A RULE-BASED REASONING DECISION SUPPORT SYSTEM FOR AS-532 COUGAR HELICOPTERS' MAINTENANCE PERSONNEL

The Institute of Economics and Social Sciences
of
Bilkent University

by

Hüseyin Kurtuluş DOĞAN

In Partial Fulfillment of the Requirements for the Degree
of
MASTER OF BUSINESS ADMINISTRATION

in

THE DEPARTMENT OF MANAGEMENT
BİLKENT UNIVERSITY
ANKARA

June 2004

I certify that I have read this thesis and have found that it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Business Administration.

Assistant Professor Yavuz GÜNALAY

Supervisor

I certify that I have read this thesis and have found that it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Business Administration.

Professor Dilek ÖNKAL

Examining Committee Member

I certify that I have read this thesis and have found that it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Business Administration.

Associate Professor A. Kadir VAROĞLU

Examining Committee Member

Approval of Institute of Economics and Social Sciences.

Professor Kürşat AYDOĞAN

Director

ABSTRACT

A RULE-BASED REASONING DECISION SUPPORT SYSTEM FOR AS-532 COUGAR HELICOPTERS' MAINTENANCE PERSONNEL

DOĞAN, Hüseyin Kurtuluş

M.B.A. Thesis

Supervisor: Assist. Prof. Yavuz GÜNALAY

June 2004

Improvements in the aviation technologies have made the aircrafts more reliable, more capable in terms of operational necessities and as a result more technological. Today, Turkish Army Aviation units possess helicopters that are equipped with the technologically up-to-date equipments. Those new generation helicopters brought their advantages in terms of performance, speed, and reliability as well as the disadvantages in terms of maintenance load, expensive parts and required deep system knowledge. At this point, there is a trade off that takes place between the development and the system integration and maintenance necessities of these ultimate equipments. Helicopter systems turn out to be more interactive. Those developments make it easier to fly for the user pilots, but more difficult to perform the maintenance necessities for the maintenance personnel.

New maintenance necessities of the developed equipments and interactive systems make it more difficult to keep the helicopter fleets flying and operational. Furthermore this situation increases the need for the domain experts. Number of unexpected failures and the required time for fault isolation are increasing by the acquisitions of developed helicopters.

Acquired experience is an important reference for the maintenance personnel in solving the problems as well as the technical documents of the helicopters. In this study, a decision support system in rule-based reasoning is formed in order to help AS-532 COUGAR helicopters' maintenance personnel. Extraordinary failures, their possible causes, warning to users and recommended solution procedures are presented in the design of "case". Users can reach to these cases by following the failure related attribute-value pairs. With the help of this decision support system, it will be possible to share the expertise of the COUGAR experts with the inexperienced personnel. As a result, the percentage of true "fault identification" will increase and required time for fault isolation will decrease. This expertise can also be used in training procedures.

Keywords: Knowledge Management, Rule-Based Reasoning, Decision Support System.

ÖZET

AS-532 COUGAR HELİCOPTERLERİ BAKIM PERSONELİ İÇİN KURAL TABANLI KARAR DESTEK SİSTEMİ

DOĞAN, Hüseyin Kurtuluş

M.B.A. Tezi

Tez Yöneticisi: Yard. Doç. Dr. Yavuz GÜNALAY

Haziran 2004

Havacılık teknolojilerindeki gelişmeler hava araçlarını daha güvenilir, operasyonel gereklilikler açısından daha kabiliyetli ve daha teknolojik yapmıştır. Günümüzde Kara Havacılık birlikleri teknolojinin son ürünü olan teçhizatlarla donatılmış helikopterlere sahiptir. Bu yeni nesil helikopterler performans, hız ve güvenilirlik açısından daha avantajlı olmakla birlikte, bakım yükü, pahalı parçalar ve derin sistem bilgisi gibi dezavantajlara da sahiptirler. Bu noktada, gelişim ve sistem entegrasyonu ile son model teçhizatın bakım yükleri arasında bir tercih söz konusudur. Helikopter sistemleri arasındaki etkileşim artmıştır. Bütün bunlar kullanıcı pilotlar için uçuş kolaylığı sağlarken, bakımcı personelin işini güçleştirmektedir.

Gelişmiş teçhizatın yeni bakım gereklilikleri ve etkileşim halindeki sistemler helikopter filolarının muharebeye hazır uçabilir durumda tutulmasını zorlaştırmaktadır. Daha da ötesi, bu durum birliklerde uzman personele duyulan ihtiyacı arttırmaktadır. Gelişmiş sistemlerle donatılmış helikopterlerin envantere girmesiyle beklenmeyen arızalar ve bunların teşhis zamanları artmaktadır.

Edinilmiş tecrübe arızaların giderilmesinde helikopter teknik dokümanları kadar önemlidir. Bu çalışmada, AS-532 COUGAR helikopterleri bakım personeli için kural tabanlı muhakeme karar destek sistemi oluşturulmuştur. Sıra dışı arızalar, muhtemel sebepleri, kullanıcıya uyarılar ve tavsiye edilen çözüm önerileri "durum" yapısında sunulmuştur. Kullanıcılar bu "durum" yapılarına arızayla ilgili olan sembol-değer çiftlerini takip ederek ulaşabilir. Bu karar destek sistemi sayesinde, COUGAR uzmanlarının tecrübelerini uzman olmayan tecrübesiz personelle paylaşmaları mümkün olacaktır. Bu sayede personelin arızalara doğru teşhis yüzdesi yükselecek ve arızaların giderilme süresi kısalacaktır. Aynı zamanda, bu sistem uzmanlık eğitim alanında da kullanılabilecektir.

Anahtar Kelimeler: Bilgi Yönetimi, Kural Tabanlı Muhakeme, Karar Destek Sistemi.

ACKNOWLEDGEMENT

I would like to thank to my thesis advisor Assist. Prof. Yavuz GÜNALAY for his guidance comments and help during this thesis study.

I also would like to thank to Mr. Kalyan Moy GUPTA from ITT Industries, AES Division, Alexandria. He has always welcomed my applications for the basic logic of expert systems in this thesis.

My friend M. Fatih SÖZBİR has continuously helped me in forming the rules and in building the logic of the program. I also would like to thank to him for his great contributions.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

1.INTRODUCTION

Specialists have always had great interest to solving the problems in less time and with more precision. Most of the new procedures, models, processors, programs and products are introduced to the market with claims of being more accurate and faster than the preceding ones. Those new programs have made the necessities easier to implement and the products have made the duties easier to perform. But with the improvements in technology, it has been more difficult to perform the maintenance necessities of those equipment. It has been a rule to maintain the equipment properly in order to be able to use them efficiently.

These technological improvements have resulted in more capable products, as well as more complex and costly maintenance procedures. Complex equipments often require sophisticated maintenance and handling procedures. While the maintenance procedures of these equipment, personnel encounter problems. Some of the basic problems can be listed as:

1. The system failure may be related with more than one subsystem. Solutions to those problems necessitate well-designed fault isolation procedures.

2. Beyond these procedures, problem solutions necessitate the knowledge of an expert (Report of a Working Party Council for Science and Society, 1989:44-59). The need for that expertise will especially heighten at unusual problems, which are difficult to handle, costly to overcome, and hard to solve. Within a big company or organization, there might be employed site experts for those problems. But it is almost impossible to use the domain experts on every event on time even it is planned carefully. And also it might not be possible for small size companies to employ an expert to be able to handle those unexpected failures. The accuracy of the solution depends on the performance of the expert and that performance may vary from time to time as well as on the speed of the solution.

3. Another problem is the cost of acquiring those experiences. Duplicative wrong decisions will always cause increments in the cost as well as increments in the "out of order" time periods of the equipments. Almost all of the wrong decisions and their applications to the problem on hand have caused different costs to the companies, associations and different types of organizations. Nobody wants to repeat the failures and bear their repetitive costs. So, some of the expertise, especially the technical ones, are extremely valuable for people who want to benefit from them.

4. Acquisition phase of the expertise also necessitates a certain period of time. It is not easy and not short to acquire an expertise that will be enough to handle with a problem about the matter. The ability of benefiting from a cumulated

expertise will surely save a certain amount of time to the problem facers. In other case, everybody will have to experience the same "wrong decision and application" period to handle with the problem correctly and acquire the same expertise. Such duplications in cost and time consuming will increase the overall incurred costs of the system malfunctions and will cause some problems to stay as "hidden" in the systems and will increase the possibility of reoccurrence of the same problem in the future. It is not a desired situation for anybody.

5. Another problem is the experts themselves. It is not easy and sometimes may not even possible to employ an expert for some of the companies. Furthermore it may not be feasible to do so sometimes. If a company or organization is sure that they will infrequently need the help of an expert, then they will prefer to hire those experts rather than employing. They will only need the help of the expert(s) at the time of trouble. However, there will be an uncertainty about time precision of hiring. Experts should be available at the time of react.

To overcome those problems, there was always a great interest to create a machine or a program that can think like a man and correct the faulty of people. Since it is almost impossible to use those experts in every event on time, the main focus should be on their knowledge. Specialists have thought of creating a computer program that would be able to carry the knowledge of the domain experts and would interact with the users whenever it is needed. After the inventions of some kind of package programs, which are used in computers, specialists tried to form a program that can make inferences from some foggy clues and can conclude about a problem (FORD, 1991:8-9).

So, the computers are used to get a result for that idea. Computer programmers created different types of programs and tried to solve problems using them. Most of the existing conventional computer programs were designed to solve the problems in a predetermined way. Programs could process with their preset structures and people were able to foresee the results of the processes; if the same processes were undertaken by the people, then the same results could be taken. There was only one true result and it could not be manipulated. And those programs were in need of precise and complete information to solve the problem on hand truly. Still, most of the programs on computers are based on the same principles; certain results. The most important advantage of this style is its timesaving for complex problems, which are very time consuming to solve manually.

Thereafter, the importance of ability at managing the knowledge has gained importance. Nigel Ford claims that:

> In an increasing number of organizations, information management is evolving to embrace the activities of creating, acquiring, sharing and maximizing the impact of resources, which enable the storage and retrieval not only of information, but also of *expertise* (FORD, 1991:4).

And he calls this concept as **knowledge management.**

By the time, to be able to manage the knowledge, intelligent systems were produced. As the basic subgroup of the intelligent systems, **expert systems** have been built for different purposes and in many different forms (FORD, 1991:8-10).

> "A more accurate description that summarizes actual usage would be that expert systems are *software systems for specialized applications that require expertise,* generally based *Artificial Intelligence (AI)* techniques". (MOHAN, 2000:1)

In this thesis, helicopter domain will be handled by rule-based reasoning technique. A great portion of helicopter maintenance demands can be considered as daily, urgent and hard to guess beforehand, like in the situation of Gölcük earthquake. These demands are important and should be met properly by the suppliers. So, the helicopter fleets should be kept ready for service and maintenance, and AOG (Aircraft On Ground) times should be lessened. A decision support system can be used to speed up some fault isolation procedures and to decrease the total costs of maintenance due to wastes. By the help of decision support systems for maintenance personnel, the time for maintenance procedures and fault isolation will decrease and "ready to operate" percentages of Cougar helicopters fleets will increase.

Cougar helicopters have been in service since 1995. Since 1995, the personnel in the army have gained a certain amount of expertise. Since I am a test pilot of Cougar helicopters I thought that preparing a knowledge base in order to be used in an expert system would be helpful for the users. Army personnel have sufficient expertise to try to form such knowledge bases for different type of aircrafts. It is concluded that rule-based reasoning would be the most suitable style of expert systems to build such a study considering its advantages. The reasons for selecting rule-based reasoning and its features will be discussed in the following chapters. Remaining of the thesis is organized as follows. Chapter 2 for "Literature Review", Chapter 3 for "Brief Review of Need for a DSS in the Army", Chapter 4 for "Rule-Based Decision Support System", Chapter 5 for "Test Results" and the last chapter, Chapter 6, is reserved for concluding remarks and future research directions.

CHAPTER 2

2. LITERATURE REVIEW

2.1.DECISION SUPPORT SYSTEMS:

Decision Support Systems are popular and have a wide application area in the world. They are the computerized systems that process the data and recommend possible actions for solutions by using analytical methods (IGNIZIO, 1991: 25-27). Managers are provided with the possible actions and they are free to choose or not to choose the recommended solutions. Since the program operates under some reasonable constraints, solutions may be optimal or not, depending on the degree of the suppositions.

2.2. ARTIFICIAL INTELLIGENCE:

A normal computer is able to solve the problems mathematically with its preloaded logic programs. Solutions are performed by calculations of numbers rather than by thinking on verbal expressions. An artificial intelligent computer is known basically as a computer that is able to solve the problems verbally. The main goal of

"Artificial Intelligence" is to make the computers able to reason about the problems similar to human reasoning (DURKIN, 1994:3).

With the help of artificially intelligent computers, it is possible to solve the verbal questions. A man gives the input problem to the computer in the form of verbs and the computer finds the answers to these inputs (RIESBECK, 1989:1-5).

2.3. KNOWLEDGE-BASED SYSTEMS:

Knowledge management is based on the knowledge–based systems, in which the problems are handled and solved like a human (LARI, 2003). A knowledge-based system basically includes three components. These are the basic programming languages that include convenient environment for usual data processing, knowledge itself with its own type of structure and the application techniques to be able to apply the knowledge to the problem on hand. Application part is known as "Inference Engine". Inference engine and the knowledge itself are separated from each other to make it easier to maintain and to provide robustness. By the time, the knowledge may change but nobody wants to change the application method.

There are two basic questions about the knowledge. The first question is about how it deals with the representation of the knowledge. Moreover, how that knowledge can be stored in the computer. Second question is about the expert systems. The structures of the expert systems are designed to store various types of information (FORSYTH, 1988:59-61). And the Inference engine can process the knowledge and gives us the initiative for action to solve the problem.

Expert systems are the commonly known examples of knowledge-based systems. They are much broader in term less sophisticated in structure. Also they deal with the matter in a much broader scope (DOHERTY, LEIGH, 1986:288).

2.4. EXPERT SYSTEMS:

2.4.1. Description:

The intelligent systems have a wide application area. There are many examples of them in application area. Almost in all of the application areas the use of expert systems brought decreases in cost and increment in profit, competitiveness and flexibility for the companies. As major areas they are used to make the existing expertise available for others, make some progresses at the existing level of expertise, free the existing experts for more serious problems, collect and form a database for further uses after the retirements of the expertise and for training purposes (FORSYTH, 1991:13-19).

As a subgroup of Intelligent Systems, expert systems are expected to solve the problems that require significant expertise. The computers are modeled according to human expert's abilities at problem solving (DURKIN, 1994:7). These systems are able to deal with uncertain problems and may lead uncertain or multiple conclusions. The results are far from foreseeing since the program may end up with some surprising results. It is different from the conventional programs in terms of its unpredictable solutions.

> " 'Expert systems' are examples of a particular class of computer programs which generally use heuristics to perform tasks previously restricted to *human* expertise" (FORD, 1991:8).

Expert systems are built on the knowledge of the domain experts. Their expertise is transferred to rules or other forms of the programs and then the real problems are tried to be solved by these programs whenever they are needed in the future. That expertise may be for any kind of topic. All of the domain issues in the world may be the bases to build an expert system upon them and the system interacts with the user and asks some questions and using the answers to those questions it responds the user with recommendations. They can be considered as simulations of human decision behaviors.

2.5. BASIC KNOWLEDGE REPRESENTATIONS IN EXPERT SYSTEMS:

Basic forms are **Rule-Based Reasoning Systems, Semantic Networks, Frames, Logic, Neural Networks** and **Case-Based Reasoning Systems**. All of them have some advantages and disadvantages over the others and are built on the human expertise. Since the systems are built upon human's expertise, it is prone to errors. For that reason, accuracy in knowledge acquisition into the systems by knowledge engineers is the most important matter of concern in the development of the expert systems. This is valid for all of the forms. It should be clear, understandable and true, and use common terminology with the other potential end-users. It is not a requirement for the knowledge engineers to know the domain and its specialties but the engineers should be careful about obeying to clear specifications of experts about the domain. The information from the expertise can be collected by different ways such as *Structured Interviews, Unstructured Meetings, Nominal-Group Technique, Delphi Method, Blackboarding* and *Case studies* (MOHAN, 2000: 10-11, 208-212). The selected method depends on the needs of the system builder, availability of the expert, time and the domain structure.

2.5.1. Rule-Based Reasoning Systems:

Rule-based reasoning is the most popular expert system. In this technique, the expertise of the domain expert is transferred into the program in the form of "IF-THEN" rules. All of the domain knowledge is studied as whole complete information and all of the possibilities are though during the domain transferring period. The engineer builds the system upon the information of domain expert. Since there might be some missed points in the system domain, maintenance is valid for the engineer. If a need for correction arises, then the relevant additional node is added. Rule-based reasoning systems are most widely used expert systems in the world (FORD, 1991:26). Conclusions are reached by trying the system nodes and answering them. Here is a typical figure for rule-based system (FORD, 1991 :25):

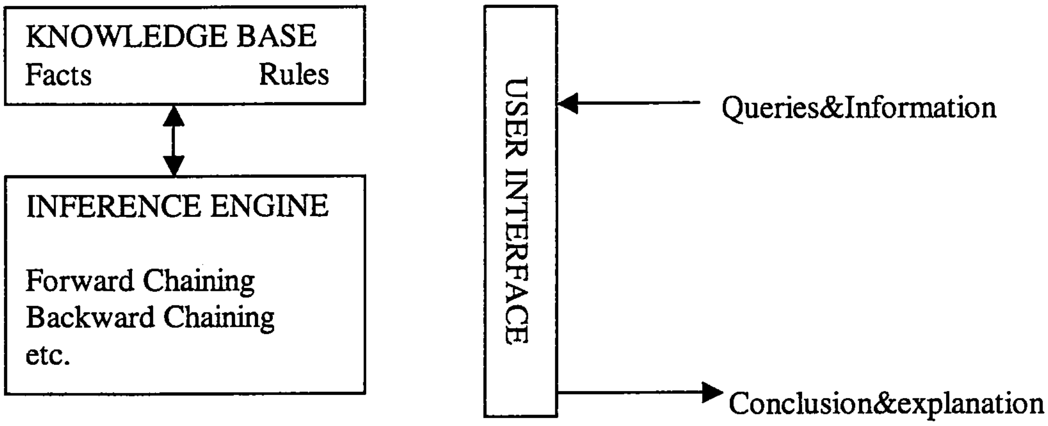


Figure 2.1. ***Architecture of a Rule-Based System.***

Here, domain knowledge forms the knowledge base and inference engine instructs how to use the domain knowledge. All of the facts about the domain are expressed as rules, which might be in backward chaining or forward chaining. Thereafter, the user interacts with the system and answers the questions for the values of the predetermined attributes. With simple exposition, the knowledge of the domain expert is formed as "IF-THEN" rules and controlled by inference engine (FORD, 1991:26).

There is always possibility for a change in the knowledge. These changes may originate from the changes in premises, changes in the features of the domain or changes in the thoughts of the expert. Changes in the knowledge base are dealt with as they occur. The relevant rules are changed or new rules are adapted to the system. In large domains it is much more important to build the rules very carefully. If not, the system maintenance will be difficult and complex since the domain is too large to examine, to find and to build the new rules appropriately for the relevant node.

In the rule-based systems, newly encountered problems are solved by the help of these prepared "IF-THEN" rules. The Artificial Intelligence community also calls IF-THEN rules as the production rules (IGNIZIO, 1991:48-51). If the domain is simple, it is better and easier to work with a rule-based system. The main form of a rule-based reasoning system is like (DURKIN, 1994:167):

ANTECEDENT		CONSEQUENT
Situation	....	Action

or

IF Situation	THEN Action

"IF 1. the infection is primary-bacteriaemia, and
2. the site of the culture is a sterile site, and
3. the suspected portal of entry of the organism is the gastro-intestinal tract,
THEN
There is suggestive evidence (0,7) that the identity of the organism is bacteroides."

(FORSYTH 1989:6)

Attributes and their values are the basic representation units for the knowledge. The attributes are formed according to the domain and the appropriate values are assigned to them. The values of the attributes are tested at the nodes and the system directs the user depending on the answer (IGNIZIO, 1991:76-79).

These attribute and value pairs have some properties in common. These are (IGNIZIO, 1991:76-79):

- Name (name of the attribute)
- Type (the class of values associated with the attribute, that is, symbolic or numeric)
- Prompt (the query presented to the user, when necessary)
- Legal values (the set of acceptable values for the attribute)
- Specified values (one or multiple)
- Confidence factors (i.e., as associated with the attribute value, or values)

NAME is the selected definition of the attribute to the system. These names are usually the primary determinants of the domain itself. For a helicopter domain, possible attribute names are bleed valve's position, servo control's position, alternators, engines and so on. TYPE is designated for the attribute in order to determine whether the attribute is numeric or symbolic. PROMPT is sometimes assigned to certain attributes in order to get the answer. The user can answer the query easily and that answer is processed by the system. LEGAL VALUES are the logical values for the attribute. All of the nonnegative real numbers will be the legal values for the weights of the human. SPECIFIED VALUES is the determinant of actual set of values to be tested in the system. CONFIDENCE FACTOR can be used if the package has that specialty and permits the user to deal with the uncertainties (IGNIZIO, 1991:77-81).

2.5.1.1. Advantages of Rule-Based Systems:

It is almost a usual way to reason about a problem in IF-THEN type rules for everybody. So, it is easy to understand and to form the rules. This specialty made the rule-based reasoning one of the most attractive expert systems to design the domains.

Knowledge and inference engines are separated from each other. This permits the engineers to perform changes on the domain or inference engine separately in order to fulfill the requirements of the changes in the domain or in the inference engine methodology.

The rules can also be expanded easily if the engineer keeps obeying the domain features and the rule interrelations. As the rule number increases in the system, it includes more facts about the domain knowledge and turns to be more useful in term of benefits it can supply (DURKIN, 1994:171-172).

Packages for development of a rule-based system are cheaper than the other types and have a widespread availability. Furthermore, knowledge representation is easy and it is easy to maintain the system if the system is built on by the engineer carefully. The most significant advantage of a rule-based system is its easiness for validation and test (IGNIZIO, 1991:74-76).

2.5.1.2. Disadvantages of Rule-Based Systems:

The most important disadvantage is the domain language, which is used during building the program. Users can describe the situation by different verbal descriptions. However, the Rule-based systems require “exact match” for firing the relevant node in the memory.

It will be possible to deal with all the possible solutions of the problem with "IF-THEN" rules. A user can find what he looks for, but as the domain becomes larger and wider then it will be difficult to prepare a perfect rule base for rule-based systems. Then it would be inefficient to handle with a wide real-world problem with complete set of rules, which covers all the possible actions. Moreover, by adding new rules to the system to capture all the possibilities one can only make the system slower and more complex.

Another disadvantage of a rule-based system is that it is hard to put the experiences of experts in perfect "IF-THEN" rules. These rules may work efficiently but cannot give the reasons for the conclusion. In order to give some clues about the reasons the engineer should form more rules, which means to form more steps to conclusion. The user can find the answer for "What to do" type problems but not for "Why" type questions.

Another problem with the rule-based system is the scattered structure of the domain. All of the rules capture the minimum-sized domain to be able to know more about the domain knowledge. However, it causes the domain to include hundreds of scattered pieces of rules. It may cause delays in the system during the operation and uncertainties about the rule locations in large domains because of the interdependency between the rules (DURKIN, 1994:173-174).

2.5.1.3. Inference Engine:

The knowledge base in the system can be supposed as the pool of information. All of the data about the domain are collected in that pool. And the inference engine's mission is to see this data and to make some inferences about the subject. Known information is stored to the knowledge base and new information is derived from this. The information on hand should be convenient to be used to infer or reason about a problem. Also it should be precise and in correct way. A basic form of inference can be illustrated as follow (FORSYTH, 1988:61-63):

Pit is a dog

Sweet is a cat

Tom is a bird

If all of those are true then it will be true to say:

Pit is a dog and Sweet is a cat and Tom is a bird.

This is an inference of the preceding three sentences.

There are two types of inference, which are forward chaining and backward chaining. In forward chaining, to find the consequences of a claim is the main idea and in backward chaining; the causes of a goal or a consequence are looked for (FORSYTH, 1988:71). However, most of the systems use the combination of both types.

2.5.1.3.1. Forward Chaining:

In forward chaining, inference engine processes the available knowledge starting from the beginning. It is also known as "Deductive Reasoning" and uses the known facts to infer about the unknown matter (DURKIN, 1994:91-101). The rules are examined from the starting point. Chaining goes from the premises to the conclusion (FORSYTH, 1988:63-67).

p = the sun is shining, q=it is hot

then,

If the sun is shining then it is hot.

This inference may be an example for forward chaining. The premise is used for the conclusion of "it must be hot".

2.5.1.3.2. Backward chaining:

Backward chaining works in an opposite way of forward chaining; in "Inductive Reasoning" way. The evidences for the hypothesis are looked for in this time. In consultation type reasoners, it is better to use backward chaining since the ultimate goal can be given at any stage of the process. Moreover, when there is abundant data for process, then it is advised to use backward chaining. The process starts with the desired goal and the evidences supporting this goal are looked for. The other propositions are not dealt with during the process. So, the ultimate goal of reasoning is always ready to be presented to the user.

q=it is hot, p=the sun is shining

then,

If it is hot then the sun is shining.

Evidence of "the sun is shining" for "it is hot" can be inferred.

Although these inference techniques are considered as separate from each other, in practice most of the programs use a combination of both (FORSYTH, 1988:14, 67-68).

2.5.1.4. Application Examples of Rule-Based Systems:

Rule-based reasoning systems are known as the most popular applications of the expert systems. Since the 1960s many of expert system applications have been seen in the world. These systems have widely been used for different purposes until today. Some of the famous application examples of rule-based reasoning can be ordered as:

2.5.1.4.1. DENDRAL was designed to analyze the Mars soil. Human expertise about the organic chemistry was transferred to the system in order to be used in Mars exploration program of NASA. This program is one of the pioneers of expert systems and it is proven that the real expertise can be captured by the computer programs efficiently.

2.5.1.4.2. MYCIN was developed to diagnose the blood diseases in 1970s. Actually, the basis of building a rule-based system was understood by MYCIN.

2.5.1.4.3. PROSPECTOR was formed to help geologists in exploration of the possible ore deposits.

Since 1980s, there have been many expert system shells in the markets and people have used the available program shells that are suitable for their goals. (DURKIN, 1994:163-197).

2.5.1.5. Rule-based System Planning:

2.5.1.5.1. Knowledge Base:

This part of rule-based systems contains the set of rules. Entire domain is expressed in the form of rules.

2.5.1.5.2. Working Memory:

This part of the program operates like human's short-term memory. It contains the facts of the situation and the inferring results.

2.5.1.5.3. Inference Engine:

Here, the facts of the problem are connected with the rules of the knowledge base in order to be able to infer new information. And this chaining goes on until no rule is to be fired.

2.5.1.5.4. User Interface:

All of the programs have a user interface to interact with the users.

2.5.1.5.5. Developer Interface:

The engineer builds the program on this face and the maintenance requirements are executed here.

2.5.1.5.6. Explanation Facility:

Here, the program has some information about the reasons of inferring. The answers for *Why* and *How* are given here to the users.

25.1.5.7. External Programs:

Some of the shells are equipped with this specialty. The program can call external programs to run.

2.5.2. Semantic Networks:

Semantic Networks are the first examples of the expert system applications. This is a representation of the knowledge in the form of nodes and arcs. The arcs stand for the relations between the objects and the nodes represent the objects. Problem is characterized with the nodes and the arcs, in which the properties and the relationships are described. The nodes are labeled with the names and the arcs define the relations between the nodes by taking some labels like *"IS-A"* and *"HAS"*. So, properties of the domain subject can be listed.

In Semantic Networks the specialties are ordered in a hierarchical level and it is possible to expand the features from specific to general, general to specific and at the same similarity level.

It works by asking questions to the nodes and getting the answer by the arcs. For instance, the answer to the question "What is lion?" will be "Wild animal" by the arc of "IS A". If there is an exceptional feature related with the object, then the exceptional specialty is dealt with adding an arc related with only that object (DURKIN, 1994:69-70). An example for the Semantic Networks may be shown as:

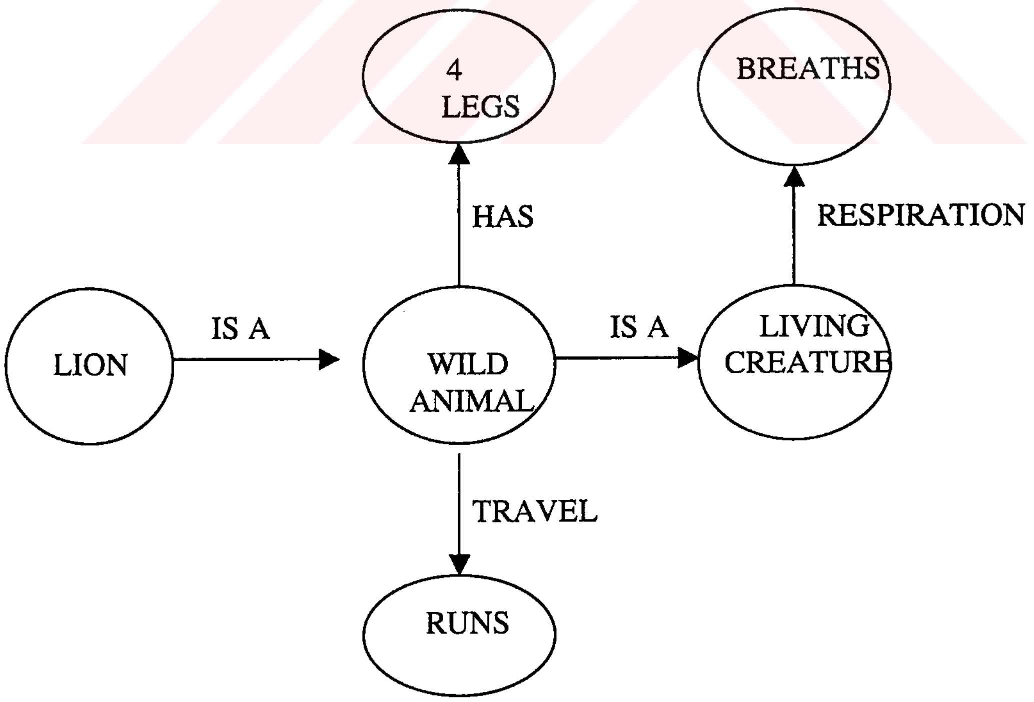


Figure 2.2. An Example of Semantic Network.

2.5.3. Frames:

Frames are a different way to present the knowledge. All of the features of an object are presented in a form related with that object. All of us are accustomed with this kind of representation in our daily lives. It is just like a sheet of student information. A name is determined for the frame and the specialties are shown with their own values.

By using the Frames, it is possible to deal with the objects at hierarchical levels. The frames can be ordered from general specialties to more specific specialties. The shared values can be shown at higher-level frames whereas the peculiar values at lower levels (DURKIN, 1994:73-80). It is possible to assign some links between the values of the specialties. So by filling the values in the higher-level frame the lower level frames can take the same shared specialty values (FORSYTH, 1989:146-149). An example for Frames is presented below:

Frame Name:	HORSE	
Class :	ANIMAL	
Properties :	Color	Brown
	Number of legs	four legs
	Hungry	Grass
	Travel	Runs, Walks
	Sound	Neigh

Figure 2.3. An Example of Frame Representation.

2.5.4. Logic:

In Logic representation, the knowledge is presented in the form of symbols. These symbols are used to infer logical reasoning. There are two types of representation; *Propositional Logic* and *Predicate Calculus*. In propositional Logic the symbols are used to present the “false” or “true” statements of the existing proposition. Propositions are evaluated according to AND, OR, NOT, IMPLIES and EQUIVALENCE logics, which we are used to from mathematics courses.

For example an AND evaluation will be like:

PROPOSITIONS		
A	**B**	**A AND B**
F	F	F
T	F	F
F	T	F
T	T	T

F:FALSE
T:TRUE

Figure 2.4. An Example for Logic Representation.

In Predicative Calculus, the objects are dealt with some predicates. These predicates describe the relation between the objects. If there are two objects (Military Expenditures, GDNP) and dependence relation between them, then the relation can be shown as:

Dependence (Military Expenditures, GDNP)

This kind of knowledge presentation also allows for reasoning like in forward chaining (DURKIN, 1994:80-87).

2.5.5. Neural Networks:

Neural Networks are designed as the operation principles of the human brain's neurons. System is designed in layers and these layers are classified as input layer, hidden layers and the output layer.

The user can manipulate the inputs and the solutions, but cannot understand the operation characteristics at the hidden layers. Input layer nodes only transmit the inputs and these inputs are taken as the input at the hidden layer nodes. After the manipulation and processes at the hidden nodes, these inputs are transformed to output and transmitted to the other connected nodes. At the last step, the output nodes take the activation signals from the hidden layer nodes and perform the input to present the solutions.

All of the nodes are connected to each other by connection rules and some weights are given to the connections. Those weights determine the accumulated activation signal at the relevant node, which decides whether to activate or kill the signal according to determined threshold of the node.

During the learning step of the organization, the weights of the connections are adjusted to be able to take the desired solution according to relevant input (FORSYTH, 1991:103-111). After these adjustments, the program is ready to run. A usual operation diagram of a neural network can be drawn as in Figure 2.5. (FORSYTH, 1991:103-111)

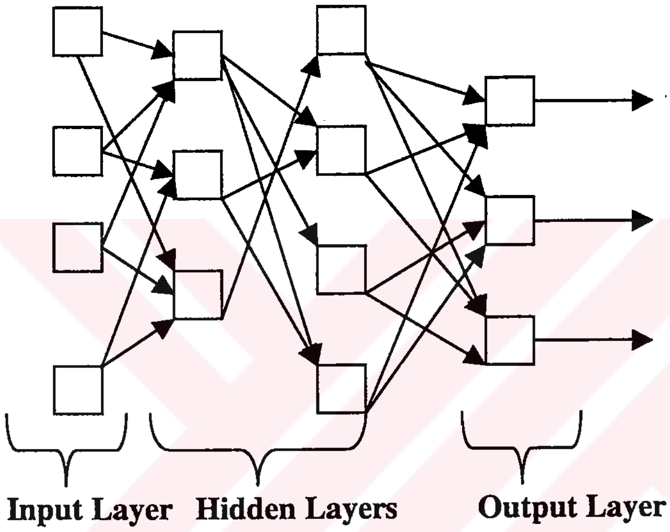


Figure 2.5. Neural Network Operation Logic.

2.5.6. Case-Based Reasoning:

> "The intuition of case-based reasoning is that situations recur with regularity"(KOLODNER, 1993:8)

Case-based reasoning was developed as an important alternative to rule-based reasoning expert systems (MOHAN, 2000:177). In the real world it is usually difficult to generate all necessary rules for a large domain. By the enlargement of the domain knowledge, it becomes more difficult to build all necessary rules to include the domain completely. In this aspect, Case-Based Reasoning (CBR) is an alternative way in the world of expert systems. CBR is appropriate for large domain knowledge with a narrow scope (MOHAN, 2000:178).

In case-based reasoning, it is the main idea that one can solve a new problem with the help of previous experiences of others or him. These experiences are transferred to a case library beforehand. Whenever a problem occurs, then the most relevant case(s) is/are retrieved from the library and used to solve the problem. The solution of the old case may or may not necessitate adaptation phase for the current situation. The former solution of the case may be used completely or it may be applied after some kind of modification for the new problem.

The knowledge of the experts is transferred to the programs in the form of cases, which serve to everybody when it is needed. All of the extraordinary experiences are thought to be cases and these cases are restored in a case library, which were prepared beforehand as the experience occurred in any time. Reposition of the cases in the case library can be thought as the knowledge of the domain expert. They are the knowledge of the system. The system can think about the problem with the help of the cases (GUPTA, 1994).

But there is a question at this point. Which of the events can we call as experience? The answer to this question is somehow subjective. In any way, the solution to the current problem should be different from the expected solution and the new solution should be unpredictable at the next times if the new experience is not recorded as a different and unique case to the library (KOLODNER, 1993:8-14). In these cases, one can find the relevant previous events and their solutions. These cases are selected from the library through the use of some different methods that are based on similarity parameters. Finding relevant cases involves; characterizing the input problem, by assigning the appropriate features to it, retrieving the cases from memory with these features and picking the case or cases that match the input best (RIESBECK, 1989:25). And the solutions of past case can be applied to the problem on hand completely or after some modifications.

> "We have a gut feeling about the situation as a whole. We can cite the arguments on both sides of the issue and then make a choice that seems best at the time" (RIESBECK, 1989:10).

Effectiveness of the system depends on both the preparation of the cases and the retrieval style (GUPTA Kalyan Moy, 2002). Retrieval has always received the ultimate interest from the CBR community. All CBR systems have a retrieval component, and success depends on the efficient retrieval of the right case at the right time (BRAMER *et al.*, 2000:90). Since it is mandatory to be able to get the correct cases just before the possible further modification, 'retrieve' phase should be very effective.

That necessity is also valid for the considerations of the case-based reasoning. These are:

- "A Case
- A similarity Index, and
- A case retrieval mechanism." (GUPTA, 1994).

Cases should be stored in the knowledge in accordance with the specialties and the usage necessities of the domain knowledge. During the storage the cases should be labeled with appropriate features in order to be able to call them during the retrieval process. All of these factors will increase the efficiency of reasoning.

Case-based reasoning (CBR) is an active learning system. As the system is used in time, the domain is captured much. And by the usage of the cases and experiencing new problems, case number gets larger and larger in the library. By the enlargement of case library, user gets an advantage also; does not need to start from the beginning. As the cases are used successfully and new cases are derived from them, the user can start from these new ones to conclude correct solution.

A basic form of case-based reasoning process can be shown as follow (KOHNO: 1997):

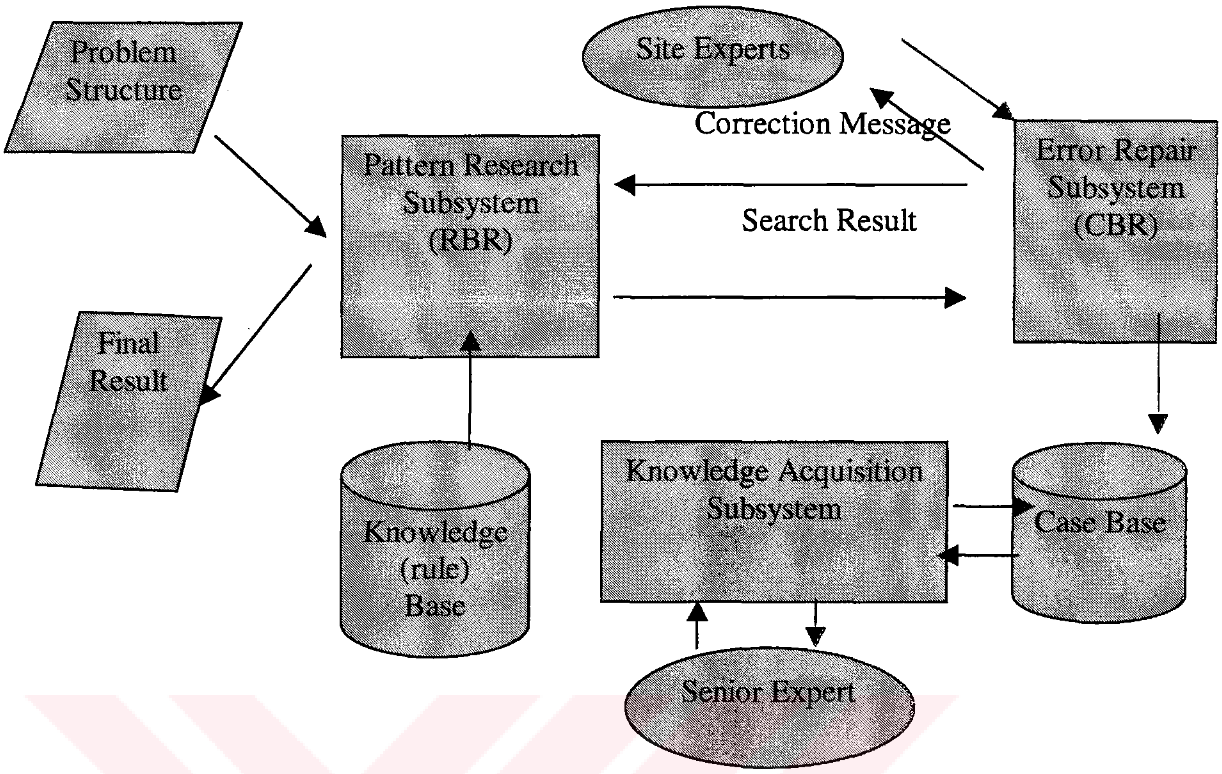


Figure 2.6. ***System Architecture and Functions.***

2.5.6.1. What is "Case"?:

> "Set of features, attributes, and relations of a given situation and its associated outcomes" (GUPTA, 1994)

Cases are past situations in which the domain experts have experienced about the problem(s) on hand. Problem definition, attributes and values (descriptors), solution to the problem and the outcome are the essential parts of a case. All of these specialties form the case (http://www.infj.ulst.ac.uk/~cbdq23/teaching /com812j/notes/cbr_ notes.doc). Attributes have a name and a group of values. At every problem, an attribute can take only one value and for every value of the specified problem values, there is a corresponding attribute. However, in some situations more than one value of corresponding attribute may be assigned to the case (REINARTZ *et al.*, 2001). An example for a case base can be as follow (REINARTZ *et al.*, 2001):

Table 2.1. ***An Example of Case Base.***

An Example Case Base						
Pi					Si	qi
C1	V12	V23	V31		S1	q1
C2	V13	V23			S2	q2
C3	V13	V23			S3	q3
C4	V13	V23	V32	V41	S4	q4
C5	V13	V23	V32		S5	q5
C6	V12		V31	V45	S6	q6

Denotation q is for additional information or data for the cases. Some of the indicators may be inserted here, as the indicator for the time the case is stored into the case base. These values under the problem (*Pi*) form the problem itself and there is a solution for each of the problems. For example, Case1 has the values V12, V23 and V31 and solution S1. Also additional information is available in q1 node for Case1.

2.5.6.2. Case-Based Reasoning Steps:

As standard application, CBR is formed from four steps; *retrieve, reuse, revise* and *retain*. However, some scientists suggested adding two more steps to this cycle due to maintenance needs. So, case-based reasoning consists of six steps, which are *retrieving, reuse, revise, retain, review* and *restore.*

A case-based reasoning cycle can be figured as below (REINARTZ *et al.*, 2001):

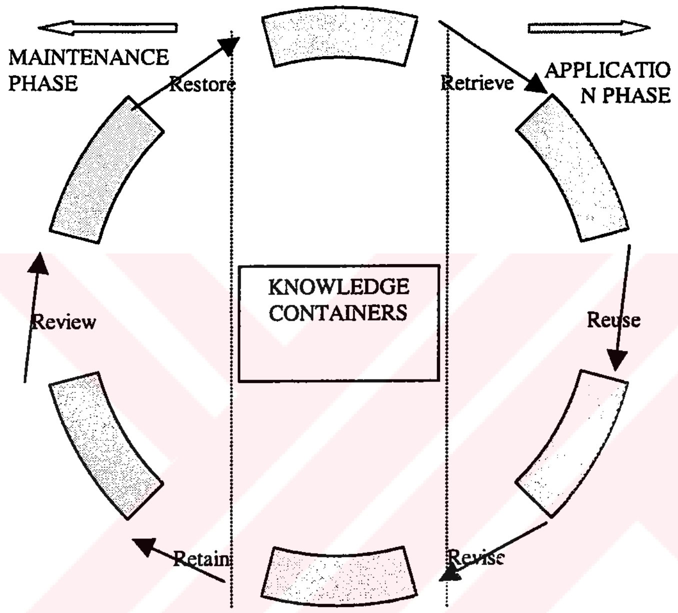


Figure 2.7. ***Six-Step CBR Cycle.***

In normal usage, only the first steps are used. Other three steps are about the maintenance cycle of CBR cycle. However, the maintenance is not applied for a well-working case base but also for efficient indexing, for enhancing similarity principles and for well applicable adaptation methods (CRAW *et al.*, 2001). There are many different types of proposals for maintenance procedures in CBR like the proposal of SHIU *et al.*, dividing the domain to small parts and then executing the maintenance, which is based on similarities of the cases in order to find redundant ones (SHIU *et al.*, 2001).

2.5.6.2.1. Retrieving:

In retrieving stage the user calls the similar cases from the library according to the features of the input problem. By the inputs of the user to the system, an algorithm looks for the best similar case or cases. These cases should also be useful. It means that the cases should be retrieved from the library in order to help the user to perform the ultimate goal, solving the problem on hand. Importance of the cases may differ from problem to problem. So, introduction of the new problem to the system is important. At the first step, cases are selected based on their potential usefulness. The features of the problem on hand and the dimensions of the past case at satisfying its solution are compared. Matching algorithms use these dimensions at determining the similarities. Indexes of the cases are used as a guide at determining the similarity since these dimensions are also on case indexes. In any time, matching should be efficient in order to benefit from the system by retrieving the best cases for the problem. Thousands of cases might exist in the case library as well as a few. Since it is not an expected situation that one of the cases in the library matches the existing situation exactly, search for an appropriate case will result with a partially matching case or cases with a big probability. All of the algorithms for searching the case libraries serve for different type of structures (KOLODNER, 1993:284-291).

Retrieval process is one of the most important steps of CBR. Further steps like reuse and revise of the results will take place after the retrieval phase (MANTARAS, PLAZA, 1997). Since the retrieval process is so important, the cases are stored in the library after an "Index-selection" procedure. Insertion algorithms do the same thing like the matching algorithms. They both determine the place for the cases. The difference is the direction of searching. During indexing, the cases are appropriately placed into the library for further usage and in selection or matching the appropriate cases are called from the case library. Both of them execute the same procedure. As logic, a user should be able to find similar cases with the new case in the library according to its place (KOLODNER, 1993:286).

Efficiency of retrieving depends on the tolerated database (library) searching complexity. However, the indexes of the cases determine the accuracy of matching process. In every case-based reasoning system and in their algorithms, the structure is partitioned in order to select the least and most accurate case set.

2.5.6.2.2. Reuse:

In this stage, the problem solution of the retrieved case(s) is/are executed to the problem on hand. Since the cases are retrieved according to specialties of the problem on hand, then the solution can be applied to our problem. This application may be either directly or after an adaptation. Possible solutions from the retrieved cases might be combined with each other during the application. Also other available data sources might be used for combining (ARSLAN, RICCI, 2004). Most of the time, old cases do not match with the new problem exactly. At these times some adaptations should be carried out on the old case solution (CRAW *et al.*, 2001).

2.5.6.2.3. Revise:

In Revising step, the solutions of the retrieved cases are adapted to the new situation in order to be able to solve it. Since the retrieved case's solution might not solve our problem, system should be able to manipulate its existing solution and adapt it to the current problem. There are two types of adaptation methods; Derivational Adaptation and Structural Adaptation (MOHAN,2000:183-185).

2.5.6.2.4. Retain:

If the solution is acceptable, then the situation can be stored as a new case to the library after the application of the revised solution to the problem in a "problem-solution" pair format with its own descriptors (attribute-value pairs). Also the system builder might choose to add additional cases to the library in order to cover the domain better (CRAW *et al.*, 2001).

2.5.6.2.5. Review:

This step includes the activities about monitoring the case-based system completely. Current situation of the knowledge is always under consideration and its quality is within the scope. For this purpose, appropriate quality measures should be determined. Furthermore, the desired quality level is acquired with the implementations of review step's recommendations (LEAKE *et al.*, 2001).

2.5.6.2.6. Restore:

In this step, available mechanisms in the system are initiated to recover the existing cases and the system itself due to unmet desired performance. CBR system's content is to be changed.

During the "Restore" operation, many different ways may be used to increase the quality level at the case base:

- a case might be *added* to the library,
- a case might be *removed* from the library,
- two different cases might be *combined*,
- A case might be *adjusted* again (by changing one of the values to another value for an attribute in a single case)
- A case might be *specialized* (By adding a new value to the problem component for an attribute, which was non-existing for that time)
- A case might be *generalized* (This action is taken by removing one or more values from the problem component of the case)
- One of the values for an attribute might be removed and another value might be added but for another attribute. This operation is called as "*Altering*"

- By crossing two cases their values are reduced to shared values. For crossing operation, the cases should be coherent. Also one of the cases must not be minimal
- Cases might be *joined.* Combining their problem component values might join two cases. Here, the values for the same attribute should not be different after the joining operation
- By *abstraction.* If two cases have the same values except one and these different values for the cases have the same unique pioneer (predecessor), then these values might be changed by abstraction and the cases are combined (REINARTZ *et al.*, 2001).

The required reaction by the system engineer to the system in different situations can be listed as (REINARTZ *et al.*, 2001):

Table 2.2. ***Relations Between Review and Restore.***

Relations Between Review and Restore for CBM	
Review	Restore
Not correct	Remove, adjust, alter
Not consistent	Remove, specialize, generalize, adjust, alter
Not unique	Remove
Not minimal	Remove, specialize, generalize, adjust, alter, cross, join
Not incoherent	Remove, specialize, generalize, adjust, alter, combine, abstract, cross, join

According to results of the "Review" step, corresponding actions might be taken for the related case in the "Restore" phase.

2.5.6.3. Indexing Cases:

Indexes are the tags of the cases. They designate the accuracy of retrieval decision of the case-based system algorithms. So, the cases should be indexed appropriately for effective usage of the library and to increase the efficiency of the system. Finding the relevant cases involves; characterizing the input problem, by assigning the appropriate features to it, retrieving the cases from memory with these features and picking the case or cases that match the input best (RIESBECK, 1989:25). Finally, those matching cases are presented to the user.

2.5.6.4. Examples of CBR Applications:

CBR has been successfully applied to a wide variety of domains, including systems for engineering structural design, learning environments for liver diseases, meteorology, cost and sales predictions and architectural design. Here are some examples of applied models:

2.5.6.4.1. IPP (Integrated Partial Parser) reads texts about terrorist activities, e.g., bombing, kidnapping, and hijacking, stores its interpretation in memory, makes generalizations, and uses these generalizations to guide future story understanding (KOLODNER, 1993: 27).

2.5.6.4.2. CYRUS is another program that focuses on how memory is used to answer questions after understanding. It derivates some other questions from the main question and tries to find an answer until a memory is retrieved with an answer (KOLODNER, 1993:121-126).

2.5.6.4.3. The **JUDGE** program works in the domain of criminal sentencing. The input is a description of the case, including the charge, the events that occurred, and the legal statutes regarding crimes of this nature, e.g., the range of imprisonment allowed and parole conditions. Then, it models a judge who is determining sentences of people who is convicted of crime. In its case library it contains the previous crimes and the sentences determined for each. And JUDGE uses the library to maintain consistent sentencing patterns (KOLODNER, 1993:102).

2.5.6.4.4. CHEF program works in the cooking domain. It generates new recipes by adapting old recipes. And this is a design domain, where an object has to be constructed to satisfy several goals simultaneously (KOLODNER, 1993:170-173).

2.5.6.4.5. The **COACH** program works in football domain to generate new football plays by improving old plays. It is also a design domain. It has significant difference from the others. It has only a few cases in the library. It creates new ones by modifying plays to form new ones.

2.5.6.4.6. The **MEDIATOR** tries to find a solution to a problem between two parties as it can be guessed from its name. It finds a new proposal to the problem on hand (KOLODNER, 1993:166-168).

2.5.6.4.7. HYPO works in the area of patent law. HYPO uses its base of precedent cases to generate plausible arguments for the prosecution or the defense (KOLODNER, 1993:179).

2.5.6.4.8. JULIA is a case-based designer working in meal domain. It has hundreds of recipes and cases in its library. There are many different situations for different customer meal habits. An interaction with the system takes place and the desires of the customers are entered as inputs to the system (KOLODNER, 1993:43-48).

CHAPTER 3

3. BRIEF REVIEW OF NEED FOR A DSS IN THE ARMY

3.1. General:

There is a tremendous progress in the technology today. It can be argued that the aviation is the first application area for these brand new technologies. Technological improvement increases the safety and makes it cheaper and faster to fly. But, it also makes it harder to keep it flying since the systems on the aircrafts are getting more complex and interactive by them. This reality necessitates more precise maintenance procedures and on-time technical supports of the producers to the customers.

There was no autopilot in 1950s but today there is almost no aircraft in the skies without an autopilot. Pioneers had been equipped with simple piston engines, which were able to produce about today's ordinary car's power. Today's aircrafts are equipped with some powerful turbine engines that can produce thousands of horsepower (HP). There were only hand fire extinguishers in the first planes. But today's airplanes have electrically supported complex systems using fire and smoke sensors located in compartments in which there is a constant risk of fire, like engine compartment or main gear box (in helicopters). Those sensors are produced as bimetal and operate according to the measured temperature automatically. As a conclusion, the systems are getting more electronic and complex. Moreover, it became a prerequisite to have highly-educated and well-trained officers and technicians to keep your fleet flying. Also, supporting the staff with up-to-date informative documents is crucial for success.

The need for vertical landing and taking off ability, which was especially aroused just before and during World War II, forced the specialists to work on an aircraft, which can move in vertical axis. And after some detailed and hard studies the first real helicopter was ready to take off. The inventor was Igor SIKORSKY, a Russian man (Inventors. The helicopters, 2004). Thereafter, technology of helicopter improved tremendously and many countries paid attention to this new technology.

3.2. Purpose of the Thesis:

In this thesis, we try to produce a decision support system for AS-532 COUGAR helicopters' maintenance personnel in fault analysis. AS-532 COUGAR utility helicopters are produced by Eurocopter and used by many countries throughout the world for different purposes including search and rescue, combat, ambulance and inner or exterior load and passenger carriage. It is equipped with two MAKILA-1A1 turbo-shaft engines.

A basic form of a decision support program will be formed in this thesis. Such a program will help the maintenance personnel in their efforts to localize and to overcome the failures. It will lessen their reflex time to unexpected failures, time for correct decision and surely the cost of maintenance. There are enough experts to use their expert knowledge to form such a system on all kinds of helicopters in Turkey. That knowledge should be collected and programmed to serve everybody for further use.

3.3. Current Failure Analysis:

The pilots report the failures by writing down the failure on the logbook of the helicopter after the daily flights. These failures are transmitted to the Maintenance Officer and the chief technician. They try to find a suitable solution by examining the technical documents and using their experience. The experience is the most important factor in lessening the reflex time and in mining the documents. If the failure is out of the authorization of the maintenance level, then the helicopter is sent to the upper level by filling a report form. If the failure is eliminated successfully, the test pilots test the helicopter in test flight or ground run and give the helicopter again to service if there is not a problem. If the failure goes on, then the steps for localization and removal of the failure are repeated.

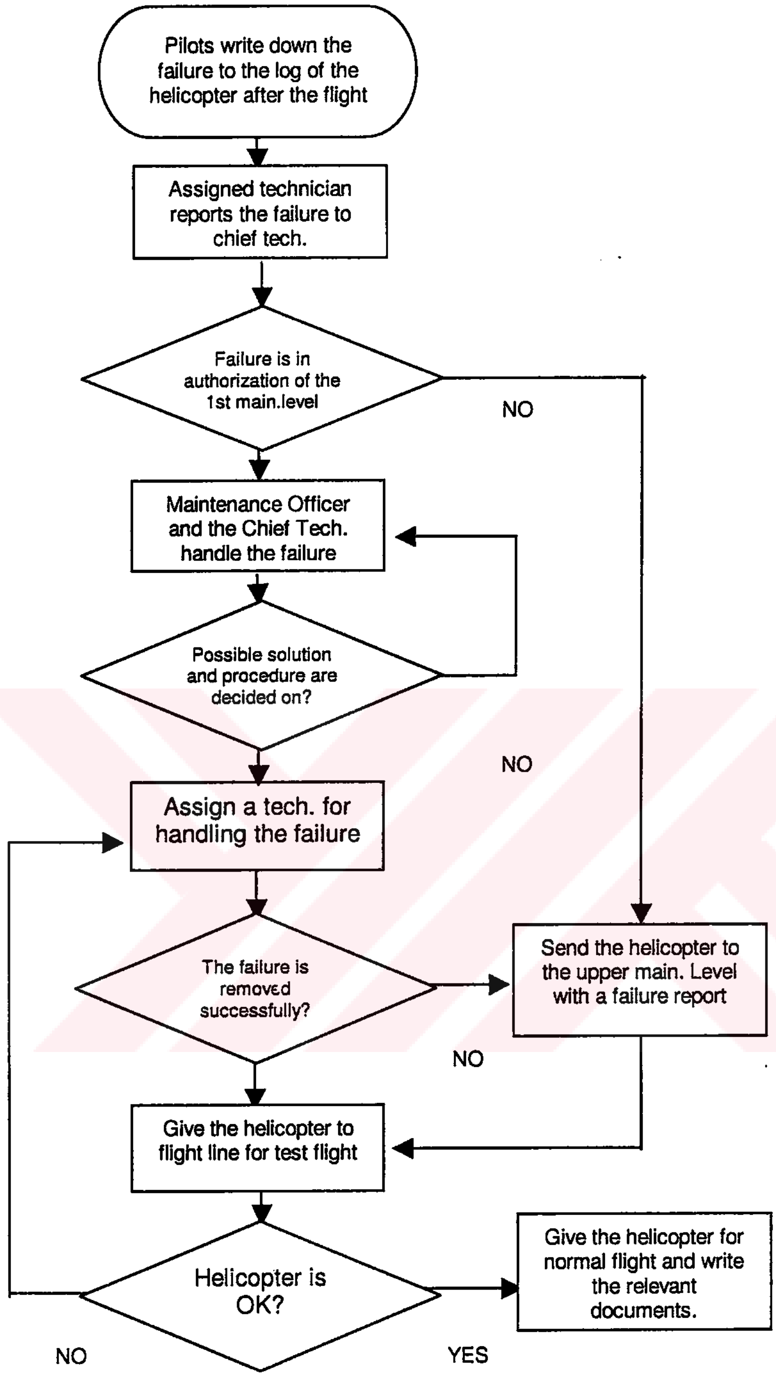
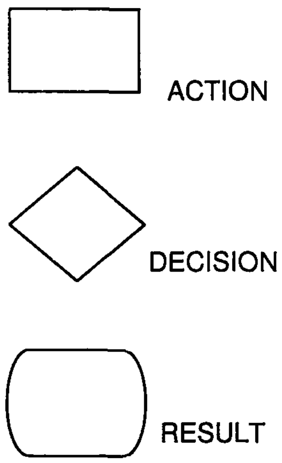


Figure 3.1. Current Failure Analysis Flowchart.

Table 3.1. Meanings of the Flowchart Figures. (RUSSELL, TAYLOR, 2003:135).



3.4. Main Reasons in Forming a DSS:

3.4.1. General:

As production logic, all types of helicopters have to make hover flight (0 ft/sn movement in any direction in the air). For that reason the power that twin MAKILA-1A1 engines in Cougar generate, should be transferred with a 90° change in direction via main gearbox (MGB) through the mast and blades to create lifting power. During the vertical lifting, which the main rotor blades perform, the main frame is exposed to a great torque force. The direction of the torque forces the main body (fuselage) to turn into opposite direction. So, there is a need for a force to eliminate the effects of torque: anti-torque. It is accomplished by a tail rotor system, which creates an anti-torque force to eliminate the torque that is created by the main-rotor. It keeps the main body in any direction under the control of the pilots. The pilots in the cockpit control the amount of the torque and the anti-torque forces by movements of the flight controls.

To transmit the generated power and the anti-torque force, and to control them, some other parts are used such as shafts on the helicopters as different from the airplanes. These parts increase the maintenance load of the personnel and make the system more complicated. All of the extra parts increase the fault possibility and need for maintenance. For that reason in manufacturing a helicopter, it is designed and used more of turning parts than a plane, which needs only forward thrust. So in helicopters it is more likely to confront a failure before, after or during a flight.

3.4.2. Factors:

To decrease the possibility and to make the helicopter more reliable, the producers limit the usage of the sub assemblies and parts by the ways of fewer cycles, fewer flight hours and less useful lives. Moreover, they try to give some controls of the systems to electronically managed control boxes to decrease the load for the pilots. But still it is more difficult to localize and to solve a fault on helicopters than airplanes due to interactions between these systems.

To monitor all the systems, the manufacturer has to add more parts and make the systems more interactive. For example, an autopilot failure might be due to a failure of autopilot system itself or due to a failure in electric system or due to a failure in hydraulic system or due to a failure in “hot air” (P2) system. The users can find the ways to solve the problems in fault analysis manuals. But some fault analyses, which require deep knowledge of the system and take much time, are experienced at that time by someone who solved it and it is prone to forget by time.

Furthermore, some of the failures can be solved by “*try and error*” method, which might cost high and take long for the organization. In line with Pareto principle, most of the time for concluding a failure with a solution and removing it are wasted for extraordinary, seldom occurring failures.

Also, the producers of the subassemblies or parts pay great attention to safety factors. They cannot put all the parts into service after the production phase. Some of the parts are found defective or inappropriate for operation. Moreover, the criteria to decide whether the part is operative or not is so strict and stipulates low tolerance limits. The costs of those parts are also transferred to the users. As a result, the parts, which are used in aviation, are too expensive to waste in “*try and error*” steps of fault diagnosis.

Another factor that creates a need for a general system to operate is posting of the personnel in Turkish Armed Forces. Some specialized personnel in any branch or type of helicopters can be posted to a different position, which might be unrelated with his expertise. On yearly base, these new assignments to different cities are performed. So, some of the acquired experience can be forgotten by time. It decreases the efficiency and increases the costs. Some duplication for the same experience and the cost is inevitable. It can be eliminated by a centrally controlled and operated system, which contains the experiences and shares it by the users.

Every year, new technicians graduate from their preparation school. They go to their assignment centers with their textual information without sufficient application on technologically developed helicopters. Moreover, they are assigned to helicopters after a short period of orientation time. They execute all the maintenance activities and get in fault analysis. Such a decision support system can be useful in decreasing their wrong diagnosis and wrong solutions while increasing the true ones.

3.5. Why Rule-Based Reasoning:

Rule-based reasoning systems can be used to support the decisions of the managers by "question-answer" pairs, in which the past experience is encoded (DURKIN, 1994:168-169). A decision tree format in the program can be very useful for the users and easy to build [(The Basics of Expert (Knowledge Based) Systems]. Acquired experiences in aviation might be good to construct a decision support system. A helicopter has full of systems, which are interacting with each other and many of operating subassemblies. Rule-based reasoning system is thought to be more suitable for creating an expert system since the advantages are charming.

First of all, the expertise can be transferred into the system as cases in which the technicians and other maintenance personnel can find the related solutions in written form instead of sole rules to apply. So, the personnel will understand the reasons for the recommended applications. Since 1995 the personnel gained a certain amount of expertise. That expertise should be collected together and shared by every user in the maintenance system.

Moreover, the cycles for time and hourly-based maintenance, changing all time-limited parts or facing with all kind of troubles are not completed yet. It means that the maintenance personnel have not experienced all of the possible fault analysis yet, which were described in the manuals. Collecting all of the experience in a program will surely help maintenance personnel while they execute their professionals.

Some kind of inferences from the system will hopefully make the technicians able to make some related comments on the subject and deal with their "first time" problems more efficiently. Most of the unnecessary part wastes or incorrect applications are done during the technician comes across with the failure for the first time. Since the system cannot assign the experts to every type of failures at every time, maintenance controllers have to count on the applicability of the maintenance documents and the understanding capacity of the inexperienced technicians. Furthermore, in order to make the technicians experienced at their profession, they should be assigned with some important applications and practices. At these times the need for experience is felt deeply. A system based on rule-based reasoning can help them during their ultimately flight safety operations or during the operations on highly cost parts.

By the time the system is used, the domain will be captured more and more. It is almost impossible for a study to be able to solve all of the potential problems at first glance. The system can maintain and update itself by the helps of the users. If it cannot find a solution to existing problem, then the system tries to collect the possible solution as a new rule to be added to the system. So, the system and its domain knowledge grow continuously.

As the final benefit, total cost of the maintenance activities and fault analysis and the time period in which the helicopter cannot fly will eventually decrease.

Existing Cougar helicopter maintenance manuals are also in the form of rule-based reasoning. Most of the procedures are in "YES/NO" tree format and according to situation on hand; the user is directed to different nodes and directives. The maintenance personnel of the Army execute the existing procedures and since 1995 they gather some great experiences. They are also accustomed with rule logic.

CHAPTER 4

4. RULE-BASED DECISION SUPPORT SYSTEM

4.1. General:

In the existing maintenance manuals there are failure analysis procedures that are applied by the technicians and supervised by the maintenance officers in the logic of rule-based. A technician is directed to the solution in the chapters by the relevant system questions and an application or change is recommended at the end. If the recommended solution solves the problem, then the analysis is ended; if not, the technician and the maintenance officer search for new indicators of the problem. Usually a new solution is decided on, in which deep system knowledge is expressed and great efforts are made.

Basically, there are two possible failure types. The failures that cause the helicopter stay on ground are called as "AOG (Aircraft on ground)" since those failures directly affect flight safety, and the failures that cause partially loss of some systems' capabilities are called as secondary failures since they do not have primary concern for flight safety.

If the failure is called as AOG, then it is strictly prohibited to give the helicopter to flight line before applying required fault analysis. This process has to end up with a successful result. Since the failure has primary concern for flight's safety, it is called as urgent. If the failure persists after applying the relevant procedures, then a second attempt of search for failure is started. This process goes on until reaching a certain and correct solution that eliminates the failure. Thereafter, the applications and the failure removing actions are recorded to relevant helicopter documents and logbook.

Second type failures are not urgent and necessary parts are ordered from the depots by official documents if it requires new parts. All of these actions are recorded to the logbook again and the helicopter can be given to flight line although the failure is not removed. Whenever the ordered parts are handed in, replacement or installation is executed and after testing the relevant system only, the records to the logbook's relevant row are noted.

All of the results for two types of failures must be affirmative at the end and the helicopters should be given to the flight line as ready to fly. Decided applications of the procedures and their results are so important in terms of flight safety and maintenance costs. Wrong decisions might end up with unnecessary high costs and unnecessary wasted parts, efforts and time. Moreover, the helicopter waits on ground as out of order. During all these activities, three possible results occur. These are "true diagnosis", "wrong diagnosis" and "wrong solution". Personnel may end up with one of these. In any case, the solution application will remove the failure. Time and procedure used make the difference.

- True Diagnosis: If a failure is ceased by taking correct steps and selecting the right choice among alternative solutions, then the diagnosis is called as true diagnosis. There is usually more than one alternative solution for a failure. All of them require different amount of effort and time. Furthermore, they might require different part removal-installation processes. The chosen solution should require minimum cost, and should dismiss the failure. It means that there is no wasted time, wasted efforts and wasted part. For example, an NG difference between the engines can be removed by adjusting the thermal compensator or by adjusting P and S screws or by adjusting drop compensator, or solely changing the FCU (Fuel control unit) with a new one. If the third alternative, to adjust drop compensator, is decided by analyzing the system manuals and system indicators in detail and applied accordingly, this will be "True diagnosis". Since this alternative eliminates the problem successfully and requires minimum effort, time and expenditure, it is the best alternative.
- Wrong Diagnosis: If a failure cannot be removed by the solution of our diagnosis and further investigation is required, then this is as wrong diagnosis. Possibility for wasted time and efforts will be certain. Also wasted parts will depend on the situation. The most important disadvantage of a wrong diagnosis is that true operating part(s) may be damaged during the unnecessary removal-installation process.

Furthermore, it is possible to defect a brand new part if it is ordered from the upper depots as an application requirement.

- Wrong Solution: It is possible to remove a failure by choosing to replace a main unit instead of making adjustments on it or on its sub assemblies. It is possible to remove a failure of FCU (Fuel control unit) by changing the unit completely with a new one. If the failure is about the sub components of FCU, then the failure will be removed successfully. This solution will be wrong solution since the failure will be removed but with unnecessary cost and wasted part. Wasting time is also possible.

Since all of the choices must end up with a successful solution, this is regarded as a primary constraint. It is the true and most advisable method to start with the least cost and easiest choice to eliminate a failure. It means that, achieving "True diagnosis" is the matter of concern. All of the others will surely cause unnecessary efforts and costs. Since all of the aviation parts are so expensive, these unnecessary costs might reach high amounts for the organization.

It is a reality that during all their maintenance and fault analysis procedures, some of the personnel gained so valuable expertise. In this thesis, those experiences are collected and presented in a program to the users. In order to achieve "True diagnosis", the acquired expertise is one of the most important factors. It will make the organization operating at minimum maintenance costs and wastes. This is the main goal of this thesis to create a "Decision Support System" for COUGAR helicopters' maintenance personnel for achieving "True diagnosis" at urgent and

AOG (aircraft on ground) failures. For that reason, this rule-based reasoning computer program is created and presented to help the staff and to eliminate unnecessary part removal and installations and to obtain minimum maintenance costs. Experience of helicopter COUGAR was selected as the domain subject for the computer program in this thesis. Knowledge base of the system is formed by acquiring the relevant experiences from the domain experts and of mine. All of the experiences were transferred to the system in the form of cases, which were connected with each other by the rules.

Table 4.1. Sub-systems of the Program.

```
1)  Auto Pilot
2)  Fuel System
3)  Hydraulic System
4)  Ice & Rain Protection
5)  Landing Gears
6)  Main Gearbox
7)  Pitot Static System
8)  Tail Rotor
9)  Thermal Load Check
10) Main Rotor
11) Engine - Stopped
12) Engine - Hot start & Indications
13) Engine - SLow Acceleration
14) Engine - Normal Acceleration
15) Engine - Running at Start Detent
16) Electric - Engines Running
17) Electric - Engines Stopped
18) ELectric - Navigation
19) Exit
```

The system is comprised of mainly 12 sub-systems that are the most interactive ones and in which most of the experiences are gained. These sub-systems and an "Exit" choice are presented in the first page of the program. "Electric" and "Engine" systems are divided to three and five sub parts respectively since they are the most important and wide systems in the helicopter. Electric has "electric with engines running", "electric with engines stopped" and "electric navigation" sub parts. Engine has "engine stopped", "engine hot start and indications", "engine with slow NG acceleration", "engine with normal acceleration", and "engine running at start detent" sub parts.

The rules were formed by "IF-THEN" rules and these rules direct the user to a solution. Since the inference was planned from the information to the final solution, it can be regarded as *forward chaining*. By answers to the questions at the inner nodes, subsequent rule is fired. This chaining goes on until reaching to final node. That node includes a solution, which might be a direct recommendation or a case that includes the experience of the expert that faced with the same problem before. The extraordinary "problem-solution" experiences are included in this study as cases, which are hard to solve or has some important and unique points to pay attention (GUPTA Kalyan Moy, 1999). The user reads the case and gets the crucial points about the problem on hand and totally free about the application of the recommended solutions since this is a decision support system and the application responsibility belongs to the user.

COUGAR Decision Support System was formed in Turbo Pascal program by "IF-THEN" rules. These rules interconnect the attributes. Since all of the attributes were used for describing a case or direct application, the user ends up with a solution. The user has to answer the attribute questions and select one of the values presented in the node. According to selected choice, user is directed to another node and the final node is a direct recommendation or a case, which includes a brief description, observations (as attribute-value pairs), problem, cause, solution and warnings to the user. If there is a discrepancy between the answers and the helicopter's system principles, then the user is warned by the phrases like "don't care", "it is not a problem", "That's normal" and "impossible" at the final nodes. So, it is a warning to the user to think over the failure and try the system again.

The system's code program was formed in Visual Basic. The user interacts with the Visual Basic page first. Then, according to the selected sub system, it operates Pascal subroutines in another page. User goes with selection of attributes and their values in this Pascal page. After reaching a solution that contains case, he interacts in Visual Basic page and selects the case tab. According to reached case number in Pascal page, the user selects the case number. After that, all of the information of selected case is presented in the page. This process can be cycled again easily.

4.2. Data Acquisition Method in the Thesis:

This program was basically built on the experiences of the Turkish Army aviators. Since I am one of the COUGAR test pilots in Turkey, I have not hesitated to contribute my own experience to this study. Helicopter technicians and the maintenance officers from the army aviation school are asked for information where

COUGAR helicopters have been in use since 1995. They have shared their own experiences and helped in this study.

There are many different ways of collecting data from the experts. *Structured Interviews, Unstructured Meetings, Nominal-Group Technique, Delphi Method, Blackboarding* and *Case studies* might be used for this purpose (MOHAN, 2000: 208-212). While collecting the data from the experts "unstructured meetings" technique is used to collect the information since the technical terms of the domain are known well and the appointments with the experts were irregular. Moreover, the possibility of misdirecting the interview phase due to lack of concentration was low.

4.3. Attribute-Value Pairs:

There are 143 attributes and their values to describe the cases. All of the attributes were prepared in line with the system requirements of helicopter. Observations are the primary descriptors for the solution and formed by those attribute-value pairs at the time of the failure. User will observe the indicators, warning lights and other components during the failure and then will use the decision support system by answering the questions according to those observations. The prepared rules will be fired by those answers according to inputs of the user. All of the attribute-value pairs go on until a final node is reached. A solution recommendation is presented directly or as a case at the final node.

An example for the attribute-value pairs is like:

Table 4.2. An Example for Attribute-Value Pairs.

ATTRIBUTE	VALUES	
Standby Artificial Horizon	Supplied	Not supplied

4.4. Cases:

In this study, the program includes 49 cases. Experiences of domain experts were transferred to the system in form of "case" and expressed by the attribute-value pairs. Only extraordinary failure-solution pairs are included into the system since they are difficult to solve by the help of the existing helicopter manuals. Furthermore, these failures occur infrequently and take long time for solution. The problems, which can be solved by following the manual instructions, are not included. It should contain an extra point and should necessitate using more than one manual. Very basic system necessities may be forgotten or not cared enough by time. For example, installation order of the tail rotor blades is a very basic requirement but can be neglected sometimes since there is much to do. If an imbalance problem is faced at tail rotor, order of installation does not come to mind as a reason. In one occasion, three days were wasted for correcting imbalance problem by implementing the relevant imbalance chapters. Since the chapters are prepared by considering that all of the prerequisites are completed beforehand, no information exists about the blade installation order. In three days, many of starts and many of efforts were wasted and the helicopter stayed as out of order. This situation deserves to be a case in this program.

All of the cases have "DESCRIPTION", "OBSERVATIONS" (as attribute-value pairs), "PROBLEM", "CAUSE", "SOLUTION" and "WARNING" sections. In "DESCRIPTION" section, information about failure related system is told about briefly for the user. In "OBSERVATIONS", observed attributes and their failure related values were written to describe the case. Conclusion of the case about the failure was stated in "PROBLEM" section. Possible causes for the failure were written in "CAUSE". There might be more than one cause for the problem on hand. For that reason all of the possible causes were listed starting from the most possible one to least. In SOLUTION, all of the possible solutions for the case were presented in an order. At the last section of the cases, important warnings to the users are told about in "WARNING" section. Cause and Warning sections contain the experts' experiences. A wrong order at blade installation can cause for an imbalance problem. This is the easiest and out of "manual warnings" cause. A closed bullet in air filters can cause out of limit observations at "T4" system during the thermal load checks since it causes the free air get into engine harder. "WARNING" includes expertise information also. The other program parts do not need for experience. One case for each main system is available in APPENDIX A.

Program also includes direct recommendations for users if the followed attribute answers do not end up with a case situation. There are 186 direct recommendations at the final nodes. "Check the circuit breakers", "Change the alternator" or "Check filter for dirt contamination" are some of the direct recommendations.

4.5. Interaction with User:

The user can start the program in WINDOWS by clicking the program file. The program file is consist of three tabs, which are for system selection, case selection and about pages.

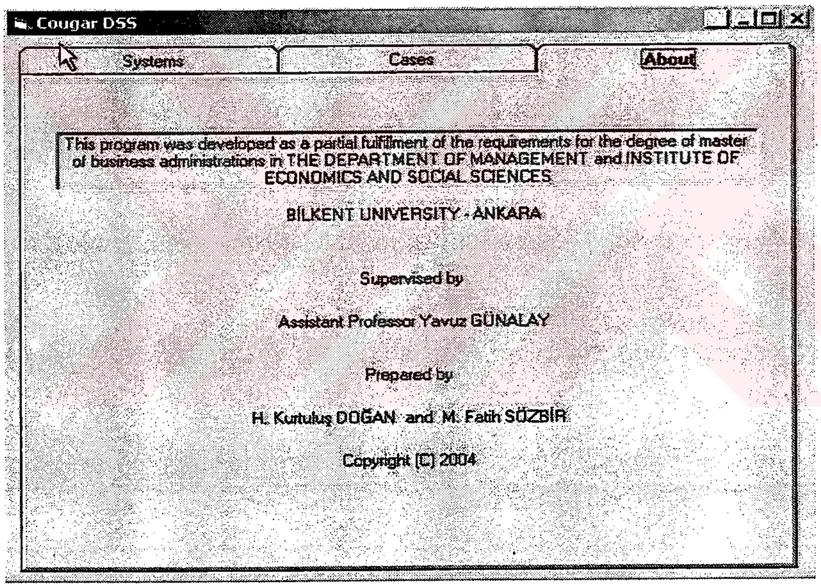


Figure 4.1. "About" Page of COUGAR DSS.

The systems page contains 18 choices and the user selects one of the systems in which he has problem. System can call other programs whenever it is needed such as case files (DURKIN, 1994:276-277). Then the program runs the "exe" file of the selected sub system's program, which was prepared in Turbo Pascal. This is a second page and the user can follow both of the pages at the same time. In this new page, user interacts with the sub system attribute-value pairs' questions. According to the answers to the questions as values, next questions are asked. "Thermal Load Check" section of the program is available in APPENDIX B.

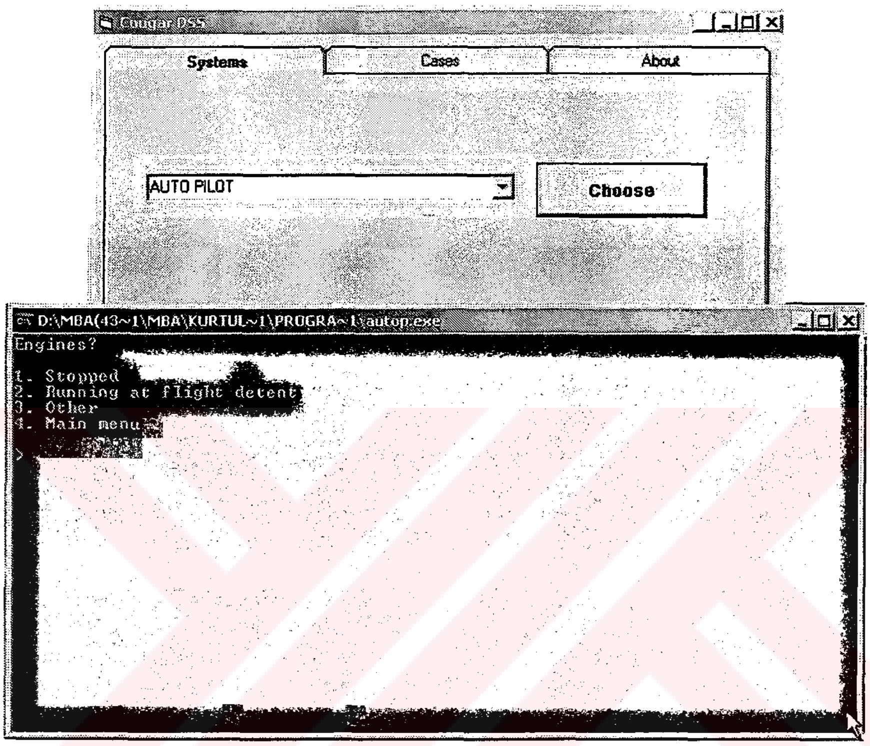


Figure 4.2. An Example for "inner nodes" in the Program.

During the process in sub system program, "back to previous question", "main menu" and "other" choices are presented in addition to the values of the question. By selecting "back to previous question" choice, the user goes to preceding node. This selection does not exist in the first nodes since the previous question would be the same with the "main menu" choice. If he selects "main menu", then the first system page comes to screen. Program asks to the user for describing his own case to system engineer if "Other" choice is selected. Then the required rules can be added to the subsystem program in Turbo Pascal and the main program is updated if it necessitates update.

4.6. DSS Maintenance:

COUGAR helicopters can be still regarded as new for the army. Therefore, it is still possible to encounter extraordinary failures and gain new experience. So the system must be capable of dealing with new experiences. For that reason, system was designed as dynamic. A routing in sub systems may end up with no solution for the problem. The user is asked for describing the situation to the system engineer for adding new case or direct recommendation in Turbo Pascal sub programs. In these times, the user must solve the problem by himself. Since the system accuracy is important and it is not possible to consider all of the solutions as cases, maintenance should be executed by authorized personnel. After the maintenance, system accuracy test should be done by the users for confirmation.

4.7. Steps in Developing Forward Chaining:

Decision support system was formed as *forward chaining* since the chaining goes from available information pieces to a complete solution. As a first step, the problems were defined by the attribute-value pairs and rules were formed. Input data are the value of the attribute. User's answer is one of those values. All of the rules were connected with each other and an answer for a node will fire the next rule.

System was tested by the builder and then by the users. Since the test users are real technicians, accuracy of the system was tested not only for correct running but also for correct recommendations. Forward chaining goes until reaching the final node correctly.

IF tail servo control's situation is **normal**

AND vibration level is **abnormal**

AND yaw pedals and feelings signs for **"there is vibration"**

THEN read **case 33**.

This example is from the knowledge base of the thesis and about tail rotor. "normal", "abnormal", and "there is vibration" are the values of the attributes of "tail servo control's situation", "vibration level" and "yaw pedals and feelings" respectively. System directs the user depending on the answers. Solution of "Read case 33" is one of the final nodes in the system.

Inference engine works in the way of forward chaining. Direct recommendations and the cases in the system are inferred by the values of the attributes. An inference example from the program can be given as:

The engines running at flight detent,

NR on triple indicator is constant at normal speeds, NF on triple indicator is more than NR+5

If all are true, then it will be true to infer:

Engine's shaft is broken.

CHAPTER 5

5. TEST RESULTS

5.1. General:

In this test procedure, we hope that our recommended program will decrease the required time in fault analysis of extraordinary failures. Moreover, we aim to decrease the percentage of "wrong diagnosis" and "wrong solution" applications. With the help of the cases and its sub sections like "causes", the system will be enlightened in the user's mind and the possible solutions will be listed in front of him. This program feature will help in decreasing "wrong diagnosis" percentages.

When a failure occurs on the helicopter systems during the flight or during the maintenance procedures, that failure is recorded by the pilots or by the technicians to the logbook of the relevant helicopter after the flight or during the maintenance procedure. Failure might be recorded by the technicians or by the fleet pilots or by the test pilots. Failure notes are written to the logbook of the helicopter and these failures are handled immediately after flight. Observations are again the primary determinants in writing down the failures in real world. Wrong or incomplete failure records can misdirect the technicians and the maintenance officers. This point is valid for both; the existing fault isolation procedures and the program.

However, the questions in the program may remind some other important points beyond the recorded observations.

5.2. Tests and Solutions:

Two groups were formed for this test. One technician was assigned for analyzing the problem using decision support system and the other group consists of two technicians who solved the presented case using in manuals as it is today. Manual group's technicians are asked different cases one for each. All of the related observations for the case are given to them. It means that 100% information was available about the case. Same observations were also given to the decision support system user technician for his case. Observations are given in verbal and were noted by the technicians. Since they have the same observations and almost the same expertise, the first ideas about the case would be equal in their minds. Then time measurement was started. Recorded times for the solutions may be longer in real life since all of the related observations are presented, which might not be available in real failure reporting.

We paid attention to some points in forming the groups. First, the technicians should be at the same expertise level. All of the technicians had at least 5 years of active experience on COUGAR helicopters. Also, they were known as skillful and capable at solving the problems. A difference in expertise could hurt reliability in the test results. Second, the foreign language levels should be almost the same for

English. All of the required maintenance manuals were kept near and ready to use by the test technicians. None of the technicians was permitted to interact with the other technicians during the test period.

All of the cases were asked to three COUGAR helicopter technicians for testing the system usefulness. The technicians were experienced enough on COUGAR and had a maintenance background on the domain. The program was run on computer and all of the manuals were available. Cases and their attributes were told the technicians in a sequence and their endeavor times were recorded until reaching the right solution for the case. Application time of the correct solution was not included in the total amount of time required for the cases. Following case and its attributes were told after the completion of the current case. During the test, usage of the computer is not permitted for the technicians with manual procedures. If they decide on a wrong part to remove and install in order to apply their decision, then the measurement was stopped and estimated removal-installation of the part was included to the measured time since in real environment, those wrong applications will take place. [1] Thereafter, possible outcomes of this application were told the technician and he was directed to the problem again by starting the measurement from where it was paused.

As mentioned in Chapter 4, those time periods are regarded as wrong diagnosis. Possible results of a wrong diagnosis application may also be different

[1] All of the required times for removal and installation of a part are estimated by the most experienced personnel who were not assigned during the test.

from only time wastes. A broken or damaged transmitter, switch, indicator or other parts may cost more than the wasted time since they are kept at limited numbers in the depots and all of them are so expensive.

If a case is reached at the final node, then the required time for reading and being knowledgeable about the case is added to the measurements for the system user technician. Since the chaining depends on the values of the case attributes, there was no need for deep knowledge about the helicopter's related main system.

Measured times for the cases were given in the below table in the test:

Table 5.1. Measured Test Times for Cases by DSS and Manual Procedure.

	WRONG DIAGNOSIS in minutes					WRONG DIAGNOSIS PENALTY TIME	WRONG SOLUTION (Times)	DSS Test Time
	1	2	3	4	RESULT			
CASE1	20	10	–	–	38	30	1	4,21
CASE2	10	–	–	–	No result	10	0	5,20
CASE3	20	–	–	–	55	20	0	3,44
CASE4	0	–	–	–	20	0	0	3,20
CASE6	25	25	–	–	No result	50	0	2,60
CASE7	20	–	–	–	48	20	0	2,68
CASE8	–	–	–	–	5	0	0	3,20
CASE9	–	–	–	–	12	0	0	2,55
CASE10	–	–	–	–	No result	0	0	3,18
CASE11	–	–	–	–	4,3	0	0	0,60
CASE12	–	–	–	–	5	0	0	2,58
CASE13	10	–	–	–	57	10	0	3,50
CASE14	–	–	–	–	11	0	0	2,49
CASE15	25	–	–	–	40	25	0	4,13
CASE16	10	–	–	–	20	10	0	5,17
CASE17	–	–	–	–	34	0	0	4,16
CASE18	–	–	–	–	5	0	1	1,42
CASE19	–	–	–	–	7	0	0	2,37
CASE20	–	–	–	–	21	0	0	3,43
CASE21	–	–	–	–	No result	0	0	3,25
CASE22	–	–	–	–	13	0	0	2,44
CASE23	–	–	–	–	12	0	0	3,56
CASE24	–	–	–	–	46	0	0	4,24
CASE25	–	–	–	–	23	0	0	4,12
CASE26	–	–	–	–	36	0	0	4,35
CASE27	–	–	–	–	No result	0	0	6,15
CASE28	15	15	–	–	57	30	0	4,52
CASE29	20	20	–	–	No result	40	1	5,20
CASE30	–	–	–	–	No result	0	0	12,26
CASE31	15	20	–	–	No result	35	0	6,35
CASE32	15	–	–	–	21	15	0	3,04
CASE33	15	15	15	10	No result	55	0	6,35
CASE34	–	–	–	–	8	0	0	5,20
CASE35	20	15	–	–	No result	35	0	2,70

Table 5.1 (cont'd)

CASE36	20	15	--	--	No result	35	0	2,70
CASE37	15	--	--	--	25	15	0	2,20
CASE38	30	--	--	--	40	30	0	5,45
CASE39	--	--	--	--	7	0	0	3,40
CASE40	20	15	--	--	45	35	0	1,50
CASE41	50	--	--	--	No result	50	0	4,00
CASE42	--	--	--	--	17	0	0	5,27
CASE43	50	--	--	--	No result	50	1	3,30
CASE44	--	--	--	--	No result	0	0	4,14
CASE45	10	--	--	--	23	10	0	5,16
CASE46	--	--	--	--	4	0	0	4,14
CASE47	10	15	--	--	No result	35	1	2,52
CASE48	--	--	--	--	18	0	0	4,45
CASE49	--	--	--	--	38	0	0	2,24
CASE50	20	--	--	--	58	20	0	2,12

It has been observed that 15 out of 49 cases could not be solved using manuals in 1 hour. All of the applications in 15 cases were incomplete and the mentioned causes were insufficient for an effective solution by the technicians. 34 of the cases could be solved manually and their solution times were well above the time of the same cases in the decision support system. If the time measurement for a correct solution went beyond 1 hour, then it was taken as "no solution"[2] although the cases must be solved in real life. It means that time observations for manual procedure are right truncated. This makes all of our calculations more conservative. Also, all of the needed parts in wrong diagnosis were regarded as available nearby.

Applying wrong diagnosis was frequent in manual procedure and all of the wasted times were added to the cases' solution times. For instance, checking and cleaning the jet pump in fuel transverse tanks is not enough for a complete and effective solution since a failure at the flap in the tank can cause to empty and open the fuel tanks again after refilling and checking the system by starting the engine.

[2] In manual procedure, 1 hour was designated for the technicians as an upper limit at concluding a correct solution.

However, those flaps can be checked at the first time when the tanks are emptied and opened if the point had been remembered by the technicians. These forgotten matters and unmentioned causes may lead duplicative time wastes and efforts.

There are 5 wrong solutions, which were described in Chapter 4 for these 49 cases. Also, penalty times (wrong diagnosis) for these 49 cases are significant and need analysis. We were in hope of decreasing these needed and wasted times at the beginning of our test. The test results confirmed our claim about the times. In these wrong diagnoses, it will be inevitable to face with wear outs on the parts. Furthermore, these parts may be completely damaged and may be out of order in unnecessary removal and installations sometimes. Since this kind of undesired results can be obtained only in real conditions, we could not include them in this thesis.

Although 5 wrong solutions for these 49 cases may seem few, their costs can be very high. Moreover, in real conditions these parts will surely necessitate great efforts. Furthermore, they may not be available in depots and the maintenance officers might have to keep the helicopter as out of order for long period of time. This is the worst situation that can be faced for maintenance personnel. That kind of application will increase the maintenance cost.

Table 5.2. Descriptive Statistics for Manual Procedure Solution Times.

MANUAL PROCEDURE	
Mean	36.1898
Standard Error	3.094606
Median	38
Mode	60
Standard Deviation	21.66224
Sample Variance	469.2526
Kurtosis	-1.63653
Skewness	-0.16959
Range	56
Minimum	4
Maximum	60
Sum	1773.3
Count	49

An average of 36.1898 minutes were needed for a solution in manual procedure. Also, 26 cases could be solved without a penalty time. It means that 47% of the cases were solved by wrong diagnosis. It is a high percentage when we consider wearing out possibility of the related parts in the cases. It is higher than the expectations. However, it is a main indicator for usefulness of the decision support system. 15 cases could not be solved by the manuals, corresponds to 30.6%. If the primary constraint is considered again, which was mentioned in Chapter 4 as all of the failures must end up with a successful failure removal, then it will true to claim that 30.6% of the cases could be solved exactly in more than 1 hour. As an experience, that time period may lead to a few days. Therefore true average time is much more than 36.19 minutes. Also, time measurements for manual procedure have a standard deviation of 21.66 minutes.

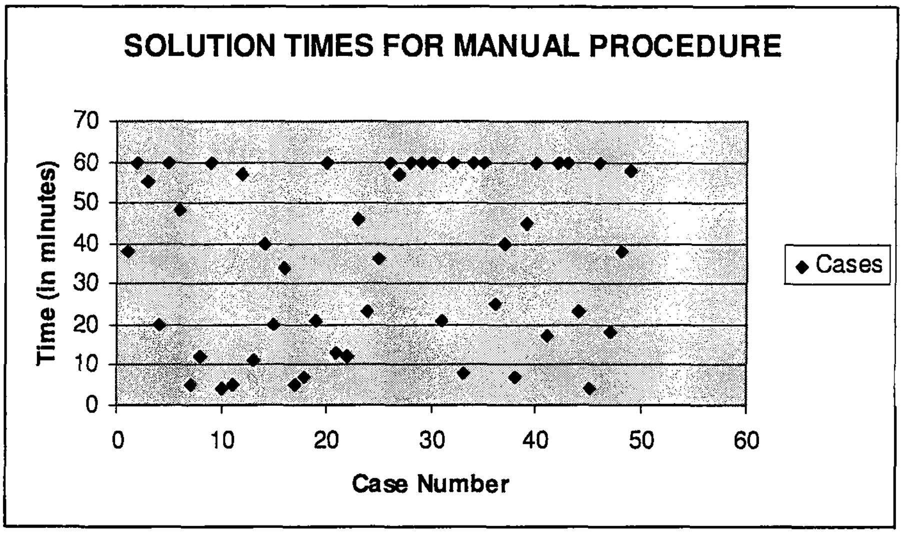


Figure 5.1. Solution Times for Manual Procedure.

It is observable that needed times in decision support system are well below of the ones measured in manual procedure. A mean of 3.804 minutes was needed to reach correct solution. Also standard deviation is 1.791 minutes. Beyond the measurements of the required time, it is more important observation that all of the cases have a solution. There is no case with any solution. Since all of the possible causes are presented to the user, wrong diagnosis penalty times diminished and possibility of duplicative efforts and wearing out the parts in unnecessary removal installations are eliminated. The user was acknowledgeable about the failure with the help of the case formation. It will also increase the efficiency and knowledge about a failure before starting the fault diagnosis in real conditions. Furthermore, the possibility for wrong solution and wrong diagnosis diminished. This will have effect at decreasing the maintenance costs by preventing wrong part changes and unnecessary removal-installations.

Table 5.3. Descriptive Statistics Values of DSS Solutions.

DSS	
Mean	3,804694
Standard Error	0,255915
Median	3,44
Mode	5,2
Standard Deviation	1,791406
Sample Variance	3,209134
Kurtosis	9,339444
Skewness	2,189883
Range	11,66
Minimum	0,6
Maximum	12,26
Sum	186,43
Count	49

It is also true that more experienced personnel can correct the failure faster than the inexperienced ones. The experience level can affect the solutions in manual procedure more sharply. This point is the main goal for the system; to decrease the needed time in failure diagnosis for inexperienced personnel.

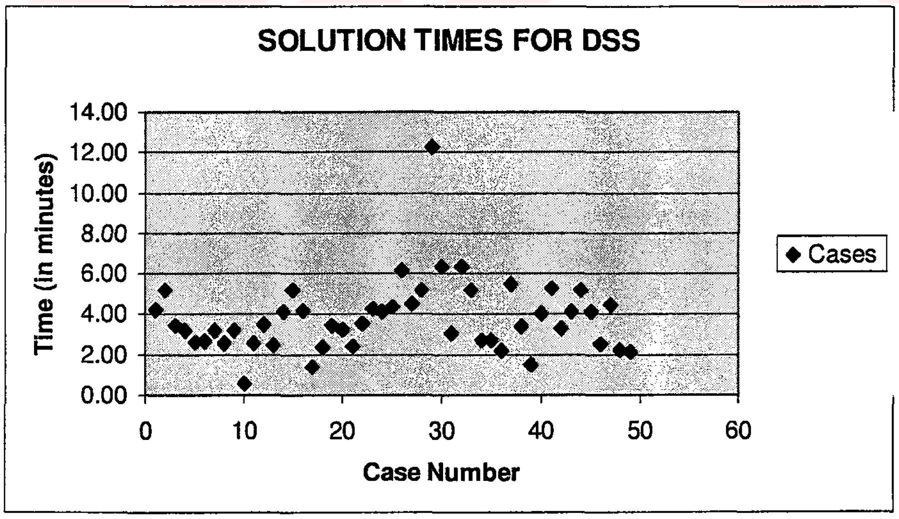


Figure 5.2. Solution Times for Decision Support System.

5.3. Hypothesis Test:

Also, it is possible to make a hypothesis test in order to see whether the system is useful or not. After the hypothesis test, the results should be examined carefully. Since the time measurements with the decision support system were recorded as a new application method in the maintenance system, these values are the second column (Observation 2) and the measurements of manual diagnosis are the values of first column (Observation 1). We assumed that relative frequency distribution of the case solution times' differences is distributed normally.

The steps in hypothesis test are presented below:

- Since the maximum time is designated as 1 hour, considering the "no result" solutions as 60 minutes will not decrease the reliability of the test. It will be more conservative approach.
- This test should be made for *Paired Difference Test of Hypothesis*.
- (Ho) was chosen as average solution time differences for manual procedure and decision support system are equal to 10, 15, 20, 25 and 30 minutes in turn in order.
- (Ha) was chosen as average solution time differences for manual procedure and decision support system are more than 10, 15, 20, 25 and 30 minutes in turn in order.
- Confidence levels were chosen as 0.01, 0.05, 0.005 and 0.001 in turn in order.

Considering "no result" solutions as 60 minutes will be a conservative decision for hypothesis test. It is claimed in Ho that the decision support system solutions' average is at least 10 minutes less than the average of solutions of manual diagnosis with a %95-confidence level.

Table 5.4. X_D Values of the Sample Groups.

MANUAL PROCEDURE	DSS	X_D
38.00	4.21	33.79
60.00	5.20	54.80
55.00	3.44	51.56
20.00	3.20	16.80
60.00	2.60	57.40
48.00	2.68	45.32
5.00	3.20	1.80
12.00	2.55	9.45
60.00	3.18	56.82
4.30	0.60	3.70
5.00	2.58	2.42
57.00	3.50	53.50
11.00	2.49	8.51
40.00	4.13	35.87
20.00	5.17	14.83
34.00	4.16	29.84
5.00	1.42	3.58
7.00	2.37	4.63
21.00	3.43	17.57
60.00	3.25	56.75
13.00	2.44	10.56
12.00	3.56	8.44
46.00	4.24	41.76
23.00	4.12	18.88
36.00	4.35	31.65
60.00	6.15	53.85
57.00	4.52	52.48
60.00	5.20	54.80
60.00	12.26	47.74
60.00	6.35	53.65
21.00	3.04	17.96
60.00	6.35	53.65
8.00	5.20	2.80
60.00	2.70	57.30
60.00	2.70	57.30
25.00	2.20	22.80
40.00	5.45	34.55
7.00	3.40	3.60
45.00	1.50	43.50
60.00	4.00	56.00
17.00	5.27	11.73
60.00	3.30	56.70
60.00	4.14	55.86
23.00	5.16	17.84
4.00	4.14	-0.14
60.00	2.52	57.48
18.00	4.45	13.55
38.00	2.24	35.76
58.00	2.12	55.88
	Average XD=	32.39

$\bar{X}_D$ value for group of case solutions was calculated as 32.39 minutes. Since the sample size is more than 30, we can use *Z-statistics*. The calculation principles for comparing paired groups in *Paired Difference Test of Hypothesis* are as below:

One-tailed Test:

$Ho = \mu_D = Do \qquad (Do = 10)$

$Ha = \mu_D > Do$ (or it may be calculated for $Ha = \mu_D < Do$)

Test Statistics :

Test Statistic: $z = \dfrac{(\bar{X}_D - Do)}{(\sigma_D / (n_D^{1/2})}$

Rejection Region will be: $z > z_\alpha$ (for $Ha = \mu_D > Do$).

σ_D will be equal to standard deviation of the differences. If we calculate the descriptive statistics for the group of differences, then we observe the standard deviation as 21.20747. Then we can start the calculations for *z-test.* Descriptive statistics table 5.6. is presented below:

Table 5.5. Descriptive Statistics for Time Differences Between Two Groups.

XD	
Mean	32.3851
Standard Error	3.029639
Median	34.55
Mode	54.8
Standard Deviation	21.20747
Sample Variance	449.7569
Kurtosis	-1.64761
Skewness	-0.15153
Range	57.62
Minimum	-0.14
Maximum	57.48
Sum	1586.87
Count	49

First, Ha = $\mu_D > 10$ was tested against Ho = $\mu_D = 10$ for $\alpha = 0.05$. After the test result, same Ha was tested for $\alpha = 0.01$, $\alpha = 0.005$ and $\alpha = 0.001$ in order to reveal highest possible confidence level for 10. After realizing that 99.9% confidence level is valid for 10, the same procedure was made for 15, 20, 25 and 30 in turn in order.

Ho = $\mu_D = 10$ (Do = 10)

Ha = $\mu_D > 10$ (or it may be calculated for Ha =$\mu_D < 10$)

$$z = \frac{(32.39) - (10)}{(21.20747) / (49^{1/2})} \Rightarrow \mathbf{z = 7.39},$$

related z value from the z-chart is **1.65** for $\alpha = 0.05$.

Rejection Region will be:

$z > z_\alpha$ (for Ha = $\mu_D >$ Do)

7.39 > 1.65 ➔ Reject.

REJECT Ho for Ha.

Since the calculated value is more than the value of *z-chart* (for α=0.05), we can reject the hypothesis of Ho (McCLAVE *et.al.*, 2001: 412-419). It means that the claim mentioned in Ho can be rejected for $\mu_D >$ Do and the claim is true for %95-confidence level. The decision support system decreased the needed amount of time for correct solution decision at least by 10 minutes. Then, the same tests for 15, 20, 25 and 30 minutes were made at the same confidence levels. Table below shows the results of these calculations:

Table 5.6. Asserted Time Savings with Different α-Values.

	Asserted Time Savings (in minutes)					
α	10	15	20	25	30	Confidence level
5%	+	+	+	+	-	95.0%
1%	+	+	+	+	-	99.0%
0.5%	+	+	+	-	-	99.5%
0.1%	+	+	+	-	-	99.9%

According to Table 5.6 we can say that the system decreased the needed time 25 minutes with a confidence level of 5% by these observed data sets. We cannot say it for other α-values in the same column. However, if we do the required calculations it is possible to say that system decreased the needed time 27.3 minutes with a confidence level of 95%. When we increase the confidence level, we see that system's expected timesavings decrease which is in line with the statistical principles. However, if we had continued to measure the time in manual procedure beyond 1 hour in cases, those results in Table 5.6 would be better for the decision support system.

Also there is a minus value among X_D values. It shows that one of the cases was solved in less time by manual procedure than by the system. That may be valid for easy cases since the technicians will not need to follow a certain route in order to reach a final node.

5.4. Correlation:

It is possible to think that a strong correlation should exist between two groups of data. If fault isolation takes long time for true solution in manual procedure, then it should take long again in decision support system. In order to understand the linear relation between the solution times, it possible to draw a scatter diagram and calculate correlation coefficient.

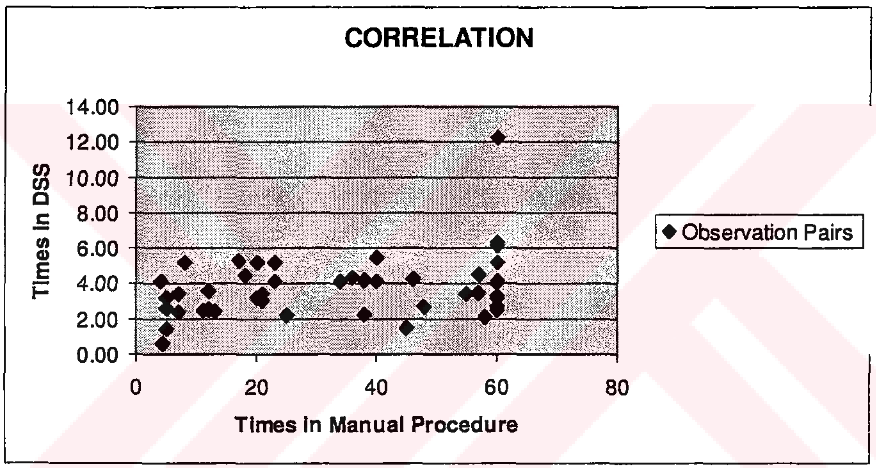


Figure 5.3. Scatter Diagram for Observation Pairs.

Table 5.7. Table for Correlation Coefficient.

	MANUAL	*DSS*
MANUAL	1	
DSS	0.292544	1
P-Value	0.041	

It is clear that there is not a strong linear relationship between the measured case times in manual procedure and in decision support system. So, we cannot say that a case with long solution time in decision support system will take long time in normal procedure. Also, P-value is significant. It is logical since the needed time in decision support system is almost constant regardless of the case difficulty level. However, difficulty of case in real life is the major determinant for the problem solution time.

CHAPTER 6

6. CONCLUSION AND FUTURE RESEARCH DIRECTIONS

6.1. General:

Almost all of the test users mentioned the usefulness of the program in different aspects. First of all, the program recommends solutions to extraordinary failures. That is a great advantage for the technicians by carrying the past experiences of the experts. “WARNING” to the users in the cases is one of the most valuable sections of the cases since there are important points and all of them are concentrated on the system realities of the helicopter.

The statistical studies of the results and the hypothesis tests showed that we are true at our claim that the system will increase the percentage of true decisions and decrease the needed amount of time and so decrease the unnecessary maintenance costs and time durations for out of order periods for helicopters. One of the most attractive solutions is elimination of 5 wrong solutions. Those wrong solutions were about expensive and big main parts like main gearbox, fuel control unit and engine. For example, engine is the most expensive part on the helicopter.

Hypothesis test is also supporting the claim in this thesis study. Test shows that such a decision support system can be beneficial for the users in real working conditions. The DSS improves the decision system by at least 27.3 minutes at significance level of 95%. This might seem small improvement. But two things have not been forgotten:

i. The system was tested against experienced technicians not new graduates.

ii. Longer failure detection times were truncated at 1 hour (which could be couple of days in some cases).

6.2. Advantages of "COUGAR DECISION SUPPORT SYSTEM":

The system has many advantages in use. It is possible to list the advantages as:

1. All of the experiences are expressed in the cases in order to be transferred to the user. These expertises are so valuable in maintenance.
2. Program gives the user some important clues about the system interactions and possible causes of the problem. Since this kind of system information is not available in existing helicopter manuals, it is going to increase the knowledge of the personnel and decrease the reflex time for starting true diagnosis.
3. Required time for a problem solution has decreased sharply. Mean value for the correct solution has decreased to 3.804694 minutes.

4. System is available for all of the technicians. So, the main point of the expert systems is accomplished. The experience is shared with the others especially with the inexperienced technicians.
5. Accuracy at reaching a solution is same for all of the personnel. This test shows even for experienced technicians there exists a 10.2% (5/49) chance of reaching a wrong solution.
6. The user is able to remember and check some crucial points while following the attributes to the solution. These attributes will show that relevant step's question should be checked before performing any fault analysis activity. Technicians will remember some important points.
7. System can be handled and maintained by the time. It is possible to add, change or delete a case or a node.
8. In "CAUSES" sections of the cases, possible causes were ordered. Although such information is available in the manuals, causes section in the program is combining long manual chapters in brief explanations.
9. It can combine the scattered experiences of the experts in the army.
10. The WARNING section is regarded as one of the most beneficial points in the cases since application related points are ordered here for the users.
11. System will be larger in terms of domain knowledge by the time passes and become more useful. Eventually, all of the experiences will be collected and a joint database will be created. Almost all of the extraordinary failures can be removed successfully at minimum efforts and costs.
12. System can also be used for training purposes.

6.3. Disadvantages of "COUGAR DECISION SUPPORT SYSTEM":

Although it has many advantages, it has some disadvantages also. Most of them are inherent to expert systems. These disadvantages can be listed as:

1. A mistake at following the steps and answering the questions will cause the user to reach a wrong node. Although program has a choice of "go to previous question", the user should be very careful at answering.
2. If the domain knowledge gets larger, then maintenance might be difficult in terms of building the correct rules.
3. Observations gained importance due to operation principles of the program. All of the direct recommendations and the cases will be available according to observations of the failure.

For further studies, it is recommended to enlarge the domain knowledge by interacting with all of the COUGAR experts via different types of tools. Also other expert system tools, like case-based reasoning, can be explored in order to see and compare the system usefulness in all kind of helicopter domains as well as COUGAR. The most attractive advantage of a case-based reasoning system is the easiness of describing the failure to the system. Since all of the cases will be stored in the system library with their descriptive tags, calling a case might be easier and more accurate by the helps of search algorithms. Observation accuracy, which may change from one person to other, might not be an important factor as it is in rule-based reasoning.

As a conclusion, there isn't any decision support system in Turkish Army Aviation units for COUGAR helicopters as it for the other types. It is concluded that the decision support system could be very useful for the users by considering the aviation parts' prices, maintenance loads, expert technician distributions in the army and the value of knowledge that can be acquired very difficult.

SELECT BIBLIOGRAPHY

AHA, D. W., and GUPTA Kalyan Moy, (2002), "Causal Query Elaboration in Conversational Case-Based Reasoning", *Proceedings of the Fifteenth International Florida Artificial Intelligence Research Society Conference,* Pensacola Beach, FL: AAAI Press, 2002, (pp. 95-100).

ARSLAN Bora, RICCI Francesco, "Case Base Session Modeling and Personalization in a Travel Advisory System", Internet, available from http://ectrl.itc.it /rpec/RPEC-Papers/07-arslan.pdf, accessed on March 4 2004.

BRAMER Max, PREECE Alun, COENEN Frans, 2001, *RESESARCH AND DEVELOPMENT IN INTELLIGENT SYSTEMS XVII*, London ENGLAND, Springer-Verlag London Limited.

CRAW Susan, JARMULASK Jacek, ROWE Ray, (2001), "Maintaining Retrieval Knowledge in a Case-Based Reasoning System", *Computational Intelligence*, Vol.: 17 Number: 2, 2001, (pp. 346-463).

DOHERTY Michael E., LEIGH William E., 1986, *DECISION SUPPORT AND EXPERT SYSTEMS,* United States of America, South-Western Publishing Co.

DURKIN John. 1994, *EXPERT SYSTEMS DESIGN AND DEVELOPMENT,* United States, Book Press.

EUROCOPTER FRANCE , 1988, Aerospatiale, SUPERPUMA, *Instruction Manual.*

EUROCOPTER FRANCE, *FLIGHT MANUAL,* Establissement de Marignane, Cedex France, Direction Technique Support.

FORD Nigel, 1991, *EXPERT SYSTEMS AND ARTIFICIAL INTELLIGENCE*, London, Library Association Publishing.

GUPTA Kalyan Moy, (1999), "Case-Based Troubleshooting Knowledge Management", *Preceedings of AAAI Spring Symposium in AI in Equipment Maintenance and Service Support,* Stanford, CA, March 22-24, 1999.

GUPTA Uma G., (1994), "How Case-Based Reasoning Solves New Problems", *INTERFACES 24: 6 November-December 1994,* (pp. 110-119).

IGNIZIO James P., 1991, *INTRODUCTION TO EXPERT SYSTEMS THE DEVELOPMENT AND IMPLEMANTATION OF RULE-BASED EXPERT SYSTEMS*, United States of America, R.R. Donnelley& Sons Company.

"Inventors, The Helicopters", Internet, available from http://inventors.about.com/library/inventors/blhelicoptor.htm, accessed on 01 May 2004.

KOHNO Takeshi, HAMADA Susumu, ARAKI Dai, KOJIMA Shoichi and TANAKA Toshikazu, (1997), "Error Repair and Knowledge Acquisition via Case-Based Reasoning", *Artificial Intelligence*, Vol.: 91, 1997, (pp. 85-101).

KOLODNER Janet, 1993, *CASED-BASED REASONING*, United States of America, Morgan Kaufmann Publishers, Inc.

LARI, Alizera, (2003), "A Decision Support System for Solving Quality Problems Using Case-Based Reasoning", *TQM&BUSINESS EXCELLENCE*, Vol.: 14, No: 6, August 2003, (pp. 733-745).

LEAKE David B., SMYTH Barry, WILSON David C., YANG Quiang, (2001), "INTRODUCTION TO THE SPECIAL ISSUE ON MAINTAINING CASE-BASED REASONING SYSTEMS", *Computational Intelligence*, Vol.: 17 Number: 2, 2001, (pp. 193-196).

MANTARAS Ramon Lopez de, PLAZA Enric, (1997), "Case-Based Reasoning: an Overview", *AI Communications*, 1997, (pp. 21-29), IOS Press.

Mc CLAVE James T., BENSON P.George, SINCICH Terry, 2001, *STATISTICS FOR BUSINESS AND ECONOMICS,* United States of America, Prentice Hall, Inc.

MOHAN, Chilukuri Krishna, 2000, *FRONTIERS OF EXPERT SYSTEMS: REASONING WITH LIMITED KNOWLEDGE*, United States, Kluwer Academic Publishers.

REINARTZ Thomas, IGLEZAKIS Ioannis and ROTH-BERGHOFER Thomas, (2001), "Review And Restore for Case-Base Maintenance", *Computational Intelligence* Vol.: 17, Number: 2, 2001, (pp. 215-234).

Report of a Working Party Council for Science and Society, 1989, *BENEFITS AND RISKS OF KNOWLEDGE-BASED SYSTEMS,* Great Britain, Biddles Ltd, Guildford& King's Lynn.

RUSSELL Roberta S., TAYLOR III Bernard W., 2003, *OPERATIONS MANAGEMENT*, United States, Prentice Hall.

SHIU Simon C.K., YEUNG Daniel S., SUN Chai, H., WANG Xi Z., (2001), "Transferring Case Knowledge To Adaptation Knowledge: An Approach for Case-Based Maintenance", *Computational Intelligence,* Vol.: 17, Number: 2, 2001, (pp. 295-314).

Table of Contents, Internet, available from http://www.infj.ulst.ac.uk/~cbdq23/teaching /com812j/notes/cbr_notes.doc, accessed on 10 January 2004.

"The Basics of Expert (Knowledge Based) Systems", Internet, available from http://www.arjhem.com/EXPERT.pdf, accessed on 23.05.2004.

TURBOMECA, *ENGINE MAINTENANCE MANUAL*, (Original Issue: March 1981, update no: 21 dated 30 May 2003), BORDES CEDEX, FRANCE.

TURBOMECA TRAINING CENTER, *MAKILA 1A1 TURBOSHAFT ENGINE Training Manual,* TARNOS-FRANCE, Centre d'Instruction.

APPENDIX A. CASES

CASES:

All of the cases in the decision support system can be referenced to topic related documents and maintenance manuals, which are listed below and in "Select Bibliography":

- TURBOMECA, (Original Issue: March 1981, Update no: 21 dated 30 May 2003) *ENGINE MAINTENANCE MANUAL*, BORDES CEDEX, FRANCE.
- TURBOMECA TRAINING CENTER, *MAKILA 1A1 TURBOSHAFT ENGINE Training Manual,* Centre d'Instruction TARNOS, FRANCE.
- EUROCOPTER FRANCE Aerospatiale, 1988, SUPERPUMA, *Instruction Manual,*.
- EUROCOPTER FRANCE, *FLIGHT MANUAL*, Establissement de Marignane, Direction Technique Support, Cedex, France.

Case 1:

Description: In electric system, there are one main battery, two AC alternators, two transformer rectifiers, two transformers and an emergency battery. Those electrical parts feed all of the systems' components on helicopter. Main suppliers are the alternators during the normal flight and on ground with rotor spinning above 245 RPM (rotation per minute). If the RPM is below 245, then the alternators cannot engage to the system and the main battery meets the entire electrical load, which is not a desirable situation.

Observation 1: "ELEC" warning light on 32α? **On.**

Observation 2: NR Indicator? **More than 245 less than 285.**

Observation 3: Circuit breakers? **In.**

Observation 4: Resetting the system? **Yes.**

Observation 5: "WARN" light? **On.**

Observation 6: "ALT1" or "ALT2" warning light? **On.**

Observation 7: Engines? **Running at flight detent.**

Observation 8: AC Voltmeter? **At 0.**

Problem: Alternator(s) does/do not engage into the system.

Cause 1: If the electrical connections of the alternator outlet are misconnected with the system input socket then the alternator phases (A-B-C) will not be able to couple to the system. (Those sockets and the alternators are usually disconnected during the long-term high-loaded maintenance periods).

Cause 2: If corrosion exist at the socket connection pins, then it may be impossible to engage the related alternator.

Cause 3: Mechanical failure of the alternator.

Solution: Stop the engine if it is running. Then unplug the alternator connections and spray with an anti-corrosion liquid and correctly plug the alternator phase connections again. Try the system again by starting the engine again. If the failure persists. Change alternator.

WARNING 1: It is enough to start one engine for controls after removing the fault.

Case 15:

Description: If there is a failure on the monocontactor (in the autopilot compartment), then it will be impossible to start the engines even the rotor brake lever is in forward position with normal automatic or manual start positions.

Observation 1: Engine acceleration? **No acceleration**.

Observation 2: Overspeed lights? **Extinguished.**

Observation 3: Rotor brake? **In forward position.**

Observation 4: Rotor brake warning light on 32 α panel? **Off**.

Observation 5: Starter S(1-2) light? **Off.**

Observation 6: Fuel flow control lever? **start detent**.

Observation 7: Noise of the starter? **No noise.**

Observation 8: DC loadmeter? **No load jump.**

Problem: It is impossible to start the engines due to free air in the fuel system.

Cause: A failure at monocontactor in the autopilot compartment causes this failure. This monocontactor has two different missions. First, it controls the illumination of “rotor brake” light on 32α warning panel. Second, it controls the “Overspeed” lights on strip panel.

Solution: Change the monocontactor in the autopilot compartment.

WARNING 1: In Phenix I helicopters monocontactor have two missions. Illumination of the rotor brake warning light on 32α and control of the Overspeed lights on the strip panel are controlled by that switch. Check the overspeed lights on the strip panel and the position of the rotor brake. If the overspeed light does not illuminate when the rotor brake control lever is in forward position then check the monocontactor by changing it with a new one.

WARNING 2: If you come across with this type of failure, there is a procedure to be able to transport the aircraft to the nearest airport or maintenance center by flying. Since the monocontactor is not operating correctly, then it will be impossible to start the engines in manual position. Only available procedure is starting the first engine by "Starting in High Wind" procedure. In this procedure another switch is used to energize the "Overspeed" circuit and lights on the strip panel. After starting the first engine, open the whole electric control panel on overhead panel. "Starting in High Wind" authorization microswitch is located between the "fuel flow control levers". Let the technician to press on that microswitch by hand and start the second engine while that microswitch is pressed. After starting both of the engines everything will operate normally. In the nearest maintenance center change the monocontactor.

Case 29:

Description: "Main system oil pressure" at "Main Gearbox" is read at the same value of "Emergency system oil pressure". In normal conditions main system's oil pressure should be above the oil pressure of the emergency system.

Observation 1: Engines? **Running at flight detent.**

Observation 2: NR value? **Between 250-256.**

Observation 3: MGB main system's oil pressure? **About emergency system's pressure.**

Observation 4: MBG.COOL warning light on 32α? **Off.**

Observation 5: MGB emergency system's pressure? **Normal.**

Observation 6: MGB temperature? **Rises and stabilizes within normal values.**

Problem: Low MGB (Main Gearbox) main system oil pressure on the indicator.

Cause: A temporary dirt contamination in the main system or a break or a specialty loss in the cable of the main system's transmitter or low oil level in the system or a damaged main system's pressure relief valve may cause this failure.

Solution: (If the oil level in the main gearbox is about 8 liters, then it is normal for such an abnormal reading. Main system cannot pump the oil to the system if the oil level is below 8 liters). Check the oil level. If necessary add oil to the system. Check the "MGB.COOL" warning light. It should be "OFF". Start and check the system again. If the oil level is normal, replace the main oil pressure transmitter with another one and start again. If the failure persists, remove the emergency system's "Check valve", and start one of the engines to check the pressure indicators. Since the main pump is adjusted for higher values of pressure to press, emergency system's oil pressure indicator will show the pressure of the main system. A reverse oil leak will be in the emergency system. If the emergency indicator shows the main pressure, then there is no failure about the pump. Check the cables and connections of the main system oil transmitter. There might be a broken cable in the socket of the transmitter or a loss on the cable. If the indicator of the emergency system still shows low values although the check valve is removed, than install the check valve of the emergency system, drop the oil level to about 8 liters and remove the check valve of the main system (near to filter body in the MGB compartment), and start one engine again to create a reverse leak in the main system to reset the system for any kind of possible dirt contamination. Then install check valve, add oil and start again normally.

WARNING 1: DO NOT accelerate the engine more than ground idle while starting the engine without the check valve in the systems.

WARNING 2: It is impossible to apply some fault isolation procedure at the failure of the main system's pressure relief valve since it is located to inner side of the MGB.

Case 30:

Description: During the main rotor balancing procedure, there are some points to pay attention.

Observation 1: Engines? **Running at flight detent.**

Observation 2: Feelings and control levers? **There is vibration.**

Observation 3: Cyclic control and feelings? **There is vibration.**

Problem: There is an imbalance problem in the main rotor. (After long-term maintenances and replacement of one of the rotor components, it is mandatory to perform the balancing procedures).

Cause: Replacement of a component at the rotor, a leakage or a play that is out of limit at one of the components or water leak into the tip-cap or a forgotten cloth part in the drop restrainer protector may result balance problems.

Solution: Check the main rotor components before performing any balancing. Especially pay great attention to "Frequency Adapters". There should be no viscoelastic gel leakage to out of the adapters. Play limits at static position should be in tolerance. See the relevant bulletins during the checks of this component. Also an out of tolerance oil leakage at the sleeve-spindle bearings will cause imbalance problems. Take all of the oil levels to the same level at sleeve-spindles. Blades should be clear at both sides, upper and down sides. Check the "tip-caps" of the blades for crack, attachment and water protection. Landing gears should be correctly adjusted in terms of hydraulic and nitrogen. Start balancing the main rotor on ground and with "Tracking". Install the targets to the tip-caps' inner rows' second screws at different shapes. Start one engine and adjust the "strobex" to the correct RPM. Wait until the MGB, IGB and TGB oil temperatures come about 40°C. Be sure that all of the blades are passing from the same track. If necessary, adjust the wrong blade(s) from the "pitch change rod". Here, all of the blades should be adjusted according to "yellow blade". It is the reference blade.

Then adjust the horizontal balance. Turn the helicopter into the wind direction. Check the clock direction and level of vibration while two or single engine running on ground, fuel flow control lever(s) are at flight detent, collective lever is at fully down, yaw pedals and cyclic are neuter. According to vibration level and clock direction find the required amount of balance weight and color of blade to put the balance weight. Put the weight to the relevant blade after stopping the engine(s). Restart after adding the weights and check again. Repeat the same procedure until the vibration level can be dropped below 0,15 IPS level on ground. It is recommended to drop the vibration below 0,05 IPS on ground. While taking the horizontal vibration levels, also take the vertical vibration levels from the special tool and follow the changes at vibration level while you are adjusting the horizontal level. After ground adjustments, fly at hover and take the horizontal vibration at hover again. It should be less than 0,15 IPS. If necessary adjust the horizontal vibration level in the same way the adjustments on ground. Also take the vertical vibration level on hover. After adjusting the horizontal at hover, perform normal flight at altitude min. AGL+1000 feet (GW should be at least 15400 lb). At 80 knots perform a level flight and take the horizontal and vertical vibration levels. Also you should look at the "track" while you are flying at 80 knots. Then, increase your speed until 16.5 pitch collective lever at the same altitude without climbing. Take the horizontal and vertical vibrations at this high speed. Also check the "track" at this speed. After landing, perform the necessary adjustments at horizontal vibration level and do the same things at hover, at 80 knots and at 16.5 pitch levels again. Always check the vertical vibration and "track". If the track is OK on ground and then the difference at the blade tracks is increasing by the increase in the speed (climbing), then an adjustment at the "trim tabs" of the blades might be necessary. It also affects the vertical vibration lever. After performing the necessary adjustments at the related parts, check the vibration levels again.

WARNING 1: If it is necessary to replace one of the adapters, it is mandatory to change the adapters as twin with the opposite one.

WARNING 2: Check the inner side of the "droop restrainers" protectors for foreign objects. Sometimes cloth parts are put in this part to prevent oil drops to MGB compartment.

WARNING 3: If the insulation of the tip-caps is not sufficient, some water may leak into the caps in rainy days and causes balance problems later on.

WARNING 4: DO NOT change the adjustments of the reference blade (yellow) except first installation. Prior to put the balance weights to the blades, check the opposite blade for weights. If there is balance weight at the opposite blade, first remove that weight and subtract that amount from the required amount. Put the new amount of the balance weight to the required blade.

WARNING 5: Max. Permitted balance weight per blade is 2 kg. max.

Case 33:

Description: Tail rotor is one of the most important components in terms of balancing. A balance problem at tail rotor will affect all the system by its high vibration level.

Observation 1: Tail servo control's situation? **Normal.**

Observation 2: Vibration level? **Abnormal.**

Observation 3: Yaw pedals and feelings? **There is vibration.**

Problem: There is excessive vibration at tail rotor. (It is also mandatory to perform tail rotor balance check after maintenance operations).

Cause 1: A changing component from the tail rotor or a forgotten sleeve for greasing or a failure at the pitch-change rods may cause this failure.

Cause 2: Wrong order of blade installation will cause this failure to go on.

Solution: Check the components of the tail rotor system. Be sure that the blades are clear. Also check the grease levels at the tail rotor blade spindles. It is better to refill the grease before starting the vibrations analyze. Check the pitch-change rods at the tail rotor. Do not permit out of limit movements at the components. Then correctly install the elements of the special tool to specified places at the tail rotor. Take a ladder and put it well out of rotor diameter. The ladder should face the tail rotor. Remove the previous weights from the blade sleeves before starting balancing. Install the target to the red blade's sleeve. Start the engine (single engine is enough for this check). Before taking the vibration level, let the TGB and IGB oil temperatures to increase until 30°C at least. While taking the value, set the "strobex" to correct value. Stand on the steps of the ladder at just opposite of the tail rotor. By the flashes of the strobex, the target on the sleeve will shine. Tune the strobex's RPM value until that target stays constant at a clock direction. Then take the value and put that value to related chart to see the required weight and blade. According to the value and clock direction designate a weight value and corresponding blades from the chart.

WARNING 1: Max. permitted weight per blade is 120 gr. At max.

WARNING 2: BE CAREFUL ABOUT THE INSTALLATION ORDER OF THE BLADES BEFORE STARTING THE BALANCING.

WARNING 3: If the failure persists, then check the connections of the IGB and TGB. There should be no movement at the bodies of TGB and IGB.

CASE 35:

Description: Hydraulic system has basically two main reservoirs and an emergency system reservoir. Main reservoirs' liquid levels are monitored continually by the transmitters and warning lights. "RH.LEVEL" or "LH.LEVEL" warning lights may illuminate although the levels are above the min. levels.

Observation 1: "BAT-EXT.PWR" Switch? **On.**

Observation 2: "RH LEV" hydraulic warning light? **On.**

Observation 3: "WARN" warning light? **On.**

Observation 4: "HYD" warning light on 32 alpha? **On.**

Problem: Hydraulic level dropping in the right hand hydraulic reservoir on ground.

Cause 1: If the hydraulic level drops below 2 liters in right hand hydraulic reservoir, then this light illuminates. The leakage should be detected and removed before the flight. Hydraulic servo controls should be examined carefully.

Cause 2: Check installation of the "tail servo shutoff solenoid valve" in detail. If the solenoid valve is not installed properly, then "RH LEV" warning light will not get to "OFF" although the level is well above 2 liters in the reservoir. This is a safety rule for the hydraulic systems.

Solution: Do not start the engine. If the level is below 2 liters then add appropriate hydraulic fluid to the system. If the level is well above 2 liters in the reservoir, then check the tail servo shutoff solenoid valve's position and installation.

WARNING 1: If you face with the warning light on ground check the installation of the solenoid valve in detail since the hydraulic levels are controlled in pre-flight checks by the pilots.

WARNING 2: Add the fluid by the ground hydraulic unit. After adding the fluid, DO NOT forget to send sample after starting the system.

CASE 40:

Description: Sometimes "operation continuing" warning light on the landing gear control panel comes on and some of three green "extended" warning lights may disappear.

Observation 1: Engines? **Running at flight detent.**

Observation 2: Main rotor? **Running at normal speeds.**

Observation 3: Left hand hydraulic main pressure indicator? **At 175 bars.**

Observation 4: Green AUX.HYD.P warning light? **Off.**

Observation 5: Yellow "operation continuing" warning light? **On.**

Observation 6: Green "Extended" lights? **disappeared.**

Observation 7: "Ground/Flight" warning light on 32α? **On after 10 seconds.**

Observation 8: WARN warning light? **On.**

Observation 9: During first two retraction? **No.**

Problem: Low nitrogen pressure or hydraulic level in the landing gears.

Cause 1: A leakage of nitrogen or hydraulic fluid from the landing gears.

Cause 2: A failure at the landing gear switches may cause this failure.

Solution: After landing, check the nitrogen pressure and hydraulic level at the landing gears. If they are correct, then check the switches.

WARNING 1: Ground/Flight warning light will affect your autopilot, pitot-head heating and aux. hydraulic pump's operation logic.

CASE 41:

Description: Fuel system consists of six tanks; 3 for left engine fuel system, 3 for right engine fuel system. All of the tanks in one engine group are interconnected and a transfer pump interconnects both engine groups.

Observation 1: Engines? **Running at flight detent.**

Observation 2: Fuel level indicator? **Above "Low Fuel" warning level threshold.**

Observation 3: Fuel "Low level" indicator light? **On.**

Observation 4: "FUEL" warning light on 32α? **On.**

Observation 5: "WARN" warning light on console? **On.**

Observation 6: Fuel control panel Test? **All digits normal (8).**

Observation 7: "Boost Pumps? **On.**

Observation 8: Fuel Pressure Indicator? **In green arc.**

Observation 9: Feeder tank levels? **Below low level threshold.**

Problem: A failure at jet pump or a jamming at feeder tank check valve.

Cause 1: A failure at jet pump in the fuel system may cause this failure.

Cause 2: Clogging at jet pump may cause this failure.

Cause 3: Jammed at close or open position (differs according to duration of the flight) of feeder tank check valve may cause this failure.

Solution: Low level warning light takes its threshold value from only the feeder tanks. A drop solely on the feeder tanks will be enough for illumination of the warning light. It is a small possibility that the jet pump fails mechanically since it is a static pump and operated according to "Bernoulli" principle. However, checking the jet pump's mechanical connections may be necessary. The tip of the jet pump is very narrow to increase the outlet pressure. Dirt contamination at this outlet inhibits the jet pump operations and the feeder tank cannot be kept full during the flight. Clean the pump input and output. Check the jet pump filter also. Feeder tank check valve ensures the fuel flow from the transverse tanks to the feeder tanks; it allows only one direction movement of the fuel. If this valve stays at close or open position, then the feeder tank cannot be kept full again. Open or close positions affect the system based on the flight duration at the time of failure.

WARNING 1: Pay great attention to the jet pump filter.

WARNING 2: Failure at Boost Pumps is also another reason for this failure. If the pumps do not operate properly, then it will be impossible to benefit from jet pump.

WARNING 3: Low level warning lights take their measurements form only feeder tanks. However in normal conditions the fuel level indicators show the total amount in the tanks. For that reason, DO NOT consider the situation as "wrong warning" before checking the tanks separately. To see the levels of the tanks separately, you should press on the buttons on fuel control panel.

CASE 43:

Description: During the test flight many of the checks are executed; like "ENGINE POWER ASSURANCE CHECK". "THERMAL LOAD CHECK" is also controlled by those values. There are some preliminary conditions, which ones should be satisfied before recording the engine values.

Observation 1: Main rotor? **Running at normal speeds.**

Observation 2: Engine(s)? **Running at flight detent.**

Observation 3: Helicopter's situation? **Level flight.**

Observation 4: P2 valve at overhead panel? **Closed.**

Observation 5: Pitch indicator? **As required.**

Observation 6: NG reading? **More than 30000.**

Observation 7: Flight duration at level flight? **At least 3 minutes.**

Observation 8: Helicopter's direction? **Headed into wind.**

Observation 9: Turbulence level? **No or Light.**

Observation 10: Situation of torquemeter? **In emergency.**

Observation 11: Bleed Valve (BLD) lights? **Off.**

Observation 12: Engine air filters? **Opened.**

Observation 13: Thermal load check? **Abnormal.**

Problem: There is thermal load on the engine beyond the specified limits when the values are entered to the relevant chart.

Cause 1: A defective compressor section may cause this failure.

Cause 2: Dirty engine's test results may be beyond the limits.

Cause 3: If the engine inlet filters are in "closed" position (bullets are in forward) during the tests, then the measurements of the engine parameters will be wrong.

Cause 4: Wrong recording will cause this failure.

Cause 5: Calibration failures of the indicators will cause this failure.

Solution: Be sure about correct indication of the indicators. Wrong measurement indications will cause the pilots record wrong values to the test sheet. If possible change the NG and T4 indicators between the engines and check the system again in flight. Before test flight wash the engine once again as specified in flight manual's 8^{th} section. Pay great attention during recording the flight values of the engines. Engine air intake filter bullets should be opened (bullets "back" position) position. If the bullets are in closed (bullets "forward" position), then T4 values will be more than normal. Test the helicopter after checking, controlling and washing the relevant parts, test the system in flight again. If necessary change the relevant part(s).

WARNING 1: Bleed valves should be in "OFF" position before recording.

WARNING 2: Wait at least three minutes in level flight to let the measurements stabilize at normal values.

WARNING 3: A defective compressor section will give some clues about itself. If there is a "T4" split between the engines despite the same NG values and torque values and you are sure about correct indications, then it may be a critical sign for a real thermal load problem.

WARNING 4: If the failure occurs on both of the engines, then it should be considered as a clue for wrong recording or a failure at P2 system.

CASE 44:

Description: Airspeed indication system controls are done during the test flight. There are some preliminary requirements to satisfy before recording the values.

Observation 1: Main rotor? **Running at normal speeds.**

Observation 2: Engine(s)? **Running at flight detent.**

Observation 3: Helicopter's situation? **Level flight.**

Observation 4: P2 valve at overhead panel? **Closed.**

Observation 5: Pitch indicator? **At 15.5.**

Observation 6: Flight duration at level flight? **At least 3 minutes.**

Observation 7: Helicopter's direction? **Headed into wind.**

Observation 8: Turbulence level? **No, Light or moderate.**

Observation 9: Bleed Valve (BLD) lights? **Off.**

Observation 10: Altimeters? **At 29.92 inchHg.**

Observation 11: Trim indicator? **At middle.**

Problem: There is a problem at airspeed indication system accuracy.

Cause 1: A failure or leakage at pitot-static system will cause this failure.

Cause 2: A failure at airspeed indicator may cause this failure.

Cause 3: A failure at electrical pitch indicator may cause this failure.

Cause 4: Wrong setting of "pitot-static emergency valve" may cause this failure.

Cause 5: Wrong recording may cause this failure.

Solution: If there is a discrepancy between two airspeed indicators' readings, then check the indicator system carefully on ground with the special tool kit as specified in the related chapter. Since the measurements are done according to electrical pitch settings, a failure at electrical pitch system will cause more or less measurements during the test flight. Pay attention to the mechanical pitch indicator also. If necessary set the pitch according to mechanical pitch to "specified pitch value-0.5

pitch" and check the system again. Check the pitot-static emergency valves for position. They should be in "normal" position.

WARNING 1: Comparison of both of the airspeed indicators will be critical at deciding the action.

WARNING 2: Emergency valves should be checked manually not just considering the position. Valve's position may be wrong from the inner side.

WARNING 3: Be careful about the position of the trim indicators (buls).

CASE 47:

Description: Autopilot unit controls the flight controls throughout the flight. Sometimes A.P. warning light may flash on 32α at the sudden control movements.

Observation 1: Engine(s)? **Running at flight detent.**

Observation 2: Main rotor? **Running at normal speeds.**

Observation 3: Electrical system control panel? **No failure indication.**

Observation 4: ALT1 and ALT2 control switches? **On.**

Observation 5: "ELEC" warning light on 32α? **Off.**

Observation 6: "GYROS" warning light on 32α? **Off.**

Observation 7: Autopilot test on ground? **0.**

Observation 8: "WARN" warning light on console? **Flashing.**

Observation 9: "A.P." autopilot warning light 32α? **Flashing.**

Observation 10: LANE1 warning lights on autopilot? **Off.**

Observation 11: LANE2 warning lights on autopilot? **Off.**

Observation 12: “PITCH”, “ROLL” and “COLL-TRIM” lights on autopilot? **One of them is flashing.**

Problem: There is an autopilot failure.

Cause 1: A loosen bulb on the autopilot control panel causes this failure.

Solution: Monitor the bulbs while you move the flight controls suddenly. AP warning light and the relevant loosen bulb on the autopilot control panel will blink at the same time. Tighten the bulb and its body. If there is no blinking bulb on the autopilot control panel, then check the “Autopilot computer” from the front battery compartment and its sockets.

WARNING 1: This failure may not be observed on ground tests since the blinking is related with the sudden movements of the flight controls. A completely loosen bulb will show itself on ground.

CASE 48:

Description: Cougar helicopters are equipped with “Rosemount Ice Detectors”. System can be tested on ground as specified in test procedure. The system has some indicating lights; “ICE.D” yellow warning light on 32α, “ICE” blue warning light on console (instrument panel) and “FAIL” red flag on the indicator itself.

Observation 1: Engine(s)? **Running at flight detent.**

Observation 2: DC electrical supply? **On.**

Observation 3: AC electrical supply? **On.**

Observation 4: "ICE DETECTOR" switch? **On.**

Observation 5: "Ice detector" on ground test? **Ok.**

Observation 6: "WARN" warning light on console? **On.**

Observation 7: "ICE.D" warning light on 32α? **On.**

Observation 8: Electrical system control panel? **No failure indication.**

Observation 9: Circuit breakers? **In.**

Observation 10: "FAIL" warning flag on "ICE DETECTOR"? **Appeared.**

Observation 11: NG indicator? **About 28000.**

Problem: There is a failure at "Ice Detector" system.

Cause 1: A failure at the system "pressure reducing valve" causes this failure.

Cause 2: A failure at 151W pressure switches block will cause this failure.

Cause 3: A failure at the control panel 149W may cause this failure.

Solution: By increasing the NG by pulling the collective lever from its "fully down" position P2 pressure increases. If the pressure reducing valve is not operating correctly, then it will cause pressure increment in the pressure switches block and activate the high pressure switch at 3.1 bars, which may correspond to the pressure of 28000 NG approximately. Check by replacing from other helicopter and test the system. If necessary change the "pressure reducing valve".

Since the failure occurs at high NG values, about 28000 NG (not on ground, engines running at flight detent, collective lever is fully down), the failure seems related with P2 system. By increment in the NG value, pressure increases at the 151W pressure switches block; "high pressure switch" detects the increment and gets to "rest" position, which causes the yellow "ICE.D" warning light to illuminate. High-pressure switch is set to detect and activate at 3.1 bars. If the high-pressure switch has lost its specialty and lost the threshold value, then it will cause the light to illuminate although the real pressure is well below 3.1 bars at the system inlet. Change the pressure switches block.

A failure of the control panel 149W at any stage of the flight will cause this failure. However, this failure will be permanent and easy to isolate.

WARNING 1: This failure cannot be observed on ground tests since the warning lights illuminate at 28000 NG, which is well above the ground values at collective lever is "fully down".

APPENDIX B. PROGRAM CODE FOR "THERMAL LOAD CHECK"

```
Procedure TLCheck;
Label
TLC,TLC2,TLC3,TLC4,TLC5,TLC6,TLC7,TLC8,TLC9,TLC10,TLC11,TLC12,TLC13,
      other,proend;
Var input:integer;
Begin
TLC:

   Clrscr;

   Writeln('Engines?');
   Writeln;
   Writeln('1. Stopped');
   Writeln('2. Running at flight detent');
   Writeln('3. Other');
   Writeln('4. Back to main menu');
   Writeln;
   Write('>');
   Read(input);

   If input<1 then begin writeln('Invalid input. Please select one
of the choices above from 1-4. Press Enter to Continue');
   read;goto TLC;end;

   If input>4 then begin writeln('Invalid input. Please select one
of the choices above from 1-4. Press Enter to COntinue');
   read;goto TLC;end;

   If input=4 then goto proend;
   If input=2 then goto TLC2;
   If input=3 then goto Other;
   If input=1 then begin writeln('Do not care');
                      writeln('Press enter to go to main menu');
                      Readln(input1);goto proend;end; ;

TLC2:
       Clrscr;

        Writeln('Main rotor?');
        Writeln;
        Writeln('1. Stopped');
        Writeln('2. Running at normal speed');
        Writeln('3. Other');
        Writeln('4. Back to previous question');
        Writeln('5. Main Manu');
        Writeln;
```

```
        Write('>');
        Read(input);

        If input<1 then begin
        writeln('Invalid input. Please select one of the choices
above from 1-5. Press Enter to Continue');
        read;goto TLC2;end;

        If input>5 then begin
        writeln('Invalid input. Please select one of the choices
above from 1-5. Press Enter to Continue');
        read;goto TLC2;end;

        If input=5 then goto proend;
        If input=4 then goto TLC;
        If input=2 then goto TLC3;
        If input=3 then goto other;
        If input=1 then begin writeln('Double shaft failure');
                         writeln('Press enter to go to main menu');
                         Readln(input1);goto proend;end;
TLC3:
       Clrscr;

        Writeln('Thermal load control?');
        Writeln;
        Writeln('1. Normal');
        Writeln('2. Abnormal');
        Writeln('3. Other');
        Writeln('4. Back to previous question');
        Writeln('5. Main Manu');
        Writeln;
        Write('>');
        Read(input);

        If input<1 then begin
        writeln('Invalid input. Please select one of the choices
above from 1-5. Press Enter to Continue');
        read;goto TLC3;end;

        If input>5 then begin
        writeln('Invalid input. Please select one of the choices
above from 1-5. Press Enter to Continue');
        read;goto TLC3;end;

        If input=5 then goto proend;
        If input=4 then goto TLC2;
        If input=2 then goto TLC4;
        If input=3 then goto other;
        If input=1 then begin writeln('No problem');
                         writeln('Press enter to go to main menu');
                         Readln(input1);goto proend;end;

TLC4:
       Clrscr;

        Writeln('NG reading?');
        Writeln;
        Writeln('1. About 28.000');
```

```
        Writeln('2. More than 30.000');
        Writeln('3. Other');
        Writeln('4. Back to previous question');
        Writeln('5. Main Manu');
        Writeln;
        Write('>');
        Read(input);

        If input<1 then begin
        writeln('Invalid input. Please select one of the choices
above from 1-5. Press Enter to Continue');
        read;goto TLC4;end;

        If input>5 then begin
        writeln('Invalid input. Please select one of the choices
above from 1-5. Press Enter to Continue');
        read;goto TLC4;end;

        If input=5 then goto proend;
        If input=4 then goto TLC3;
        If input=2 then goto TLC5;
        If input=3 then goto other;
        If input=1 then begin writeln('Increase it over 30.000');
                         writeln('Press enter to go to main menu');
                         Readln(input1);goto proend;end;
TLC5:
       Clrscr;

        Writeln('Pitch indicator?');
        Writeln;
        Writeln('1. at 15.5');
        Writeln('2. as required');
        Writeln('3. Other');
        Writeln('4. Back to previous question');
        Writeln('5. Main Manu');
        Writeln;
        Write('>');
        Read(input);

        If input<1 then begin
        writeln('Invalid input. Please select one of the choices
above from 1-5. Press Enter to Continue');
        read;goto TLC5;end;

        If input>5 then begin
        writeln('Invalid input. Please select one of the choices
above from 1-5. Press Enter to Continue');
        read;goto TLC5;end;

        If input=5 then goto proend;
        If input=4 then goto TLC4;
        If input=2 then goto TLC6;
        If input=3 then goto other;
        If input=1 then begin writeln('Increase pitch until NG goes
beyond 30.000 NG');
                         writeln('Press enter to go to main menu');
                         Readln(input1);goto proend;end;
TLC6:
```

```
        Clrscr;

         Writeln('Flight duration at level flight?');
         Writeln;
         Writeln('1. One minute');
         Writeln('2. At least 3 minutes');
         Writeln('3. Other');
         Writeln('4. Back to previous question');
         Writeln('5. Main Manu');
         Writeln;
         Write('>');
         Read(input);

         If input<1 then begin
         writeln('Invalid input. Please select one of the choices
above from 1-5. Press Enter to Continue');
         read;goto TLC6;end;

         If input>5 then begin
         writeln('Invalid input. Please select one of the choices
above from 1-5. Press Enter to Continue');
         read;goto TLC6;end;

         If input=5 then goto proend;
         If input=4 then goto TLC5;
         If input=2 then goto TLC7;
         If input=3 then goto other;
         If input=1 then begin writeln('Fly at least 3 minutes before
recording');
                          writeln('Press enter to go to main menu');
                          Readln(input1);goto proend;end;
TLC7:
        Clrscr;

         Writeln('Direction of Helicopter?');
         Writeln;
         Writeln('1. Diffirent direction with the wind');
         Writeln('2. Headed into wind');
         Writeln('3. Other');
         Writeln('4. Back to previous question');
         Writeln('5. Main Menu');
         Writeln;
         Write('>');
         Read(input);

         If input<1 then begin
         writeln('Invalid input. Please select one of the choices
above from 1-5. Press Enter to Continue');
         read;goto TLC7;end;

         If input>5 then begin
         writeln('Invalid input. Please select one of the choices
above from 1-5. Press Enter to Continue');
         read;goto TLC7;end;

         If input=5 then goto proend;
         If input=4 then goto TLC6;
         If input=2 then goto TLC8;
```

```
        If input=3 then goto other;
        If input=1 then begin writeln('Turn and fly into wind');
                        writeln('Press enter to go to main menu');
                        Readln(input1);goto proend;end;
TLC8:
       Clrscr;

        Writeln('Turbulence level?');
        Writeln;
        Writeln('1. Severe turbulence');
        Writeln('2. Light turbulence');
        Writeln('3. Other');
        Writeln('4. Back to previous question');
        Writeln('5. Main Manu');
        Writeln;
        Write('>');
        Read(input);

        If input<1 then begin
        writeln('Invalid input. Please select one of the choices
above from 1-5. Press Enter to Continue');
        read;goto TLC8;end;

        If input>5 then begin
        writeln('Invalid input. Please select one of the choices
above from 1-5. Press Enter to Continue');
        read;goto TLC8;end;

        If input=5 then goto proend;
        If input=4 then goto TLC7;
        If input=2 then goto TLC9;
        If input=3 then goto other;
        If input=1 then begin writeln('Do not care thermal load
checks with severe turbulence');
                        writeln('Press enter to go to main menu');
                        Readln(input1);goto proend;end;
TLC9:
       Clrscr;

        Writeln('P2 valve at overhead panel?');
        Writeln;
        Writeln('1. Open');
        Writeln('2. Closed');
        Writeln('3. Other');
        Writeln('4. Back to previous question');
        Writeln('5. Main Manu');
        Writeln;
        Write('>');
        Read(input);

        If input<1 then begin
        writeln('Invalid input. Please select one of the choices
above from 1-5. Press Enter to Continue');
        read;goto TLC9;end;

        If input>5 then begin
        writeln('Invalid input. Please select one of the choices
above from 1-5. Press Enter to Continue');
```

```
        read;goto TLC9;end;

        If input=5 then goto proend;
        If input=4 then goto TLC8;
        If input=2 then goto TLC10;
        If input=3 then goto other;
        If input=1 then begin writeln('Close it and check again');
                          writeln('Press enter to go to main menu');
                          Readln(input1);goto proend;end;

TLC10:
       Clrscr;

        Writeln('Torquemeter situation?');
        Writeln;
        Writeln('1. In normal position');
        Writeln('2. In emergency position');
        Writeln('3. Other');
        Writeln('4. Back to previous question');
        Writeln('5. Main Manu');
        Writeln;
        Write('>');
        Read(input);

        If input<1 then begin
        writeln('Invalid input. Please select one of the choices
above from 1-5. Press Enter to Continue');
        read;goto TLC10;end;

        If input>5 then begin
        writeln('Invalid input. Please select one of the choices
above from 1-5. Press Enter to Continue');
        read;goto TLC10;end;

        If input=5 then goto proend;
        If input=4 then goto TLC9;
        If input=2 then goto TLC11;
        If input=3 then goto other;
        If input=1 then begin writeln('Check adjustment switch for
correct operation');
                          writeln('Press enter to go to main menu');
                          Readln(input1);goto proend;end;

TLC11:
       Clrscr;

        Writeln('BLEED valve light?');
        Writeln;
        Writeln('1. On');
        Writeln('2. Off');
        Writeln('3. Other');
        Writeln('4. Back to previous question');
        Writeln('5. Main Manu');
        Writeln;
        Write('>');
        Read(input);

        If input<1 then begin
```

```
        writeln('Invalid input. Please select one of the choices
above from 1-5. Press Enter to Continue');
        read;goto TLC11;end;

        If input>5 then begin
        writeln('Invalid input. Please select one of the choices
above from 1-5. Press Enter to Continue');
        read;goto TLC11;end;

        If input=5 then goto proend;
        If input=4 then goto TLC10;
        If input=2 then goto TLC12;
        If input=3 then goto other;
        If input=1 then begin writeln('Bleed lights should be closed
before the test');
                         writeln('Press enter to go to main menu');
                         Readln(input1);goto proend;end;
TLC12:
       Clrscr;

        Writeln('Engine air filters?');
        Writeln;
        Writeln('1. Closed');
        Writeln('2. Open');
        Writeln('3. Other');
        Writeln('4. Back to previous question');
        Writeln('5. Main Manu');
        Writeln;
        Write('>');
        Read(input);

        If input<1 then begin
        writeln('Invalid input. Please select one of the choices
above from 1-5. Press Enter to Continue');
        read;goto TLC12;end;

        If input>5 then begin
        writeln('Invalid input. Please select one of the choices
above from 1-5. Press Enter to Continue');
        read;goto TLC12;end;

        If input=5 then goto proend;
        If input=4 then goto TLC11;
        If input=2 then goto TLC13;
        If input=3 then goto other;
        If input=1 then begin writeln('They should be open before
recording');
                         writeln('Press enter to go to main menu');
                         Readln(input1);goto proend;end;

TLC13:
        writeln('Solution to your problem is availablein case 43.
Please read case 43.');
        writeln('To display case 43 press Enter');
        readln(input1);
        Writeln('case43');
        Writeln;
        Writeln('Press enter to go to Main Menu');
```

```
        Readln;
        Goto proend;

Other:
        Writeln('Please describe your case to your system engineer');
        Writeln('Press enter to go to main menu');
        Readln(input1);goto proend;


proend:
end;
```