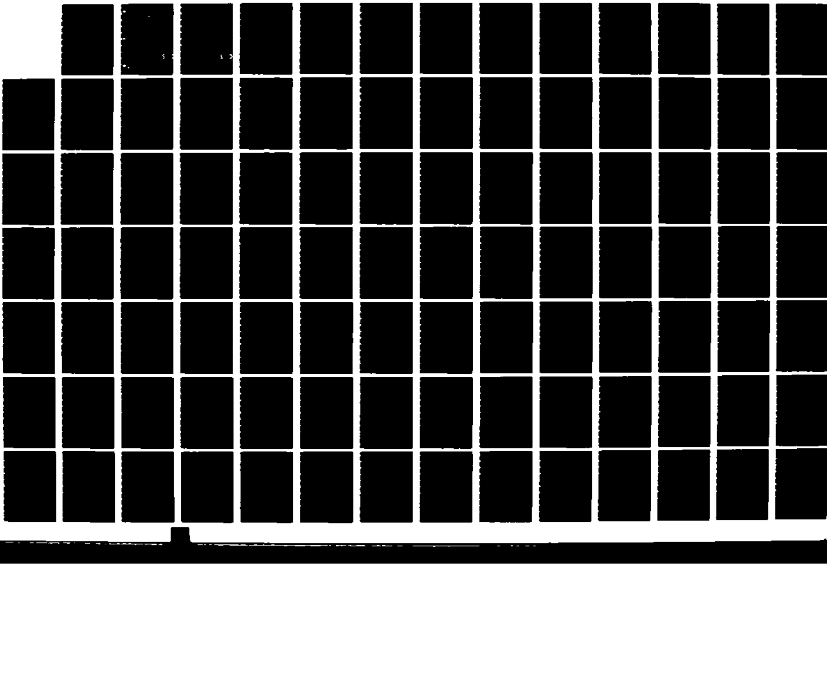
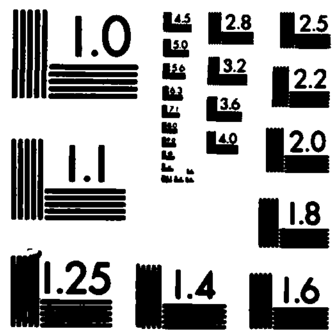
AD-A138 373 AIRCRAFT MAINTENANCE EXPERT SYSTEMS(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING G R FERGUSON NOV 83 AFIT/GCS/EE/83D-9 1/2

UNCLASSIFIED F/G 1/3 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AIRCRAFT MAINTENANCE EXPERT SYSTEMS

THESIS

AFIT/GCS/EE/83D-9 Gerald R. Ferguson
Capt USAF

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY (ATC)
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GCS/EE/83D-9

AIRCRAFT MAINTENANCE EXPERT SYSTEMS

THESIS

AFIT/GCS/EE/83D-9 Gerald R. Ferguson
Capt USAF

AFIT/GCS/EE/83D-9

AIRCRAFT MAINTENANCE
EXPERT SYSTEMS

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by
Gerald R. Ferguson, B.S.
Capt USAF
Graduate Computer Science

November 1983

Acknowledgments

Sincere thanks are given to my thesis adviser, Maj. Charles Lillie, to my committee member, Cpt. Robert Milne, and to my sponsor, Mr. Russell Genet of AFHRL at Wright-Patterson Air Force Base. Their tremendous moral support, impartial recommendations, and technical assistance is greatfully appreciated. Greatful appreciation is also extended to Mr. Brian Thompson of the 906th Tactical Fighter Group at Wright-Patterson for his valuable time and information concerning the F-4 aircraft maintenance environment.

Special gratitude is conferred upon my wife, Brenda. Without her typing assistance, continued encouragements, and devoted love, I could not have accomplished this work. I owe her a measureless debt for successfully maintaining a healthy and happy home environment during this time.

Finally, I thank all the new friends whom I have made while at AFIT. Their support and advice made this assignment less burdensome.

Contents

Contents

List of Figures

List of Tables

Abstract

The field of study known as "artificial intelligence" has gained considerable interest and support within the last few years. Machines are now capable of performing tasks that only humans were thought capable of performing. Machines have been constructed to assist humans in almost every aspect of their daily lives.

However, there are some daily activities in which the human must perform manual tasks. One such area is aircraft maintenance. The current manual procedures in this area are very strenuous and time consuming. This text investigates only one feature of those manual procedures, i.e. diagnosis of aircraft malfunctions.

This text provides design considerations for implementation of an "expert system" to assist in the diagnosis of aircraft problems. It illustrates the characteristics required for an automated diagnostic system to assist the average aircraft technician in the performance of his/her duties. The design of a "knowledge base" and "inference procedure" for such a system are presented. A working system model was developed on a microcomputer to demonstrate the feasibility for a full scale maintenance expert system.

I. Introduction

Background

With the advent of increased capabilities and decreased costs in digital computers, there has become an increased sophistication in their use. The computers built during the last two decades were huge machines which cost millions of dollars. Today, those large computer systems are being replaced by smaller, less costly computers which have the same capabilities as their "big brothers". The ultimate design and subsequent use of these newer computers has branched into two distinct areas.

One area is the continued progression toward faster and faster processing machines. These machines can quickly and accurately calculate large numbers, plot complicated graphs, and even understand the human voice. The trend of these computer systems is toward increasing the ease of computer/human communication which will tend to decrease the special training requirements for humans to interact with the computer.

The second area is the increasing sophistication of computers used in decision-making processes. These machines use complicated algorithms to correlate and disseminate information. The judgement and decision-making capabilities of these computers were formerly attained only by "intelligent" humans. Because of their reasoning capability, these computers have fallen into the field of "artificial intelligence".

Since computers are not endowed with any knowledge of their own, they must be provided with information from a human. Computers are currently being used for diagnostic applications in fields such as medicine and mineral explorations [DUD81]. These computers are supplied with a large amount of the knowledge of a human "expert" in a specific field of endeavor. These computers are then used to augment the human intellect of the "less than expert" individual in the diagnosis of a specific problem of that field.

A computer system used in this manner is called an "Expert System" or "Knowledge-Based System". The domain of factual knowledge possessed by an expert system is real; however, the knowledge is artifically generated. This knowledge can be accessed much faster and with greater accuracy than the same knowledge can be obtained from the human expert [DUD81]. For these reasons, the realm of artificial intelligence and expert systems is of significant interest to the Department of Defense (DOD) [HRL83].

Within the last few years research in the field of artificial intelligence has grown significantly, primarily in the area of medical diagnosis. Medical expert systems such as MYCIN, CASNET and INTERNIST [NAU83] have proven successful enough to warrant the investigation of similar expert systems in other fields of study. Development of artificial intelligence type systems for equipment maintenance in the commerical and industrial environments is currently underway at Boeing, Fairchild, and Hughes Aircraft Corporation [HRL83]. Also, several firms are currently contracting with such well-known centers of learning as Carnegie Mellon, MIT and Stanford to develop expert systems for the technology market.

Statement of Problem

The subject of this thesis involves the development of computerized expert systems to aid in overcoming some of the inherent problems incurred with today's weapon systems. The main problems encountered are technical complexity of the weapon, shortage of qualified technical personnel to maintain the weapon, and currency of the technical publications associated with the weapon system. This thesis will present the current characteristics of the aircraft maintenance environment and how the above problems may be reduced through the aid of a Maintenance Expert System (MES).

Due to the sophistication and rapid technological advances of today's DOD weapon systems, there is an ever increasing need for highly qualified technicians to maintain these systems. The incorporation of advanced technology, in both new and existing weapon systems, has made the accurate and timely assessment of damaged or malfunctioning equipment an extremely complex task. As the complexity of these systems increases, there will inevitably be fewer and fewer so called "technical experts" for a particular system.

The shortage of technical experts will have a drastic effect on the capability of a military unit to fulfill its mission. Since the critical step in repairing equipment is the assessment of damages, this step is normally accomplished by a technical expert of the specific system. The personnel performing this task must possess both a good technical background and a great deal of training and experience on the particular system being repaired. Within the Department of Defense, personnel who have attained these qualities are usually advanced to another job or transferred to another system within a few years. Therefore, with the increased complexity of new weapon systems and the movement of personnel, the human technical experts for a system are fast becoming a scarce resource.

Maintenance technicians, even the experts, are normally aided with their assessment of a system through the use of technical publications and manuals. However, these manuals are bulky, difficult to understand, and usually not updated with the current information pertaining to the system. Therefore, it is evident that some method must be found that will provide current information on a weapon system, will be easy to use, and will provide a quick and accurate assessment of the particular weapon system problem.

Thus, the purpose of this thesis is to assess the feasibility of augmenting the aircraft technician in the maintenance of aircraft with a computerized diagnostic system. Based on current diagnostic expert systems such as

MYCIN [YAS80], the maintenance expert system (MES) will contain a data base composed of factual knowledge obtained from technical manuals and technical experts for a specific aircraft. With these facts stored in the system, it should be possible for the less qualified technicians to quickly and accurately assess aircraft problems through interaction with the MES.

Updates of technical information concerning an aircraft system should be easier and faster to accomplish using the automated MES than with the current manual "paperflow" system. This will immediately provide current aircraft information to the technicians with the manual update system used as a backup. With this information in the MES, it is possible that the MES could be used for training both new and current technicians [HRL83]. Therefore, the goal of the MES is to overcome the deficiency of a weapon system caused by the shortage of technically qualified maintenance personnel [HRL83]. The MES could provide the means by which the average technician can make faster, more accurate assessments and repairs of the aircraft.

Scope

The scope of this thesis is to obtain information pertaining to the design and operational characteristics of current artificial intelligence systems and to apply that information to the general design of an expert system to assist in the maintenance of military aircraft. This thesis will present current maintenance problems and will present a plausible design for a MES. This effort primarily focuses on the high-level design characteristics of a maintenance expert system for use in any military aircraft maintenance organization. The major area of concentration of this thesis will be the design considerations which must be given to maintenance expert systems in general.

The actual database structure, programming language, and type of computer system best suited for the aircraft maintenance environment will be specifically addressed. The actual implementation of a small scale maintenance expert system will be constructed to determine the feasibility of a larger scale system. This small scale implementation will involve several elements of the engine system of the F-4 aircraft. Information from this simulation will be analyzed to provide conclusions and recommendations for full scale implementation of a serviceable maintenance expert system.

Due to its availability, the computer system used for this simulation will be an Apple II microcomputer using the Control Program for Microcomputers (CPM) operating system. The specific implementation language used in this simulation model will be LISP (List Processing), since it is available and is a well-known language used in artificial intelligence projects. The appropriate database will be determined following an initial investigation and data analysis of the aircraft system diagnostic information.

General Approach

A detailed literature search was conducted to find recent documents concerning experts systems, knowledge-based systems, and diagnostic systems. The results of the literature search indicate that most of the "state of the art" information pertaining to artificial intelligence and expert systems is limited to journals and papers written within the last five years. The "Background" section of this chapter provides an overview of the information found in this search.

The next step is to gather as much information as possible concerning the types of data structures applicable to artificial intelligence and expert systems. This information, along with the aircraft input information, will provide a means to forecast a specific database structure for use within an expert system for the aircraft maintenance environment. The selection, if any, of a specific Data Base Management System (DBMS) will be determined from the evaluation of this information.

The knowledge obtained in the previous step will also aid in determining the application language to be used for the aircraft maintenance expert system. Several programming languages currently used in artificial intelligence projects, such as LISP, ROSIE, and PROLOG, will be investigated to insure compatability with the requirements for the selected database design.

Concentration on selection of a typical host computer for an expert system will be toward minicomputers and microcomputers. These smaller computer systems initially appear to be likely candidates since they have capabilities equivalent to large mainframe systems, cost less to purchase, and some are highly portable.

Next, a detailed system design will be developed for a simulation model to determine the feasibility of an aircraft maintenance expert system. This model will be developed to gather information from technical experts on a specific aircraft system. The expert system will then be tested against the human technical experts on a real-life aircraft maintenance problem. Results of this simulation will be used to provide conclusions and recommendations for full-scale implementation of an aircraft maintenance expert system.

Contents

The remaining chapters of this thesis provide specific information and a detailed analysis for a maintenance expert system. Chapter two presents the "System Descriptions" of the aircraft maintenance environment and of a MES. It contains current aircraft maintenance procedures and shows how these procedures will be represented within the components of a MES. The third chapter concentrates on the "System Design" for a MES and presents considerations for the database, implementation language, and hardware for a MES. Chapter four contains a "System Model" used to simulate the implementation of a MES. This chapter contains the model design, a test plan, and the test results. The final chapter contains "Conclusions and Recommendations" for a maintenance expert system used within the realm of aircraft maintenance.

II. System Descriptions

Purpose

This chapter presents the reader with a description of the aircraft maintenance environment and a detailed view of the components of an expert system. Since the emphasis of this thesis is involved with improving the diagnostic capabilities of the maintenance technicians, the reader should understand the manual diagnostic processes and terminology of the aircraft maintenance environment. The reader should also understand what comprises an expert system and how an expert system might be used within the maintenance environment.

Aircraft Maintenance Environment

An initial meeting was conducted with Mr. Brian Thompson, Quality Assurance APG Inspector for the 906th Tactical Fighter Group (F-4D aircraft) at Wright-Patterson AFB, to determine current aircraft maintenance procedures. Information was gathered on the F-4D jet aircraft maintenance environment, but similar information applies to any aircraft maintenance complex.

In order to keep the F-4D flying, maintenance is performed under a "crewchief" concept within an organizational environment. The crewchief is the maintenance expert responsible for the aircraft's flyable condition. He/she is constantly aware of any/all problems associated with the aircraft. He/she is "assigned" to the aircraft as its "owner" to instill pride and boost morale among maintenance personnel. This concept has proven itself through competitive efforts of crewchiefs to keep their aircraft in perfect flying condition. However, the crewchief is assisted in maintaining the aircraft by other "specialists" within the organizational maintenance complex.

Organizational maintenance consists of operational checks of the aircraft's systems, isolating failures, adjustments, and removal and replacement of line item units as necessary in accordance with specified maintenance manuals. The maintenance manuals used by the Air Force are known as Technical Orders (T.O.s). These T.O.s provide the maintenance personnel with required technical documentation to maintain the aircraft. This documentation includes general aircraft information, ground operations, environmental conditions, and extensive troubleshooting procedures.

Each series of aircraft, such as the F-4, must be maintained in accordance with its own set of specified T.O.s. For instance, the nomenclature for one series of the set of T.O.s used to maintain the powerplant and propulsion system for the F-4 is "T.O. 1F-4C-2-8". The "1F-4C" indicates the primary aircraft identification and the "-2-8" specifies the aircraft system to which this T.O. applies.

This T.O. is only one of a series for the fourteen major systems of the F-4 aircraft. This T.O. consists of over 900 pages of technical information associated with the general upkeep and maintenance of the aircraft's two engines. The other T.O.s for this aircraft are similar in size.

Through the use of these T.O.s the crewchief and his specialists insure that the F-4 aircraft is maintained in excellent flying condition. For any specific problem which might plague the aircraft, the crewchief uses his/her technical expertise along with the proper T.O. to "nurse" the aircraft back into its flyable state.

More specific information pertaining to the complexity and amount of information contained in these T.O.s is provided in Chapter III of this text.

Expert System Concepts

An expert system is derived from two primary concepts which should be recognized prior to a discussion of the system itself. The first concept is that the expert system will contain specific knowledge about a particular field or domain that the system is designed to model. This "knowledge" is an incorporation of existing facts from human experts and documentation within that specific field. In order to artificially generate "knowledge", the expert system is able to apply reasoning skills in an attempt to provide a solution(s) to the problem(s) generated by its user [DUD81].

Although the expert system will closely model the human expert, it is not a replacement for the human. Instead, it will only act as his/her assistant. Since an expert system does not have the innovative and imaginative abilities to derive new problem solutions, it can not be considered as a "replacement" for the human expert [YAS80, AIS82, HRL83].

The second concept involves the classification of expert systems. Duda and Gaschnig [DUD81] have classified the most well-known expert or knowledge-based systems into eight general categories according to each system's function. However, these categories are more appropriately subdivided into only two distinct classifications: diagnostic or problem solving systems, and pedagogic or teaching systems.

Diagnostic systems are application oriented. The purpose of this type system is to assist the user in the resolution of a specific problem of a particular field. The pedagogic system is less concerned with problem solving and more concerned with how specific knowledge or information should be taught. Both types of systems are educational in nature which is one of the major focuses of research and development funding in artificial intelligence [CLA79, HRL83].

Expert System Components

What distinguishes an expert system from an ordinary applications program? As stated by Nau [NAU83], "the main difference is that in most expert systems, the model of problem-solving in the application domain is explicitly in view as a separate entity or knowledge base rather than appearing only implicitly as part of the coding of the program". Duda and Gaschnig [DUD81] suggest "another characteristic of most expert systems is that they try to mimic the way human experts make decisions" through a valuable set of rules.

Most authors [FEI80, YAS80] agree that an expert system consists of two parts. One is the knowledge base that contains the facts of the domain. The second part is an inference procedure which is used to "reason" about the knowledge base. However, a third part is required to act as the interface between the system and its user [FEI80, NAU83].

Inference Procedure. The inference procedure is the mechanism which provides the central control for the expert system. It may be called the main program, inference system, or system driver, but overall, it is the "brains" of the system [DUD81]. For simplicity the inference procedure will be referred to as the "driver" within the following discussions.

The driver's primary effort is toward reasoning and making inferences based upon the application of rules contained in the knowledge base. It accomplishes this effort through the implementation of a control strategy. There are two basic control strategies implemented in current expert systems [GRA79, DUD81]. The implementation of one selected strategy is based upon the type of expert system, either diagnostic or pedagogic, and the specific domain of application.

One of the simpliest strategies is known as forward searching or data-driven searching [NAU83]. This strategy employs knowledge base rules on some initial state or condition of the data and continually applies those rules to new conditions until the desired goal is attained. This strategy is also known as "forward chaining" since rules are chained together to produce new conditions [DUD81].

A second strategy, employed in most diagnostic expert systems, is known as backward searching [NAU83, WIN79]. This strategy begins with the selection of a specific goal and then scans the rules to find those whose consequent actions can achieve that goal. Since this is a goal oriented strategy, it is also known as goal-driven, backward-chaining, or consequent reasoning [DUD81].

Two techniques are generally used by the driver in state-space searches. One technique called backtracking uses procedures which explore one path as far as possible. If the path reaches a "dead-end" or it is somehow determined that no goal can be reached, the procedure backtracks to a previous state and chooses a path in a different direction. These procedures are usually written recursively to avoid redundancy and complexity of code [DUD81, NAU83].

Another technique is referred to by Nilson [NIL80] as graph-searching. This method searches several paths simultaneously while keeping track of several "current states" of the system. Some paths may be explored faster than others or, as with a breadth-first search, all paths are searched at the same speed.

An alternative technique to state-space searching is a technique commonly referred to as problem reduction [NAU83]. This method partitions or decomposes the problem into smaller subproblems which can be solved separately. The combination of the subproblem solutions will yield a solution to the overall problem. Scientists at MIT and Carnegie-Mellon are currently developing "parrallel-processing" computers to employ this technique [AIS82]. These computers will break a problem into many parts and solve each part simultaneously.

The driver of the expert system must also be capable of managing knowledge acquisition. It must include facilities for entering new information into the knowledge base, updating current information, and deleting unwanted/outdated information from the knowledge base. This portion of the driver is required for a quality expert system [NAU83].

Knowledge Base. The knowledge base is that portion of the expert system which the driver calls upon to accomplish its reasoning process [NAU83]. The knowledge base itself is passive. It only serves as a holding place for "knowledge" accessed during resolution of a problem.

The most important parts of the knowledge base are the facts of the domain and the rules governing their use. The facts are those items of information obtained from the human expert and the documents which he/she would use. These documents contain facts such as definitions, descriptions, and relational data. The rules are those decision processes by which the expert formulates those facts into usable "knowledge".

The representation of knowledge within the knowledge base is acquired through the use of interpretive procedures according to the domain of the expert system [BAR80, DUD81, NAU83]. The three procedural techniques or strategies most commonly used are finite state machine, predicate calculus, and production rules.

An expert system which is modeled as a finite state machine proceeds from some initial state to some goal state. The transition from one state to another is determined by the collection of previous states and use of decision rules to determine the next state. Predicate calculus based systems use formal symbolic notation for expression of logical relationships and assertions within the knowledge base. Most diagnostic systems employ the use of production rules, much as the human expert does, in the diagnosis of a problem [DUD81, NAU83].

Production rules specify a conditional action in the form of an "IF...THEN..." construct. The condition part is usually a conjunction of patterns to be matched in the knowledge base and the action part is the conclusion (or action to be performed) based on given conditions [DUD81]. The actions can result in changes to the knowledge base assertions, query the user for more information, or enable and disable decision rules.

Knowledge based expert systems have also been developed to aide in the development of other knowledge based systems. One such system, GUIDON, was developed at Stanford to teach facts and problem-solving strategies used within the MYCIN system [DUD81, FEI82]. Another system, AGE, was specifically designed to allow the implementation of a broad spectrum of knowledge bases [DUD81, NAU83]. It gives the designer a set of separate preprogrammed modules to be used in representing the knowledge base and the inference procedure (driver).

Interface. The interface is that part of the expert system which allows and controls communication with the user. As previously mentioned, the interface functions to interact with the user during diagnostic sessions. Also, it is used in acquisition of new knowledge for the system [FEI80, NAU83].

The user interface supports the natural language used in formulating system queries. This interface provides the user with consultative information within the realm of the specific knowledge domain. Current diagnostic expert systems such as MYCIN [FEI80] and INTERNIST [DUD81] rely on the user to be extremely competent in his/her field in order to understand the information provided. However, as Yasaki

[YAS80] points out, the advancement of speech systems in AI research will enhance these system inquiries.

The knowledge acquisition interface is the most critical part of the expert system [FEI80, AIS82]. It is usually the human expert of the problem domain who uses this interface to input or update the expert system's knowledge base. The task of knowledge acquisition is currently a "very tedious, time-consuming, and expensive procedure" [FEI80] and is the "bottleneck" problem in Artificial Intelligence.

Summary

To gain an understanding of the quantity and quality of data required for accurate diagnostics, an explanation is needed of the human maintenance expert's thought process during diagnosis of an aircraft problem. Diagnosis of the engine system of a jet aircraft is used for illustration; however, these same processes apply to any system of the aircraft. This will also provide the reader with a view of how the expert system functions.

In order to diagnose a problem, a problem must exist. For example, suppose that the aircraft has aborted its mission and the pilot has entered "The #2 engine does not start." into the aircraft forms. From this point the maintenance crew will diagnose and repair the problem to return the aircraft to its flyable condition.

There are numerous reasons that a jet engine will not start, anything from an electrical failure to a faulty fuel system. The maintenance expert has two choices of progression to obtain the correct reason. The first choice is that he/she can begin from the ignition switch being turned on and work toward the failure point. The second choice is that he/she can start at the point of failure and trace backwards to find the cause of the problem. Most expert systems such as MYCIN [DUD81, NAU83] use a variation of the latter approach.

The following questions and answers illustrate the thought process of a typical human expert for a jet aircraft engine with the stated problem:

Q: Was the engine rotating?
A: Yes

Q: What was the rotation speed?
A: 35%

Q: What was the throttle setting?
A: Idle

Q: What was the EGT rating?
A: 90 degrees

Problem cause: Bad fuel ignitors

In order for the expert to determine the "probable cause", he/she must have "knowledge" of the system under diagnosis. The "knowledge" of the system consists of factual and heuristic information [FEI80]. The factual information is the shared knowledge which is written in the aircraft technical manuals and publications. The heuristic information is the knowledge which constitutes the judgmental rules of the field, or "the art of good guessing". These are the two most important aspects of information pertaining to solving a given problem. If the facts are not known then the proper decision cannot be made. If the proper decision is not made, then subsequent efforts may be fruitless.

The previous jet engine diagnostics example illustrates the use of production rules within the control stategy that allowed the expert to arrive at a probable cause for the engine malfunction. The example shows the route taken for a "yes" answer to the first question; however, a "No" answer would lead to other possible questions. Different answers to any of the questions would possibly prompt other follow-up questions until the probable cause was determined.

The above question and answer sequence illustrates how an expert system might operate to provide the same information and results that the human expert obtained. The questions would be generated according to the answers provided by the expert system user. It is the function of the expert system to artifically duplicate the thought process (questions) in a manner similar to that of the human expert. The following chapter will provide more specific guidelines and characteristics for an expert system within the domain of aircraft maintenance.

III. System Design

Purpose

The design of any system must consider the storage of data and presentation of information about that data to the user. This chapter presents this author's view of the general design of a diagnostic expert system to be effectively used within the realm of aircraft maintenance. The main effort of this design is software oriented, however some hardware considerations are presented.

This chapter will include a discussion of a generalized database design and an implementation language for a maintenance expert system (MES). The three parts of an expert system (knowledge base, inference procedure, and user interface) will be discussed in this design. The current manual processes within the aircraft environment will be described for contrast to this system design.

Knowledge Base

The sophistication of modern aircraft weapon systems has generated enormous amounts of data which must be accurately analyzed to keep those systems in operation. In order to effectively diagnose a problem within the weapon system, the data accessed must be logically structured to provide the fastest and most efficient solution. The following description of the manual data storage system currently used within most aircraft maintenance complexes is presented to provide a comparison with the design of an automated knowledge base to replace the manual system.

Manual Knowledge Base. In order to acquire an accurate realization for the amount and complexity of information required to maintain any modern aircraft, the information system used with the F-4D jet fighter aircraft will be described. As presented in the previous chapter, this information system is similar in size and structure to most other modern aircraft systems within the Air Force. Some military flying organizations have more modern and/or larger aircraft, but the operational characteristics of each organization is virtually the same.

The "database" for this manual information system (knowledge base) is comprised of a set of publications called Air Force Technical Orders (T.O.s). These T.O.s provide all the information that the organizational personnel need to fly and maintain the aircraft. This information ranges from general aircraft characteristics such as weight and dimensions, to specific instructions of how the aircraft is to be repaired for any known problem.

The knowledge base (T.O. series) for each aircraft is generated through years of research and development efforts. The information pertaining to a given aircraft is formulated by the aircraft manufacturer and is supplied to the Air Force with the aircraft as part of the contract. Highly proficient Air Force personnel monitor and assist the manufacturer in the T.O. development to insure all systems are thoroughly documented prior to delivery. These T.O.s are published and maintained under the direct authority of the Secretary of the Air Force to insure consistency throughout all aircraft organizations.

The T.O. system is sub-divided into separate categories according to their application to the specific aircraft. For example, T.O. manuals designed for maintenance inspection purposes are separate from the manuals designed to provide information to the pilot about the aircraft's flying characteristics. Table III-1 provides a partial list of the T.O.s which comprise the database for the F-4D aircraft.

F-4D Technical Orders	
1F-4D-1	General Aircraft Information Manual
1F-4D-2-1	Aircraft General Maintenance Instructions
1F-4D-2-2 through 1F-4D-2-38	Maintenance Instructions Technical Manuals
1F-4D-3	Structured Repair instructions Manuals
1F-4D-4	Illustrated Parts Breakdown Manuals
1F-4D-5	Basic Weight Checklist and Loading Data
1F-4D-6	Inspection Manuals, Charts, and Work Cards

TABLE III-1

As can be seen in this list, some T.O. categories contain several sub-levels, such as the "-2" series. In fact, the "-2" T.O. series contains the largest amount of information of any of the T.O.s. Since this series of T.O.s is primarily used by the aircraft maintenance personnel, the "-2" series will be the prime reference for the expert system design in this chapter.

A conceptual view of the overall structure of this aircraft database shows it to be "hierarchical". Each series of aircraft within the Air Force inventory has its own set of T.O.s. In order to use any of these T.O.s, the user must know how the T.O. system is sub-divided. This will enable him/her to logically traverse the hierarchical structure to obtain the specific information desired. Figure III-1 illustrates this hierarchical structure and shows the lowest level at which a particular T.O. is selected. (This figure depicts the logical path to information about the Powerplant and Propulsion System contained in T.O. 1F-4D-2-8.)

At the individual T.O. level the information is further structured into sections and paragraphs. In some cases the lower level information is cross-referenced to other T.O.s to form a "network" of information. This usually causes complications in the access to the desired information, but is necessary due to the complexity of these weapon systems and to avoid duplication of information.

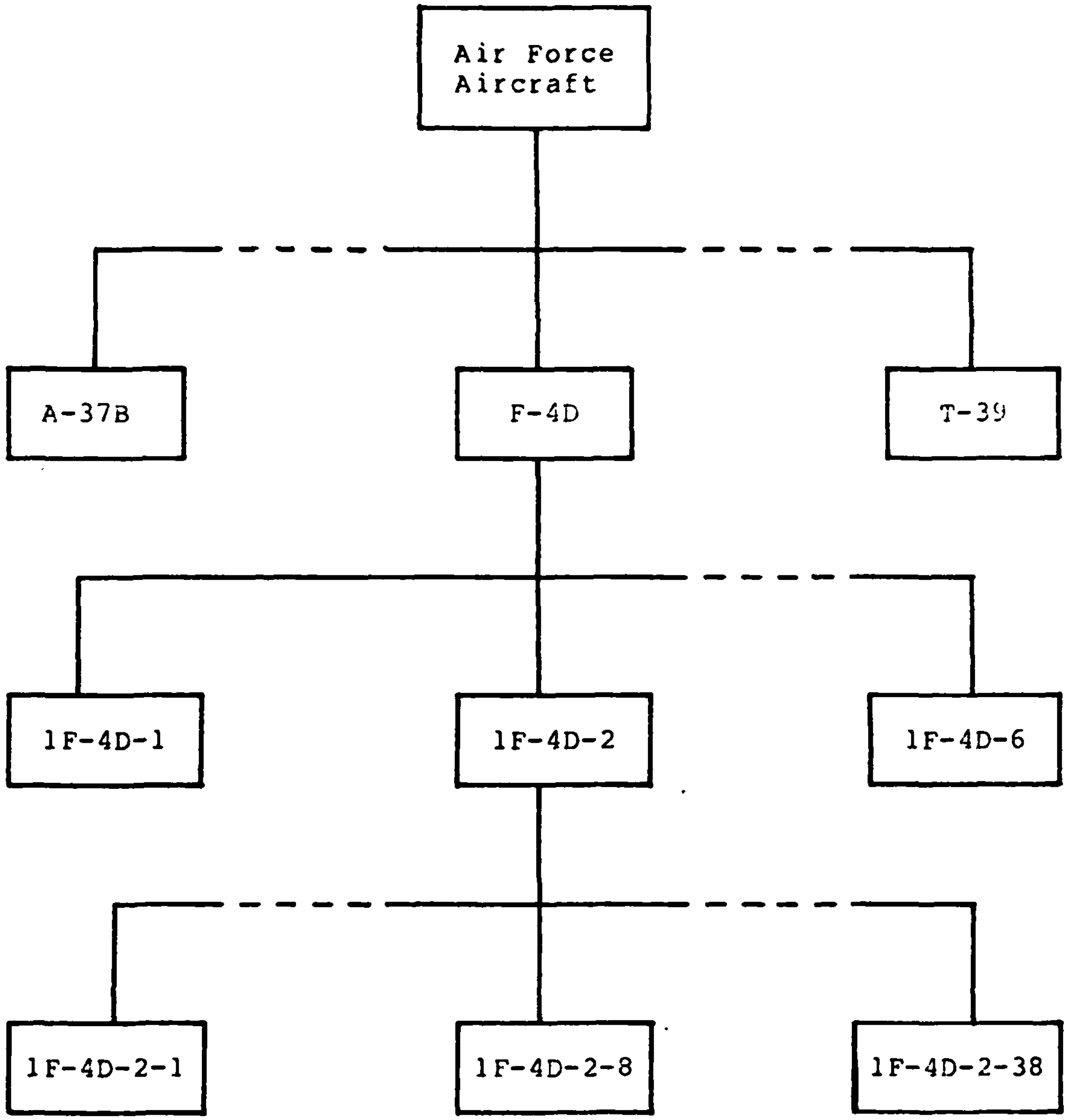


Figure III-1. Aircraft Technical Order Structure

The primary characteristic of the T.O. information is that it is easy to use. Most of the information is instructional and/or informative. It is written in standard English of the high school level. Detailed explanations are given for every system and component of the aircraft and are usually accompanied by charts, diagrams, and/or pictorial representations.

As previously mentioned, the bulk of the T.O. series for any aircraft is the "-2" series. This T.O. series contains all the necessary information for "flightline" maintenance of the aircraft. Flightline maintenance is maintenance usually performed within the organizational environment.

The "-2" series for the F-4D aircraft is composed of thirty-eight separate T.O. volumes. Each of these T.O.s contains a troubleshooting section which provides procedures for identifying malfunctions, isolating the cause to the smallest line replaceable unit and correcting the malfunction. These procedures are presented in logic tree form for systematic troubleshooting.

A "master" troubleshooting manual, T.O. 1F-4D-2-34, is available and contains "symptom lists", troubleshooting procedures, and troubleshooting schematics. This T.O. lists all flight and ground operation symptoms for which troubleshooting procedures have been prepared. The symptom lists consist of eighty-nine tables containing symptoms and references to troubleshooting procedures within the applicable -2 system T.O..

Each -2 system T.O. has its own "symptom index". For example, if a problem occurred within the engine system, the master T.O. would refer the maintenance technician to T.O. 1F-4D-2-8. This T.O. contains more specific references from the symptom index to the appropriate troubleshooting procedure. Table III-2 is an exact duplicate of page 2-165 of T.O. 1F-4D-2-8. This table lists several engine areas with troubles which occur in those areas and references to the appropriate troubleshooting procedure.

From the sympton index the maintenance technician can find the paragraph or section to repair the trouble. Suppose the aforementioned engine problem was "Rough or vibrating engine". This trouble is found in the index under "Compressor" and the troubleshooting procedure refers to "Paragraph 2-103". Table III-3 shows what the technician will find in paragraph 2-103.

At this point the manual database structure has transformed from a hierarchical to a "binary-tree" structure. The troubleshooting procedures generally found in the T.O.s consist of questions and answers. The technician responds to each question with a "yes/true" or "no/false" answer and the procedure then refers the technician to another question in the diagnostic tree structure or possibly to the problem solution.

Symptom Index

Indication of Trouble	Troubleshooting Procedure (Refer to)
AFTERBURNER	
.No afterburner light	Paragraph 2-99
.Slow afterburner light	Paragraph 2-108
.Slow afterburner termination	Paragraph 2-100
.Afterburner surge	Paragraph 2-109
ANTI-ICING	
.No anti-icing indication	Paragraph 2-101
.Anti-icing light flickers or comes on at high power settings with anti-icing switch OFF	Paragraph 2-137
CIRCUIT BREAKER POPS	
.R Main Ignition(5CB304)	Section III
.L Afterburner Ignition(5CB305)	Section III
.R Afterburner Ignition(5CB306)	Section III
.L Main Ignition(5CB307)	Section III
.L Ign Unit #1(5CB311)	Section III
.R Ign Unit #1(5CB312)	Section III
.L Ign Unit #2(5CB313)	Section III
.R Ign Unit #2(5CB314)	Section III
.Anti-ice (39CB301)	Section X
COMPRESSOR	
.Refer to T.O. 1F-4D-2-8-CL-2 before troubleshooting this symptom	Paragraph 2-102
.Rough or vibrating engine	Paragraph 2-103
FUEL	
.High EGT	Paragraph 2-105
.Low EGT	Paragraph 2-104
.Fuel flow out of limits	Replace Fuel Control
.No or slow acceleration to Idle	Paragraph 2-124
.Slow aceleration Idle to MIL	Paragraph 2-106
.No or slow fuel dump on shutdown ...	Replace drain valve

TABLE III-2

2-1Ø3. ROUGH OR VIBRATING ENGINE

	No	Yes
a. Does compressor or turbine show evidence of FOD?.	b	c
b. Are engine mounts secure?	d	e
c. Refer to section XII limits	-	-
d. Replace or retorque as necessary	-	-
e. Are hydraulic pump clamps and adapters secure? ..	f	g
f. Replace defective component	-	-
g. Do variable vanes follow schedule?	h	i
h. Rig variable vanes	-	-
i. Perform SOAP check on engine and CSD oil. Is check satisfactory?	j	k
j. Return engine to next higher maintenance level ..	-	-
k. Return for test stand vibration check of engine..	-	-

TABLE III-3

Any skilled aircraft maintenance technician is usually able to use this T.O. system to troubleshoot and repair the aircraft malfunction. However, some complicated problems require the assistance of a more knowledgable technician or expert. The following knowledge base design will incorporate the factual information from the T.O.s with the heuristic information of the expert. This combined information within the knowledge base of an expert system will provide the skilled technician with the ability to repair the aircraft when the human technical expert is not available.

Automated Knowledge Base. The logical model for the design of an automated knowledge base will follow the same structure as the manual system. Since the total amount of information for a specific aircraft is quite large, the upper level of this knowledge base will consist of a "directory" to provide access to the lower level structures. Each sub-level shown in Figure III-1 will consist of other directories for each immediately subordinate level.

The lowest level of the structure, individual T.O.s, will contain the user information. That information which is informative or instructional will be maintained by sections and paragraphs within those sections. This allows the knowledge base to maintain its hierarchical structure. However, due to the nature of the troubleshooting information, it will be structured as separate sub-trees within the T.O. paragraphs. The heuristic knowledge from the technical experts will also be included as a separate troubleshooting paragraph at the sub-tree level.

Since the hierarchical structure of the directory information of the knowledge base is at most four levels deep, a separate Database Management System (DBMS) to maintain these "directories" will not be considered. The access, manipulation, and updates to these levels can be conviently handled through routines of the expert system program. The directory levels and the information on each level can be linked together by manipulation of logical pointers to simulate the appearance of an internal DBMS.

The physical model of this knowledge base will consist of several files. A single file is used to maintain the hierarchical structure for the directories. Multiple files at the T.O. level will maintain the information from the current set of aircraft T.O.s. This allows easy access and manipulation of the automated T.O. system by the internal DBMS. The DBMS will provide the user with mechanisms for retrieving information about the T.O. system. It will also provide the user with access to the specific T.O. file, but not access to the information within that file.

The data and diagnostic information at the T.O. file level is stored in sentence, table, and figure forms and should be presented to the user in those same forms. The informative and instructional data can be accessed through features of the programming language [BAR82]. This is also true for the troubleshooting procedures in the tree form. This suggests that the implementation language for this expert system must be capable of effectively processing "strings" of information rather than individual pieces of data from the knowledge base.

Implementation Language

Due to the original scope of this text, the choice of languages is limited to those that will operate on a microcomputer. Possible languages available are Pascal, FORTRAN, PL-1, and LISP (LISt Processing). Of these, only LISP has built-in language functions to provide the string processing capabilities desired for AI programming. Other languages which have evolved in artificial intelligence applications, such as ROSIE and PROLOG [BAR82], have these same capabilities, but are not currently available for microcomputer use.

Since its conception in 1958, LISP has become the primary AI programming language [BAR82]. It has been used by the vast majority of AI researchers in all subfields and all students in AI laboratories learn LISP, so that it has become a "shared" language. This language is a "natural vehicle for AI research because there are features of LISP that are critically important in AI programming" [BAR82].

Symbol manipulation is required to make computers appear to be intelligent [BAR82, CHA80]. This intelligent behavior appears in programs which apply common-sense reasoning, provide natural language interfaces, and support educational systems. Most of these seemingly intelligent programs are written in LISP.

Symbolic expressions are used within a LISP program to provide the program with the capability of "remembering" and working with data and procedures. This allows the program to perform the same symbol-manipulation that people perform with a pencil and paper. A typical LISP program has sections that recognize particular symbolic expressions, tear old expressions apart, and assemble new expressions. Figure III-2 illustrates the basic structure of a LISP expression.

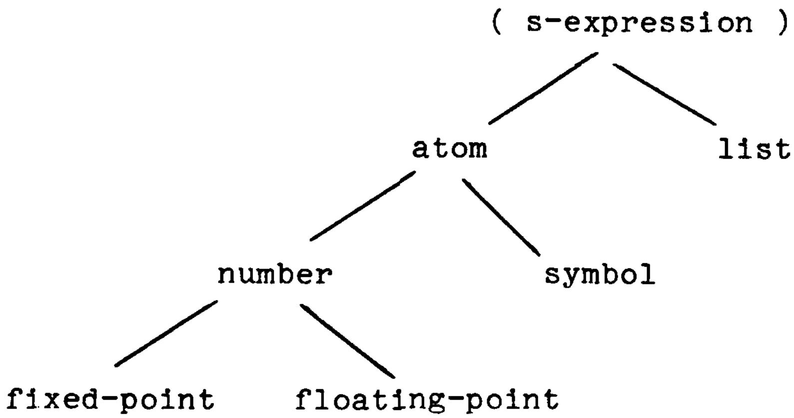


Figure III-2. LISP Expression Structure

This basic structure is used to define both data and executable statements (functions) within LISP. The "atom" is the smallest accessible element. The "list" can be one atom or a list of atoms. The "symbol" can be a user created item or an available function of LISP. Expressions can also be nested through the use of the "list" element.

Executable statements within LISP programs consist of functions defined in a rather mathematical format. Each function call is represented as a list. The first element of the list is the name of the function and the other elements of the list are the function arguments. The arguments can also be calls to other functions or to itself thereby providing a recursive process.

Each function call in LISP is embedded in parentheses. The parentheses are used to indicate the program or function structure and gives LISP programs a distinctly different appearance from those of other languages. In fact, "people who know other programming languages frequently have difficulty learning LISP, while many people with a mathematical background find LISP an easy first language to learn" [BAR82].

Within LISP there are approximately seventeen "primitively" defined categories of functions which can be used alone or can be used to build other functions. A total of eighty-three LISP functions are defined in machine language for maximum effiency on the microcomputer. Table III-4 shows a list of some of these categories with their defined use and available function names.

LISP Functions

SELECTOR -- used to select a desired element of a data structure. CAR, CDR, CADR, CAAR, CDDR

CONSTRUCTOR -- used to generate the data structures. CONS, LIST, OBLIST, REVERSE

COMPARATOR -- used to compare data structures. EQUAL, MEMBER, GREATERP, LESSP

LOGICAL -- permits Boolean combinations of truth values. NOT, AND, OR

ASSIGNMENT -- used to assign values to program variables. SET, SETQ, POP, PUSH

NUMERICAL -- provides capability for mathimatical calculations. NIMUS, PLUS, TIMES, DIVIDE

READER/PRINTER -- used to provide input/ouput capabilities.
RDS, READ, READCH, READLIST
WRS, PRINT, SPACES, TERPRI

ENVIRONMENT -- (only available with the microcomputer LISP due to limited computer memory)
-- used to save the current operating environment or load a previously saved environment. SAVE, LOAD

AUXILLIARY -- used to define the evaluation and function definition functions. QUOTE, EVAL, APPLY, COND, LOOP

TABLE III-4

Since the primary data structure in LISP is the expression, two functions (CAR and CDR) provide the capabilities needed to manipulate the expressions. CAR provides access to only the leftmost element of the expression and CDR provides access to only the rightmost elements, excluding the leftmost element. For example, suppose from the following expression (EXPRS) the second element in the list ("COMPUTERS") was to be obtained:

```
(SETQ EXPRS  (SMART COMPUTERS ARE GREAT))
```

In order to access only the second element, a combination of the CAR and CDR functions is needed:

```
(SETQ ELEM (CAR (CDR EXPRS)))
```

LISP expressions are evaluated from the innermost parentheses outward. For this example, the rightmost (innermost) elements ("COMPUTERS ARE GREAT") are accessed and then the leftmost of these elements is accessed to get the element "COMPUTERS". This is the basic process in which LISP is able to process and manipulate symbolic data.

INFERENCE PROCEDURE

The Maintenance Expert System (MES) used within the aircraft maintenance environment must perform in much the same manner as the human technical expert. To do this, the inference procedure (or driver of the MES) must be capable of accomplishing these basic functions:

1. Accept and process problem queries from the user.
2. Engage the user in a dialogue when a solution to the problem is not apparent from the query.
3. Provide the user with explanations of terms and the reasoning used to arrive at a solution.
4. Allow accurate updates to the knowledge base.

When a problem exists in an aircraft system, there are usually several components of the system which must be checked to arrive at a solution. In order for the MES to properly diagnose a problem, it must have as much information about the problem as possible. The MES will use this information to traverse through its knowledge base until a solution is found or it determines that more information is required.

The control strategy used to enable the MES to "reason" should be choosen based on the contents of the knowledge domain [BAR82, CHA80]. Since production rules are employed in the representation of the data in the knowledge base of the aircraft MES, the control strategy should be either "forward" chaining or "backward" chaining. However, due to the method in which information must be obtained from the knowledge base, the MES diagnostic control strategy must be that of forward chaining.

It is forseen that most uses of a MES will be of the dialogue nature. This requires that the MES be extremely "user friendly". The MES will ask questions pertaining to the components of the aircraft system involved and will search for a solution from the knowledge base according to the user responses.

During this dialogue the MES will be required to provide explanations to the user as needed. Terms which are unfamiliar to the user must be clarified. Explanations of "why" the MES has asked a specific question must also be provided. This will provide a mechanism by which the user can evaluate the MES as well as further his/her knowledge of the aircraft system.

Since the knowledge base contains all the information pertaining to the aircraft, it must be kept current. As previously discussed, only the information and instructions contained in the T.O.s are "approved" for use in maintaining the aircraft. When approved changes to the T.O. are generated, it is sometimes several weeks before those changes are published in printed form and distributed to all affected aircraft organizations. The MES will provide the means for quickly updating the knowledge base at the time the changes are approved.

As a separate feature of the inference procedure, the user will be able to use an internal database management system (DBMS) to update the knowledge base. However, the use of this DBMS must be restricted to only those maintenance staff personnel who normally perform the T.O. update function. This central control of the knowledge base will continue to insure the quality of the MES information.

Database management facilities are present within LISP [BAR82]. This language provides a "property-list" feature for performing such operations as pattern-matching, recursive data manipulation, and context assession. However, there is one drawback. The property-list structures are indexed in only one direction which makes backtracking of information very cumbersome. Normally, the requirements of the MES will dictate that processing only proceeds in one direction.

The MES may also perform additional functions such as maintaining statistics on failure rates of certain aircraft system components. This information is vital to the aircraft organization and may be of great use in future updates of the knowledge base. It may also be used to isolate manufacturer defects and/or costly maintenance errors to the organization management personnel.

USER INTERFACE

The MES must be capable of communicating effectively with the average maintenance technician. The dialogue must be interactive and as natural as possible. Terms used in the dialogue must be familiar to the user or should be explained as previously mentioned. The majority of this interface should be incorporated within the inference procedure at the points where a dialogue is required.

A natural language interface is most appropriate for a MES. The user will be able to input the aircraft problem directly from the aircraft maintenance forms. The MES will be capable of analyzing this "problem statement" and proceed to a specific portion of the knowledge base for diagnosis. However, it is perceived that most aircraft problems will be of such a nature that the MES will require additional information from the user. At this point the question and answer sequence, as previously described, will be invoked.

The inference procedure and user interface will act together to provide the "intelligence" capability of the MES. A certain amount of "remembering" must take place while diagnosing a specific problem in order to provide accurate diagnosis and explanations to the user. Answers to diagnostic questions as well as production rules accessed from the knowledge base must be remembered. The inherent recursive capabilities of LISP allow this function to be effectively accomplished.

HARDWARE CONSIDERATION

In order to implement a MES using a microcomputer, several hardware considerations must be made. The limiting factors on a microcomputer are normally the disk storage space and the internal memory capacity. Another factor to consider is the transportability of the microcomputer. Chapter V of this text presents some of the futuristic technological "visions" for AI, but at this point the current "state of the art" hardware is considered for use within a MES.

The knowledge base for a MES will be extremely large. Hard disk drives are currently available for microcomputers and will hold approximately five to ten megabytes of data. These disk drives are more cost effective than floppy disks which hold up to only two megabytes of data and cost about the same. However, even with five megabytes of on-line storage, the entire knowledge base may not fit on one disk. In that case the diagnostic information should be stored on a disk separate from the more general aircraft information.

Most microcomputers are available with at least 64k (64 kilobytes) of internal memory. This limitation should not be a problem due to the compactness of the object code generated by the LISP interpreter. The MES should be capable of operating within this memory space without any problems. If not, there are microcomputers currently available with as much as 128k of resident memory.

The MES must be portable and durable to perform under aircraft maintenance conditions. There are several aircraft maintenance organizations that are highly mobile, such as Reserve units. When these units deploy to another operating location, they also take all their T.O.s with them. For this reason, the MES must be capable of being transported with its using organization. Most microcomputers are small enough that they can be easily transported to a new location and be operational within a few minutes after arrival.

Also, some microcomputers are capable of being operated from only battery power while others must have 110 volts of AC electrical power. During wartime conditions the need for a MES which will operate on battery power may be great.

SUMMARY

This chapter has presented a view of the current aircraft maintenance procedures and the design considerations for a MES to be used to augment those procedures. The details for a full scale design of a specific MES are too numerous to be presented in this text. However, the next chapter discusses how these design considerations are implemented within a small scale MES for the F-4D aircraft maintenance environment.

IV. System Model

Introduction

This chapter presents the detailed design of a small scale expert system for the F-4D aircraft maintenance organization. This design includes the knowledge base, inference procedure, and user interface. The predominance of the information presented lies in the methods by which the LISP language can be used to construct such a system. Hardware features, such as main and secondary storage, are discussed at appropriate times for comparison of this model to a possible full scale system implementation.

Knowledge Base

As discussed in the previous chapter, the knowledge base for the aircraft maintenance environment has a hierarchical structure. Since the mainstay of the expert system is to assist in the diagnosis of aircraft problems, this design will present a model which will easily traverse the heirarchical structure and allow access to the diagnostic information. Concentration of effort will be within the use and management of that information at its lowest level in the structure.

As shown in Figure III-1, each Air Force aircraft has numerous technical orders (T.O.s) to aide the aircraft technicians. The T.O. series used in the diagnosis of aircraft problems is the "-2" series, as previously discussed. Due to the bulk of diagnostic information contained in this series of T.O.s, this design will be limited to the information contained in T.O. 1F-4D-2-8 for the F-4D Powerplant and Propulsion System.

This T.O. contains over 900 pages of maintenance information which includes more than 200 illustrations and tables. Within the Symptom Index presented in Table III-2, there are approximately 50 categories of trouble areas which can plague the propulsion system of the aircraft. The troubleshooting procedures for these trouble areas consist of approximately 190 questions and 225 related corrective actions. The MES knowledge base implemented in this design model includes only 44 of those questions and 48 corrective actions. This accounts for almost 25 percent of the diagnostic information for this single T.O. and provides a basis for estimating the size and complexity of a full scale MES.

One should remember that this amount of information pertains only to the Powerplant and Propulsion System of the aircraft. The other thirteen major systems of the F-4D each contain approximately the same amount of information. Therefore, in this model only two percent (2%) of the total diagnostic information for the entire "-2" T.O. series is represented. However, this small knowledge base provides enough data for a realistic assessment of the MES storage requirements of the total aircraft diagnostic information.

Storage and retrieval of information in the form of questions and corrective actions from the MES knowledge base is accomplished without the aid of a separate DBMS. Since database facilities are present within LISP features, the physical structure of the knowledge base could be designed as a LISP data structure enclosed in parentheses. However, due to the enormous amount of diagnostic information required in a full scale MES, this design model incorporates one physical structure for each of the five sub-systems of the propulsion system being modeled.

These five physical structures are stored as separate files on disk (refer to Appendix E). This allows the MES to load into memory only that portion of the knowledge base which is required. These structures could be further decomposed into smaller structures, if required, to further reduce the amount of main memory needed. This design feature provides an "overlay" capability for more effective use of the usually small main memory space, usually 64K, associated with microcomputers.

Each data structure, disk file for this MES knowledge base, occupies less than 1K bytes of disk storage. In fact, all five disk files occupy less than 4K bytes of storage. Based on the previously described amount of diagnostic information in this design model, the entire "-2" T.O. series for the F-4D aircraft will consume approximately 200K bytes of disk storage.

Several considerations must be given to the physical design of each data structure since each disk file will contain numerous questions and their associated corrective actions. But, how does the inference procedure decide which part of the structure is a "question" and which part is an "action"? Also, how can the structure be designed in order to logically represent the hierarchical tree structure for the knowledge base?

In order to design the structure of the MES knowledge base to incorporate questions and associated actions, several schemas were evaluated. One method, which is possibly the simplest to design, would require each question to be prefaced by a "Q" or some other identifier to enable the inference procedure to determine if the information is to be used as a question or action. This method would require more storage space and may require additional checks during diagnostic processing.

Another method, which is implemented in this model, allows the smallest parts of the data structure to be grouped by the "true" and "false" parts of each question. Table III-3 presented the logic used with the diagnostic questions from the T.O. manual. This logic provided a branch to another question or action based on the answer being either "yes" or "no". This is the same construct used in this design model. Since the actions are determined at the lowest level of the hierarchy, it seems plausible that the grouping of true and false parts should begin at that level. However, analysis of the design of this model MES indicates that the groupings can begin either at the top or bottom level.

Within this knowledge base the groupings are constructed from the top level, down the false branch nodes, and then to the true branch nodes. Each grouping of a question with its false and true parts constitutes a separate sub-tree of this production system. The rules by which these productions are accessed are embedded in the design characteristics of the knowledge base and in the logic of the inference engine/procedure. The "IF...THEN" logic characteristic of production systems or rule-based expert sytems is envoked in the question and answer sequence of this MES. For example, "IF" the answer to a question is 'true', "THEN" the procedure continues by examining data down the 'true' side of the tree structure. Otherwise, the 'false' side of the tree structure will be processed.

Each element within LISP must be enclosed in parentheses. This requires that each string of characters representing a question or action must be enclosed in parentheses. As previously discussed, LISP uses two functions, CAR and CDR, to access the left and right atoms/elements of a list. With this in mind, each logical construct of the tree structure for this MES knowledge base was designed with a node representing a question, followed by a node representing the left/false part, and terminated with a node representing the right/true part of the question. This primitive structure is recursively generated for each question and action grouping to the lowest level of each true and false branch of the diagnostic tree structure.

At any level of this tree structure there are only two possible routes. At any level a "true" route will only lead to more questions unless the corrective action is obtained. The same feature holds for the "false" route. Therefore, the grouping of these questions and actions is critically sensitive to ensure that proper traversal of the knowledge base occurs.

The groupings of these primitive structures further implements the basic design found in other production systems [BAR80, BAR82]. However, this design feature is implemented in this MES without the necessity of comparing several conditions, which must be satisfied prior to a decision to traverse a specific route, at each node of the tree. This MES requires only one conditional action at each node in the tree traversal. This small scale MES was designed based on the logic used in production systems, but it is actually implemented as a discrimination network. Therefore, redundancy of data comparison and cluttering of the knowledge base is eliminated.

Figure IV-1 displays the basic tree structure for one of the symptom index references of the F-4D propulsion system used in this design. This figure is drawn based on the questions and actions found in Table III-3. The questions at the top of this structure are the primary questions for this area and help to eliminate unneccessary traversals early in the diagnostic procedures.

Tree Structure for
ROUGH OR VIBRATING ENGINE

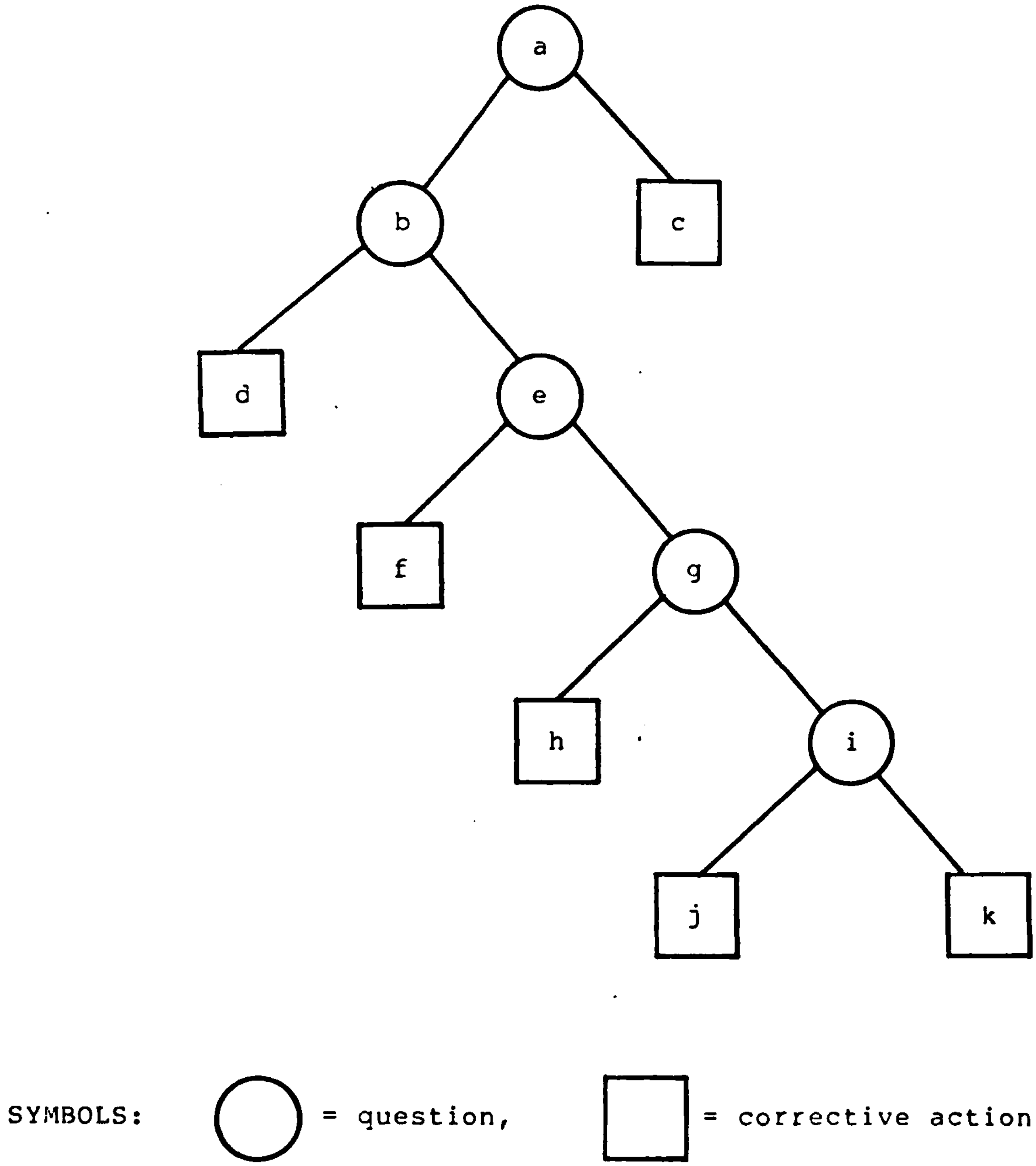


FIGURE IV-1

From this figure, the data structure is represented in its LISP form as follows:

((a (b d) (e f) (g h) (i j) k) (c))

With this basic format each data structure can easily be accessed by a combination of CAR and CDR LISP functions. Suppose a symbolic data name such as 'DSTRUC' is given to this data structure. To access the first question of this structure, 'a', the LISP statement is '(CAR (CAR DSTRUC))'. This effectively accesses the 'left element' of the 'left element'. In other words, a recursive process is being utilized working on each higher level element of the data structure from left to right. From that point the true and false branches are traversed by recursively using '(CDR (CAR DSTRUC))' for the false branches and '(CDR DSTRUC)' for the true branches.

As each level of the original sub-tree structure is traversed, the size of DSTRUC is actually being reduced. This recursive process continues until a terminal node of the tree is reached. At that point the most probable corrective action has been obtained.

Updates and modifications to this knowledge base are accomplished in much the same manner. The internal DBMS features of LISP allow the knowledge base elements to be easily modified through traversals of each sub-tree using the CAR and CDR functions repeatedly. The specific question or action to be modified is first located and then the updated information is written into the structure at that point thereby replacing the previous information. However, as previously discussed, this modification process should be carefully performed only by an expert technician who has authorization to modify the knowledge base according to new T.O. changes.

Inference Procedure and User Interface

The design of this procedure is based on the contents and structure of the knowledge base. The primary purpose of the MES is to act upon the knowledge base in regards to diagnostic evaluations and provide the user with a solution to the aircraft problem. Therefore, the design of the inference procedure must be centered on the knowledge base.

The small scale MES constructed in this design incorporates three primary modules, or functions as described in LISP terms. The modules associated with the heirarchical design of this MES are illustrated in Figure IV-2 through IV-4 and are also included as Appendix A. The first module, INIT, provides all preliminary housekeeping routines for all the other modules. A second module, DIAG, provides the actual diagnostic capabilities of the MES. This is the module of main emphasis in this design. The third module, KBMS, provides the MES with knowledge base modification capabilities.

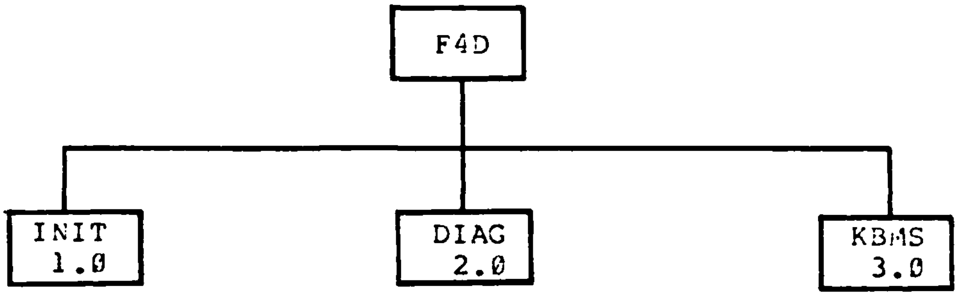


FIGURE IV-2. Maintenance Expert System

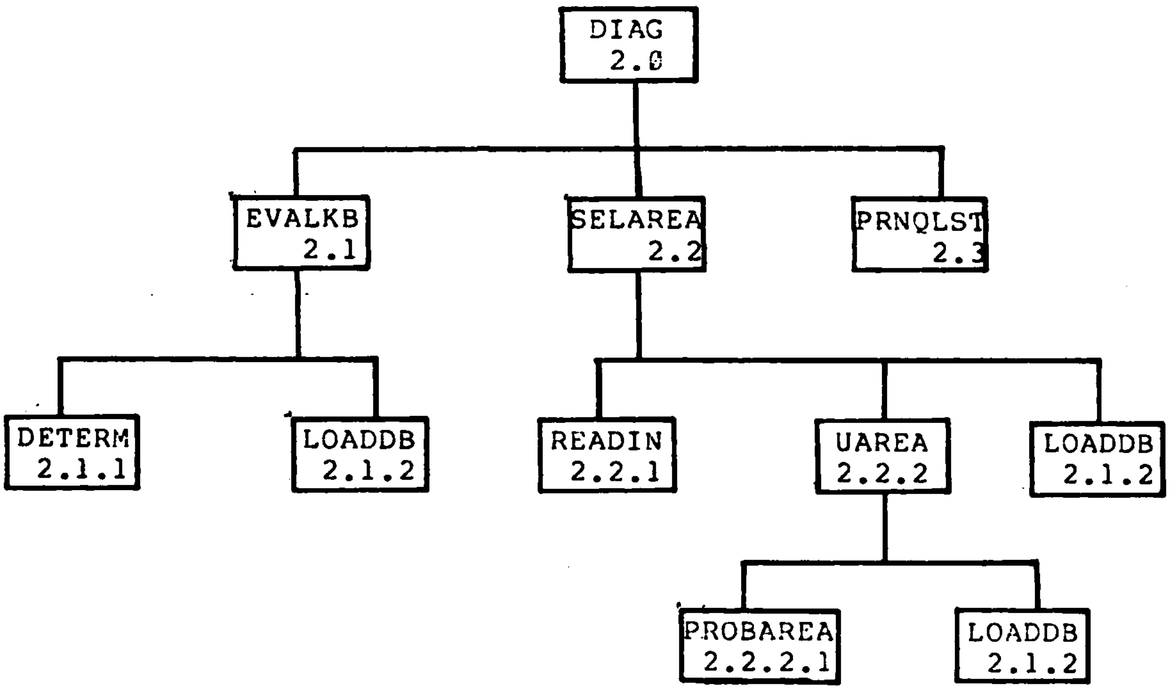


FIGURE IV-3. Aircraft Diagnostics

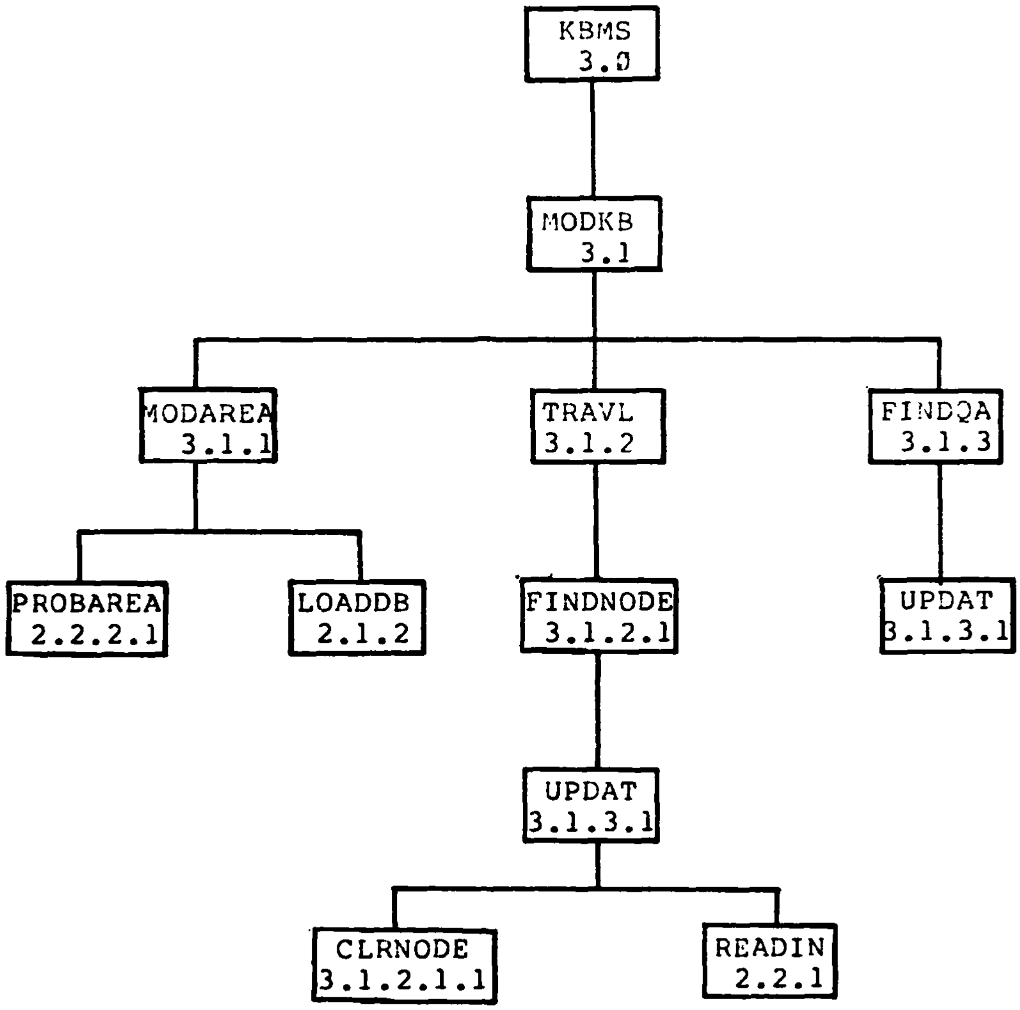


FIGURE IV-4. Knowledge Base Management System

Each of these prime modules rely on other sub-modules to accomplish their tasks. One such sub-module is LOADDB which loads the specified knowledge base information from disk. This module first determines which area of the knowledge base is currently loaded in memory and then decides if it needs to access the disk. This module is used by the several other major units of the MES except INIT.

The following scenario provides a view of how this author designed the inference procedure for this MES and provides ideas of how these functions could be expanded for a full scale MES. In addition to the diagnostic features of a MES, this design incorporates a knowledge base management facility to provide modifications to the diagnostic information. All output to the user is via the CRT, but may optionally be directed to the printer for hardcopy documentation.

Upon entry to the MES the user is provided with a message stating the capabilities of the MES. After reading this message, the user then selects the appropriate MES feature. For this scenario the "Aircraft Diagnostics" feature is chosen.

Since the MES is primarily designed to assist in isolating faulty components of the aircraft, the MES must have some knowledge of the problem encountered by the aircraft crew member. At this point the MES queries the user for the "problem statement". This statement can be entered exactly as written in the aircraft maintenance forms or can be an abbreviated version from the user. The MES will use this statement to determine which area of the knowledge base should be used to diagnose the problem.

A unique group of key words is associated with each area of the knowledge base. The MES analyzes the problem statement by matching each word of the problem statement on any of these key words. This gives the MES the appearance of being "intelligent" by having a pseudo natural language interface. A full scale MES should have a more sophisticated natural language interface.

The appropriate area of the knowledge base is loaded via two methods. If a key word match occurs, the MES then builds the knowledge base file name for that area and loads that knowledge base information into memory. If a key word match can not be made, the user is presented with a menu of "suspective" areas in which the diagnosis is to begin. It is at this point that the MES could not "understand" the problem statement in terms of locating a specific knowledge base area to begin diagnosis.

The actual file name for each area of the knowledge base is built as needed. Since there are only five knowledge base areas in this model, each file name terminates with a number in the range of 1 to 5. The file names used in this design are DB1 through DB5 as shown in Appendix E. As a key word is matched or a specified menu area is selected, the MES builds the file name by appending the associated matched area or menu number to "DB". This provides the full file name as stored on disk. This same procedure could be expanded in the full scale MES implementation.

Once the knowledge base information is loaded, the diagnosis actually begins. This MES engages the user in a dialogue of questions and answers until the corrective action is determined. A full scale MES may allow the user to input results of any system checks which have been made prior to envoking the MES for this particular problem. This would allow the MES to guide itself through the knowledge base and only query the user when it needed more information. The user input could be only a simple "true" or "false" response as in this MES or it could be in the form of specific values for certain aircraft system checks.

During this interactive dialogue the questions and answers are retained by the MES. Once the corrective action is determined, it is displayed to the user. The user is then asked if he/she would like to review the questions and answers used to arrive at the given corrective action. This feature may be helpful in determining if modifications to the knowledge base are necessary for some intermediate level in the question and answer sequence. This information could also be used by the knowledge base manager to ensure proper diagnosis is being accomplished according to current T.O.s for the aircraft.

This scenario is terminated by querying the user for another aircraft problem at which time the diagnostic process would repeat itself. If no further problems are to be diagnosed, the MES again prompts the user with a menu of the MES features. At this time the diagnostics feature can be entered again or another MES feature can be selected.

The user interface throughout this scenario is quite friendly. All messages are displayed on an 80-column display with special characters used to emphasize important information. With the development of this MES on an Apple II+ microcomputer, additional features such as sound are available. Sound in the form of varied frequencies and durations are used to attract the user's attention at specific points of the dialogue. For example, depressing a key which can not be determined as a true or false response will trigger an error message to be displayed and a warning sound to be heard. The added enhancement of graphic displays is also available on most microcomputers, but was not developed as part of this MES.

The disk and memory space required for this small scale MES were minimal. The disk space, discussed previously, will increase as the amount of diagnostic information in the knowledge base increases. The memory space requirements for the MES source code is approximately 14K of disk space. Once the source code is converted to object code for execution, storage requirements decrease to only 11K of memory. As the diagnostic process evolves, additional memory space is occupied by the information from the knowledge base. For this MES, each area of the knowledge base occupies only 1K of additional memory.

A word of caution is needed at this point! Most LISP programs are written with numerous recursive functions. For the implementation of a MES on a microcomputer, these recursive functions should be used with caution. Recursive calls to a function tend to quickly exhaust memory and memory is one limiting feature on a microcomputer. The MES implemented in this design uses both iterative and recursive functions without any problems.

The source code and associated documentation for this MES is listed in Appendix C. A user's guide and system requirements for operation of this MES on an Apple II+ microcomputer is included as Appendix D.

Summary

The MES designed and implemented in this chapter provides a basis for the development of a full scale MES to be used within any aircraft maintenance environment. Although the knowledge base for an operational MES will be much larger, the same data structure and similar inference procedure modules could be implemented.

V. Conclusions and Recommendations

Conclusions

As proposed in Chapter I, a general design for a maintenance expert system for use within the aircraft maintenance environment has been presented. A small scale system has been designed and implemented on a microcomputer to demonstrate that such a MES can be constructed. The structure charts, data-flow diagrams, and source code for this system are contained in Appendices A, B, and C respectively. A user's guide is supplied in Appendix D for operation of this system on an Apple II+ microcomputer using the CPM operating system. The structure and contents of the data files used in this model are included in Appendix E.

An analysis of the operation of this MES indicates that a large scale MES is feasible. The technology, both hardware and software, is currently available with microcomputers to make such a system a reality. Disk and resident memory capabilities on these machines seem to increase daily. Therefore, it appears feasible that a MES of any size could be constructed.

Recommendations

It is recommended that the complete design and implementation of an operational MES should be considered. Demonstrations of this small MES to local aircraft maintenance personnel were favorable and indicate that such a system is desirable within aircraft maintenance. In addition to the diagnostic capabilities, several other features are desired.

One feature that is greatly needed is the capability to provide general and specific technical information to the user. Aircraft technicians are faced daily with new challenges and desire to know more about their aircraft. As presented in Chapter III, a separate knowledge base could be provided with the MES to accomodate user queries for aircraft information. This would allow the technician to ask questions of the MES to increase his/her knowledge about the aircraft systems.

Other futuristic ideas and desires of a MES are astounding. As presented in Johnson's paper [JOH81], aircraft of the future will have their own diagnostic systems onboard the aircraft. In fact, this idea is currently a reality with the MADAR system of the C-5A transport aircraft. This system monitors numerous components and subsystems of the aircraft during flight. It gathers information such as airspeed, altitude ranges, stress forces, and data on component malfunctions. It has been refined to the point that it can even inform the maintenance technician that the air pressure is low in one of its 28 tires!

The MADAR system provides immediate and delayed information reporting. If a malfunction occurs during flight, the system determines what action to take. If the malfunction is detrimental to the flight, the system immediately warns the flight crew. Other malfunctions are simply stored on computer tape and downloaded after flight onto printouts for maintenance crews.

As Johnson points out, mobility conditions dictate that computers used in maintenance activities must operate in a "stand-alone" mode. These computers will perform their normal functions from their own "data bank" and may be linked to other larger systems by satellite communications.

The graphics capabilities necessary to fully desribe an aircraft system component must also be available. A technician, who is unfamiliar with specific maintenance procedures, could be shown the location of the component and be shown step-by-step procedures for removal and replacement of the component. Color, rotational positioning, and three-dimensional graphics will greatly enhance the MES.

The learning capability of a MES is a must. The AI capabilities of the system will allow it to make necessary adjustments in its troubleshooting strategies based on diagnostic successes and failures. According to Davis [DAV82], work in expert systems has successfully developed to the point that learning is possible. However, reconstruction of learned knowledge to fit other situations within the domain still lies on the frontier.

A MES with the aforementioned capabilities could also interface with other automated systems. Most aircraft maintenance organizations have some form of automated maintenance documentation facility. By interfacing the MES with this documentation system, an immediate accounting of maintenance activities could be accomplished. The maintenance records for all aircraft could be updated on a real-time basis. This information could then be used to provide commanders with the current status of all aircraft assigned to the organization.

Development and implementation of an expert system takes time. As Davis points out, "even for the best-understood problems, experienced researchers using the best-understood technologies still require at least five man-years to develop a system that begins to be robust". If this is actually the case, development of a MES with the above capabilities for use within the aircraft maintenance complexes should begin immediately!.

Bibliography

AI-M-332. Ideas About Management of LISP Data Bases. Cambridge, Massachusetts: Artificial Intelligence Laboratory, Massachusetts Institute of Technology, May 1975 (AD-A013312).

"Artificial Intelligence: The Second Computer Age Begins," Business Week, 66-75 (March 1982).

Barr, A. and J. Davidson. Representation of Knowledge. Stanford, California: Computer Science Department, Stanford University, March 1980. (AD-A089073).

Barr, A. and E.A. Feigenbaum. The Handbook of Artificial Intelligence. Stanford: Heuris Tech Press, 1982.

Berliner, H. "The B* Tree Search Algorithm: A Best-First Proof Procedure," Computer Science Department, Carnegie-Mellon University, 1981.

Charniak, E., et al. Artificial Intelligence Programming. Hillsdale, New Jersey: Lawrence Erlbaum, 1980.

Clancey, W.J., et al. Applications-oriented AI Research: Education. Stanford, California: Computer Science Department, Stanford University, July 1979. (AD-A075517).

Clocksin, W.F. and C.S. Mellish. Programming in PROLOG. New York: Springer-Verlag, 1981.

Davidson, J., et al. Natural Language Understanding. Stanford, California: Computer Science Department, Stanford University, July 1979. (AD-A076873).

Davis, R. "Expert Systems: Where are we? and where do we go from here?" Cambridge, Massachusetts: Artificial Intelligence Laboratory, Massachusetts Institute of Technology, June 1982 (AI memo 665).

Davis, R. and D. Lenat. Knowledge-based Systems in Artificial Intelligence. New York: McGraw-Hill, 1982.

Duda, R.O. and J.G. Gaschnig. "Knowledge-based expert Systems come of age," Byte, 6: 238-281 (September 1981).

Feigenbaum, E.A. Knowledge Engineering: The Applied Side of Artificial Intelligence. Stanford, California: Department of Computer Science, Stanford University, September 1980. (AD-A092574).

Ferrentino, A.B. Software Acquisition Manager's Knowledge-Based Expert System. Arlington, Virginia: Software Architecture and Engineering, Inc., June 1982. (AD-A117091).

Graham, N. Artificial Intelligence. Blue Ridge Summit, Pennsylvania: Tab Books, 1979.

Halpern, E. and J. Slagle. An Intelligent Control Strategy for Computer Consultation. NRL Memorandum Report 4789. Washington, D.C.: Information Technology Division, Naval Research Laboratory, April 1982. (AD-A113388).

Hofstadter, D.R. "Metamagical Themas," Scientific American, 248: 14-28 (January 1983).

Hofstadter, D.R. "Metamagical Themas," Scientific American, 248: 22-30 (February 1983).

Hofstadter, D.R. "Metamagical Themas," Scientific American, 248: 14-28 (March 1983).

Horn, B. K. and P. H. Winston. LISP. Reading, Massachusetts. Addison-Wesley Publishing Company, 1981

Johnson, R. C. "Integrated Maintenance Information System: An Imaginary Preview," Human Resources Laboratory Report. Wright-Patterson AFB, Ohio: AFHRL, September 1981 (Technical Paper 81-18).

McCarthy, J. "History of LISP", SIGPLAN Notices, 13: 217-223 (August 1978).

Miner, L. and R. Webster. "Expert Systems: Programming Problem-Solving," Technology, 62-73 (January 1982).

Minker, J. "An experimental relational data base system based on logic". In H. Gallaire and J. Minker, (Eds.), Logic and Databases. New York: Phenum, 1978.

Moore, R.C. Reasoning About Knowledge and Action. Stanford, California: Artificial Intelligence Laboratory, Stanford University, 1980.

Nau, D.S. "Expert Computer Systems," Computer, 2: 63-84 (February 1983).

Nilsson, N.J. Principles of Artificial Intelligence, Tioga, Palo Alto, California, 1980.

Robinson, J.A. and E.E. Sibert. Logic Programming in LISP. RADC report. New York: School of Computer and Information Science, Syracuse University, January 1981. (AD-A096042).

Sowa, I. F. "A Conceptual Schema for Knowledge-Based Systems," Journal of ACM, 6: 193-195 (January 1980).

Stefik, M., et al. "The Organization of Expert Systems, A Tutorial." Artificial Intelligence, 18,135-173, 1982.

Summers, P.D. "A Methodology for LISP Program Construction from Examples," Journal of ACM, 24: 161-175 (January 1977).

Use of Artificial Intelligence Methods for Avionics Maintenance, Training, and Diagnostics. Human Resources Laboratory report. Wright Patterson AFB, Ohio: Human Resources Laboratory, January 1983.

Winston, P.H. Artificial Intelligence. Reading, Massachusetts: Addison-Wesley Publishing Company, 1979.

Wood, R.J. The Direct LISP Approach to Functional Environmental Manipulation. TR-907. Report on LISP control structures. College Park, Maryland: Department of Computer Science, Maryland University, June 1980. (AD-A089671).

Yasaki, E.K. "AI Comes of Age," Datamation, 26: 48-54 (October 1980).

References

[AIS82] "Artificial Intelligence: The Second Computer Age Begins," Business Week, 66-75 (March 1982).

[BAR80] Barr, A. and J. Davidson. Representation of Knowledge. Stanford, California: Computer Science Department, Stanford University, March 1980. (AD-A089073).

[BAR82] Barr, A. and E.A. Feigenbaum. The Handbook of Artificial Intelligence. Stanford: Heuris Tech Press, 1982.

[CHA80] Charniak, E., et al. Artificial Intelligence Programming. Hillsdale, New Jersey: Lawrence Erlbaum, 1980.

[CLA79] Clancey, W.J., et al. Applications-oriented AI Research: Education. Stanford, California: Computer Science Department, Stanford University, July 1979. (AD-A075517).

[DAV82] Davis, R. "Expert Systems: Where are we? and where do we go from here?" Cambridge, Massachusetts: Artificial Intelligence Laboratory, Massachusetts Institute of Technology, June 1982 (AI memo 665).

[DUD81] Duda, R.O. and J.G. Gaschnig. "Knowledge-based expert Systems come of age," Byte, 6: 238-281 (September 1981).

[FEI80] Feigenbaum, E.A. Knowledge Engineering: The Applied Side of Artificial Intelligence. Stanford, California: Department of Computer Science, Stanford University, September 1980. (AD-A092574).

[HRL83] Use of Artificial Intelligence Methods for Avionics Maintenance, Training, and Diagnostics. Human Resources Laboratory report. Wright-Patterson AFB, Ohio: Human Resources Laboratory, January 1983.

[JOH81] Johnson, R. C. "Integrated Maintenance Information System: An Imaginary Preview," Human Resources Laboratory report. Wright-Patterson AFB, Ohio: AFHRL, September 1981 (Technical Paper 81-18).

[NAU83] Nau, D.S. "Expert Computer Systems," Computer, 2: 63-84 (February 1983).

[NIL80] Nilsson, N.J. Principles of Artificial Intelligence, Tioga, Palo Alto, California, 1980.

[WIN79] Winston, P.H. Artificial Intelligence. Reading, Massachusetts: Addison-Wesley Publishing Company, 1979.

[YAS] Yasaki, E.K. "AI Comes of Age," Datamation, 26: 48-54 (October 1980).

APPENDIX A:

Maintenance Expert System

Structure Charts

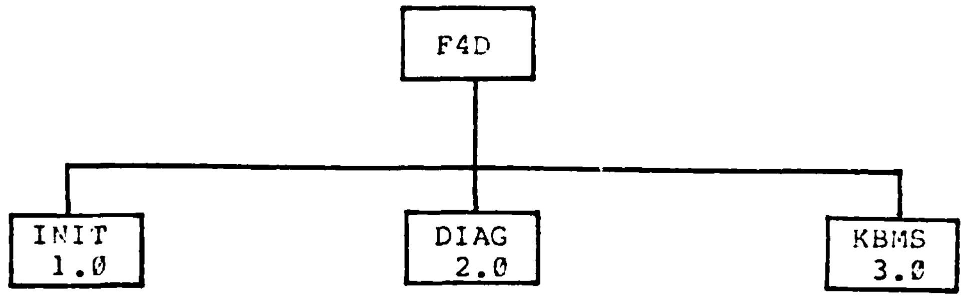


FIGURE IV-2. Maintenance Expert System

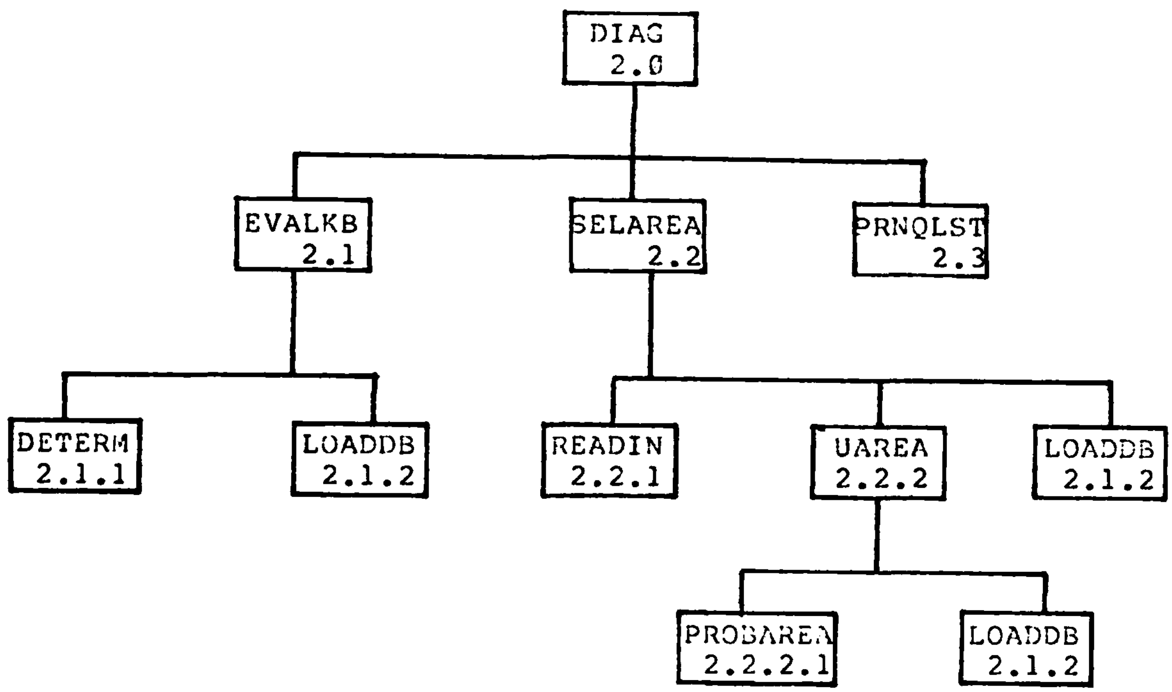


FIGURE IV-3. Aircraft Diagnostics

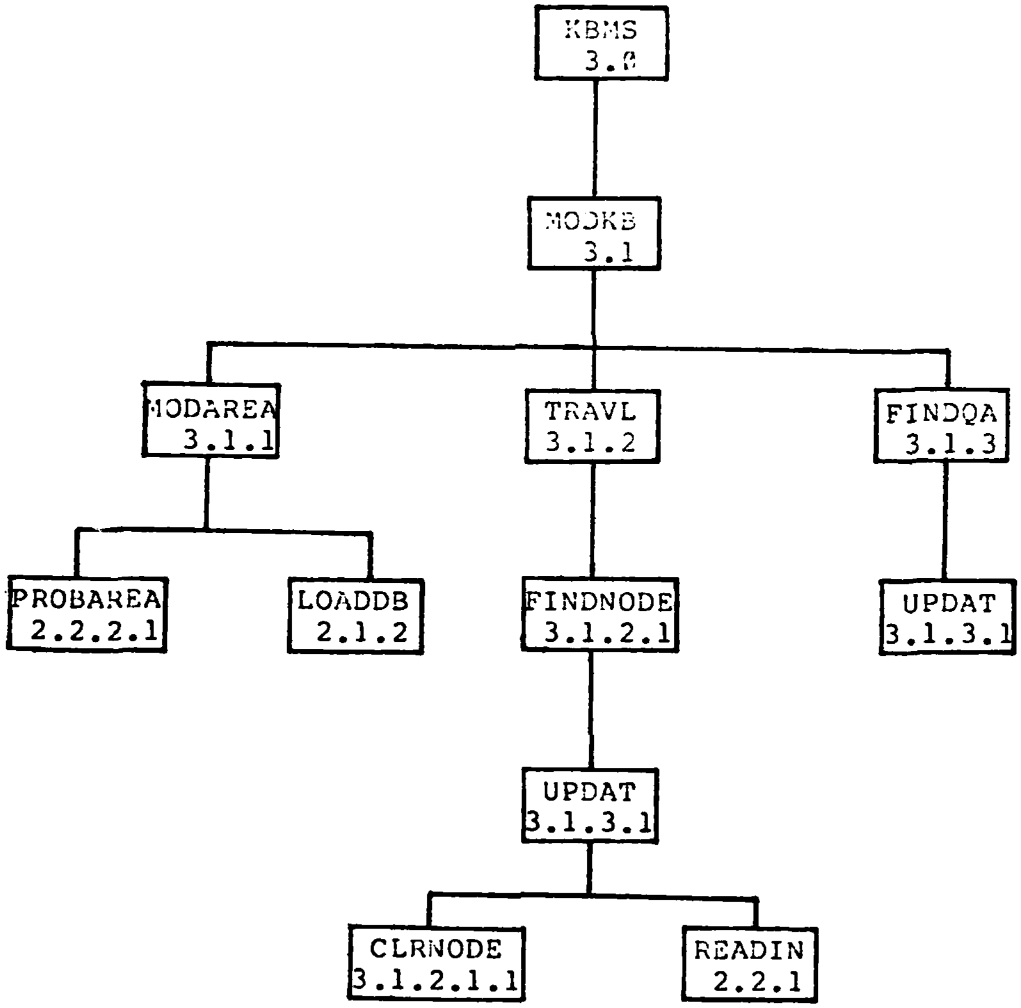


FIGURE IV-4. Knowledge Base Management System

APPENDIX B:

Data Flow Diagrams

for

Maintenance Expert System

The following data flow diagrams illustrate the operational flow of information within the MES. An overview of the system is shown in figure B-1. The remaining figures show the logical paths of data flow for each system module. The module numbers correspond to the module numbers within the heirarchy structure illustrated in Appendix A.

The interface shown as 'Technician' represents the user of the system at a terminal. Messages and data are displayed to the user on a CRT monitor. User responses are input via the terminal keyboard.

The 'Knowledge Base' represents the disk storage area of the technical order information being used within the MES. This knowledge base consists of several files as discussed in Chapter IV of this text.

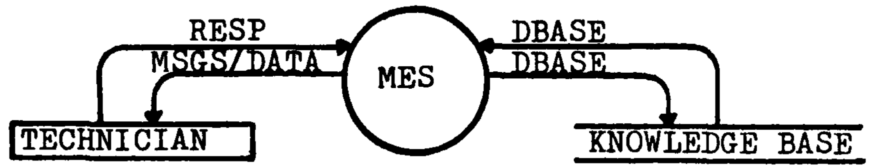
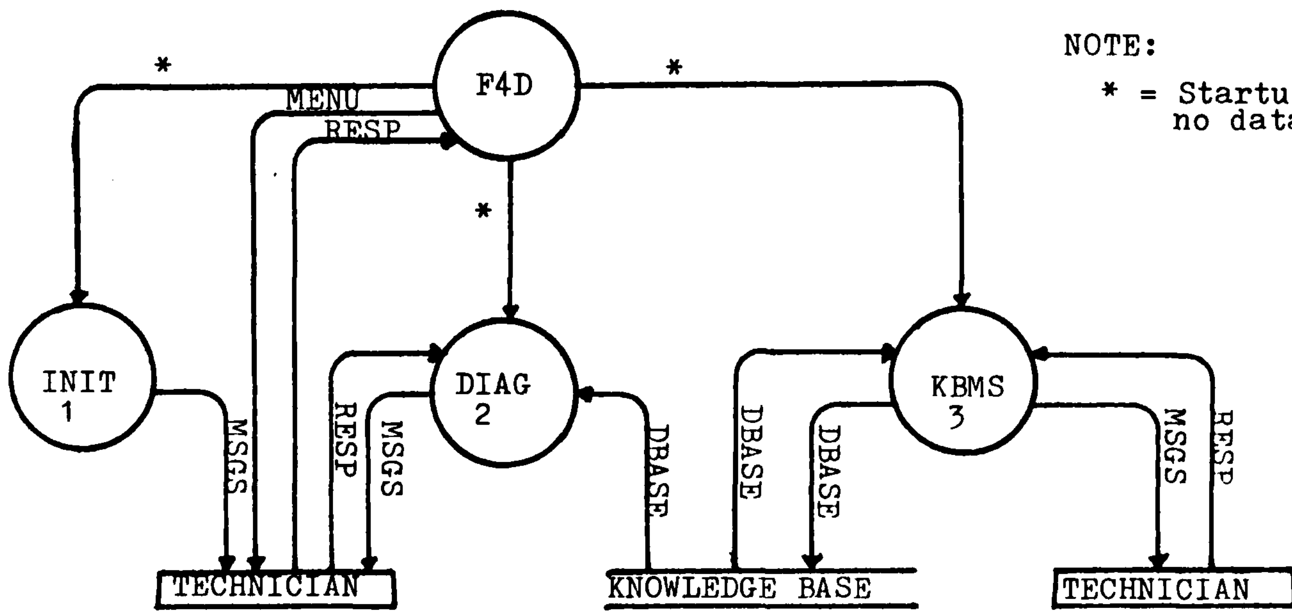


Figure B-1. MES Overview



NOTE:

* = Startup only, no data passed

Figure B-2. MES Driver Modules

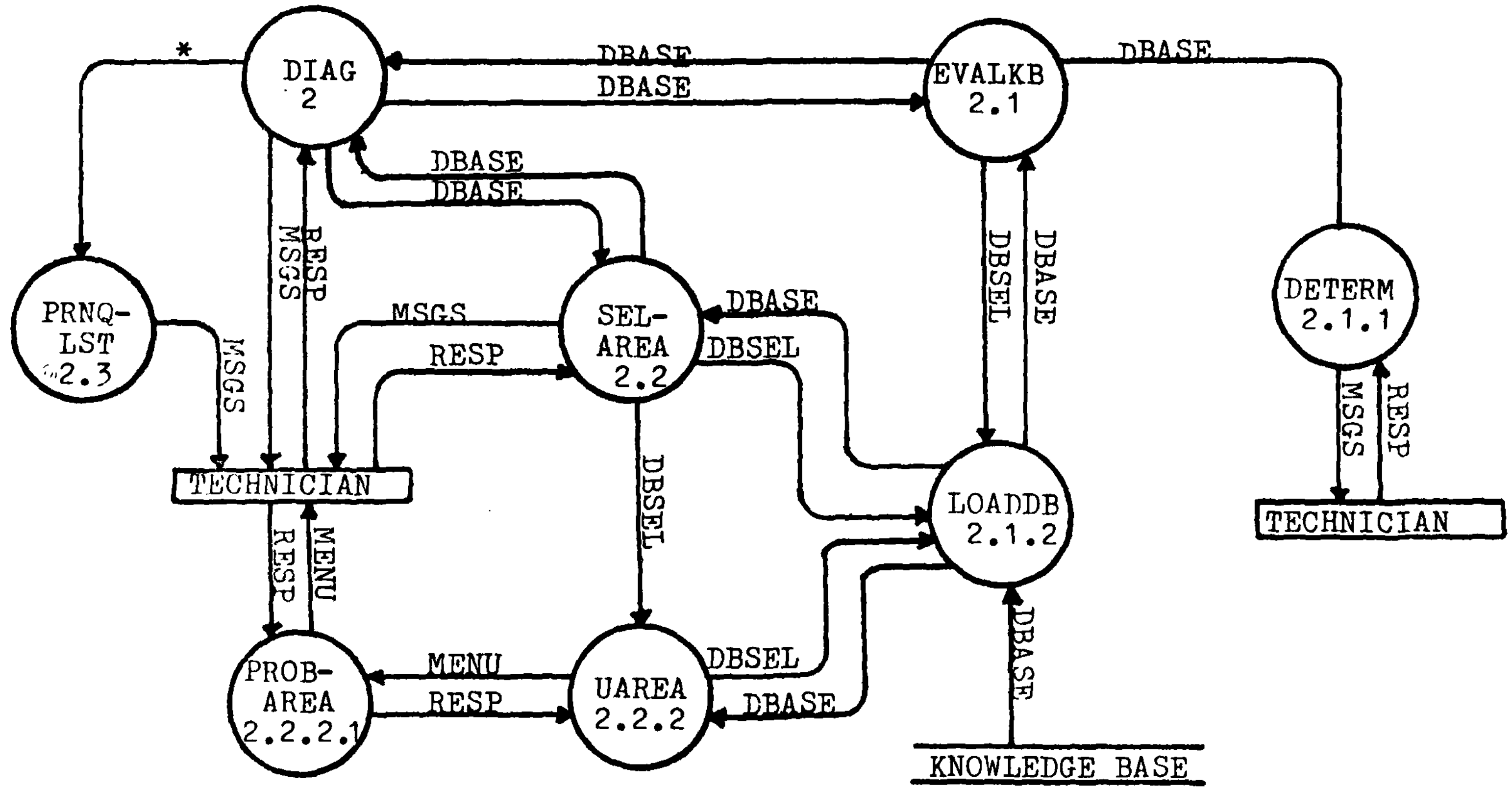


Figure B-3. Aircraft Diagnostics

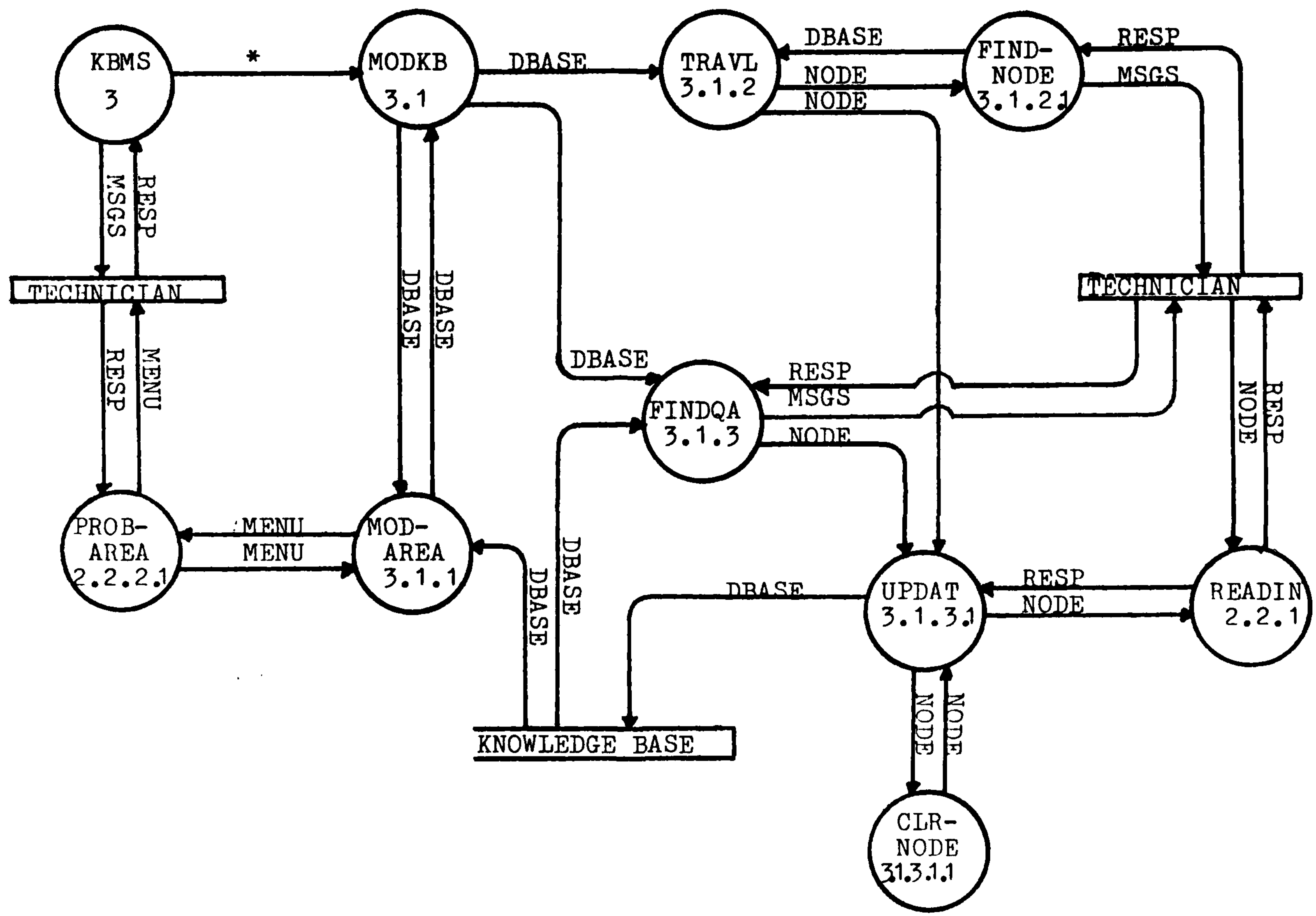


Figure B-4. Knowledge Base Management System

APPENDIX C:

LISP Source Code

for

Maintenance Expert System

```
%  ******************** F4TEST.LIB ****************  %


% MAINTENANCE EXPERT SYSTEM FOR DIAGNOSTIC ASSISTANCE ON THE F-4D AIRCRAFT.
   DEVELOPED ON AN APPLE II+ MICROCOMPUTER USING THE CPM OPERATING SYSTEM.
   LANGUAGE --- LISP   (MULISP-80 BY MICROSOFT, VERSION 2.15, 1982)
%


% FUNCTION:  INITIALIZES THE GLOBAL VARIABLES AND LISTS USED BY MOST OF
             THE SYSTEM MODULES.  IT ALSO PROVIDES THE USER WITH INSTRUCTIONS
             FOR OPERATION OF THE SYSTEM.

   MODULES CALLED:  NONE
   CALLING MODULES: F4D
   MODULE NUMBER:   1.0
%

(DEFN INIT (LAMBDA ()
(SETQ B1 (AFTERBURNER IGNITER SPRAYBARS AB A/B LIGHT))
(SETQ B2 (VIBRATES VIBRATION ROUGH VIBRATING COMPRESSOR))
(SETQ B3 (OVERTEMP EGT TEMPERATURE FUEL SLOW))
(SETQ B4 (STARTUP NO-START START FALSE IGNITION IGNITERS))
(SETQ B5 (OIL PRESSURE LEAKAGE CONSUMPTION))
(SETQ A1 (AFTERBURNER))
(SETQ A2 (COMPRESSOR))
(SETQ A3 (FUEL))
(SETQ A4 (IGNITION))
(SETQ A5 (LUBRICATION))
(SETQ TERM (!))
 (SETQ DBASES (QUOTE (1 2 3 4 5)))  % NUMBERS ASSOCIATED WITH DATABASE AREAS %
 (SETQ DWSSAP EE799)      % PROTECTION FOR KNOWLEDGE BASE %
 (SETQ TRESP (QUOTE (T TRUE Y YES YO YEA)))     % VALID 'TRUE/YES' USER RESPONSES %
 (SETQ FRESP (QUOTE (F FALSE N NO NIL NONE)))   % VALID 'FALSE/NO' RESPONSES %
 (SETQ VALRESP (NCONC (QUOTE (T TRUE Y YES YEA YO)) FRESP))  % ALL RESPONSES %
 (SETQ MVAL (0 1 2 3 4 5))              % MENU VALUES %
 (LINELENGTH 79)                        % SETUP FOR 80 COLUMN SCREEN %
 (CLRSCRN) (SOUND 9 15)
 (DISPLAY (QUOTE (> > > > > F - 4 D"  "E X P E R T"  "S Y S T E M < < < < <)))
(TERPRI 2)
(DISPLAY (QUOTE (* * * * * TECHNICAL ORDER SERIES ---  1F-4D-2-8  * * * * *)))
 (TERPRI) (DISPLAY (QUOTE (= = = = = = = =  F E A T U R E S  = = = = = = = =)))
 (TERPRI 2)  (DISPLAY (QUOTE (-- AIRCRAFT DIAGNOSTICS --- THIS FEATURE WILL ASSIST WITH)))
 (DISPLAY (QUOTE (TROUBLESHOOTING THE 'PROPULSION SYSTEM'"."  THE DATABASE INFORMATION FOR)))
 (DISPLAY (QUOTE (THIS SYSTEM IS TAKEN DIRECTLY FROM TECHNICAL ORDER "1F-4D-2-8.")))
 (TERPRI)
 (DISPLAY (QUOTE (-- KNOWLEDGE BASE MANAGER --- THIS FEATURE MUST BE USED ONLY BY A)))
 (DISPLAY (QUOTE (HUMAN 'EXPERT' FOR THE F-4D AIRCRAFT"!"  THE DATABASE SHOULD ONLY BE)))
 (DISPLAY (QUOTE (CHANGED OR UPDATED WHEN CHANGES TO THE T"."O"." ARE APPLICABLE".")))
 (DISPLAY (QUOTE (N O T E:  ACCESS TO THIS FEATURE ONLY BY SECURITY PASSWORD)))
 (TERPRI 2)
 (DISPLAY (QUOTE (=======  PLEASE PRESS ANY KEY AND 'RETURN' TO CONTINUE  =======)))
 (READ) ))
```

```
% FUNCTION:  PERFORMS THE ACTIONS OF AN 'INFERENCE PROCEDURE. IT
  CONTROLS STARTING AND STOPPING OF THE "EXPERT SYSTEM", AS WELL AS ACCESS
  TO THE FUNCTIONAL MODULES OF THE SYSTEM.
   MODULES CALLED: INIT, DIAG, KBMS
   CALLING MODULES: NONE
   MODULE NUMBER: (SYSTEM DRIVER)
%

(DEFN F4D (LAMBDA ()
  (INIT)
  (LOOP
     (CLRSCRN)
      (DISPLAY (QUOTE (--- PLEASE SELECT A NUMBER TO ENTER/EXIT THE SYSTEM ---)))
   (LOOP
      (TERPRI)
      (DISPLAY (QUOTE (0 = EXIT THE SYSTEM)))
      (DISPLAY (QUOTE (1 = AIRCRAFT DIAGNOSTICS)))
      (DISPLAY (QUOTE (2 = KNOWLEDGE BASE MANAGEMENT)))
      (SETQ SECT (READ))
      ((MEMBER SECT (QUOTE (0 1 2))) T)
      (SOUND 8 15)
      (TERPRI)
      (DISPLAY (QUOTE (* * * SELECT ONLY FROM THE MENU GIVEN * * *)))
      (DEELAY 222) (CLRSCRN))      %END OF INNER LOOP %
    (COND ((EQUAL SECT 1) (DIAG))
          ((EQUAL SECT 2) (KBMS)))
    ((EQUAL SECT 0)
```

```
     (DISPLAY (QUOTE (*********** PROGRAM EXIT *********** ))) T))))

% FUNCTION:  DETERMINES THE APPROPRIATE AREA OF THE ENGINE SYSTEM TO
             BE DIAGNOSED VIA USER INPUT.  LOADS THAT AREA OF THE DATABASE
             AND CALLS "DETERM" TO NAVIGATE THROUGH THE PRODUCTIONS TO
             FIND A SOLUTION TO AN AIRCRAFT PROBLEM.
  MODULES CALLED: EVALKB, SELAREA, PRNQLST
  CALLING MODULES: F4D
  MODULE NUMBER: 2.0
%

(DEFN DIAG (LAMBDA ()
(CLRSCRN)
(DISPLAY ("     "***** F-4D  PROPULSION SYSTEM DIAGNOSTICS  *****))
(TERPRI 2)
(LOOP
  (SETQ Q-LST NIL)
  (SETQ MENU 99)       % NULL ENTRY IN MENU %
  (SELAREA)       % GET PROBLEM STATEMENT AND SELECT PORTION OF DATABASE %
 (COND ((NOT (EQUAL MENU 0))
 (SETQ RSLT NIL)
   (CLRSCRN)
   (DISPLAY (QUOTE (> > > > > PLEASE ANSWER THESE QUESTIONS  < < < < <)))
   (TERPRI)
  (EVALKB )    % EVALUATE THE KNOWLEDGE BASE AREA SELECTED %
  (TERPRI)
  (SOUND 21 12)
  (DISPLAY (========= ACTION RECOMMENDED TO CORRECT AIRCRAFT PROBLEM =========))
  (TERPRI) (DISPLAY (CAR RSLT)) (TERPRI)
  (DISPLAY (=================================================================))
  (TERPRI 2)
  (DISPLAY (???  WOULD YOU LIKE TO REVIEW THE QUESTIONS ASKED ???))
  (COND ((MEMBER (READ) TRESP)
      (CLRSCRN) (PRN-Q-LST))
      (T (CLRSCRN)))
  (TERPRI)))          % END OF OUTER CONDITION %
  (TERPRI 2)
  (DISPLAY (QUOTE (??? WOULD YOU LIKE TO DIAGNOSE ANOTHER AIRCRAFT PROBLEM ???)))
  ((NOT (MEMBER (READ) TRESP)) T)
  (CLRSCRN))                         % END OF LOOP %

 (CLRSCRN)))
```

```
% FUNCTION: CALLS THE ROUTINE 'DETERM' TO TRAVERSE THE SELECTED KNOWLEDGE BASE
            AREA.  IF THE CORRECTIVE ACTION IS NOT FOUND IN THAT AREA, ANOTHER
            KNOWLEDGE BASE AREA IS LOADED AND THE ROUTINE IS REPEATED.
   MODULES CALLED: DETERM, LOADDB
   CALLING MODULES: DIAG
   MODULE NUMBER: 2.1
%

(DEFN EVALKB (LAMBDA ()
  (DETERM )                     % DETERMINE SOLUTION TO PROBLEM %
((NOT (MEMBER (CAAR RSLT) DBASES)) T)
                  % SETUP FOR LOADING NEW AREA OF KNOWLEDGE BASE %
  (SETQ DBSEL (PACK (D (PACK (B (CAAR RSLT))))))  % GET NEW DATABASE AREA NAME %
  (SOUND 10 15)
  (SETQ A1-5 (CAR (PACK (A (CAAR RSLT)))))
  (DISPLAY (QUOTE
  (*** CORRECTIVE ACTION CAN NOT BE FOUND IN THIS AREA ***)))
  (TERPRI)
  (DISPLAY (APPEND (QUOTE (** TRY THE ))
           (APPEND (CAR A1-5) (QUOTE ( SUB-SYSTEM **)))))
  (LOADDB)                     % LOAD THE NEW DATABASE AREA     %
  (EVALKB)         % REPEAT THE EVALUATION ROUTINE %
  (SETQ PREDBA DBSEL)))   % REMEMBER LAST KNOWLEDGE BASE AREA LOADED %

% FUNCTION: OBTAINS SUSPECTED KNOWLEDGE BASE AREA FROM USER %
%  MODULES CALLED: PROBAREA, LOADDB
   CALLING MODULES: SELAREA
   MODULE NUMBER: 2.2.2
%
(DEFN UAREA (LAMBDA ()
(TERPRI 2) (SOUND 9 18)
(DISPLAY "MORE INFORMATION IS NEEDED TO DETERMINE WHICH SUBSYSTEM TO TROUBLESHOOT!!")
   (TERPRI 2)
    (DISPLAY (QUOTE (???  IN WHICH AREA DO YOU SUSPECT THE PROBLEM TO BE LOCATED ???)))
   (PROBAREAS)         % ASK USER FOR HELP IN DETERMINING PROBLEM AREA %
  ((EQUAL MENU 0) T)               % EXIT IF 0 %
   (CLRSCRN)
    (SETQ DBSEL (PACK (D (PACK (B MENU)))))   % BUILD DATABASE NAME %
    (LOADDB)             % LOAD SELECTED DATABASE %
  ))             % END OF UAREA %
```

```
% FUNCTION:  QUERY USER FOR KNOWLEDGE BASE AREA TO USE WITH
               DIAGNOSTICS AND KNOWLEDGE BASE MODIFICATIONS.
   MODULES CALLED: NONE
   CALLING MODULES: UAREA, MODAREA
   MODULE NUMBER: 2.2.2.1
%

(DEFN PROBAREAS (LAMBDA ()
    (TERPRI)
    (DISPLAY "==== PLEASE SELECT THE DESIRED AREA:  ====")
   (LOOP
    (TERPRI)
                          % DETERMINE AREA FOR DIAGNOSIS %
    (DISPLAY (QUOTE (0 = RETURN TO START OF SYSTEM)))
    (DISPLAY (QUOTE (1 = AFTERBURNER AREA)))
    (DISPLAY (QUOTE (2 = COMPRESSOR AREA)))
    (DISPLAY (QUOTE (3 = FUEL AREA)))
    (DISPLAY (QUOTE (4 = IGNITION AREA)))
    (DISPLAY (QUOTE (5 = LUBRICATION AREA)))
    (DISPLAY (QUOTE (?? YOUR CHOICE ??)))
    (SETQ MENU (READ))
    ((MEMBER MENU MVAL) T)           % DETERMINE IF SELECTION IS ON MENU %
    (SOUND 7 21)
    (DISPLAY (QUOTE (* * *  PLEASE MAKE YOUR SELECTION FROM THE MENU PROVIDED  * * *)))
    (DEELAY 200) (CLRSCRN))    % END OF INNER LOOP %
   ))      % END OF PROBAREAS %




% FUNCTION:  LOAD SELECTED KNOWLEDGE BASE AREA FROM DISK      %
%  MODULES CALLED: NONE
   CALLING MODULES: UAREA, EVALKB, MODAREA
   MODULE NUMBER: 2.1.2
%

(DEFN LOADDB (LAMBDA ()
  (COND ((NOT (EQUAL DBSEL PREDBA))  %CHECK PREVIOUSLY LOADED DATABASE %
      (TERPRI)
      (DISPLAY (NCONC
        (** PLEASE WAIT WHILE LOADING:  ) (CONS DBSEL LIB)))
      (SOUND 7 14)
      (SETQ DBASE (READ (RDS DBSEL LIB)))  % LOAD SELECTED KNOWLEDGE BASE AREA %
      (RDS))              % RETURN INPUT CONTROL TO KEYBOARD %
      (T (DISPLAY "*** KNOWLEDGE BASE AREA CURRENTLY LOADED ***")
         (DEELAY 225)))
  (SETQ PREDBA DBSEL)      % REMEMBER LAST DATABASE LOADED %
  ))              % END OF LOADDB %
```

```

% FUNCTION:  INTERACTS WITH THE USER TO DETERMINE A SOLUTION TO THE AIRCRAFT
               PROBLEM.  THIS IS THE 'PRODUCTION INTERPRETER' FOR THE SYSTEM.
   MODULES CALLED: NONE
   CALLING MODULES: EVALKB
   MODULE NUMBER: 2.1.1
%

(DEFN DETERM (LAMBDA ()                    % PRODUCTIONS USED FOR DIAGNOSIS %
                                           % ARE OBTAINED FROM CURRENT DATABASE %
  (SETQ PROD DBASE)
   (LOOP
      (SETQ QUES (CAAR PROD))   % FIND QUESTION PART %
      (COND ((EQUAL (CADR QUES) "")
             (SETQ QUES (CAR QUES)))
            ((NULL (CDR QUES))
             (SETQ QUES (CAR QUES))))
      (LOOP
        (DISPLAY (APPEND QUES (QUOTE (???))))
        (TERPRI)
        (SETQ RESP (READ))             % GET ANSWER FROM USER %
        ((MEMBER RESP VALRESP)         % CHECK FOR VALID ANSWER %
          ((MEMBER RESP TRESP)
              (SETQ PROD (CDR PROD)))       % TRUE PART OF PRODUCTION %
          (T (SETQ PROD (CDAR PROD))))      %FALSE PART OF PRODUCTION %
        (SOUND 7 21)
        (DISPLAY (QUOTE (* * * *  I DID NOT UNDERSTAND YOUR ANSWER  * * * *)))
        (TERPRI)  (DISPLAY (QUOTE
        (* * *"  "THE ONLY VALID ANSWERS WHICH I UNDERSTAND ARE ":")))
        (DISPLAY (NCONC (QUOTE ("     ")) VALRESP))
        (TERPRI))                                  % END OF INNER LOOP %
      (SETQ L-LST (APPEND (QUOTE (QUESTION ==)) QUES))
      (SETQ R-LST (APPEND (QUOTE (? ---- ANSWER ==)) (RESP)))
      (SETQ QR-LST (APPEND L-LST R-LST))
      (SETQ Q-LST (CONS QR-LST Q-LST))     %SAVE QUESTIONS AND ANSWERS  %
      (SETQ RSLT PROD)             % SAVE RESULT %
      ((NULL (CDR PROD)) PROD)      % RESULT IS FOUND ON TRUE SIDE %
      ((NULL (CDAR PROD)) PROD))))) % RESULT FOUND ON FALSE SIDE %
```

```
% FUNCTION:  SHOWS THE LIST OF QUESTIONS ASKED AND THE ANSWER INPUT BY
             THE USER.  THIS MAY BE HELPFUL IN DETERMINING ALTERNATE
             SOLUTIONS FOR UPDATE TO THE DATABASE.
   MODULES CALLED: NONE
   CALLING MODULES: DIAG
   MODULE NUMBER: 2.3
%

(DEFN PRN-Q-LST (LAMBDA ()
 (DISPLAY (QUOTE (< < < <  QUESTIONS AND ANSWERS USED TO DETERMINE SOLUTION  > > > >)))
 (TERPRI)
 (SETQ HLD (REVERSE Q-LST))
 (LOOP
   ((NULL HLD) T)
   (DISPLAY (CAR HLD)) (TERPRI)
   (SETQ HLD (CDR HLD)))))








% FUNCTION: KNOWLEDGE BASE MANAGEMENT DRIVER ROUTINE.  QUERIES
            USER FOR VALID PASSWORD AND ENVOKES MODIFY ROUTINES.
   MODULES CALLED: MODKB
   CALLING MODULES: F4D
   MODULE NUMBER: 3.0
%

 (DEFN KBMS (LAMBDA ()
   (SOUND 15 20)  (TERPRI)
   (DISPLAY (QUOTE (PASSWORD ???)))
   ((EQUAL (READ) DWSSAP)   % MATCH ON USER PASSWORD %
      (MODKB))
    (DISPLAY (QUOTE
     (*** INVALID PASSWORD, NO ACCESS TO KNOWLEDGE BASE ***)))
   (SOUND 30 50)  (DEELAY 100)))
```

```
% FUNCTION: DETERMINE WHICH METHOD OF MODIFICATION IS DESIRED.
   MODULES CALLED: MODAREA, TRAVL, FINDQA
   CALLING MODULES: KBMS
   MODULE NUMBER: 3.1
%

(DEFN MODKB (LAMBDA ()
 (CLRSCRN) (DISPLAY "= = = KNOWLEDGE BASE MANAGEMENT OPTIONS: ") (TERPRI)
 (DISPLAY (QUOTE (0 = EXIT)))
 (DISPLAY (QUOTE (1 = REVIEW QUESTIONS TO THE POINT OF MODIFICATION)))
 (DISPLAY (QUOTE (2 = PROCEED DIRECTLY TO THE POINT OF MODIFICATION)))
 (TERPRI)  (DISPLAY (QUOTE (WHICH ?)))
 (SETQ KBACT (READ))
 ((EQUAL KBACT 1)
   (MODAREA) (TRAVL) (MODKB))
 ((EQUAL KBACT 2)
    (MODAREA) (FINDQA) (MODKB))
 ((NOT (EQUAL KBACT 0))
   (DISPLAY "*** INVALID SELECTION, TRY AGAIN ***")
   (SOUND 10 20)    (DEELAY 200)
   (MODKB))))

% FUNCTION: QUERIES USER FOR KNOWLEDGE BASE ARE TO BE
            USED DURING MODIFICATION.
   MODULES CALLED: PROBAREA, LOADDB
   CALLING MODULES: MODKB
   MODULE NUMBER: 3.1.1
%

(DEFN MODAREA (LAMBDA ()
(SETQ MENU 0)
(PROBAREAS)       % SHOW LIST OF KNOWLEDGE BASE AREAS %
((NOT (EQUAL MENU 0))
(SETQ DBSEL (PACK (DB MENU)))
(LOADDB))))
```

```
% FUNCTION: PROVIDE USER WITH DIRECTIONS FOR MODIFICATION.
   MODULES CALLED: FINDNODE
   CALLING MODULES: MODKB
   MODULE NUMBER: 3.1.2
%

(DEFN TRAVL (LAMBDA ()
 (CLRSCRN)
(DISPLAY (QUOTE
  (*** ANSWER THE FOLLOWING QUESTIONS UNTIL THE POINT IS REACHED ***)))
(DISPLAY (QUOTE
  (*** AT WHICH MODIFICATION IS TO BE MADE".  "AT THAT TIME TYPE A ***)))
(DISPLAY (QUOTE
  (***"  "'Q' IF THE QUESTION IS TO BE MODIFIED"," OTHERWISE NO "  "***)))
(DISPLAY (QUOTE
  (*** MODIFICATION WILL BE MADE AND QUESTIONS WILL CONTINUE TO  ***)))
(DISPLAY (QUOTE
  (*** BE ASKED UNTIL A 'CORRECTIVE ACTION' IS DETERMINED".   " AT ***)))
(DISPLAY (QUOTE
  (*** THAT POINT THE CORRECTIVE ACTION CAN BE MODIFIED".        "***)))
(TERPRI)
(DISPLAY (QUOTE
  (== PRESS ANY KEY AND 'RETURN' TO CONTINUE ==)))
 (READ) (CLRSCRN)
 (SETQ DQUES "CORRECTIVE ACTION")
 (FINDNODE DBASE)))

% FUNCTION: RECURSIVELY TRAVERSE THE KNOWLEDGE BASE AREA
             UNTIL THE SELECTED QUESTION/ACTION IS FOUND.
   MODULES CALLED: UPDAT
   CALLING MODULES: TRAVL
   MODULE NUMBER: 3.1.2.1
%

(DEFN FINDNODE (LAMBDA (NODE)
((NULL (CDR NODE)) (UPDAT (CAR NODE)))
((NULL (CDAR NODE)) (UPDAT (CAR NODE)))
 (COND ((EQUAL (CADR (CAAR NODE)) "")
        (SETQ QUEST (CAAAR NODE)))
       ((NULL (CDR (CAAR NODE)))
        (SETQ QUEST (CAAAR NODE)))
       (T (SETQ QUEST (CAAR NODE))))
 (DISPLAY (APPEND QUEST "???")) (TERPRI)
 (SETQ RESP (READ))
 (COND ((MEMBER RESP TRESP)
        (FINDNODE (CDR NODE)))
      ((MEMBER RESP FRESP)
        (FINDNODE (CDAR NODE)))
  ((EQUAL RESP Q)
    (SETQ DQUES "QUESTION")
    (UPDAT (CAAR NODE)))
 (T  (SOUND 8 23)
  (DISPLAY "*** THAT IS NOT A VALID RESPONSE, TRY AGAIN ***")
  (TERPRI)
```

```
% FUNCTION: QUERY USER TO DETERMINE IF CORRECT QUESTION OR
            ACTION HAS BEEN FOUND AND MODIFY IF DESIRED.
   MODULES CALLED: NONE
   CALLING MODULES: FINDNODE, FINDQA
   MODULE NUMBER: 3.1.3.1
%

(DEFN UPDAT (LAMBDA (OLDNODE)
 (CLRSCRN) (SOUND 13 13)
 (DISPLAY (APPEND "= = =  " (APPEND DQUES "TO BE MODIFIED: ")))
 (TERPRI)
 (COND ((EQUAL (CADR OLDNODE) "")
        (SETQ DNODE (CAR OLDNODE)))
       ((NULL (CDR OLDNODE))
        (SETQ DNODE (CAR OLDNODE)))
       (T (SETQ DNODE OLDNODE)))
 (DISPLAY DNODE) (TERPRI)
 (DISPLAY (QUOTE
  (??? REPLACE THIS INFORMATION WITH NEW INFORMATION ???)))
((MEMBER (READ) TRESP)
    (CLRNODE OLDNODE)
    (DISPLAY
   "== PLEASE ENTER THE NEW INFORMATION AND TERMINATE WITH A ! ==")
    (TERPRI)
    (SET OLDNODE (READIN))
    (SOUND 10 15) (TERPRI 2)
    (DISPLAY "*** PLEASE WAIT WHILE KNOWLEDGE BASE IS UPDATED ***")
  (WRS DBSEL LIB) (PRINT DBASE) (WRS))
  (DISPLAY (QUOTE (*** MODIFICATION TERMINATED ***)))
  (TERPRI 3)))




% FUNCTION: QUERIES USER FOR UNIQUE KNOWLEDGE BASE INFORMATION
            TO BE MODIFIED.  PROCEEDS DIRECTLY TO THAT NODE.
   MODULES CALLED: UPDAT
   CALLING MODULES: MODKB
   MODULE NUMBER: 3.1.3
%

(DEFN FINDQA (LAMBDA ()
 (CLRSCRN)
 (DISPLAY "*** THIS FEATURE IS NOT FULLY OPERATIONAL, PRESS ANY ***")
 (DISPLAY "*** KEY AND 'RETURN' TO CONTINUE.                   ***")
 (SOUND 15 15) (READ)))
```

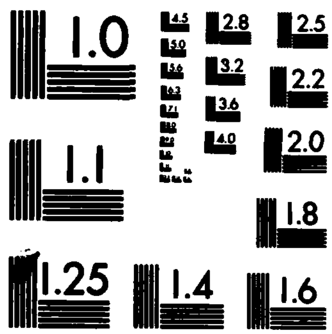
```

% FUNCTION:  QUERY THE USER FOR A PROBLEM AND THEN SELECTS THE MOST PROBABLE
            AREA OF THE KNOWLEDGE BASE TO USE IN DIAGNOSIS.  IT SCANS THE
            PROBLEM STATEMENT FOR KEY WORDS ASSOCIATED WITH AREAS OF THE
            KNOWLEDGE BASE.  IF A KEY WORD IS MATCHED, THAT PORTION OF THE
            KNOWLEDGE BASE IS LOADED.  IF A KEY WORD IS NOT MATCHED, THE
            USER IS QUERIED TO ASSIST IN ISOLATING A SPECIFIC AREA.
   MODULES CALLED: READIN, LOADDB, UAREA
   CALLING MODULES: DIAG
   MODULE NUMBER: 2.2
%

(DEFN SELAREA (LAMBDA ()
(DISPLAY (QUOTE (**** PLEASE STATE THE AIRCRAFT PROBLEM IN CONCISE WORDS ****)))
(DISPLAY (QUOTE (*** TERMINATE YOUR STATEMENT WITH A '!' AND A RETURN ***)))
(DISPLAY (QUOTE (WHAT IS THE PROBLEM?)))
(TERPRI)
(SETQ PROBLM (READIN))
(SETQ CNT 1)
(COND ((LOOP
         (SETQ TPROB PROBLM)
         (SETQ AREA (PACK (SETQ AREA (B CNT))))

         ((LOOP
           (SETQ CKPR (CAR TPROB))
           ((MEMBER CKPR (EVAL AREA)) T)
           (SETQ TPROB (CDR TPROB))
           ((EQUAL TPROB NIL) NIL)) T)   % END OF INNER LOOP %
         (SETQ CNT (PLUS CNT 1))
         ((EQUAL CNT 6) (SETQ AREA NIL) NIL))))
(COND ((NOT (EQUAL AREA NIL))
    (SETQ A1-5 (CAR (PACK (A CNT))))
    (TERPRI)
    (SETQ AREAMSG (APPEND (QUOTE (** PROBLEM APPEARS TO BE IN THE ))
                    (APPEND (CAR A1-5) (QUOTE ( SUBSYSTEM **)))))
    (DISPLAY AREAMSG)
    (TERPRI)
    (SETQ DBSEL (PACK (D AREA)))     % BUILD DATABASE NAME %
    (LOADDB) T)           % LOAD SELECTED DATABASE AREA %
     (T (UAREA)))))    % QUERY USER FOR MORE INFORMATION PRIOR TO %
                       % LOADING SELECTED DATABASE AREA          %
```

```

% FUNCTION: READ LISTS OF INPUTS FROM USER KEYBOARD.
    MODULES CALLED: NONE
    CALLING MODULES: SELAREA, UPDAT
    MODULE NUMBER: 2.2.1
%
(DEFN READIN (LAMBDA ()
(SETQ LST NIL)
(LOOP
  ((MEMBER (RATOM) TERM) LST)
  (SETQ LST (NCONC LST (LIST RATOM))))))

% FUNCTION:  DELETES ELEMENTS OF THE KNOWLEDGE BASE AS POINTED TO BY
             'WHER'.
  MODULES CALLED:  NONE
  CALLING MODULES: UPDAT
  MODULE NUMBER:  3.1.2.1.1
%

(DEFN CLRNODE (LAMBDA (WHER)
((NULL WHER) T)
(SET WHER "")
(CLRNODE (CDR WHER))))
```

AD-A138 373 AIRCRAFT MAINTENANCE EXPERT SYSTEMS(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING G R FERGUSON NOV 83 AFIT/GCS/EE/83D-9 2/2

UNCLASSIFIED F/G 1/3 NL

END



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
% ****************************************************************

  THE FOLLOWING FUNCTIONS ARE USED TO PRODUCE SPECIAL EFFECTS TO
  THE USER DURING OPERATION OF THE SYSTEM.  THESE FUNCTIONS APPLY
  MAINLY TO THE APPLE II+ MICROCOMPUTER AND ARE NOT NECESSARILY A
  FEATURE OF THE EXPERT SYSTEM.
%



(DEFN SOUND (LAMBDA (F1 D1)
(SETQ CNT 1)
(LOOP
  ((EQUAL CNT 5) T)
  (BEEP F1 D1)
  (SETQ CNT (PLUS CNT 1)))))


% FUNCTION:  DISPLAYS MESSAGES ON CRT WITHOUT PARENTHESES
%

(DEFN DISPLAY (LAMBDA (LST)
((NULL LST) (TERPRI))
(PRIN1 (PACK (" " (CAR LST))))
(DISPLAY (CDR LST))))


% FUNCTION: CLEARS CRT AND RETURNS CURSOR TO LEFT UPPER CORNER
%

(DEFN CLRSCRN (LAMBDA ()
(SETQ HOME (ASCII 30))
(TERPRI 25) (DISPLAY HOME) (PRINT HOME)))


% FUNCTION:  PROVIDES DELAYS FOR MESSAGES DISPLAYED ON CRT
%

(DEFN DEELAY (LAMBDA (WHILE)
(SETQ CNT 0)
(LOOP
  ((EQUAL CNT WHILE) T)
  (SETQ CNT (PLUS CNT 1)))))


% FUNCTION: RETURNS INPUT TO KEYBOARD AND BEGINS EXECUTION OF PROGRAM.
%
(F4D (RDS))
```

APPENDIX D:

User's Guide

This guide provides step-by-step procedures for operation of the Maintenance Expert System (MES) developed in this text. The disk containing the MES can be obtained by contacting any member of the committee for this thesis. The MES was developed and implemented using an Apple II+ microcomputer with the following resources:

CPM operating system

LISP language interpreter
(Microsoft, Version 2.15, 1982)

64K RAM

Dual Disks (256K) (only 1 drive required)

80 Character Display

NOTE: From this point onward, it is assumed that the user is familiar with the operations of the Apple and the CPM operating system. Please read completely through this guide prior to using the system.

System Boot

1. Place the CPM Master Disk in drive A.

2. Turn the power on to the monitor and computer. At this time the computer will boot from drive A and when finished, an 'A>' will appear on the CRT screen. If this does not occur, reboot or seek assistance.

3. (Dual drive systems) Place the LISP language disk in drive B and proceed with step #4.

3. (Single drive systems) Remove the CPM master disk and replace it with the LISP language disk. Omit step #4.

4. Type 'B:' and press 'Return'. This assigns drive 'B' as the default disk while using the system. A 'B>' will appear on the screen.

Loading the MES

1. Type 'MULISP F4MES'. This loads the LISP interpreter and the system loadable file containing the expert system. The MULISP logo will appear on the screen while the LISP disk is being read. After a few seconds the '$' prompt of LISP will appear on the screen.

2. Type '(F4D)' and press 'RETURN'. This envokes the MES driver routine for execution of the system.

NOTE:

The disk drive containing the LISP language disk will be the disk used during read/write operations of the MES. If an error message appears on the CRT, take appropriate actions. Otherwise, the MES will be loaded and execution will begin by displaying a screen full of information about the MES to the user. From this point the user simply follows the directions provided by the MES.

Exiting the MES and LISP

The MES can be exited by following the menu directions provided on the CRT. This form of exit will return the operation level to LISP at which time a '$' will appear on the CRT. A 'CTRL-C' can then be entered to exit LISP and return the user to the CPM operating level.

CAUTION: The CPM master disk must be in one of the disk drives when the 'CTRL-C' is used.

Output to Printer

The user can direct the information viewed on the CRT to the printer to obtain a hardcopy document of his session with the MES. To toggle the printer on/off, enter a 'CTRL-P' at any point within the operation of the MES.

APPENDIX E:

Structure and Contents of MES Data Files

The following pages provide the structure and contents of the five data files used in the development and implementation model for this MES. The file names and associated subsystems of the aircraft propulsion system are as follows:

DB1 -- Afterburner
DB2 -- Compressor
DB3 -- Fuel
DB4 -- Ignition
DB5 -- Lubrication

```
% ************** DB1.LIB ************** %

(((DELAYED OR NO AFTERBURNER LIGHT) ((ENGINE SLOW OUT OF AFTERBURNER)
   (2))
((THROTTLE SYSTEM RIGGED PROPERLY) (REPLACE WORN PARTS AND RERIG SYSTEM))
((A/B PUMP VENT DRAINING PROPERLY) (REPLACE A/B PUMP VENT VALVE))
 ((IGNITOR FILTER UNION CLEAN AND POSITIONED PROPERLY)  (CLEAN AND REPOSITION UNION))
 ((SIGNAL PRESENT FROM MAIN FUEL CONTROL TO A/B FUEL PUMP ON RETARD)
  (REPLACE A/B FUEL PUMP)) (CLEAN A/B FUEL SHUTOFF VALVE))
 ( (TORCH IGNITOR LIGHTING PROPERLY)
 ((IGNITOR PLUG DEPTH CORRECT) (ADJUST PLUG DEPTH))
 ((IGNITOR ORIFICES AND BASKET CLEAN) (CLEAN OR REPLACE PARTS))
 ((A/B IGNITION SWITCH OPERATING PROPERLY)
 (REPLACE A/B IGNITION SWITCH)) (CHECK POWER SOURCE AND CIRCUITS))
  ((SPRAYBARS BULGED OR BROKEN) (CHANGE A/B PRESSURE VALVE))
 (REPLACE SPRAYBARS))


 % ***************** DB2.LIB ***************** %


(((ENGINE ROUGH OR VIBRATING) (3)) (((FOD EVIDENCE IN COMPRESSOR AREA))
 ((ENGINE MOUNTS SECURE) (REPLACE OR RETORQUE MOUNTS))
((HYDRAULIC PUMP COMPONENTS SECURE) (REPLACE DEFECTIVE COMPONENTS))
  ((VARIABLE VANES FOLLOW SCHEDULE) (RIG VARIABLE VANES))
 ((SOAP CHECK GOOD) (SEND ENGINE TO HIGHER LEVEL MAINTENANCE))
 (RUN ENGINE ON TEST STAND FOR VIBRATION CHECK)) (REFER TO -2 -8 SECTION XII FOR REPAIR))
```

```
% ****************** DB3.LIB ****************** %
(((EGT HIGH AT MIL POWER) (((EGT LOW AT FULL THROTTLE)) ((ENGINE SLOW TO ACCELERATE)
 (5))     % TRY AREA 5 %
 ((FOD IN COMPRESSOR OR TURBINE AREA) ((SPECIFIC GRAVITY ADJUSTMENT CORRECT)
 (ADJUST SPECIFIC GRAVITY FOR FUEL))
 ((CDP LINE LEAKING) ((BLEED AIR DUCTS LEAKING)
((VARIABLE VANE TRACKING PROPERLY) (RERIG VARIABLE VANES))
(CHECK EXHAUST NOZZLE RIGGING)) (TIGHTEN COMPONENTS))
 (REPLACE WASHERS AND TORQUE FITTINGS)) (REFER TO -2 -8 SECTION XII FOR REPAIR))
 (REFER TO -2 -8 PARA 2 -104))
 ((JETCAL ANALYSIS SHOWS SYSTEM READS CORRECT) (REPLACE FAULTY PARTS))
 ((OVERTEMP EXCEEDED SERVICE LIMITS)
  ((ENGINE CONTROL CIRCUITS WITHIN LIMITS ON JETCAL) (REPLACE FAULTY PARTS))
(ADJUST EGT AT AMPLIFIER AND CHECK TEMP)) (SEND ENGINE TO HIGHER LEVEL MAINTENANCE))
 % *************** DB4.LIB **************** %
(((ENGINE ROTATION GOOD) ((CHECK AUXILIARY POWER CART AND TRY RESTART)  ))
   ((FUEL FLOW GOOD) ((MANIFOLD SHUTOFF VALVE OPERATIONAL)
      (RUN ENGINE TO CHECK THROTTLE CUTOFF SWITCH AND REEVALUATE PROBLEM))
 ((POWERPLANT RIGGING GOOD) (RERIG SYSTEM))
    (INSPECT DRIVE AND REPLACE FUEL PUMP))
 ((BOTH IGNITOR PLUGS INOPERATIVE)
     ((28 VDC AT INPUT) (CHECK IGNITION WIRING))
 (REPLACE IGNITION UNIT)) (CHECK 28 VDC IGNITION CIRCUIT))
 % ****************** DB5.LIB **************** %
(((HIGH OIL CONSUMPTION) ((HIGH OIL PRESSURE)
      ((LOW OIL PRESSURE)
     (THERE IS NOTHING WRONG WITH THIS AIRCRAFT, CHECK THE CREW))
       ((OIL TANK SERVICED PROPERLY) (SERVICE OIL TANK AND RECORD))
       ((OIL FLOWS FROM PUMP) (FLUSH OIL SYSTEM AND REPLACE PUMP))
        (REFER TO -2-11 FOR REPAIR))
       ((HIGH PRESSURE DETECTED IN A/B) (FLUSH SYSTEM AND RUN ENGINE))
        (NORMAL CONDITION IN A/B))
       ((ENGINE APPEARS TO MAKE OIL)
           ((ENGINE HAS HISTORY OF HIGH OIL USE)
          ((MORE THAN 1 QUART IN GEARBOXES) (OIL USE MAY BE OK))
           (CLEAN AND REPAIR GEARBOXES))
              (HIGHER LEVEL MAINTENANCE REQUIRED))
           (REPAIR LEAKING LUBE OR HYDRAULIC VALVES))
```

VITA

Capt Gerald R. Ferguson was born in Moultrie, Georgia on 15 Sept 1949. He graduated from Moultrie High School in 1967 and attended the University of Georgia.

He enlisted in the Air Force in Sept 1969 as a Jet Aircraft Maintenance Specialist. In 1974, he retrained into the computer career field as a Computer Maintenance Repairman. While serving as an instructor with the 3390th Technical Training Group at Keesler Air Force Base, MS, he completed a Bachelor of Science Degree in Computer Technology at the University of Southern Mississippi.

Capt Ferguson was commissioned through Officer Training School in Aug 1978. He was reassigned to Keesler AFB as an instructor member of the Worldwide Military Command and Control System (WWMCCS) Mobile Training Team (MTT). In this capacity, he instructed both national and foreign students in basic and advanced programming techniques, operating systems software, and database implementations.

Capt Ferguson is one of the initial developers of the only Air Force Specialty Code (AFSC) granting course for the multishred 5135 officer AFSC. He served both as an instructor and instructor supervisor for that course. In May 1982 he was assigned to the Air Force Institute of Technology, School of Engineering where he pursued a Masters Degree in Computer Systems.

Permanent Address:

Box 231, Route 5, Moultrie, Georgia 31768

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) UNCLASSIFIED

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER AFIT/GCS/EE/83D-9	2. GOVT ACCESSION NO. AD-A138373	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) AIRCRAFT MAINTENANCE EXPERT SYSTEMS		5. TYPE OF REPORT & PERIOD COVERED MS Thesis
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Gerald R. Ferguson		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT-EN) Wright-Patterson AFB Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Human Resources Laboratory (AFHRL) Wright-Patterson AFB Ohio 45433		12. REPORT DATE November 1983
		13. NUMBER OF PAGES 104
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Approved for public release: IAW AFR 190-17.
LYNN E. WOLAVER 7 Feb 84
Dean for Research and Professional Development
Air Force Institute of Technology (ATC)
Wright-Patterson AFB OH 45433

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Expert Systems
Knowledge Base
Aircraft Maintenance
Artificial Intelligence
LISP
Diagnotics

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The field of study known as "artificial intelligence" has gained considerable interest and support within the last few years. Machines are now capable of performing tasks that only humans were thought capable of performing. Machines have been constructed to assist humans in almost every aspect of their daily lives.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(Blk 20 continued)

However, there are some daily activities in which the human must perform manual tasks. One such area is aircraft maintenance. The current manual procedures in this area are very strenuous and time consuming. This text investigates only one feature of those manual procedures, i.e. diagnosis of aircraft malfunctions.

This text provides design considerations for implementation of an "expert system" to assist in the diagnosis of aircraft problems. It illustrates the characteristics required for an automated diagnostic system to assist the average aircraft technician in the performance of his/her duties. The design of a "knowledge base" and "inference procedure" for such a system are presented. A working system model was developed on a microcomputer to demonstrate the feasibility for a full scale maintenance expert system.

FILMED

3-84

DTIC