İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

INTERPRETATION OF AIRCRAFT ENGINE CONDITION MONITORING DATA WITH AN EXPERT SYSTEM

M.Sc. Thesis by

Yalçın ŞENLİOL, B.Sc.

Department : **Aeronautical & Astronautical Engineering**

Programme: **Aeronautical & Astronautical Engineering**

JUNE 2008

<u>İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY</u>

INTERPRETATION OF AIRCRAFT ENGINE CONDITION MONITORING DATA WITH AN EXPERT SYSTEM

M.Sc. Thesis by
Yalçın ŞENLİOL, B.Sc.
(511031045)

Date of submission : 5 May 2008

Date of defence examination: 9 June 2008

Supervisor (Chairman): Prof. Dr. İbrahim ÖZKOL

Members of the Examining Committee Assoc. Prof.Dr. Erol Uzal

Assist. Prof.Dr. Melike N. Bayraktar

JUNE 2008

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

UÇAK MOTORUNUN DURUM İZLEME VERİLERİNİ YORUMLAYAN UZMAN SİSTEM

YÜKSEK LİSANS TEZİ

Müh. Yalçın ŞENLİOL
(511031045)

Tezin Enstitüye Verildiği Tarih : 5 Mayıs 2008
Tezin Savunulduğu Tarih : 9 Haziran 2008

Tez Danışmanı :	**Prof.Dr. İbrahim ÖZKOL**
Diğer Jüri Üyeleri	**Doç.Dr. Erol Uzal**
	Yrd. Doç. Dr. Melike N. Bayraktar

HAZİRAN 2008

PREFACE

I am dedicating this thesis to my dear parents who provided their all support to me during my lifetime.

I want to thank to Prof. Dr. İbrahim Özkol who always supported and encouraged me with his guidance and criticises and kept my motivation at high level.

I also want to thank Eng. Utku Ataç for his great support for providing maintenance data and his expertise.

Finally I want to thank to Eng. Erhan Kurubaş who guided me with his proficiency during development of the expert system.

2 May 2008 Yalçın Şenliol

CONTENTS

ABBREVIATIONS

AI	: Artificial Intelligence
IPC	: Intelligent Project Coach
ES	: Expert System
KBS	: Knowledge Based Systems
ECM	: Engine Condition Monitoring
FOQA	: Flight Operational Quality Assurance
FAA	: Federal Aviation Administration
SAGE	: System for the Analysis of Gas Turbine Engines
FOD	: Foreign Object Damage
EGT	: Eghaust Gas Temperature
FF	: Fuel Flow
N2	: Core Speed
DEGT	: Delta Eghaust Gas Temperature
DFF	: Delta Fuel Flow
DN2	: Delta Core Speed

LIST OF FIGURES

INTERPRETATION OF AIRCRAFT ENGINE CONDITION MONITORING DATA WITH AN EXPERT SYSTEM

SUMMARY

There is no doubt that one of the most important issues of aircraft maintenance is the engine maintenance. Some engine data are acquired in different part of a flight and considered in engine maintenance. This is called as engine condition monitoring and it assists in managing and forecasting engine maintenance. Today, interpreting of engine condition monitoring data is done by human experts employed in aircraft engine maintenance departments. Human experts check and consider the engine condition monitoring data which are formed by some different engine condition monitoring software. They are interpreting some parameters with their expertise and technical documents of engine manufacturer and then they decide action plans and schedule maintenance programs for the engines. The expert system which is presented in this thesis, can interpret the same parameters like a human expert. The system can monitor momentary condition of the engine and inferences advices for engine maintenance. The expert system can be used as assistant to human experts and can be used for training purposes.

UÇAK MOTORUNUN DURUM İZLEME VERİLERİNİ YORUMLAYAN UZMAN SİSTEM

ÖZET

Uçak bakımlarının en önemli kısmı uçak motorunun bakımıdır. Uçuşun farklı safhalarında motordan bazı veriler okunarak bakım esnasında değerlendirilmek üzere hafızada tutulur. Bu verilerin uçak bakımı sırasında değerlendirilmesi motor durumunun izlenilmesi olarak adlandırılır ve motor bakımlarının yapılması ve belirli bir dönem için planlanmasında kullanılır. Günümüzde motor durum izleme verilerinin yorumlanması uçak motor bakım bölümlerindeki insan uzmanlar tarafından yapılmaktadır. İnsan uzmanlar, çeşitli motor durum izleme yazılımları tarafından oluşturulan motor durum izleme verileri incelemektedirler. Uzmanlık bilgilerini, deneyimlerini ve motor üreticisi firmanın teknik dokümanlarını bazı parametrelerin yorumlanmasında kullanan insan uzmanlar, motorların bakımı esnasında yapılması gerekenler için kararlar vermekte ve motor bakımlarının planlamasını yapmaktadırlar. Bu çalışmada geliştirilen uzman sistem, insan uzmanların yorumladıkları bu parametreleri aynı insan uzmanlar gibi yorumlayabilmektedir. Uzman sistem, uçak motorunun o anki durumunu gösterebilmekte ve motor bakımı için tavsiyeler verebilmektedir. Bu uzman sistem, insan uzmanların iş yardımcısı olarak kullanılabileceği gibi eğitim amaçlı olarak da kullanılabilir.

1. ARTIFICIAL INTELLIGENCE

1.1. Introduction

Computer systems are becoming commonplace in last decade and they are almost ubiquitous. They can be found central to the functioning of most business, governmental, military, environmental, and health-care organizations. They are also a part of many educational and training programs. But these computer systems, while increasingly affecting our lives, are rigid, complex, and incapable of rapid change. To help us and our organizations cope with the unpredictable eventualities of a volatile world, these systems need capabilities that will enable them to adapt readily to change. They need to be intelligent. The computer systems used for such purposes must also be intelligent. Health-care providers require easy access to information systems so they can track health-care delivery and identify the most recent and effective medical treatments for their patients' conditions. Crisis management teams must be able to explore alternative courses of action and support decision making. Educators need systems that adapt to a student's individual needs and abilities. Businesses require flexible manufacturing and software design aids to maintain their leadership position in information technology, and to regain it in manufacturing. [1]

Advanced information technology can help meet these and many other needs in our society. Advances in computer and telecommunications have made available a vast quantity of data, and given us computational power that puts the equivalents of mainframes on our desktops. However, raw information processing power alone, like brute strength, is useful but insufficient. To achieve their full impact, computer systems must have more than processing power-they must have intelligence. They need to be able to assimilate and use large bodies of information and collaborate with and help people find new ways of working together effectively. The technology must become more responsive to human needs and styles of work, and must employ more natural means of communication.

To address the critical limitations of today's systems, we must understand the ways people reason about and interact with the world, and must develop methods for incorporating intelligence in computer systems. By providing computer programs that amplify human cognitive abilities and increase human productivity, reach, and effectiveness, we can help meet needs in industries like health care, education, service, and manufacturing [1].

Artificial intelligence (AI) is a field that studies intelligent behavior in humans using the tools-theoretical and experimental-of computer science. The field simultaneously addresses one of the most profound scientific problems-the nature of intelligence-and engages in pragmatically useful undertakings: developing intelligent systems. The concepts, techniques, and technology of AI offer us a number of ways to discover what intelligence is-what one must know to be smart at a particular task-and a variety of computational techniques for embedding that intelligence in a program.

1.2. AI Research Areas

Researches have been approaching AI techniques in much kind of application areas. These can be shown below [2].

- Games
- Artificial Life
- Theorem Proving
  - Prolog
  - Constraint Logic Programming
  - Parallel Prolog
- Natural Language Processing
  - Machine Translation
- Knowledge Based Systems
  - Expert Systems
  - Knowledge Based Simulation
  - General Knowledge Systems
- Machine Learning
  - Classifiers
  - Genetic Algorithms
  - Neural Networks
- Machine Discovery
  - Knowledge Discovery
  - Scientific Discovery
- Robotics
  - Task Planning

- Vision

- Pattern Recognition
  - Object Recognition
  - Optical Character Recognition
  - Voice Recognition

1.3. Recently Popular Application Models of AI Techniques

1.3.1. Intelligent Simulations

Systems that generate realistic simulated worlds would enable extensive, affordable training and education that can be made available anytime and anywhere. A new generation of intelligent simulation capabilities could support the construction of programs that model complex situations, involving both complicated devices and significant numbers of intelligent simulated people. Uses of these capabilities range from crisis management to product evaluation and entertainment. Many educational, commercial, military, entertainment, and scientific applications require the capability of generating realistic simulated worlds [3].

1.3.2. Intelligent Information Resources

Information resource specialist systems would support effective use of the vast resources of the information infrastructure [3]. These systems would work with their users to determine users' information needs, navigate the information world to locate appropriate data sources-and appropriate people-from which to extract relevant information. They would adapt to changes in users needs and abilities as well as changes in information resources. They would be able to communicate in human terms in order to assist those with limited computer training.

1.3.3. Intelligent Project Coaches

An intelligent project coach system can assist with design of a complex device (such as an airplane) or a large software system by helping to preserve knowledge about tasks, to record the reasons for decisions, and to retrieve information relevant to new problems [3]. It could help at the operational level to improve diagnosis, failure detection and prevention, and system performance. Project coach systems do not need to be experts themselves; rather, they could significantly boost capability and productivity by collaborating with human experts, assisting them by capturing and delivering organizational memory. They could function as coworkers, assisting and collaborating with design or operations teams for complex systems. They could also supply institutional memory. The IPC could remember and recall the rationale of previous decisions, and, in times of crisis, explain the methods and reasoning previously used to handle that situation. IPCs would typically incorporate intelligent simulation and information resources systems as components.

1.3.4. Robot Teams

Intelligent robot system teams can perform tasks that are dangerous, such as environmental clean-up, mine removal, fire-fighting, and rescue operations. They can also perform those tasks that, while essential to the smooth functioning of a society, are mundane, repetitive, or unappealing to human workers. Individual robot team members may have limited capabilities; the teams need not be fully independent. Instead they can work together under human supervision, with robots doing the work and people providing direction and guidance. AI techniques can enable robots to evolve from objects that must be strictly controlled to objects that can be managed [4]. Progress requires more sophisticated perception, integration of input from a range of sensors, and the joint use of symbolic and sensory information; enriched capabilities for robots to communicate with each other and with humans; abilities to plan individual and collective actions and to monitor and control their execution; and the ability to acquire new behaviors by learning or by being told.

1.3.5. Learning, Information Elicitation, and Automatic Adaptation

Systems that can generalize, learn from experience, and adapt to new circumstances have the potential to reach higher levels of performance. All high impact application systems can be more powerful if they can learn from experience. For example, information assistants that can learn will be able to tailor their information-retrieval process to a user's needs without having to be told exactly what to do; they will instead generalize from previous interactions with the user. Learning skills will enable an intelligent coach system to deal with new types of problems, for example, drawing on its experience in the design of one type of automobile and applying it to the design of another. Networks of robots and computer-based agents in simulated worlds can avoid future coordination problems by learning from their experience in interacting with other robots.

1.3.6. Coordination of Perception, Planning, and Acting

Intelligent systems must be able to plan-to determine appropriate actions for their perceived situation, then execute them and monitor the results. Planning requires advanced capabilities to represent and reason about time, action, perception and the mental states of other agents. To cope with realistic situations, systems must be able to deal with incomplete, uncertain, and rapidly changing information and must have mechanisms for allocating resources between thinking and acting. Information-resource systems, for example, need to plan the best way to acquire information, trading off the urgency of the request against the cost of accessing different databases, the expected time required, and the likelihood of success. Simulated agents, intelligent coaches for operating complex systems, and taskable robots will all have to cope with complex physical processes and situations in which the actions of other agents conspire to create a complex, unpredictable, dynamic environment [4].

1.3.7. Coordination and Collaboration

All of the high-impact application systems will require both human-computer and computer-computer collaborations [4]. Machines can be designed to follow fixed protocols, but any collaboration that involves people will have to take into account the ways that people work. The amount and types of communication between agents will also vary widely, depending on both the types of participants and the costs of communication.

1.3.8. Perception

Most of the high-impact application systems require an ability to handle several types of perceptual information. For example, teams of robots need vision, language, and touch capabilities to function realistically. Design associates will need to interact with their environments and other team members; operational associates will need to monitor the behavior of complex systems using a variety of sensing devices. Humanlike communication that will make computers more accessible to everyone will require advances in perceptual capabilities, such as image interpretation, gesture recognition, and spoken-language understanding.

1.3.9. Content-Based Retrieval

The Internet is already populated with enormous amounts of multimodal information, from pages containing images, text, and graphics to video with sound track. Existing language-processing and vision techniques can provide a starting point for intelligent, content-based retrieval of information [3]. Significant research challenges include determining the kinds of image annotations, video, and audio data that are needed to enable efficient and effective access; developing techniques for automatically processing raw data to produce these annotations; providing a means of representing multimodal queries, whether in query languages for users or as target translation languages for sophisticated human-computer communication systems; developing capabilities for performing these tasks quickly enough so that users can afford to search many images or videos; and integrating multiple access techniques.

1.3.10. Reasoning and Representation

Research in reasoning and representation is needed to support the full range of high-impact applications systems. For example, in software engineering, AI representation and reasoning techniques can be used to describe the interfaces of software components and to find how to connect components from different sources to achieve a complex goal. Intelligent information systems must deal with data that is imprecise, incomplete, uncertain, and time varying. They must be able to manage with domain knowledge that is incomplete, and they must do as they meet pressing real-time performance requirements [4]. Finding a solution that is guaranteed to be optimal-under any reasonable interpretation of optimal can be shown to be computationally intractable: it cannot be done efficiently no matter how much faster we make our computers. Consequently, we must develop fast heuristics that can be shown to lead to good-if not necessarily optimal- solutions.

1.3.11. Building Large Systems

The high-impact applications require solutions that retain efficiency and robustness in large-scale, demanding environments. The advances in AI technology necessary for large-scale applications can be achieved by researchers working collaboratively. It is important to develop and maintain large-scale knowledge-bases and program libraries and to create knowledge representation capabilities that will allow these shared resources to be used successfully. Building these systems will require combining AI methods with non-AI approaches and embedding AI technology within larger systems [4].

Brittleness has been a perennial problem with intelligent systems constructed to date: they are good at their task but their performance falls off drastically as they move away from that task. Human expertise is far more flexible; it rests on a large stock of common-sense knowledge about the world, a very large collection of basic facts and inferences. A substantial common-sense knowledge base would lend an important improvement to the performance of many systems. In addition, many of the fundamental scientific challenges require collaborative, interdisciplinary efforts in the cognitive sciences and engineering.

2. HISTORY OF AI

With early twentieth century inventions in electronics and the post–World War II rise of modern computers in Alan Turing's laboratory in Manchester, the Moore School at Penn, Howard Aiken's laboratory at Harvard, the IBM and Bell Laboratories. As a result of their awesome calculating power, computers in the 1940s were frequently referred to as giant brains. Although robots have always been part of the public's perception of intelligent computers, early robotics efforts had more to do with mechanical engineering than with intelligent control. Recently, though, robots have become powerful vehicles for testing our ideas about intelligent behavior. Moreover, giving robots enough common knowledge about everyday objects to function in a human environment has become a daunting task. It is painfully obvious, for example, when a moving robot cannot distinguish a stairwell from a shadow. Nevertheless, some of the most resounding successes of AI planning and perception methods are in NASA's autonomous vehicles in space.

But AI is not just about robots. It is also about understanding the nature of intelligent thought and action using computers as experimental devices. By 1944, for example, Herb Simon had laid the basis for the information-processing, symbol-manipulation theory of psychology: "Any rational decision may be viewed as a conclusion reached from certain premises…. The behavior of a rational person can be controlled, therefore, if the value and factual premises upon which he bases his decisions are specified for him."[5].

AI in its formative years was influenced by ideas from many disciplines. These came from people working in engineering, biology, experimental psychology, communication theory, game theory, mathematics and statistics, logic and philosophy and linguistics. These lines of work made their mark and continue to be felt, and our collective debt to them is considerable. But having assimilated much, AI has grown beyond them and has, in turn, occasionally influenced them. Only in the last half century have we had computational devices and programming languages powerful enough to build experimental tests of ideas about what intelligence is.

Turing's 1950 seminal paper in the philosophy journal Mind is a major turning point in the history of AI. The paper crystallizes ideas about the possibility of programming an electronic computer to behave intelligently, including a description of the landmark imitation game that we know as Turing's Test.

Symbol manipulation languages such as Lisp, IPL, and Prolog and time sharing systems gave programmers new power in the 1950s and 1960s. Nevertheless, there were numerous impressive demonstrations of programs actually solving problems that only intelligent people had previously been able to solve.

Arthur Samuel's checker-playing program, described in that collection but written in the 1950s, was a tour-de-force given both the limitations of the IBM 704 hardware for which the program was written as a checkout test and the limitations of the assembly language in which it was written. Checker playing requires modest intelligence to understand and considerable intelligence to master. Samuel's program is all the more impressive because the program learned through experience to improve its own checker-playing ability from playing human opponents and playing against other computers.

Newell and Simon [5] acknowledge the convincingness of Oliver Selfridge's early demonstration of a symbol-manipulation program for pattern recognition [6]. Selfridge's work on learning and a multi agent approach to problem solving, plus the work of others in the early 1950s, were also impressive demonstrations of the power of heuristics. The early demonstrations established a fundamental principle of AI.

Minsky summarized much of the work in the first decade or so after 1950 [7]: "The most central idea of the pre-1962 period was that of finding heuristic devices to control the breadth of a trial-and-error search. A close second preoccupation was with finding effective techniques for learning. In the post-1962 era the concern became less with "learning" and more with the problem of representation of knowledge and with the related problem of breaking through the formality and narrowness of the older systems. The problem of heuristic search efficiency remains as an underlying constraint, but it is no longer the problem one thinks about, for we are now immersed in more sophisticated subproblems, e.g., the representation and modification of plans" [7].

Knowledge representation has become a cornerstone of every AI program. [9] makes the case for a declarative knowledge representation that can be manipulated easily. McCarthy has been an advocate for using formal representations, in particular extensions to predicate logic, ever since. Research by McCarthy and many others on nonmonotonic reasoning and default reasoning, as in planning under changing conditions, gives us important insights into what is required for intelligent action and defines much of the formal theory of AI.

Another turning point came with the development of knowledge-based systems in the 1960s and early 1970s. Dendral program [9] in the mid-1960s as a “paradigm shift” in AI toward knowledge-based systems. Prior to that, logical inference, and resolution theorem proving in particular, had been more prominent.

Mycin and the thousands of expert systems following it became visible demonstrations of the power of small amounts of knowledge to enable intelligent decision-making programs in numerous areas of importance. Although limited in scope, in part because of the effort to accumulate the requisite knowledge, their success in providing expert-level assistance reinforces the old adage that knowledge is power.

3. EXPERT SYSTEMS

One major insight gained from early work in problem solving was the importance of domain-specific knowledge. Expert knowledge is a combination of a theoretical understanding of the problem and a collection of heuristic problem-solving rules that experience has shown to be effective in the domain. Expert systems are constructed by obtaining this knowledge from a human expert and coding it into a form that a computer may apply to similar problems.

An expert system is a set of programs that manipulate encoded knowledge to solve problems in a specialized domain that normally requires human expertise. An expert system's knowledge is obtained from expert sources and coded in a form suitable for the system to use its inference or reasoning processes. The expert knowledge must be obtained from specialists or other sources of expertise, such as texts, journal articles and data bases. This type of knowledge usually requires much training and experience in some specialized field such as medicine, geology, system configuration or engineering design. Once a sufficient body of expert knowledge has been acquired, it must be encoded in some form, loaded into a knowledge base, then tested, and refined continually throughout the life of system.

This reliance on the knowledge of a human domain expert for the system's problem solving strategies is a major feature of expert systems. Although some programs are written in which the designer is also the source of the domain knowledge, it is far more typical to see such programs growing out of collaboration between a domain expert such as a doctor, chemist, geologist, or engineer and a separate artificial intelligence specialist. The domain expert provides the necessary knowledge of the problem domain through a general discussion of her problem-solving methods and by demonstrating those skills on a carefully chosen set of sample problems. The AI specialist, or knowledge engineer, as expert systems designers are often known, is responsible for implementing this knowledge in a program that is both effective and seemingly intelligent in its behavior. Once such a program has been written, it is necessary to refine its expertise through a process of giving it example problems to solve, letting the domain expert criticize its behavior, and making any required changes or modifications to the program's knowledge. This process is repeated until the program has achieved the desired level of performance.

3.1. Background History

The first expert system to be completed was DENDRAL developed at Stanford University in the late 1960s. This system was capable of determining the structure of chemical compounds given a specification of the compound's constituent elements and mass spectrometry data obtained from samples of the compound. DENDRAL used heuristic knowledge from obtained from experienced chemists to help constrain the problem and thereby reduce the search space. During tests, DENDRAL discovered a number of structures previously unknown to expert chemists.

As researchers gained more experience with DENDRAL, they found how difficult it was elicit expert knowledge from experts. This led to the development of Meta-DENDRAL, a learning component for DENDRAL which was able to learn rules from positive examples, a form of inductive learning described later in detail.

Shorter after DENDRAL was completed, the development of MYCIN began at Stanford University. MYCIN is an expert system which diagnoses infectious blood diseases and determines a recommended list of therapies for the patient. As part of the Heuristic Programming Project at Stanford, several projects directly related MYCIN were also completed including a knowledge acquisition component called THEIRESIUS, a tutorial component called GUIDON, and a shell component called EMYCIN. EMYCIN was used to build other diagnostic systems including PUFF, a diagnostic expert for pulmonary diseases. EMYCIN also became the design model for several commercial expert system building tools.

MYCIN's performance improved significantly over a several year period as additional knowledge was added. Tests indicate that MYCIN's performance now equals or exceeds that of experienced physicians. The initial MYCIN knowledge base contained about only 200 rules. This number was gradually increased to more than 600 rules by the early 1980s. The added rules significantly improved MYCIN's performance leading to a 65% success record which compared favourably with experienced physicians who demonstrated only an average 60% success rate [10].

Other classic expert systems include the PROSPECTOR program for determining the probable location and type of ore deposits based on geological information about a site, the INTERNIST program for performing diagnosis in the area of internal medicine. Numerous other expert systems are currently solving problems in areas such as medicine, education, business, design, and science.

It is interesting to note that most expert systems have been written for relatively specialized, expert level domains [11]. These domains are generally well studied and have clearly defined problem-solving strategies. Problems that depend on a more loosely defined notion of "common sense" are much more difficult to solve by these means. In spite of the promise of expert systems, it would be a mistake to overestimate the ability of this technology. Current deficiencies include:

1. Difficulty in capturing "deep" knowledge of the problem domain. MYCIN, for example, lacks any real knowledge of human physiology. It does not know what blood does or the function of the spinal cord. Folklore has it that once, when selecting a drug for treatment of meningitis, MYCIN asked whether the patient was pregnant, even though it had been told that the patient was male. Whether this actually occurred or not, it does illustrate the potential narrowness of knowledge in expert systems.

2. Lack of robustness and flexibility. If humans are presented with a problem instance that they cannot solve immediately, they can generally return to an examination of first principles and come up with some strategy for attacking the problem. Expert systems generally lack this ability.

3. Inability to provide deep explanations. Because expert systems lack deep knowledge of their problem domains, their explanations are generally restricted to a description of the steps they took in finding a solution. For example, they often cannot tell "why" a certain approach was taken.

4. Difficulties in verification. Though the correctness of any large computer system is difficult to prove, expert systems are particularly difficult to verify. This is a serious problem, as expert systems technology is being applied to critical applications such as air traffic control, nuclear reactor operations, and weapons systems.

5. Little learning from experience. Current expert systems are handcrafted; once the system is completed, its performance will not improve without further attention from its programmers, leading to doubts about the intelligence of such systems.

In spite of these limitations, expert systems have proved their value in a number of important applications.

3.2. The Building Blocks of Expert Systems

Every expert system consists of two principal parts: the knowledge base; and the reasoning, or inference, engine.

The knowledge base of expert systems contains both factual and heuristic knowledge. Factual knowledge is that knowledge of the task domain that is widely shared, typically found in textbooks or journals, and commonly agreed upon by those knowledgeable in the particular field.

Heuristic knowledge is the less rigorous, more experiential, more judgmental knowledge of performance. In contrast to factual knowledge, heuristic knowledge is rarely discussed, and is largely individualistic. It is the knowledge of good practice, good judgment, and plausible reasoning in the field. It is the knowledge that underlies the "art of good guessing."

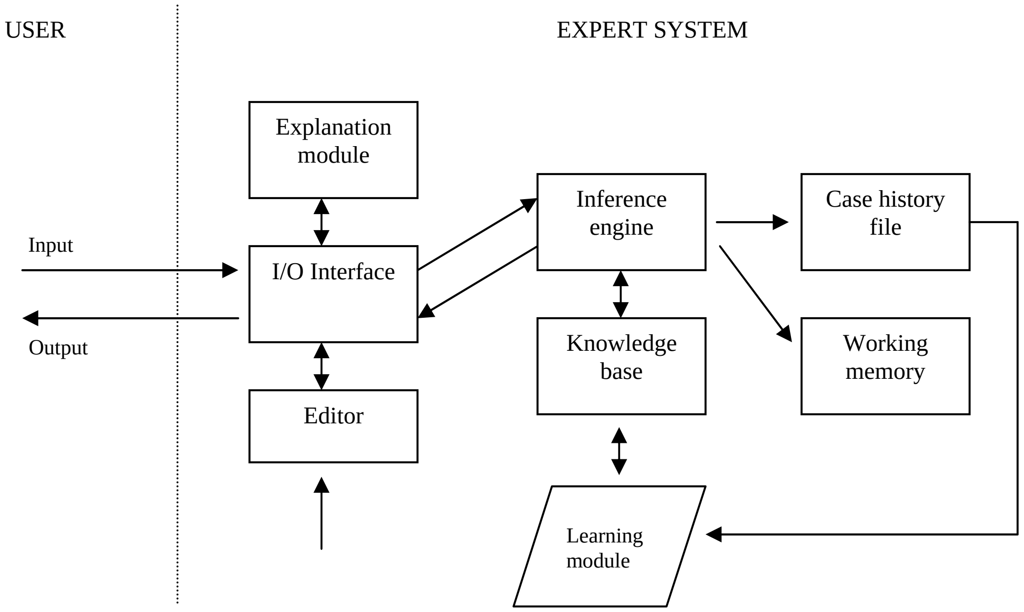


Figure 3.1: Components of a typical expert system

The main components of a typical expert system are depicted in figure 3.1. The solid lined boxes in the figure represent components found in most systems whereas the broken lined boxes are found in only a few such systems.

Though an expert system consists primarily of a knowledge base and an inference engine, a couple of other features are worth mentioning: reasoning with uncertainty, and explanation of the line of reasoning. Because an expert system uses uncertain or heuristic knowledge its credibility is often in question. Most expert systems have the ability to answer questions of the form: "Why is the answer X?" Explanations can be generated by tracing the line of reasoning used by the inference engine [12].

3.3. Knowledge engineering

Knowledge engineering is the art of designing and building expert systems, and knowledge engineers are its practitioners. Theoretically, a knowledge engineer is a computer scientist who knows how to design and implement programs that incorporate artificial intelligence techniques.

A knowledge engineer interviews and observes a human expert or a group of experts and learns what the experts know, and how they reason with their knowledge. The engineer then translates the knowledge into a computer-usable language, and designs an inference engine, a reasoning structure, that uses the knowledge appropriately. He also determines how to integrate the use of uncertain knowledge in the reasoning process, and what kinds of explanation would be useful to the end user.

Next, the inference engine and facilities for representing knowledge and for explaining are programmed, and the domain knowledge is entered into the program piece by piece. It may be that the inference engine is not just right; the form of knowledge representation is awkward for the kind of knowledge needed for the task; and the expert might decide the pieces of knowledge are wrong. All these are discovered and modified as the expert system gradually gains competence.

3.4. The Inference Process

The inference engine accepts user input queries and responses to questions through the I/O interface and uses this dynamic information together with the static knowledge stored in the knowledge base. The knowledge in the knowledge base is used to derive conclusions about the current case or situations as presented by the user's input. The inferring process is carried out recursively in three stages: match, select, and execute as can be seen in Figure 3.2.

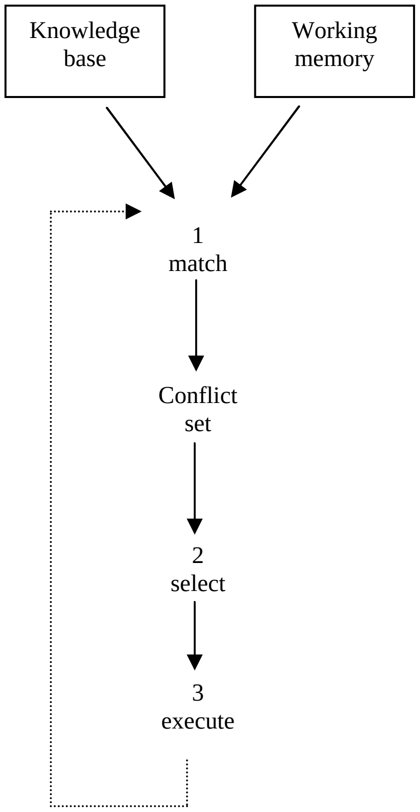


Figure 3.2: The production system inference cycle

3.5. Explaining How or Why

The explanation module provides the user with an explanation of the reasoning process when requested. This is done in response to a how query or a why query.

To respond to a how query, the explanation module traces the chain of rules tired during a consultation with the user. The sequence of rules that led to the conclusion is then printed for the user to actually see the reasoning process followed by the system in arriving in arriving at the conclusion. If the user does not agree with the reasoning steps presented, they may be changed using the editor.

To respond to a why query, the explanation module must be able to explain why certain information is needed by the inference engine to complete a step in the reasoning process before it can proceed. For example, in diagnosis a car that will not start, a system might be asked why it needs to know the status of the distributor spark. In response, the system would reply that it needs this information to determine if the problem can be isolated to the ignition system. Again, this information allows the user to determine if the system's reasoning steps appear to be sound. The explanation module programs give the user the important ability to follow the inferencing steps at any time during the consultation.

3.6. The Input - Output Interface

The input-output interface permits the user to communicate with the system in a more natural way by permitting the use of simple selection menus or the use of a restricted language which is close to a natural language. This means that the system must have special prompts or a specialized vocabulary which encompasses the terminology of the given domain of expertise. For example, MYCIN can recognize many medical terms in addition to various common words needed to communicate. For this, MYCIN has a vocabulary of some 2000 words.

Personal Consultant Plus, a commercial PC version of the MYCIN architecture, uses menus and English prompts to communicate with the user. The prompts, written in standard English, are provided by the developer during the system building stage. How and why explanations are also given in natural language form.

The learning module and history file are not common components of expert systems. When they are provided, they are used to assist in building and refining the knowledge base.

3.7. Characteristic features of expert systems

Expert systems differ from conventional computer systems in several important ways.

Expert systems use knowledge rather than data to control the solution process. ``In the knowledge lies the power`` is a theme repeatedly followed and supported. Much of the knowledge used is heuristics in nature rather than algorithmic.

The knowledge is encoded and maintained as a separate from the control program. As such, it is not compiled together with the control program itself. This permits the incremental addition and modification of the knowledge base without recompilation of the control programs. Furthermore, it is possible in some cases to use different knowledge bases with the same control programs to produce different types of expert systems. Such systems are known as expert system shells since they may be loaded with different knowledge bases.

Expert systems are capable of explaining how a particular conclusion was reached, and why requested information is needed during a consultation. This is important as it gives the user a chance to assess and understand the system's reasoning ability, thereby improving the user's confidence in the system.

Expert systems use symbolic representation for knowledge and perform their inference through symbolic computations that closely resemble manipulations of natural language.

Expert systems often reason with meta-knowledge; that is, they reason with knowledge about themselves, and their own knowledge limits and capabilities.

3.8. The Applications of Expert Systems

The spectrum of applications of expert systems technology to industrial and commercial problems is so wide as to defy easy characterization. The applications find their way into most areas of knowledge work. They are as varied as helping salespersons sell modular factory-built homes to helping NASA plan the maintenance of a space shuttle in preparation for its next flight. Applications tend to cluster into seven major classes [13].

3.8.1. Diagnosis and Troubleshooting of Devices and Systems of All Kinds

This class comprises systems that deduce faults and suggest corrective actions for a malfunctioning device or process. Medical diagnosis was one of the first knowledge areas to which ES technology was applied, but diagnosis of engineered systems quickly surpassed medical diagnosis. There are probably more diagnostic applications of ES than any other type. The diagnostic problem can be stated in the abstract as: given the evidence presenting itself, what is the underlying problem/reason/cause?

3.8.2. Planning and Scheduling

Systems that fall into this class analyze a set of one or more potentially complex and interacting goals in order to determine a set of actions to achieve those goals, and/or provide a detailed temporal ordering of those actions, taking into account personnel, materiel, and other constraints. This class has great commercial potential, which has been recognized. Examples involve airline scheduling of flights, personnel, and gates; manufacturing job-shop scheduling; and manufacturing process planning.

3.8.3. Configuration of Manufactured Objects from Subassemblies

Configuration, whereby a solution to a problem is synthesized from a given set of elements related by a set of constraints, is historically one of the most important of expert system applications. Configuration applications were pioneered by computer companies as a means of facilitating the manufacture of semi-custom minicomputers. The technique has found its way into use in many different industries, for example, modular home building, manufacturing, and other problems involving complex engineering design and manufacturing.

3.8.4. Financial Decision Making

The financial services industry has been a vigorous user of expert system techniques. Advisory programs have been created to assist bankers in determining whether to make loans to businesses and individuals. Insurance companies have used expert systems to assess the risk presented by the customer and to determine a price for the insurance. A typical application in the financial markets is in foreign exchange trading.

3.8.5. Knowledge Publishing

The primary function of the expert system is to deliver knowledge that is relevant to the user's problem, in the context of the user's problem. The two most widely distributed expert systems in the world are in this category. The first is an advisor which counsels a user on appropriate grammatical usage in a text. The second is a tax advisor that accompanies a tax preparation program and advises the user on tax strategy, tactics, and individual tax policy.

3.8.6. Process Monitoring and Control

Systems falling in this class analyze real-time data from physical devices with the goal of noticing anomalies, predicting trends, and controlling for both optimality and failure correction [14]. Examples of real-time systems that actively monitor processes can be found in the steel making and oil refining industries.

3.8.7. Design and Manufacturing

These systems assist in the design of physical devices and processes, ranging from high-level conceptual design of abstract entities all the way to factory floor configuration of manufacturing processes.

3.8.8. Other Applications

Since the introduction of these early experts, the range and depth of applications has broadened dramatically. Applications can now be found in almost all areas of business and government. They include such areas as [15]:

- Different types of medical diagnoses,

- Diagnosis of complex electronic and electromechanical systems,

- Diagnosis of diesel electric locomotion systems,

- Diagnosis of software development projects,

- Planning experiments in biology, chemistry compounds,

- Forecasting crop damage,

- Identification of chemical compound structures and chemical compounds,

- Location of faults in computer and communication systems,

- Scheduling of customer orders, job shop production operations, computer resources for operating systems, and various manufacturing tasks,

- Evaluation of loan applicants for lending institutions,

- Assessment of geologic structures from dip meter logs,

- Analyses of structural systems design or as a result of earthquake damage,

- The optimal configuration of components to meet given specifications for a complex system,

- Estate planning for minimal taxation and other specific goals,

- Stock and bond portfolio selection and management,

- The design of very large scale integration systems,

- Numerous military applications ranging from battlefield assessment to ocean surveillance,

- Numerous applications related to space planning and exploration,

- Numerous areas of law including civil case evolution, product liability, assault and battery, and general assistance in locating different law precedents,

- Planning curricula for students,

- Teaching students specialized tasks.

3.9. Advantages Of Expert Systems

A speed-up of human professional or semi-professional work, typically by a factor of ten and sometimes by a factor of a hundred or more.

Within companies, major internal cost savings. For small systems, savings are sometimes in the tens or hundreds of thousands of dollars; but for large systems, often in the tens of millions of dollars and as high as hundreds of millions of dollars. These cost savings are a result of quality improvement, a major motivation for employing expert system technology.

Improved quality of decision making. In some cases, the quality or correctness of decisions evaluated after the fact show a ten-fold improvement.

Preservation of scarce expertise. ESs are used to preserve scarce know-how in organizations, to capture the expertise of individuals who are retiring, and to preserve corporate know-how so that it can be widely distributed to other factories, offices or plants of the company.

Introduction of new products. A good example of a new product is a pathology advisor sold to clinical pathologists in hospitals to assist in the diagnosis of diseased tissue.

An expert system does not need to rest, sleep, etc. like a human expert and can operate the complete day and year. In many cases, having a human expert in related area is very hard but having couple of expert system in the same area is not so hard. Human experts bring their expertise information with their selves but expertise information of expert system will remain in the company because it can be easy to copy and save the information.

An expert system operates with a high performance but a human expert's performance decreases when he is ill or tired. But a well developed expert system inferences the best advices in its information and problematic area.

Human expertise needs a long time to be enough to give advices in an expertise area so it is not easy to train a human expert. But it is very easy to copy an expert system's information to another system.

4. KNOWLEDGE-BASED SYSTEMS

The basic categories of research in knowledge-based systems include: knowledge representation, knowledge use, and knowledge acquisition.

4.1. Knowledge Representation

In knowledge representation, the key topics are concepts, languages, and standards for knowledge representation. There are many issues involved in scaling up expert systems: defining the problems encountered in the pursuit of large knowledge bases; developing the infrastructure for building and sharing large knowledge bases; and actually accumulating a large body of knowledge, for example, common sense knowledge or engineering and technical knowledge.

4.1.1. Rule Based Representation

Rule based representation [16], [17] includes the following basic elements: Rules, facts, interpreters, translators, explanations and explanation functions or sub functions. Many rule based systems allow for multiple representations of rules; one representation might be for data entry, another for explanations. Explanations are usually generated by translating the rules taking part into natural language. In rule based systems, knowledge can be represented in a clear and understandable, in terms of simple and compound propositions. In addition to its memory of facts and rules, a rule based system uses a working memory to store temporary assertions. Rule based systems can use both forward and backward chaining in their interference rules of procedures.

Rule based applications have been helpful to refine and clarify existing knowledge in their domain and by such refinements to add new knowledge. They have proved invaluable as a practical means for transforming poorly understood knowledge into a coherent knowledge base. In contrast to conventional programming, rule based programming requires one to think more analytically than procedurally. Current rule based systems owe their success to a small search space of a specialized knowledge intensive system. However, rule based systems have some shortcomings. [16] Identifies the following as lacking in current rule based systems:

1- A precise analytic foundation for deciding which problems are solvable,

2- A suitable verification methodology or a technique for testing the consistency of a rule set,

3- A system of knowledge organisation that would facilitate scaling up without loss of intelligibility or performance,

4- High-grade rule compilers and specialized hardware accelerators,

5- Methods for easy integration into conventional data processing systems.

In a rule based systems, simple predicate statements are considered as `facts` and conditional statements as rules. Currently, there is not a well developed formalism to classify facts and rules in rule based systems. Many system combine rule based and frame based representation. The Dendral, meta-dendral, molgen and bacon systems can be considered as rule based systems.

4.1.2. Frame based representation

In frame representation systems [18], knowledge is represented in data structures called ``frames`` which can incorporate sets of attribute descriptions called ``slots``. A frame system provides constructs for organising classes in taxonomic structures. In this way, frames provide structured representations of objects or classes of objects. A distinguishing characteristic of frame based systems in that a frame that represents a class, can contain prototype descriptions of members of the class, as well as descriptions of the class as a whole. The description of an object type can contain a prototype description of individual objects of that type and these prototypes can be used in creating default descriptions of the objects.

Frame based systems are particularly powerful in inheritance based interface, because the taxonomic relationships among frames enable descriptive information to be shared among multiple frames by inheritance. Frame systems use the following types of inference [17]: inferred existence, inferred generic properties, inferred default values, recognition of abnormal cases and inference by analogy. These will now be described in turn.

Inferred existence: When a match is made between an entity and a frame, then the existence of the entity is inferred by the frame.

Inferred generic properties: when a match is made between an entity and a frame, then it can be inferred that the entity has all the generic properties associated with the frame.

Default properties: When a match is made between an entity and a frame, and the value for some attribute in the frame is not known, but there is a default value for the attribute, then the system can infer that the entity has this value for the attribute.

Recognition of errors and abnormal cases: When a match is made between an entity and a frame, the absence of a value for an attribute or the presence of an inappropriate value may signify an unusual about the entity or an error in the frame.

Inference by analogy: If there is an analogy between two entities e1, e2, then the attribute values of an e1 can be inferred from the values of the attributes in the slots of the frame of e2.

Frame representation systems are generally integrated with condition-action rules (production rules) to provide the necessary, effective inference mechanisms. In knowledge systems that use a large number of condition-action rules it may be difficult to understand and control the behaviour of the rules. A frame based representation facility can serve as an important component in the design of a rule based system and can provide help in rule-management by providing a means of organising and indexing rules according their intended tasks. Frames can be used to represent the rules themselves and in such cases rules can easily be grouped into classes and the description of a rule can include the attributes of the rule. Each rule can be represented as a frame with conditions, conclusions and action slots. Other slots that provide descriptions, such as rationalizations for the rule, records of usage and the tasks the rule is used for, could be included in the frame. Frames can also be used in representing qualitative processes and events.

4.1.3. Predicate Logic Based Representation

The emergence of logic programming and in particular, Prolog [19] as a predicate logic based programming language, followed by the subsequent developments have increased the importance of predicate logic as a powerful basis for symbolic representation of knowledge. Symbolic logic had been used in representing high-level concepts like infinity, continuity and causality, and also in developing various axiom systems in mathematics, physics and biology.

The adoption of Prolog by the Japanese Future Generation Computing Systems project as its core language [19] has also contributed the consolidation of this language in scientific and technical programming. Subsequent developments in metalogical programming in Prolog further increases the importance of this language. However, the predicate calculus does not have facilities for defining complex constructs such as can be defined in frame representation. The generality of the predicate calculus somewhat reduces the deductive efficiency in knowledge systems. On the other hand, with additional formalism, predicate logic can integrate rule based and frame based representations.

Reference [17] provides an interesting idea that formulae of a logical theory can be organised in such a way as to look like a frame system. Accordingly, a- all formulae which contain a particular unary predicate could be stored together, b- all formulae containing the ``is-a`` predicate could be stored in one place, c- all formulae relating to a particular entity could be stored in one place. He states that such a system would have many of the efficient implementation properties of a frame based system in addition to a well-defined semantics from formal logic.

4.1.4. Semantic Networks

Semantic networks [20], also called associative networks, are another major form of knowledge representation. There is a strong structural parallelism between the semantic network and predicate logic representation. Knowledge that can be expressed by relations and terms can be represented in semantic networks, with arcs representing relations and nodes representing terms.

Because of the structural similarities in representation, semantic networks can be formed in a logic programming language. However, there are some aspects of semantic networks, like indexing which are not readily available in predicate logic representation. An indexing scheme gives pointers from one node to another. Semantic networks provide a scheme of pointers and back pointers that facilitate accessing information easily. Nevertheless, indexing schemes can be indirectly incorporated into predicate logic based representations. Thus, a predicate logic based notation plus an appropriate indexing scheme gives almost everything in the semantic network formalism.

In semantic networks, hierarchies are used to transfer inheritance properties and relations. Examples of such hierarchies are `is a` and `instead of` hierarchies, `instance of` designates the membership of an individual to some class, while `is a` says that one class is a subclass of another. Semantic networks are used in language comprehension, pattern recognition and relational databases.

4.1.5. Knowledge representation in classifiers

Classifiers are inductive learning systems. In these systems knowledge is represented as vectors or matrices of values to a set of parameters. Classifiers are low level representation systems and therefore they may need complex input and output interfaces in translating external knowledge into internal representation and internal representation into externally readable form.

4.1.6. Knowledge representation in neural networks

Neural nets [20] are composed of a large number of interconnected processing units. Each processing unit computes a weighted sum of its input and sends its output to its output unit. Neural nets are trained by providing training examples. Learning in these systems takes place by the adjustments of the weights or strengths of the connections between the processing units. Therefore, when learning is complete, knowledge is said to be represented by to the totality of the connections. In other words, knowledge is represented by the entire network.

The difference between in the symbolic and connectivist models in the representation of knowledge. In the symbolic model, it is represented with labelled symbols, which allow symbol level rules be applied. In the connectionist model, knowledge is represented directly and affects the processing without undergoing any interpretative process.

As classifiers, neural nets are also low level representation systems and therefore need elaborate input and output interfaces for higher level representations (such as representing complex propositions and frame-like concepts) and reasoning ([such as planning, symbolic, mathematical and theoretical reasoning).

Frame based representation, classifier systems and neural networks support massive parallelism. However, parallelism is not a panacea for many symbolic computational problems, because most of the difficult tasks in knowledge rich areas cannot be highly parallelized. They argue that the time to perform a symbolic computational task often increases exponentially with its size. Parallelism may reduce search by a constant factor, whereas knowledge based approach may add a constant factor, but it may reduce the exponent.

4.2. Knowledge Use

Knowledge use, or problem-solving, research efforts involve the development of new methods for different kinds of reasoning, such as analogical reasoning, reasoning based on probability theory and decision theory, and reasoning from case examples.

The first generation of expert systems was characterized by knowledge bases that were narrow and, hence, performance that was brittle. When the boundary of a system's knowledge was traversed, the system's behaviour went from extremely competent to incompetent very quickly. To overcome such brittleness, researchers are now focusing on reasoning from models, principles and causes. Thus, the knowledge-based system will not have to know everything about an area, as it were, but can reason with a broader base of knowledge by using the models, the principles, and the causation.

4.3. Knowledge Acquisition

The quest for a large knowledge base boils down to the problem of access to distributed knowledge bases involving multiple expert systems and developers. The effort to develop the infrastructure needed to obtain access is a research area called knowledge sharing. The goal of the knowledge sharing research is to overcome the isolation of first-generation expert systems, which rarely interchanged any knowledge. Hence, the knowledge bases that were built for expert systems in the 1980s did not accumulate.

5. APPLICATION

5.1. Current Engine Monitoring Concept

The current concept of engine condition monitoring was developed in the early 1970s. Since that time engine condition monitoring (ECM) has become the cornerstone of many airlines' Flight Operational Quality Assurance (FOQA) and Continuous Airworthiness Management programs [21].

Federal aviation administration (FAA) Advisory Circular AC-120-17A ``Maintenance Control by Reliability Methods`` lists the three primary maintenance processes as hard time, On-Condition and condition monitoring [22].

FAA AC-120-17A further states that Condition Monitoring ``is accomplished by appropriate means, and enables an operator to find and solve problem areas.`` [22]. Simply, to put engine parameter trend analysis is one aspect of engine condition monitoring. It is the technique of looking at measured or inferentially calculated data for changes from the normal operation condition.

A successful engine trend monitoring program has been composed of three fundamental functions; Data Collection, Performance Plotting, and the Analysis [23].

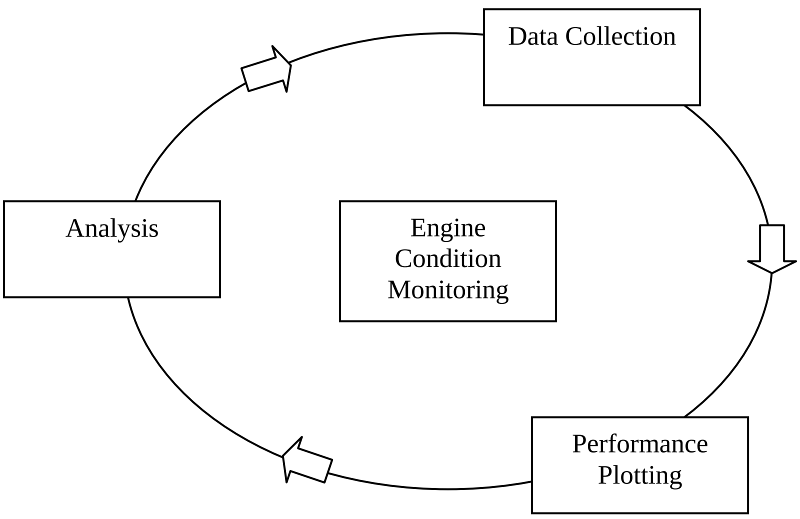


Figure 5.1: Engine condition monitoring

Data Collection : Data must be accurately and regularly recorded during flight

Performance Plotting : Changes in Altitude, Outside Air Temperature, Indicated Air Speed, and different power settings are a few of the factors that can affect the engine's performance from day to day. To realize and control the effects on these factors collected data needs to be processed through analytical software.

Analysis : An accurate and timely interpretation of the trend monitoring graphs is essential to the success of a trend monitoring program. Upon completion of the analysis, adverse results should be addressed and acted accordingly.

5.2. Engine Condition Monitoring Process

Data collection is done in sequence. First of all flight data is acquired and recorded. Flight data includes data during takeoff, climb and cruise. It is very important to acquire and record the good quality data.

Then, acquired and recorded flight data is transmitted to ground systems. Then those are applied to System for the Analysis of Gas turbine Engines (SAGE) software. SAGE software converts the data in sensible values with attached formulas which are developed by the engine manufacturer. Output values vary in described boundaries.

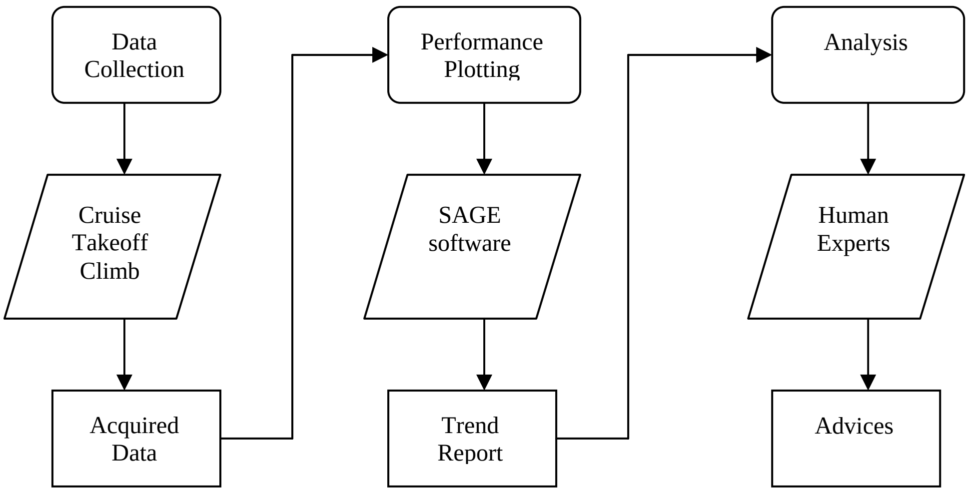


Figure 5.2: Engine condition monitoring process

SAGE software examines the data and reports as analysis of monitoring data. Now, this report is meaningful for human experts who are experienced in interpreting in engine condition monitoring data. Human experts consider the report according to their experiments and instructions of engine manufacturer. They look for a certain trend shift in related parameters. Levels of parameters are also very important for them to decide. They can consider the results with other human experts.

After human experts finished interpreting SAGE reports, they decide what they will do then. They can ask for additional inspections. They can prepare an engine maintenance schedule. They can also put the engine to the watch list [23].

When interpretation of SAGE reports are performed properly by qualified and experienced personnel, trend monitoring is capable of detecting faults such as; hot section deterioration, hot starts, faulty fuel nozzles, dirty or eroded compressors, FOD damage, bleed leaks, instrumentation errors.

5.3. Reasons for Performing Engine Trend Monitoring

Effective engine trend monitoring process assists in managing and forecasting engine maintenance [23]. Condition monitoring is an important aspect of safe engine operation and effective maintenance practises in assessing engine performance and health, reducing unexpected problems and unscheduled maintenance by detecting abnormal trend shifts or levels of concern and monitoring performance trends.

Software available monitoring applications and acquiring good quality monitoring data are strongly recommended by engine manufacturers.

5.4. Concept of Engine Trend Monitoring

Current ``health`` of engines must be monitored. Engine condition monitoring data should be collected on a required routine and continuing basis. Current engine ``health`` and trend data should be easy to observe. Monitoring process should be relatively easy to implement.

Engine performance and hardware condition should be evaluated both. Evaluation can be based on assessment of engine trends (without engine disassembly).

It is very important to provide early detection of potential problems. Basic engine and instrumentation faults have identifiable trend characteristics. These can be easily detected [23].

Only the available data should be considered to monitor engine ``health``. Suitable data can be recorded during revenue service operation. Data can be recorded manually by the flight crew, or automatically by means of on-board data acquisition systems.

Overall performance and mechanical parameters which are available with standard instrumentation should be monitored. Data must be normalized to detect abnormal shifts and trends. Aircraft operating conditions vary significantly because of too much scattering.

Engine trend monitoring software typically used and computer programs available from engine manufacturers

5.5. Expert System Application

5.5.1. Problematic Area and Applications

Interpreting of engine condition monitoring data is a very important issue to decide correct maintenance actions in aircraft engine maintenance departments.

The data which shows engine condition is collected in some special parts of a flight. This data is acquired in takeoff, climb and cruise and recorded in every flight. Acquired data is applied to SAGE software which converts the data to a graphical report of varying in limited values. SAGE software provides this graphical report with its integrated formulas.

These graphical reports are considered by human experts. Human experts use their expertise in the engines and technical issues of engine manufacturer. Interpreting of engine condition monitoring data is such a complex procedure and totally dependent to human experts. Wrong interpretations or incorrect decisions which can be born of workload, illness, tiredness or carelessness will result with bigger breakdowns. Aircraft engines are one of the most valuable and vital part of an aircraft and so on, it is very important to determine and fix the faults and to schedule maintenances.

There is another problem in maintenance departments of companies. It is very clear that the most valuable thing of a maintenance department is the expertise information of its human experts. Generally, human experts can not explain their expertise as a document. This will be a handicap on transferring valuable expertise information to the company’s know-how. The result is highly dependency to the human experts at interpreting of engine condition monitoring data. If this happens, work plans can not be planned effectively and work load can be more sometimes. In those times, human experts may not examine the reports enough and this can be a problem on their inferences.

Additionally, to educate new human experts can be very hard if the expertise information does not exist enough in a maintenance department. And if a human expert resigns, his expertise will go with him.

The expert system interprets the SAGE software reports. This expert system operates as a human expert and follows the same steps with the human expert for interpreting.

5.5.2. The way of thinking of a human expert : Analysis of engine trends

Analysis related parameters in SAGE reports should be approached with two main concepts. The first one is to pay attention on a trend step-shift beyond pre-set values. If there is a trend shift, human expert conducts a maintenance action within a pre-determined time frame. The second one is checking a more gradual change in the values. If there is a more gradual change, human expert alerts maintenance department to conduct maintenance at the next convenient time and/or put the engine in an ``on-watch`` status.

To make a trend interpretation human expert reviews trend plots in timely manner and looks for a trend shift. Shift means that a deviation from most recent smoothed trend, a shift can also be detected by examination of individual ``raw`` points. If

there is a trend shift human expert investigate further. If there is no shift change, no action is taken.

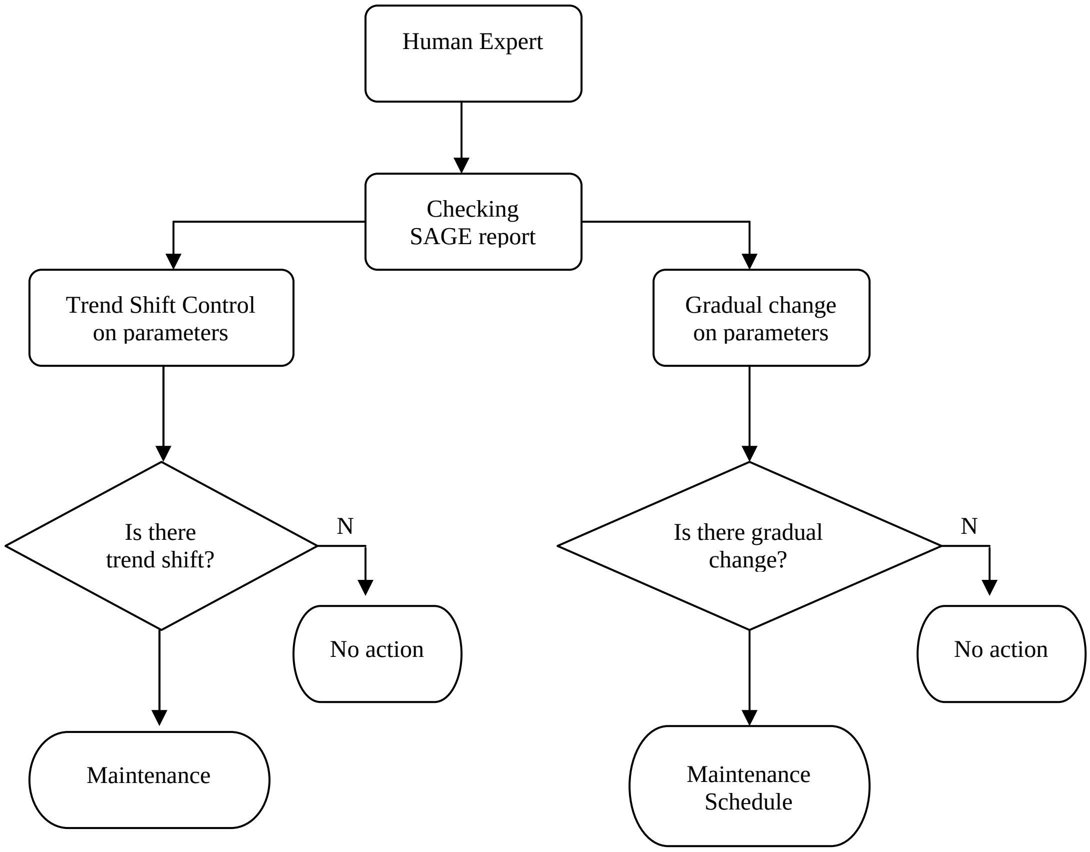


Figure 5.3: Human expert's thinking way

For further investigation, human expert checks other engines on aircraft. If all engines have similar shifts at same time, the problem induces by aircraft. So, there is a problem on measurement instrumentation or input is error. If all engines have not similar shifts at same time, problem induced by engine or engine instrumentation.

When the problem appears to be due to engine or engine instrumentation, human expert looks to the consistency of parameter shifts. If the parameter shifts consistent, the problem is in instrumentation or in engine control, or hardware failure or performance deterioration. If the parameter shifts inconsistent, the problem is in instrumentation or there is a input error.

5.5.3. Operation Principle of Expert System

The main parameters in technical sheets of aircraft engine which are published by the manufacturer of the engine are exhaust gas temperature (EGT), fuel flow (FF) and core speed (N2) values.

These technical sheets were used during the development of this expert system. A human experts working and thinking principle are mentioned above parts. Human experts thinking and working principles are applied in expert system as a operation principle. During developing of expert system, a human expert who is working in an aircraft maintenance department was helped in some critical concepts. These critical concepts are considering the EGT, FF and N2 parameters in the practical SAGE reports with the theoretical technical sheets of the aircraft engine manufacturer.

Before passing to the interpreting of EGT, FF and N2 parameters, it will be very helpful to explain the meaning of these parameters. EGT is the abbreviation of exhaust gas temperature, FF is the abbreviation of fuel flow and N2 is the initial of engine core speed.

Like a human expert, expert system checks EGT, FF and N2 parameters for its inference process. These parameters should be entered in a text box as seen in the SAGE report. The text box is the expert system’s input. When the user starts the expert system, expert system reads these EGT, FF and N2 values in the text box. Expert system calculates the amount of trend shift which is called delta (D) and decides if the trend shift’s direction is up or down. Measurements and directions are monitored in the system.

The engine manufacturer defined some causes that can be reason of some delta values and directions of trend shifts. Each parameter’s trend shift direction and delta value is affected from these causes. These are integrated in the expert system’s knowledge base. After measuring the trend shift amounts and deciding their directions, results are compared with the integrated results for reaching correct causes. Expert system can monitor all these causes without any deficiency. So, the user does not need to check some datasheets or technical sheets which are in separated sources.

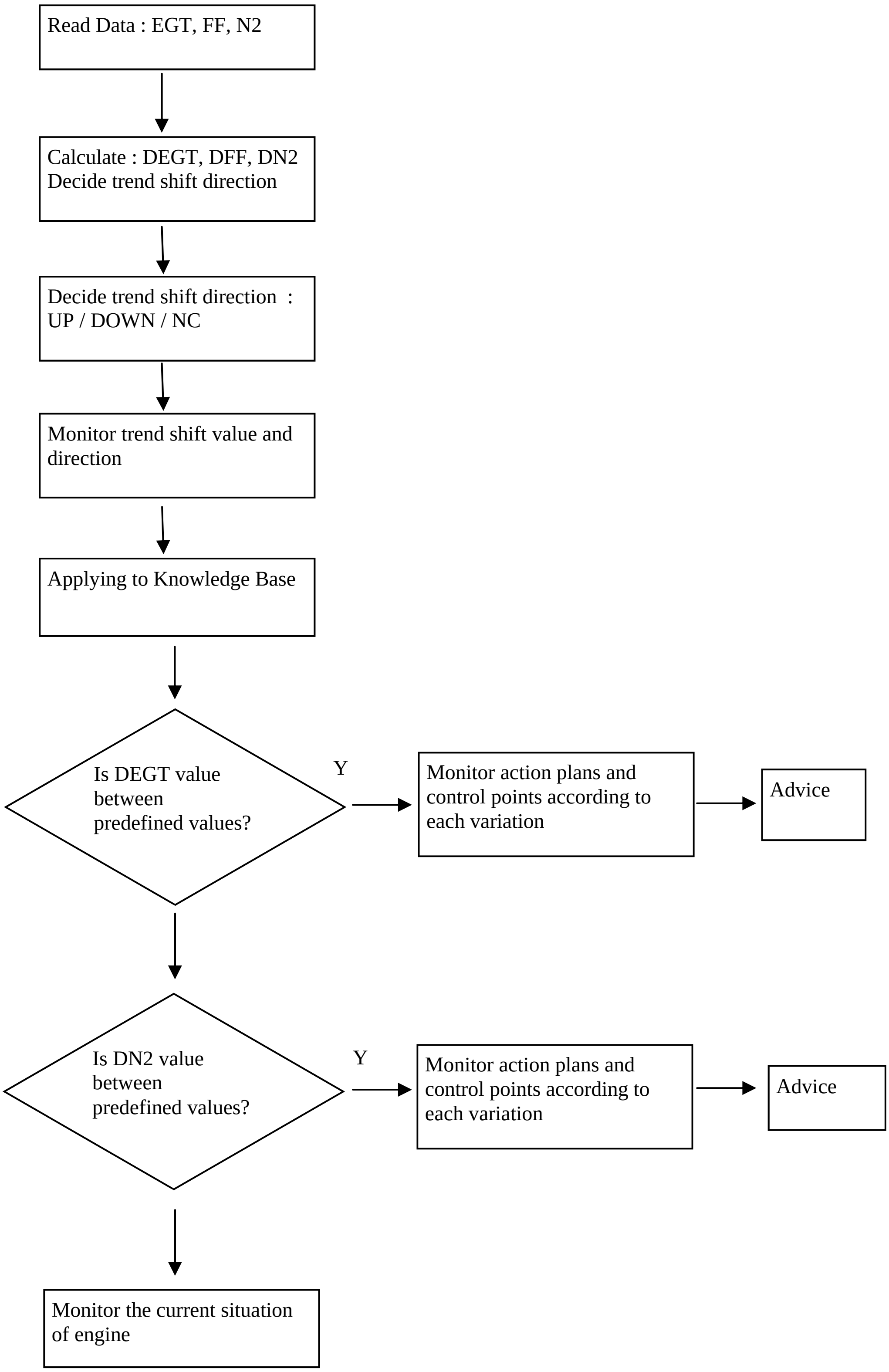


Figure 5.4: Operation Principle of Expert System

Then, expert system checks the delta EGT (DEGT) value. Engine manufacturer defined critical points for interpreting DEGT value and its direction. These critical points are;

DEGT is more than 10°C but less than 20°C in UP direction,

DEGT is equal or more than 20°C in UP direction,

DEGT direction is more than 10°C in DOWN direction.

For each situation, there are special action which should be taken and control points which should be checked. All of these are monitored as an output of expert system.

In next step of interpreting EGT parameter, expert system monitors all action plans for EGT, FF and N2 direction changes as UP, DOWN or NOT CHANGE. Users can now easily see all necessary actions which they should be done in the maintenance.

Delta FF (DFF) is used for a confirmation parameter of EGT trend shift in normal application but it is not considered in the expert system. However, it could be easily integrated in the software.

Delta N2 (DN2) is a also an important parameter which is always under control for interpreting engine condition monitoring data. The engine manufacturer declares in their technical documents that if the DEGT is less than 10°C in UP direction or less than 10°C in DOWN direction, DN2 should be considered mainly. Engine manufacturer defined critical points for interpreting DN2 value and its direction. These critical points are;

DN2 is equal or less than 0.5% in UP direction, or equal or less than 0.5% in DOWN direction,

DN2 is more than 0.5% in UP or DOWN direction but between the maximum and minimum guidelines,

DN2 is less than minimum guideline,

DN2 is more than maximum guideline.

For all above conditions there are some actions which should be taken and points which should be checked. All these information will be monitored by the expert system.

In next step of interpreting N2 parameter, expert system monitors all action plans for EGT, FF and N2 direction changes as UP, DOWN or NOT CHANGE. Users can now easily see all necessary actions which they should be done in the maintenance.

DEGT and DN2 values and explanations these values are also monitored in the main screen.

5.5.4. Knowledge Representation of Expert System

Rule based knowledge representation is used in the expert system. In these knowledge representation method; facts, rules and explanations are used. Knowledge is represented as simple and compound propositions and condition-action rules. The knowledge can be represented by a set of production rules. The rules have the following structure.

IF [antecedent]   THEN [consequent]

The main purpose of the inference engine is to link the rules given in the knowledge base and the associated condition input by the user with data given in the database in order to solve the problem.

DEGT, DFF and DN2 are calculated from the entered data sets of EGT, FF and N2. After calculation of DEGT, DFF and DN2, they are applied to the rules in the knowledge base. This principle is the expert system's inference mechanism.

All facts and rules are called from a table in the knowledge base and the inference in other words results are monitored to the main screen. Explanations are generated by translating the rules taking part in the decision into natural language. In addition to its memory of facts and rules, rule based system uses a working memory [or dynamic memory] to store temporary assertions. Rule based systems have been helpful to refine and clarify existing knowledge in their domain. They have proved invaluable as a practical means for transforming poorly understood knowledge into a coherent knowledge base.

5.5.5. Software Development Tool and Technique

The expert system is developed in Borland C++ Builder 5.0 programming language. Developed system firstly reads the data sets which are entered by user in three different array then measures the DEGT, DFF and DN2. Measured values are applied on rules in the knowledge base to inference results to advice user.

5.5.6. Expected Benefits of the Expert System

Usage of the expert system in explained application area will decrease the human expert's workload. By this way; human experts will not make mistakes in interpreting of engine condition monitoring data which born of high workload. Some basic interpreting jobs can be done by the expert system so human experts will have more enough time to investigate complex SAGE reports. This will increase the human expert's expertise and knowledge also.

Expert system can be used if the human expert is not available or absent at the moment. Dependency to a human expert can be decreased in many situations at a maintenance department.

The expert system can be easily used for education of new human experts effectively.

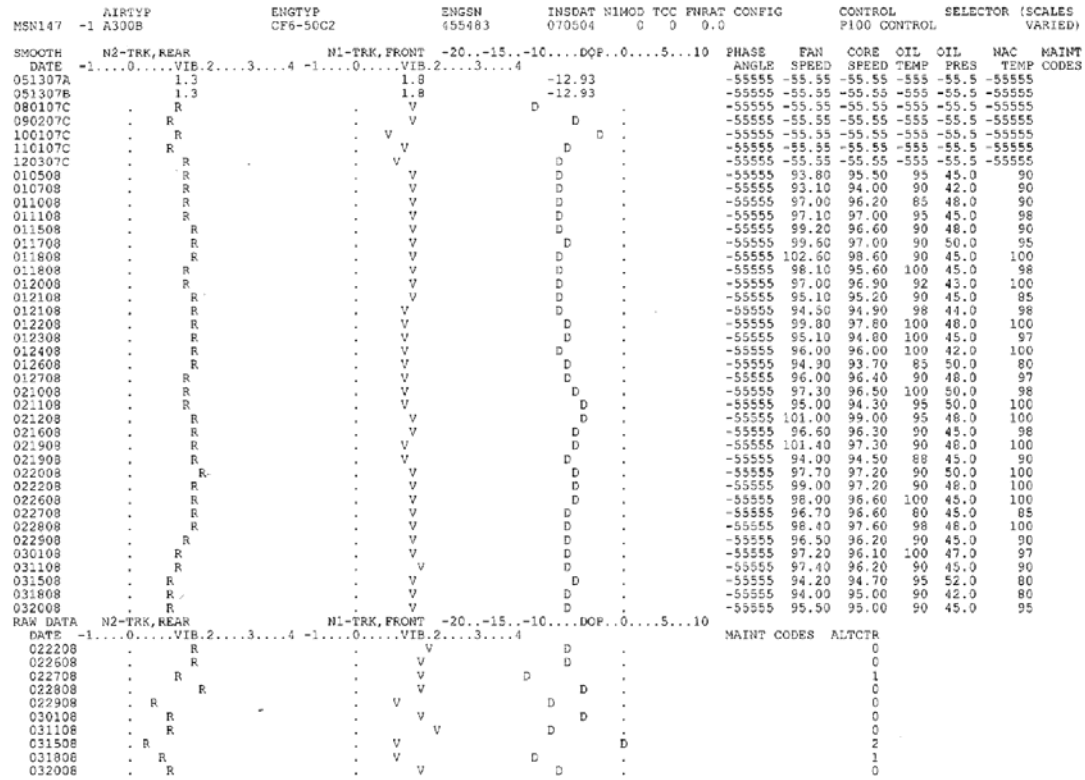
Usage of the expert can prevent interpreting mistakes causes by humanitarian effects like carelessness, illness etc. This will also prevents high maintenance costs causing by wrong action plans which caused by interpretation mistakes.

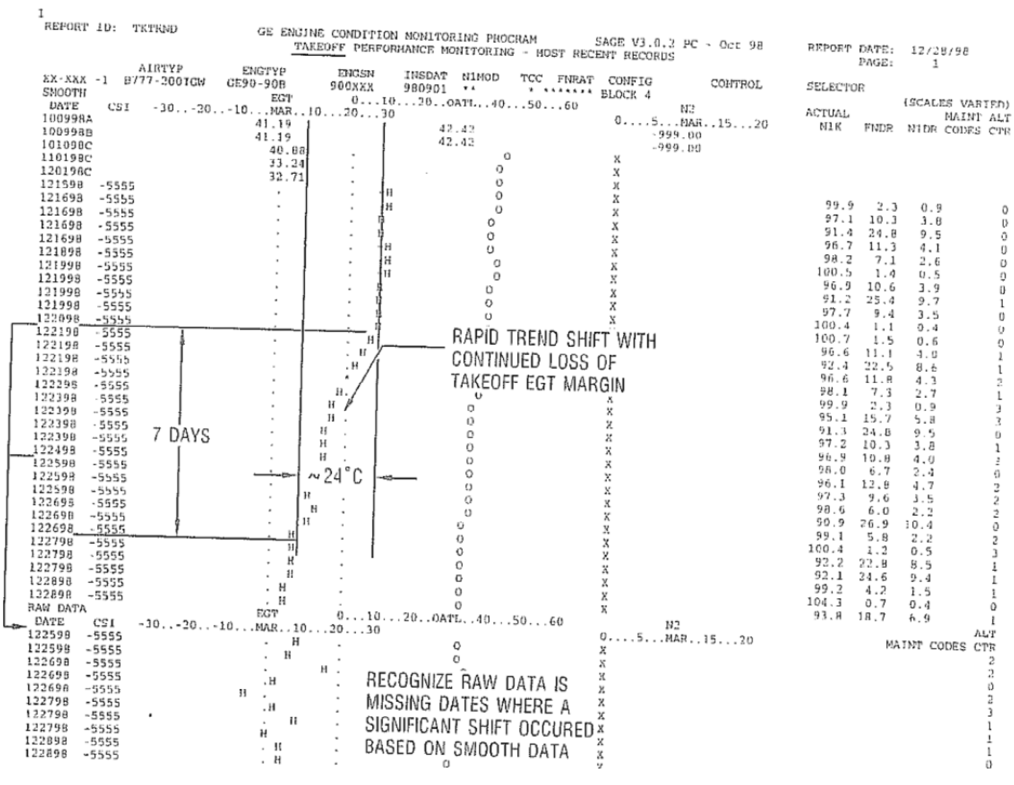
Additionally, human expertise will be transferred to the maintenance department's knowledge base.

5.5.7. Deficiencies of the Expert System

First of all, this expert system considers the user entered data totally true. It could not realize the insensible data values in the data sets. Human experts can easily notice the insensible data and they do not consider this data during interpretation. DFF value is used for a confirmation parameter by human experts in usage of DEGT values. This expert system does not use DFF value during interpretation processes. But these can be easily integrated in the software. Design allows much kind of supplementary features.

Appendix 1 – Examples of SAGE reports





Appendix 2 – Explanations of parameter shifts

IV. TREND ANALYSIS (Continued)

C. Typical Causes of Cruise Trend Parameter Shift

PARAMETER SHIFT			PROBABLE CAUSES
DEGT	DFF	DN2	(DEGT = Delta EGT, DFF = Delta Fuel Flow, etc.)
UP	UP	UP	1. High-Pressure Compressor Damage 2. Excessive/Abnormal Air Bleed or Leakage * 3. VBV Tracking Shift (OPEN) 4. Low-Pressure Turbine Deterioration/Damage 5. Combustor Deterioration 6. N1 Indication Error (Indicates Low)
UP	UP	DN	1. High-Pressure Compressor Efficiency Loss 2. High-Pressure Turbine Efficiency Loss/Damage 3. Combustor Deterioration (Fishmouth Seals) * 4. VSV Tracking Shift OPEN
DN or NC	DN or NC	UP	* 1. VSV Tracking Shift CLOSED 2. N2 Indication Error (High) If no change in EGT and Fuel Flow.
UP or NC	UP or NC	DN	* 1. VSV Tracking Shift OPEN 2. N2 Indication Error (Low) If no change in EGT and Fuel Flow.
DN	DN	DN	1. N1 Indication Error (Indicates High)
NC	UP	NC	1. Fuel Flow (Wf) Indication Error (High)
NC	DN	NC	1. Fuel Flow Indication Error (Low)
UP	NC	NC	1. EGT Indication Error (High)
DN	NC	NC	1. EGT Indication Error (Low)
DN	UP	DN	1. TAT Indication Error (High)
UP	DN	UP	2. TAT Indication Error (Low)

* Not Applicable to FADEC Engines.

Appendix 2 – Explanations of parameter shifts

IV. TREND ANALYSIS (Continued)

D. Trend Monitoring Axioms

Several axioms can be applied to Trend Monitoring Analysis that are helpful in determining whether trend shift is valid. If the data measurement is accurate, the following axioms are usually always valid:

1. EGT Shift Upward (Over 15°) will be confirmed by an upward shift in Fuel Flow Trend.

2. EGT Downward Shift (Over 15°C) will be confirmed by a downward shift in Fuel Flow Trend.

* 3. VBV Tracking Shift OPEN will cause N2 to trend higher.

NOTE: VBV Tracking CLOSED may not produce an N2 shift because, in cruise, VBVs are normally near FULL CLOSED.

* 4. VSV Tracking Shift CLOSED will cause N2 to trend higher.

* 5. VSV Tracking Shift OPEN will cause N2 to trend lower.

* 6. Significant Downshift in N2 trend (over 1%) is usually caused by a VSV OPEN shift.

NOTE: Compressor damage usually causes N2 to trend higher. Deterioration that causes loss of HPC or HPT efficiency will usually cause N2 down-shift no more than 1.0 %.

* Not applicable to FADEC engines. VSV or VBV tracking shift will be reported as a fault by the Electronic Control Unit.

E. Maintenance Actions Based On Trend Monitoring

(1) EGT Trend Shift	ACTION
UP SHIFT more than 10°C but less that 20°C	1. Check for indication of bird strike or FOD at Inlet and Exhaust. 2. Check Last Stage LPT Blades. 3. Place engine On-Watch for next three flights. If average shift is greater than +10°C, perform troubleshooting listed below.
UP SHIFT 20°C or more	Perform troubleshooting prior to next flight.

Appendix 2 – Explanations of parameter shifts

IV. E. 1. (Continued)

EGT Trend Shift	ACTION
DOWN SHIFT more than 10°C	Place On-Watch for next 3 flights. If downshift averages 20°C or more, troubleshoot indication system.

EGT TREND INDICATION			ACTION
DEGT	DFF	DN2	
UP	UP	UP	1. BSI HPC (Suggest Stgs. 1,7,12) 2. BSI LPT (First and Last Stage) * 3. Check VBV System/Rig (Except CF6-6) 4. BSI Combustor and Stg. 1 HPT Nozzle. 5. Check for excessive air bleed or leaks. 6. Check N1 indication system.
UP	UP	DN	1. BSI HPT Stg 10 Air Seal (CF6-50 only). 2. BSI HPT Stg 1 and 2 for damage. 3. BSI Combustor and Stg 1 HPT Nozzle. 4. Check for mini-nozzle failure (CF6-50 only) 5. Coke clean engine.
UP	NC	NC	1. Check EGT Indication System. 2. BSI HPC and HPT 3. BSI HPT Stg 10 Air Seal (CF6-50 Only) 4. Coke clean engine.
DN	NC	NC	- Check EGT Indication System.
DN	UP	DN	- Check TAT Indication (Indicates High).
UP	DN	UP	- Check TAT Indication (Indicates Low).

* Not Applicable to FADEC engines.

(2) Fuel Flow Trend Shift

No maintenance actions should be performed based only on Fuel Flow (Wf) trend shifts except for obvious indication system problems.

Fuel Flow trend should be used as a confirming indicator for troubleshooting EGT trend shifts.

Appendix 2 – Explanations of parameter shifts

REP TIP NO. 373, CF6-6/-50/-45/-80A/80C2 GUIDELINES FOR PARAMETER TREND MONITORING
PAGE 7 of 18 — 29 August 1989, Rev. 1, 17 January 1990

IV. E. (Continued)

(3) N2 Trend Shift

NOTE: The following applies if the EGT shift is less than 10°C. If the EGT shift is more than 10°C, refer to the EGT Guide.

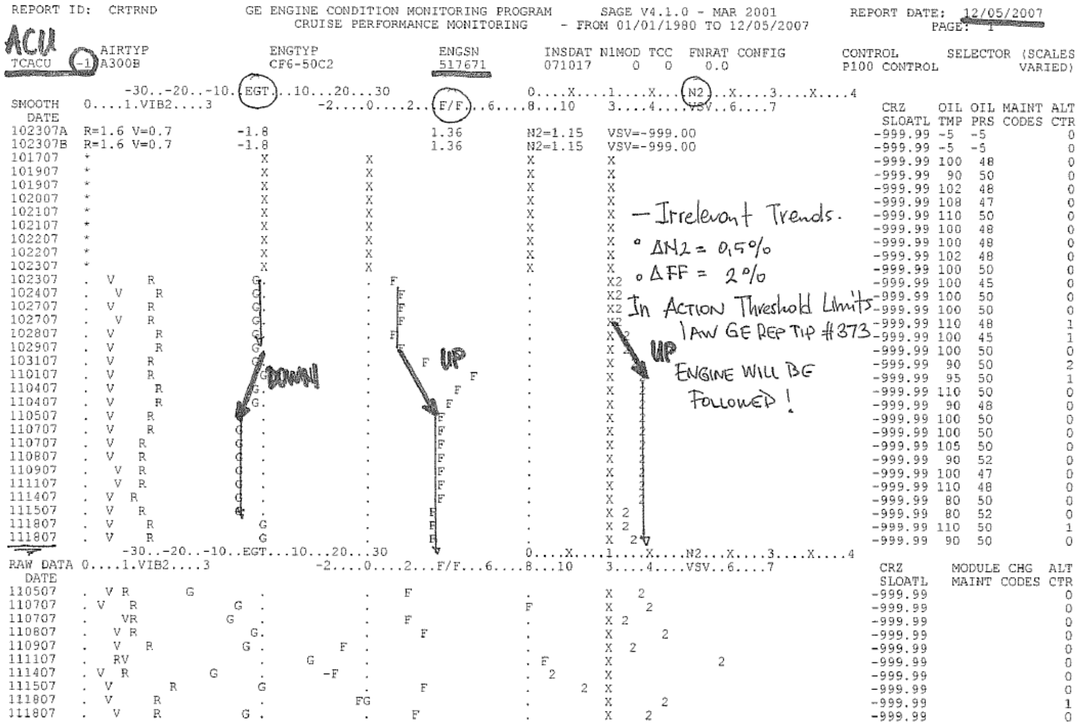
	ACTION
- UP Shift or DOWN Shift 0.5% or less.	- No action recommended.
- UP shift or DOWN shift more than 0.5% but within Min-Max guideline band.	- Place On-Watch for at least three flights. If shift is confirmed, schedule trouble-shooting at Operator's convenience.
- N2 Trend below Minimum guideline.	- Perform troubleshooting prior to next flight.
- N2 Trend above Maximum guideline.	- Place On-Watch for next 3 flights. If N2 is still trending above Max guideline, schedule troubleshooting at the Operator's convenience.

TREND INDICATION			ACTION
DEGT	DFF	DN2	
UP or NC	UP or NC	DN	* 1. VSV Tracking shift OPEN, Check Rig 2. BSI HPT Stg 10 airseal (-50 only) 3. Check CIT Sensor for Cold shift. 4. Coke-clean engine.
DN or NC	DN or NC	UP	* 1. VSV Tracking shift CLOSED, Check Rig 2. Check CIT Sensor for Warm Shift or plugged Sensor Fuel Line Screens.

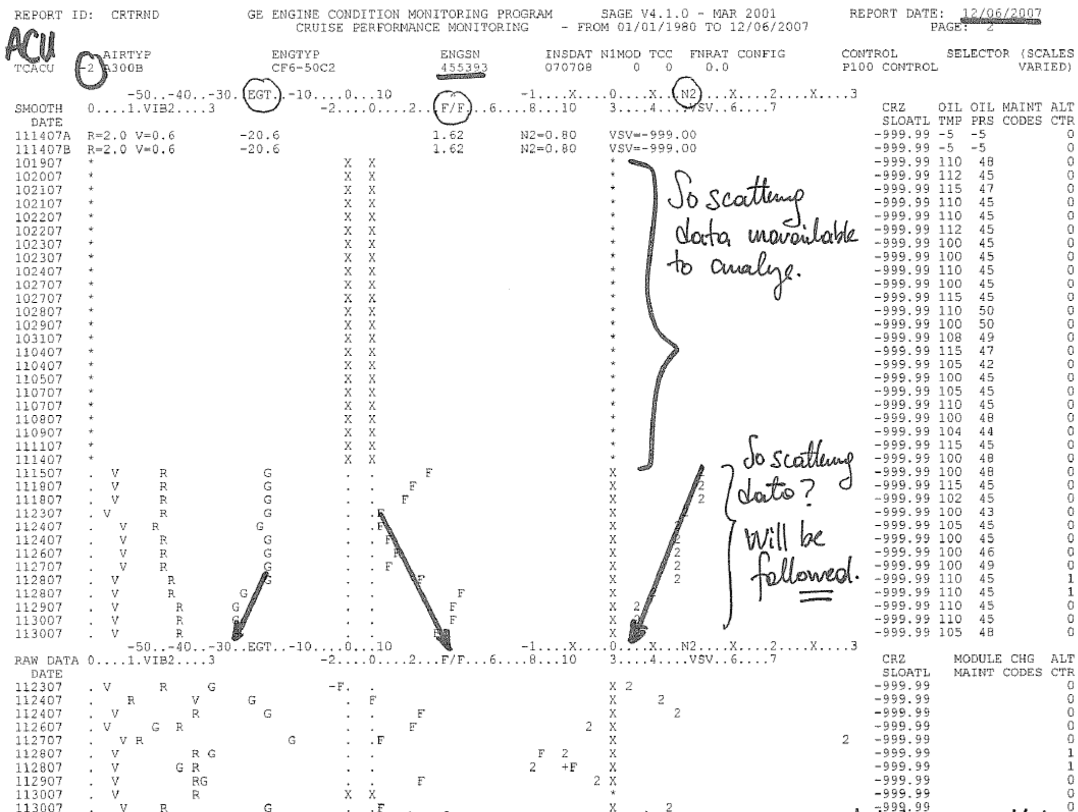
NOTE: N2 trend Min-Max guideline for each engine model and application is included in guideline limits of Appendix I.

* Not applicable to FADEC engines.

Appendix 3 – Some examples of human experts decisions

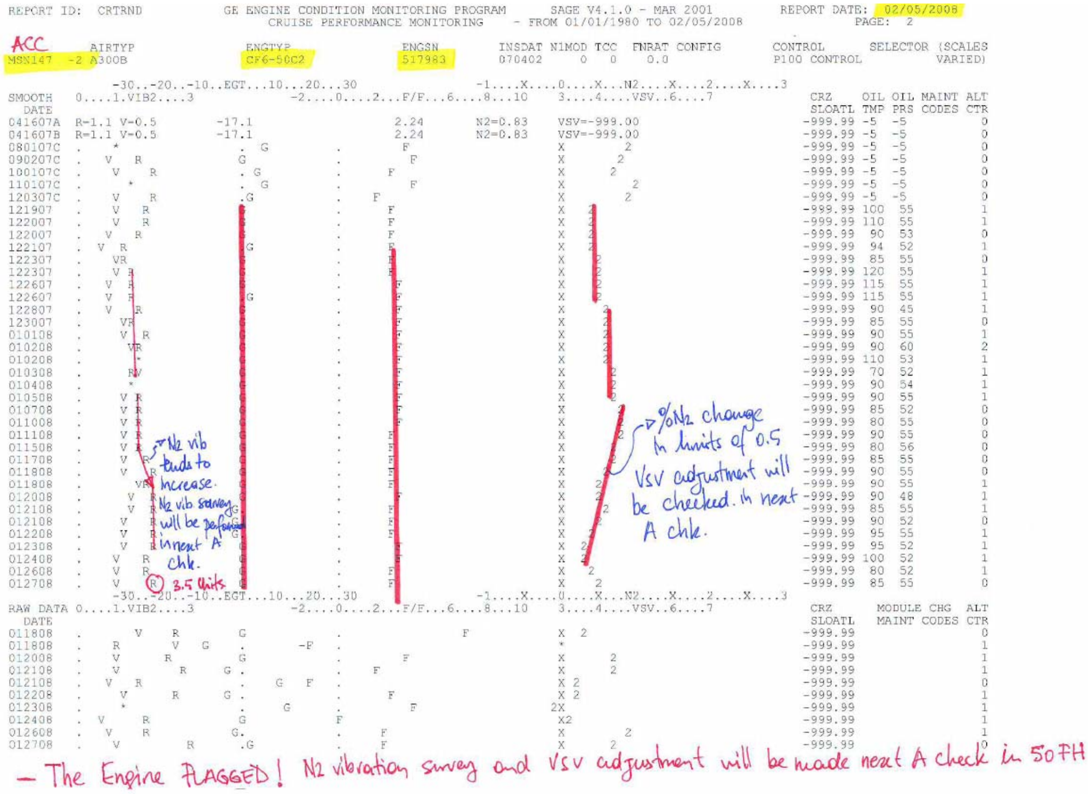


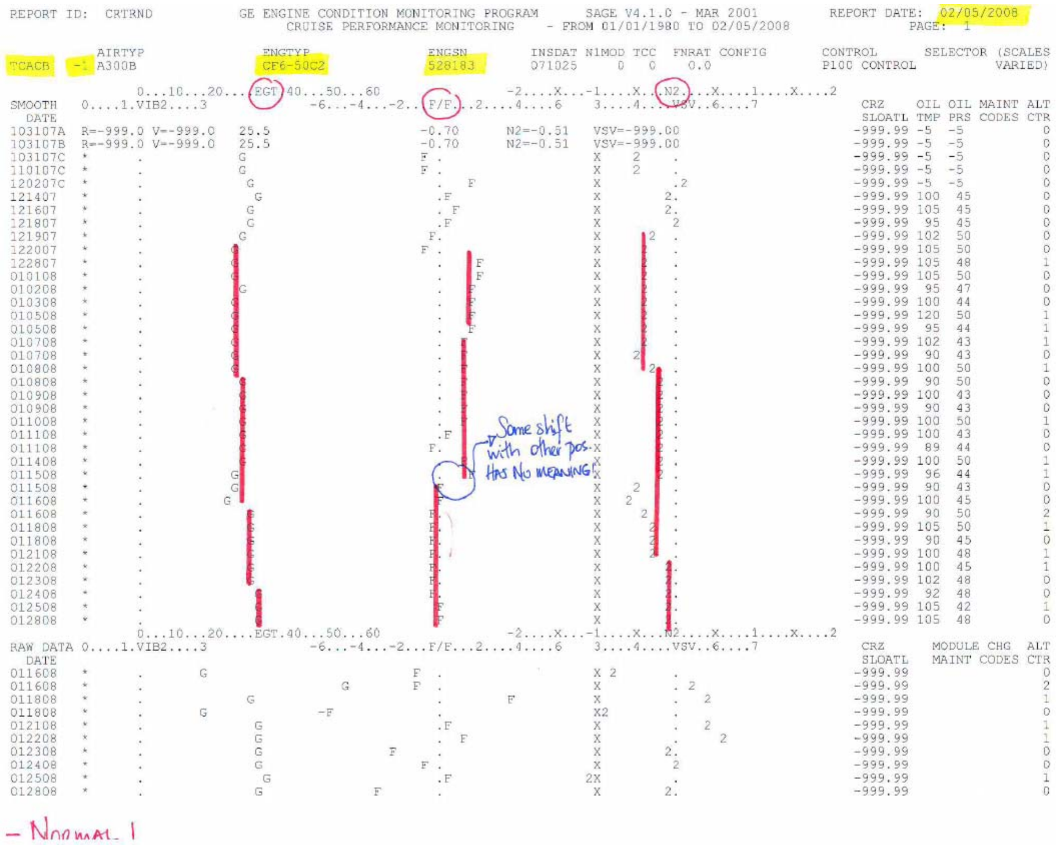
– Engine removed from TC-ACU @ 19/11/07



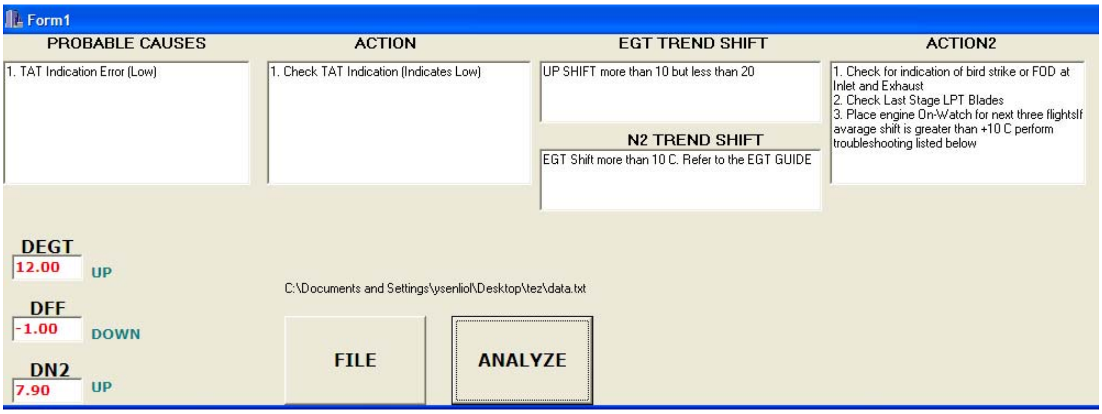
– Trend has no meaning. Only TAT Indication error would be possible

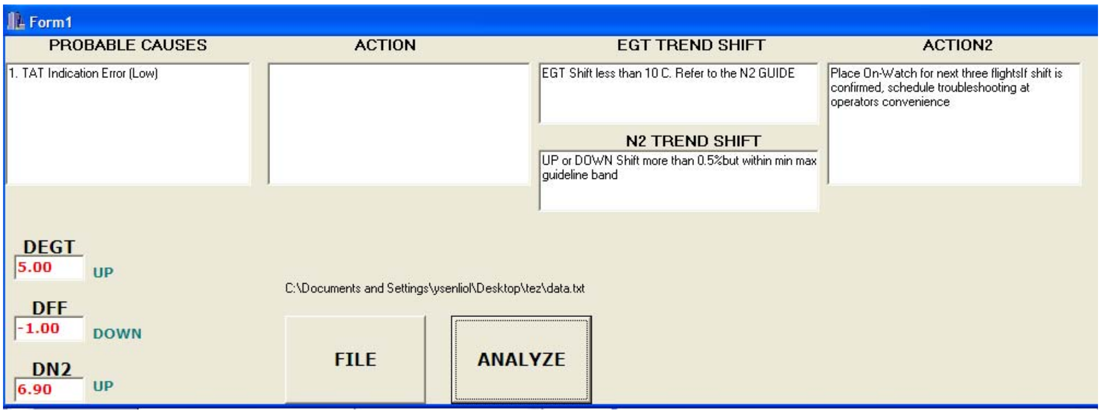
Appendix 3 – Some examples of human experts decisions

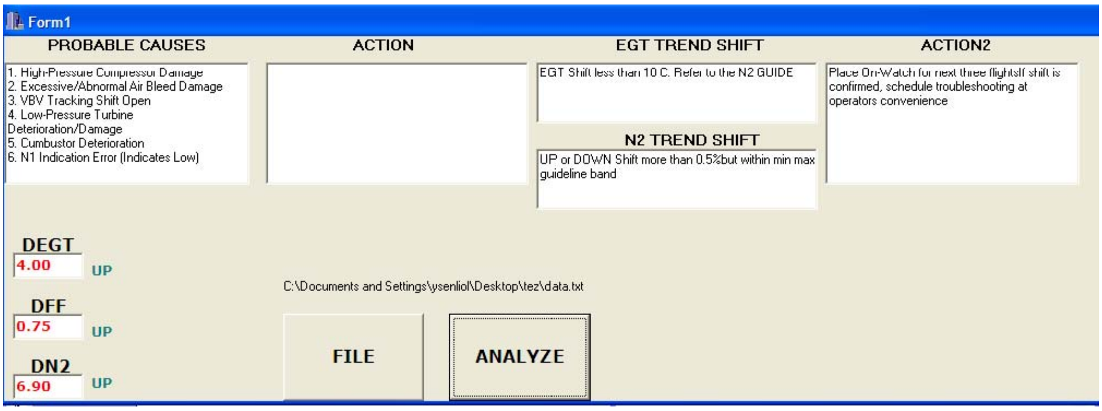




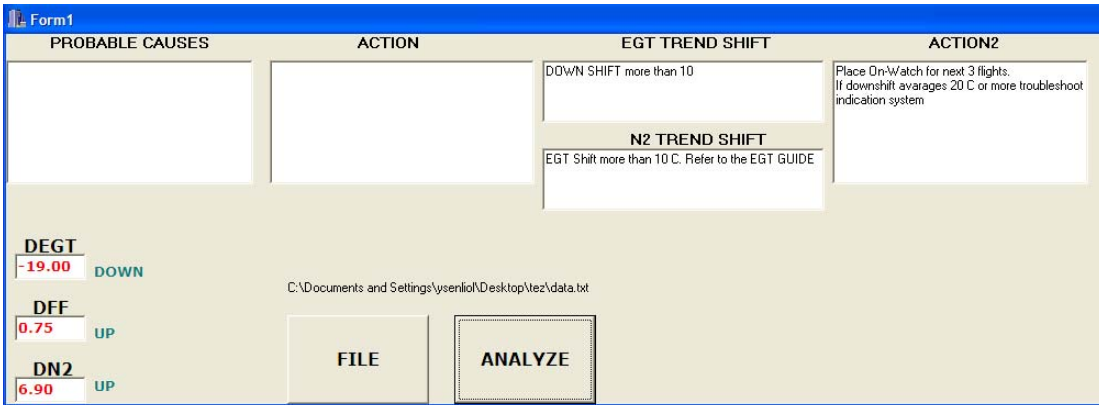
Appendix 4 – Some examples of expert system interpretations

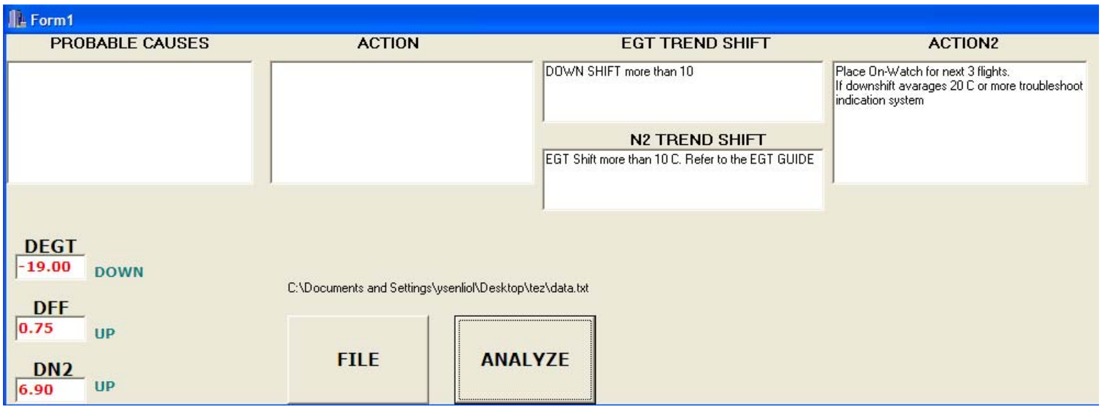


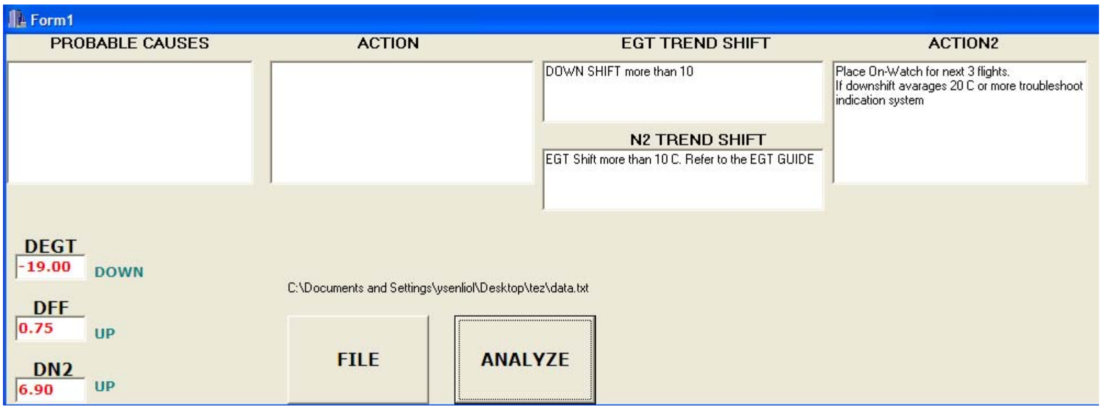




Appendix 4 – Some examples of expert system interpretations







REFERENCES

[1] **Firebaugh, M.W.**, 1988. Artificial Intelligence : A Knowledge Base Approach, Boyd & Fraser, Boston

[2] **Derek, P.D.**, 1991. A New Guide to Artificial Intelligence, Ablex Pub. Corp., Norwood, N.J.

[3] **IITA Task Group**, 1994. Information Infrastructure Technology and Applications. Office of Science and Technology Policy, Washington D.C.

[4] **Blake, D.V. and Uttley. A.M.,** 1959. AAAI 2005 Topics, American Association for Artificial Intelligence, CA.

[5] **Newell, A. and Simon, H. A.,** 1972.Human problem solving, Prentice Hall, Englewood Cliffs, NJ.

[6] **Feigenbaum, E.A. and Feldman, J.,** 1995. Computers and Thought, AAAI Press, Cambridge.

[7] **Minsky, M.,** 1968. Semantic Information Processing, MA:MIT Press, Cambridge.

[8] **McCarty, J.,** 1959. Programs With Common Sense, *Mechanisation of Thought Processes, Proceedings of the Teddington Conference on the Mechanization of Thought Processes,* pp. 77-84, Her Majesty's Stationery Office, London

[9] **Lindsay, R.K. and Buchanan, B.G. and Feigenbaum E.A. and Lederberg, J.,** 1980. Applications of Artificial Intelligence for Organic Chemistry: The Dendral Project, McGraw-Hill Book Company, N.Y.

[10] **Lenat, D.B.**, 1984. Why AM and EURISKO Appear to Work, *Artificial Intelligence,* **23,** 269-294.

[11] **Persaye, K. and Chignell, M.**, 1988. Expert Systems for Experts, pp. 29-60, John Wiley & Sons, New York, USA.

[12] **Wick, M.R. and Slagle, J.R.,** 1989. An Explanation Facility for Today's Expert Systems, *IEEE Expert,* **4,** 26-36.

[13] **Holden, A.D.C.,** 1976. Trends in Artificial Intelligence, *IEEE Transactions on Computers,* **C-25,** 313-316.

[14] **Shirley, R.,** 1987. Some Lessons Learned Using Expert Systems For Process Control, *IEEE Control Systems Magazine,* **7,** 11-15.

[15] **Girgis, A.A. and Johns, M.B.,** 1989. A Hybrid Expert System for Faulted Section Identification, Fault Type Classification and Selection of Fault Location Algorithms, *IEEE Power Engineering Review,* **9,** 56-57.

[16] **Hayes-Roth, B.**, 1985. A Blackboard Architecture for Control, *Artificial Intelligence,* **26,** 251-321.

[17] **Frost, R.A.,** 1986. Introduction to Knowledge Base Systems, William Collins Sons and Co., U.K.

[18] **Fikes, R. and Kehler, T.**, 1985. The Role of Frame Based Representation in Reasoning, *ACM,* **28,** 904-920.

[19] **Kowalski, R.A.**, 1984. Software Engineering and Artificial Intelligence in New Generation Computing, *Future Generation Computer Systems,* **1,** 39-49.

[20] **Nabiyev, V.V.,** 2003. Yapay Zeka, Seckin Yayincilik, Ankara.

[21] http://www.flightsafety.org/foqa.html

[22] http://www.faa.gov/regulations_policies/

[23] **Ataç, U.,** 2008. Personal Interview.

RESUME

Yalcin Senliol was born in Kastamonu on the 7th of November, 1979. He obtained his B.Sc. degree from Yildiz Technical University, Department of Electronics and Communication Engineering in 2003. After one year, he attended M.Sc in Aeronautical and Astronautical Engineering at Institute of Science and Technology of Istanbul Technical University, in 2004. He has been working in AVS Technology which is a project company in telecommunication sector as a sales engineer since 2007.