comparison with known things for describing new ones.
7. The taxonomic structure of the inference mechanism (inheritance) for many applications in frame systems strike the right balance between expressive power and efficiency.
8. Default reasoning, information, if it exists, and stored with the object and thus retrieved first. It can be decided that this information is present, so undecidability does not arise here, although we will sometimes have to conclude that we do not know the answer [21].
9. Frames offer us a powerful tool for combining declarative and procedural knowledge, although they leave the knowledge engineer with difficult decisions to make about the hierarchical structure of the system and its inheritance paths [30].
10. Easier structuring, abstracting which means that the knowledge can be structured and organized hierarchically to inherit procedural or declarative knowledge in less number of cases. Faster retrieval reduces complexity.
11. Appropriate classification in a single class and easy classification of knowledge can be used to contain default values that can constrain allowed values clearly documented [30].

2.3.4 Disadvantages of Frame-Based Reasoning

1. Procedural attachment: It provided a translation for the procedures into logics or by giving up the enterprise to handle the procedural aspects in another way.
2. It is not clear how one could distinguish between a member and an owner of the slots. Owner's slots in the class frames cannot be dealt with in the frame work.

3. Disjunctions between values of different slots cannot be expressed in frame-based reasoning.
4. Existential knowledge is not in the scope of the universal quantifier that a problem such as a murderer in the detective story, who is unknown. In a frame based reasoning, an instance frame for the murderer would be distinct from all suspects' frames).
5. Not all frame-based in Knowledge Representation languages will be used only as a default inheritance [21].
6. Negation cannot be represented, and disjunction cannot easily be represented as well. That means qualification is not a part of the language and the difficulty modeling of n-ary relations will appear as well.
7. Problems encountered when we building a frame systems if the following cases hold:
   - Classes are not defined at useful levels of abstraction.
   - Overloading of concept definitions has been done [21].

9. Difficult to define appropriate frames – requires a lot of domain knowledge, to deal with multiple faults if they are not independent.
10. It could be inefficient at runtime which lead to "procedural fever" and Highly iterative design process.
11. The design must be created carefully to suitable taxonomies [21].
12. Frame based systems hardly distinguish between essential properties (those that an instance must have in order to be considered as a member of a class) and accidental properties (those that all the instances of a class just happen to have) [30].

Chapter 3

Inexact Reasoning Methods

The process of reasoning with uncertain, vague, and incomplete information is known by several different names depending on the emphasis of the authors and the types of uncertain information under consideration. These names include evidential reasoning, approximate reasoning, uncertain reasoning, and probabilistic reasoning. There have recently been a number of heated debates in the literature which argue strongly for one method over one or more other methods. The implication of these articles is generally that one of these methods is inherently superior for representing all types of uncertainty in expert systems.

Although it might be most convenient to identify one technique that could be used consistently for the expert system. We do not have a priority limited to ourselves for using a single method of representation for all uncertain information in the domain. We will consider a limitation using multiple methods if that approach has clear advantages and also if techniques can be developed for integrating the different representations.

In this chapter we consider methods for reasoning when our knowledge is unreliable or incomplete. Also we are looking at how we can use previous experience to reason up current problems.

3.1 Case-Based Reasoning (CBR)

3.1.1 Introduction

Case-Based Reasoning (CBR) is a problem of solving paradigm that solves a new problem by remembering a previous similar situation and by reusing information and knowledge of that situation [2]. More specifically, CBR uses a database of problem-solving to resolve new problems. The database can be built through the Knowledge Engineering (KE) process or it can be collected from previous cases [49].

In a problem-solving system, each case would describe a problem and a solution to that problem. The problem solving methods solve new problems by adapting relevant cases from the library [2]. CBR can mean different things depending on the intended use of the reasoning: adapt and combine old solutions to solve a new problem, explain new situations according to previously experienced similar situations, critique of new solutions based on old cases, reason from precedents to understand a new situation, or build a consensued solution based on previous cases [42].

CBR is both, a model of human reasoning, and also a method used to create "intelligent" systems. As a model of CBR which is based on a number of key observations stated based on the following:

- The first observation is the fact that most of the problems decision maker that has to handle it isn't unique. When we are encountered with a new problem the novices and the experts often reason by analogy that companies the current situation with earlier problems encountered.
- The second observation is that when solving new problems, people typically reuse solutions from similar problems, adapting the solution to suit the current

circumstances. In short term, the CBR model of human reasoning suggests that the people reason by analogy, remembering past experiences [32].

3.1.2 Characteristics of the CBR-Based Reasoning

We introduce some of the most important aspects mainly the benefits that CBR can provide as a reasoning technique which include the following:

1. It provides many benefits over other AI-based approaches for easing knowledge acquisition, easing knowledge maintenance, and increasing user acceptance.
2. CBR is an approach for developing knowledge-based systems.
3. CBR is an approach for learning from experience (examples, cases).
4. CBR is an explanation model for human problem solving, learning and increasing problem solving efficiency.
5. CBR is a very natural approach for the development of knowledge especially in the context of teaching and tutoring.
6. CBR is a knowledge based extension of the nearest neighbor classifier paradigm known from pattern.
7. CBR is a holistic system approach where it is different form many approaches related to machine learning community that are more oriented towards developing optimized algorithms for specific problems and, therefore, do not consider requirements (for building software systems) in knowledge/software engineering very much.
8. CBR is an open environment for integrating different kinds of techniques.
9. Knowledge based information retrieval.
10. A case base becomes useful in the first case to increase quality of solutions.

11. A case base captures knowledge easily (there is no need to discover complex interrelationships between cases).
12. Case bases are understandable. It is logical and easy to follow.
13. CBR augments human capabilities (comprehensive case storage and tracking).
14. The case format is flexible and may be modified over time without impacting the methodology.
15. CBR is amenable to being incorporated into a knowledge management process.
16. The CBR cycle is similar to experience reuse in project management.
17. Organizational learning is similar to the CBR cycle and so can be supported by CBR technology [41].

3.1.3 The Basic CBR Cycle

The CBR process can be represented by a schematic cycle, as shown in Figure 3.1. We described CBR typically as cyclical process comprising in the following steps [2]:

1. RETRIEVE

The CB Reasoner searches the knowledge base to find the most approximate case to the current situation.

2. REUSE

This process includes use the retrieved case and adapting it to a new situation. At the end of this process, the reasoner might propose a solution.

3. REVISE

The proposed solution could be inadequate this process can correct the first proposed solution.

4. RETAIN

The new solution as a part of a new case, this process enables CBR to learn and create a new solution should be added to the case based.

It should be noted that the RETRIEVE process in CBR is different from the process in a knowledge base. If we want to query data, the database only retrieves some data using an exact matching while a CBR may retrieve data using an approximate matching.

As shown in Figure 3.1, the CBR cycle starts with the description of a new problem which can be solved by retrieving previous cases and reusing solved cases. If possible, giving a suggested solution or revising solution, retaining the repaired case and incorporating it into the new case based. This cycle will rarely occur without human intervention which is usually involved in the RETAIN step. Many application systems and tools act as a case retrieval system, such as some help desk systems and customer support systems [49].

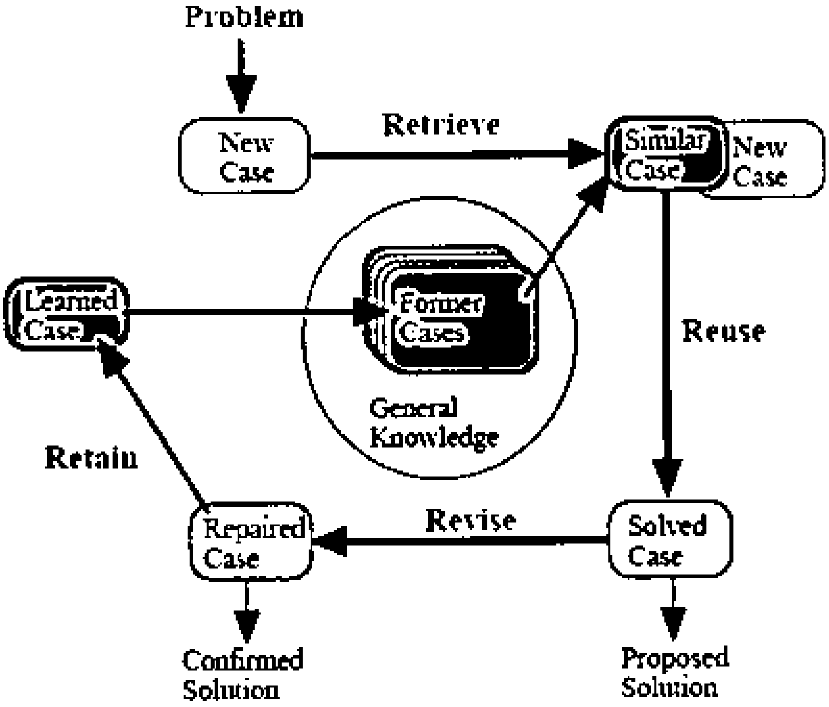


Figure 3.1 The Case-Based Reasoning cycle developed by Aamodt

The quality of the new case(s) extracted by the CBR depends upon some of the following criteria [10]:

a) The usefulness of the case(s) extracted and selected;

b) The ease of use this or (these) case(s);

c) The validity of the reasoning process;

d) The improvement of knowledge through experience.

When setting a CBR one has to take into account some design decisions or static problems such as follow:

a) How to describe the domain problems?

b) Which will be the case structure?

c) Which will be the Case Library structure?

d) How to deal with the missing information problem?

e) CBR will be the criterion for indexing the Case Library?

f) How to assess similarity between cases?

3.1.4 Main types of CBR methods

According to the methodologies mentioned in [2] there are four different types of CBR methods, although they share similar features, each of them is more appropriate for a particular type of problem. The different four types are case of exemplar based reasoning; instance based reasoning, memory-based reasoning, and analogy-based reasoning.

The term CBR is often used in both first as a generic one for several types of more specific approaches, and the second as a single approach. In its specific meaning, typical case usually has a certain degree of richness of information contained in it, and certain complexity with respect to its internal organization. General background knowledge is used during reasoning process in order to modify or adapt a retrieved solution when applied in a different problem-solving context [23].

3.1.4.1 Exemplar-based reasoning

This category of CBR employs nothing more that classification of a new case based on how previous cases were classified [11]. In the exemplar view, the concept is defined extensionally, as the set of its exemplars. CBR methods that address the learning of concept definitions.

In this approach, solving a problem is a *classification task*, finding the right class for the unclassified exemplar. The class of the most similar past case becomes the solution to the classification problem. The set of classes constitutes the set of possible solutions. Modification of a solution found is therefore outside the scope of the CBR methods [2].

Using the case base, the learner could look for similar moves with similar attributes and see how they were classified. Based upon how these were classified, the learner could then classify the new more similarly [11].

3.1.4.2 Instance-based reasoning

Instance-based reasoning (IBR) can be considered to be a type of exemplar-based reasoning in highly syntax-dependent problem areas [23]. This is a specialization of exemplar-based reasoning into a highly *syntactic* CBR-approach.

To compensate for lack of guidance from general background knowledge, a relatively large number of instances are needed in order to close in on a concept definition. The representations of the instances are usually simple, since a major focus to study *automated learning* with no user in the loop. Instance based reasoning serves to distinguish their methods from more knowledge-intensive exemplar-based approaches [2].

This type of CBR system focuses on problems in which there are a large number of instances which are needed to represent the whole range of the domain and where there is a lack of general background knowledge. The case representation can be made with feature vectors and the phases of the CBR cycle are normally automated as much as possible, eliminating human intervention [23].

3.1.4.3 Memory-based reasoning

This approach emphasizes a collection of cases as a *large memory*, and reasoning as a process of accessing and searching in this memory. Memory organization and access is a concern of the case-based methods. The utilization of parallel processing techniques is a characteristic of these methods which distinguishes this approach from the others [2].

In memory-based reasoning, decisions are based upon memories of specific events versus that of using relationships or rules built up from experience. The remindings are syntactic in nature versus the rich semantic nature of alternative CBR methods. Thus, computational processes that get the learner of the most relevant cases based upon a matching of the index attributes are the ones employed in this method [11]. The memory-based reasoning hypothesis is that reasoning may be accomplished by searching a database of worked problems for the "best match" to the problem at hand. Memory-based reasoning degrades gracefully when it cannot come up with a definitive answer to a problem: It may respond that no answer is possible when giving one or more plausible answers, or asking for more information.

The aim of memory-based reasoning is to fill in the goal fields of a target record by retrieving records from a database. The basis of the method is finding the dissimilarity between the target record and each of the data records. The dissimilarity measure is calculated by assigning a weight to each field and the value difference measure to each value occupying those fields [12].

3.1.4.4 Analogy-based reasoning

Analogy -based Reasoning (ABR) is similar to CBR except that the learner solves the new situations with past experiences from a different domain. Transfer of knowledge is a major issue in the education of a learner. For example, we often use cross-domain analogies in education. In solving new computing applications we often look to the biology and the human cognition to help creating new solutions. Providing the learner with these cross-domain analogies is a quite complex task, but it offers a tremendous area of growth in the field [11].

Analogy-based reasoning reflects natural human reasoning that is based on the ability to associate concepts and facts by analogy [5]. Research on analogy reasoning is therefore a subfield concerned with mechanisms for identification and utilization of cross-domain analogies. The major focus of study has been on the *reuse* of a past case, which is called the mapping problem: Finding a way to transfer, or map, the solution of an identified analogue where we used to called source or base to the present problem which is called target [2].

3.1.5 Advantages of Case-Based Reasoning

1. The ability to encode historical knowledge directly. In many domains cases it can be obtained from existing case histories, repair logs or other sources to eliminate the need for intensive knowledge acquisition with human expert.

2. Allows shortcuts in reasoning. If an appropriate case can be found then new problems can often be solved in much less time than it would take to generate a solution from rules or models.
3. Extensive analysis of domain knowledge is not required. Unlike a rule-based system, where the knowledge engineer must anticipate rule interactions which CBR allows a simple additive model for knowledge acquisition. This requires an appropriate representation for cases in such a useful retrieval index, and a case adaptation strategy.
4. Appropriate indexing strategies add insight and problem-solving power to the ability to distinguish differences in target problems and select an appropriate case, important source of a case-based reasoners power; often, indexing algorithms can provide this functionality automatically [19].
5. CBR doesn't require extensive analysis of domain knowledge. CBR permits problem solving even if domain knowledge is incomplete. The most important thing is to know how to compare two cases [49].
6. CBR allows a reasoner to propose solutions in domains that are not completely understood. This is a particular importance to the advanced planning that is necessary to design and build complex facilities such as hospitals.
7. The knowledge acquisition for a CBR system is natural. Concrete examples rather than piecemeal rules can be used. Experts (experienced practitioners) find it difficult to report the knowledge they use to solve problems. They are quite at home reporting their experiences and discussing the ways in which cases are different from each other [13, 27].

8. CBR should be considered when it is difficult to formulate domain rules but cases are available. Formulating rules is difficult in weak-theory domains such as architectural briefing and design. In this domain knowledge is incomplete, uncertain or inconsistent. It is impossible to formulate rules when there is a great amount of variability in design situations that have the same outcome.

9. CBR can be considered when rules that can be formulated require more input information that is normally available. This may be due to incomplete specified problems or the fact that the knowledge required is not available at problem-solving time. This is often the case in the construction industry and fast track projects where all project information is not available up-front.

10. CBR should be used when generally applicable knowledge is not sufficient to solve the problem. This could be due to the fact that knowledge changes with context or because some of the knowledge required solving the problem is used only under special circumstances [11].

11. When there is no fast computational method for evaluating a solution or when there are so many unknowns that evaluation methods are unusable or difficult to use, CBR provides an alternative [13].

3.1.6 The disadvantages of Case-Based Reasoning

CBR has several disadvantages and caveats in architectural design that should also be considered. The list below has been collated and adapted from [13,19, 42]:

1. CBR requires cases. Traditionally the effort in building a CBR system went into case collection. It is apparent from a study and interview with the designers of ARCHIE that it was an enormous effort. To be successful in the architectural

profession and the construction industry it should not require such extraordinary efforts. The case library should be automatically assembled during the normal professional design activities.

2. For CBR to be useful and reliable, cases with similar problem statements should have similar solutions. CBR is based on the premise that situations recur in a predictable way.
3. Adaptation modifies old solutions to fit new situations. If a domain is discontinuous where similar situations require wildly different kinds of solutions, then CBR cannot be used and would be misleading. This is unfortunately only partially true in architecture. Creative designers do not always solve related design problems in a similar way.
4. An inexperienced case-based reasoner might be tempted to use old cases blindly, relying on previous experience without validating it in the new situation.
5. A case-based reasoner might allow cases to bias him or her too much in solving a new problem.
6. Case libraries require considerable storage space. In the design of CBR systems special consideration must be given to ensure a long life of the case with changing technology. A large sum of money in terms of intellectual capital time and effort is encapsulated in the case library. Persistence of data is therefore paramount importance.
7. Inexperienced people are often not reminded of the most appropriate sets of cases when they are reasoning.
8. Cases do not often include deeper knowledge of the domain .This handicaps explanation facilities, and in many situations it allows the possibility that cases may be misapplied, leading to wrong or poor quality advice.

9. A large case base can suffer problems from store/compute trade-offs.

10. It is difficult to determine good criteria for indexing and matching cases currently, to retrieval vocabularies and similarity for matching algorithms that must be carefully hand crafted; this can offset many of the advantages CBR offers for knowledge acquisition.

3.2 Model Based Reasoning (MBR)

3.2.1 Introduction

Model-Based Reasoning (MBR) concentrates on reasoning about a system's behaviour from an explicit model of a mechanism by modelling that behaviour. Because they employ models that are compact axiomatic systems from which large amounts of information can be deduced. MBR techniques can very succinctly represent knowledge more completely and at a greater level of detail than techniques that encode experience.

Model-based reasoning does not restrict to AI fields as it is, in fact, a technique widely used by engineer and scientist of all fields, as well as in economist and politics. For example, it can be based on simulation techniques, but it can as well not even involve computers. Analytical modelling for example is quite broadly used for simple systems. In the aeronautical field, material objects reduced copies of the original which are often used to reason and perform experiments that help analyse the original system [10]. Model-based reasoning takes knowledge about the entities structures and interactions in a particular domain and uses that knowledge as a foundation for generating a description of the behaviour of some systems. The key feature is a model that it maintained which mirrors an important feature of the domain.

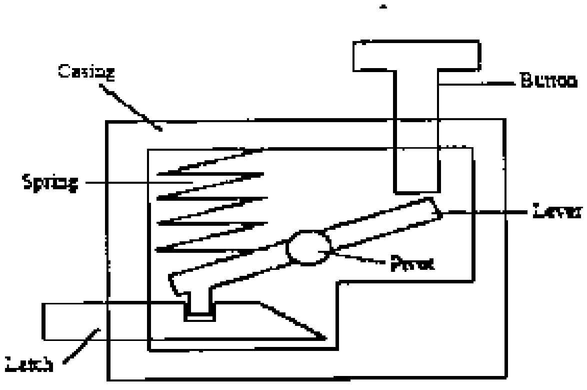


Figure 3.2 Simple model for latch lock (adaptive from [10])

For example, consider the very simple latch lock illustrated in the previous figure 3.2. If the structure of this device and the behaviour of its constituent parts are represented in an appropriate form, then it should be possible to synthesise the behaviour of the whole system.

The lock consists of six components, connected together in a particular manner. From a structure/behaviour model of this device it should be possible to determine that if the button is pressed then the latch will become free to move.

MBR is an inferring process using models abstracted from the reality of a physical system. MBR is the symbolic processing of an explicit representation of the internal working on the system in order to predict, simulate and explain the resultant behavior of the system from the structure, causality, functional and behavior of its components. Qualitative models aim to capture the fundamental aspects of a system or mechanism, while suppressing much of the detail. Methods such as abstraction and approximation are often used to build models based on symbolic rather than numeric quantity spaces. Such models are based on a sound domain theory that provides a systematic, consistent and complete knowledge base for the aspects of the system being modeled [10].

The main differences in emphasis between these models and the conventional models used in science and engineering are the incorporation of explanation structures and the requirement for the system structure to be explicitly modelled. Such models often turn out to exhibit much better structural similarities with the conceptual basis of the real system.

Model-based reasoning developed in response to pressing problems in automated reasoning. Earlier methods encoded knowledge about how mechanisms failed, but did not encode knowledge of how correctly functioning mechanisms worked [16].

3.2.2 Reasoning techniques

The basic paradigm of Model-Based Reasoning lies in the interaction of observation and prediction. It consists in comparing the model results with some observations of the corresponding system. It involves making a series of experiments with the model until you finally identify what is your system made-up with, or what is going wrong in your system, or what is the best design of your system...etc. Figure 3.3 describes how this process can be carried on [10].

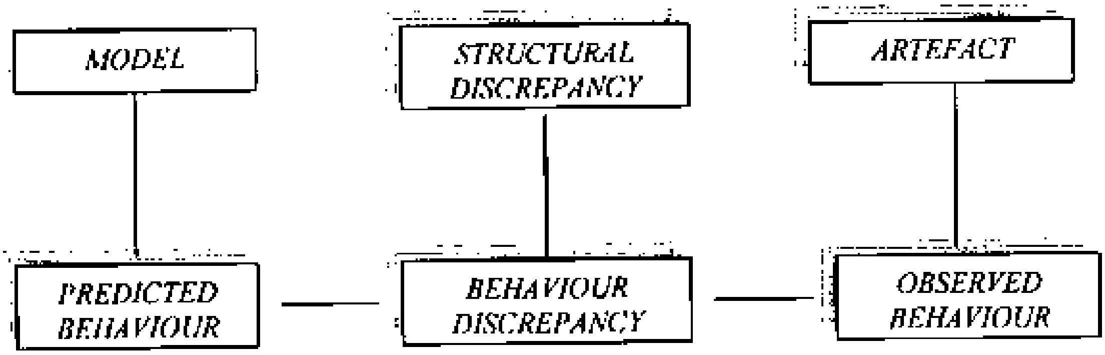


Figure 3.3[1] Model-Based Reasoning paradigm.

To predict behaviour, different reasoning techniques can be used depending on the values that the variables of the model can take.

3.2.3 Types of Model-Based Reasoning

The way in which a model represents some system will vary, depending on the type of system. Used to which the model is to be put and the information available below show several modelling methods are considered. For more detailed discussions on this subject and comparisons of the structural, process and constraint model approaches are mentioned in [25].

[1] This figure is borrowed from [10].

3.2.3.1 Causal models

Causal models have been used in the last few years in the design of some diagnostic expert systems. In this approach, knowledge is represented by means of causal relationships between observations (finding) and diagnostic hypotheses. Causal model consists of the representation of the possible states of a system (mechanical, physical or physiological) and of the relationships, between these states .Kinds of relationships may differently depend on a particular domain of application.

Different forms of reasoning can be performed on causal models .A simple possibility is when analyzing the consequences of an external perturbation on a system under specific conditions that could search for a possible cause of some particular state [36].

In a causal model, the behaviour of the whole device is explicitly embedded in the representation, which essentially describes links between causes and effects.

The model builder must therefore correctly specify all possible causes and their effects. Very little structure is imposed on these models and this can lead to problems with maintainability and reusability. In order to analyse a causal model, it is merely necessary to find the appropriate cause and follow the associated links to find what effects it might have.

This is often the simplest (and easiest) way to build a model of a system in a new domain. However, having built such a model it can be difficult to reuse much of that model in the construction of the next one. It can also be difficult to extend the model at a later date. Finally, it is necessary to be able to model separate interacting systems; causal models provide no way to represent such modularity [26].

3.2.3.2 Physical structural models

Such models attempt to capture the important physical structural features of a system and what components are present within the system also and how they are connected together. The behaviour of the individual component is described independently of any particular system (this means that the components can often be reused in different systems within the same domain). The way in which component behaviour is described varies depending upon the domain and the specific technique being used. For example, in [26] he described the behaviour of mechanical components using force propagation and resolution rules which differed depending upon the type of component being defined (e.g. the behaviour of a solid slab and the behaviour of a spring were defined by different sets of rules). Alternatively, specify component behaviour using qualitative versions of differential equations taken from systems dynamics (called confluences). These equations describe relations between parameters that represent the characteristics of the materials affected by the component. A component can have more than one set of confluences, which represents the fact that physical systems can behave in significantly different ways under different circumstances.

Using the component behaviour descriptions, the operation of the whole system is then generated by analysing how the components interact within the specified structure. This type of model is often applicable when devices can be described in terms of their components and the relations that exist between components. Such descriptions are often available for man-made systems, and in particular for engineered systems. Thus, mechanical and electrical systems, which are strongly component oriented, are good candidate problem domains. The strong mapping between these models and the real device makes it possible to alter the structure and/or component behaviour to match most alterations to the real device. The resulting new device simulation will then automatically reflect those changes.

This type of model is not suitable for domains where behaviour of the whole cannot easily be synthesised from the behaviour of the component parts. This occurs where processes are embedded in the domain, for example, in biological and medical domains. Another restriction is that it is only applicable for devices with fairly small numbers of discrete components reasoning about the behaviour of gases, liquids or polystyrene beads in this way is often not sensible. This means, for instance, that reasoning about chemical or continuous process engineering using purely structural models is unlikely to be fruitful [26,37].

3.2.3.3 Logical models

These models are very similar to the physical structural models, except that instead of representing the actual structure of the system, the underlying logic of the system is modelled. For example, in digital electronics a half adder circuit might be represented in terms of the logic of the operations, rather than, as an implementation in terms of actual electrical components [25]. However the physical world can not be completely ignored, for example, when bridging faults occur between particular spatial locations. [26] Attempts can be made to overcome these problems by using information about the physical structure in parallel with logical models.

The advantages and drawbacks of logical models are similar to those for physical models. They are however more remote from the actual system, and depend on careful mapping of logical assertions to significant system features. This is much more appropriate for electronic devices than it is for mechanical devices. In mechanical devices, the physical structure is usually of paramount importance.

3.2.3.4 Constraint models

The constraint model approach considered in [9, 26] does not directly represent information about the structure of a system or the behaviour of its components. Instead, variables are identified whose values indicate the state of the system, and relationships between those variables often synthesised from the systems structure are determined. These variables and relationships can then be represented as qualitative constraints, which are, in effect, qualitative versions of differential equations that are used in traditional physics. Analysis is then performed by considering the effects of a given initial state on these equations. This can lead to further states being generated, which are again analysed. This cycle continues until no new states are generated and all transitions between states have been determined.

This approach assumes that everything that occurs can be appropriately described in terms of constraints. As these models are based solely upon the concept of qualitative versions of differential equations, it can be difficult to produce a model for a system which cannot easily be so described using differential equations. This is often the case in physiological and medical systems, for example in the heart, and biological systems. This means that a constraint model approach is inappropriate method for representing relationships between system variables in such cases. The approach taken by [26] is also based on qualitative differential equations and so would appear to suffer from the same limitations as the constraint approach.

3.2.3.5 Process models

Many natural phenomena cannot easily be characterised using the structural models described above. Examples include physical effects and processes such as boiling, freezing and evaporation. In contrast to device centred methods, languages have been developed for describing physical processes [26]. These languages allow processes to be defined in a modular manner, each module specifies exactly the conditions under which it is applicable. The concept of a process becomes the overriding cause of all changes in the system, with the behaviour of the components becoming irrelevant. It is therefore processes which create, remove and manipulate the substances present in a system.

For mechanical and electrical devices, mappings from a real device to a process model often prove unsuitable or difficult. Process is more a system level concept, rather than a component-level one. It is difficult to predict from a process model, when a change of a component may make a difference to behaviour of the whole device [16].

3.2.3.6 Functional models

Knowledge of the function of a device, or what the device is intended to do, is important for very many tasks (for example, diagnosis, failure mode effects analysis and intelligent tutoring systems etc.). Yet many model-based systems fail to capture such knowledge or only capture it implicitly. For example, they rely solely on the knowledge of the structure of the system and the way in which the components operate. This type of model only represents knowledge on *how the system works*, not *what it is for*. However, engineers also rely on knowledge of the *function* of a device in order to successfully complete many of their tasks. This has led to a number of researchers to develop models which allow the known functionality of a device to be represented [37].

Functional models have been extensively investigated in the context of several problem-solving tasks such as device diagnosis. The model representing the functioning of a problem solver explicitly specifies how the knowledge and reasoning of the problem solver result in the achievement of its goals. One of the major advantages of exploiting functional models comes from their ability to focus on a particular subsystem of a system. This subsystem can then be analysed in detail. This can reduce the complexity of the model to be analysed and potentially reduce any ambiguity present in the resulting analysis [26].

3.2.4 Advantages of Model Based Reasoning

1. The ability to use functional/structural knowledge of the domain in problem solving. This increases the reasoners ability to handle a variety of problems, including those that may not have been anticipated by the systems designers.
2. Model-based reasoners tend to be very robust .For the same reasons that humans often retreat to first principles when confronted with a novel problem, model-based reasoners tend to be robust and flexible problem solvers.
3. Some knowledge is transferable between tasks. Model-based reasoners are often built using scientific, theoretical knowledge. Because science strives for generally applicable theories; this generality often extends to model-based reasoners.
4. Model-based reasoners often can provide causal explanations. These can convey a deeper understanding of the fault to human users, and can also play an important tutorial role [19].
5. Dynamic generation of problems. limited problem set has been recently recognized as a potential drawback of encoding a finite number of problems into a tutor [6]. Using Model-Based Reasoning for domain modeling can easily address this drawback. Since domain models based on MBR are capable of solving problems on their own without being told the correct solution, a tutor using such models need not to be restricted to administering only the problems

that have been encoded into it. When coupled with a scheme for generating problems, such a tutor can potentially administer an unlimited number of problems to the learner [6].

6. Consistency: Using one model for two tasks assures that both tasks are reasoning about the same object. Changes made to the model in the course of doing one task are accessible to the other; don't need to worry about the change failed to propagate.
7. Task integration: If the diagnostic system is available to the design system, a designer may inquire about what failures are possible. He may propose undesirable behaviors of the mechanism and then allow the diagnostic unit to discover faults which could cause such behavior [16].
8. The models are closer to the domain. One of the advantages of qualitative models is that they do not require special expertise in fields such as statistics and numerical analysis. In mathematical models we have both statistics and numerical analysis model for such skills are needed both to construct the model and also to interpret the results. Indeed, a great deal of practice and experience are needed to use mathematical models as effective tools. However, with qualitative methods, the terms used by the models and the results generated are more directly linked to the domain.
9. Explanations can be generated at the user's level. With model-based representations and qualitative reasoning, the results produced, are more likely to be comprehensible by the domain user (and/or expert) [26].
10. Even with inexact models, results can be produced. Similarly, in situations where the information available about some process is incomplete, qualitative models and functional models can still be exploited. For example, the fact that

plants produce phytoalexins in response to pathogen attack is well known, but the exact details of the process are unknown. This process cannot be modelled numerically; however it could be modelled causally or by a qualitative structure and behaviour model.

11. Easy of modeling: Even when detailed numerical models of a particular process are available and exact data can be obtained, qualitative models can still be preferable. This is because the symbolic representation of a qualitative model closely matches the structure of the device being modeled to make it easier for qualitative systems to produce explanations and justifications at an appropriate level for human comprehension than is the case with numerical models of a domain.

12. Speed of analysis. A qualitative simulation can often be performed much more rapidly than a comparable numerical one, and usually needs much less computing power. This is partly caused of the comparative simplicity of the model, and partly because modifications are rapid and do not require extensive reformulation and set-up time [25].

3.2.5 Disadvantages of Model Based Reasoning

1. A lack of experiential (descriptive) knowledge of the domain. The heuristic methods used by rule based approaches reflects a valuable class of expertise.

2. It requires an explicit domain model. Many domains, such as the diagnosis of failures in electronic circuits that have a strong scientific basis which supports model-based approaches. However, many domains, such as some medical specialtiesin in most design problems, or in many financial

applications a lack of well-defined scientific theory. Model-based approaches cannot be used in such cases.

3. High complexity. Model-based reasoning generally operates at a level of detail that leads to significant complexity; this is, after all, one of the main reasons that human experts develop heuristics in the first place.
4. Exceptional situations. Unusual circumstances, for example, bridging faults or the interaction of multiple failures in electronic components, can alter the functionality of a system in ways difficult to predict a priori [19].
5. Generates all possibilities. This means that a qualitative simulation traverses the complete space of possible system states. This is computationally very expensive and is the main cause of the inefficiency inherent in qualitative systems.
6. Creates unreal ambiguity. As a qualitative diagnostic system generates all states which are qualitatively possible, it also generates some states which could not, actually occur in the physical device. This is a sub state of qualitative representation which is ambiguous in some situations (where the actual system is not), and partly due to the local viewpoint of each decision made by the simulator.
7. Cannot handle quantitative problems. Qualitative representations are abstractions of information which is essentially quantitative in nature. It is not reasonable to attempt problems that are beyond the precision of the qualitative representation. For example, a qualitative diagnostic system that is attempting to diagnose the failure of a hot, paint stripper would most likely be unable to distinguish between the values of ambient

temperature and a temperature which is just below that required to burn the paint.

8. Demands considerable modelling effort. When a qualitative model is initially built in a new domain, it may require greater effort than that needed to build a heuristic knowledge base in the same domain. This is because of the building process is information intensive and requires all the knowledge that underlies the domain to be made explicit. However once a qualitative model has been built it is easier to maintain and reuse than a heuristic knowledge base.
9. In some domains there are no recognised models of the real system, in which case it may be easier to build a set of heuristics to represent system behaviour rather than to attempt to model it [25].

3.3 Fuzzy Reasoning

3.3.1 Introduction

In this section we introduce the basic notations and definitions needed for fuzzy reasoning. That attempts to answer the question (How can we represent expert knowledge that uses vague and ambiguous terms in the computer?).

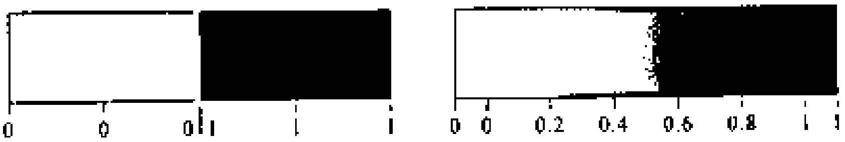
In 1965 Lotfi Zadeh, published his famous paper "Fuzzy sets". Zadeh extended the work on possibility theory into a formal system of mathematical logic, and introduced a new concept for applying natural language terms. This new logic for representing and manipulating fuzzy terms has been called fuzzy logic [30].

In 1979 Zadeh introduced the theory of fuzzy reasoning. This theory provides a powerful framework for reasoning in the face of imprecise and uncertain information. The use of fuzzy logic and fuzzy reasoning methods are becoming more and more popular in intelligent information systems [45].

Knowledge representation and uncertainty reasoning are the essential techniques in an expert system or a decision support system. The most commonly used method of uncertainty reasoning is based on fuzzy set theory. By using fuzzy rules in the form of " IF A THEN B", where "A" and "B" are linguistic terms, fuzzy reasoning is accomplished by fuzzy operations, such as "Min" and "Max" operations on membership functions [29].

The essential goal of approximate or fuzzy reasoning is how suitably treat pieces of fuzzy information given via fuzzy data, via fuzzy values of fuzzy variables and via fuzzy relationships between their values [20].

Fuzziness rests on fuzzy set theory, and fuzzy logic is just a small part of that theory. Fuzzy logic is not logic that is fuzzy, but logic that is used to describe fuzziness. Fuzzy logic is the theory of fuzzy sets, sets that calibrate vagueness. Fuzzy logic is based on the idea that all things admit of degrees. Temperature, height, speed, distance, beauty all come on a sliding scale. Tom is a *very tall*. Fuzzy logic is a set of mathematical principles for knowledge representation based on degrees of membership. Unlike two valued Boolean logic, fuzzy logic is multi-valued. It deals with degrees of membership and degrees of truth. Fuzzy logic uses the continuum of logical values between 0 (completely false) and 1 (completely true). Instead of just black and white, it employs the spectrum of colours, accepting that things can be partly true and partly false at the same time [30, 38].



(*a*) Boolean Logic. (*b*) Multi-valued Logic.

Figure 3.4[1] Boolean logic and fuzzy logic.

3.3.2 Fuzzy Sets and Logic

A fuzzy set is a set with fuzzy boundaries.

Let *X* be the universe of discourse and its elements be denoted as *x*. In the classical set theory, crisp set *A* of *X* is defined as function *fA*(*x*) called the characteristic function of *A*

$fA(x)$: $X \to \{0, 1\}$, where $$f_A(x) = \begin{cases} 1, \text{if } x \in A \\ 0.\text{if } x \notin A \end{cases}$$

This set maps universe *X* to a set of two elements. For any element *x* of universe *X*, characteristic function *fA*(*x*) is equal to 1 if *x* is an element of set *A*, and is equal to 0 if *x* is not an element of *A*.

[1] This figure is borrowed from [30].

In the fuzzy theory, fuzzy set A of universe X is defined by function $\mu A(x)$ called the *membership function* of set A

$$\mu A(x): X \rightarrow [0, 1], \text{ where } \mu A(x) = 1 \text{ if } x \text{ is totally in } A;$$
$$\mu A(x) = 0 \text{ if } x \text{ is not in } A;$$
$$0 < \mu A(x) < 1 \text{ if } x \text{ is partly in } A.$$

This set allows a continuum of possible choices. For any element x of universe X, membership function $\mu A(x)$ equals the degree to which x is an element of set A. This degree, a value between 0 and 1, represents the degree of membership, also called membership value, of element x in set A described in [30].

Fuzzy set theory generalizes classical set theory in that the membership degree of an object to a set is not restricted to the integers 0 and 1, but may take on any value in [0,1] [35].

❖ Linguistic variables and hedges

- At the root of fuzzy set theory lies the idea of linguistic variables.
- A linguistic variable is a fuzzy variable. For example, the statement "John is tall" implies that the linguistic variable *John* takes the linguistic value *tall*.
- The range of possible values of a linguistic variable represents the universe of discourse of that variable. For example, a man who is 185 cm tall .He is member of the tall men set with a degree of membership of 0.5.However , he is also a member of the set of very tall men with a degree of 0.15 which is fairly reasonable see figure 3.5.
- A linguistic variable carries with it the concept of fuzzy set qualifiers, called *hedges*.

- Hedges are terms that modify the shape of fuzzy sets. They include adverbs such as *very*, *somewhat*, *quite*, *more or less* and *slightly* [30, 38].

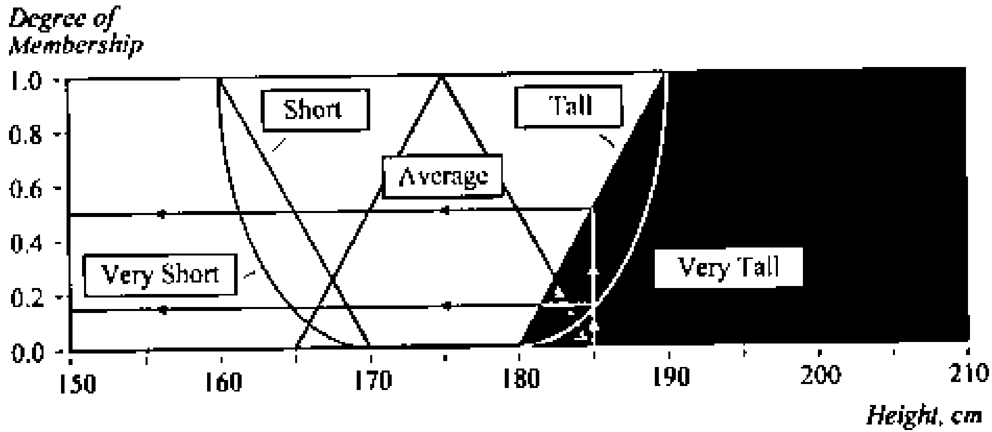


Figure 3.5[1] Fuzzy sets with the hedge *very*

3.4.3 Fuzzy rules

In 1973, Lotfi Zadeh published his second most influential paper. This paper outlined a new approach to analysis of complex systems, in which Zadeh suggested capturing human knowledge in fuzzy rules. A fuzzy rule can be defined as a conditional statement in the form of a very classical rule denotation as follow:

IF x is A
THEN y is B

Where x and y are linguistic variables; and A and B are linguistic values determined by fuzzy sets on the universe of discourses X and Y, respectively [30].

Fuzzy rules are the cornerstone of fuzzy logic systems. Rules are a form of proposition. A *proposition* is an ordinary statement involving terms that have been defined, e.g., "The damping ratio is low." Consequently, we could have the following rule: "IF the damping ratio is low, THEN the system's impulse response oscillates a long time before it dies out." In traditional propositional logic, the proposition must be

[1] This figure is presented in [38].

meaningful to call it "true" or "false," whether or not we know which of these terms properly applies. Logical reasoning is the process of combining given propositions into other propositions, and then doing this over and over again [35].

❖ How to reason with fuzzy rules?

Fuzzy reasoning includes two distinct parts: evaluating the rule antecedent the IF part of the rule and implication or applying the result to the consequent and the THEN part of the rule which include the action when ever the condition of the IF part stands a true value.

In fuzzy systems where the antecedent is a fuzzy statement, all rules fire to some extent, or in other words they fire partially. If the antecedent is true to some degree of membership, then the consequent is also true to that some degree.

Fuzzy reasoning can be defined also as a process of mapping from a given input to an output, using the theory of sets [30].

3.4.4 Fuzzy Logic

Fuzzy logic begins by borrowing notions from crisp logic, just as fuzzy set theory; however, doing this is inadequate for engineering applications of fuzzy logic. In engineering, cause and effect is the cornerstone of modeling, whereas in traditional propositional logic it is not. Ultimately, this will cause to define "engineering" implication operators. Before doing this, let us develop an understanding of why the traditional approach fails us in engineering. As in our extension of crisp set theory to fuzzy set theory, replacing the bivalent membership functions of crisp logic with fuzzy membership functions makes our extension of crisp logic to fuzzy logic [35].

Fuzzy reasoning in fuzzy logic is basically generalized from traditional logic with the exception of its computational process [48].

3.4.4.1 Advantages of Fuzzy Logic

1. Fuzzy logic converts complex problems into simpler problems using approximate reasoning. The system is described by fuzzy rules and membership functions using human type language and linguistic variables. Thus, one can effectively use his/her knowledge to describe the system's behavior.
2. A fuzzy logic description can effectively model the uncertainty and nonlinearity of a system. It is extremely difficult, if not impossible, to develop a mathematical model of a complex system to reflect nonlinearity, uncertainty, and variation over time. Fuzzy logic avoids the complex mathematical modeling.
3. Fuzzy logic is easy to implement using both software on existing microprocessors or dedicated hardware [29].
4. Fuzzy logic based solutions are cost effective for a wide range of applications (such as home appliances) when compared to traditional methods [30].

3.4.4.2 Disadvantages of Fuzzy Logic

Fuzzy logic has been proven successful in solving problems in which conventional mathematical model based approaches are either difficult to develop or inefficient and costly. Although easy to design, fuzzy logic brings with it some critical problems.

1. The use of fixed geometric-shaped membership functions in fuzzy logic limits system knowledge more in the rule base than in the membership function base. This results in requiring more system memory and processing time.
2. Fuzzy logic uses heuristic algorithms for defuzzification, rule evaluation, and antecedent processing. Heuristic algorithms can cause problems mainly because

heuristics do not guarantee satisfactory solutions that operate under all possible conditions. Moreover, the generalization capability of fuzzy logic is poor compared to neural nets. The generalization capability is important in order to handle unforeseen circumstances.

3. Conventional fuzzy logic cannot generate rules (users cannot write rules) that will meet a pre-specified accuracy. Accuracy is improved only by trial and error.
4. Conventional fuzzy logic does not incorporate previous state information (very important for pattern recognition, like speech) in the rule base [29].

3.4.5 Fuzzy Reasoning techniques

There are several fuzzy inference techniques used based on fuzzy logic. We are willing to describe some of these techniques as follow:

3.4.5.1 Mamdani-Style technique

The most commonly used fuzzy inference technique is the so-called Mamdani method. In 1975, Professor Ebrahim Mamdani of London University built one of the first fuzzy systems to control a steam engine and boiler combination. He applied a set of fuzzy rules supplied by experienced human operators.

The Mamdani style fuzzy inference process is performed in four steps as follow [30]:

- Fuzzification of the input variables,
- Rule evaluation;
- Aggregation of the rule outputs, and finally
- Defuzzification.

We examine a simple two-input one-output problem that includes three rules:

Rule: 1		Rule: 1	
IF	x is $A3$	IF	*project funding* is *adequate*

OR *y* is *B*1	OR *project staffing* is *small*
THEN *z* is *C*1	THEN *risk* is *low*
Rule: 2	Rule: 2
IF *x* is *A*2	IF *project funding* is *marginal*
AND *y* is *B*2	AND *project staffing* is *large*
THEN *z* is *C*2	THEN *risk* is *normal*
Rule: 3	Rule: 3
IF *x* is *A*1	IF *project_funding* is *inadequate*
THEN *z* is *C*3	THEN *risk* is *high*

Step 1: Fuzzification

The first step is to take the crisp inputs, *x*1 and *y*1 (*project funding* and *project staffing*), and determine the degree to which these inputs belong to each of the appropriate fuzzy sets.

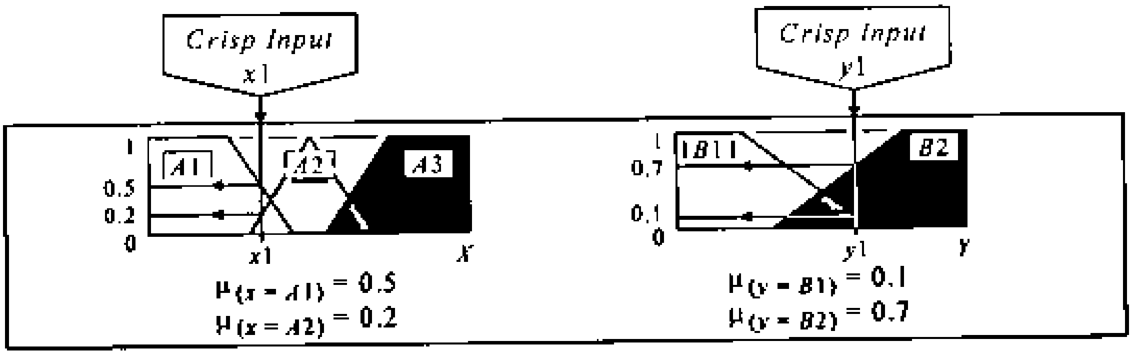


Figure 3.6[1] Fuzzification

Step 2: Rule Evaluation

The second step is to take the fuzzified inputs, $\mu(x{=}A1) = 0.5$, $\mu(x{=}A2) = 0.2$, $\mu(y{=}B1) = 0.1$ and $\mu(y{=}B2) = 0.7$, and apply them to the antecedents of the fuzzy rules. If a given fuzzy rule has multiple antecedents, the fuzzy operator (AND or OR) is used to obtain a single number that represents the result of the antecedent evaluation. This number (the truth value) is then applied to the consequent membership function.

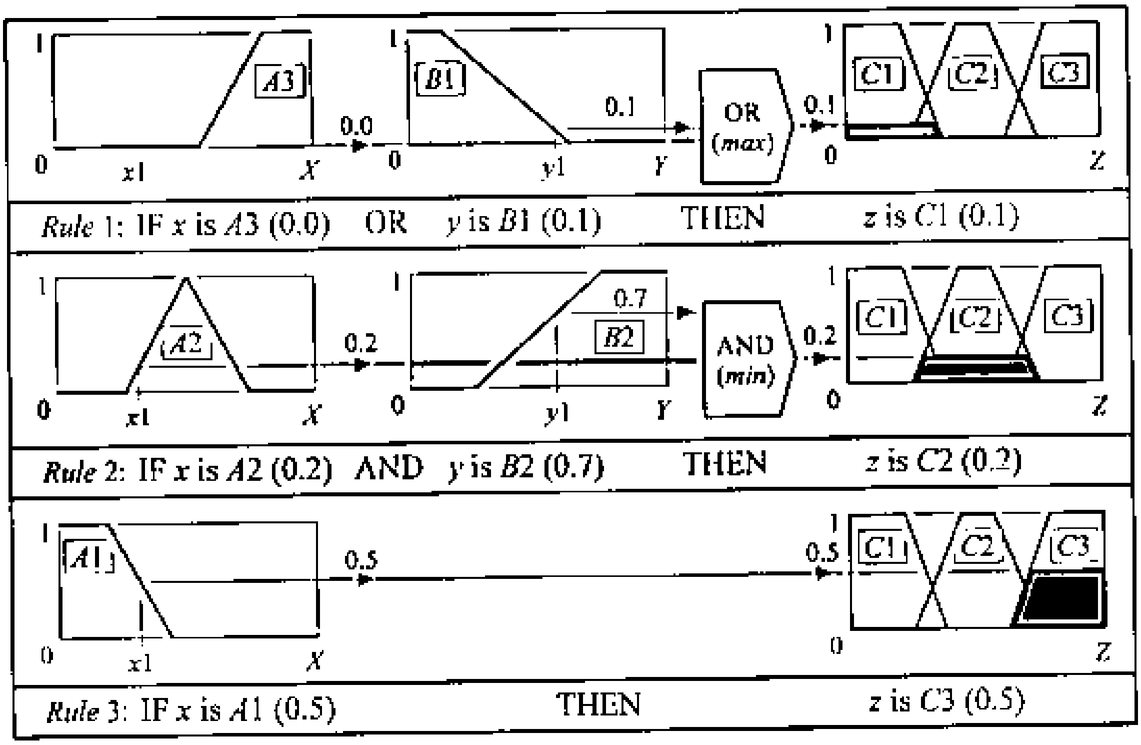
[1] All figures in this section were borrowed from [30].

To evaluate the disjunction of the rule antecedents, we will use the OR fuzzy operation. Typically, fuzzy expert systems make the use of the classical fuzzy operation union:

$$\mu A \cup B(x) = max\ [\mu A(x), \mu B(x)]$$

Similarly, in order to evaluate the conjunction of the rule antecedents, we will apply the AND fuzzy operation intersection:

$$\mu A \cap B(x) = min\ [\mu A(x), \mu B(x)]$$



Rule 1: IF *x* is *A*3 (0.0) OR *y* is *B*1 (0.1) THEN *z* is *C*1 (0.1)

Rule 2: IF *x* is *A*2 (0.2) AND *y* is *B*2 (0.7) THEN *z* is *C*2 (0.2)

Rule 3: IF *x* is *A*1 (0.5) THEN *z* is *C*3 (0.5)

Figure 3.7 Rule Evaluation

Now the result of the antecedent evaluation can be applied to the membership function of the consequent.

The most common method of correlating the rule consequent with the truth value of the rule antecedent is to cut the consequent membership function at the level of the antecedent truth. This method is called clipping, since the top of the membership function is sliced, then the clipped fuzzy set loses some information. However, clipping is still often preferred because it involves less complex and faster mathematics, and generates an aggregated output surface that is easier to defuzzify.

While clipping is a frequently used method, scaling offers a better approach for preserving the original shape of the fuzzy set. The original membership function of the rule consequent is adjusted by multiplying all its membership degrees by the truth value of the rule antecedent. This method, which generally loses less information, can be very useful in fuzzy expert systems.

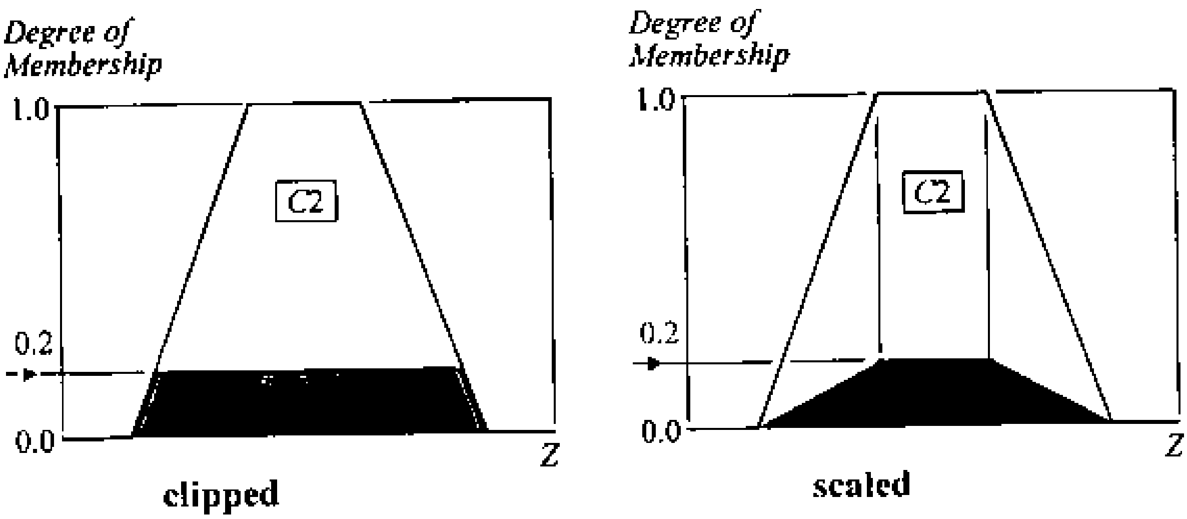


Figure 3.8 Clipped and scaled membership functions

Step 3: Aggregation of the rule outputs

Aggregation is the process of unification of the outputs of all rules. We take the membership functions of all rule consequents previously clipped or scaled and combine them into a single fuzzy set.

The input of the aggregation process is the list of clipped or scaled consequent membership functions, and the output is one fuzzy set for each output variable.

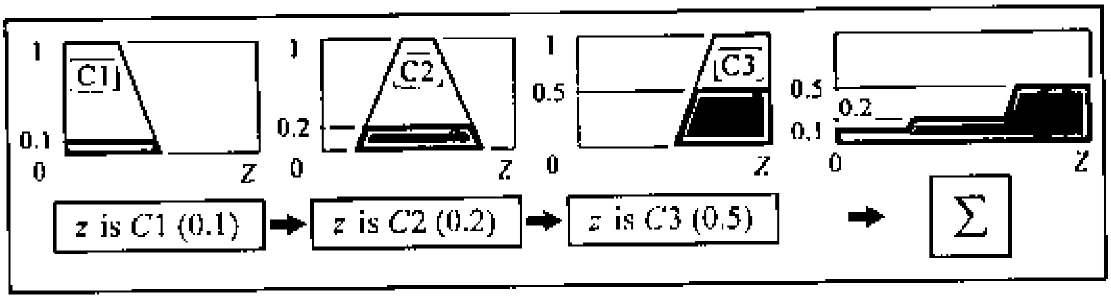


Figure 3.9 Aggregation of the rules

Step 4: Defuzzification

The last step in the fuzzy inference process is defuzzification. Fuzziness helps us to evaluate the rules, but the final output of a fuzzy system has to be a crisp number. The input for the defuzzification process is the aggregate output fuzzy set and the output is a single number.

Defuzzification: conversion of fuzzy set produced by composition stage into a crisp value.

Several defuzzification methods exist, but probably the most popular one is the centroid technique. Which finds the **Centre Of Gravity (COG)** of the aggregate set:

$$COG = \frac{\int_a^b \mu_A(x)\, x\, dx}{\int_a^b \mu_A(x)\, dx}$$

Centre Of Gravity (COG): In practice, a reasonable estimate is obtained by calculating it over a sample of points:

$$COG = \frac{(0+10+20)\times 0.1 + (30+40+50+60)\times 0.2 + (70+80+90+100)\times 0.5}{0.1+0.1+0.1+0.2+0.2+0.2+0.2+0.5+0.5+0.5+0.5} = 67.4$$

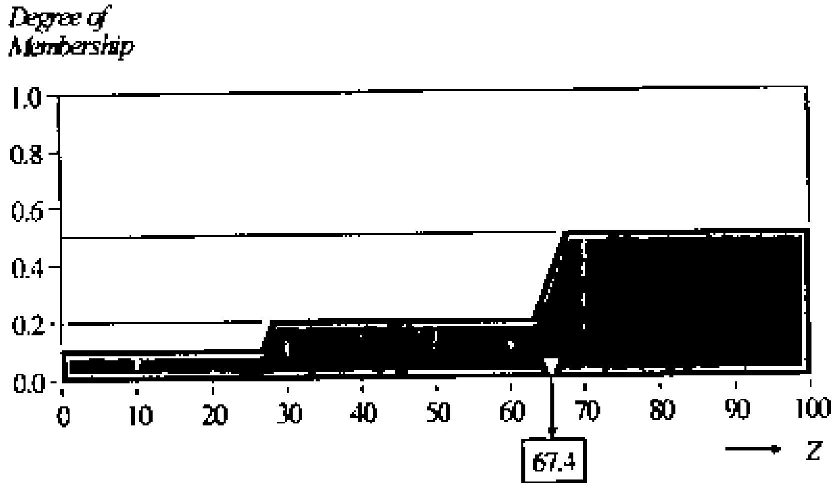


Figure 3.10 Defuzzifying the solution variable's fuzzy set

3.3.5.2 Sugeno-Style technique

Mamdani-style inference, as we have just seen, requires us to find the centroid of a two-dimensional shape by integrating across a continuously varying function. In general, this process is not computationally efficient.

Sugeno-style fuzzy inference is very similar to the Mamdani method. Sugeno changed only a rule consequent. It's using a single spike, a singleton, as the membership function of the rule consequent. A singleton, or more precisely a fuzzy singleton, is a fuzzy set with a membership function that is unity at a single particular point on the universe of discourse and zero everywhere else. Instead of a fuzzy set, he used a mathematical function of the input variable. The format of the Sugeno-style fuzzy rule was denoted as follow:

IF x is A
AND y is B
THEN z is f (x, y)

where x, y and z are linguistic variables; A and B are fuzzy sets on universe of discourses X and Y, respectively; and f (x, y) is a mathematical function.

The most commonly used zero-order Sugeno fuzzy model applies fuzzy rules in the following form:

IF x is A
AND y is B
THEN z is k

where k is a constant.

In this case, the output of each fuzzy rule is constant and all consequent membership functions are represented by singleton spikes.

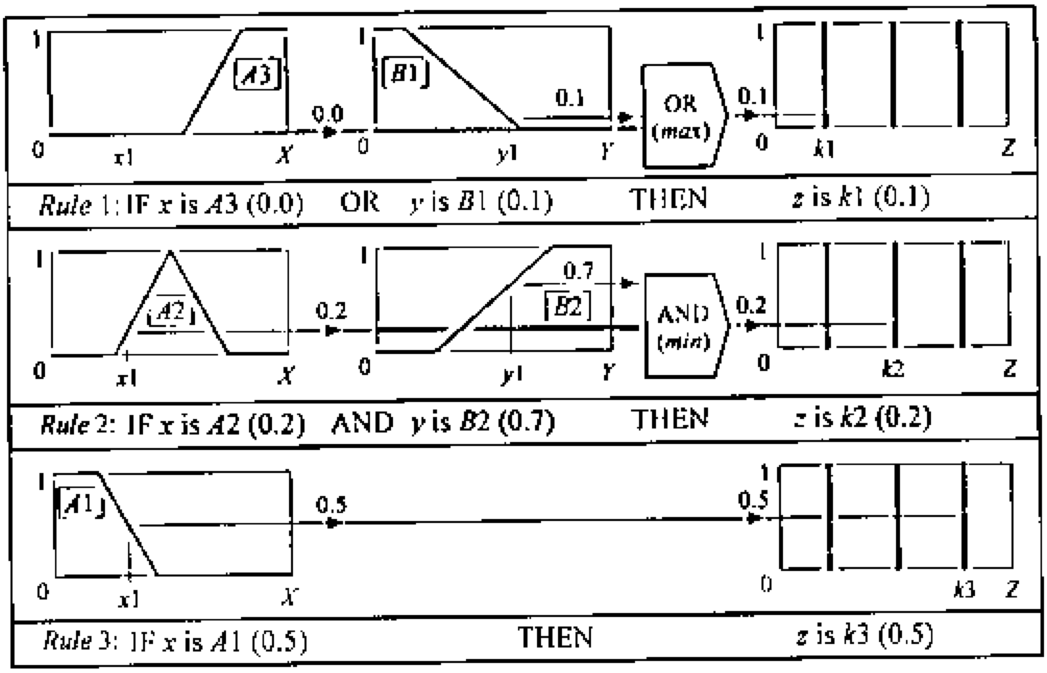

A3
0.0
B1
0.1
OR (max)
0.1
k1
Rule 1: IF x is A3 (0.0) OR y is B1 (0.1) THEN z is k1 (0.1)
A2
0.2
0.7
B2
AND (min)
0.2
k2
Rule 2: IF x is A2 (0.2) AND y is B2 (0.7) THEN z is k2 (0.2)
A1
0.5
0.5
k3
Rule 3: IF x is A1 (0.5) THEN z is k3 (0.5)
x1
X
y1
Y
Z
0
1

Figure 3.11 Sugeno-style rule evaluation

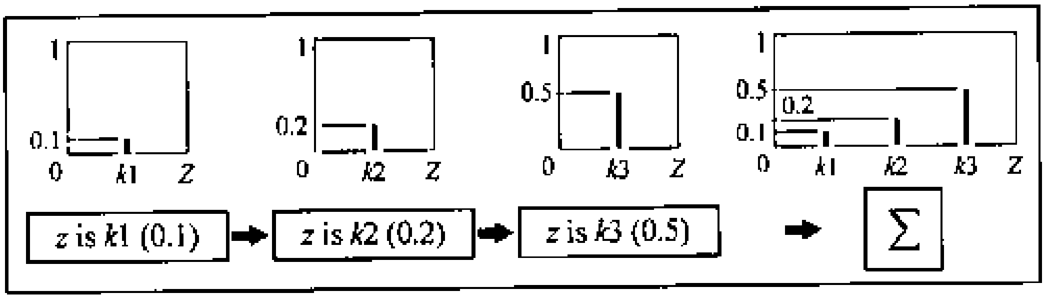


Figure 3.12 Sugeno-style aggregations of the rule outputs

Chapter 4

Comparison of Reasoning Methods

In order to evaluate the applicability of different reasoning methods, a set of criteria for evaluating the methods are used to analyze the types of certain or uncertain information that need to be represented, to determine how experts combine the different types of information when solving a problem.

4.1 Evaluation Criteria

A number of different evaluation criteria have been used for determining which method of reasoning is most appropriate. Some groups attempt to apply these criteria without regard to the application, but in general, the requirements of the application determine the applicability of a technique. The criteria that we are using to evaluate the methods for reasoning are mentioned below [10, 23, 43]. Note that our goals in this study were not only to determine how well each method for reasoning meets each of these criteria, but to determine which of the criteria are most important for this particular task.

1. Knowledge unit. In this unit we are concerned with how we may represent the knowledge in each reasoning methods and how easily we can map from the information that was provided by experts to the representation?, Does the method provide a means of representing the type(s) of uncertainty found in the domain? And what is power of the representation for different types of evidence?

2. Knowledge acquisition. It expresses how many "numbers" must one acquire from the expert and/or data?, How easy is it for the expert to provide these numbers?

And how sensitive is the system to small changes in the numbers provided by the experts?

3. Explanation mechanism. Is the theoretical basis for the representation and reasoning methods? Which can answer question such as: What is the mathematical foundation for the representation and reasoning system?. What are the assumptions made by the method (e.g. independence of evidence, mutually exclusive hypotheses, etc.)?, Are these assumptions valid or important in this domain?, Explain Why the Answer is a Good Answer (Justification), How the System Reached the Answer (Transparency), and Explain Why a Question Asked is Relevant (Relevance).

4. Knowledge transfer across problems. It may explain several questions related to: How can we get a packet of knowledge from one part of the organization to another (or all other) parts of the organization? It is considered to be more than just a communication problem. If it were merely that, then a memorandum, an e-mail or a meeting would accomplish the knowledge transfer. When the Knowledge was transferred it will be in a more complex form.

5. Domain requirement. Is the applicability of the reasoning methods to this domain? Does the method provide operators for combining information from multiple sources that yield results consistent with those provided by experts? Are expert system "shells" readily available that support this type of reasoning? Alternatively, is the approach relatively easy to implement in a standard programming language?

4.1.1 Evaluation of the Rule Based Approach

1.A knowledge unit issue was represented in rule where rules are patterns. Patterns in as rules naturally represent heuristic knowledge.

2.Knowledge Acquisition, rules are applied in an iterative cycle of micro events. Rules are also retrieved that match the input exactly some rule-based systems employ pattern matching algorithms to handle more complex forms of reasoning that can be expressed by other methods [34].

3.Explanation mechanism is the backtrack of rule firings when rules are small and ideally independent but consistent pieces of domain knowledge. Individual unit, independent of the other rules consistent piece of field of interest. Good explanation facilities are the basic rule-based in the framework which support flexible problem specific explanations. It must be mentioned that ultimate quality of these explanations depends upon the structure and content of the rules .Explanation facilities differ widely between data- and goal-driven systems. Rules map into state space search. Explanation facilities also support debugging [19].

4.Knowledge transfer across problems is high, in backtracking. Linear, deterministic rules are applied in an iterative cycle of micro events.

5.Domain requirement, rules are not applicable when a domain is well and not so well understood. Although individual rules are generally easy to understand, it is often difficult to understand how a set of rules works together and to anticipate how changes in a single rule will influence system behavior [21].

4.1.2 Evaluation of the Logic Based Approach

1. Knowledge unit issues a logic based inference procedure which begins with a set of axioms and a theorem to be proved (or refuted). If a logical representation is complete then all the logical consequences that follow from the axioms are derivable (as theorems) [40].

2. Knowledge Acquisition. If the acquisitional efficiency is high then new knowledge could be easily added to the system [45]. This was caused by the modularity of the representation which simplifies very much the problem of integration of the new knowledge into the existing knowledge.

3. Explanation mechanism, logic allows us to get across all the important points about what logic and how it can be used to perform inference that eventually results in action. The main problem is that there are just too many propositions to handle [17].

4. Knowledge transfer across problems, logic-based systems typically support some kinds of logical reasoning, but not others. For example, most logic-based systems can reason using propositional logic, but do not permit the more powerful predicate logic.

5. Domain requirement, logic is too rigorous, and inflexible, to be use in all AI problem domains [45].

4.1.3 Evaluation of the Frame Based Approach

1. Knowledge unit issues; Frames naturally represent structured knowledge. In a frame-based system, we can express that this fact is the default value and that it may be overridden [30].

2. Knowledge Acquisition: Frame-based reasoning requires an extensive knowledge of the domain and has a difficult maintenance. The main advantage of using frame-based systems for expert systems over the rule-based approach is that all the information about a particular object is stored in one place.
3. Explanation mechanism, Match the problem against classified frames that describe typical explanations and have slots describing their solutions. Fine explanation frame (schema) has the best match. The explanation can be produced by inheritance from the relevant explanation schema.
4. Knowledge transfer across problems, in frame based reasoning differs from a case based reasoning only in the way of storing. The easiest way to explain the difference is to give an example. A judicial information system would contain every judicial case as a precedent case without any exceptions. Opposite medical information system would only contain diseases only as prototypes (frames) and not every occurrence of the disease [39]. This difference becomes particularly clear when we consider frames that have a very large number of slots and where a large number of relationships exist between frames (i.e., a situation in which objects have a lot of properties, and a lot of objects are related to each other). Clearly, many real-world situations have these properties [30].
5. Domain requirement. The Frame-Based Reasoning is generally used as a support for defining terms or storing concepts within more complex systems that uses other mechanisms for reasoning about assertional knowledge also excellent for representing cases in case-based reasoning systems. Frames have been used to represent knowledge in programs for natural language

understanding where they have proven to have great economy of representation. Frames have been employed in sociology to represent the qualitative reasoning by theorists such as Goff man. The complexity of reasoning with frames, involving multiple levels of abstraction and complex semantic relationships offers a sophisticated form of computational reasoning and appears to address many of the concerns. Frame-based reasoning requires an extensive knowledge of the domain and more difficult maintenance than for CBR [21].

4.1.4 Evaluation of the Case Based Approach

1.Knowledge unit issues .In knowledge unit will focus on a libraries cases to describe how we can evaluate the work in real problems these cases could be as in the following case1: Libraries are constants that describe the way things work. Case2: Collection of data and constants. Cases are retrieved which match the input partially, in the CBR system. Many cases can not be matched exactly in all details. But some patterns may be used to recognize and store generalizations about the case, but they also are not themselves considered to be case. Furthermore, partial matching will lead to case adaptation [11].

2.Knowledge Acquisition. Cases are easier to remember because experts usually prefer explaining specific examples of the problems they have encountered and their solutions to those problems. In fact, several people building expert systems that know how to reason using cases have found it easier to build Case-based expert systems than traditional ones. The knowledge acquisition for a CBR system is natural. Concrete examples rather than piecemeal rules can be used. Experts (experienced practitioners) find it difficult to report the knowledge they use to solve problems. They are quite at home reporting their experiences and discussing the ways in which cases are different from one another [13].

3.Explanation mechanism. Cases do not often include deeper knowledge of the domain. This handicaps explanation facilities in many situations which allow the possibility that cases may be misapplied, leading to wrong or poor quality advice. [27] In simple CBR, displaying the retrieved case as a form of explanation provides complete transparency into the reasoning process. When more advanced methods like feature weighting and complex similarity measures are introduced, however, it will be necessary to provide additional information in order to fulfill the transparency goal.

4.Knowledge transfer across problems is low when it is provided for efficient solution generation and evaluation is based on the best case available. It needs a means of evaluating its solutions, guiding its adaptation and knowing when two cases are similar [43].

5.Domain requirement. CBR can be used when a domain is well and not so well understood. In the latter case it assumes the role of a generalized model. CBR does not require an explicit domain model and so elicitation becomes a task of gathering case histories, to some extent .Implementation will reduce the identifying of significant features that describe a case in an easier task than creating an explicit model. Cases are large chunks of domain knowledge and it's quite likely redundant, in the part with other cases. Based on idiosyncratic knowledge, specific to episodes but mostly not normative, provides methods for constructing solutions [19, 49].

4.1.5 Evaluation of the Model Based Approach

1.Knowledge unit issues. Store causal models of devices or domains. Causal Model Base stores the process models representing the causal relationship between process variables [6].

2.Knowledge Acquisition issues. MBR knows everything from scratch and that is why the model used should be complete. All possible situations can be diagnosed or generated at which it can actually be an alternative to a case base that is by nature incomplete [10]. It does not have to acquire knowledge, however, it can make use of first guess probabilities that may be adjusted when confronted with results. MBR may learn from other systems like CBR.

3.Explanation mechanism. Explanations can be generated at the user's level, with model-based representations and reasoning. The results produced are more likely to be comprehensible by the domain user (and/or expert) [19, 26].

4.Knowledge transfer across problems .Some knowledge is transferable between tasks. Model-based reasoners are often built using scientific theoretical knowledge. Because of science strives for generally applicable theories; this generality often extends to model-based reasoners.

5.Domain requirement. Is used when a domain is well enough understood to enumerate a causal model [6, 43].

4.1.6 Evaluation of the Fuzzy Approach

1.Knowledge unit issues. The expert seemed to be very comfortable providing ranges for membership in different categories (fuzzy sets). Unlike the other approaches, however, fuzzy reasoning allows a particular value to be a member of more than one of the categories (fuzzy sets) [43.48]. Simple piecewise linear functions with a prescribed degree of overlap appeared to be good representations of these imprecise concepts.

2.Knowledge acquisition issues. In a fuzzy reasoning, one must acquire membership functions rather than single numbers [38]. Research has shown that simple functions are usually adequate representations and can be readily supplied by Domain Experts Sensitivity Analysis which will need to be conducted to determine the effect of small changes in membership functions on decisions.

3.Explanation mechanism. Fuzzy has been widely criticized for the lack of a theoretical foundation for its reasoning methods. The wide variety of inference procedures, combining rules, and defuzzification techniques that have been developed is often used as an illustration of the variety of ad hoc methods that must be adopted in order to make fuzzy work in different situations [29].

4.Knowledge transfer across problems. The fuzzy system is rule based and, like the rule based approach that has the same strength and weakness of any rule-based system. In addition, the interaction of membership functions of sets used in premises and conclusions of different rules is sometimes difficult to anticipate [30].

5.Domain requirement. the major drawback of fuzzy for reasoning appears to be the lack of a convenient method for representing the level of importance of evidence. In particular, there is no convenient method for weighting different pieces of evidence [20].

Chapter 5

Design and Implementations of Case Studies

5.1 Implementation of a Computer Diagnostic Expert System (CDES)

The problem type influences our choice of the method and tool for implementing an intelligent system. We are willing to develop a prototype system that can help in diagnostic process. Computer manuals include troubleshooting sections which consider possible problems with the system start-up for detecting computer and their peripherals such as hard disk, keyboard, monitor, sound card, disk drivers such (floppy disk, CD-ROM,), and networks tools. Peripherals such as printer, scanner....etc.

This implementation clearly belongs to a diagnosis inferring malfunctions of an object from its behaviour and recommending solutions. Domain knowledge for solving such problems can often be represented by production rules, where a rule based expert system might be the right candidate for the job.

5.1.1 Designing process of an Expert System

5.1.1.1 Objective Data and Knowledge Acquisition Structure

To develop such a computer diagnostic system, we need to acquire knowledge about troubleshooting based on the known problems that occurs in computers. We use a troubleshooting manual to provide step-by-step procedures for detecting and fixing a variety of faults. We will consider troubleshooting sections in a following order:-

1. Motherboard/CPU/RAM
2. Power Supply
3. IDE Drives
4. CD and DVD

5. Sound Card
6. Video Card
7. Modems and Networks

✦ Knowledge Base Structure

In our CDES implementation an efficient rules in rule based expert system are required in our prototype in order to accommodate all the generated candidate diagnostic faults via troubleshooting to be used in later stage. According to flowcharts in [31] we develop a general rules and objects structure which can handle the diagnosing processes. These rules will be listed in Appendix C:

✦ Objective Data

CDES uses numbers of linguistic objects described in terms such as: fault, live_screen, power_diagnostic, video_diagnostic, RAM, CPU, CMOS, freezes_boot,. ..). Each object can tale one of the allowed values (for example ,object fault can take the value of Motherboard, Power Supply, Sound Card, IDE Drives, CD_ DVD Video Card, Modems, object screen has two value YES or NO).An object and its value constitute a fact.

5.1.1.2 Programming Language Used in the implementation

We have used Visual Rule Studio shell, in the development of our system. The reasons behind the usage of such language are the following:

- It is classified as a declarative programming language that is very easy to learn and use.
- It is compatible with windows operating system and platform rule development environment for Windows co-operations.

- It can interact with the user via Graphical User Interface, which gave the language the ability to support procedure languages as well.
- It is the ultimate object-oriented programming language.
- It is the most productive way to create component reusable rule-based objects.
- It represents a generational leap in the application of expert system technology to the problem of component-oriented rules representation.
- Visual Rule Studio is based on the Production Rule Language (PRL) and Inference Engines of LEVEL5 Object. The PRL classes of our RuleSet become objects in Visual Basic as shown in appendix A figure A.1.

Visual Rule Studio Inferencing Strategies

The Visual Rule Studio inference engines control the strategies that determine how, from where, and what order a knowledge base draws its conclusions. These inference strategies model the reasoning processes an expert uses when solving the problem.

By using Visual Rule Studio we indicate three types of inferencing strategies pointed as follow:

- Backward-Chaining.
- Forward-Chaining.
- Hybrid-Chaining.

Each of these inferencing strategies acts on specific knowledge base components. Backward-chaining inferencing starts with a specific hypothesis, or set of hypotheses, called the agenda. In backward chaining, also the inference engine works backward from the agenda, pursuing a hypothesis via its search order strategies, which can be composed of the session context, the knowledge base rules . WHEN NEEDED methods, Query Objects, or the default value. Forward chaining inferencing starts with known data or conditions and determines what can be concluded from that data. Forward chaining uses also a knowledge base's demons and WHEN CHANGED methods as well. Hybrid inferencing is a mixed strategy that combines backward chaining and forward chaining.

Visual Rule Studio uses rule-based reasoning techniques on an application's object model to create, delete, manipulate variables and objects, and to determine their values. A set of rules (RuleSet Editor) for the expert system are shown in appendix A figure A.2.

➜ RuleSet Tree View

The RuleSet Tree View provides a diagrammatic representation of the relationships among the classes, attributes, rules, and demons of a RuleSet.

To display the RuleSet Tree View, click on the Tree View button in the RuleSet Designer. Figure A.3 in appendix A showns the Tree View for our expert system.

➜ Tree View Node Types

Each node in the RuleSet TreeView has an icon indicating its type, as shown below. The RuleSet TreeView has five basic node types: RuleSet Node, Class Node, Attribute Node, Rule Attribute Conclusion Node, and Rule Node as shown in appendix A figure A.4.

5.1.1.3 Create the User Interface

As in any kind of expert systems having a good user interface in critical to determined classes, attributes and rules in principal. To make a success reasoning system we need better user interface to carry out all the information, and queries reports in unsophisticated manner. The reply of the submitted queries will be represented in a visualization mode. The role of the knowledge, structures trends and outliers may be identified, statistical tests tend incorporate isolated instances into a broader model as they attempt to formulate global features and then there is no requirement for a hypothesis, but the techniques can also support the formulation of hypotheses if wanted. The user will be attempted to select answers presented by the expert system interface shown in appendix A figure A.5.

The user has the advantage to interact the appropriate answer(reply) to the submitted query (questions) recently asked via the CDES system ,the result of the retrieval queries will be issued in the following script in appendix A figure A.6.

5.1.1.4 Evaluate the expert system

In the dialogue shown below, the expert system will demand the user to select the answers to seek for a convincer solution to the problem based on his/her demand. The answers which are supplied by the user answers which were indicated by arrows. Our expert system will apply the matched rules from its knowledge base to infer the result of diagnostic, the process will be shown in appendix A figure A.7.

Requested problem (Query)

Please enter the exact fault type?

==> Motherboard

Is live screen?

==> YES

Is freezes on boot screen?

==> YES

Is freezes bare bones?

==> YES

Is freezes on swapped ram?

==> YES

Is cmos settings default?

==> YES

Is heatsink active?

==> YES

Is runs on bench?

==> NO

Is CPU swap work?

⇨ YES

Replied solution (Result)

Diagnosis:

Bad CPU, watch voltage heat sink on replacement.

The most important recommendations about building expert system using rule based reasoning are listed below:-

1. Natural knowledge representation. An expert usually explains the problem solving procedure with such expressions such as: 'in such-and-such situation, I do so-and –so'. These expressions can be represented quite naturally as IF-THEN rules.
2. Uniform structure. Rules have the uniform IF–THEN structure. Each rule is an independent piece of knowledge .The very syntax of production rules enables them to be self documented.
3. Opaque relations between rules .Although the individual rules tend to be relatively simple and self-documented their logical interactions within the large set of rules may be opaque.

4. Ineffective search strategy. Starts when the inference engine applies an exhaustive search through all the rules during each cycle.
5. Inabilities to learn. Learning cannot automatically modify its knowledge base, or adjust existing rules or add new ones.

5.2 Implementation of a Decision-Support Expert System (DSES)

A service centre keeps spare parts and repairs failed ones. A customer brings a failed item and receives a spare of the same type. Failed parts are repaired and placed on the shelf, and thus become spares.

The objective here is to advise a manager of the service centre on certain decision policies to keep the customers satisfied. We need to decide which problem is a candidate for any reasoning methods. In this problem we cannot define a set of exact rules for each possible situation; inherently imprecise properties of the problem make it a good candidate for fuzzy technology. Fuzzy systems are particularly well suited for modelling human decision making.

To build our fuzzy expert system, we will use one of most popular tools used for such a purpose, the MATLAB. It provides a systematic framework for computing with fuzzy rules and graphical user interfaces as well.

5.2.1 Designing Process of a Fuzzy Expert System

To design a fuzzy expert system we have to go through several steps in a designing process to identify all system basic requirements which will be order as follow:

1. Objective of define linguistic variables.
2. Determine fuzzy sets.
3. Elicit and construct fuzzy rules.

4. Encode the fuzzy sets, fuzzy rules and procedures to perform fuzzy inference into the expert system.
5. Evaluate and tune the system.

5.2.1.1 Objective of Define Linguistic Variables

There are four main linguistic variables: average waiting time (mean delay) *m*, repair utilisation factor of the service centre ρ, number of servers *s*, and initial number of spare parts *n*.

The customer's average waiting time, *m*, is the most important criterion of the service centre's performance. The actual mean delay in service should not exceed the limits acceptable to customers.

The repair utilization factor of the service centre, ρ, is the ratio of the customer arrival tare , λ ,to the customer departure rate , μ. Magnitudes of λ and μ indicate the rates of an item's failure and repair, respectively. Apparently, the repair rate is proportional to the number of servers, *s*. To increase the productivity of the service centre its manager will keep the repair utilization factor as high as possible.

The number of servers, *s*, and the initial number of spares, *n*, directly affect the customer's average waiting time, and thus have a major impact on the centre's performance. By increasing *s* and *n*, we achieve lower values of the mean delay, but, at the same time we increase the costs of employing new servers, building up the number of spares and expanding the inventory capacities of the service centre for additional spare.

Let us determine the initial number of spares *n*, given the customer's mean delay *m*, number of servers *s*, and repair utilization factor, ρ .

Thus, in the decision model considered here, we have three inputs m, s, and ρ, and one as an output n.

In other words, a manager of the service centre wants to determine the number of spares required to maintain the actual mean delay in customer service within an acceptable range.

Now we need to specify the ranges of our linguistic variables, by supposing we obtain the results shown in appendix B Table B.1 where the intervals for m, s and n are normalized to be within the range of [0,1] by dividing base numerical values by the corresponding maximum magnitudes.

Note that, for the customer mean delay m, we consider only three linguistic values such as: *Very Short*, *Short* and *Medium* because other values such as *Long* and *Very Long* are simply not practical mentioned. A manager of the service centre cannot afford to keep customers waiting longer than a medium time.

5.2.1.2 Determine Fuzzy Sets

Fuzzy sets can have a variety of shapes. However, a triangle or a trapezoid can often provide an adequate representation of the expert knowledge, and at the same time, significantly simplifies the process of computation as showing in appendix B figures B.1,B.2,B.3,B.4.

5.2.1.3 Elicit and construct fuzzy rules

To accomplish this task, we might ask the expert to describe how the problem can be solved using the fuzzy linguistic variables motivated in section 5.2.1.1.

There are three inputs and one output variables in our expert system. It is often to represent fuzzy rules in a matrix form. A two-by-one system is depicted as a M x N matrix of the input variables .The linguistic values of one input variable form the horizontal axis and the linguistic values of the other input variable form the vertical axis. In the intersection of a row and a column lies the linguistic value of the output variable. But for the three-by-one system the representation takes the cube shapes of an M x N x K cube .This form of representation is called a fuzzy associative memory (FAM).

Let us make use of a very basic relation between the repair utilisation factor ρ, and the number of spares n, assuming that other input variables are fixed .This relation can be expressed in the following form: if ρ increases, then n will not decrease .Thus we could write the following three rules:

1. IF (utilisation_factor is L) THEN (number_of_spares is S),
2. IF (utilisation_factor is M) THEN (number_of_spares is M),
3. IF (utilisation_factor is H) THEN (number_of_spares is L).

Now we can develop the 3 x 3 FAM that will represent the rest of the rules in a matrix form. The results of this effort are shown in appendix B figure B.5.

A detailed analysis of the service centre operation may enable us to derive 27 rules that represent complex relationships between all variables used in the expert system .Table B.2 in appendix B contains these rules and figure which shows the cube (3 x 3 x 3) FAM representation.

First we developed 12 rules calculated as (3 + (3 x 3)) rules, but then we obtained the rest of 27 rules via the multiplication of (3 x 3 x 3) rules. If we implement both schemes, then we will compare both results. Figure B.6 in appendix B shows how this process stars for.

5.2.1.4 Encode the Fuzzy Sets, fuzzy rules and procedures to perform fuzzy inference into the expert system

To build our system by using fuzzy rules and procedures to perform fuzzy inference into the expert system we will use the MATLAB ToolBox. Such a tool usually provides complete environments for building and testing fuzzy systems .For example, the MATLAB has five integrated graphical editors: the fuzzy inference system editor, the rule editor, the membership function editor, the fuzzy inference viewer, and the output surface viewer. All these features make designing fuzzy systems much easier to be visualized in an acceptable manner for user real application.

5.2.1.5 Evaluate and tune the system

The last, task is to evaluate and tune the system. We want to see whether our fuzzy system meets the requirements mentioned in this thesis work.

Several test situations depend on the mean delay, number of servers and repair utilisation factor.

The Tool can generate surface to help us analyse the system's performance. Figure B.7 in appendix B represents three-dimensional plots graph for the two-input and one-output system in rule base system that we used.

But our system has three inputs and one output which we may move beyond three dimensions. When we move beyond three dimensions, we encounter difficulties with displaying the results. The tool has a special capability that it can generate a three dimensional, output, surface by varying any two of inputs and keeping other inputs constant. Thus we can observe the performance of our three-input, one-output system on two /or three-dimensional plots.

Although the fuzzy system works well, we may attempt to improve it by applying Rule Base 2.The results are shown in appendix B figure B.8,B.9,B.10. The expert might not be satisfied with system performance. To improve it, may suggest additional sets – Rather Small and Rather Large - and to extend the rule base according to the FAM presented in figure B.11 in appendix B, the rule base 3 shown in figure B.12. The ease with which a fuzzy system can be modified and obtain results in figure B.13.

If we suppose, a service centre is required to supply its customers with spare parts within 24 hours, the service centre employs 8 servers and the repair utilisation factor is 60%. The inventory capacities of the centre are limited by 100 spares. The values for the mean delay, number of servers and repair utilisation factor are 0.7, 0.8 and 0.6, respectively.

n=round ((evalfis([0.7 0.8 0.6],a))*100)   n = 20

Suppose, now the manager of the service centre wants to reduce the customer's average waiting time to 12 hours it can be calculated in the following aspects.

n=round ((evalfis([0.35 0.8 0.6],a))*100)   n = 35

To Tuning fuzzy expert system may involve a number of actions as review model input and output variables, and if required redefine their ranges, review the fuzzy sets, and if required define additional sets, review the existing rules, and if required add new rules to the rule base.

Chapter 6

Conclusion and Recommendation

This chapter summarizes the main work in this thesis, the development and results of this thesis would contribute to the reasoning methods fields in several ways. The formulation of a set of heuristics rules selections will help to clarify the matching between specific problem and the set of best suited algorithms or techniques for solving it. These guidelines are expected to be useful and applicable to real problem solving projects.

The conclusion of thesis work can be summarized in four components to locate a common comparative task for some different types of reasoning which it can be mentioned as follow:-

1. Providing a framework for describing and exploring all features of different reasoning schemes with emphasis on benefits, drawbacks, applicability, criteria of choosing methods for a specific application.
2. Identifying categories, structures, or properties of knowledge or tasks for which different reasoning techniques are appropriate or advantageous.
3. Providing a comparison between different reasoning schemes are used as well to guide designers to select the appropriate method for a particular domain.
4. Implementing case studies that applied two types of expert systems the first one is called Computer Diagnostic Expert System (CDES) which deal with rule based reasoning. The second expert system was named as Decision-Support Expert System (DSES) using fuzzy reasoning. Both systems were developed to formulate the requirements of this thesis work and for problem solving

respectively. Also both expert systems should contain an expressive, extendible representation system for one or more method of reasoning.

6.1 Summary of the thesis work

- A *process reasoning* is a reasoning paradigm that infers information about a domain using process or multistage methods that exists a traditional reasoning paradigm which plays a vital role in every main stage of the process.
- The RBR breaks a problem down into a set of individual rules that each solves a part of the problem. We have combined rules together to solve the whole problem. However, to create these rules by hand, we have to know how to solve the problem and the task that can be extremely complex at the time consuming. Our CBR development differs basically in the use of the rules that does not need to know how to solve a problem but only know how to recognize it if a similar problem was solved in the past. If so, the CBR used instead of the RBR to easily solve the current problem. There are real differences between our CBR development systems and the RBESs in partial matching where the CBR system many cases can not be matched exactly in all details. Instead we used Patterns to recognize and store generalizations about cases, but they are not themselves considered to be cases some times. Furthermore, partial matching leads us to case adaptation.
- Frame-based reasoning required us an extensive knowledge of the domain that had more difficult maintenance than the CBR. The main advantages of using frame-based system in expert systems over the rule-based approach in all the information about a particular object were stored in one place.

- Frame based reasoning also differs from a case based reasoning only in the way of storing it. The easiest way to explain the difference is to give an example which judicial information system would contain and every judicial case has a precedent case without any exceptions.
- In Knowledge Acquisition Cases are easier to remember than rules because experts usually prefer explaining specific examples of the problems they have encountered their solutions to those problems, than it is for them to describe their problem solving technique in term of potentially large numbers of rules. In fact, several people building expert systems know how to reason those using cases that have found it easier to build the Case-based expert systems then the traditional ones.
- Cases may guide interpretation of rules; cases may be used to focus RBR; or the CBR system that may be one of the components in the RBES.
- The CBR system may provide an alternative to RBESs, and is 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. Furthermore, traditional RBESs and CBR systems share the common theoretical foundation based on mathematical logic; that is, derived from propositional logic and predicate logic, because RBESs and CBR systems basically perform deductive reasoning. More specifically, RBESs perform reasoning based on modus ponens, while CBR systems perform reasoning based on generalized modus ponens. Because generalized modus ponens is an extended form of modus ponens were CBR systems can be considered as a general form of RBES.

- MBR is often considered in our thesis as a subtype of CBR which can be viewed as fundamentally analogical. MBR systems also solve problems by retrieving stored cases (precedents) as a starting point for new problem-solving.
- The essential difference between RBR and LBR lies in which the former possesses much more knowledge than the latter, mean while the latter is richer in inference rules and can perform a multistep process of reasoning.
- Fuzzy reasoning in fuzzy logic is basically generalized from traditional logic expressions with the exception of its computational process. Fuzzy reasoning is basically one-step reasoning in order to avoid such fuzzy degeneration. Successful applications of fuzzy reasoning do not come from its multistep process of reasoning based on its quasi-transitivity, but from its linguistic computation and more precise understanding of the fuzzy characteristics of certain domain knowledge.
- Reasoning in CBR does not belong to the form that mentioned reasoning paradigms above. It can be considered as a new kind of reasoning; which is classified as a process reasoning.
- The fuzzy reasoning system is rule based system just, like the certainty factor approach but it has the same strengths and weakness of any rule-based system. In addition, the interaction of membership functions of the sets used in premises and conclusions of different rules is sometimes difficult to anticipate.
- Fuzzy reasoning is attractive because much of the knowledge in domain appears to be imprecise. However, more work must be done to determine if adequate methods for weighting different pieces of evidence can be

developed. Another alternative that must be considered is that of using more than one of these methods. The modular design of the knowledge base for different decisions appears to make this a reasonable alternative.

6.2 Recommendation

Some inserting future research problems will be presented respectively in this section. The main focus of my future work will be to improve all the related components of reasoning aspects in a comparative way. We are willing to develop prototypes using different methods for reasoning to determine which method(s) is most appropriate.

Hybrid methods deserve the attention of researchers and application developers to build expert system that require a resolution problems regarding deterministic of reasoning methods to apply in a given situation. Resolving differences between reasoning methods and designing representation allow knowledge to be shared based on reasoning methods decision.

Methods with hybrid architecture for a single or more than one paradigms were integrated to get comparative effects which well play an alternative challenge to us, when the strengths of one of the mentioned methods will be compensate for the weakness of anther . More over, my future tasks will be listed as follow:

- We will try to improve the combining reasoning methods in a single application.
- Supporting or guiding anther method will be needed in case of combining reasoning in a single application as well.
- We are also interested to improve reasoning experience by compiling one form into another via reasoning knowledge.

- Transferring successful methods from one form of reasoning to another form is a very important task to be private.
- Interoperability of applications based on different reasoning technology can be implemented in future as well.
- Switching among alternative forms of reasoning can be a good challenge in future.
- We suggest improving the comparison and evaluation of reasoning alternatives for specific problem domains.
- Demonstrating practical advantages of a multimodal approach for real problems. These problems lead us to think of it as a major future task.
- Identifying and exploiting commonalities will finally be considered as a future work.

References

[1] A. Aamodt, *A Knowledge-Intensive, Integrated Approach to Problem Solving and Sustained Learning*, PhD thesis, University of Trondheim, Norwegian Institute of Technology (NTH), 1991.

[2] A. Aamodt, E. Plaza, *Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches*, AI Communications. IOS Press, Vol. 7: 1, pp. 39-59, 1994.

[3] A. Badiru, John Cheung, *Fuzzy Engineering Expert Systems with Neural Network Applications*, *ISBN* 0-471-29331-8, NewYork, 2002.

[4] A. Dix, J. finlay, *Introduction to Artificial Intelligence*, British library, londen, 2002.

[5] A. Wojna, *Analogy-based reasoning in classifier construction*, PhD thesis, Institute of Mathematics, Warsaw University, December 2004.

[6] A.N. Kumar, *Model-Based Reasoning for Domain Modeling*, Explanation Generation and Animation in an ITS to help Students Learn C++, International workshop at ITS, San Sebastian, Spain, 2002.

[7] B. Coppin, *Artificial Intelligence Illuminated*, ISBN: 0763732303, Jones and Bartlet Publishers, UK, 2004.

[8] B. D'Ambrosio, *Extending the Mathematics in Qualitative Processes Theory, Artificial Intelligence*, Simulation & Modeling, L. E. Widman, K. A. Loparo and N.R. Nielsen, John Wiley & Sons Ltd, New York, pp. 133-158, 1989.

[9] B. Kuipers, *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*, Cambridge, MA: MIT Press, ISBN 0-262-11190-X, 1994.

[10] C. Campos, E Rollón, *CRB and MBR techniques: review for an application in the emergencies domain*, RIMSAT Project IST—28655, The University of Girona, 2000.

[11] C. Schmidt, *Foundations of Case Based Reasoning and Applications to Hypertext*, journal of Educational computing, volume 5, USA, 2004.

[12] C. Stanfill, D.Waltz, *Memory-Based Reasoning*, White paper, Volume 29, Number 12, Communications of the ACM, 1986.

[13] D. Conradie, *The use of software systems to implement Case-Based Reasoning enabled intelligent components for architectural briefing and design*, PhD Thesis, University of Pretoria, 2000.

[14] D. Leake, *Experience, Introspection, and Expertise: Learning to Refine the Case-Based Reasoning Process*, Journal of Experimental and Theoretical Artificial Intelligence, USA, 1995.

[15] D. Poole, A. Mackworth, R.Goebel, *Computational Intelligence: A Logical Approach*, New York Oxford, Oxford University Press, 1998.

[16] D.R.Throop, *Model-Based Diagnosis of Complex Continuous Mechanisms*, PhD thesis, The University of Texas at Austin, 1991.

[17] E. Jaynes. *Probability Theory: The Logic of Science*, Cambridge University Press, ISBN 0521592712, 2003.

[18] G. Brajnik, M. Lines, *Qualitative Modeling and Simulation of Socio Economic Phenomena*, 1992.

[19] G. Luger, *Artificial Intelligence, structures and strategies for complex problem solving*, fourth edition, 2002.

[20] H. Bandemer, S. Gottwald, *Fuzzy sets, Fuzzy logic, Fuzzy methods with Applications*, ISBN 0471956368,John Wiley &Sons Ltd,England,1995

[21] H. Reichgelt, *Knowledge Representation*, 1991.

[22] J. Stannard , *Building Expert Systems in Prolog*, 2000.

[23] J. Corchado , C. Fyfe, *A Comparison of Kernel Methods for Instantiating Case Based Reasoning Systems*, Computing and Information Systems, pp. 29-42, University of Paisley, 2000.

[24] J. Crawford, *Access-Limited Logic - A Language for Knowledge Representation*, the Qualitative Reasoning Group at the Artificial Intelligence Laboratory, The University of Texas at Austin, 1994.

[25] J. Hunt, *Model-Based Diagnosis*, Hartham Park Corsham, Copyright JayDee Technology Ltd.UK, 2002

[26] J.E. Hunt, M.H. Lee, CJ. Price, *Applications Of Qualitative Model-Based Reasoning*, White paper, Control Enq. Practice, Vol. 1, No. 2, pp. 253-266, Persamen Preu Ltd,UK,1993.

[27] K.D.Althoff, *Case-Based Reasoning*, Handbook on Software Engineering and Knowledge Engineering,D-67661 Kaiserslautern, Germany.

[28] K.P.Mohanan, *Types of Reasoning: Relativizing the Rational Force of Conclusions.*

[29] L. Deren , D. Kaichang , L. Deyi , *Knowledge Representation And Uncertainty Reasoning In Gis Based On Cloud Models*, ,PhD thesis, Technical University of Surveying and Mapping,2000.

[30] M. Negnevitsky, *Artificial Intelligence: A Guide to Intelligent Systems*, 2nd Edition, Pearson education Ltd, England, ISBN: 0321204662, 2005.

[31] M. Rosenthal, *Computer Repair with Diagnostic Flowcharts*, ISBN 0972380116, 2005.

[32] M. Sollenborn, *Clustering and Case-Based Reasoning for User Stereotypes*, white paper, Mälardalen University, Västerås, Sweden.

[33] N. Kasabov, *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering*, A Bradford Book The MIT Press Cambridge, Massachusetts London, England, Second printing, 1998.

[34] P. Buzing, Hybrid *Systems: Two Examples of the Combination of Rule-Based Systems and Neural Nets*, 2001.

[35] P. Dadone ,*Design Optimization of Fuzzy Logic Systems*, PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 2001.

[36] P. Torasso, L. Console, *Diagnostic problem solving: combining heuristic approximate and causal reasoning*, Horth Oxford Academic, London, 1989.

[37] R. Davis, R. W. Hamscher, *Model-Based Reasoning: troubleshooting*. In Shrobe, H., editor, *Exploring Artificial Intelligence*, pages 297–346, Morgan- Kaufmann, 1988.

[38] R. Fuller, *On fuzzy reasoning schemes*, TUCS General Publications, No. 16, ISBN 952-12-0492-3, 85-112, 1999.

[39] R. Ifikes, T.Kehler, *The Role Of Frame-Based Representation In Reasoning*. Communications of the ACM, Volume 28 Number 9, 1985.

[40] R. J. Schalkoff, *Artificial Intelligence An Engineering Approach*, Clemson University, McGraw-Hill publishing Co.

[41] R. Karni ,M. Kaner, *CBR-based Knowledge-focusing for Training in Decision making within Project Management Processes*, white paper, Robert Gordon University, Aberdeen, Scotland, 2002.

[42] R. L. pezde, M. ntaras , E. Plaza, *Case-Based Reasoning : An Overview*, IIIA-Artificial Intelligence Research Institute, CSIC-Spanish National Research Council ,paper.

[43] S. Bridges, J. Hodges, C. Sparrow, *A Comparative Study of Different Methods for Representing and Reasoning with Uncertainty for a Waste Characterization Task*, Technical Report No. MSU-970613, Mississippi State University, USA, 1997.

[44] S. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, ISBN 0-13-103805-2, 2003.

[45] S. Russell, P. Norvig,, survey: pp. 200-213 (*Logic, Propositional Logic: A Very Simple Logic, Reasoning Patterns in Propositional Logic*), survey: 240-260.

[46] S. Schafrik. *A New Style of Simulation Model for Mining Systems*, M.Sc Thesis, Faculty of the Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 2001.

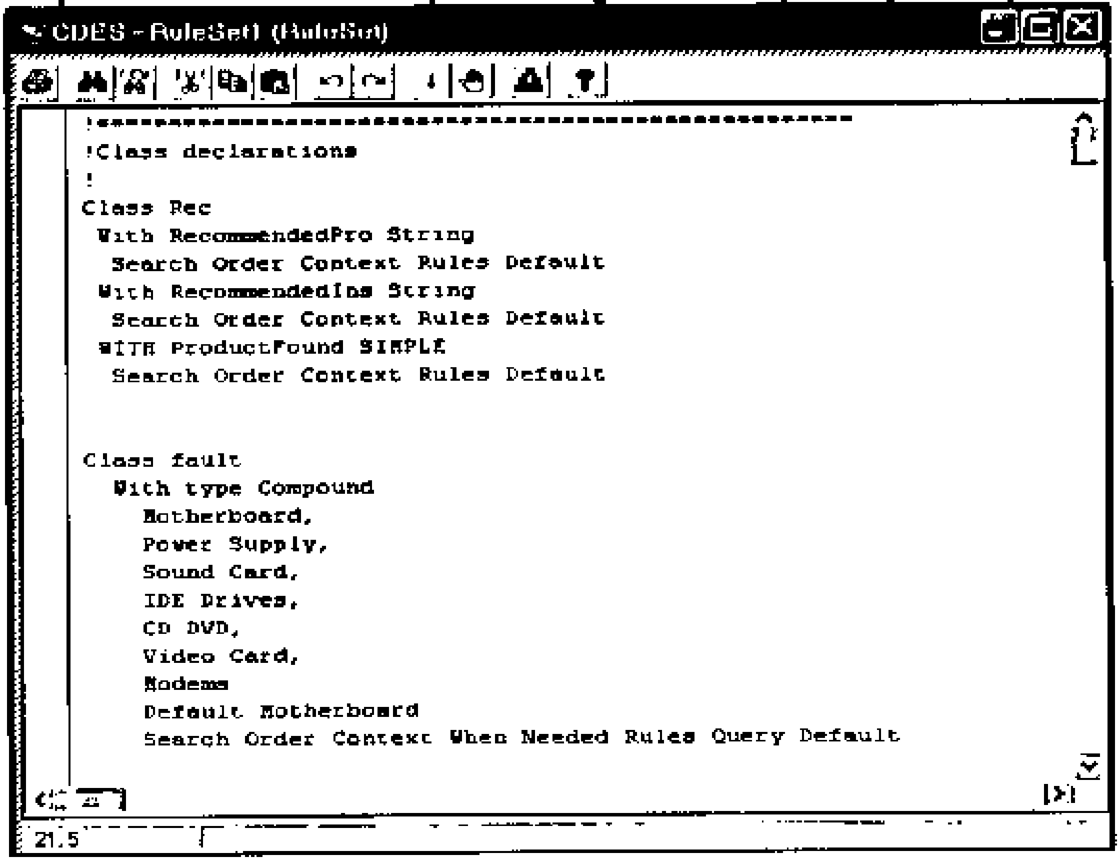
[47] W. POHL, *Logic-Based Representation and Reasoning for User Modeling Shell Systems*, User Modeling and User-Adapted Interaction 9: 217–282, Kluwer Academic Publishers. Printed in the Netherlands, 1999.

[48] W. Siler , J. Buckley, *Fuzzy Expert Systems and Fuzzy Reasoning*, ISBN 0-471-38859-9 , John Wiley & Sons Inc, 2005.

[49] W. Xijun, *A Web-Based Case-Based Reasoning Shell*, M.Sc thesis, University of Wyoming, 2000.

Appendix A

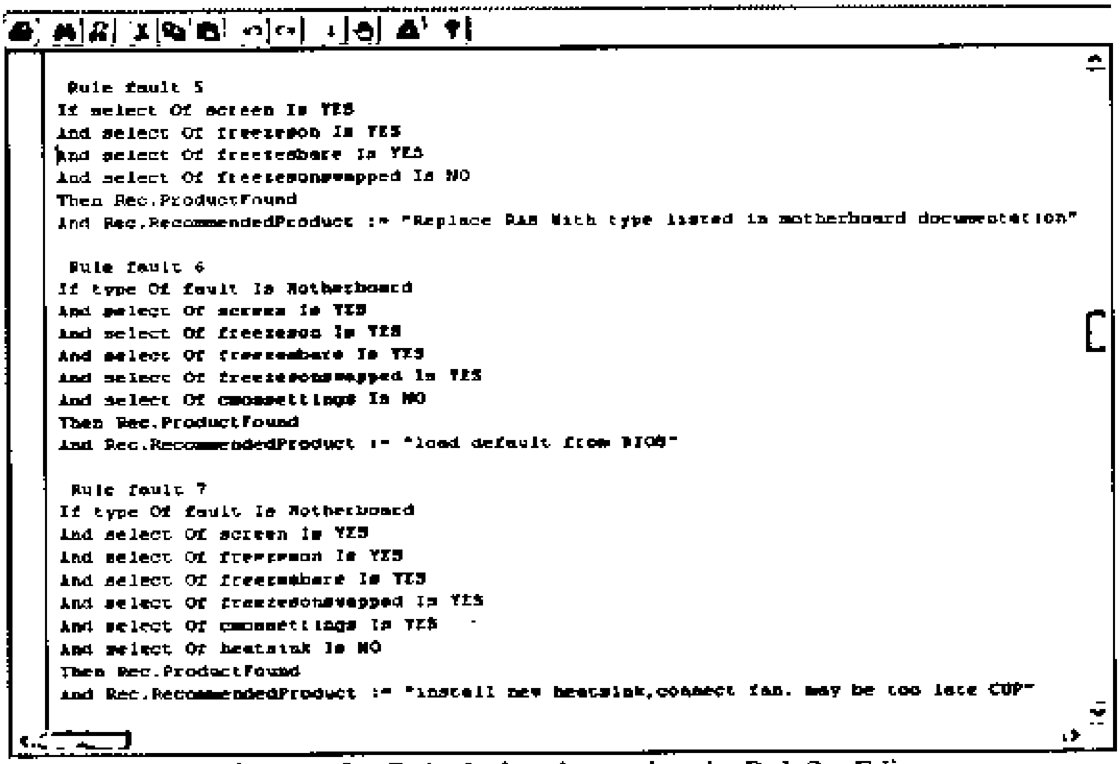
Implementation of Computer Diagnostic Expert System (CDES)



```
!=======================================================
!Class declarations
!
Class Rec
 With RecommendedPro String
  Search Order Context Rules Default
 With RecommendedIns String
  Search Order Context Rules Default
 WITH ProductFound SIMPLE
  Search Order Context Rules Default


Class fault
  With type Compound
    Motherboard,
    Power Supply,
    Sound Card,
    IDE Drives,
    CD DVD,
    Video Card,
    Modems
    Default Motherboard
    Search Order Context When Needed Rules Query Default
```

Figure A.1 Classes declaration in ruleset



```
 Rule fault 5
If select Of screen Is YES
And select Of freezeson Is YES
And select Of freezesonbare Is YES
And select Of freezesonswapped Is NO
Then Rec.ProductFound
And Rec.RecommendedProduct := "Replace RAM With type listed in motherboard documentation"

 Rule fault 6
If type Of fault Is Motherboard
And select Of screen Is YES
And select Of freezeson Is YES
And select Of freezesonbare Is YES
And select Of freezesonswapped Is YES
And select Of cmossettings Is NO
Then Rec.ProductFound
And Rec.RecommendedProduct := "load default from BIOS"

 Rule fault 7
If type Of fault Is Motherboard
And select Of screen Is YES
And select Of freezeson Is YES
And select Of freezesonbare Is YES
And select Of freezesonswapped Is YES
And select Of cmossettings Is YES
And select Of heatsink Is NO
Then Rec.ProductFound
And Rec.RecommendedProduct := "install new heatsink,connect fan. may be too late CUP"
```

Figure A.2 Rule declarations using the RuleSet Editor

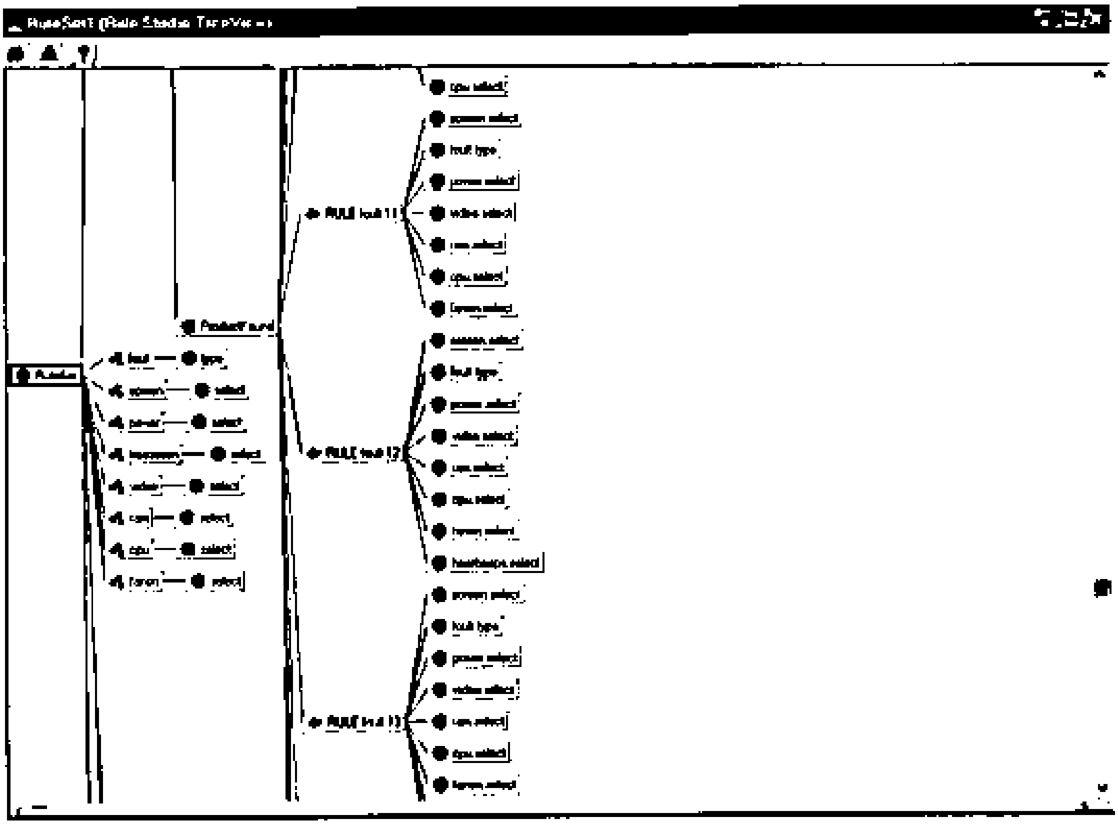


Figure A.3 RuleSet Tree View

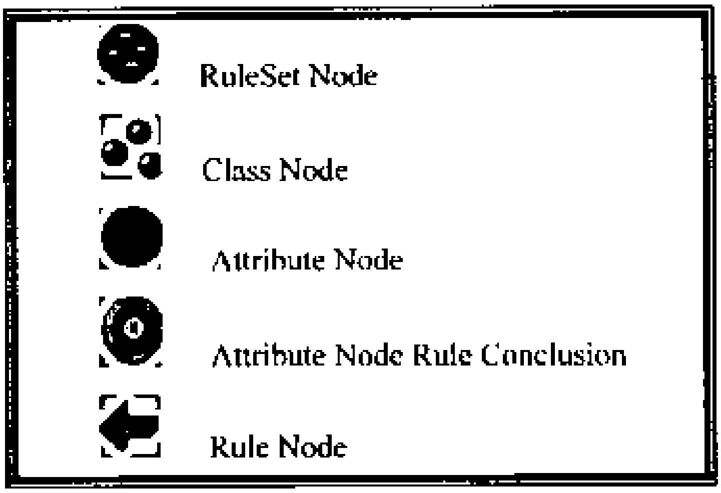
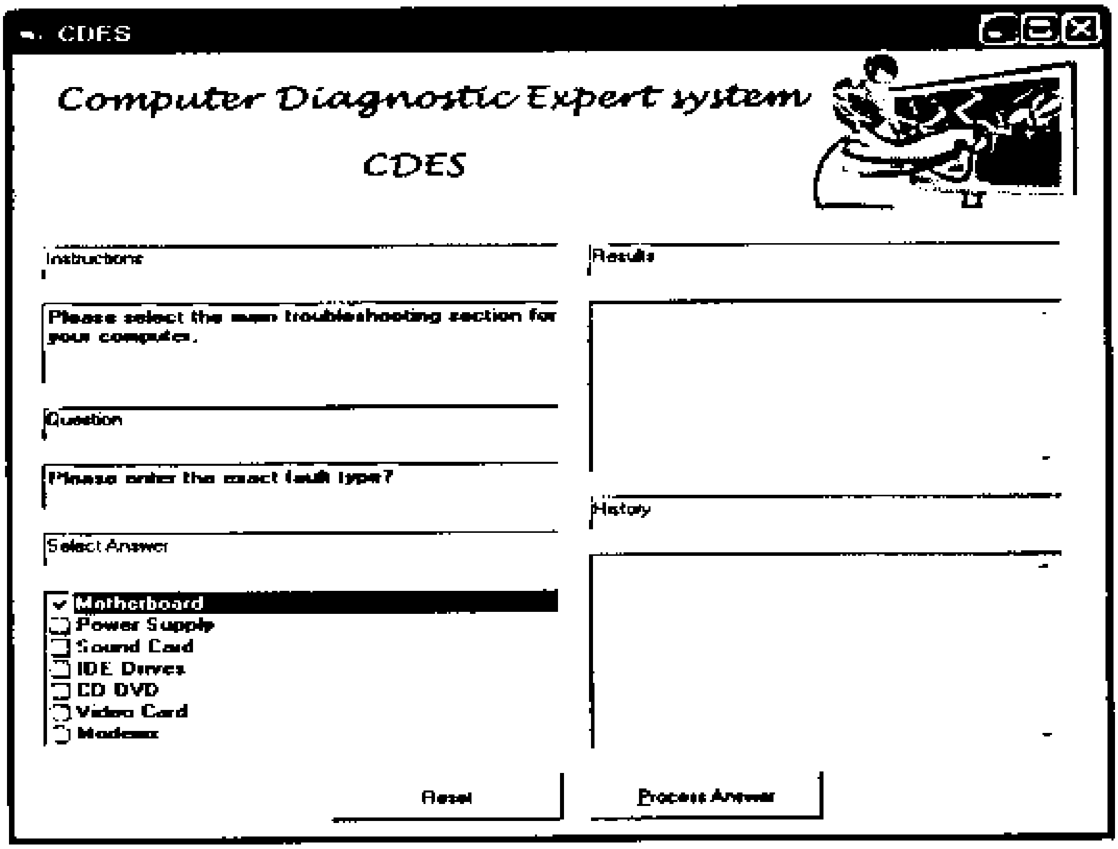


Figure A.4 Tree View Node Types



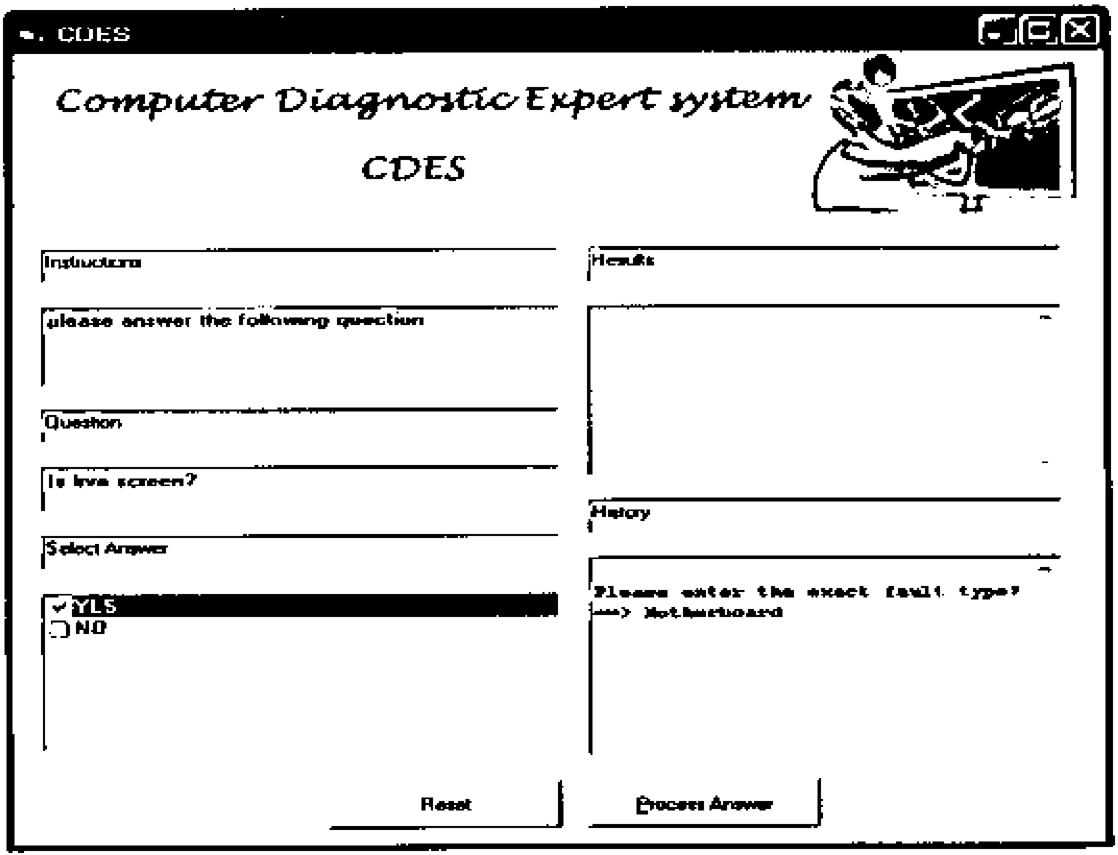
Instructions: Please select the main troubleshooting section for your computer.

Question: Please enter the exact fault type?

Select Answer:

- ☑ Motherboard
- □ Power Supply
- □ Sound Card
- □ IDE Drives
- □ CD DVD
- □ Video Card
- □ Modems

Figure A.5 CDES user interface



Instructions: please answer the following question

Question: Is live screen?

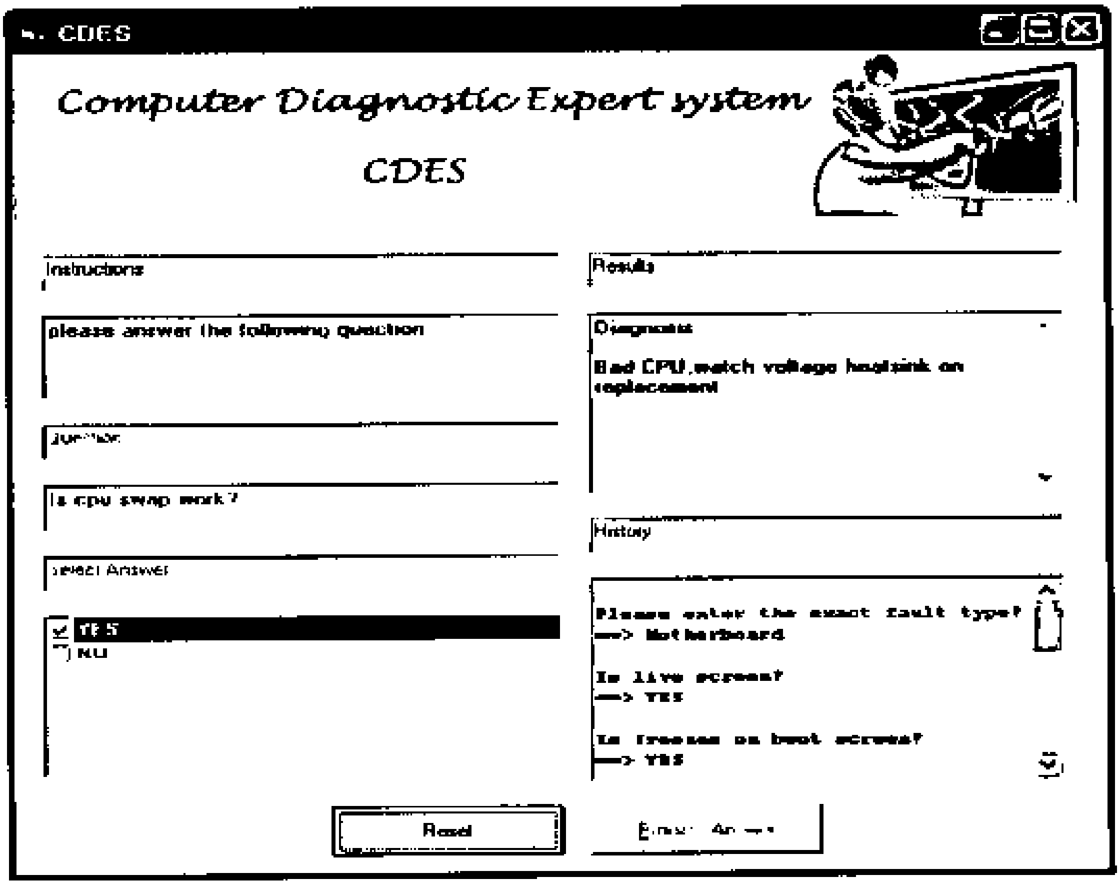
Select Answer:

- ☑ YES
- □ NO

History:

```
Please enter the exact fault type?
---> Motherboard
```

Figure A.6 Select the answer to question display



Computer Diagnostic Expert system

CDES

Instructions

please answer the following question

Question

Is cpu swap work ?

Select Answer

☑ YES
□ NO

Results

Diagnosis

Bad CPU,watch voltage heatsink on replacement

History

```
Please enter the exact fault type?
==> Motherboard

Is live screen?
==> YES

Is freezes on boot screen?
==> YES
```

Figure A.7 Visualizing the Results of Diagnostics

Appendix B

Implementation of Decision-Support Expert System (DSES)

Linguistic Variable: *Mean Delay, m*		
Linguistic Value	Notation	Numerical Range (normalised)
Very Short	VS	[0, 0.3]
Short	S	[0.1, 0.5]
Medium	M	[0.4, 0.7]
Linguistic Variable: *Number of Servers, s*		
Linguistic Value	Notation	Numerical Range (normalised)
Small	S	[0, 0.35]
Medium	M	[0.30, 0.70]
Large	L	[0.60, 1]
Linguistic Variable: *Repair Utilisation Factor*, ρ		
Linguistic Value	Notation	Numerical Range
Low	L	[0, 0.6]
Medium	M	[0.4, 0.8]
High	H	[0.6, 1]
Linguistic Variable: *Number of Spares, n*		
Linguistic Value	Notation	Numerical Range (normalised)
Very Small	VS	[0, 0.30]
Small	S	[0, 0.40]
Rather Small	RS	[0.25, 0.45]
Medium	M	[0.30, 0.70]
Rather Large	RL	[0.55, 0.75]
Large	L	[0.60, 1]
Very Large	VL	[0.70, 1]

Table B.1 Linguistic variables and their ranges

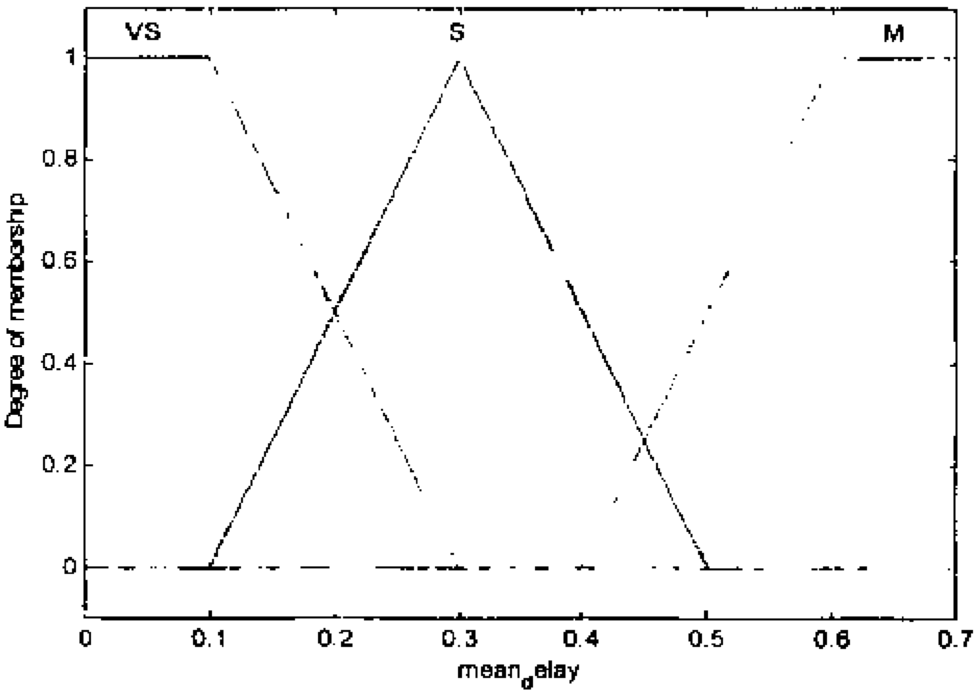


Figure B.1 Fuzzy sets of Mean Delay m

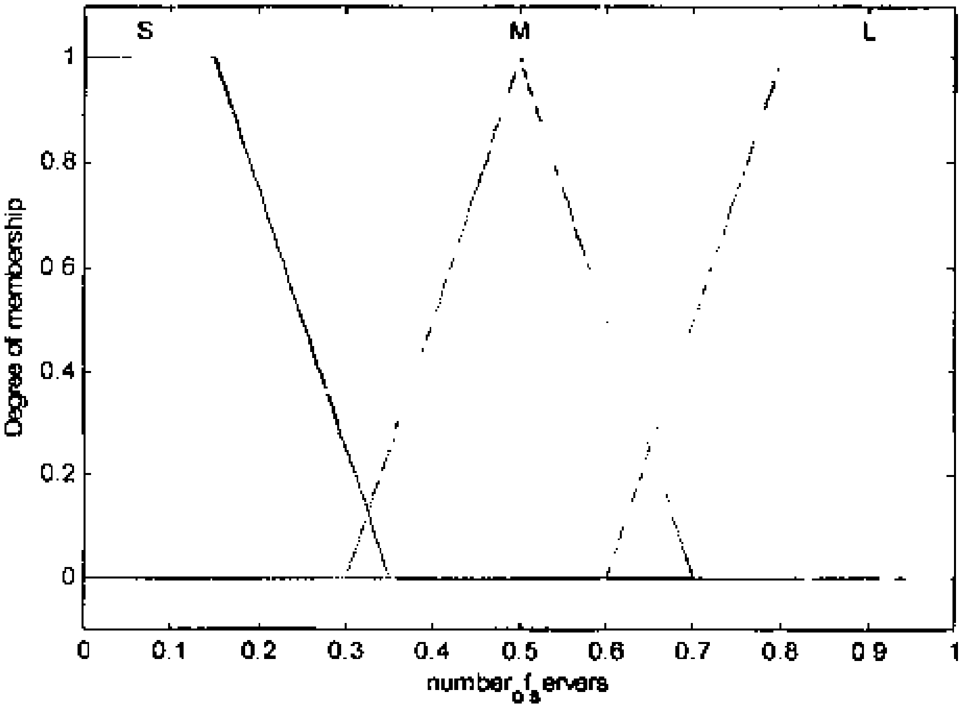


Figure B.2 Fuzzy sets of Number of Servers s

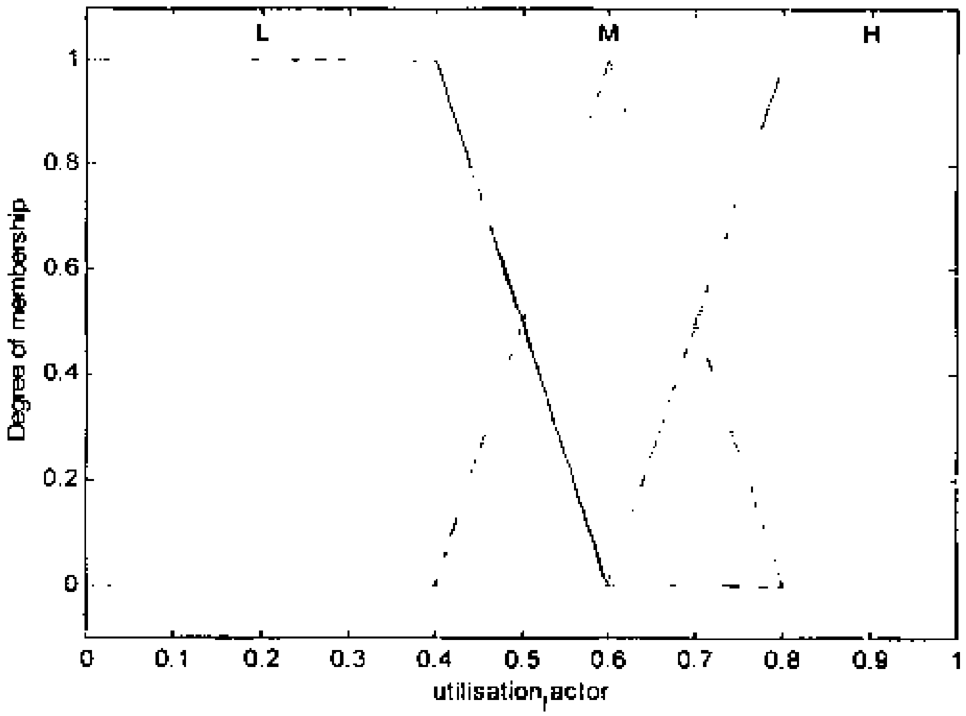


Figure B.3 Fuzzy sets of Repair Utilisation Factor ρ

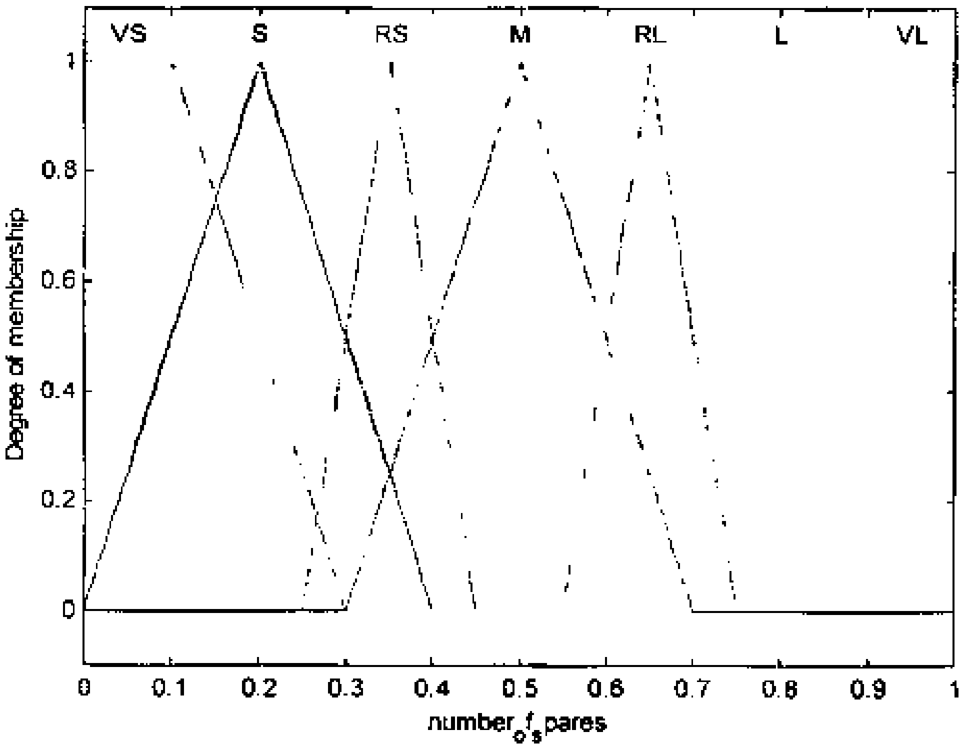
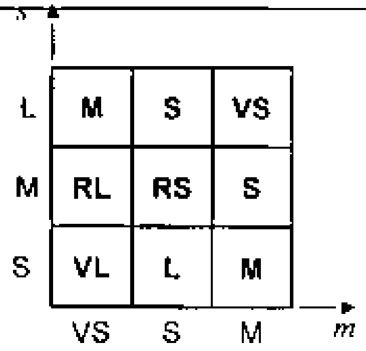


Figure B.4 Fuzzy sets of Number of Spares n

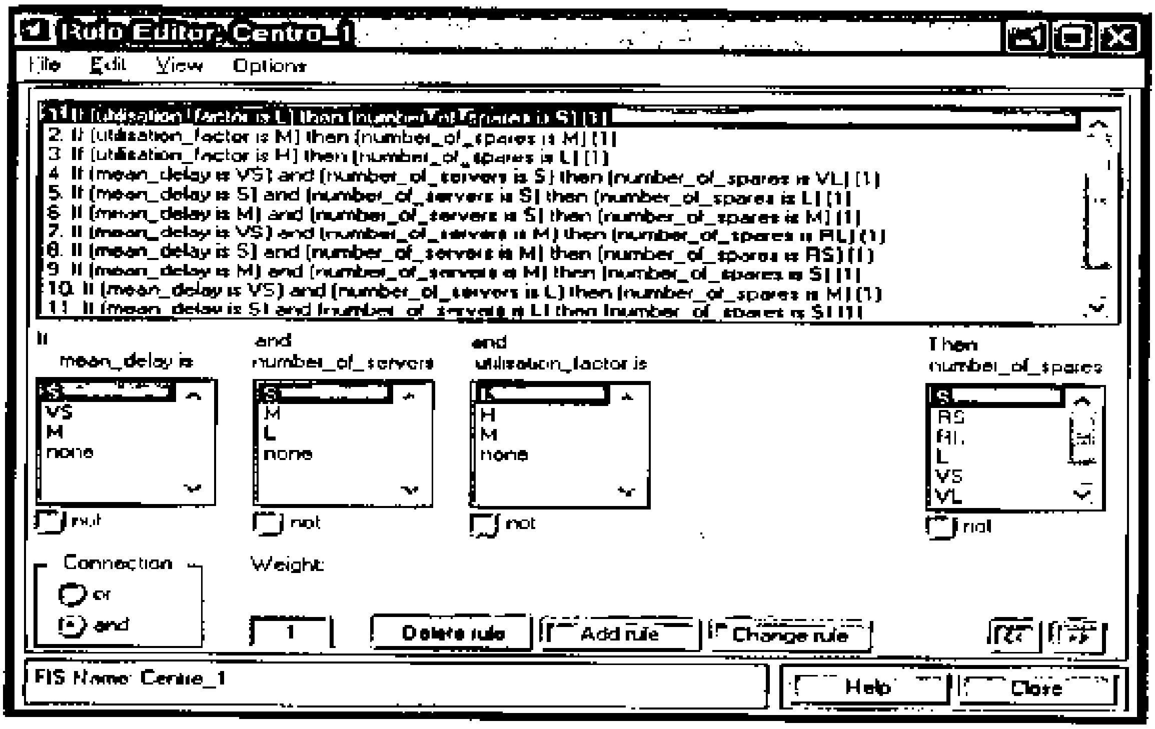


s				
L	M	S	VS	
M	RL	RS	S	
S	VL	L	M	
	VS	S	M	m

Figure B.5 The square FAM representation

Rule	m	s	ρ	n	Rule	m	s	ρ	n	Rule	m	s	ρ	n
1	VS	S	L	VS	10	VS	S	M	S	19	VS	S	H	VL
2	S	S	L	VS	11	S	S	M	VS	20	S	S	H	L
3	M	S	L	VS	12	M	S	M	VS	21	M	S	H	M
4	VS	M	L	VS	13	VS	M	M	RS	22	VS	M	H	M
5	S	M	L	VS	14	S	M	M	S	23	S	M	H	M
6	M	M	L	VS	15	M	M	M	VS	24	M	M	H	S
7	VS	L	L	S	16	VS	L	M	M	25	VS	L	H	RL
8	S	L	L	S	17	S	L	M	RS	26	S	L	H	M
9	M	L	L	VS	18	M	L	M	S	27	M	L	H	RS

Table B.2 The rule table



1. If (utilisation_factor is L) then (number_of_spares is S) (1)
2. If (utilisation_factor is M) then (number_of_spares is M) (1)
3. If (utilisation_factor is H) then (number_of_spares is L) (1)
4. If (mean_delay is VS) and (number_of_servers is S) then (number_of_spares is VL) (1)
5. If (mean_delay is S) and (number_of_servers is S) then (number_of_spares is L) (1)
6. If (mean_delay is M) and (number_of_servers is S) then (number_of_spares is M) (1)
7. If (mean_delay is VS) and (number_of_servers is M) then (number_of_spares is RL) (1)
8. If (mean_delay is S) and (number_of_servers is M) then (number_of_spares is RS) (1)
9. If (mean_delay is M) and (number_of_servers is M) then (number_of_spares is S) (1)
10. If (mean_delay is VS) and (number_of_servers is L) then (number_of_spares is M) (1)
11. If (mean_delay is S) and (number_of_servers is L) then (number_of_spares is S) (1)

If mean_delay is: S, VS, M, none; □ not

and number_of_servers: S, M, L, none; □ not

and utilisation_factor is: L, H, M, none; □ not

Then number_of_spares: S, RS, RL, L, VS, VL; □ not

Connection: ○ or ● and

Weight: 1

FIS Name: Centre_1

Figure B.6 Rule Base 1

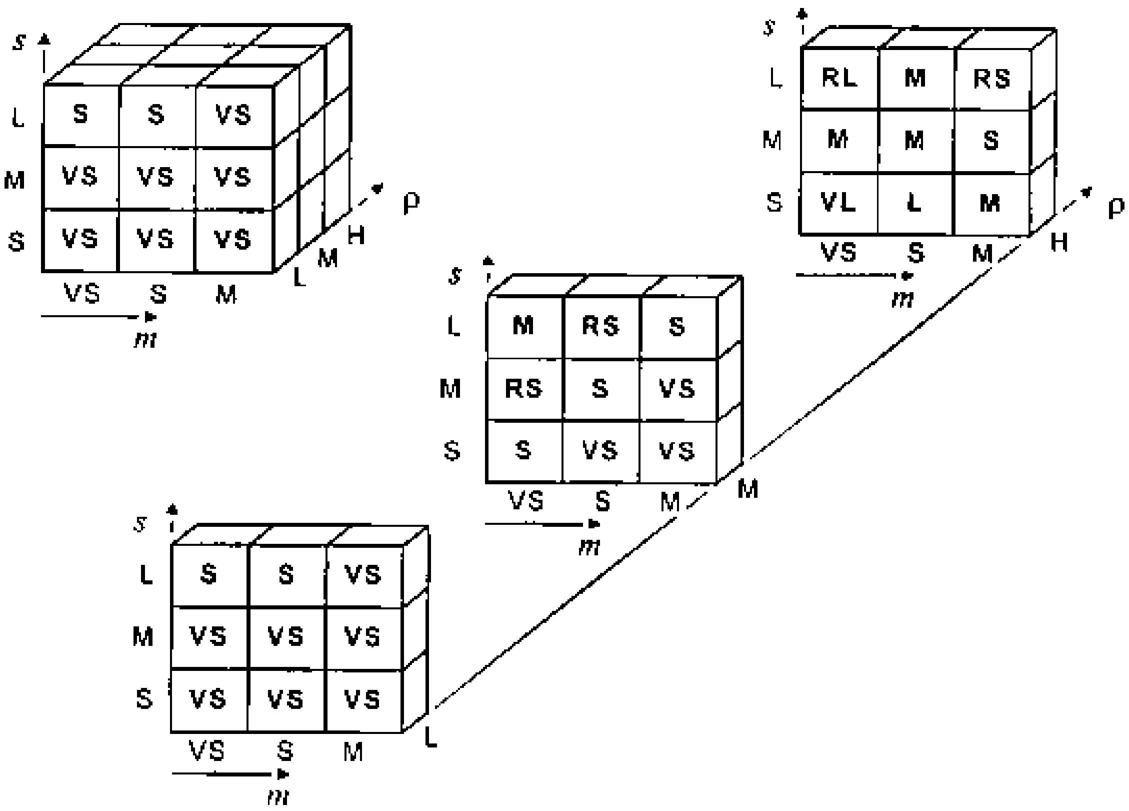
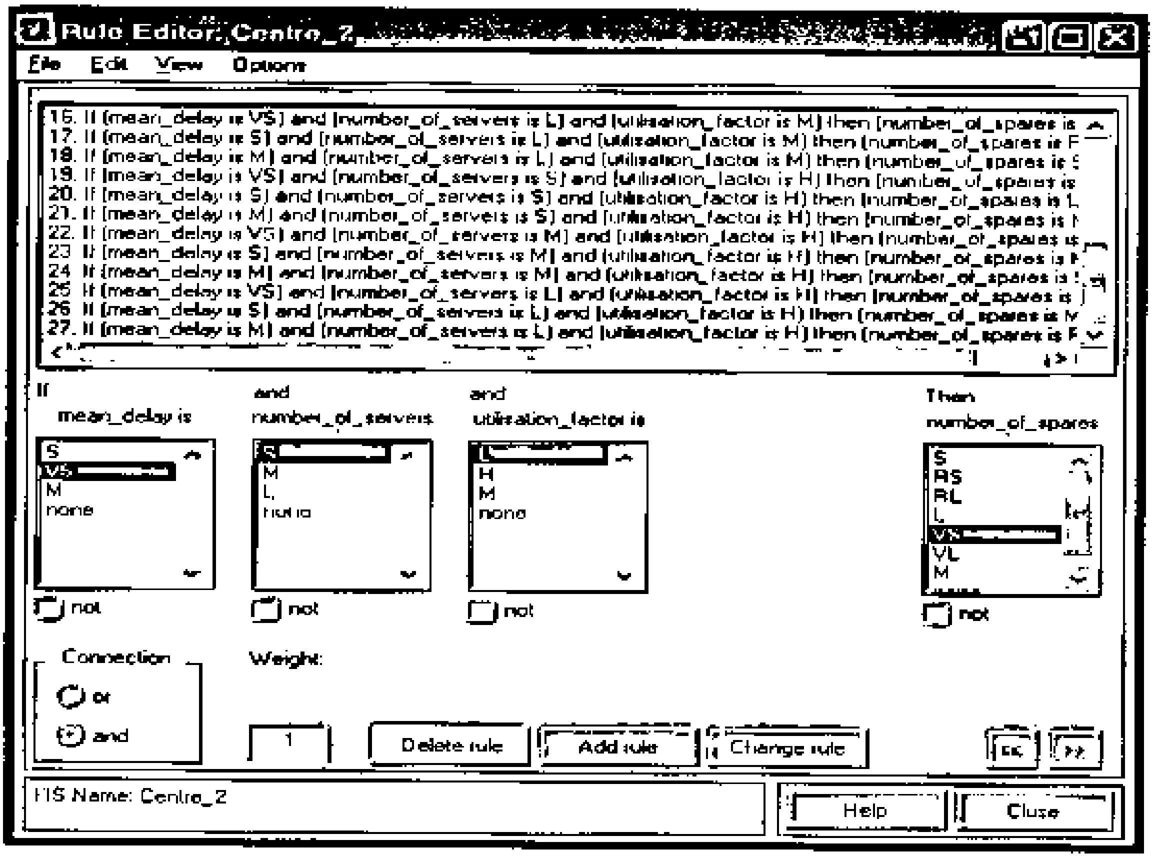


Figure B.7 Cube FAM of Rule Base 2



16. If (mean_delay is VS) and (number_of_servers is L) and (utilisation_factor is M) then (number_of_spares is
17. If (mean_delay is S) and (number_of_servers is L) and (utilisation_factor is M) then (number_of_spares is F
18. If (mean_delay is M) and (number_of_servers is L) and (utilisation_factor is M) then (number_of_spares is S
19. If (mean_delay is VS) and (number_of_servers is S) and (utilisation_factor is H) then (number_of_spares is
20. If (mean_delay is S) and (number_of_servers is S) and (utilisation_factor is H) then (number_of_spares is L
21. If (mean_delay is M) and (number_of_servers is S) and (utilisation_factor is H) then (number_of_spares is M
22. If (mean_delay is VS) and (number_of_servers is M) and (utilisation_factor is H) then (number_of_spares is
23. If (mean_delay is S) and (number_of_servers is M) and (utilisation_factor is H) then (number_of_spares is M
24. If (mean_delay is M) and (number_of_servers is M) and (utilisation_factor is H) then (number_of_spares is S
25. If (mean_delay is VS) and (number_of_servers is L) and (utilisation_factor is H) then (number_of_spares is
26. If (mean_delay is S) and (number_of_servers is L) and (utilisation_factor is H) then (number_of_spares is M
27. If (mean_delay is M) and (number_of_servers is L) and (utilisation_factor is H) then (number_of_spares is F

If mean_delay is: S, VS, M, none — selected: VS

and number_of_servers: S, M, L, none — selected: S

and utilisation_factor is: L, H, M, none — selected: L

Then number_of_spares: S, RS, RL, L, VS, VL, M — selected: VS

□ not □ not □ not □ not

Connection: ○ or ● and

Weight: 1

FIS Name: Centre_2

Figure B.8 Rule Base 2

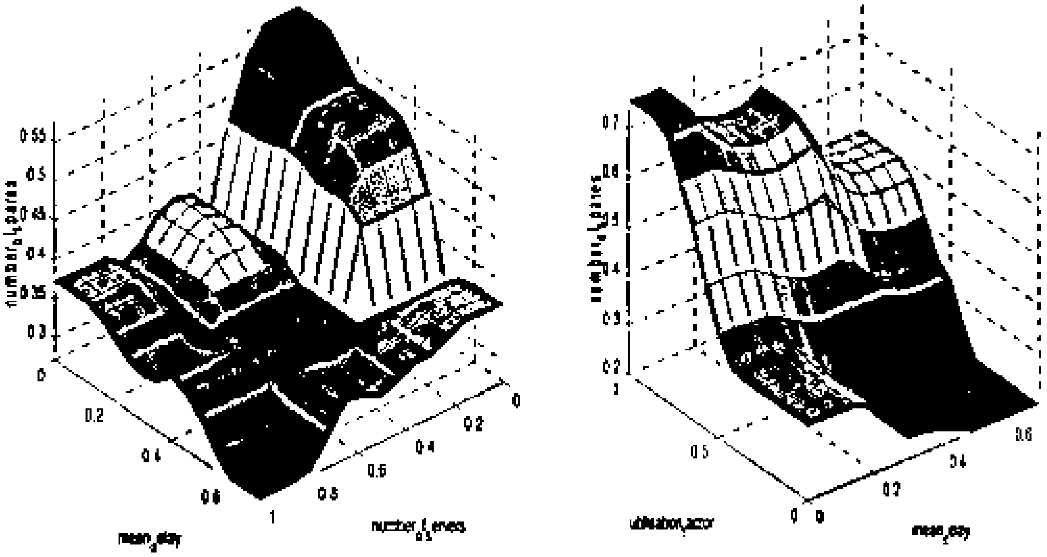


Figure B.9 Three-dimensional plots for Rule Base 1

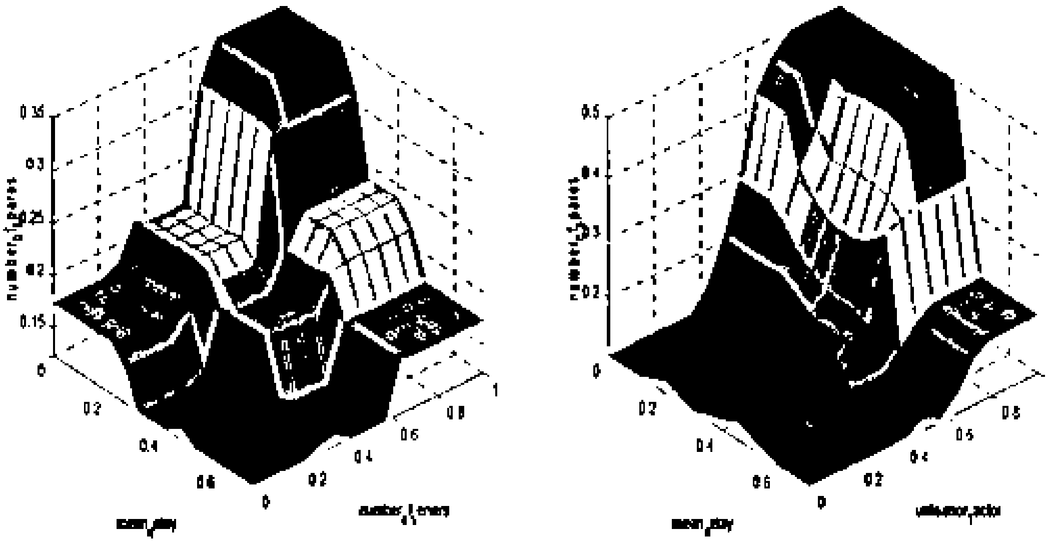
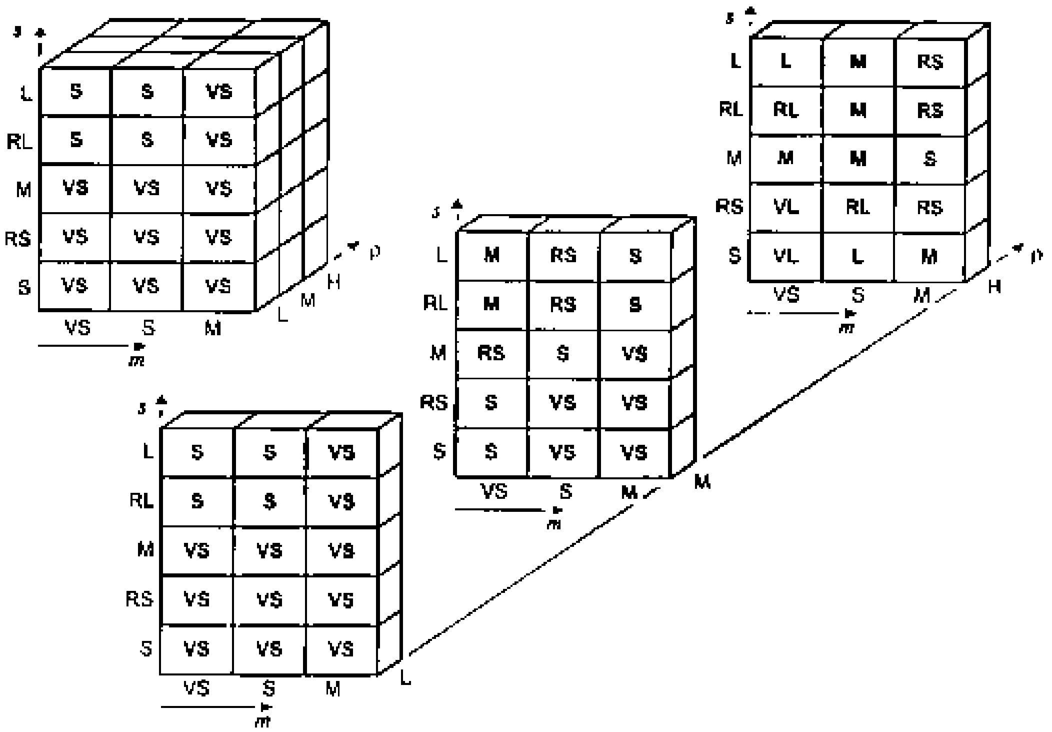


Figure B.10 Three-dimensional plots for Rule Base 2

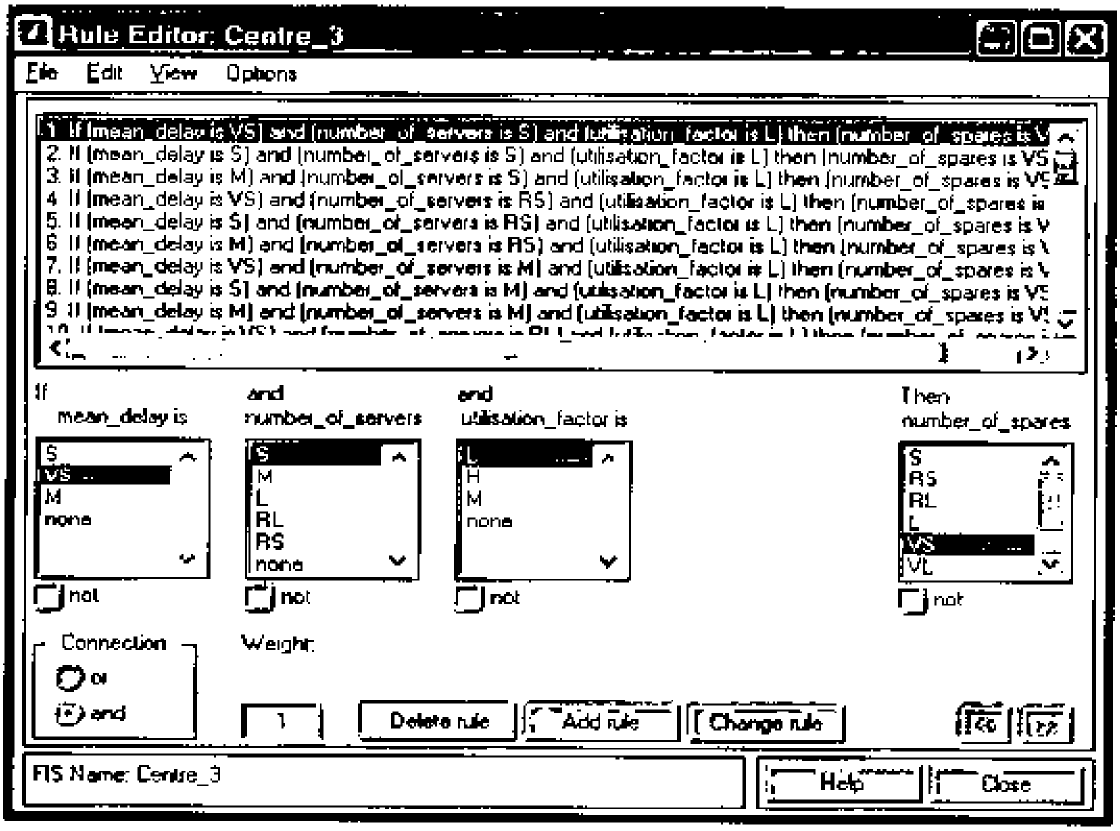


	VS	S	M
L	S	S	VS
RL	S	S	VS
M	VS	VS	VS
RS	VS	VS	VS
S	VS	VS	VS

	VS	S	M
L	M	RS	S
RL	M	RS	S
M	RS	S	VS
RS	S	VS	VS
S	S	VS	VS

	VS	S	M
L	L	M	RS
RL	RL	M	RS
M	M	M	S
RS	VL	RL	RS
S	VL	L	M

Figure B.11 Cube FAM of Rule Base 3



1. If (mean_delay is VS) and (number_of_servers is S) and (utilisation_factor is L) then (number_of_spares is V
2. If (mean_delay is S) and (number_of_servers is S) and (utilisation_factor is L) then (number_of_spares is VS
3. If (mean_delay is M) and (number_of_servers is S) and (utilisation_factor is L) then (number_of_spares is VS
4. If (mean_delay is VS) and (number_of_servers is RS) and (utilisation_factor is L) then (number_of_spares is
5. If (mean_delay is S) and (number_of_servers is RS) and (utilisation_factor is L) then (number_of_spares is V
6. If (mean_delay is M) and (number_of_servers is RS) and (utilisation_factor is L) then (number_of_spares is \
7. If (mean_delay is VS) and (number_of_servers is M) and (utilisation_factor is L) then (number_of_spares is \
8. If (mean_delay is S) and (number_of_servers is M) and (utilisation_factor is L) then (number_of_spares is VS
9. If (mean_delay is M) and (number_of_servers is M) and (utilisation_factor is L) then (number_of_spares is VS

If mean_delay is: S, VS (selected), M, none; □ not

and number_of_servers is: S (selected), M, L, RL, RS, none; □ not

and utilisation_factor is: L (selected), H, M, none; □ not

Then number_of_spares: S, RS, RL, L, VS (selected), VL; □ not

Connection: ○ or ● and

Weight: 1

FIS Name: Centre_3

Figure B.12 Rule Base 3

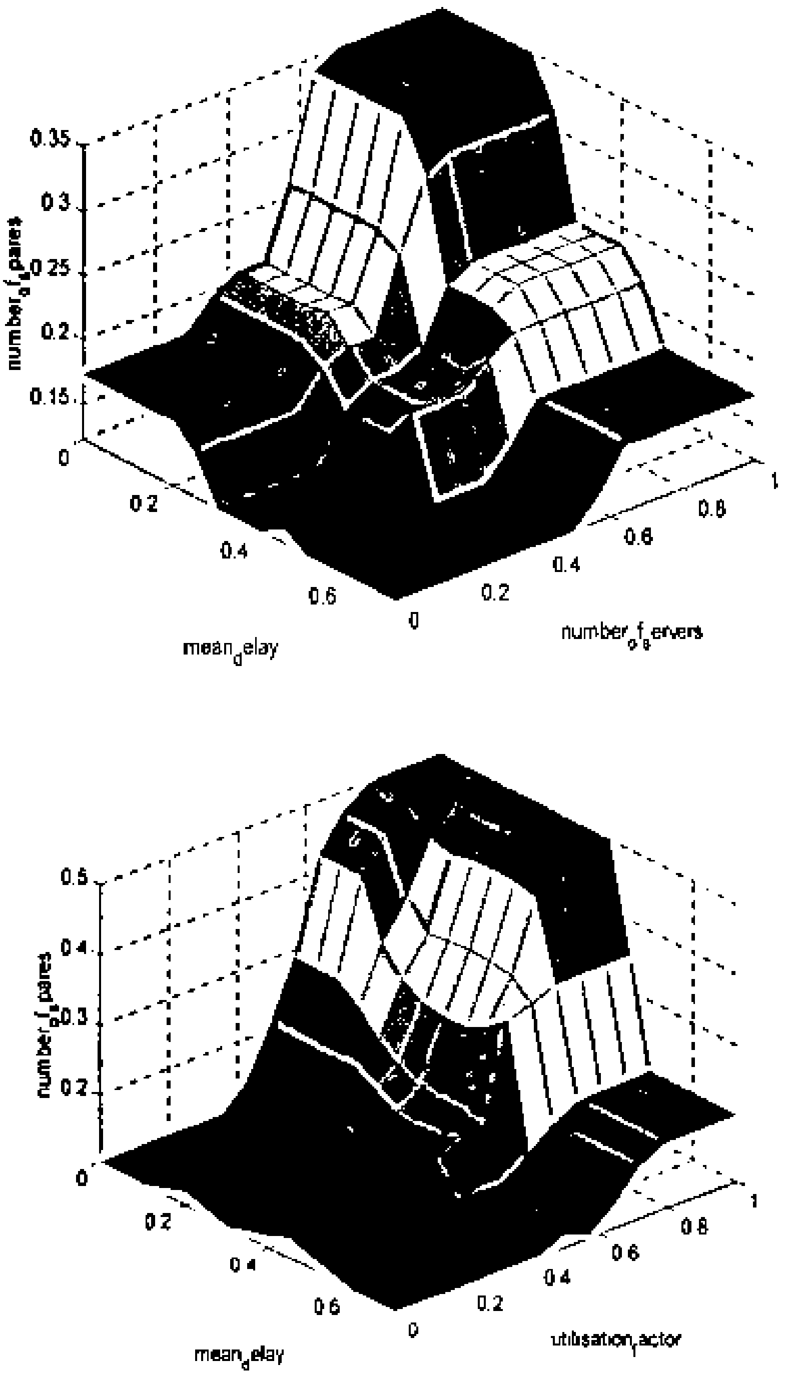


Figure B.13 Three-dimensional plots for Rule Base 3

Appendix C

Preliminary CDES Rules

RULE 1

IF type of fault is Motherboard
AND live of screen is YES
AND freezes on boot screen is NO
THEN the diagnostic is "proceed to motherboard performance chart"

RULE 2

IF type of fault is Motherboard
AND live of screen is NO
AND power diagnostic done is NO
THEN the diagnostic is "proceed to power supply failure"

RULE 3

IF type of fault is Motherboard
AND live of screen is NO
AND power diagnostic done is YES
AND video diagnostic done is YES
AND RAM matched seated is NO
THEN the diagnostic is "Reseat RAM positive lever lock. Install pairs if required"

RULE 4

IF type of fault is Motherboard
AND live of screen is NO
AND power diagnostic done is YES
AND video diagnostic done is NO
THEN the diagnostic is "proceed to video failure diagnostic chart"

RULE 5

IF type of fault is Motherboard
AND live of screen is YES
AND freezes on boot screen is YES
AND freezes bare bones is NO
THEN the diagnostic is "proceed to conflict resolution chart"

RULE 6

IF type of fault is Motherboard
AND live of screen is YES
AND freezes on boot screen is YES
AND freezes bare bones is YES
AND freezes on swapped RAM is NO
THEN the diagnostic is "Replace RAM with type listed in motherboard documentation"

RULE 7
IF type of fault is Motherboard
AND live of screen is YES
AND freezes on boot screen is YES
AND freezes bare bones is YES
AND freezes on swapped RAM is YES
AND CMOS setting default NO
THEN the diagnostic is "load default from BIOS"

RULE 8
IF type of fault is Motherboard
AND live of screen is YES
AND freezes on boot screen is YES
AND freezes bare bones is YES
AND freezes on swapped RAM is YES
AND CMOS setting default YES
AND heat sink active is NO
THEN the diagnostic is "install new heat sink, connect fan. May be too late CUP"

RULE 9
IF type of fault is Motherboard
AND live of screen is YES
AND freezes on boot screen is YES
AND freezes bare bones is YES
AND freezes on swapped RAM is YES
AND CMOS setting default YES
AND heat sink active is YES
AND runs on bench is YES
THEN the diagnostic is locate short binding, or swap case"

RULE 10
IF type of fault is Motherboard
AND live of screen is YES

```
        AND    freezes on boot screen is YES
        AND    freezes bare bones is YES
        AND    freezes on swapped RAM is YES
        AND    CMOS setting default YES
        AND    heat sink active is YES
        AND    runs on bench is NO
        AND    CPU swap work is NO
        THEN   the diagnostic is "Motherboard Bad"

RULE 11
        IF      type of fault is Motherboard
         AND   live of screen is YES
        AND    freezes on boot screen is YES
        AND    freezes bare bones is YES
        AND   freezes on swapped RAM is YES
        AND    CMOS setting default YES
        AND    heat sink active is YES
        AND    runs on bench is NO
        AND    CPU swap work is YES
        THEN   the diagnostic is "Bad CPU, watch voltage heat sink on replacement"

RULE 12
        IF      type of fault is Motherboard
         AND   live of screen is NO
        AND    power diagnostic done is YES
        AND   video diagnostic done is YES
        AND    RAM matched seated is YES
        AND   CPU seated flat is NO
        THEN   the diagnostic is "Reseat CPU and heat sink"

RULE 13
        IF      type of fault is Motherboard
         AND   live of screen is NO
        AND    power diagnostic done is YES
        AND    video diagnostic done is YES
        AND    RAM matched seated is YES
        AND    CPU seated flat is YES
        AND    fan on heat sink active is NO
        THEN   the diagnostic is "check fan power point replace"
```

```
RULE 14
    IF     type of fault is Motherboard
    AND    live of screen is NO
    AND    power diagnostic done is YES
    AND    video diagnostic done is YES
    AND    RAM matched seated is YES
    AND    CPU seated flat is YES
    AND    fan on heat sink active is YES
    AND    hear beeps is YES
    THEN   the diagnostic is "Replace RAM with type listed in motherboard documentation"

RULE 15
    IF     type of fault is Motherboard
    AND    live of screen is NO
    AND    power diagnostic done is YES
    AND    video diagnostic done is YES
    AND    RAM matched seated is YES
    AND    CPU seated flat is YES
    AND    fan on heat sink active is YES
    AND    hear beeps is NO
    AND    default motherboard settings is NO
    THEN   the diagnostic is "Restore default motherboard settings with jumpers or switches"

RULE 16
    IF     type of fault is Motherboard
    AND    live of screen is NO
    AND    power diagnostic done is YES
    AND    video diagnostic done is YES
    AND    RAM matched seated is YES
    AND    CPU seated flat is YES
    AND    fan on heat sink active is YES
    AND    hear beeps is NO
    AND    default motherboard settings is YES
    AND    runs on bench is NO
    AND    CPU swap work is YES
    THEN   the diagnostic is "Bad CPU, watch voltage heat sink on replacement"

RULE 17
    IF     type of fault is Motherboard
    AND    live of screen is NO
```

AND power diagnostic done is YES

AND video diagnostic done is YES

AND RAM matched seated is YES

AND CPU seated flat is YES

AND fan on heat sink active is YES

AND hear beeps is NO

AND default motherboard settings is YES

AND runs on bench is NO

AND CPU swap work is NO

THEN the diagnostic is "Motherboard Bad"

RULE 18

IF type of fault is Motherboard

AND live of screen is NO

AND power diagnostic done is YES

AND video diagnostic done is YES

AND RAM matched seated is YES

AND CPU seated flat is YES

AND fan on heat sink active is YES

AND hear beeps is NO

AND default motherboard settings is YES

And runs on bench is NO

THEN the diagnostic is locate short binding, or swap case"

الفصل الأول:- يحتوي على مقدمة للذكاء الاصطناعي والاستنتاج وتعريفه وبعض المفاهيم الأساسية في الدراسة واهم الأبحاث المتعلقة بموضوع الدراسة وأهدافها.

الفصل الثاني:- يقدم طرق الاستنتاج التام وهي طريقة الاستنتاج المبني على القواعد، المنطق والإطارات.

الفصل الثالث:- يوضح طرق الاستنتاج الغير التام وهي طريقة الاستنتاج المبني على الحالة، النموذج والاستنتاج الضبابي.

الفصل الرابع:- يضمن مقارنة وتقييم لطرق الاستنتاج المذكورة.

الفصل الخامس:- يحتوي على تصميم وتطبيق الحالات الدراسية المقدمة.

الفصل السادس:- يقدم الاستنتاجات من الدراسة وملخص البحث.