NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

DEVELOPMENT OF A PROTOTYPE
H-46 HELICOPTER
DIAGNOSTIC EXPERT SYSTEM

by

Thomas J. Gadzala

September 1987

Thesis Advisor Nancy C. Roberts

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION Unclassified	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; Distribution is Unlimited
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School — 6b OFFICE SYMBOL (If applicable) Code 54	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000	7b ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION — 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
DEVELOPMENT OF A PROTOTYPE H-46 HELICOPTER DIAGNOSTIC EXPERT SYSTEM (u)

12 PERSONAL AUTHOR(S)
Gadzala, Thomas J.

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month Day)	15 PAGE COUNT
Master's Thesis	FROM ______ TO ______	1987 September	125

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	H-46 Helicopter, Expert Systems, Computer-Aided Diagnostic System, M.1, Diagnostic Expert System

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

This study was undertaken to demonstrate the feasibility of applying expert system technology to the Navy's H-46 helicopter maintenance process. A microcomputer-based prototype known as a computer-aided diagnostic system (CADS) was developed for this purpose. Given a helicopter electrical or hydraulic system discrepancy, the troubleshooter interacts with CADS to find the cause. The prototype CADS was developed utilizing the M.1 knowledge-based system development tool by Teknowledge, Inc.

The complexity of helicopter systems diagnosis, and inadequacies of the maintenance manuals, often result in unnecessary removal of system components. The prototype CADS is intended to demonstrate that a fully developed system, containing all the formal and heuristic knowledge of H-46 diagnostic information, could eliminate these problems. Also, such a diagnostic system could provide a comprehensive, stable diagnostic knowledge base, regardless of personnel turnover.

This study includes a description of current helicopter maintenance procedures, and how the integration of CADS could improve this process. Also included are

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION Unclassified
22a NAME OF RESPONSIBLE INDIVIDUAL Prof. Nancy C. Roberts	22b TELEPHONE (Include Area Code) (408) 646-2742 — 22c OFFICE SYMBOL 54Rc

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete

#19. ABSTRACT (continued)

descriptions of expert systems and the M.1 knowledge-based system development tool: how they work, and their applicability to structured selection problem-solving. The development and testing strategies used for CADS are discussed in detail. Results, conclusions, and recommendations for further study are provided.

Development of a Prototype
H-46 Helicopter
Diagnostic Expert System

by

Thomas J. Gadzala
Lieutenant Commander, United States Navy
B.S., United States Naval Academy, 1975
M.S., University of Southern California, 1982

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
September 1987

ABSTRACT

This study was undertaken to demonstrate the feasibility of applying expert system technology to the Navy's H-46 helicopter maintenance process. A microcomputer-based prototype known as a computer-aided diagnostic system (CADS) was developed for this purpose. Given a helicopter electrical or hydraulic system discrepancy, the troubleshooter interacts with CADS to find the cause. The prototype CADS was developed utilizing the M.1 knowledge-based system development tool by Teknowledge, Inc.

The complexity of helicopter systems diagnosis, and inadequacies of the maintenance manuals, often result in unnecessary removal of system components. The prototype CADS is intended to demonstrate that a fully developed system, containing all the formal and heuristic knowledge of H-46 diagnostic information, could eliminate these problems. Also, such a diagnostic system could provide a comprehensive, stable diagnostic knowledge base, regardless of personnel turnover.

This study includes a description of current helicopter maintenance procedures, and how the integration of CADS could improve this process. Also included are descriptions of expert systems and the M.1 knowledge-based system development tool: how they work, and their applicability to structured selection problem-solving. The development and testing strategies used for CADS are discussed in detail. Results, conclusions, and recommendations for further study are provided.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

A. GENERAL

Three objectives of the Naval Aviation Maintenance Program's (NAMP) aviation material readiness standards established by the Chief of Naval Operations (CNO) are:

1. Achieve, for all Navy and Marine Corps aircraft, a 78% mission capable rate for deployed aircraft and a 73% mission capable rate overall; a 61% full systems capable rate for deployed aircraft and a 56% full systems capable rate overall.
2. Reduce the growth rate in Maintenance Man-hours per Flight-Hour (MMH/FH) to zero and maintain a level no higher than the FY-78 average.
3. Reduce the growth rate in maintenance and support costs per flight-hour to zero and maintain a level no higher than the FY-78 average. [Ref. 1: p. 2-1]

It is the responsibility of Naval aviation squadrons to meet these objectives with optimum use of manpower, material, and funds. Such is the case with the United States Navy's and United States Marine Corps' 21 H-46 helicopter squadrons. The H-46 Sea Knight is shown in Figure 1.1.

The H-46 Sea Knight was developed and manufactured by the Boeing Aircraft Company's Vertol Division in the 1960s. Since that time many of the helicopter's systems have been modified or changed completely. Increasingly complex helicopter systems and the diverse nature of Naval aviation operating requirements place a high premium on the ability of squadron maintenance people to achieve the aircraft material readiness standards. This ability is a function of the training that these people receive and the tools at their disposal. Any means that can be used to increase the effectiveness of the maintenance process should enhance the ability of achieving these objectives.

Artificial intelligence techniques offer a promising means for helping maintenance personnel. Specifically, the type of computer program known as an expert system deserves consideration for this purpose. A prototype expert system program known as the Computer-Aided Diagnostic System (CADS) has been developed by the author to test the value of such a system; that is, to demonstrate feasibility and applicability of expert systems technology to Naval aviation maintenance.

Figure 1.1 H-46 "Sea Knight" Helicopter.

B. BACKGROUND

To understand the application of CADS a brief familiarization of the helicopter squadrons' current maintenance troubleshooting process is necessary. When a helicopter system discrepancy is discovered by pilots, aircrew, or maintenance personnel, the discrepancy is documented on a Maintenance Action Form (MAF). The MAF is processed by the squadron's Maintenance Control Office, which routes the MAF to the appropriate maintenance shop, i.e., hydraulics shop, power plants shop, or electrical shop. The maintenance personnel of the shop, who are trained in that specific aircraft system, follow the Maintenance Information Manuals' (MIMs) troubleshooting procedures to isolate the problem or symptom, find the cause, then take the appropriate corrective action. There are approximately 15 separate MIMs volumes. Each volume contains troubleshooting procedures for a specific helicopter system.

C. APPLICATION OF CADS

CADS is designed as a prototype microcomputer-based expert system to aid maintenance personnel (or other squadron personnel) diagnose helicopter equipment failures. Given a helicopter system discrepancy, CADS will help find the cause of the discrepancy.

Helicopter aircraft systems are complex. Finding the cause of a helicopter system discrepancy can involve difficult and time-consuming troubleshooting techniques. Yet maintenance people must be able to repair helicopter problems quickly and efficiently to provide aircraft that are fully systems-capable for the next operational commitment. When the aircraft problem is complicated, difficult to troubleshoot, and not covered adequately in the MIMs, troubleshooting sometimes becomes the mere changing of black boxes and hoping the problem will become fixed. This can result in curing the symptom but not the cause, aggravation of the problem, and time and money wasted in maintenance man-hours and parts costs. It is anticipated that the use of a fully developed CADS could reduce or even eliminate this problem.

CADS also can be used in the squadron as a training tool for maintenance troubleshooting procedures. A fully developed CADS would contain knowledge representing the entire maintenance diagnostic expertise of an aircraft community. Utilizing the trace feature of this expert system, maintenance persons perhaps could observe CADS' reasoning process in finding the cause of the problem, given the discrepancy.

Such a fully developed CADS would also provide a comprehensive, stable diagnostic knowledge base, regardless of personnel turnover. The nature of military tours of duty creates the problem of loss of corporate knowledge as experienced people are transferred elsewhere. The diagnostic skills of maintenance experts would remain in the squadron's CADS long after the experts have departed.

The H-46 helicopter has recently undergone significant systems modifications in an effort to extend its serviceable life. Under the Surviveability, Reliability, and Maintainability (S,R,&M) Program all H-46 "A" and "D" models will have significant systems modifications, and be redesignated CH-46D. A problem associated with the S,R,&M Program is the lack of knowledge concerning the modified helicopter systems among operational squadron personnel. The CADS developed for this project contains the troubleshooting process as described in the updated MIMs for S,R,&M helicopters (CH-46D), and knowledge acquired from identified H-46 experts.

Although the CADS developed for this project is not truly fully developed, it can serve as a prototype to demonstrate the feasibility of applying expert system technology to the United States Navy's H-46 helicopter maintenance diagnostic process.

D. OBJECTIVES

The objectives of this study are as follows:

1. Develop a prototype CADS to demonstrate the feasibility of applying expert system technology to a maintenance diagnostic process. The diagnostic expert system program is written by the author utilizing the M.1 Knowledge System Development Tool (version 2.1) by Teknowledge, Inc.
2. Demonstrate the CADS applicability to the United States Navy's H-46 helicopter maintenance process.

E. RESEARCH QUESTIONS

The main question addressed in this study can be summarized as:

Is a computer-aided diagnostic system feasible and applicable to the United States Navy's H-46 helicopter maintenance process?

Secondary questions pertaining to the research areas are:

1. Squadron applications.
   - How would the maintenance personnel use CADS?
   - How could the CADS knowledge base provide a valuable training tool for maintenance personnel?
   - Could the CADS knowledge base contain the corporate knowledge of the H-46 helicopter community's maintenance diagnostic expertise, regardless of personnel turnover?
   - Can other people in the squadron use CADS program besides maintenance personnel? How will they use CADS?
   - Why is development of CADS particularly germane in view of the recent H-46 systems modifications (S,R,&M)?
2. Use of CADS.
   - What interaction is required between CADS and the user?
   - What are the hardware, software, and memory requirements?

F. SCOPE AND LIMITATIONS

This research focuses on the development of the CADS prototype program, and on its applicability to the helicopter maintenance process. The prototype CADS includes knowledge related to diagnoses of the hydraulic and electrical systems of the S,R,&M CH-46D helicopter. The prototype addresses every symptom and problem cause for these two systems as specified in the MIMs troubleshooting tables. It also contains a limited amount of the heuristic knowledge of Naval Aeronautical Engineering Service Unit (NAESU) technical representatives, and squadron maintenance experts. A fully developed CADS is beyond the time and monetary constraints of this research. The prototype CADS software is intended as an aid to using the MIMs. The prototype is by no means a replacement for these technical publications. However, ultimately a fully developed CADS could preclude the use of written maintenance manuals.

This research will not include a discussion of CADS implementation issues in the helicopter squadrons. Testing the diagnostic expert system program in the squadron will be limited; therefore the program cannot be considered completely validated.

G. ORGANIZATION OF STUDY

Chapter II contains a description of the Naval Aviation Maintenance Program's Maintenance Data System (MDS). Discussion includes maintenance documentation procedures, troubleshooting procedures, and how integrating CADS could enhance the process, thereby improving readiness of the squadrons. Chapter III discusses knowledge-based expert systems, and the M.1 knowledge-based system development tool. Chapter IV describes in detail the development of the prototype CADS, and how it works. Chapter V discusses the results and conclusions of this research. It also provides suggestions for further study, and recommendations for the use of CADS and M.1. Chapter VI provides an executive summary of this study and its results. Appendix A contains a glossary of acronyms used in this study. A sample diagnostic consultation session is contained in Appendix B. The prototype CADS knowledge base is contained in Appendix C.

II. SQUADRON MAINTENANCE: CURRENT PROCESS AND CADS INTEGRATION

Prior to describing how expert systems work and the CADS development process, it is necessary to describe the aviation MDS used by the helicopter troubleshooters, in order to understand how CADS would be integrated. This chapter explains MDS, maintenance action documentation procedures, troubleshooting procedures, and integrating CADS into the troubleshooting process.

A. MAINTENANCE DATA SYSTEM

The MDS was developed to provide data input to the NAMP for the purpose of meeting the aviation material readiness standards established by the CNO. The description of MDS is quoted directly from volume II of NAMP:

> The MDS is a Management Information System (MIS) designed to provide statistical data for use at all management levels relative to:
>
> a. Equipment maintainability and reliability.
>
> b. Equipment configuration, including alteration and Technical Directive (TD) status.
>
> c. Equipment mission capability and utilization.
>
> d. Material usage.
>
> e. Material nonavailability.
>
> f. Maintenance and material processing times.
>
> g. Weapon system and maintenance material costing. [Ref. 2: p. 11-1]

The MDS requires that any work done on an aircraft is to be documented by the person performing the work. The maintenance person converts a brief narrative description of the job into codes and enters the coded information onto standard forms or source documents. The source documents are collected and forwarded to a data service facility, where the data are converted to machine records. From these records Maintenance Data Reports (MDR) are produced, which are periodic report listings summarizing the maintenance data. The MDRs are supplied to squadron maintenance supervisors to assist them in planning and directing the maintenance effort.

Application of the management information provided by the MDR assists in identifying, among other things, trend analysis of the following:

- Areas in which there are skill or training deficiencies.
- Efficient or inefficient use of available manpower.
- Parts with high failure rates.
- Inadequate troubleshooting.

These four areas are specifically noted as they are most applicable to this research.

B. MAINTENANCE ACTION DOCUMENTATION PROCEDURES

One type of MDS source document is the Visual Information Display System/Maintenance Action Form (VIDS/MAF) shown in Figure 2.1. VIDS/MAFs are used to document over 17 specific types of maintenance actions, as outlined in NAMP. Included are such maintenance actions as troubleshooting man-hours, removal and replacement of aircraft parts, repair, the performance of scheduled inspections, and accumulated man-hours as a result of work stoppage for parts or maintenance.

A VIDS/MAF is originated by the squadron Maintenance Control Office personnel or by aircrew, when an aircraft discrepancy is discovered. The form is also used to document scheduled periodic maintenance or special aircraft inspections that become due. The initiated VIDS/MAF provides for recording, among others, the following types of data, quoted directly from NAMP:

a. Job Control Number (JCN).

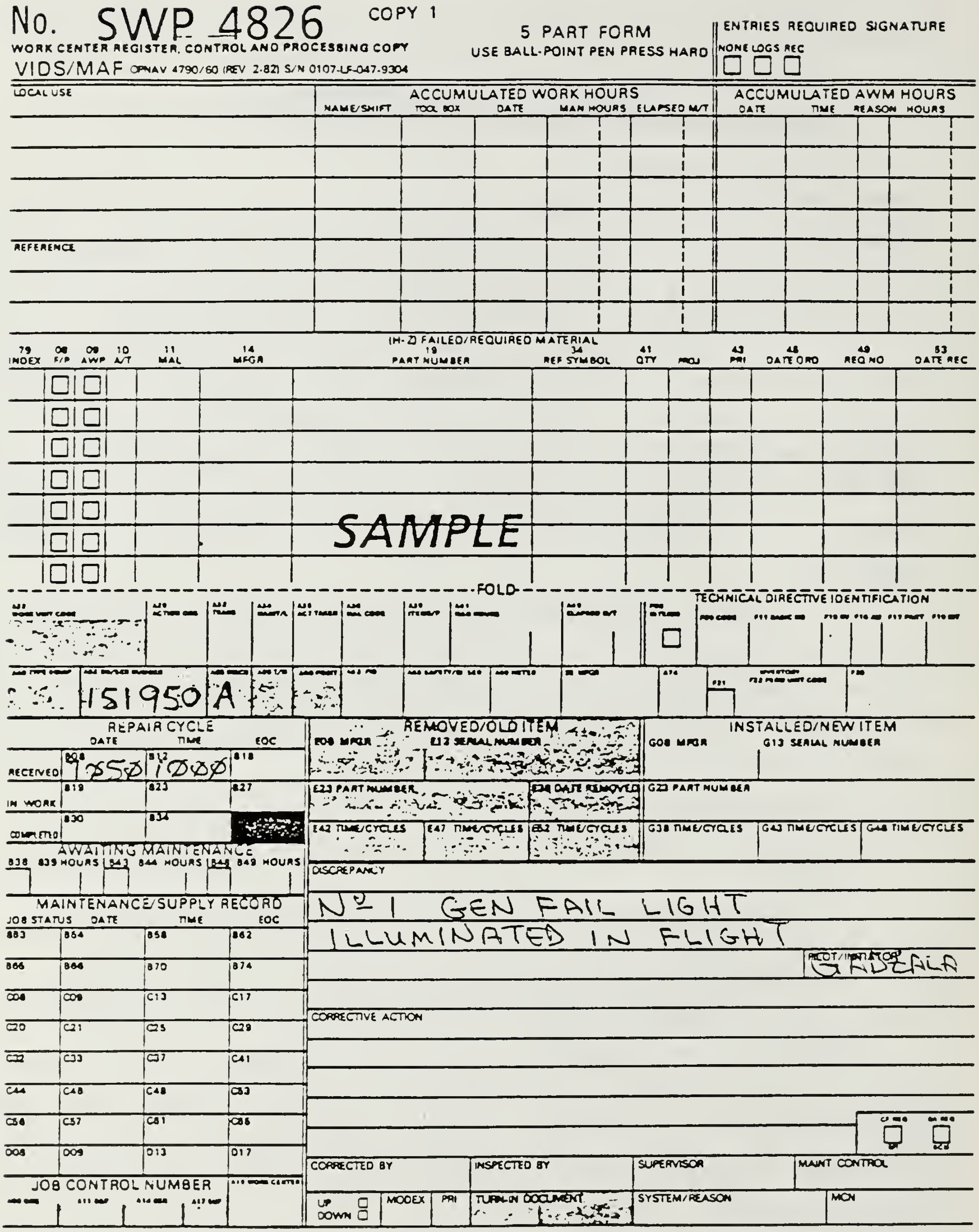
b. Identity of the organization and the work center in which the work is being performed.

c. The type equipment, system, subsystem, and component upon which work is being performed.

d. How the malfunction/discrepancy/failure occurred, and when it was discovered, and the action taken to correct it. [Ref. 2: p. 11-3]

Maintenance Control forwards a copy of the VIDS/MAF to the appropriate work center. For example, if the discrepancy is "No.1 Generator failure light illuminated in flight", the VIDS/MAF copy will be routed to the electrical shop. The work center supervisor screens the document, enters applicable data, and assigns workers to the task.

When complex problems occur, troubleshooting often requires a great amount of time. In these cases troubleshooting time is separated from repair time by



No. SWP 4826 COPY 1

WORK CENTER REGISTER, CONTROL AND PROCESSING COPY

VIDS/MAF OPNAV 4790/60 (REV 2-82) S/N 0107-LF-047-9304

5 PART FORM

USE BALL-POINT PEN PRESS HARD

ENTRIES REQUIRED SIGNATURE

NONE LOGS REC

LOCAL USE

REFERENCE

ACCUMULATED WORK HOURS

NAME/SHIFT | TOOL BOX | DATE | MAN HOURS | ELAPSED M/T

ACCUMULATED AWM HOURS

DATE | TIME | REASON | HOURS

(H-Z) FAILED/REQUIRED MATERIAL

79 INDEX | 08 F/P | 09 AWP | 10 A/T | 11 MAL | 14 MFGR | 19 PART NUMBER | 34 REF SYMBOL | 41 QTY | PROJ | 43 PRI | 45 DATE ORD | 49 REQ NO | 53 DATE REC

SAMPLE

FOLD

TECHNICAL DIRECTIVE IDENTIFICATION

151950 A

REPAIR CYCLE

DATE | TIME | EOC

RECEIVED B08 9250 | B12 1000 | B18

IN WORK B19 | B23 | B27

COMPLETED B30 | B34

AWAITING MAINTENANCE

B38 B39 HOURS | B43 B44 HOURS | B48 B49 HOURS

MAINTENANCE/SUPPLY RECORD

JOB STATUS | DATE | TIME | EOC

B83 | B64 | B58 | B62

B66 | B66 | B70 | B74

C08 | C09 | C13 | C17

C20 | C21 | C25 | C29

C32 | C33 | C37 | C41

C44 | C48 | C48 | C53

C56 | C57 | C61 | C65

D08 | D09 | D13 | D17

JOB CONTROL NUMBER

REMOVED/OLD ITEM

E08 MFGR | E13 SERIAL NUMBER

E23 PART NUMBER | E38 DATE REMOVED

E42 TIME/CYCLES | E47 TIME/CYCLES | E52 TIME/CYCLES

INSTALLED/NEW ITEM

G08 MFGR | G13 SERIAL NUMBER

G23 PART NUMBER

G38 TIME/CYCLES | G43 TIME/CYCLES | G48 TIME/CYCLES

DISCREPANCY

NO 1 GEN FAIL LIGHT

ILLUMINATED IN FLIGHT

PILOT/INITIATOR GALDEALA

CORRECTIVE ACTION

CORRECTED BY | INSPECTED BY | SUPERVISOR | MAINT CONTROL

UP / DOWN | MODEX | PRI | TURN-IN DOCUMENT | SYSTEM/REASON | MCN

Figure 2.1 VIDS/MAF Showing No.1 Generator Failure Light Discrepancy.

documenting two VIDS/MAFs: one to reflect troubleshooting time, the other to record repair time.

C. TROUBLESHOOTING PROCEDURES

The maintenance personnel assigned to correct the discrepancy consult the appropriate MIMs volume. In the above example, it would be the Electrical Systems volume. Within each of the 15 MIMs volumes are either troubleshooting tables or troubleshooting sections, which outline isolation procedures for system symptoms. Maintenance people from the work center perform the isolation procedures, diagnosing the discrepancy to find the cause of the problem. Consider the example symptom of a No.1 Generator failure light that has illuminated on the helicopter's cockpit caution light panel. This symptom description would be entered on the VIDS/MAF by the aircrewman who discovered it, as shown in Figure 2.1. It is similarly listed in the MIMs troubleshooting tables as "GEN FAIL light on (both generators on) rotors at flight RPM" [Ref. 3: p. 3].

The maintenance workers search the troubleshooting tables of the Electrical System MIM for this specific symptom, then perform the isolation procedures. A portion of the troubleshooting tables, which they would use for this particular problem, is contained in Figure 2.2 [Ref. 3: p. 3]. Other than what is contained in the troubleshooting tables, the MIMs do not fully describe each step of action and its consequences. The troubleshooters learn this through training and experience. This experience could easily be included in CADS. A flow diagram of this troubleshooting procedure for this example is shown in Figure 2.3 to assist the reader.

According to the MIMs troubleshooting tables, there are two possible outcomes from this isolation procedure, i.e., 1) the generator comes on line, and the No.1 GEN FAIL light extinguishes, or 2) the No.1 GEN FAIL light remains illuminated. For the first outcome, the MIM gives the cause as a supervisory panel malfunction. For the second, the troubleshooters perform subsequent isolation checks, to find the possible cause of the generator failure. See Figures 2.2 and 2.3.

This structured, methodical process of searching for the possible cause of an aircraft problem is universal in Naval aviation maintenance. Having found the exact cause of the problem after performing all applicable isolation procedures, the maintenance workers again refer to the MIMs. Each problem cause has specific corrective actions, which are briefly mentioned in the troubleshooting tables. Detailed corrective action procedures are contained in other sections of the MIMs.

6. Table 1 includes isolation procedures and corrective actions for symptoms identified by letters as follows:

A. GEN FAIL light on (both generators on) rotors at flight RPM.

B. One ESSENTIAL BUS light on (either or both generators on) rotors at flight RPM.

C. One or both ESSENTIAL BUS lights on (associated with one generator) rotors at flight RPM.

D. One DC BUS light on (either or both generators on) rotors at flight RPM.

E. Opposite DC BUS light on (one generator on) rotors at flight RPM.

7. When troubleshooting the AC/DC generation systems refer to the following schematics:

a. AC System – WP 003 00, figure 2.

b. DC System and Battery – WP 004 00, figure 2.

Table 1. AC System Troubleshooting

CAUSE	ISOLATION PROCEDURE	CORRECTIVE ACTION
A. Gen Fail Light On (Both GEN's On) Rotors At Flight RPM.		
Supervisory Panel Malfunction	With generators "OFF" cross connect the opposite Supervisory Panel into malfunctioning system using both test cables (figures 1 and 2) turn "ON" discrepant AC system.	If generator comes on line, remove and replace supervisory panel.
Generator Malfunction	With main generators selected and both generator switches "ON", check for 115 to 120 vac pins A, B, and C to ground of the related generator test receptacle (161J3) (161J4) or terminals M, P, and R respectively of the terminal board on generator test cable (figure 1).	If voltages are correct, check phasing, contactor control and switch control. If all voltages are missing or very low, check excitation, PMG output and feeder fault. If one or two voltages are zero, check generator feeder wires. (NOTE: Voltage readings taken at test receptacles with generator switch in test position are no load voltage indications.) These readings may not indicate an RFI generator.
PMG Output Incorrect	With generators off, connect the generator test cable (figure 1) between the supervisory panel receptacle and connector plug (241P5) (241P6) on the malfunctioning system. At the test terminal board, check for approximately 38 vac between ground and terminal E and terminal H.	If voltage is low, disconnect test cable receptacle at helicopter cable plug and repeat measurement. If voltage does not rise to approximately 38 vac, the pm section of the generator is defective. If voltage is normal, check exciter winding.

Figure 2.2 Electrical Systems MIMs Troubleshooting Table.

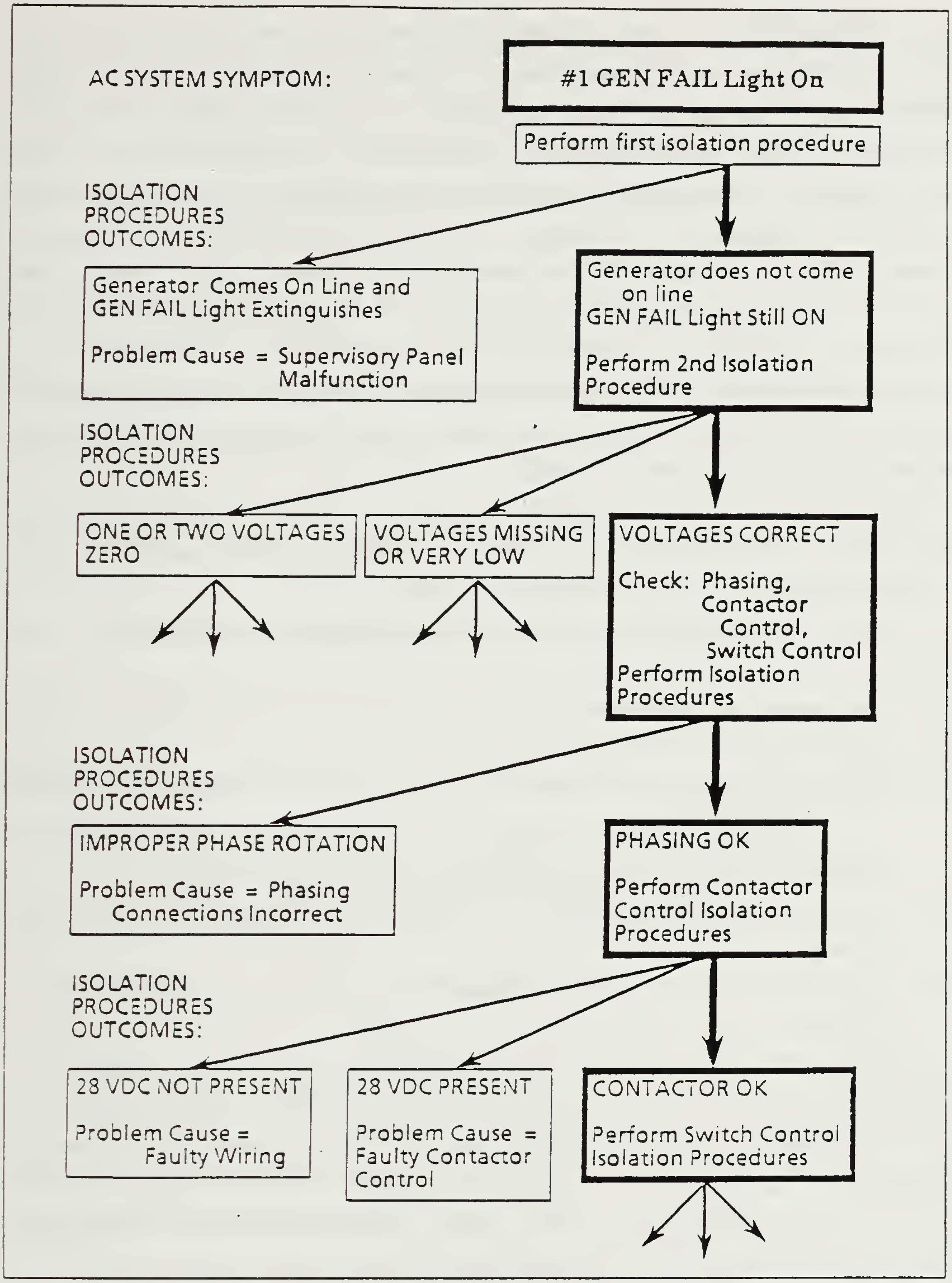


Figure 2.3 Troubleshooting Flow Diagram.

D. INTEGRATING CADS INTO THE TROUBLESHOOTING PROCESS

Diagnosing helicopter discrepancies is a process which involves more than simply following isolation procedures in MIMs. It takes a person with formal knowledge and experience to diagnose accurately and to know exactly which parts are defective. Very few mechanics in any aircraft squadron are recognized as good troubleshooters. Individuals aspiring to be good troubleshooters will constantly refer to the maintenance manuals. They will also ask the squadron's recognized experts how effective troubleshooting is performed. What happens when the manuals contain incomplete information, or have not been updated with current airframes changes? Worse yet, when the experts in the unit are transferred, who will maintain their knowledge and experience needed for the maintenance effort, and who will impart that knowledge to the people aspiring to attain those skills?

CADS may solve some of these problems, but may result in others as well. How would CADS be integrated into the maintenance troubleshooting process? How would the maintenance personnel use CADS in this process?

Several assumptions are necessary before these questions can be answered. These include:

1. Each squadron work center has (or has access to) a Zenith 248 or IBM PC compatible microcomputer.
2. Each work center has a copy of the CADS program contained on 5.25 inch floppy diskettes.
3. All maintenance personnel in the squadron have received training on how to operate the microcomputer, and can boot up the CADS program.
4. All maintenance personnel have received training in the use of the CADS program.
5. There is a qualified knowledge engineer in the squadron to maintain the CADS program.

When the work center receives a VIDS/MAF, the workers assigned can immediately boot up CADS to begin troubleshooting. The MIMs will still be required, as the prototype CADS does not contain the corrective-action repair procedures. The advantage of CADS is that its use could possibly reduce the time required in troubleshooting the discrepancy. CADS offers a much faster means of searching for the correct isolation procedures for a problem symptom.

Using the MIMs is often difficult because the maintenance persons constantly must turn pages between the troubleshooting tables, the appropriate part diagrams and figures, and the corrective action procedures section. Using CADS on a microcomputer with the MIMs would eliminate the need for constantly referring back and forth between sections. The maintenance people can interact with CADS, while leaving the MIMs open to the appropriate figures and diagrams. Of course, the ultimate system would integrate a fully developed CADS, all maintenance procedures, and include all figures and diagrams. Through the use of "pull down windows", all information would be available to the maintenance person on the computer, literally at his/her finger tips. This would eliminate the need for manuals.

Cases of rarely occurring or particularly difficult problems may be beyond the expertise of squadron maintenance personnel. The symptoms and isolation procedures may not be contained in the MIMs troubleshooting tables. The squadron Maintenance Officer may request assistance from the NAESU technical representatives. The NAESU technical representatives are civilian systems engineers with in-depth training and experience with specific aircraft systems. They combine formal knowledge with heuristics (rules of thumb) to solve the helicopter's problem.

The heuristics of these experts are not contained in the MIMs, but could be contained in a fully developed CADS. Historic cases of problems not addressed in the MIMs could be acquired from the experts, then programmed into CADS. Should new problems occur, they also could be programmed into CADS. This would further expand and improve the CADS knowledge base. This aspect of CADS is particularly important when there are no technical representatives readily available.

Availability of this troubleshooting expertise is often crucial to deployed H-46 units of the Navy and Marine Corps. The Navy has five Helicopter Combat Support Squadrons with H-46 helicopters. Four of the five squadrons deploy units known as detachments on ships at sea for extended periods (six to nine months). Each detachment contains two H-46 helicopters, six pilots, and approximately 20 maintenance men. It is hoped that the detachment members will possess enough knowledge and experience to fix any H-46 helicopter problem that could occur, without support of the mother squadron or technical representatives. But what if they have a problem that is beyond their expertise? Having a fully developed CADS onboard ship would prove to be of immeasureable value to deployed H-46 units.

There are also situations in which non-maintenance people in the squadron could utilize CADS. A common scenario involves an aircraft and crew that fly cross country to another military airfield, which has limited or no H-46 maintenance support capabilities. Upon post-flight inspection it is found that hydraulic fluid is covering the entire inboard aft section of the helicopter. The exact source of the leak is not readily apparent. None of the crew are aviation hydraulic systems technicians. They call the mother squadron (on a Sunday morning) for help. The duty personnel are all administrative technicians. Although unfamiliar with maintenance procedures and the use of MIMs, the duty persons could provide immediate information to the stranded crew. With the CADS diskettes and a microcomputer in the squadron duty office, they could boot up CADS, and tell the aircrew the isolation procedures for the symptom.

The author believes that integrating CADS into the squadron environment could improve the effectiveness and efficiency of the squadrons' maintenance efforts. Quite possibly the improvement would be reflected in the MDRs. The MDRs generated after CADS integration could be compared to those generated prior to CADS. Reductions of troubleshooting man-hours, removal and replacement of aircraft parts, repair times, and accumulated man-hours as a result of work stoppage for maintenance would be possible indications of improvement.

III. DESCRIPTION OF EXPERT SYSTEMS AND M.1

A definition of an expert system is given by Professor Edward Feigenbaum of Stanford University, one of the leading researchers in expert systems:

> . . . an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solution. Knowledge necessary to perform at such a level, plus the inference procedures used, can be thought of as a model of the expertise of the best practitioners of the field.
>
> The knowledge of an expert system consists of facts and heuristics. The "facts" constitute a body of information that is widely shared, publicly available, and generally agreed upon by experts in the field. The "heuristics" are mostly private, little-discussed rules of good judgment (rules of plausible reasoning, rules of good guessing) that characterize expert-level decision making in the field. The performance level of an expert system is primarily a function of the size and the quality of a knowledge base it possesses. [Ref. 4: p. 5]

A. EMPHASIS ON KNOWLEDGE-BASED EXPERT SYSTEMS

Knowledge-based expert systems can be considered the most popular type of applied Artificial Intelligence (AI) systems. Why knowledge-based expert systems? AI research has been aimed primarily at creating a machine with the capability to perform problem solving by attempting to duplicate the human thinking process. Less abstract are knowledge-based expert systems, which focus on domain-specific knowledge used for solving narrowly defined problems.

Fredrick Hayes-Roth, Donald A. Waterman, and Douglas B. Lenat contend that:

> Machines that lack knowledge seem doomed to perform intellectually trivial tasks. Those that embody knowledge and apply it skillfully seem capable of equaling or surpassing the best performance of human experts. Knowledge provides the power to do work; knowledge engineering is the technology that promises to make knowledge a valuable industrial commodity. [Ref. 5: p. 3]

Knowledge engineering can be defined as the technology of building expert systems. Knowledge engineers develop expert systems by performing activities of problem assessment, knowledge acquisition, and representing the acquired knowledge as rules in the system.

There are several reasons for emphasis on knowledge-based systems in comparison with cognitive reasoning methods, or even conventional programs. First, many of the difficult and interesting problems cannot be reduced to an adaptable algorithm. Conventional programs process data by means of complex algorithms which yield specific quantifyable results. They are written in code only programmers understand. Knowledge-based systems must be user friendly by their very nature. Highly interactive, they require user-supplied answers to questions generated by the program in order to function. Knowledge-based systems allow the user to halt the processing at any time and ask why a particular line of questioning is being pursued or how a particular conclusion was reached.

Second, human experts have had a good track record at problem-solving in their fields because they are knowledgeable. If a computer can be programmed to have this knowledge it follows that it too can have good problem-solving performance.

Third, knowledge engineers and experts maintain knowledge-based systems, while programmers maintain conventional programs. The action of a knowledge-based system is to reason through a problem, not merely execute a mathematical model. Consequently, the skills of experts in the given field are required in order to transfer their abilities to the computer and keep those abilities current.

Fourth, the knowledge base is readable, and can be modified easily. The experts' knowledge can be represented by sets of "if-then" rules written in plain English.

Barr and Feigenbaum suggest that in order for expert systems to be useful and perform at a significant level of expertise they must include:

- Facts about the domain.
- Hard and fast rules or procedures.
- Problem situations and what might be good things to do when you are in them (heuristics).
- Global strategies (methods of approaching any problem within the overall domain).
- Differential diagnoses (methods of breaking specific large problems into smaller ones to solve).
- Possible theories about the domain itself (how and why the domain is the way it is). [Ref. 6: p. 81]

Aircraft troubleshooting readily fits into the context of a knowledge-based expert system. Fault diagnosis involves inferring possible causes from a list of observable conditions and potential flaws in system components, following an "if-then" format. In the H-46 troubleshooting tables, the observable conditions are the results of the isolation procedures listed for each symptom (see Figure 2.2). Each of these "observables" infers a possible problem cause. Given the advantages of knowledge-based systems (compared to conventional programming), Barr and Feigenbaum's must-have items, plus the "if-then" structure of troubleshooting, a diagnostic system fits perfectly into the framework of a knowledge-based expert system.

B. DESCRIPTION OF M.1.

According to Joseph S. Yavorsky, the M.1 expert system shell operates as follows:

> M.1 is a sophisticated knowledge engineering tool to aid development of knowledge-based expert systems that are diagnostic/prescription oriented. [It is capable of containing up to 2500 rules and facts [Ref. 7: p. 1-10]. Typical applications contain 100-200 rules]. It is designed to seek a goal defined by the programmer to present a single solution or multiple possible solutions to a problem. M.1 will accept UNKNOWN as an answer, answer questions about its reasoning during a consultation, and will calculate certainty factors for conclusions. It has a sophisticated user interface with windowing capabilities which makes it user friendly, and eases the development of a system. M.1 requires an IBM PC, XT, AT, or compatible microcomputer, using PC DOS 2.0 or later with a minimum RAM of 512K bytes and two disk drives. [Ref. 8: p. 24]

CADS was developed using a series of microcomputers (IBM PC, IBM AT, COMPAQ, ZENITH 248), with hard disk drives and monochrome and color monitors.

Although earlier versions of M.1 were implemented in PROLOG, M.1 version 2.0 and later is written in the C programming language, which is transparent to the user. This affords M.1 the capability of accessing database and calculating programs using C language patches. This capability was not investigated in this research. M.1 can access other M.1 programs. The knowledge base program code written by the developer can be prepared on any standard word processor such as Wordstar, and is in the form of facts and "if-then" rules similar to written English.

1. Inference Engine

Facts are represented as attribute-value pairs called expressions that describe the attributes and relationships of objects. The rules represent application of those objects in certain situations. Facts and rules characterize formal and heuristic knowledge. The M.1 inference engine is described by Yavorsky as follows:

The inference engine will seek values for expressions by methodically considering previously stored conclusions (cached values), relevant knowledge base entries, and information supplied by the user. Previously stored conclusions can be those facts that never change that are resident in the program, or values that have been determined previously during the run of the program. These conclusions are stored in what is known as the cache. Relevant knowledge base entries are the rules and processes in the program which will determine through inferencing, the values for the expression. If values have not been determined by either the search through the cache, or by inferencing, then M.1 asks the user

What is the value of: Expression?

The reference manual for M.1 gives a succinct example of the order in which a value is sought for an expression [Ref. 7: pp. 4-2 - 4-3]:

As an example, consider the simplest possible knowledge base, consisting of a single knowledge base entry:

goal = advice

When you begin a consultation using this knowledge base, the following events take place:

1. The inference engine identifies the goal expression of the consultation and begins to seek a value for **advice.**
2. M.1 checks to see if **advice** is an arithmetic expression for which it can simply compute the value. It is not.
3. M.1 searches the cache for a prior conclusion for **advice.** As no such conclusion yet exists, the search is unsuccessful.
4. M.1 searches through the knowledge base for an entry that can help conclude a value for **advice.** No such entry exists, so again the search fails.
5. M.1 asks a question:

   What is the value of: advice?

   to which you may respond:

   > > sell < Enter >
6. The system has found a value for its goal expression. M.1 displays the conclusion, along with its justification, and returns you to the top-level interpreter.

   advice = sell (100%)
   because you said so

   M.1 >

The method M.1 uses to seek values for expressions via knowledge base entries is called backward chaining. [Ref. 8: pp. 24 - 26]

Backward chaining is a control strategy that starts with a goal or an expected conclusion and works backwards, looking for evidence (other goals or expressions) that support or contradict the expectation. Backward chaining is shown in Figure 3.1.

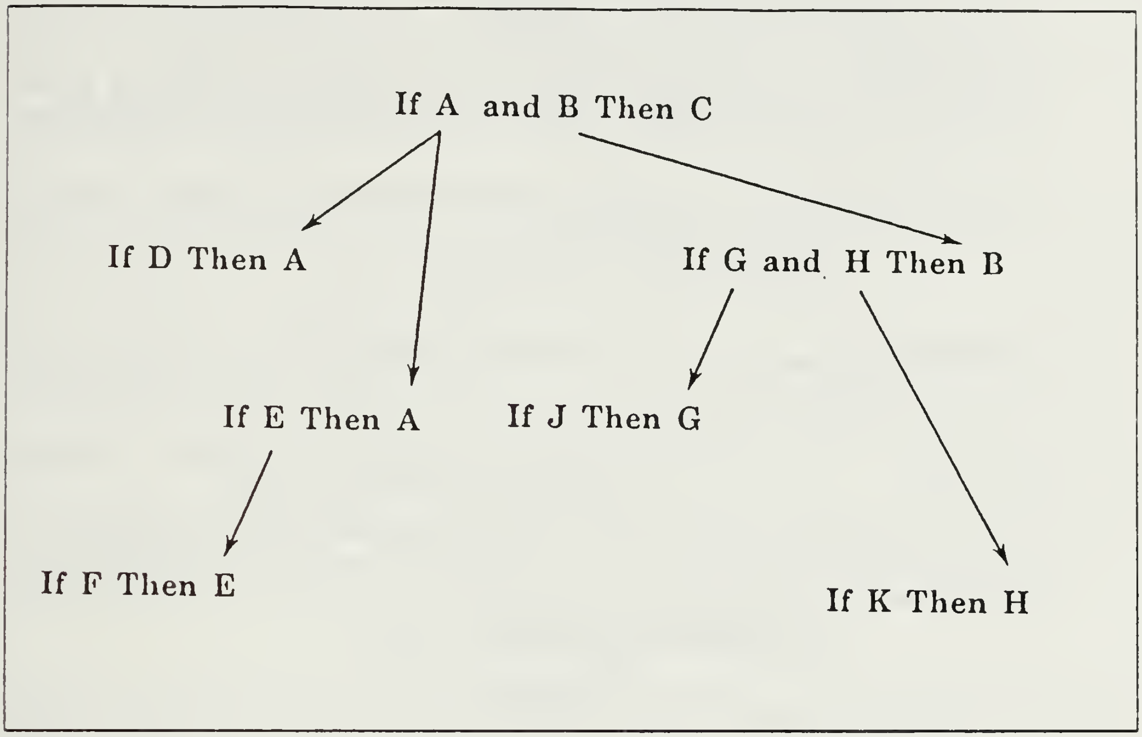


Figure 3.1 Backward Chaining.

The M.1 reference manual gives a good example of its use of backward chaining:

. . . consider the following simple knowledge base:

kb-1: goal = best-color.
kb-2: if main-component = fish
then best-color = white.
kb-3: if day-of-week = friday
then main-component = fish.
kb-4: question(day-of-week) =
'What is the day of the week?'.
kb-5: if best-color = white
then wine = chablis.

When a consultation is run with this knowledge base, the following takes place:

1. M.1 begins seeking the goal expression, **best-color.** After first checking the cache, the inference engine tries to find a knowledge base entry that might conclude a value for **best-color.**
2. Finding **kb-2**, the inference engine then tests the premise of that rule by trying to find a value for **main-component.**
3. After checking the cache and finding no conclusion mentioning **main-component**, the inference engine locates **kb-3** and tries to use it. **kb-3** causes M.1 to seek **day-of-week.**
4. The only knowledge base entry that can help find a value for **day-of-week** is **kb-4**, so you are asked the question:

   What is the day of the week?
   > >

5. If you answer **friday**, M.1 concludes that **day-of-week** is equal to **friday**, and notes that fact in the cache.
6. This causes **kb-3** to succeed, and M.1 notes that **main-component** = **fish** in the cache.
7. This causes **kb-2** to succeed, and the inference engine notes that **best-color** = **white.** Since this is the goal of the consultation, M.1 displays its conclusions and returns you to the top-level interpreter:

   best-color = white (100%)
   because kb-2
   M.1 >

Had you answered anything other than **friday**, all the rules would have failed and M.1 would have indicated that it could not find a value for the goal expression:

best-color was sought, but
no value was concluded.

Note that M.1 does not invoke **kb-5** even though logically it could use **kb-5** to infer that **wine** = **chablis** after the last conclusion was noted. It does not do so because nothing caused it to seek the value of **wine.** M.1 never invokes a rule unless its conclusion provides a value for the expression currently being sought. An expression is never sought unless it is explicitly declared to be a goal (or initial data) or unless it is sought as a result of backchaining from a goal. [Ref. 7: pp. 4-11 - 4-13]

M.1 also has a limited forward chaining capability. Forward chaining starts with known or available information provided by the user, stored in the cache, or with facts in the knowledge base. The goal or conclusions are based on pre-determined rules and the available information. Forward chaining is shown in Figure 3.2. This control mechanism can also be used to seek expressions and conclusions that are in a

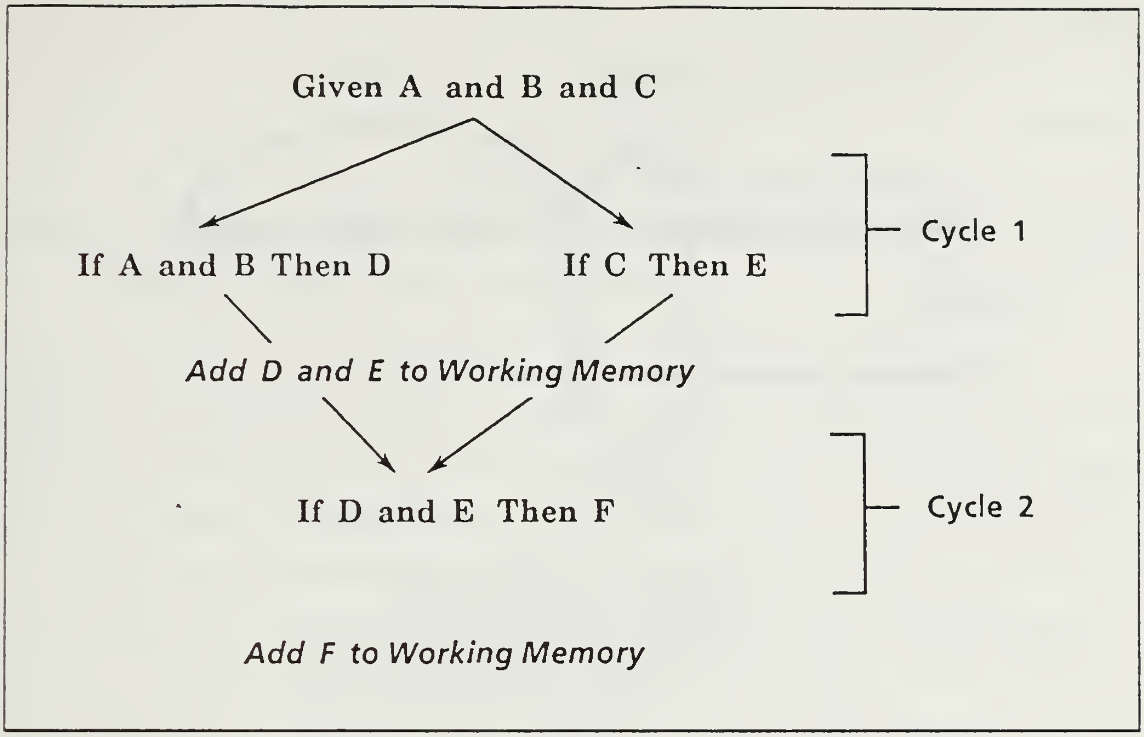


Figure 3.2 Forward Chaining.

sequence-specific pattern. The M.1 command **whenfound** or **whencached** forces the inference process to search rules in a particular sequence. Used in the form

whenfound(EXPRESSION = VALUE) = LIST

means that if the EXPRESSION has the specified VALUE, then the LIST, which can be one or more other EXPRESSIONs, is true. For example, if the discrepancy on an H-46 is "No.1 Generator failed in flight", this is an Alternating Current (AC) electrical system problem. A control mechanism is needed to force M.1 to exclusively search rules that deal with the AC electrical system:

whenfound(major-system = electrical) =

[electrical-sub-system, electrical-sys-symptom].

M.1 will interrupt its backward-chaining process to seek values for the two EXPRESSIONs in the brackets. In the above example, the user would be prompted for values of the two EXPRESSIONs

Which electrical sub-system is the problem located?

1. AC system
2. DC system
3. APU system

The correct response being

> > AC system <Enter>

therefore

electrical-sub-system = AC system.

The value for **electrical-sys-symptom** is found similarly. When it has found those values it resumes its backward-chaining search through rules with those specific values.

2. **Uncertainty**

There are cases when the troubleshooters are not absolutely certain that they have found the specific cause of a problem. Even having followed the proper diagnostic procedures, which led to one cause, there exists the uncertainty that there may be another cause of the problem. M.1 has the ability to represent and use uncertain knowledge. Yavorsky discusses M.1's handling of uncertainty as follows:

> Certainty factors indicate the degree to which a fact is believed as indicated by an integer between -100 and +100, where:
>
> - + 100 represents complete certainty.
> - 20 represents a minimum threshold of belief.
> - 0 represents no evidence for or against.
> - Negative numbers represent belief that the fact is false.
> - - 100 represents complete certainty that the fact is false.
>
> Within M.1 certainty factors less than 100 (the default value) may arise because:
>
> - the answer to a question is qualified by a certainty factor, or,
> - a fact in the knowledge base has an attached certainty factor, or,
> - the conclusion of a rule contains a certainty factor. [Ref. 7: p. 4-16]

As evidence accumulates during a consultation, certainty factors must be combined to come up with a single level of confidence for the final conclusion. In combining two positive certainty factors, the formula used is:

CF-noted = CF1 + (CF2)% of (100 - CF1).

An example is shown in Figure 3.3 [Ref. 7: p. 4-17]. Certainty Factor 1 (CF1) = 50 and Certainty Factor 2 (CF2) = 30. So the Certainty of the conclusion (CF-noted) = 65 or:

65 = 50 + (.30) * (100 - 50).

The combination of two pieces of negative evidence is the same as that for two pieces of positive evidence, with the exception that after the calculation, the negative is taken of the result. The formula is thus:

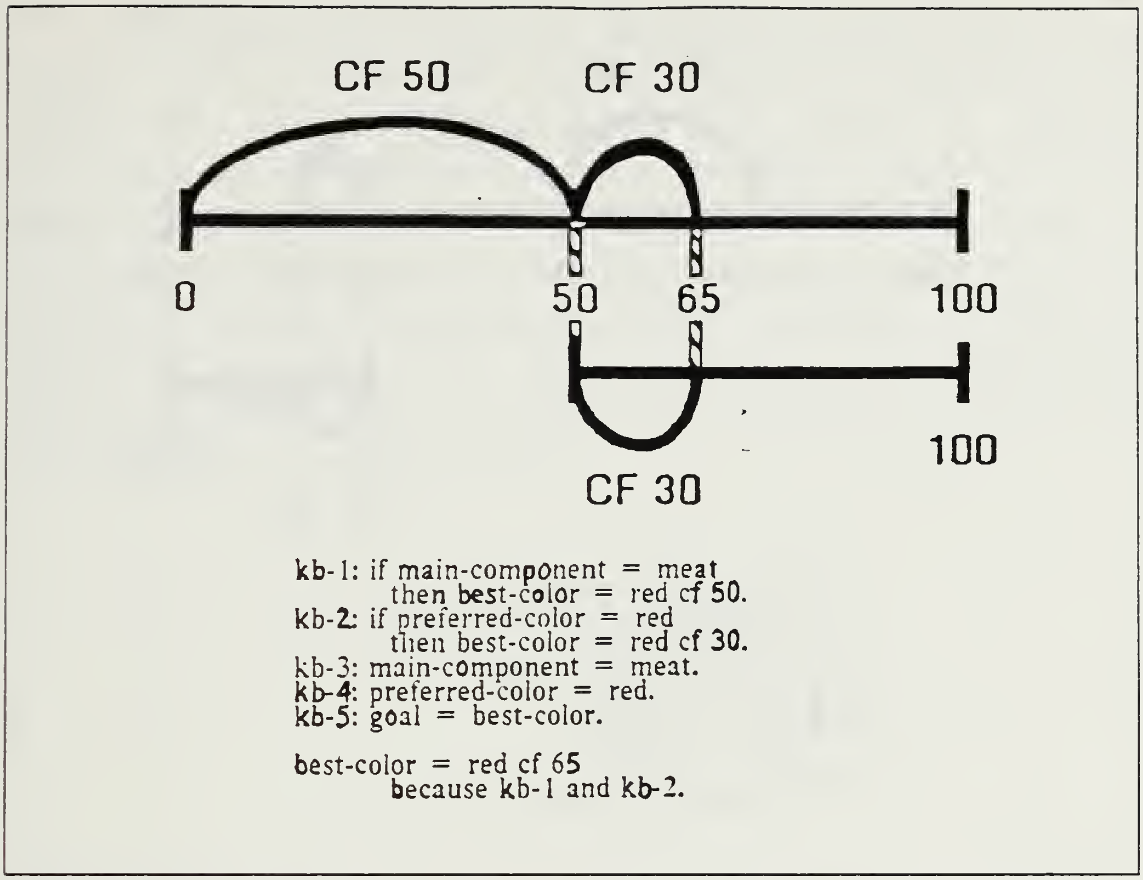


Figure 3.3 Combining Two Positive Certainty Factors.

$$\text{CF-noted} = -(|\,\text{CF1}\,| + |\,\text{CF2}\,|\% \text{ of } (100 - |\,\text{CF1}\,|))$$
$$= \text{CF1} + \text{CF2}\% \text{ of } (100 + \text{CF1}).$$

For example, for a Certainty Factor 1 (CF1) = -50 and a Certainty Factor 2 (CF2) = -30 the Certainty Factor concluded (CF-noted) = -65.

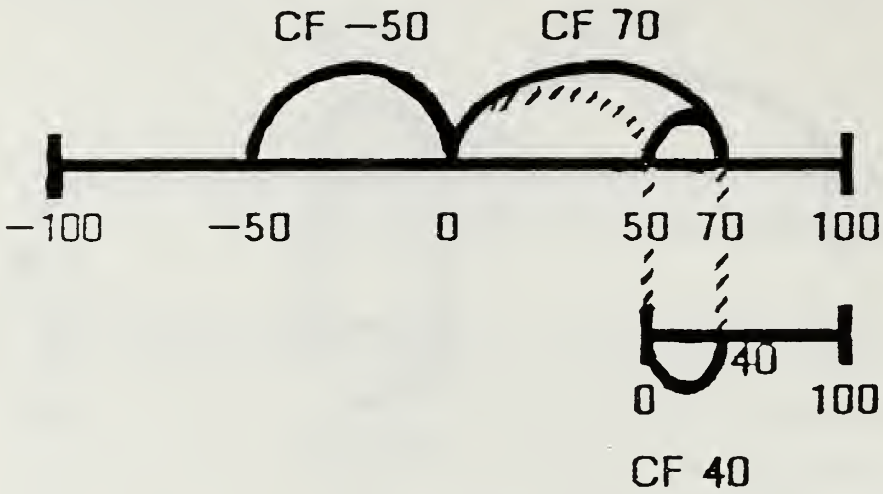
$$-65 = -50 + (-30) * (100 + (-50))$$

To combine both positive and negative evidence, the two certainty factors are added, then the result multiplied by a scaling factor of 100/(100 - A) where A is the smaller of the absolute values of the two factors. The formula:

$$\text{CF-noted} = (\text{CF1} + \text{CF2}) * 100/100 - \text{A}),$$
$$\text{A} = \min(|\,\text{CF1}\,|, |\,\text{CF2}\,|).$$

An example is shown in Figure 3.4 [Ref. 7: p. 4-19]. Certainty Factor 1 (CF1) = -50 and Certainty Factor 2 (CF2) = 70. A = min(| -50 |, | 70 |) = 50. So the certainty of the conclusion (CF-noted) = 40 or:

$$40 = (-50 + 70) * (100/(100 - 50)).$$ [Ref. 8: pp. 27 - 29]




```
kb-1: if main-component = fish
        then best-color = red cf -50.
kb-2: if sauce = tomato
        then best-color = red cf 70.
kb-3: main-component = fish.
kb-4: sauce = tomato.
kb-5: goal = best-color.

best-color = red cf 40
        because kb-1 and kb-2.
```

Figure 3.4 Combining Positive and Negative Certainty Factors.

The Teknowledge Reference Manual provides a list of interesting consequences, when using the above method for calculating the combination of certainty factors:

- The final certainty factor is independent of the order in which evidence is found.
- As positive evidence accumulates, the resulting certainty factor approaches, but cannot pass, 100. Similarly, accumulating negative evidence can approach, but not pass, -100.
- Once the certainty of a conclusion reaches 100 or -100, it cannot be changed again by additional incoming evidence.
- Certainties below 100 cannot combine to produce 100. Certainties above -100 cannot combine to produce -100.
- 0 combined with any certainty factor leaves the certainty factor unchanged.
- Equal positive and negative evidence will exactly cancel out (except +100 and -100, that cannot be changed once concluded). [Ref. 7: pp. 4-19 - 4-20]

This chapter gave Feigenbaum's definition of an expert system. It also provided a discussion of why knowledge-based systems have become popular for problem-solving situations, the advantages of expert systems over conventional programming, and why diagnostic systems meet the criteria for use in a knowledge-based system as defined by Barr and Feigenbaum. The remainder of the chapter was a description of the basic workings of the knowledge engineering tool M.1, based on Yavorsky's discussions. This included the hardware and software requirements for running M.1, how the M.1 inference engine works, and how M.1 handles uncertainty. The next chapter will describe the strategy used to develop the prototype CADS knowledge-based expert system using M.1.

IV. CADS DEVELOPMENT STRATEGY

The strategy used to develop CADS was derived from a combination of three methodologies for building expert systems. They are Harmon and King [Ref. 4: p. 178], Hayes-Roth, Waterman, and Lenat [Ref. 5: pp. 159 - 160], and Teknowledge, Inc. [Ref. 9: pp. 11-1-11-4]. These methodologies can be summarized in the following six steps:

1. Identify and define a problem or task. Test for expert system suitability.
2. Select a tool to build the expert system.
3. Acquire and analyze knowledge of the domain-specific area.
4. Construct several example cases that the expert system will solve.
5. Build a prototype.
6. Test, revise, and expand the prototype.

A. STEP 1: IDENTIFY AND DEFINE A TASK

The task of troubleshooting helicopter system problems was chosen for for this study for several reasons. The author's experience as a helicopter squadron Post Maintenance Functional Check Pilot has indicated that "good" troubleshooting skills are particularly difficult to obtain. These skills are difficult to obtain because of the extensive training and experience required. The lack of comprehensive documentation of helicopter problems and troubleshooting procedures intensifies this difficulty.

Once identified, the task area was tested for expert system feasibility using the criteria outlined by Hayes-Roth, Waterman, and Lenat [Ref. 5: p. 160], and the Teknowledge "M.1 Sample Knowledge Systems" manual [Ref. 9: pp. 11-5 - 11-8]:

1. *The task area should ". . .focus on a narrow specialty that does not involve a lot of common sense knowledge"* [Ref. 5: p. 160]. Most of what is known about H-46 diagnostic procedures is formal knowledge contained in the maintenance manuals. It is well defined, specific, and accepted by the experts. For these reasons this task meets the domain-specific criteria, and does not require the exclusive use of common sense knowledge for problem-solving.
2. *Good domains and tasks are knowledge-intensive.* Knowledge-intensive tasks are domain areas in which higher competence levels are gained through

increased knowledge of and experience in the task. Because of this relationship between competence and knowledge, knowledge-intensive tasks are characterized by considerable differences in peoples' performance ability. The knowledge domain of troubleshooting helicopter problems fits this definition of a knowledge-intensive task.

3. *Facts are accessible and generally understood by typical users.* Facts and terminology about troubleshooting H-46 helicopters are familiar to maintenance people. Novice troubleshooters generally know how to hook up and use equipment such as voltage meters to gather facts about a problem. The diagnostic expert system could ask the troubleshooters for voltage meter test results, which they would understand.

4. *Facts are stable for the duration of a consultation.* Facts about a helicopter problem generally remain true or false throughout a troubleshooting session. For example, if a troubleshooter is using a voltage meter to gather information about a failed generator, the voltage readings will remain the same during a particular troubleshooting procedure. The problem does not change.

5. *Solutions are enumberable.* If all possible solutions to a problem can be listed in advance the solutions are *enumerable.* The maintenance manuals' troubleshooting tables list most of the possible causes for all system symptoms. Those not covered in the tables can be acquired from the experts.

6. *Entities in the domain are discrete.* Helicopter system problems can be readily described as facts, values, and objects. For example, the isolation procedures for troubleshooting the symptom "No.1 GEN FAIL light On" is to test for *generator terminal voltages*, and cross-connect *supervisory panels*, checking if the *generator comes on line.* These conditions are discrete entities with specific values. The values can be "true" or "false", "yes" or "no", or numeric. They are not continuous or "fuzzy" like trying to describe the "attractiveness" of an financial investment.

7. *The task passes the "telephone test".* The domain area of a troubleshooting problem passes the telephone test as described in the Teknowledge "M.1 Sample Knowledge Systems" manual:

    A competent performer could help the intended user of your knowledge system over the phone. This means that a verbal dialog is sufficiently rich to capture the important facets of the problem. If perception plays an important role in problem solving, then the task will fail the telephone test. The knowledge of a master mechanic who depends on the sounds and odors of the engine will be difficult to capture, for example, unless the mechanic and end-user are able to communicate about these cues unambiguously. [Ref. 9: p. 11-8]

After identifying a problem and testing for expert system feasibility, the scope and limitations of the knowledge base were defined. These are described in Chapter I.

B. STEP 2: SELECTING A TOOL

The knowledge system development tool marketed by Teknowledge, Inc., known as M.1, was selected for use on this project after consideration of the available tools. The M.1 tool has many advantages. First, M.1 is designed to prototype diagnosis/prescription consultations [Ref. 4: p. 179]. The problem selected for this study concerns such consultations.

Second, programming with M.1 is relatively easy to learn, and resulting programs are easy to use. It is programmed using "if-then" rules written in near plain English.

Third, M.1 has been applied to automotive diagnostic systems. Diagnostic processes are basically similar, which makes M.1 an excellent tool for the problem considered in this study.

Fourth, M.1 is available for educational purposes at the Naval Postgraduate School by agreement with Teknowledge. The documentation and teaching materials that accompany the M.1 program diskettes are clear and easy to read.

Last, the knowledge base program can be written using any IBM PC compatible word processor. Both SIDEKICK by Borland International, Inc., and WORDSTAR by MicroPro International Corp., which have similar editing commands, were used. SIDEKICK has the advantage of allowing the author to alternate rapidly between the running program and the text editor to make changes to the knowledge base. This facilitates debugging and adding rules. However, SIDEKICK has a limit of approximately 1300 lines of text or program code per notepad file. When a knowledge base module exceeded this limit, WORDSTAR was utilized.

C. STEP 3: ACQUIRE AND ANALYZE KNOWLEDGE

Knowledge acquisition (KA) is probably the most difficult step in building expert systems. It involves ". . .the transfer and transformation of problem-solving expertise from some knowledge source to a program" [Ref. 5: p. 129]. Mark D. Grover describes KA as a three phase process: domain definition, fundamental knowledge formulation, and basal knowledge consolidation. The three phases are shown in Figure 4.1 [Ref. 10].

KA is pertinent throughout expert system development. Because of this importance, Grover's KA cycle was integrated with the six steps of the CADS development strategy. The relationship between the KA cycle and the six development steps is shown in Figure 4.2.

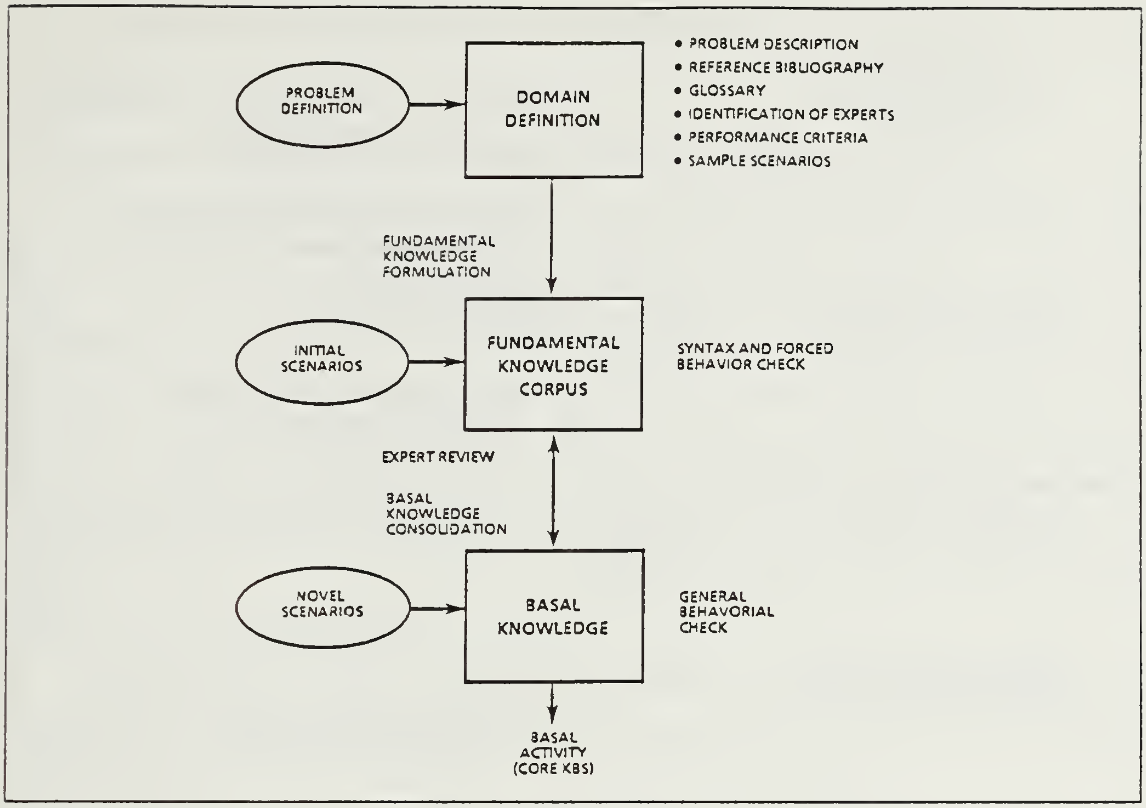


Figure 4.1 Knowledge Acquisition Cycle.

The first phase of the KA cycle is described by Grover as:

> . . . the careful understanding and recording of the domain. The goal is the production of a Domain Definition Handbook containing:
>
> - General problem description,
> - Bibliography of reference documents,
> - Glossary of terms, acronyms, and symbols,
> - Identification of authoritative "experts,"
> - Definition of appropriate and realistic performance metrics, and
> - Description of example reasoning scenarios. [Ref. 10]

This research documentation is considered the Domain Definition Handbook for this project. The general problem description and the identification of reference documentation were accomplished in step one of the development process. The knowledge engineer then must become familiar with the knowledge domain. This is done by acquiring knowledge from books, manuals, technical publications, and

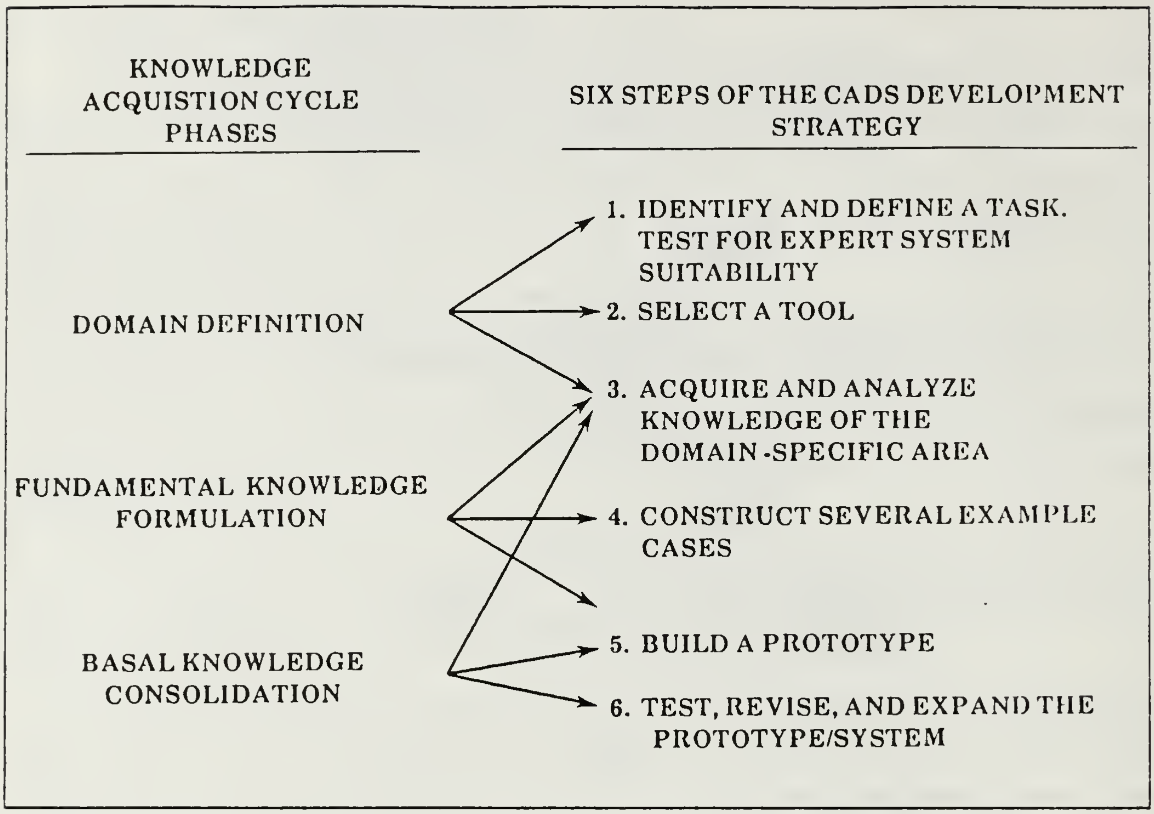


Figure 4.2 KA Cycle and CADS Development Relationship.

pamphlets. The objective is to investigate and become familiar with any structured problem-solving methodology that the domain utilizes. The author as the knowledge engineer familiarized himself with the H-46 diagnostic process. The knowledge and process were analyzed to prepare for the construction of sample cases, which are discussed in step four. The bibliography of reference documents for CADS consists of the H-46 maintenance manuals.

A glossary was initiated (contained in Appendix A) to include acronyms and expressions which are characteristic of Naval aviation maintenance.

Identification of experts in the domain was accomplished by asking the Naval Air Systems Command H-46 Program Manager (NAVAIRSYSCOM PMA-261) to identify technical representatives who are knowledgeable about the recently modified H-46. Five Naval Aeronautical Engineering Service Unit (NAESU) technical representatives were identified. They are located at Marine Corps Air Station Tustin, California, and Naval Air Station North Island, California.

A discussion of the performance metrics and formulation of sample cases is in the next section.

D. STEP 4: CONSTRUCT SEVERAL SAMPLE CASES

Phase two of Grover's KA cycle is fundamental knowledge formulation, which includes steps three, four, and five of the CADS development process. In this phase initial cases are selected and formulated for solution by the expert system. Fundamental knowledge should represent situations that are typical, well understood, and have expected conclusions. When the initial cases have been formulated, the identified domain expert reviews the cases for correctness. "This review forms a baseline for minimum performance, predictable testing and correction, and careful delineation of capabilities which can be expanded and subjected to experimentation" [Ref. 10]. This *core* of reviewed initial cases becomes what Grover calls the Fundamental Knowledge Corpus.

Constructing the initial sample cases can be considered the first step toward designing the system and identifying the performance criteria. The performance metrics of the system were defined as follows:

1. The goal of the initial cases, and ultimately the entire system, should be identified. What is the system supposed to solve?
2. The isolation procedures should be displayed to the user.
3. The system should ask the user questions about the outcomes of the performed isolation procedures.
4. The user should be given a list of possible answers to the questions that the system asks.
5. Based on the user's responses the system should display the correct conclusion about the helicopter problem (goal).

What is the troubleshooter's goal? The goal is to find the *cause* of a helicopter system problem. In terms of M.1:

goal = problem-cause.

Finding the actual cause of a helicopter problem is paramount. Once the actual cause is known then the correct appropriate action can be taken, instead of merely changing components, without really identifying whether the component contains the fault.

After the goal of the system's knowledge base was identified, the initial scenarios were constructed from the symptoms and isolation procedures of the maintenance manuals' troubleshooting tables. The problem causes listed in the troubleshooting tables were examined, then the isolation procedures were analyzed to identify the conditions that would lead a troubleshooter to those causes (goal). The outcomes and conditions of the isolation procedures became the antecedents of rules for which the problem-cause was the conclusion. Several rules were then drafted on paper, written in pseudo-English, i.e., the "if-then" format acceptable to M.1.

Refering to the example discrepancy discussed in Chapter II, the No.1 electrical generator malfunctions on the helicopter. One of the symptoms of this problem is that the No.1 generator failure light has illuminated on the cockpit caution light panel. This symptom is listed in the troubleshooting tables as "GEN FAIL light On". See Figure 2.2. According to the troubleshooting tables, there are two outcomes of the isolation procedure (cross-connecting the supervisory panels): 1) the generator comes on line and the No.1 GEN FAIL light extinguishes, or 2) the generator does not come on line and the light remains illuminated. If the generator comes on line then the supervisory panel should be replaced, implying that the cause is a faulty supervisory panel. The symptom and each of these outcomes are conditions which would lead a troubleshooter to this problem-cause. The rule can be written thus:

if symptom = 'GEN FAIL light On'
and 'generator comes on line'
then problem-cause = 'Faulty Supervisory Panel'.

The CADS program prompts the user for the symptom and possible values for the conditions. In M.1 each of these conditions is recognized as an *expression.* M.1 then uses meta-facts to determine the values of the *expressions.* Meta-facts are knowledge base entries that provide information or directions about how to determine a value. . . [Ref. 9: p. 4-6]. The most common meta-fact used is a **question:**

question('generator comes on line') =
['Having performed the isolation procedure,
does the generator come on line?'].

The answers to this question are displayed using the M.1 meta-fact **legalvals,** which list the legal values of the *expression* 'generator comes on line'. The possible responses to this question are written in the meta-fact:

legalvals('generator comes on line') = ['yes','no'].

Had the user answered 'yes', the *expression* 'generator comes on line' would be true, and the problem-cause displayed, ending the consultation.

Rules and meta-facts were written on paper following this logic for several problem-causes. A set of meta-facts and rules leading to a conclusion (goal) are considered a case or scenario. These meta-facts and rules were then coded into the program for an initial prototype, which is discussed next.

E. STEP 5: BUILDING A PROTOTYPE

The strategy for building the initial CADS prototype was to use the adaptive design method, better known as prototyping. Using the prototyping method, a knowledge engineer can 1) build a working system quickly, 2) evaluate the appropriateness of the initial design quickly without investing large amounts of time and energy into an unworkable design, and 3) make adaptions to the design, user interfaces, and outputs. Once an initial working prototype is developed, the knowledge engineer can modify and refine the program as necessary.

The ability to make changes and adaptions to the CADS knowledge base system design is important. When the initial CADS prototype was completed, interviews with the H-46 maintenance experts were conducted to acquire their knowledge of troubleshooting techniques and procedures. They reviewed the MIMs troubleshooting tables for correctness and added new information. The prototype CADS knowledge base was modified accordingly, to include this expert knowledge.

Using the prototyping approach, the first set of rules, questions, and meta-facts were written, entered into M.1, and run. Logic flow and solutions to helicopter discrepancies were compared with those given in the maintenance manuals. As CADS yielded the correct problem-cause, more rules and knowledge base entries were written. The knowledge bases for each sub-system of the electrical and hydraulic systems were completed in this iterative manner.

The logical and physical structures of the CADS knowledge base are similar. The knowledge base is divided into modules. Each module represents a subsystem knowledge base. The electrical system is divided into three subsystems: AC system, DC system, and APU system. The hydraulic subsystems are the flight control system, flight control pressure indicating system, utility power system, and the utility pressure indicating system. This structure is shown in Figure 4.3.

Using this "divide and conquer" approach, each knowledge base module was written independently. This structure was chosen over the option of having one, immense knowledge base. Modularizing the CADS knowledge base allowed the knowledge engineer to concentrate on a specific subsystem, enhancing the ability to debug and revise.

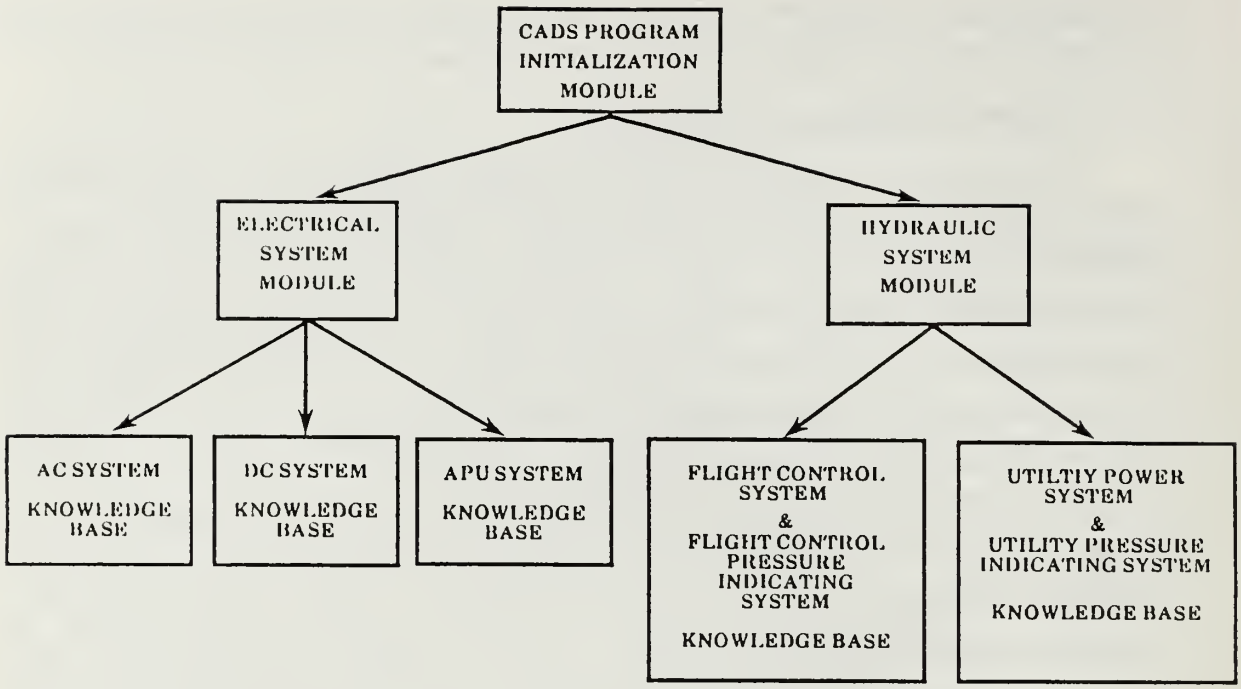


Figure 4.3 CADS Knowledge Base Structure Chart.

Modularizing requires that there be a top-level control module to initialize the CADS program, and to load the appropriate knowledge base file. The top-level module begins the CADS consultation by offering the user directions and asking in which major system and subsystem the aircraft problem is located. The appropriate knowledge base is then loaded.

F. STEP 6: TEST, REVISE, AND EXPAND THE PROTOTYPE

Grover's third phase of the KA cycle is basal knowledge consolidation, which relates to steps three, five, and six of the CADS development strategy. This phase is an iterative process with the objective of testing, improving, and expanding what Grover calls the Fundamental Knowledge Corpus. Basal knowledge can be considered a core knowledge base system that meets the minimum performance metrics listed in step four of the system-building process.

1. **Testing and Verification**

Testing the CADS prototype involved evaluating the advice that was displayed (isolation procedures), the data acquisition process (questioning). and the conclusions reached.

In defining the test criteria the following questions were considered:

- Does the top level module initialize the CADS program properly?
- Is the correct knowledge base loaded?
- Are the correct isolation procedures provided in the correct sequence?
- Are the right questions being asked in the correct sequence, and for the right reasons?
- Are the conclusions correct? Given the responses to the questions, is CADS finding and displaying the correct problem-cause for that combination of conditions?
- Are the isolation procedures, conditions, and conclusions consistent with the experts?

To answer these questions, testing was performed in two phases. The first phase of testing involved evaluating the system for consistency with the maintenance manuals' troubleshooting tables. Because of the known inadequacies of the maintenance manuals (out of date, inaccurate, lack of information), a second phase of testing and verification was necessary. This second phase was to check the prototype knowledge base for consistency with the experts: the five NAESU technical representatives.

The first phase of testing was conducted throughout prototype development. As the rules were written, each conclusion was manually checked to determine that the antecedents (conditions) leading to it were correct. Each antecedent must have a value either solved for by the system, provided by the user, or stored in the cache, for the correct conclusion to be reached. This ensures that questions prompting the user for values are asked at the right time and under the right conditions. CADS has been tested and works according to the information contained in the troubleshooting tables.

As mentioned above in step four, problem causes listed in the troubleshooting tables were examined, and the isolation procedures were analyzed to identify the set of conditions that would lead to those causes. Many inconsistencies were noted in the troubleshooting tables. The sequence of isolation procedures was sometimes vague. Outcomes to procedures were missing or suspect. A copy of the electrical and hydraulic systems troubleshooting tables, with these notations, was sent to the experts for clarification, initiating phase two testing.

The objectives of phase two testing were a review of the CADS knowledge base by the experts, and acquisition of the heuristics of the experts. The experts had approximately three weeks to review and comment on the noted discrepancies of the MIMs tables. Then face-to-face interviews with the experts were conducted.

During the interviews, the experts and knowledge engineer reviewed the discrepancies noted in the troubleshooting tables. The experts provided up-to-date, correct information of the electrical and hydraulic systems troubleshooting procedures. They also had the opportunity to run and evaluate the CADS prototype.

2. **Revision**

Based on the experts' review of the CADS prototype and the troubleshooting tables, revision of the knowledge base took place. Rules, display of isolation procedures, and questions were rewritten as necessary. After this revision, CADS was considered a completed prototype for the purposes of this project. Its knowledge base contains both formal and heuristic knowledge of the experts.

As discussed above, basal knowledge consolidation is an iterative process. An expert system's knowledge base is continually reviewed, up-dated, and improved by the knowledge engineer in collaboration with the experts. However, further iterations of CADS are beyond the scope of this project.

3. **Expansion**

The rules and meta-facts were revised after the experts' review. There are no plans to expand the CADS knowledge base further at this time.

V. RESULTS, CONCLUSIONS, AND RECOMMENDATIONS

A. RESULTS

Improper troubleshooting techniques result in lost time, unnecessary removal of system components, and in wasted resources. These problems are not exclusive to the Navy's H-46 helicopter community, but are common throughout the military. A possible solution to these problems is the application of expert systems technology -- not only to the Navy's H-46 helicopter community, but to other military maintenance processes.

This study was undertaken to demonstrate the feasibility of applying expert system technology to the Navy's H-46 helicopter maintenance process. A microcomputer-based prototype diagnostic system known as CADS was developed for this purpose. The prototype CADS does what it was designed to do: aid H-46 squadron maintenance personnel find the cause of helicopter electrical and hydraulic systems problems. Therefore, the answer to the main question addressed in this study is "yes", a computer-aided diagnostic system is feasible and and applicable to the Navy's H-46 helicopter maintenance process. The success of CADS suggests that expert system technology may be appropriate for general application to military maintenance systems.

B. CONCLUSIONS ABOUT M.1 AND CADS

1. Application of M.1 to structured selection problem-solving

The M.1 knowledge system development tool is particularly suitable for building knowledge bases that use a structured selection approach to problem-solving. Structured selection is a problem-solving methodology by which the problem-solver systematically progresses through a richly structured search space of enumerated problems. These enumerated problems have been seen and solved before. The task is to select and refine a solution for a particular case rather than create it anew.

It has been shown in this research that diagnosing helicopter problems uses such a structured selection approach to problem-solving. Symptoms and causes of helicopter problems are enumerated and contained in the maintenance manuals troubleshooting tables. Most or all of the problems have been solved throughout the history of the H-46 helicopter. Troubleshooting of these problems, plus any new ones that arise, is conducted daily. As a result, diagnosing helicopter problems provides a good example of an M.1 problem-solving application.

2. **Memory requirements for CADS**

The prototype CADS developed for this research is contained on two 5.25-inch, 360K byte floppy diskettes. One diskette contains the CADS initialization module and the CADS executable file, using 246K bytes of disk space. The second diskette contains the modularized subsystem knowledge bases, which use 139K bytes of disk space. These knowledge base files can be called by the initialization module whether they remain on the diskette or have been copied to a hard disk drive.

The CADS executable file can be configured for use on two floppy disk drives or on a hard disk drive. Use of floppy diskettes in the two drive configuration ensures portability of the program. However, the recommended configuration is for use on a hard disk drive, for several reasons. First, a CADS knowledge base is loaded much faster, and the inference process executes much faster on a hard disk drive: knowledge base files can be called and loaded within 30 seconds by the initialization module. This operation can take over five minutes in the two drive configuration, depending on the knowledge base size (number of rules and meta-facts).

Second, the hard drive configuration is recommended in view of the recent contract for Navy-wide implementation of Zenith 248 microcomputers. Currently, each squadron has one such microcomputer. It is anticipated that more will be installed in the future.

3. **Modularizing the CADS knowledge base**

Modularizing the CADS knowledge bases has several advantages. First, a more efficient use of memory space is realized by having only the initialization module and the executable file (both on CADS diskette #1) remain resident in RAM. Much less microcomputer memory space is required with this configuration. The knowledge base files remain on CADS diskette # 2, and are accessed as necessary.

Second, the inference process is more efficient. Since only one knowledge base module is loaded at a time, the inferencing process searches for values and rules pertinent to that particular helicopter system. Computer processing time and memory is not wasted in searches down branches of irrelevant systems.

Third, modularity allows for expansion of the system, without having to increase the requirement for the microcomputer's RAM space. Because information about the helicopter systems is divided into separate knowledge base files, additional modular knowledge bases could be developed and written, independent of the original prototype. These additional knowledge bases could be integrated into the system, with little modification required of the initialization module. The modification basically requires the addition of valid answers to existing questions, and the addition of a rule to load each new knowledge base.

4. **Using the trace function to demonstrate reasoning**

CADS provides an excellent training tool for inexperienced troubleshooters. However, the original proposal for using the trace function to show the reasoning process has not been satisfactory in this M.1 implementation.

M.1's inference tracer displays information describing the inference process. Also displayed are questions, menus, prompts, expressions being sought, which rules have succeeded or failed, and which knowledge base entries are being fired. The trace function is intended for use by the knowledge engineer in the development stage, for finding subtle faults in the knowledge base. As a result, M.1's trace function traces through the knowledge base seeking values of rules which are not relevant to finding the cause of the problem. This reasoning is not necessarily similar to an expert's reasoning method.

Another disadvantage of the trace function is that rules and expressions are displayed exactly as they are written in the knowledge base. Readability of the rules and expressions displayed becomes a function of the programmer's writing style. Although the rules are written in "if-then" format using near plain English, they may seem cryptic to a user unfamiliar with the language of expert systems and M.1.

5. **Training knowledge engineers for the Navy**

The possibility exists for new system discrepancies to appear throughout the H-46 fleet. If CADS were implemented, there would be a need for people within the helicopter squadrons who are able to document and program this new information into existing CADS knowledge bases. These people must be familiar with the language of expert systems, and in particular with the use of the M.1 knowledge development tool.

Because of the ease of developing expert systems utilizing this tool, people with little or no programming experience could be trained for this purpose. The U.S. Army Signal School at Ft. Gordon, Georgia, has such a training program. Army personnel with expertise in particular technical areas, but with no programming experience, attend a two-week training course. They learn how to use M.1, and develop small useable systems within that time. The Navy could set up a similar school for this purpose. People with this valuable training could return to their commands where they could maintain existing CADS software, plus apply this technology to other problem-solving areas, developing new and useful systems.

Knowledge gained from this type of training would be particularly important for improving and expanding the "corporate knowledge" of troubleshooting expertise. As new problems arise and are diagnosed, and as existing procedures are improved and updated, this new knowledge can be programmed and put to immediate use. This could prove to be an advantage over printing changes on paper, for incorporation into updated printed manuals.

C. SUGGESTIONS FOR FURTHER STUDY

1. Knowledge base expansion

The next step after prototype development is to expand the prototype CADS into a fully developed expert system containing all the diagnostic information of all the helicopter systems. A fully developed CADS could also contain the corrective action procedures for each helicopter system discrepancy, and other amplifying maintenance information. Such a long term project might involve a more structured analysis and design of diagnostic information about the entire helicopter. However, the prototyping development approach could be integrated with the information structuring.

Additional knowledge acquisition would also be required to extract all pertinent knowledge from the maintenance manuals and the experts. This would involve more interviewing and survey hours than that required for the development of the CADS prototype.

If CADS were expanded to include other helicopter systems, more secondary memory would be required for the program. Including all helicopter subsystems as additional knowledge bases would increase the requirement for floppy diskettes significantly. If the user has a microcomputer with only two floppy disk drives, it will be necessary to load and unload the diskettes, whenever troubleshooting different helicopter systems is required. Also, loading and inferencing times will be significantly slower. Although the expanded CADS would be portable, it is recommended that an expanded system be used on a computer with a hard disk drive.

2. Enhanced use of M.1 explanation facility

The prototype CADS could be improved by providing more specific explanations to users, enhancing readability and making the system more expert. This could be accomplished by utilizing the M.1 **explanation** facility. Customized explanations, amplifying information about a path of reasoning, or elaboration of heuristic knowledge would be displayed when users enter **why** in response to a question. Ordinally, if a user entered **why**, M.1 would only display the knowledge base entry under consideration.

3. **Enhanced use of the "unknown" response**

The term **unknown** is always an acceptable answer to questions posed in M.1. This gives the M.1 (and CADS) inference process the advantage of continuing to search for values even though a value has not been concluded. It causes the inferencing process to change its search to other branches of the knowledge base, allowing for the solution of other values, to seek a conclusion. The prototype CADS knowledge base could be expanded to provide additional information should the user select **unknown** as an answer. This is accomplished utilizing the M.1 control command **whenfound** to display a text string of additional information.

4. **Integration with graphics**

Ultimately, a fully developed CADS could integrate all diagnostic information, figures, diagrams, and maintenance procedures. Through the use of "pull down" windows, all information would be available on the computer for the maintenance person. This could eliminate the need for printed manuals.

Integration with graphics software could be an initial step toward such a fully developed system. M.1 has the ability to access graphics programs through C language patches. The use of graphics to describe system components would greatly enhance the ability of maintenance personnel to troubleshoot problems. With graphics, component location and troubleshooting procedures could guide personnel through maintenance procedures. This could be extremely useful if the maintenance personnel are unfamiliar with maintenance procedures.

Graphics enhancements such as color, rotating three-dimensional views of components, and rotating transparent views of the helicopter are also possibilities for future study. The use of these graphics techniques would facilitate troubleshooting. Used in a training environment, such graphics techniques could make learning helicopter troubleshooting an exciting, motivating experience, especially for a generation of young mechanics raised with video games.

5. **Integration with CD-ROM technology**

Other futuristic possibilities exist for a CADS integration with Compact Disc Read Only Memory (CD-ROM) technology. This not only would provide an extremely portable medium, but would significantly reduce or even eliminate literally tons of printed paper. This consideration is of particular importance in relation to helicopters deployed aboard ship. Eliminating the need for shipboard office space containing tons of technical manuals is critical.

6. **Study of CADS implementation in the squadrons**

Implementation of CADS in the squadrons and in deployed units offers many interesting possibilities for further study. This research could be conducted using the prototype or a fully developed system. However, a study using a fully developed system would be more desirable, to evaluate the success of implementation. Prior to intiation of any studies, an implementation plan should be developed. Studies could include the amount of training required to familiarize maintenance personnel in the use of microcomputers and CADS. Comparisons could be made of the learning curves for computer literate and illiterate users.

D. RECOMMENDATIONS

The following is a list of recommendations concerning the use of CADS and M.1 in general:

1. Use the M.1 knowledge-based system development tool for building knowledge bases that use a structured selection approach to problem-solving.
2. Implement CADS on a hard disk drive system for efficient use of memory.
3. Keep the knowledge bases modular in large system applications for efficient use of memory space, efficient inference processing, and ease of system expansion.
4. Train Navy personnel to use M.1 to develop immediate and useful systems, and to maintain implemented systems.
5. Expand the CADS knowledge base to include the rest of the helicopter's systems.
6. Improve the M.1 explanation facility, and the "unknown" response to provide more specific information to users, enhance readability of the system, and make the system more expert.
7. Add graphics capabilities as a first step toward a fully developed system. Helicopter component descriptions utilizing graphics would enhance troubleshooting.
8. Investigate the possibilities of integration with CD-ROM technology.
9. Develop a plan for implementation of the system in the helicopter squadrons.

VI. SUMMARY

A microcomputer-based prototype diagnostic system, known as CADS, has been developed to demonstrate the feasibility of applying expert system technology to helicopter maintenance. This system diagnoses electrical and hydraulic system problems in the Navy's CH-46D S,R,&M helicopter.

As an interactive program, CADS prompts the user for information with a sequence of questions, and provides valid answers. The sequence of questions depends upon the answers that the user selects. As the questions become more specific, the program searches the knowledge base for valid causes of the helicopter system problem. Searching is accomplished utilizing a backward chaining, decision tree technique to find solutions.

Chapter II discusses the Naval aviation maintenance process, current troubleshooting procedures, and how CADS could be integrated into the troubleshooting process. The written references for the CADS knowledge domain are two volumes of 15 maintenance information manuals. These two volumes contain the electrical and hydraulic systems. The prototype CADS solves over 39 specific symptoms referred to in over 28 pages of troubleshooting tables. Although this represents only a small portion of the entire published helicopter diagnostic information, it provides enough data for a realistic assessment of CADS. The secondary research questions listed in Chapter I, pertaining to squadron applications, are also discussed.

Chapter III contains a description of knowledge-based expert systems, and the M.1 knowledge-based system development tool. A comparison of knowledge-based systems and conventional programs is made. The criteria necessary for expert systems to be useful and to perform at a significant level of expertise (as suggested by Barr and Feigenbaum) is provided. The description of M.1 includes its capacity of rules and facts, how the inferencing process works, and how M.1 handles uncertainty.

Chapter IV discusses in detail the development of the prototype CADS, and how it works. The strategy used to develop CADS is a combination of three methodologies, resulting in six development steps. Because of the important role that knowledge acquisition has throughout expert systems development, the knowledge acquisition cycle was integrated with the six steps. This relationship is discussed in detail.

Logically, the CADS knowledge base is divided into modules. Each module consists of the knowledge base for a helicopter subsystem, such as the AC electrical subsystem. The advantages of modularizing the knowledge base are discussed. Criteria for a two phase test plan were developed. After testing the prototype CADS, revision of the knowledge base modules was accomplished.

A glossary of acronyms used throughout this research is contained in Appendix A. Appendix B contains a sample CADS consultation using an example electrical system problem. The CADS knowledge base (source code) is in Appendix C.

Based on this discussion, it has been shown that expert systems are well suited for diagnostic applications. The technology, hardware and software exist and are available for use in such an expert system.

APPENDIX A
GLOSSARY OF ACRONYMS

AC	Alternating Current.
AI	Artificial Intelligence.
APU	Auxillary Power Unit.
CADS	Computer-Aided Diagnostic System.
CD-ROM	Compact Disc Read Only Memory.
CF	Certainty Factor, the degree of confidence one has in a fact or relationship. In M.1, CF is assigned a value between +100 and -100.
CH-46D	Designation of S,R,&M Sea Knight helicopter.
CNO	Chief of Naval Operations.
DC	Direct Current.
FY	Fiscal Year.
GEN FAIL	Generator failure.
HC	Designation for Helicopter Combat Support Squadron.
H-46	Designation for the Sea Knight helicopter used in the United States Navy and Marine Corps for utility missions.
JCN	Job Control Number.
KA	Knowledge Acquisition.
kb	Knowledge base entry.
MAF	Maintenance Action Form, for documentation of aircraft discrepancies.
MDR	Maintenance Data Report.
MDS	Maintenance Data System.
MIM	Maintenance Information Manual.
MIS	Management Information System.
MMH/FH	Maintenance Man-hours per Flight Hour.
NAESU	Naval Aeronautical Engineering Service Unit.
NAMP	Naval Aviation Maintenance Program.
NAVAIRSYSCOM	Naval Air Systems Command.
PMA-261	NAVAIRSYSCOM Program Manger for H-46 squadrons.
RAM	Random-access memory: computer's high speed (main) memory.

S,R,&M	Surviveability, Reliability, and Maintainability Program, that extends the service life of the H-46 helicopter through the year 2000.
TD	Technical Directive.
VIDS/MAF	Visual Information Display System/Maintenance Action Form.

APPENDIX B
INTRODUCTORY USER'S GUIDE TO CADS

1. INTRODUCTION

CADS is a prototype microcomputer-based expert system designed to help squadron and detachment maintenance people troubleshoot S,R,&M CH-46D helicopter system problems. The goal of the CADS prototype is to find the exact cause of electrical and hydraulic system discrepancies ("gripes"). CADS contains limited corrective action procedures.

The purpose of this Introductory User's Guide is to get a troubleshooter started and somewhat familiar with use of the CADS prototype. It is not meant to be a comprehensive instruction manual explaining all the various capabilities and peculiarities of the CADS software.

The following assumptions apply to using the CADS prototype:

a. Users are familiar with the use of a microcomputer.
b. Users have access to a Zenith 248 or IBM PC compatible microcomputer, with two disk drives or a hard drive.
c. Users have a copy of the CADS prototype program, either on the two 5.25 inch floppy diskettes, or on the hard drive of the microcomputer.

2. WHAT TO EXPECT

CADS helps troubleshooters find the cause of a helicopter discrepancy by displaying isolation procedures for the troubleshooters to perform, then asking questions about the outcomes of the performed procedures. CADS lists all the possible answers to questions it asks. You as the troubleshooter select the appropriate answer based on the outcomes of the isolation procedures you have performed.

CADS is a computerized version of the maintenance manuals' troubleshooting tables. The intention is to demonstrate that maintenance information can be made accessible on disks and microcomputers, so you do not have to deal with the maintenance manuals.

The best way to approach CADS is to imagine that you are consulting the troubleshooting tables. The difference is the added feature that CADS will ask you the right questions about the isolation procedures you have performed, at the right time, and for the right reason. You will not have to waste time searching the manuals for procedures that are not there anyway. CADS will guide you to troubleshoot discrepancies correctly, completely, and efficiently.

3. WHAT YOU NEED TO USE CADS

To use CADS you will need the following:

a. A Zenith 248, or IBM PC compatible microcomputer with 512K bytes of memory running PC-DOS 2.0 or later.

b. The two CADS prototype diskettes.

c. Two disk drives or a hard drive.

d. A color monitor is recommended, but a monochrome monitor is sufficient.

4. GETTING STARTED WITH CADS.

The initial setup procedures are as follows:

a. Turn on the microcomputer and the monitor, and "boot up" with PC-DOS 2.0 or later.

b. Insert CADS diskette #1 into drive A: .

c. Insert CADS diskette #2 into drive B: .

or

d. If the microcomputer you are using has a hard drive (the Zenith 248 does) copy both diskettes onto drive C: . To do this, at the A> prompt type:

A> **copy a:*.* c:**

Remove CADS diskette #1 from drive A:, and insert CADS diskette #2. Use this copy command again for CADS diskette #2. This puts all the CADS files onto the hard drive (drive C:).

e. At the A> or C> prompt type:

A> **cads** (if you are using two disk drives)

or

C> **cads** (if you are using a hard drive)

in upper or lower case letters.

A full screen banner is displayed momentarily, then a full screen of text welcoming you to CADS is displayed.

5. CADS USER INTERFACE

a. Windows

Refer to Figure B.1. The screen is divided into three *windows*. These permanent areas of the screen are:

1. TROUBLESHOOTING PROCEDURES & POSSIBLE CAUSE window.

This window will display the isolation procedures that you need to perform for the "gripe" you are troubleshooting. After a CADS consultation the cause of the "gripe" will be displayed in this window. A few corrective action procedures are also displayed.

2. QUESTION window

CADS asks you questions during a consultation. The questions will automatically appear in this window as the consultation proceeds.

3. SELECT AN ANSWER window

Answer the questions CADS asks you by highlighting the appropriate answer displayed in this window, and pressing "Enter". Use the arrow (cursor) keys to move the highlighting up or down through the list of answers. The values displayed are the answers acceptable to CADS.

The term **unknown** is always included as a possible answer. However, in this CADS prototype do not select **unknown** when asked for the major system and the subsystem in which the gripe is located.

At the bottom of the window is the term "Space to Mark". This is just a reminder for you to mark the response you have selected by highlighting.

4. CF window

This is the certainty factor window. It allows you to enter numbers from -100 to +100 representing the confidence you have in your selected response. You can enter numbers by using the right arrow key, which moves the highlighter to the CF window. When you have typed in a CF number, press "Enter".

b. **Message Line**

The message line appears at the bottom of the screen. It displays the function keys and the CADS status indicator.

1. F2 Scroll

You can press the **F2** key and the arrow (cursor) keys to scroll backwards and forwards (up and down) through the TROUBLESHOOTING PROCEDURES & POSSIBLE CAUSE and QUESTION windows. The following keys are used for scrolling:

F2	**enable scrolling/move to next window**
↓	**scroll forward (down)**
↑	**scroll backward (up)**
Esc	**disable scrolling and return to the SELECT AN ANSWER window**

2. F10 Help

Refer to Figure B.2. The following help commands are available anytime by pressing the **F10** key (an "on/off" key), and highlighting a help command (described below) and by pressing "Enter", or by pressing the noted keys.

(1) *options for response.* Sometimes you will not be able to read the entire list of all acceptable answers because the SELECT AN ANSWER window is too small, such as when a gripe has a long symptom description. Or, you may want to know what the acceptable responses are to a question. Highlight "options for response", and press "Enter". The "pop up" CADS help menu disappears and the message "Your response must be chosen from the following:" will appear in the TROUBLESHOOTING PROCEDURES & POSSIBLE CAUSE window, with a full list of acceptable answers. When you have found the appropriate response, scroll up or down through the SELECT AN ANSWER window to highlight the appropriate response, then press the "Enter" key.

(2) *abort consultation/ALT - A.* If you want to quit during a CADS consultation, highlight "abort consultation", and press "Enter". A "pop up" box appears asking you whether you want to abort the consultation. If "yes", press the "y" key. This will end the consultation, and clear the windows. To restart the consultation press "Alt" and "G" simultaneously.

(3) *Restart CADS.* If you want to exit from CADS and return to the operating system, or return to the main menu, highlight "Restart CADS". A "pop up" box appears asking "Restart CADS (main menu)/Exit to DOS? (y/n)". If "yes" press the "y" key. This will end the consultation, clear the screen, and return you to the DOS A> or C> prompt. To return to the main menu type **cads** at the A> or C> prompt.

(4) *go run consultation/ALT - G.* If you have aborted or ended a consultation and you want to start another consultation with the same major system and subsystem, highlight "go run consultation", or press the "Alt" and "G" keys simultaneously.

(5) *restart consultation.* This allows you to continue with a consultation where you left off. It keeps the answers you have already given in the CADS memory.

(6) *why is CADS asking/ALT - W.* If you want to know why CADS is asking you a particular question, highlight "why is CADS asking". CADS will then display the rules in the knowledge base that it is considering, as it tries to determine if

they are applicable to the consultation. You can do the same during a consultation by pressing the "Alt" and "W" keys.

(7) *ESC to Cancel.* If you want to escape out of the CADS help menu or any of the help commands, press the "ESC" key. This will cancel the help command, cause the CADS help menu to disappear, and return you to the screen display of the consultation where you left off.

3. Status indicator

The status indicator on the right side of the message line indicates the status of CADS. This will be one of three modes: **READY, LOADING,** or **CONSULTATION.**

6. SAMPLE CADS CONSULTATION

This consultation demonstrates finding the cause of the gripe:

No. 1 generator failure light illuminated in flight

Figures B.3 to B.9 show what the microcomputer screen will look like as the consultation proceeds. Good luck, and remember: DO NOT BE AFRAID OF ANYTHING THAT YOU CAN PULL THE PLUG ON!

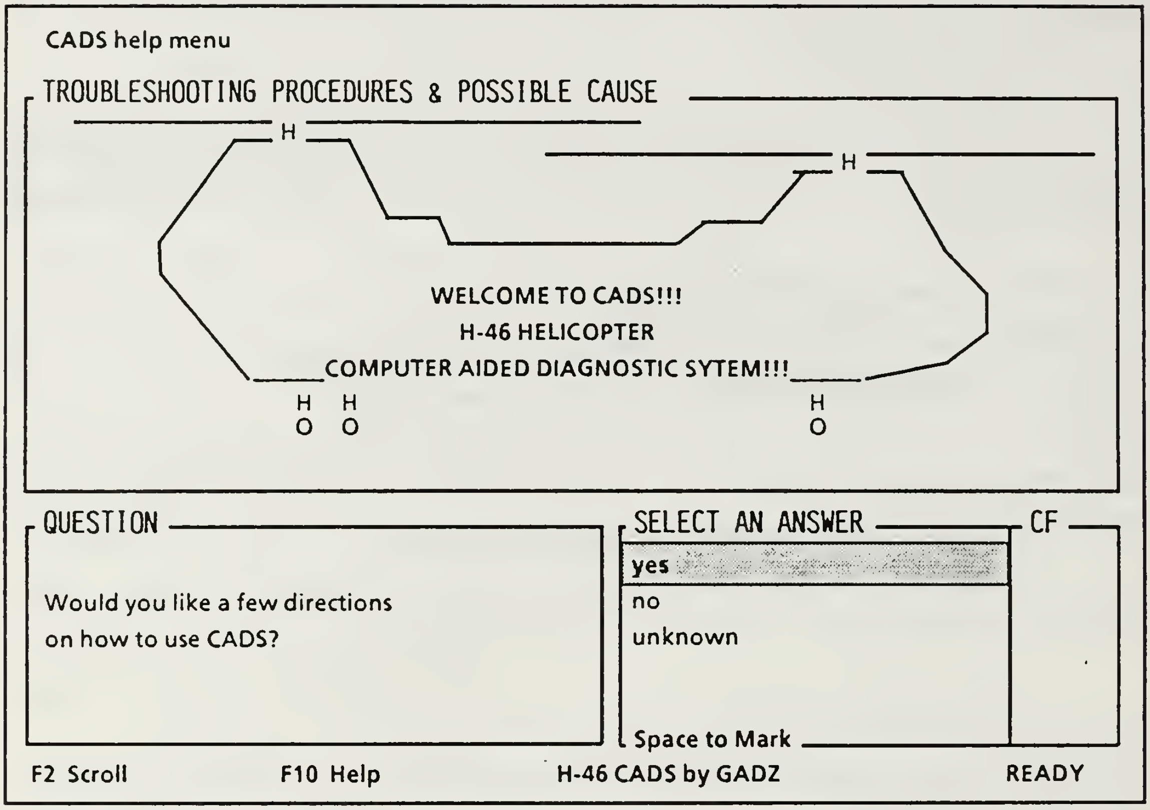


Figure B.1 Initial CADS Screen.

After typing **cads** at the A> or C> prompt, a banner screen displays momentarily, then this initial text screen appears. You can obtain a few brief directions by highlighting "yes" and pressing "Enter". Or, begin the CADS consultation without directions by highlighting "no" and pressing "Enter".

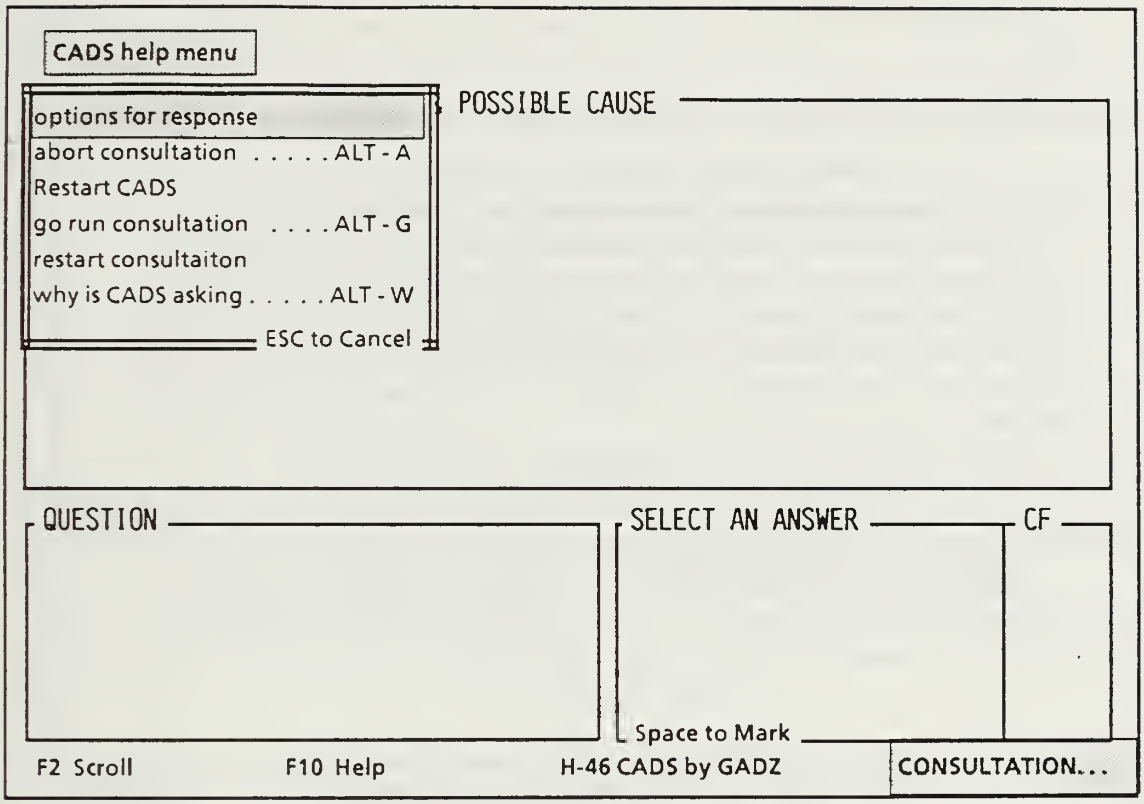


Figure B.2 CADS Help Menu.

The **CADS help menu** is available anytime you press the **F10** key. You may then highlight a help command and press "Enter". Refer to the **F10 Help** section for an explanation of these commands. Notice the status indicator in the message line at the bottom of the CADS screen has changed from **READY** to **CONSULTATION . . .** It will automatically change back to **READY** when you have selected a help command or pressed the "ESC" key. If you want to escape out of the **CADS help menu** press the "ESC" key.

CADS help menu

TROUBLESHOOTING PROCEDURES & POSSIBLE CAUSE

This system is designed to be a prototype microcomputer - based expert system to aid squadron maintenance personnel diagnose aircraft system problems.

1. You will be asked a series of questions about the "gripe" you are troubleshooting.
2. Answer the questions by highlighting your response. Responses are found in the "SELECT AN ANSWER" window in the lower right hand corner of the screen. Use the up and down arrow keys to highlight your response, then press the "Enter" key.
3. To see the list of possible answers to a question, or to see the entire answer, press "F10" then select "options for response".
4. If you get stuck, press the "F10" key to get help from the CADS help menu.

QUESTION

In what major aircraft system is the problem or "gripe" located?

SELECT AN ANSWER — CF

electrical
hydraulic
unknown

Space to Mark

F2 Scroll   F10 Help   H-46 CADS by GADZ   READY

Figure B.3 Second CADS Screen Giving Directions.

The second screen provides brief directions, and asks the first question about the gripe. As a troubleshooter, you know that a No. 1 generator failure light illuminating in flight is associated with the H-46 helicopter's electrical system. Highlight 'electrical' in the SELECT AN ANSWER window, then press "Enter".

Caution: DO NOT select **unknown** at this point. If you do, press the "Alt" and "G" keys simultaneously to return to the main menu.

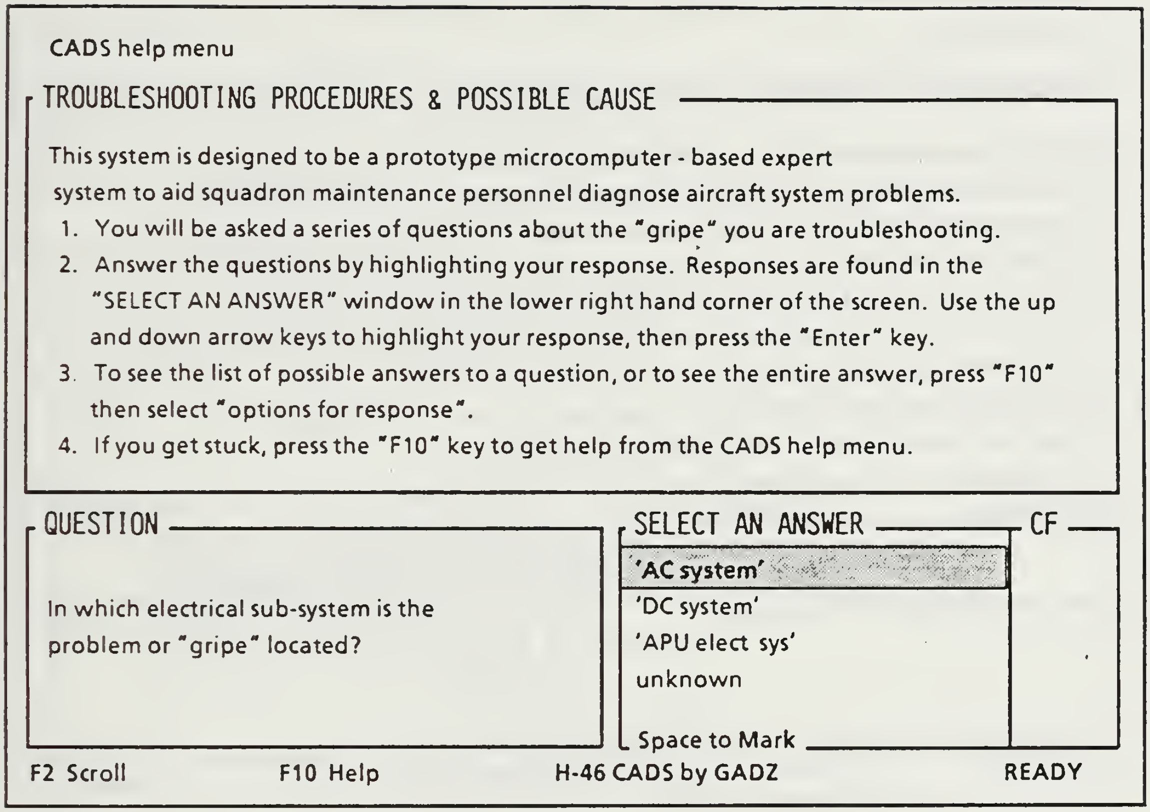


Figure B.4 Third CADS Screen Asking for Electrical Sub-system.

The third screen changes little from the previous, except for the question asking in which electrical sub-system the gripe is located. Knowing that the generators are part of the AC system, highlight 'AC system' and press "Enter". Wait momentarily while CADS loads the AC system knowledge base. You will notice that the status lamp in the lower right corner has changed to **CONSULTATION . . .** When the AC system knowledge base is loaded, it will change back to **READY**, then you can continue.

Caution: DO NOT select **unknown** at this point. If you do, press the "Alt" and "G" keys simultaneously to return to the main menu.

CADS help menu

TROUBLESHOOTING PROCEDURES & POSSIBLE CAUSE

system to aid squadron maintenance personnel diagnose aircraft system problems.

1. You will be asked a series of questions about the "gripe" you are troubleshooting.
2. Answer the questions by highlighting your response. Responses are found in the "SELECT AN ANSWER" window in the lower right hand corner of the screen. Use the up and down arrow keys to highlight your response, then press the "Enter" key.
3. To see the list of possible answers to a question, or to see the entire answer, press "F10" then select "options for response".
4. If you get stuck, press the "F10" key to get help from the CADS help menu.

begin CADS consultation = yes (100%) because kb-18.

QUESTION

What is the symptom in the AC system?

SELECT AN ANSWER — CF

'GEN FAIL LT ON - both Gens
'One ESS BUS LT ON - either
'One or both ESS BUS LT ON -
'One DC BUS LT ON - either
'Opposite DC BUS LT ON - one
unknown

Space to Mark

F2 Scroll   F10 Help   H-46 CADS by GADZ   READY

Figure B.5 Fourth CADS Screen Asking for the AC System Symptom.

The AC system knowledge base is now loaded. You are asked for the AC system symptom. These symptoms are the same as the symptoms listed in the maintenance manuals' troubleshooting tables. The SELECT AN ANSWER window is too small to display the entire symptom. That is no problem. Press the F10 key for the **CADS help menu.** Highlight "options for response", and press "Enter". The full list of acceptable responses will appear in the TROUBLESHOOTING PROCEDURES & POSSIBLE CAUSE window. When you have selected your response, highlight the appropriate response in the SELECT AN ANSWER window. In this case highlight 'GEN FAIL LT ON - both Gens ON - rotors at flight rpm', and press "Enter".

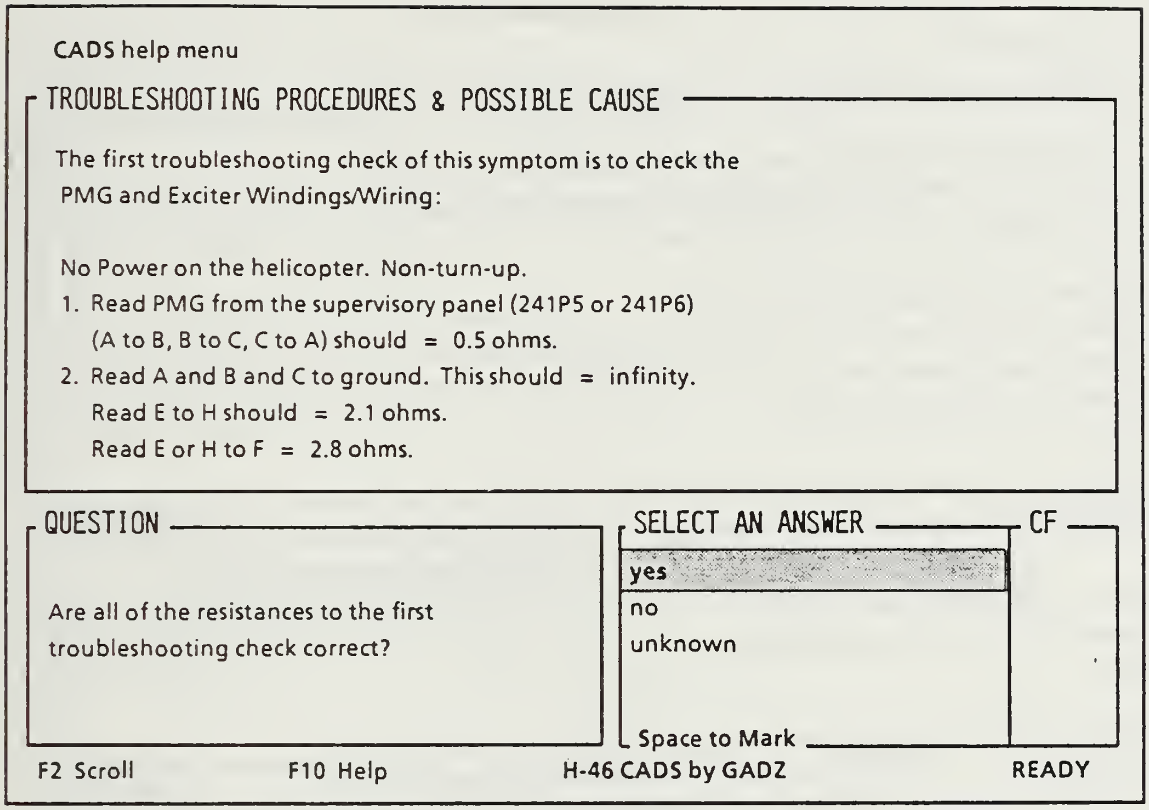


Figure B.6 Fifth CADS Screen Displaying First Isolation Procedure.

The first isolation procedure for this gripe is displayed, along with a question asking about the outcome of this procedure. You now need to perform the procedure to answer the question. Let us say that you performed the isolation procedure and found that all the resistances are correct. Highlight "yes", and press "Enter".

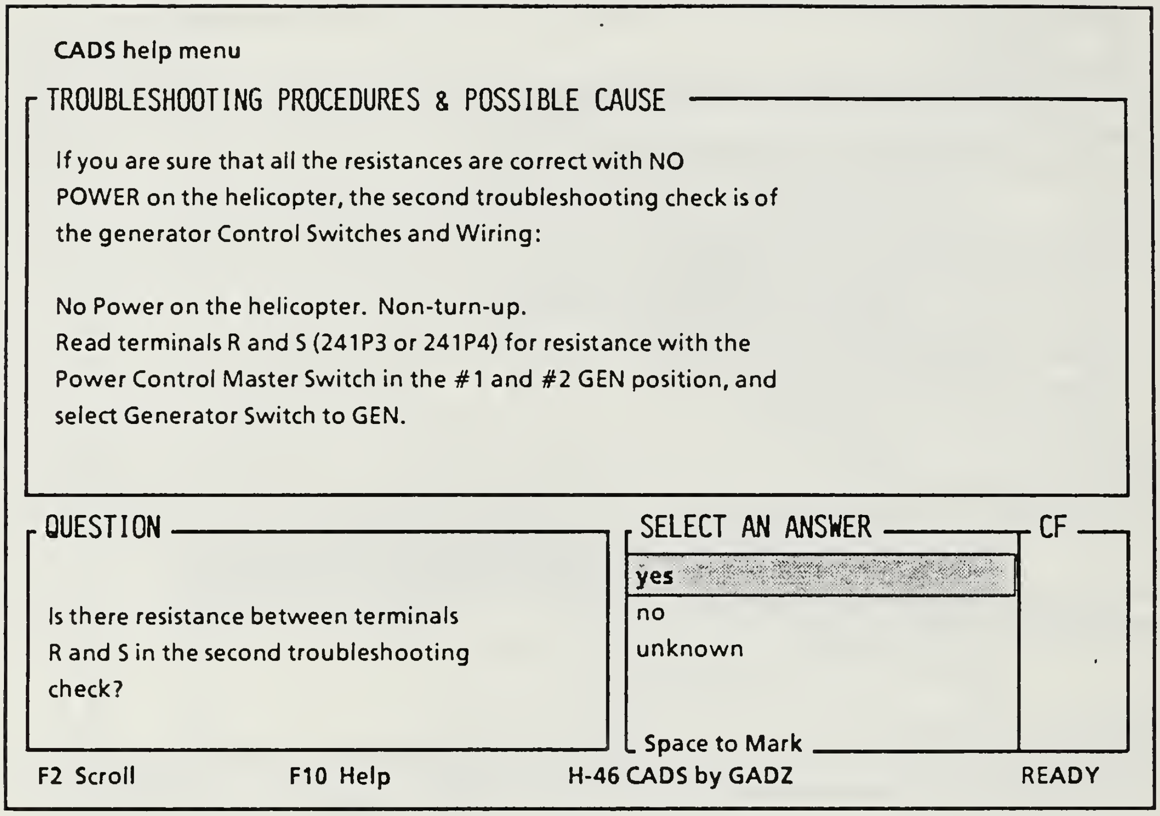


Figure B.7 Sixth CADS Screen Displaying Second Isolation Procedure.

You must now check for resistance between terminals R and S. Let us say that there is resistance. Highlight "yes", and press "Enter".

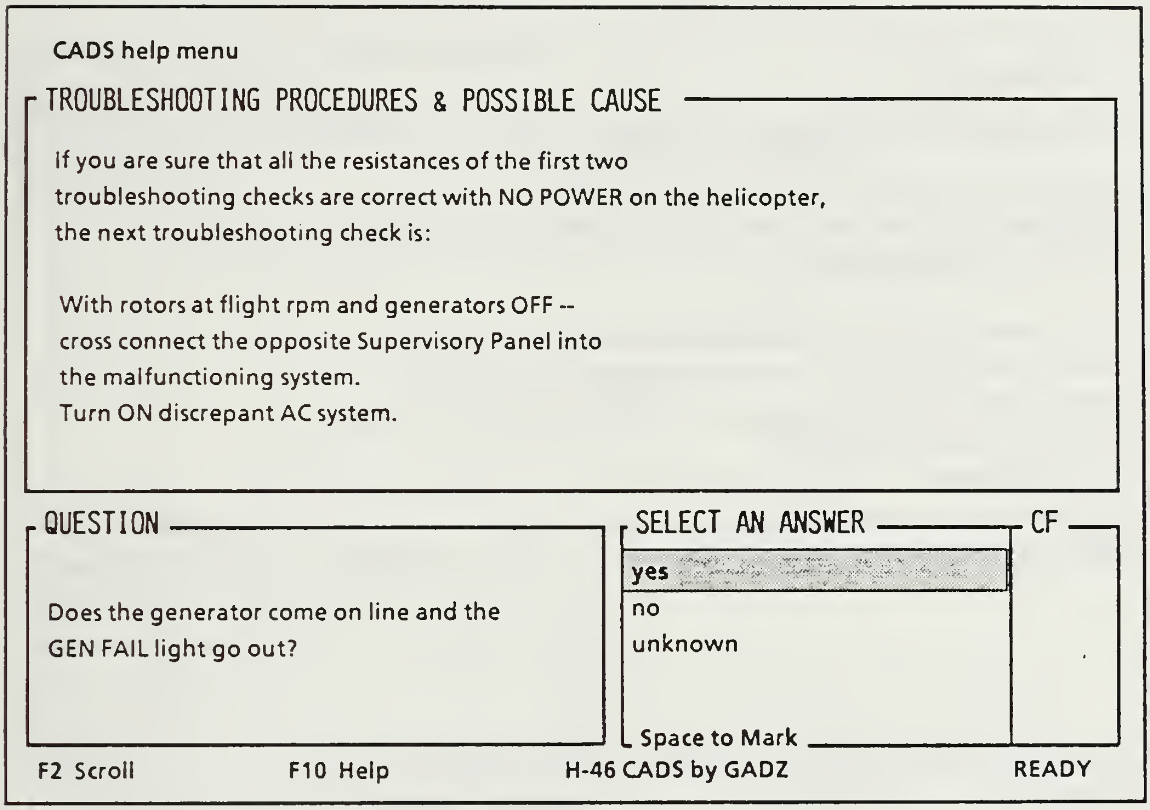


Figure B.8 Seventh CADS Screen Displaying Third Isolation Procedure.

Since your first two troubleshooting checks resulted in correct resistances, you must perform this third isolation procedure. Let us say that you performed it, and found that the generator did come back on line and the GEN FAIL light on the cockpit caution panel went out (you are a pretty good troubleshooter!). Highlight "yes", and press "Enter".

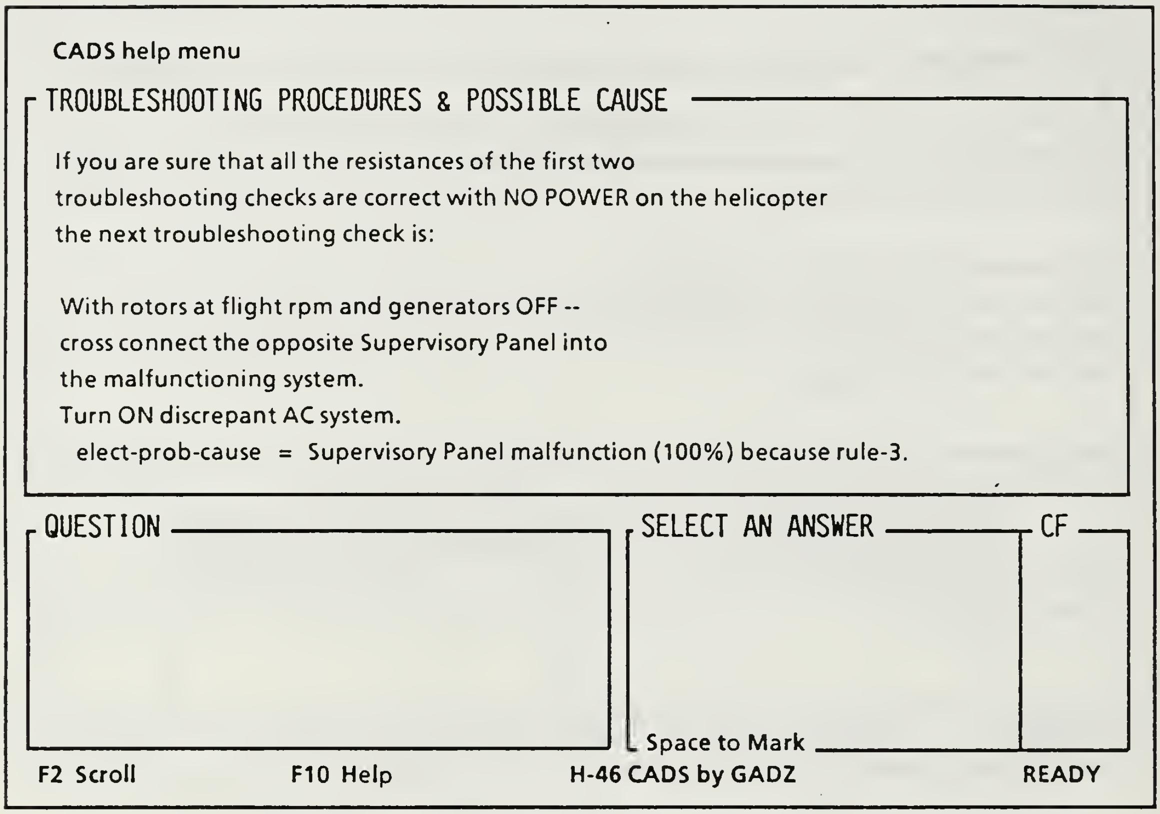


Figure B.9 Eighth CADS Screen - Electrical Problem Cause Found.

Notice that CADS displays a message "elect-prob-cause = Supervisory Panel malfunction (100%) because of rule-3". The consultation has ended, and this message is CADS's way of telling you that the cause of the gripe has been found. The message also informs you that CADS is 100% certain of the cause, and this information is contained in rule-3 of the knowledge base.

APPENDIX C

CADS KNOWLEDGE BASE MODULES

```
/*--------------------START PROGRAM-----------------------*/

goal = 'begin CADS consultation'.
automaticmenu(ALL).
enumeratedanswers(ALL).
initialdata = [helo-display,directions,major-system].
if display([nl,'
    ______________________________
          ___H____          _______________________________________
         /        )                                  ___H___
       /           )                                /       )
      /             )__                          __/         )
     /                 >___________________/                  )
    |                                                          )
     (                  WELCOME TO CADS!!!                      |
        (                H-46 HELICOPTER                       )
          (_____COMPUTER AIDED DIAGNOSTIC SYSTEM!!!_____ /
                I  I                             I
                O  O                             O
   ',nl])
then helo-display.

question(directions) = ['
Would you like a few directions
on how to use CADS?'].

legalvals(directions) = ['yes','no'].

whenfound(directions = 'yes') =
  display(-nl,'
This system is designed to be a prototype microcomputer-based expert
system to aid squadron maintenance personnel diagnose aircraft system
problems.
 1.  You will be asked a series of questions about the "gripe" you are
      troubleshooting.
 2.  Answer the questions by highlighting your response.  Responses
      are found in the "SELECT AN ANSWER" window in the lower
      right hand corner of the screen.  Use the up and down arrow
      keys to highlight your response, then press the "Enter" key.
 3.  To see the list of possible answers to a question, or to see
      the entire answer, press "F10" then select "options for response".
 4.  If you get stuck, press the key marked "F10" to get help from
      the help menu.',nl-).

whenfound(directions = 'no') =
   display([nl,'
   --------------------------------------------------
   After answering the questions shown below
   wait momentarily while CADS loads the Knowledge Base of the
   major system that you want to troubleshoot.',nl,nl]).


/*-------------------MAJOR SYSTEM--------------------------*/


question(major-system) = ['
  In what major aircraft system is the
  problem or "gripe" located?',nl].

legalvals(major-system) = -'electrical','hydraulic'-.
```

```
/*-----------    SUB-SYSTEM QUESTIONS          --------------*/

whenfound(major-system = 'electrical') = electrical-sub-system.
question(electrical-sub-system) = [nl,'
   In which electrical sub-system is
   the problem  or "gripe" located?',nl].
legalvals(electrical-sub-system) = ['AC system','DC system',
   'APU elect sys'].
whenfound(major-system = 'hydraulic') = hydraulic-sub-system.
question(hydraulic-sub-system) = [nl,'
   In which hydraulic sub-system is
   the problem  or "gripe" located?',nl].
legalvals(hydraulic-sub-system) =
  ['flight control hydraulic system',
   'flight control pressure indicating system',
   'Utility Power System',
   'utility pressure indicating system'-.

/*--------------LOADING ELECTRICAL FILES---------------------*/


if major-system = 'electrical'
   and electrical-sub-system = 'AC system'
   and do( load ac-elect )
then 'begin CADS consultation'.
if major-system = 'electrical'
   and electrical-sub-system = 'DC system'
   and do( load dc-elect )
then 'begin CADS consultation'.
if major-system = 'electrical'
   and electrical-sub-system = 'APU elect sys'
   and do( load apu-elec )
then 'begin CADS consultation'.


/*-------------LOADING HYDRAULIC AND UTILITY FILES----------*/


if major-system = 'hydraulic'
   and hydraulic-sub-system = 'flight control hydraulic system'
   or  hydraulic-sub-system =
   'flight control pressure indicating system'
   and do( load hydr )
then 'begin CADS consultation'.
if major-system = 'hydraulic'
   and hydraulic-sub-system = 'Utility Power System'
   or  hydraulic-sub-system = 'utility pressure indicating system'
   and do( load utility )
then 'begin CADS consultation'.

/**************  END OF INITIALIZATION MODULE  ****************/
```

```
/*****************  MAJOR-SYSTEM:  ELECTRICAL  ******************

        ----ELECTRICAL SUB-SYSTEM:  AC SYSTEM SYMPTOMS---------*/

goal = elect-prob-cause.

automaticmenu(ALL).
enumeratedanswers(ALL).

presupposition(ac-sys-symptom) =
   major-system = 'electrical'
   and electrical-sub-system = 'AC system'.

question(ac-sys-symptom) = [nl,nl,'
   What is the symptom in the AC system?',nl].

legalvals(ac-sys-symptom) = [
   'GEN FAIL LT ON - both Gens ON - rotors at flight rpm',
   'One ESS BUS LT ON - either or both Gens ON - rotors at flight rpm',
   'One or both ESS BUS LT ON - assoc. with one Gen ON - flight rpm',
   'One DC BUS LT ON - either or both Gens ON - rotors flight rpm',
   'Opposit DC BUS LT ON - one Gen ON - rotors flight RPM'].


/*--------             AC SYSTEM SYMPTOM:
      GEN FAIL LT ON - BOTH GENS ON - ROTORS AT FLIGHT RPM  ------*/

whenfound(ac-sys-symptom =
   'GEN FAIL LT ON - both Gens ON - rotors at flight rpm') =
   display([nl,nl,'
   The first troubleshooting check of this symptom is to check the
   PMG and Exciter Windings/Wiring:

   No Power on the helicopter.  Non-turn-up.
   1.  Read PMG from the supervisory panel (241P5 or 241P6)
        (A to B, B to C, C to A) should = 0.5 ohms.
   2.  Read A and B and C to ground.  This should = infinity.
        Read E to H = 2.1 ohms.
        Read E or H to F = 2.8 ohms.',nl,nl,nl]).

rule-1:  if ac-sys-symptom =
   'GEN FAIL LT ON - both Gens ON - rotors at flight rpm'
   and 'first ts check'= no
then elect-prob-cause =
'Discrepant Generator PMG or Exciter Windings/Wiring Faulty'.

question('first ts check') = [nl,'
Are all of the resistances of the first
troubleshooting check correct?',nl].

legalvals('first ts check') = ['yes','no'].

whenfound('first ts check' = yes) =
   display([nl,nl,'
   If you are sure that all the resistances are correct with NO
   POWER on the helicopter the second troubleshooting check is of
   the generator Control Switches and Wiring:

   No Power on the helicopter.  Non-turn-up.
   Read terminals R to S (241P3 or 241P4) for resistance with the
   Power Control Master Switch in the #1 and #2 GEN position, and
   select Generator Switch to GEN.',nl,nl,nl]).

rule-2:  if ac-sys-symptom =
   'GEN FAIL LT ON - both Gens ON - rotors at flight rpm'
   and 'first ts check'= yes
   and 'second ts check' = no
then elect-prob-cause =
'Discrepant Generator Control Switches and Wiring Faulty'.
```

```
question('second ts check') = [nl,'
Is there resistance between terminals
R and S in the second troubleshooting check?',nl].

legalvals('second ts check') = ['yes','no'].

whenfound('second ts check' = yes) =
   display([nl,nl,'
   If you are sure that all the resistances of the first two
   troubleshooting checks are correct with NO POWER on the helicopter
   the next troubleshooting check is:

   With rotors at flight rpm and generators OFF --
   cross connect the opposite Supervisory Panel into
   the malfunctioning system.
   Turn ON discrepant AC system.',nl,nl,nl]).

rule-3: if ac-sys-symptom =
   'GEN FAIL LT ON - both Gens ON - rotors at flight rpm'
   and 'first ts check'= yes
   and 'second ts check' = yes
   and 'generator comes on line'
then elect-prob-cause = 'Supervisory Panel malfunction'.

presupposition('generator comes on line') =
   ac-sys-symptom =
   'GEN FAIL LT ON - both Gens ON - rotors at flight rpm'.

question('generator comes on line') = [nl,'
   Does the generator come on line and the
   GEN FAIL light go out?',nl].

legalvals('generator comes on line') = ['yes','no'].

whenfound('generator comes on line' = no) =
   display([nl,nl,'
   Since the generator did not come on line and the GEN FAIL light
   is still on, the sup panel is probably OK.
   You must continue to troubleshoot the AC system.
   ----------------------------------------------------------------

   With main generators selected and both
   generator switches ON, check for 115 to 120 vac pins A,B, and
   C to ground of the related generator test receptacle
   161J3  161J4  or terminals M, P, and R respectively of the
   terminal board on generator test cable.  See fig. 1,
   A1-H46AE-420-000, section 008 00, page 8.
   ----------------------------------------------------------------',nl,nl]).

presupposition(gen-voltage-reading) =
   ac-sys-symptom =
   'GEN FAIL LT ON - both Gens ON - rotors at flight rpm'.

question(gen-voltage-reading) = [nl,'
   What are the generator voltages reading?',nl].

legalvals(gen-voltage-reading) = ['voltages correct',
   'voltages missing or low','one or two voltages = 0'].

whenfound(gen-voltage-reading = 'voltages correct') =
   display([nl,nl,'
   If you are sure that the generator voltages are correct,
   the generator malfunctions because the phasing connections are
   incorrect or because of a faulty contactor control.

   You must now check the phasing, contactor control, and switch
   control.
   ----------------------------------------------------------------

   Using a phase detector, check for proper phase rotation at
   related test receptacle 161J3 161J4.',nl,nl,nl]).

rule-4:  if ac-sys-symptom =
   'GEN FAIL LT ON - both Gens ON - rotors at flight rpm'
   and not('generator comes on line')
   and gen-voltage-reading = 'voltages correct'
then 'phase rotation' is sought.
```

```
whenfound(gen-voltage-reading = 'voltages missing or low') =
   display([nl,nl,'
   If you are sure that the generator voltages are missing or low,
   the generator malfunctions possibly because the exciter
   windings are defective, PMG output is incorrect, or because of
   a feeder fault.

   You must now troubleshoot the generators PMG output, exciter windings,
   and feeder wires:

   With generators OFF, connect the generator test
   cable - see fig.1 - between the supervisory panel receptacle and
   connector plug  241P5  241P6 on the malfunctioning system.
   At the test terminal board, check for approximately 38 vac
   between ground and terminal E and terminal H.',nl]).

rule-5: if ac-sys-symptom =
   'GEN FAIL LT ON - both Gens ON - rotors at flight rpm'
   and not('generator comes on line')
   and gen-voltage-reading = 'voltages missing or low'
then 'PMG output' is sought.


whenfound(gen-voltage-reading = 'one or two voltages = 0') =
   display([nl,nl,'
   If you are sure that the generator one or two voltages = 0,
   the generator malfunctions possibly because of a feeder fault.

   You must now troubleshoot the generators feeder wires:

   Perform the following with the helicopter shut down and all
   power removed:

   1  Disconnect cables from T4, T5, and T6.
   2  Check for infinite resistance to ground at terminals T1, T2,
       and T3.
   3  If resistance at any terminal is measurable, disconnect
       wire from terminal to determine whether ground fault
       is in generator or on feeder wire.
   4  Check for infinite resistance between terminals T1, T2, and T3.
   5  If any resistance between any combination of terminals is
       measurable, disconnect wires from those terminals to
       determine whether ground fault is in the generator or on
       feeder wires.
   6  Check for zero resistance to ground on wires
       that were removed from terminals T4, T5, and T6.',nl]).

rule-6: if ac-sys-symptom =
   'GEN FAIL LT ON - both Gens ON - rotors at flight rpm'
   and not('generator comes on line')
   and gen-voltage-reading = 'one or two voltages = 0'
then 'feeder fault' is sought.


/*----------------AC:  PHASING CONNECTIONS-----------------*/


question('phase rotation') = [nl,nl,'
   Is the phase rotation correct?',nl].

legalvals('phase rotation') = ['yes','no'].

whenfound('phase rotation' = 'yes') =
   display([nl,nl,'
   If phasing is correct, verify ground phase reference wires are
   connected correctly to generator terminals T4, T5, and T6
   routed through their correct cores in current transformer 241T3
   241T4.

   You must also check the contactor.

   When proper voltage is 115 - 120 vac and
   proper phase is present at terminals A1, B1, and C1 respectively
   but not A2, B2, and C2 of related contactor
   242K1 242K2 check for 28 vdc between pins A and B of contactor
   connector plug 242P1 242P2.
   ------------------------------------------------------------',nl]).
```

```
whenfound('phase rotation' = 'no') =
   display([nl,nl,'
   If phasing is incorrect, locate crossed connections - generator
   terminals T1, T2, T3, or contactor terminals 242K1 242K2 A1,
   B1, and C1.

   You must also check the contactor.

   When proper voltage is 115 - 120 vac and
   proper phase is present at terminals A1, B1, and C1 respectively
   but not A2, B2, and C2 of related contactor
   242K1 242K2 check for 28 vdc between pins A and B of contactor
   connector plug 242P1 242P2.
   -------------------------------------------------------------',nl]).

/*-----------------AC:  CONTACTOR CONTROL------------------*/

question('contactor voltage') = [nl,nl,'
   Is there 28 vdc between pins A and B.',nl].

legalvals('contactor voltage') = ['yes','no'].

rule-7: if gen-voltage-reading = 'voltages correct'
   and 'phase rotation'
   and 'contactor voltage'
   and display([nl,nl,'
   If the 28 vdc is correct, replace the contactor.',nl])
then elect-prob-cause = 'gen malfunction - faulty contactor'.

rule-8: if gen-voltage-reading = 'voltages correct'
   and 'phase rotation'
   and not('contactor voltage')
   and display([nl,nl,'
   If you are sure that the voltage is not 28 vdc, verify pin B to
   ground and pin A of the supervisory panel connector plug 241P3
   241P4.
   Correct discrepant wiring.',nl])
then elect-prob-cause = 'gen malfunction - contactor wiring faulty'.

rule-9: if gen-voltage-reading = 'voltages correct'
   and not('phase rotation')
   and 'contactor voltage'
   and display([nl,nl,'
   If the 28 vdc is correct, replace the contactor.
   The phasing is incorrect also.  Locate the crossed
   connections.',nl,nl])
then elect-prob-cause =
   'generator malfunction - phasing incorrect and faulty contactor'.

rule-10: if gen-voltage-reading = 'voltages correct'
   and not('phase rotation')
   and not('contactor voltage')
   and display([nl,nl,'
   If you are sure that the voltage is not 28 vdc, verify pin B to
   ground and pin A of the supervisory panel connector plug 241P3
   241P4.
   Correct discrepant wiring.
   The phasing is incorrect also.  Locate the crossed connections.',nl,nl])
then elect-prob-cause =
   'generator malfunction - phasing incorrect and faulty contactor'.

/*---------------------AC:  PMG OUTPUT------------------*/

question('PMG output') = [nl,nl,'
   What is the value of the pm voltage
   between ground and terminal E and
   terminal H?

   Enter a number between 0 and 38.',nl].
```

```
legalvals('PMG output') = number(0,38).

whenfound('PMG output' = 38) =
   display([nl,nl,'
   You just entered 38 which is normal voltage - 38 vac.

   Now you need to troubleshoot and check the exciter winding:

   Using the generator test cable as in the PMG output check,
   check for approximately 2.3 vdc between ground and terminal F
   of test terminal board.',nl,nl]).

rule-11: if gen-voltage-reading = 'voltages missing or low'
   and 'PMG output' = X
   and X < 38
   and display([nl,nl,'
      You just entered a voltage below 38 vac, if you are sure that
      the voltage is low, disconnect test cable receptacle at helicopter
      cable plug and repeat measurement.  If voltage does not rise to
      approximately 38 vac, the pm section of the generator is
      defective.',nl,nl])
then elect-prob-cause = 'PMG output incorrect'.


rule-12: if gen-voltage-reading = 'voltages missing or low'
   and 'PMG output' = X
   and X = 38
   and 'exciter dc voltage' = Y
   and Y < 2.3
   and display([nl,nl,'
   You just entered a voltage below 2.3 vdc.  Since the dc voltage
   is low, operate the switch on test cable to OFF position and
   repeat measurement.

   If the voltage remains low, the regulator in the supervisory
   panel is defective.',nl,nl])
then elect-prob-cause =
   'Regulator module in Supervisory Panel Defective'.


/*---------------------AC: EXCITER WINDING------------------*/


question('exciter dc voltage') = [nl,nl,'
   What is the value of the voltage between
   ground and terminal F of test terminal
   board?

   Enter a number between 0 and 38.',nl].

legalvals('exciter dc voltage') = number(0,38).


rule-13: if gen-voltage-reading = 'voltages missing or low'
   and 'PMG output' = X
   and X = 38
   and 'exciter dc voltage' = Y
   and Y > 2.3
   and display([nl,nl,'
   You just entered a voltage above 2.3 vdc.

   Since the dc voltage is in excess of 2.3 vdc, the
   exciter winding in the generator is defective.',nl,nl])
then elect-prob-cause = 'Exciter Winding Defective'.

whenfound('exciter dc voltage' = 2.3) =
   display([nl,nl,'
   You just entered voltage = 2.3 vdc.

   The exciter winding is probably OK.  You now need to check for
   feeder faults.',nl,nl]).
```

```
rule-14: if gen-voltage-reading = 'voltages missing or low'
   and 'PMG output' = X
   and X = 38
   and 'exciter dc voltage' = Y
   and Y = 2.3
   and display([nl,'
   Are we supposed to troubleshoot for FEEDER FAULTS now?',nl,nl])
then elect-prob-cause = 'UNKNOWN - NOT PROGRAMMED YET'.

/*---------------------AC:  FEEDER FAULT-----------------*/

question('feeder fault') = [nl,nl,'
   Is there any measurable resistance
   between any combination of terminals?',nl,nl].

legalvals('feeder fault') = ['yes','no'].

/*-----------------------------

if gen-voltage-reading = 'one or two voltages = 0'
   and 'PMG output' = X
   and X = 38
   and 'exciter dc voltage' = Y
   and Y = 2.3
   and 'feeder fault'
   and display([nl,nl,'
   When you have found whether ground fault is in the generator or
   on feeder wire, correct wire fault or replace defective
   generator as appropriate.',nl,nl])
then elect-prob-cause = 'Feeder Fault'.
-------------------------------------------*/

rule-15: if gen-voltage-reading = 'one or two voltages = 0'
   and 'feeder fault'
   and display([nl,nl,'
   If you are sure that there is measureable resistance between
   any combination of terminals then
   disconnect wires from those terminals to determine whether
   there is ground fault in the Generator or on the feeder wires.

   When you have found whether ground fault is in the
   generator or on feeder wire, correct wire fault or replace
   defective generator as appropriate.',nl,nl])
then elect-prob-cause = 'Feeder Fault or Generator malfunctioning'.

rule-16: if gen-voltage-reading = 'one or two voltages = 0'
   and not('feeder fault')
   and display([nl,nl,'
   If there is no measurable resistance - infinite resistance -
   then the problem is beyond the scope of this program.  Sorry.
   Do not pass go.  Do not collect $300.',nl,nl])
then elect-prob-cause = 'UNKNOWN - NOT PROGRAMMED YET'.

/*------------        AC SYMPTOM:
    ONE ESSENTIAL BUS LT ON - EITHER OR BOTH GENS ON - ROTORS AT
                                            FLIGHT RPM  --------------*/

whenfound(ac-sys-symptom =
   'One ESS BUS LT ON - either or both Gens ON - rotors at flight rpm') =
   display([nl,nl,'
   This is an indication problem.  To troubleshoot:

   Check for 115-120 vac at terminals A2, B2, and C2 to ground,
   of ac main line contactor 242K1  242K2.',nl,nl]).
```

```
rule-17: if ac-sys-symptom =
   'One ESS BUS LT ON - either or both Gens ON - rotors at flight rpm'
   and 'mainlinecont voltage correct'
   and 'wiring continuity'
   and display([nl,nl,'
   If you are sure that the voltage is correct, and the wiring
   continuity readings are good, then
   replace the defective BUS FAIL - phase sensing - RELAY 242k3
   242K4.',nl])
then elect-prob-cause = 'Essential Bus Fail Relay'.

question('mainlinecont voltage correct') = [nl,nl,'
   Is the voltage correct?',nl].

legalvals('mainlinecont voltage correct') = ['yes','no'].

whenfound('mainlinecont voltage correct' = 'yes') =
   display([nl,'
   Check for wiring continuity between the ac main line contactor
   242K1  242K2  A2, B2, and C2 to L1, L2, and L3 of
   BUS FAIL -phase sensing - RELAY  242K3  242K4.']).

question('wiring continuity') = [nl,nl,'
   Is the wiring continuity good?',nl].

legalvals('wiring continuity') = ['yes','no'].

rule-18: if ac-sys-symptom =
   'One ESS BUS LT ON - either or both Gens ON - rotors at flight rpm'
   and not('mainlinecont voltage correct')
   and display([nl,nl,'
   Check for 115-120 vac at terminals A1, B1, and C1 to ground of
   AC MAIN LINE contactor 242K1  242K2.  If those voltage readings
   are correct, replace AC MAIN LINE CONTACTOR.',nl])
then elect-prob-cause = 'Essential Bus Fail Relay'.

rule-19: if ac-sys-symptom =
   'One ESS BUS LT ON - either or both Gens ON - rotors at flight rpm'
   and 'mainlinecont voltage correct'
   and not('wiring continuity')
   and display([nl,nl,'
   If you are sure that the wiring continuity is incorrect repair
   the wiring.',nl])
then elect-prob-cause =
    'Faulty Wiring between AC Main Line Contactor and BUS FAIL RELAY
    terminals'.

/*------------              AC SYMPTOM:

   ONE OR BOTH ESSENTIAL BUS LT ON - ASSOCIATED WITH ONE GEN ON -
                                    ROTORS AT FLIGHT RPM           ------*/


whenfound(ac-sys-symptom =
   'One or both ESS BUS LT ON - assoc. with one Gen ON - flight rpm') =
   display([nl,nl,'
   Check for 115-120 vac at terminals A1, B1, and C1 to ground,
   of ac main line contactor 242K1  242K2.',nl,nl]).

rule-20: if ac-sys-symptom =
   'One or both ESS BUS LT ON - assoc. with one Gen ON - flight rpm'
   and 'ac main line contactor voltage correct'
   and not('A2,B2,C2 voltage')
then elect-prob-cause = 'AC MAIN LINE Contactor'.

rule-21:  if ac-sys-symptom =
   'One or both ESS BUS LT ON - assoc. with one Gen ON - flight rpm'
   and 'ac main line contactor voltage correct'
   and 'A2,B2,C2 voltage'
   and display([nl,nl,'
   The knowledge engineer is pretty sure that with both sets of
   voltage readings correct the ESS BUS indicator is probably
   faulty.',nl])
then elect-prob-cause = 'Faulty ESS BUS indicator light' cf 80.
```

```
question('ac main line contactor voltage correct') = [nl,nl,'
Is the voltage 115-120 vac from terminals
A1, B1, C1 to ground?',nl].

legalvals('ac main line contactor voltage correct') = ['yes','no'].

whenfound('ac main line contactor voltage correct' = 'yes') =
   display([nl,nl,'
   If you are sure that the voltage is 115-120 vac from terminals A1,
   B1, and C1 to ground you have an indicator problem.

   Next, check for 115-120 vac from terminals A2, B2, and C2 to ground, of
   AC MAIN LINE CONTACTOR 242K1.',nl,nl]).

question('A2,B2,C2 voltage') = [nl,nl,'
Is the voltage 115-120 vac from terminals
A2, B2, C2 to ground?',nl].

legalvals('A2,B2,C2 voltage') = ['yes','no'].


whenfound('ac main line contactor voltage correct' = 'no') =
   display([nl,nl,'
   If you are sure that the voltage between terminals A1,B1,and C1
   and ground is NOT 115-120 vac, then you must now
   Check for wiring continuity of generator wiring from
   disconnected terminals T1, T2, and T3 of #1 GEN to terminals
   A1, B1, and C1 of AC MAIN LINE Contactor.',nl,nl]).

question('ac main line contactor wiring continuity') = [nl,nl,'
   Is the wiring continuity good?',nl].

legalvals('ac main line contactor wiring continuity') = ['yes','no'].

rule-22: if ac-sys-symptom =
   'One or both ESS BUS LT ON - assoc. with one Gen ON - flight rpm'
   and not('ac main line contactor voltage correct')
   and 'ac main line contactor wiring continuity'
   and display([nl,nl,'
   Since the first voltages of the first check were incorrect, and
   the wiring continuity is good, then the AC Mainline Contactor
   is faulty.',nl])
then elect-prob-cause = 'Defective AC MAIN LINE CONTACTOR' cf 80.


rule-23: if ac-sys-symptom =
   'One or both ESS BUS LT ON - assoc. with one Gen ON - flight rpm'
   and not('ac main line contactor voltage correct')
   and not('ac main line contactor wiring continuity')
then elect-prob-cause =
   'Defective AC MAIN LINE Contactor wiring continuity defective'.


/*----------          AC SYMPTOM:

    ONE DC BUS LT ON - EITHER OR BOTH GENS ON - ROTORS AT FLIGHT
                                                          RPM ------------*/


whenfound(ac-sys-symptom =
   'One DC BUS LT ON - either or both Gens ON - rotors flight rpm') =
   display([nl,nl,'
   Check for 28 vdc between X1 and X2 of

   BUS FAILURE RELAY  161K3 161K4.',nl,nl]).

rule-24: if ac-sys-symptom =
   'One DC BUS LT ON - either or both Gens ON - rotors flight rpm'
   and 'X1,X2 dc voltage correct'
then elect-prob-cause = 'Defective BUS FAILURE RELAY'.

question('X1,X2 dc voltage correct') = [nl,nl,'
   Is the voltage between X1 and X2  = 28 vdc?',nl].

legalvals('X1,X2 dc voltage correct') = ['yes','no'].
```

```
whenfound('X1,X2 dc voltage correct' = 'no') =
   display([nl,nl,'
   If you are sure that there is no voltage present,
   check for 28 vdc at left or right hand junction box at the DC
   circuit breakers bus bar.',nl,nl]).

presupposition('DC C/B bus bar voltage') =
   ac-sys-symptom =
   'One DC BUS LT ON - either or both Gens ON - rotors flight rpm'.

question('DC C/B bus bar voltage') = [nl,nl,'
   Is the voltage present at the
   DC circuit breaker bus bar?',nl].

legalvals('DC C/B bus bar voltage') = ['yes','no'].

rule-25: if 'DC C/B bus bar voltage' = 'yes'
   and display([nl,nl,'
   If you are sure that there is no voltage present between X1 and X2,
   and
   that voltage IS present at the DC circuit breakers bus bar
   then
   use multimeter and A1-H46AE-WDM-000, WP 004 00 or
   A1-H46AE-WDM-010, WP 004 00
   to fault isolate wiring and repair as necessary.',nl,nl])
then elect-prob-cause = 'Defective BUS FAILURE RELAY'.

rule-26: if ac-sys-symptom =
   'One DC BUS LT ON - either or both Gens ON - rotors flight rpm'
   and not('DC C/B bus bar voltage')
   and display([nl,nl,'
   If you are sure that there is no voltage present between X1 and X2,
   and
   that voltage IS NOT present at the DC circuit breakers bus bar,
   the DC BUS Light indicator is faulty.',nl])
then elect-prob-cause = 'Faulty DC BUS Light indicator' cf 80.


/*-------------         AC SYMPTOM:
 OPPOSITE DC BUS LT ON - ONE GENERATOR ON - ROTORS AT FLIGHT RPM--*/


whenfound(ac-sys-symptom =
   'Opposit DC BUS LT ON - one Gen ON - rotors flight RPM') =
   display([nl,nl,'
   Check for 28 vdc between X1 and X2 at

   DC CROSS-TIE RELAY  161K5.',nl,nl]).

rule-27: if ac-sys-symptom =
   'Opposit DC BUS LT ON - one Gen ON - rotors flight RPM'
   and 'DC XTIE X1,X2 voltage correct'
then elect-prob-cause = 'Defective DC CROSS-TIE RELAY'.

question('DC XTIE X1,X2 voltage correct') = [nl,nl,'
   Is the voltage between X1 and X2  = 28 vdc?',nl].

legalvals('DC XTIE X1,X2 voltage correct') = ['yes','no'].

whenfound('DC XTIE X1,X2 voltage correct' = 'no') =
   display([nl,'
   If voltage is not present, check for 28 vdc between terminal B2
   and ground of DC MAINLINE CONTACTOR  161K1  161K2.']).

rule-28: if ac-sys-symptom =
   'Opposit DC BUS LT ON - one Gen ON - rotors flight RPM'
   and not('DC XTIE X1,X2 voltage correct')
   and 'B2 voltage'
   and display([nl,'
   If you are sure that there is 28 vdc between terminal B2 and
   ground of the DC MAINLINE CONTACTOR, then
   use multimeter and A1-H46AE-WDM-000, WP 004 00 or
   A1-H46AE-WDM-010, WP 004 00
   to fault isolate wiring and repair as necessary.',nl,nl])
then elect-prob-cause = 'Defective DC CROSSTIE RELAY wiring' cf 60.
```

```
question('B2 voltage') = [nl,nl,'
   Is there 28 vdc between B2 and ground?',nl].

legalvals('B2 voltage') = ['yes','no'].

whenfound('B2 voltage' = 'no') =
   display([nl,'
   If you are sure that voltage is not present between B2 and
   ground, then check for 28 vdc between terminal B3 and ground of
   DC MAINLINE CONTACTOR 161K1  161K2.']).

rule-29: if ac-sys-symptom =
   'Opposit DC BUS LT ON - one Gen ON - rotors flight RPM'
   and not('DC XTIE X1,X2 voltage correct')
   and not('B2 voltage')
   and 'B3 voltage'
   and display([nl,'  '])
then elect-prob-cause = 'Defective DC MAINLINE CONTACTOR'.

question('B3 voltage') = [nl,nl,'
   Is there 28 vdc between B3 and ground?',nl].

legalvals('B3 voltage') = ['yes','no'].

rule-30: if ac-sys-symptom =
   'Opposit DC BUS LT ON - one Gen ON - rotors flight RPM'
   and not('DC XTIE X1,X2 voltage correct')
   and not('B2 voltage')
   and not('B3 voltage')
   and display([nl,'
   If you are sure that there is 28 vdc between terminal B3 and
   ground of the DC MAINLINE CONTACTOR, then
   use multimeter and A1-H46AE-WDM-000, WP 004 00 or
   A1-H46AE-WDM-010, WP 004 00
   to fault isolate wiring and repair as necessary.',nl,nl])
then elect-prob-cause = 'Defective DC CROSSTIE RELAY wiring' cf 60.

/*************  END OF AC ELECTRICAL MODULE  *******************/
```

```
/****************** MAJOR SYSTEM: ELECTRICAL **************
          --- ELECTRICAL-SUB-SYSTEM: DC SYSTEM -----------*/

goal = elect-prob-cause.

automaticmenu(ALL).
enumeratedanswers(ALL).

presupposition(dc-sys-symptom) =
  major-system = 'electrical'
  and electrical-sub-system = 'DC system'.

/*----ELECTRICAL SUB-SYSTEM: DC SYSTEM SYMPTOMS---------*/

question(dc-sys-symptom) = [nl,nl,'
  What is the symptom in the DC system?',nl].

legalvals(dc-sys-symptom) = [
  'Equipment Connected To Battery Bus Inoperative',
  'No.1 DC BUS Caution LT ON - rotors operating'].


/*--------DC SYMPTOM: EQUIPMENT CONNECTED TO THE
                      BATTERY BUS INOPERATIVE---------------*/

whenfound(dc-sys-symptom =
  'Equipment Connected To Battery Bus Inoperative') =
  display([nl,nl,nl,nl,'
  ---------------------------------------------------------
  Check battery for defective cells, electrolyte leakage,
  and loose connections.  Check for 21.7 vdc at the battery.',nl,nl,nl]).

presupposition('battery') =
  dc-sys-symptom =
  'Equipment Connected To Battery Bus Inoperative'.

presupposition('battery relays') =
  dc-sys-symptom =
  'Equipment Connected To Battery Bus Inoperative'.

presupposition('battery bus voltage') =
  dc-sys-symptom =
  'Equipment Connected To Battery Bus Inoperative'.

presupposition('battery voltage') =
  dc-sys-symptom =
  'Equipment Connected To Battery Bus Inoperative'.

presupposition('battery switch') =
  dc-sys-symptom =
  'Equipment Connected To Battery Bus Inoperative'.

rule-1:  if dc-sys-symptom =
  'Equipment Connected To Battery Bus Inoperative'
  and 'battery'
then elect-prob-cause = 'Defective Battery'.

question('battery') = [nl,'
  Does the battery have any of these
  problems?',nl].

legalvals('battery') = ['yes','no'].

whenfound('battery' = no) =
  display([nl,nl,'
  Since the battery does not have defective cells, electrolyte
  leakage, loose connections, and the voltage is 21.7 vdc,

  Check the following battery relays for proper operation:
          battery relay
          battery transfer relay
          transformer/rectifier transfer relay.',nl,nl,nl]).
```

```
rule-2: if not('battery')
   and 'battery relays'
   and display([nl,nl,'
   If you are sure that a battery relay, battery transfer relay
   or transformer/rectifier transfer relay is defective, replace it.',nl,nl])
then elect-prob-cause = 'Defective Battery Relay'.

question('battery relays') = [nl,'
   Are any of these battery relays defective?',nl].

legalvals('battery relays') = ['yes','no'].

whenfound('battery relays' = no) =
   display([nl,nl,'
   If you are sure that the battery relays are operating properly,
   then

   Check for 25 vdc at the battery bus.',nl,nl,nl]).

rule-3: if dc-sys-symptom =
   'Equipment Connected To Battery Bus Inoperative'
   and not('battery')
   and not('battery relays')
   and not('battery bus voltage')
   and display([nl,nl,'
   If you are sure that 25 vdc is not present at the battery bus,
   troubleshoot the bus and repair wiring as required.',nl,nl,nl])
then elect-prob-cause = 'Defective Battery Bus'.

question('battery bus voltage') = [nl,'
   Is 25 vdc present at the battery bus?',nl].

legalvals('battery bus voltage') = ['yes','no'].

whenfound('battery bus voltage' = 'yes') =
   display([nl,nl,'
   If you are sure that the battery bus voltage is 25 vdc,
   then

   Check for 25 vdc at the battery.',nl,nl,nl]).

rule-4: if not('battery')
   and not('battery relays')
   and 'battery bus voltage'
   and not('battery voltage')
   and display([nl,nl,'
   If you are sure that 25 vdc is not present at the battery,
   repair or replace wiring.',nl,nl,nl])
then elect-prob-cause = 'Defective Wiring'.

question('battery voltage') = [nl,'
   Is 25 vdc present at the battery?',nl].

legalvals('battery voltage') = ['yes','no'].

whenfound('battery voltage' = 'yes') =
   display([nl,nl,'
   If you are sure that the battery voltage is 25 vdc,
   then

   Check the ground connection to the battery switch for security.',nl,nl,nl]).

rule-5: if not('battery')
   and not('battery relays')
   and 'battery bus voltage'
   and 'battery voltage'
   and not('battery switch')
then elect-prob-cause = 'Defective Battery Switch'.

question('battery switch') = [nl,'
   Is the ground connection to the
   battery switch secure?',nl].

legalvals('battery switch') = ['yes','no'].
```

```
rule-6: if not('battery')
   and not('battery relays')
   and 'battery bus voltage'
   and 'battery voltage'
   and 'battery switch'
   and display([nl,nl,'
   These conditions are not covered in the MIMs.',nl,nl,nl])
then elect-prob-cause = 'UNKNOWN'.

/*----------------DC SYMPTOMS:  NO.1 DC BUS CAUTION LT ON-----*/

whenfound(dc-sys-symptom =
   'No.1 DC BUS Caution LT ON - rotors operating') =
   display([nl,nl,'
   Check for shorted No.1 DC BUS.',nl,nl,nl]).

rule-7: if dc-sys-symptom =
   'No.1 DC BUS Caution LT ON - rotors operating'
   and 'shorted 1 DCBUS'
then elect-prob-cause = 'DC overcurrent condition'.

question('shorted 1 DCBUS') = [nl,'
   Is the No.1 DC BUS shorted?',nl].

legalvals('shorted 1 DCBUS') = ['yes','no'].

whenfound('shorted 1 DCBUS' = 'no') =
   display([nl,nl,'
   Since the No.1 DC BUS not shorted, you must now check
   for proper operation of the No.1 SUPERVISORY PANEL, by
   using the substitution method.',nl,nl,nl]).

presupposition('shorted 1 DCBUS') =
   dc-sys-symptom =
   'No.1 DC BUS Caution LT ON - rotors operating'.

presupposition('sup panel') =
   dc-sys-symptom =
   'No.1 DC BUS Caution LT ON - rotors operating'.

rule-8: if not('sup panel')
then elect-prob-cause = 'Defective No.1 SUPERVISORY PANEL'.

question('sup panel') = [nl,'
   Is the No.1 SUPERVISORY PANEL
   operating properly?',nl].

legalvals('sup panel') = ['yes','no'].

whenfound('sup panel' = 'yes') =
   display([nl,nl,'
   Since the No.1 DC BUS not shorted, and the SUPERVISORY PANEL
   is operating properly you must now check
   for proper operation of the CROSS TIE RELAY performing the following
   troubleshooting procedures:

   Power the helicopter with the APU.
   Check if the No.1 DC BUS stays ON and No.2 DC BUS
   stays OFF.
   If so, the CROSS TIE RELAY or related wiring is defective.

   * Shut down the APU immediately *',nl,nl]).

rule-9: if 'sup panel'
then elect-prob-cause = 'Defective CROSS TIE RELAY'.

/****************  END OF DC ELECTRICAL MODULE  *************/
```

```
/**************** MAJOR SYSTEM:  ELECTRICAL  *****************
 -----  ELECTRICAL SUB-SYSTEM:  APU ELECTRICAL SYSTEM SYMPTOMS   --*/
goal = elect-prob-cause.

automaticmenu(ALL).
enumeratedanswers(ALL).

presupposition(apu-sys-symptom) =
   major-system = 'electrical'
   and electrical-sub-system = 'APU elect sys'.

question(apu-sys-symptom) = [nl,'
   What is the symptom in the APU
   electrical system?',nl].

legalvals(apu-sys-symptom) = [
   'APU ON-ESS BUS + DC BUS LTS ON',
   'APU ON-GRD PWR/APU ON, No.2 DC BUS LT ON',
   'APU Operating Ground PWR/APU ON Both DC Lights ON'].


/*  ---------------APU SYSTEM SYMPTOMS:
      APU OPERATING - GROUND POWER -GRD POWER APU READY BOTH
      ESS BUS LIGHTS AND BOTH DC BUS LIGHTS ON.
      GROUND PWR/APU ON INDICATOR OUT. --------------------------*/


whenfound(apu-sys-symptom = 'APU ON-ESS BUS + DC BUS LTS ON') =
   display([nl,'
   You just choose the following symptom:

     The APU is operating on ground power.
     The switches are in GRD power and APU Ready.
     Both ESS BUS lights and
     both DC BUS lights are ON.
     The Ground Pwr/APU ON indicator is Out.
   ---------------------------------------------------------

     First check for 115-120 vac at terminals L11, L12, L13
     L1, L2, L3  of contactor 243K2.',nl,nl,nl]).

question(apu-volt-rdg) = [nl,'
   What are the APU terminal voltages
   reading?'].

legalvals(apu-volt-rdg) = ['voltages correct','voltages missing or low',
   'one or two voltages = 0'].

whenfound(apu-volt-rdg = 'voltages correct') =
   display([nl,nl,nl,nl,'
   If you are sure that the APU voltages at terminals L11  L12  L13 -
   L1  L2  L3 - are correct  the APU malfunctions because the
   phasing connections are incorrect,
   the contactor control is faulty,
   or the switch control is faulty.
   You must now check the phasing contactor control and switch control.
   -----------------------------------------
   First:
   using a phase detector, check for proper phase rotation at terminals
   L11  L12  L13 - L1 L2 L3 - of contactor 243K2.
   -----------------------------------------',nl,nl,nl]).

rule-1: if apu-sys-symptom = 'APU ON-ESS BUS + DC BUS LTS ON'
   and apu-volt-rdg = 'voltages correct'
then 'apu phase rotation' is sought.

whenfound(apu-volt-rdg = 'voltages missing or low') =
   display([nl,nl,'
   If you are sure that the APU voltages are missing or low,
   the APU might be malfunctioning because of incorrect PMG
   output.

   Troubleshoot by performing the following checks:
```

```
   With the APU operating, measure voltages at the voltage regulator
   plug 242P2 between pins A, B, C and ground.  This voltage should
   be 38 - 40 vac.',nl,nl]).

rule-2: if apu-sys-symptom = 'APU ON-ESS BUS + DC BUS LTS ON'
   and apu-volt-rdg = 'voltages missing or low'
then 'PMG voltage' is sought.

whenfound(apu-volt-rdg = 'one or two voltages = 0') =
   display([nl,nl,'
   If you are sure that one or two APU voltages = 0,
   check the APU generator feeder wires.
   Troubleshoot by performing the following checks:
   When only one or two phases are missing or low,
   the integrity of those wires must be verified.
   With all electrical power off the helicopter, check
   for continuity and no shorts to each of the following
   terminals:  L11, L12, L13 - L1,L2,L3 - of contactor
   242K2, to the disconnected terminals T1, T2, T3 of the
   auxiliary generator - or ground power receptacle pins
   A, B, C.  Check generator post T1, T2, T3 to ground.',nl,nl]).

rule-3: if apu-sys-symptom = 'APU ON-ESS BUS + DC BUS LTS ON'
   and apu-volt-rdg = 'one or two voltages = 0'
then 'APU feeder wires' is sought.


/*--------------APU: PHASING------------------------*/


question('apu phase rotation') = [nl,'
   Is the phase rotation correct?'].

legalvals('apu phase rotation') = ['yes','no'].

whenfound('apu phase rotation' = 'yes') =
   display([nl,'
   If phasing is correct, now check the contactor control by:

   When proper voltages and phasing are present at terminals
   L11, L12, L13 - L1,L2,L3 - contactor 243K2,
   Check for 24 vdc and ground at x - y - coil of contactor
   243K2.
   ----------------------------------------------------- ',nl,nl]).

rule-4: if apu-sys-symptom = 'APU ON-ESS BUS + DC BUS LTS ON'
   and apu-volt-rdg = 'voltages correct'
   and not('apu phase rotation')
then elect-prob-cause = 'Phasing Connections Incorrect'.


/*--------------APU:  CONTACTOR CONTROL-----------------*/


/*---------------------
The following rule is temporarily blocked out because the
knowledge engineer must find out whether combinations of
phasing, contactor, and switch defects are possible.
As the program runs now, it is phasing or contactor or
switch.

whenfound('apu phase rotation' = 'no') =
   display([nl,nl,'
   If phasing is incorrect, locate crossed connections - generator
   terminals T1, T2, T3, or ground power unit, or
   contactor terminals 243K2 L11, L12, L13
   L1, L2, L3.

   Now check the contactor control by:

   When proper voltages and phasing are present at terminals
   L11, L12, L13 - L1,L2,L3 - contactor 243K2,
   Check for 24 vdc and ground at x - y - coil of contactor
   243K2.
   ------------------------------------------------------------',nl]).
    */
```

```
/*-----------------APU:  CONTACTOR CONTROL-----------------*/

question('APU contactor voltage') = [nl,nl,'
   Is there 24 vdc and ground at
   x - y - coil of the contactor?',nl].

legalvals('APU contactor voltage') = ['yes','no'].

rule-5: if apu-sys-symptom = 'APU ON-ESS BUS + DC BUS LTS ON'
   and apu-volt-rdg = 'voltages correct'
   and 'apu phase rotation'
   and 'APU contactor voltage'
then elect-prob-cause = 'APU malfunction - faulty CONTACTOR'.

rule-6: if apu-sys-symptom = 'APU ON-ESS BUS + DC BUS LTS ON'
   and apu-volt-rdg = 'voltages correct'
   and 'apu phase rotation'
   and not('APU contactor voltage')
   and 'battery voltage' = 'battery voltage is good'
   and display([nl,nl,'
   If you are sure that the voltage or ground is incorrect
   fault isolate using a meter and A1-H46AE-WDM-000 - WP 004 00
   or A1-H46AE-WDM-010 - WP 004 00.',nl])
then elect-prob-cause =
   'APU malfunction - CONTACTOR and/or wiring faulty'.

question('battery voltage') = [nl,'
   CHECK THE BATTERY!

   What is the battery voltage?'].

legalvals('battery voltage') = ['battery voltage is good',
   'battery is weak'].

rule-7: if apu-sys-symptom = 'APU ON-ESS BUS + DC BUS LTS ON'
   and apu-volt-rdg = 'voltages correct'
   and 'apu phase rotation'
   and not('APU contactor voltage')
   and 'battery voltage' = 'battery is weak'
   and display([nl,nl,'
   Remember:  A good troubleshooter always verifies that the
   battery is good before doing these checks!
   Get a good battery and begin troubleshooting again!',nl,nl,nl])
then elect-prob-cause =
   'WEAK BATTERY'.

rule-8: if apu-sys-symptom = 'APU ON-ESS BUS + DC BUS LTS ON'
   and apu-volt-rdg = 'voltages correct'
   and 'apu phase rotation'
   and not('APU contactor voltage')
   and 'battery voltage' = 'battery voltage is good'
   and 'APU switch control' = 'continuity good'
then elect-prob-cause = 'APU switch control transfer unit faulty'.

/*---------------APU:  SWITCH CONTROL-----------------------*/

whenfound('apu phase rotation' = 'yes'
   and 'contactor voltage' = 'no'
   and 'battery voltage' = 'battery voltage is good') =
   display([nl,'
   With all electrical power OFF the helicopter,
   check the continuity of ground power/APU power S243S2.
      With switch a APU power,
         read continuity between pins
         Y and W, S and U of connector 243P1 of the transfer unit.
      With switch at GRD PWR,
         read continuity between pins Z and W, and T and U of
         connector 243P1.']).
```

```
rule-9: if apu-sys-symptom = 'APU ON-ESS BUS + DC BUS LTS ON'
   and apu-volt-rdg = 'voltages correct'
   and 'apu phase rotation'
   and not('APU contactor voltage')
   and 'battery voltage' = 'battery voltage is good'
   and 'APU switch control' = 'continuity bad'
   and display([nl,'
   If continuity is incorrect:

   fault isolate switch or wiring using a meter and
   A1-H46AE-WDM-000 - WP 004 00
   or A1-H46AE-WDM-010 - WP 004 00.',nl,nl,nl])
then elect-prob-cause = 'APU switch control continuity defective'.

/*------------APU:  PMG OUTPUT--------------------*/


question('APU PMG output') = ['
   What is the value of the voltage
   at the voltage regulator plug
   243P2 between pins A, B, C
   and ground?

   Enter a number between 0 and 40.',nl].

legalvals('APU PMG output') = number(0,40).

rule-10: if apu-sys-symptom = 'APU ON-ESS BUS + DC BUS LTS ON'
   and apu-volt-rdg = 'voltages missing or low'
   and 'APU PMG output' = X
   and X < 38
   and display([nl,nl,'
   You just entered a value less than 38 vac.
   If you are sure that the voltage is less than 38 vac,
   replace the APU.',nl,nl])
then elect-prob-cause = 'APU PMG output incorrect - DEFECTIVE APU'.


rule-11: if apu-sys-symptom = 'APU ON-ESS BUS + DC BUS LTS ON'
   and apu-volt-rdg = 'voltages missing or low'
   and 'APU PMG output' = X
   and X >= 38
   and X <= 40
   and display([nl,nl,'
   You just entered a voltage between 38 and 40 vac, which is
   normal.
   Replace the voltage regulator.',nl,nl])
then elect-prob-cause =
   'APU PMG output incorrect - DEFECTIVE VOLTAGE REGULATOR'.


/*---------------------APU:  FEEDER FAULT-----------------*/


question('APU feeder wires') = [nl,nl,'
Do you have:
   1. proper continuity AND
   2. no shorts AND
   3. generator post grounded between
      terminals?',nl,nl].

legalvals('APU feeder wires') = ['yes','no'].

rule-12: if apu-sys-symptom = 'APU ON-ESS BUS + DC BUS LTS ON'
   and apu-volt-rdg = 'one or two voltages = 0'
   and 'APU feeder wires'
   and display([nl,nl,'
   If you are sure that there is continuity, no shorts, and that
   the generator post is grounded, then
   there does not seem to be a problem here!',nl,nl])
then elect-prob-cause = 'UNKNOWN NOT PROGRAMMED YET'.
```

```
rule-13: if apu-sys-symptom = 'APU ON-ESS BUS + DC BUS LTS ON'
   and apu-volt-rdg = 'one or two voltages = 0'
   and not('APU feeder wires')
   and display([nl,nl,'
   Locate and repair discrepant wire or connection.
   If open exists inside APU generator remove and replace APU.',nl,nl])
then elect-prob-cause = 'Generator Feeder Wires faulty'.

/*------------APU:  APU OPERATING, GROUND POWER/APU ON
                         NO.2 DC BUS LIGHT ON--------------------*/

whenfound(apu-sys-symptom =
   'APU ON-GRD PWR/APU ON, No.2 DC BUS LT ON') =
   display([nl,'
   Check for 28 vdc at No.2 DC BUS in right hand junction
   box.',nl]).
question('APU DC BUS voltage') = ['
Is the voltage 28 vdc at the No.2 DC BUS
in the right hand junction box?',nl].
legalvals('APU DC BUS voltage') = ['yes','no'].
whenfound('APU DC BUS voltage' = 'no') =
   display([nl,nl,'
   If you are sure that the voltage is not 28 vdc,
   now you must check for 28 vdc at X1 at DC CROSS TIE CONTACTOR
   161K5.
   If 28 vdc is present check ground wiring.',nl,nl]).
rule-14: if apu-sys-symptom =
   'APU ON-GRD PWR/APU ON, No.2 DC BUS LT ON'
   and 'APU DC BUS voltage' = 'no'
   and 'DC CROSS TIE CONTACTOR voltage'
   and 'ground wiring correct'
then elect-prob-cause = 'APU DC CROSS TIE CONTACTOR defective'.
whenfound('APU DC BUS voltage' = 'yes') =
   display([nl,nl,'
   If you are sure that the voltage is 28 vdc,
   now you must check for 28 vdc at X2 of BUS FAIL RELAY
   161K4.
   If 28 vdc is present check ground wiring.',nl,nl]).
question('DC CROSS TIE CONTACTOR voltage') = [nl,'
   Is the voltage 28 vdc at X1 at the
   DC CROSS TIE CONTACTOR?',nl].
legalvals('DC CROSS TIE CONTACTOR voltage') = ['yes','no'].
question('ground wiring correct') = ['
   Is the ground wiring correct?',nl].
legalvals('ground wiring correct') = ['yes','no'].
rule-15: if apu-sys-symptom =
   'APU ON-GRD PWR/APU ON, No.2 DC BUS LT ON'
   and 'APU DC BUS voltage' = 'yes'
   and 'BUS FAIL RELAY voltage'
   and 'ground wiring correct'
then elect-prob-cause = 'APU BUS FAIL RELAY defective'.
question('BUS FAIL RELAY voltage') = ['
   Is the voltage 28 vdc at X2 at the
   BUS FAIL RELAY?',nl].
legalvals('BUS FAIL RELAY voltage') = ['yes','no'].
rule-16: if apu-sys-symptom =
   'APU ON-GRD PWR/APU ON, No.2 DC BUS LT ON'
   and 'APU DC BUS voltage' = 'no'
   and not('DC CROSS TIE CONTACTOR voltage')
then elect-prob-cause = 'APU DC CROSS TIE RELAY defective'.
```

```
rule-17: if apu-sys-symptom =
   'APU ON-GRD PWR/APU ON, No.2 DC BUS LT ON'
   and 'APU DC BUS voltage' = 'yes'
   and not('BUS FAIL RELAY voltage')
then elect-prob-cause = 'APU DC CROSS TIE RELAY defective'.

/*-----------APU:  APU OPERATING GROUND PWR/APU ON BOTH
                       DC BUS LIGHTS ON------------------------*/

whenfound(apu-sys-symptom =
   'APU Operating Ground PWR/APU ON Both DC Lights ON') =
   display([nl,'
   Check for 28 vdc at the transformer rectifier relay terminal
   A3.',nl,nl]).

question('trans rect relay voltage') = [nl,'
   Is the voltage at the transformer
   rectifier relay terminal A3 = 28 vdc?'].

legalvals('trans rect relay voltage') = ['yes','no'].

whenfound('trans rect relay voltage' = 'yes') =
   display([nl,'
   If you are sure that the voltage is 28 vdc,
   replace relay 161K7, then
   check the voltage of terminal A2 of relay 161K7.',nl,nl]).

whenfound('trans rect relay voltage' = 'no') =
   display([nl,'
   If you are sure that the voltage is not 28 vdc,
   fault isolate wiring, transformer rectifier, and
   input wiring.',nl,nl-).

question('A2 voltage') = [nl,'
   Is the voltage at terminal A2
   of relay 161K7 correct?'].

legalvals('A2 voltage') = ['yes','no'].

rule-18: if apu-sys-symptom =
   'APU Operating Ground PWR/APU ON Both DC Lights ON'
   and 'trans rect relay voltage'
   and 'A2 voltage'
then elect-prob-cause =
    'Relay 161K7 in APU transformer rectifier defective'.

rule-19: if apu-sys-symptom =
   'APU Operating Ground PWR/APU ON Both DC Lights ON'
   and not('trans rect relay voltage')
then elect-prob-cause = 'APU transformer rectifier defective'.

/***************  END OF APU ELECTRICAL MODULE  *****************/
```

```
/****************  UTILITY POWER SYSTEM ******************/

goal = hydr-prob-cause.

automaticmenu(ALL).
enumeratedanswers(ALL).

presupposition(util-pwr-sys-symptom) =
   major-system = 'hydraulic'
   and hydraulic-sub-system = 'Utility Power System'.

presupposition(util-press-ind-symptom) =
   major-system = 'hydraulic'
   and hydraulic-sub-system =  'utility pressure indicating
system'.

question(util-pwr-sys-symptom) = ['
  What are the symptoms of the problem?',nl].

legalvals(util-pwr-sys-symptom) = [
  'Handpump Operation Difficult and Ineffective (Filling System)',
  'Handpump Operation Difficult and Ineffective (Pressurizing System)',
  'Pressure Fluctuates Between 0 and 2500 psi After APU Starts',
  'Accumulator Pressure Drops Off to Gas Precharge',
  'APU Starts But Does Not Accelerate -Runs 40-60% RPM',
  'Continuous UTIL HYD HOT Warning Light',
  'Fan Does Not Operate',
  'Fluid Overheating',
  'Accumulator Gas Precharge Bleeds Off',
  'Low System Pressure -- Transmission Turning',
  'Power Supply Pressure Fluctuates',
  'APU Will Not Start',
  'Subsystems Not Receiving Pressure',
  'Unable to Pressurize Accumulator With Handpump',
  'Repeated Extension of Filter Contamination Indicators',
  'Hydraulic Chattering or High Frequency Vibrations in System'-.

/*-----------UTILITY POWER SYSTEM SYMPTOM:
     HANDPUMP OPERATION DIFFICULT AND INEFFECTIVE (FILLING
     SYSTEM)                                              ------ */

whenfound(util-pwr-sys-symptom =
'Handpump Operation Difficult and Ineffective (Filling System)') =
  display(['
  The first check of this problem is:

  Check that the arrow on the check valve points toward the tee
  above the depressurizing valve.',nl,nl,nl]).

question(hand-pump-ck-valve) = [nl,'
Which way is the arrow on the check
valve pointing:

"towards tee" or "away from tee"

above the depressurizing valve?'].

legalvals(hand-pump-ck-valve) = ['towards tee','away from tee'].

rule-1: if util-pwr-sys-symptom =
  'Handpump Operation Difficult and Ineffective (Filling System)'
  and hand-pump-ck-valve = 'away from tee'
then hydr-prob-cause =
  'Check valve installed backwards in tee above
   depressurizing valve'.

whenfound(hand-pump-ck-valve = 'towards tee') =
  display([nl,nl,'
  If you are sure that the check valve is installed properly in
  the tee above the depressurizing valve, then:

  check which way the arrow points on the check valve in the
  bottom filler.',nl,nl,nl]).
```

```
rule-2: if util-pwr-sys-symptom =
   'Handpump Operation Difficult and Ineffective (Filling System)'
   and hand-pump-ck-valve = 'towards tee'
   and bottom-filler-ck-valve = 'towards filter'
then hydr-prob-cause =
   'Check valve installed backwards in bottom of filler'.

question(bottom-filler-ck-valve) = ['
Which way is the arrow on the bottom
filler check valve pointing:

"towards filter" or "away from filter"?'].

legalvals(bottom-filler-ck-valve) = ['towards filter','away from filter'].

whenfound(bottom-filler-ck-valve = 'away from filter') =
   display([nl,nl,'
   If you are sure that the check valve is installed properly in
   the tee above the depressurizing valve, and
   If you are sure that the check valve is installed properly in
   the bottom filler, then:

   check which way that the arrow points on the check
   valve in the out port of the return filter.',nl,nl,nl]).

rule-3: if util-pwr-sys-symptom =
   'Handpump Operation Difficult and Ineffective (Filling System)'
   and hand-pump-ck-valve = 'towards tee'
   and bottom-filler-ck-valve = 'away from filter'
   and return-filter-ck-valve = 'towards filter'
then hydr-prob-cause =
   'Check valve installed backwards in Out Port of Return Filter'.

question(return-filter-ck-valve) = ['
Which way is the arrow on the out port
of return filter check valve pointing:

"towards filter" or "away from filter"?'].

legalvals(return-filter-ck-valve) = ['towards filter',
'away from filter'].

whenfound(return-filter-ck-valve = 'away from filter') =
   display([nl,nl,'
   If you are sure that the check valve is installed properly in
   the tee above the depressurizing valve, and
   If you are sure that the check valve is installed properly in
   the bottom filler, and
   If you are sure that the check valve is installed properly in
   the out port of the return filter, then:

   check the Depressurizing Valve.',nl,nl,nl]).

rule-4: if util-pwr-sys-symptom =
   'Handpump Operation Difficult and Ineffective (Filling System)'
   and hand-pump-ck-valve = 'towards tee'
   and bottom-filler-ck-valve = 'away from filter'
   and return-filter-ck-valve = 'away from filter'
   and depress-valve = 'bad'
   and display([nl,nl,'
   Replace the Depressurizing Valve using
   A1-H46AE-450-000, section WP 028 00.',nl,nl])
then hydr-prob-cause = 'Faulty Depressurizing Valve'.

question(depress-valve) = ['
Is the depressurizing valve
   "good" or "bad"
or have you
   "already replaced" it?'].

legalvals(depress-valve) = ['good','bad','already replaced'].
```

```
rule-5: if util-pwr-sys-symptom =
   'Handpump Operation Difficult and Ineffective (Filling System)'
   and hand-pump-ck-valve = 'towards tee'
   and bottom-filler-ck-valve = 'away from filter'
   and return-filter-ck-valve = 'away from filter'
   and depress-valve = 'good'
   or  depress-valve = 'already replaced'
   and display([nl,nl,'
   If you are sure that the Depressurizing Valve is "good", or
   if replacing the Depressurizing Valve did not correct the
   problem, replace the Handpump using A1-H46AE-450-000,
   section WP 029 00.',nl,nl])
then hydr-prob-cause = 'Faulty Handpump'.

/*--------------UTILITY POWER SYSTEM SYMPTOMS:
      HANDPUMP OPERATION DIFFICULT AND INEFFECTIVE (PRESSURIZING
      SYSTEM)                                          -----------*/


whenfound(util-pwr-sys-symptom =
'Handpump Operation Difficult and Ineffective (Pressurizing System)') =
   display(['
   The first check of this problem is:

   Check the check valves in the tee in the reservoir suction line
   at station 533, WL 44.
   Check the arrow at the side of the tee, and
   check the arrow on the valve at the bottom of the tee.',nl,nl,nl]).

question(res-suct-line-ck-valve) = [nl,'
Which way are the arrows on the
reservoir suction line check
valves pointing?'].

legalvals(res-suct-line-ck-valve) = [
'side tee arrow - away from tee, and bottom tee arrow - towards tee',
'side tee arrow - towards tee, and bottom tee arrow - away from tee'].

rule-6: if util-pwr-sys-symptom =
   'Handpump Operation Difficult and Ineffective (Pressurizing System)'
   and res-suct-line-ck-valve =
   'side tee arrow - towards tee, and bottom tee arrow - away from tee'
   or  res-suct-line-ck-valve = unknown
then hydr-prob-cause =
   'Check valves are installed backwards in tee in reservoir
   suction line at station 533, WL 44'.

whenfound(res-suct-line-ck-valve =
'side tee arrow - away from tee, and bottom tee arrow - towards tee') =
   display(['
   If you are sure that the arrow at the side of the tee points
   AWAY from the tee, and that the arrow on the valve at the
   bottom of the tee points TOWARDS the tee, then:

   Check for Air in the Utility Reservoir.',nl,nl,nl]).

question(air-in-util-res) = [nl,'
Did you find air in the utility
reservoir?'].

legalvals(air-in-util-res) = ['yes','no'].

rule-7: if util-pwr-sys-symptom =
   'Handpump Operation Difficult and Ineffective (Pressurizing System)'
   and res-suct-line-ck-valve =
   'side tee arrow - away from tee, and bottom tee arrow - towards tee'
   and air-in-util-res = 'yes'
   and display([nl,nl,'
   Bleed the Utility Reservoir using A1-H46AE-450-000,
   section WP 013 00.',nl,nl,nl])
then hydr-prob-cause = 'Air in Utility Reservoir'.
```

```
whenfound(air-in-util-res = 'no') =
   display(['
   If you are sure that the check valves are properly installed in
   the tee at the reservoir suction line, and
   that there is no air in the utility reservoir, then:

      1.  Disconnect the line between the isolation manifold and the
           package manifold.
      2.  Cap the isolation manifold and plug line.
      3.  Operate the handpump.',nl]).


rule-8: if util-pwr-sys-symptom =
   'Handpump Operation Difficult and Ineffective (Pressurizing System)'
   and res-suct-line-ck-valve =
   'side tee arrow - away from tee, and bottom tee arrow - towards tee'
   and air-in-util-res = 'no'
   and hand-pump-ops = 'yes'
   and display([nl,nl,'
   If the hand pump operates properly, the package manifold is
   faulty.',nl,nl,nl])
then hydr-prob-cause = 'Faulty Relief Valve in Package Manifold'.

question(hand-pump-ops) = [nl,'
Does the hand pump operate properly?'].

legalvals(hand-pump-ops) = ['yes','no'].

whenfound(hand-pump-ops = 'no') =
   display([nl,nl,nl,'
   If you are sure that the check valves are properly installed in
   the tee at the reservoir suction line, and
   that there is no air in the utility reservoir, and
   that the hand pump does not operate properly, then:

   Check the Depressurizing Valve, and
   Check the reservoir for external damage.',nl,nl,nl]).

rule-9: if util-pwr-sys-symptom =
   'Handpump Operation Difficult and Ineffective (Pressurizing System)'
   and res-suct-line-ck-valve =
   'side tee arrow - away from tee, and bottom tee arrow - towards tee'
   and air-in-util-res = 'no'
   and hand-pump-ops = 'no'
   and depress-valve = 'bad'
   and res-ext-damage = 'no'
then hydr-prob-cause = 'Faulty Depressurizing Valve'.


rule-10: if util-pwr-sys-symptom =
   'Handpump Operation Difficult and Ineffective (Pressurizing System)'
   and res-suct-line-ck-valve =
   'side tee arrow - away from tee, and bottom tee arrow - towards tee'
   and air-in-util-res = 'no'
   and hand-pump-ops = 'no'
   and depress-valve = 'good'
   or  depress-valve = 'already replaced'
   and res-ext-damage = 'no'
   and display([nl,nl,'
   If you are sure that the Depressurizing Valve is "good", or
   if replacing the Depressurizing Valve did not correct the
   problem, replace the Handpump using A1-H46AE-450-000,
   section WP 029 00.',nl,nl])
then hydr-prob-cause = 'Faulty Handpump'.

question(res-ext-damage) = ['
Is there any evidence of damage
on the reservoir?'].

legalvals(res-ext-damage) = ['yes','no'].
```

```
rule-11: if util-pwr-sys-symptom =
   'Handpump Operation Difficult and Ineffective (Pressurizing System)'
   and res-suct-line-ck-valve =
   'side tee arrow - away from tee, and bottom tee arrow - towards tee'
   and air-in-util-res = 'no'
   and hand-pump-ops = 'no'
   and depress-valve = 'good'
   or  depress-valve = 'already replaced'
   and res-ext-damage = 'yes'
   and display([nl,nl,'
   Replace the Reservoir using A1-H46AE-450-000,
   section WP 017 00.',nl,nl])
then hydr-prob-cause =
   'Handpump Cavitation caused by Binding Piston in the Utility Reservoir'.

/*----------UTILITY POWER SYSTEM SYMPTOMS:
        PRESSURE FLUCTUATES BETWEEN 0 AND 2500 PSI AFTER APU
        STARTS                                             -------*/

whenfound(util-pwr-sys-symptom =
   'Pressure Fluctuates Between 0 and 2500 psi After APU Starts') =
   display([nl,'
   The first check for this symptom is:

   Check the contamination indicator on system filters.
      If extended:
         Examine contents of filter bowl for contamination caused
         by pump-motor failure.',nl,nl,nl]).

question(filter-contam) = ['
Is the contamination indicator on the
system filters extended AND
is there evidence of contamination
in the filter bowl?'].

legalvals(filter-contam) = ['yes','no'].

rule-12: if util-pwr-sys-symptom =
   'Pressure Fluctuates Between 0 and 2500 psi After APU Starts'
   and filter-contam = 'yes'
   and display([nl,nl,'
   Replace APU Pump-Motor, and
   Flush the lines using A1-H46AE-450-000,
   section WP 019 00.',nl,nl])
then hydr-prob-cause = 'Faulty APU Pump-Motor'.

whenfound(filter-contam = 'no') =
   display([nl,nl,'
   If you are troubleshooting:
      "Pressure Fluctuates Between 0 and 2500 psi After APU Starts"

   If you are sure that the system is not contaminated,
   then the next troubleshooting procedure you perform is:

   Pressurize the system with a teststand.
   Note whether the system holds pressure or not.
   -------------------------------------------------

   If you are troubleshooting:
      "APU Starts But Does Not Accelerate (Runs At About 40-60% RPM)"

   If you are sure that the main engine is NOT cranking
      during APU start, and the Accumulator air charge and
      hydraulic pressure are good, and the APU pump-motor is not
         cavitating, and
      the APU pump-motor is not contaminated,
   then the next check for this symptom is:

   1.  Connect a 3000 psi gauge to the case drain port.
   2.  Use a tee.
   3.  During motor mode, pressure should not exceed 250 psi.
   4.  During pumping mode, pressure should not exceed 150 psi.',nl,nl]).

question(util-sys-press) = ['
Does the Utility System hold pressure?'].
```

```
legalvals(util-sys-press) = ['yes','no'].

rule-13: if util-pwr-sys-symptom =
   'Pressure Fluctuates Between 0 and 2500 psi After APU Starts'
   and filter-contam = 'no'
   and util-sys-press = 'no'
   and display([nl,nl,'
   Since the system did not hold pressure, you must now:

   troubleshoot the flight control hydraulic system using
   A1-H46AE-450-000, section WP 010 00.',nl,nl,nl])
then hydr-prob-cause = 'Pressure Leak in No.2 Flight Control System'.

whenfound(util-sys-press = 'yes') =
   display([nl,nl,'
   If you are sure that:
      the system is not contaminated, and
      the system does hold pressure,
   then the next troubleshooting procedure you perform is:

   1.  Pressurize the accumulator with the handpump.
   2.  Check if the system maintains pressure, or
             if the system drops off to air precharge pressure.',nl,nl]).

question(accum-press) = ['
Does the Accumulator "hold" pressure,
or does it "drop off" to air
precharge pressure?'].

legalvals(accum-press) = ['hold','drop off'].

rule-14: if util-pwr-sys-symptom =
   'Pressure Fluctuates Between 0 and 2500 psi After APU Starts'
   and filter-contam = 'no'
   and util-sys-press = 'yes'
   and accum-press = 'drop off'
   and display([nl,nl,'
   If you are sure that the Accumulator did not hold pressure -
     there was a pressure "drop off", you must now:

   Restart the consultation to troubleshoot as directed in the
   symptom:

      "Accumulator Pressure Drops Off To Gas Precharge.',nl,nl,nl])
then hydr-prob-cause = 'Leakage through a Utility System Component'.

whenfound(accum-press = 'hold') =
   display([nl,nl,'
   If you are troubleshooting:
    "Pressure Fluctuates Between 0 and 2500 psi After APU Starts"
   and if you are sure that:
      the system is not contaminated, and
      the system does hold pressure, and
      the accumulator does hold pressure,
   then the next troubleshooting procedure you perform is:
      1.  Disconnect the line from the relief valve.
      2.  Cap off the valve and plug line.
      3.  Operate the APU.
   Check if the Utility System pressure fluctuates, or not.

    **  DISREGARD THIS PROCEDURE IF TROUBLESHOOTING  **
     "Accumulator Pressure Drops Off To Gas Precharge.',nl,nl]).

question(util-sys-press-fluct) = [' Does the
             Utility System Pressure fluctuate?'].

legalvals(util-sys-press-fluct) = ['yes','no'].

rule-15: if util-pwr-sys-symptom =
   'Pressure Fluctuates Between 0 and 2500 psi After APU Starts'
   and filter-contam = 'no'
   and util-sys-press = 'yes'
   and accum-press = 'hold'
   and util-sys-press-fluct = 'no'
then hydr-prob-cause = 'Faulty High Pressure Relief Valve'.
```

```
whenfound(util-sys-press-fluct = 'yes') =
   display([nl,'
   If you are sure that:
       the system is not contaminated, and
       the system does hold pressure, and
       the accumulator does hold pressure, and
       the Utility System pressure is fluctuating when you operated
       the APU,
   then the next troubleshooting procedure you perform is:
      1.  Disconnect the return line from the package manifold.
      2.  Cap off the manifold plug line.
      3.  Operate the APU.
   Check if the Utility System pressure fluctuates, or not.',nl,nl]).

question(second-ck-util-sys-press-fluct) = ['
Does the Utility System Pressure
fluctuate?'].

legalvals(second-ck-util-sys-press-fluct) = ['yes','no'].

rule-16: if util-pwr-sys-symptom =
   'Pressure Fluctuates Between 0 and 2500 psi After APU Starts'
   and filter-contam = 'no'
   and util-sys-press = 'yes'
   and accum-press = 'hold'
   and util-sys-press-fluct = 'yes'
   and second-ck-util-sys-press-fluct = 'no'
then hydr-prob-cause = 'Faulty Relief Valve in Package Manifold'.

rule-17: if util-pwr-sys-symptom =
   'Pressure Fluctuates Between 0 and 2500 psi After APU Starts'
   and filter-contam = 'no'
   and util-sys-press = 'yes'
   and accum-press = 'hold'
   and util-sys-press-fluct = 'yes'
   and second-ck-util-sys-press-fluct = 'yes'
   and display([nl,nl,'
   Sorry, this situation is beyond the scope of this program.',nl,nl,nl])
then hydr-prob-cause = 'UNKNOWN - NOT PROGRAMMED YET'.


/*----------UTILITY POWER SYSTEM SYMPTOMS:
      ACCUMULATOR PRESSURE DROPS OFF TO GAS PRECHARGE ------*/


whenfound(util-pwr-sys-symptom =
   'Accumulator Pressure Drops Off to Gas Precharge') =
   display([nl,'
   The first check for this symptom is:

   Check the pressure-operated valve for leakage.',nl,nl,nl]).

question(press-op-valve-leak) = ['
Is the pressure-operated valve leaking?'].

legalvals(press-op-valve-leak) = ['yes','no'].

rule-18: if util-pwr-sys-symptom =
   'Accumulator Pressure Drops Off to Gas Precharge'
   and press-op-valve-leak = 'yes'
   and display([nl,nl,'
   Replace Pressure-Operated Valve using A1-H46AE-450-000,
   section WP 026 00.',nl,nl])
then hydr-prob-cause = 'Faulty Pressure-Operated Valve'.

whenfound(press-op-valve-leak = 'no') =
   display([nl,'
   If you are sure that the Pressure-Operated Valve is NOT
   leaking, then the next check for this symptom is:

   1.  Disconnect APU pressure line quick disconnects.
   2.  Pressurize the accumulator.
   3.  Check for accumulator pressure drop.',nl,nl,nl]).
```

```
rule-19: if util-pwr-sys-symptom =
   'Accumulator Pressure Drops Off to Gas Precharge'
   and press-op-valve-leak = 'no'
   and accum-press = 'hold'
   and display([nl,nl,'
   Replace Pressure-Operated Valve using A1-H46AE-450-000,
   section WP 021 00.',nl,nl])
then hydr-prob-cause = 'Faulty Pressure-Operated Valve'.


whenfound(accum-press = 'drop off') =
   display([nl,nl,'
   If you are troubleshooting:
     "Accumulator Pressure Drops Off to Gas Precharge"
   and if you are sure that:
      the Pressure-Operated Valve does not leak, and
      that the accumulator pressure does "drop off",
   the next troubleshooting procedure is:

      1.  Remove the line to the manifold valve AUX PRESS port.
      2.  Cap the tee.
      3.  Pressurize the accumulator.
      4.  Check for accumulator pressure "drop off" or "hold".',nl,nl]).

rule-20: if util-pwr-sys-symptom =
   'Accumulator Pressure Drops Off to Gas Precharge'
   and press-op-valve-leak = 'no'
   and accum-press = 'drop off'
   and ramp-manifold-accum-press-ck = 'hold'
   and display([nl,nl,'
   Replace Ramp and Door Control Manifold Valve using A1-H46AE-110-000,
   section WP 015 00.',nl,nl])
then hydr-prob-cause = 'Faulty Ramp and Door Control Manifold Valve'.

question(ramp-manifold-accum-press-ck) = ['
Does the accumulator "hold" or "drop off"
pressure?'].

legalvals(ramp-manifold-accum-press-ck) = ['hold','drop off'].

whenfound(ramp-manifold-accum-press-ck = 'drop off') =
   display([nl,nl,'
   If you are sure that:
      the the Pressure-Operated Valve is not leaking, and
      the accumulator does hold pressure for the first check, and
      the accumulator "drops off" pressure in the second check,
   then the next troubleshooting procedure you perform is:

   Check for APU Pump-Motor motoring with start switch OFF.',nl,nl]).

rule-21: if util-pwr-sys-symptom =
   'Accumulator Pressure Drops Off to Gas Precharge'
   and press-op-valve-leak = 'no'
   and accum-press = 'drop off'
   and ramp-manifold-accum-press-ck = 'drop off'
   and apu-motoring = 'yes'
   and display([nl,nl,'
   Replace Solenoid Operated Shutoff Valve using A1-H46AE-450-000,
   section WP 021 00.',nl,nl])
then hydr-prob-cause = 'Faulty Solenoid Operated Shutoff (Pilot) Valve'.

question(apu-motoring) = ['
Does the APU motor with the start
switch OFF?'].

legalvals(apu-motoring) = ['yes','no'].
```

```
whenfound(apu-motoring = 'no') =
   display([nl,nl,'
   If you are sure that:
       the the Pressure-Operated Valve is not leaking, and
       the accumulator does hold pressure for the first check, and
       the accumulator "drops off" pressure in the second check,
       and the APU does not motor when the start switch is OFF,
   then the next troubleshooting procedure you perform is:

   Check for the pressure to drop off when depressurizing valve
   is set to FULL.',nl,nl]).

rule-22: if util-pwr-sys-symptom =
   'Accumulator Pressure Drops Off to Gas Precharge'
   and press-op-valve-leak = 'no'
   and accum-press = 'drop off'
   and ramp-manifold-accum-press-ck = 'drop off'
   and apu-motoring = 'no'
   and depress-valve-press-drop = 'drop off'
   and display([nl,nl,'
   Replace Controllable Check Valve using A1-H46AE-450-000,
   section WP 022 00.',nl,nl])
then hydr-prob-cause = 'Faulty Controllable Check Valve'.

question(depress-valve-press-drop) = ['
Does the pressure "hold" or "drop off"
when the depressurizing valve is set
to FULL?'].

legalvals(depress-valve-press-drop) = ['hold','drop off'].

whenfound(depress-valve-press-drop = 'hold') =
   display([nl,'
   If you are sure that:
       the the Pressure-Operated Valve is not leaking, and
       the accumulator does hold pressure for the first check, and
       the accumulator "drops off" pressure in the second check,
       and the APU does not motor when the start switch is OFF, and
       the pressure "holds" when the depressurizing valve is set
       to FULL,
   then the next troubleshooting procedure you perform is:
         1.  Disconnect line from shuttle valve to top cross between
              high pressure relief valve and pressure line filter.
         2.  Cap the valve.
         3.  Plug line.
         4.  Pressurize the Accumulator.
   Check for Accumulator pressure drop.',nl,nl]).

rule-23: if util-pwr-sys-symptom =
   'Accumulator Pressure Drops Off to Gas Precharge'
   and press-op-valve-leak = 'no'
   and accum-press = 'drop off'
   and ramp-manifold-accum-press-ck = 'drop off'
   and apu-motoring = 'no'
   and depress-valve-press-drop = 'hold'
   and shuttle-valve-accum-press-ck = 'drop off'
   and display([nl,nl,'
   Replace Shuttle Valve using A1-H46AE-450-000,
   section WP 018 00.',nl,nl])
then hydr-prob-cause = 'Faulty Controllable Check Valve'.

question(shuttle-valve-accum-press-ck) = ['
Does the accumulator pressure
"hold" or "drop off"?'].

legalvals(shuttle-valve-accum-press-ck) = ['hold','drop off'].


/*-----------UTILITY POWER SYSTEM SYMPTOM:
     APU STARTS BUT DOES NOT ACCELERATE (RUNS AT ABOUT
          40 TO 60% RPM                                   ---- */

whenfound(util-pwr-sys-symptom =
```

```
   'APU Starts But Does Not Accelerate -Runs 40-60% RPM') =
   display([nl,'
   The first check for this symptom is:

   Watch the Ng Tachometer for engine cranking during APU start.
   If trouble is on #1 Engine, set SYS SELECT switch to ENG #1.
   Check to see if the main engine cranks.',nl,nl,nl]).

question(engine-cranking) = ['
Is the Engine cranking during
APU start?'].

legalvals(engine-cranking) = ['yes','no'].

rule-32: if util-pwr-sys-symptom =
   'APU Starts But Does Not Accelerate -Runs 40-60% RPM'
   and engine-cranking = 'yes'
   and display([nl,nl,'
   Replace the main engine start valve using A1-H46AE-220-000,
   section WP 075 00.',nl,nl])
then hydr-prob-cause = 'Faulty Main Engine Start Valve'.

whenfound(engine-cranking = 'no') =
   display([nl,'
   If you are sure that the main engine is NOT cranking during APU
   start, the next check for this symptom is:

   Check the Accumulator air charge and hydraulic
   pressure.',nl,nl]).

question(accum-air-chrg-hydr-pres) = ['
Is the Accumulator Air charge or the
Hydraulic Pressure Low?'].

legalvals(accum-air-chrg-hydr-pres) = ['yes','no'].

rule-33: if util-pwr-sys-symptom =
   'APU Starts But Does Not Accelerate -Runs 40-60% RPM'
   and engine-cranking = 'no'
   and accum-air-chrg-hydr-pres = 'yes'
   and display([nl,nl,'
   Dump the Hydraulic Pressure.
   Check the gauge for proper air charge.
   Recharge as necessary.
   Handpump the accumulator to 3000 psi.  Use A1-H46AE-450-000,
   section WP 013 00.',nl,nl])
then hydr-prob-cause =
   'Accumulator Air Charge or Hydraulic Pressure is low'.

whenfound(accum-air-chrg-hydr-pres = 'no') =
   display([nl,'
   If you are sure that the main engine is NOT cranking during APU
   start, and
   the Accumulator air charge and hydraulic pressure are good,
   then the next check for this symptom is:

   Check for the APU pump-motor cavitating and chattering during
   APU start.',nl,nl]).

question(apu-cavitating) = ['
Is the APU cavitating and chattering
during start?'].

legalvals(apu-cavitating) = ['yes','no'].

rule-34: if util-pwr-sys-symptom =
   'APU Starts But Does Not Accelerate -Runs 40-60% RPM'
   and engine-cranking = 'no'
   and accum-air-chrg-hydr-pres = 'no'
   and apu-cavitating = 'yes'
   and display([nl,nl,'
   Bleed air from system using A1-H46AE-450-000,
   section WP 013 00.',nl,nl])
then hydr-prob-cause = 'Air in System'.

whenfound(apu-cavitating = 'no') =
   display([nl,'
```

```
   If you are sure that the main engine is NOT cranking during APU
   start, and
   the Accumulator air charge and hydraulic pressure are good, and
   the APU pump-motor is not cavitating,
   then the next check for this symptom is:

   Inspect the contamination indicators on system filters.
   If extended, examine contents of filter bowl for contamination
   caused by APU pump-motor failure.',nl,nl]).

rule-35: if util-pwr-sys-symptom =
   'APU Starts But Does Not Accelerate -Runs 40-60% RPM'
   and engine-cranking = 'no'
   and accum-air-chrg-hydr-pres = 'no'
   and apu-cavitating = 'no'
   and filter-contam = 'yes'
   and display([nl,nl,'
   Replace APU pump-motor.  Flush lines using A1-H46AE-450-000,
   section WP 013 00.',nl,nl])
then hydr-prob-cause = 'Faulty APU pump-motor'.

rule-36: if util-pwr-sys-symptom =
   'APU Starts But Does Not Accelerate -Runs 40-60% RPM'
   and engine-cranking = 'no'
   and accum-air-chrg-hydr-pres = 'no'
   and apu-cavitating = 'no'
   and filter-contam = 'no'
   and pump-motor-press = 'yes'
   and display([nl,nl,'
   Replace APU pump-motor.  Flush lines using A1-H46AE-450-000,
   section WP 013 00.',nl,nl])
then hydr-prob-cause = 'Faulty APU pump-motor'.

question(pump-motor-press) = ['
Are pressures excessive in either the
motor mode, or pumping mode?'].

legalvals(pump-motor-press) = ['yes','no'].

whenfound(pump-motor-press = 'no') =
   display([nl,'
   If you are sure that the main engine is NOT cranking during APU
   start, and
   the Accumulator air charge and hydraulic pressure are good, and
   the APU pump-motor is not cavitating,
   the APU is not contaminated, and
   the pump mode and motor mode pressures are not excessive,
   then:
      Troubleshooting the APU using A1-H46AE-400-000, section WP
      033 00.

   After you have done that the next check for the Utility Power
   System is:
      1.  Disconnect the line from the aft end of High Pressure
           Relief Valve.
      2.  Plug line.
      3.  Cap the valve fitting.
      4.  Operate the APU.
      5.  Check for APU acceleration.',nl,nl]).

question(apu-accel) = ['
Does the APU accelerate?'].

legalvals(apu-accel) = ['yes','no'].

rule-37: if util-pwr-sys-symptom =
   'APU Starts But Does Not Accelerate -Runs 40-60% RPM'
   and engine-cranking = 'no'
   and accum-air-chrg-hydr-pres = 'no'
   and apu-cavitating = 'no'
   and filter-contam = 'no'
   and pump-motor-press = 'no'
   and apu-accel = 'yes'
   and display([nl,nl,'
   Replace High Pressure Relif Valve using A1-H46AE-450-000,
```

```
   section WP 031 00.',nl,nl])
then hydr-prob-cause = 'High Pressure Relief Valve Leaking'.

whenfound(apu-accel = 'no') =
   display([nl,'
   If you are sure that the main engine is NOT cranking during APU
   start, and
   the Accumulator air charge and hydraulic pressure are good, and
   the APU pump-motor is not cavitating,
   the APU is not contaminated, and
   the pump mode and motor mode pressures are not excessive, and
   you have troubleshot the APU, and
   the APU does not accelerate, then:

      1.  Disconnect the line from the aft end of the manifold.
      2.  Plug line.
      3.  Cap the manifold port fitting.
      3.  Operate the APU.
      4.  Check for APU acceleration.',nl,nl]).

rule-38: if util-pwr-sys-symptom =
   'APU Starts But Does Not Accelerate -Runs 40-60% RPM'
   and engine-cranking = 'no'
   and accum-air-chrg-hydr-pres = 'no'
   and apu-cavitating = 'no'
   and filter-contam = 'no'
   and pump-motor-press = 'no'
   and apu-accel = 'no'
   and man-apu-accel = 'yes'
   and display([nl,nl,'
   Replace Manifold using A1-H46AE-450-000,
   section WP 030 00.',nl,nl])
then hydr-prob-cause = 'Utility System Package Manifold Leaking'.

question(man-apu-accel) = ['
Does the APU accelerate?'].

legalvals(man-apu-accel) = ['yes','no'].

rule-39: if util-pwr-sys-symptom =
   'APU Starts But Does Not Accelerate -Runs 40-60% RPM'
   and engine-cranking = 'no'
   and accum-air-chrg-hydr-pres = 'no'
   and apu-cavitating = 'no'
   and filter-contam = 'no'
   and pump-motor-press = 'no'
   and apu-accel = 'no'
   and man-apu-accel = 'no'
   and display([nl,nl,'
   Sorry, this situation is beyond the scope of this program.',nl,nl-)
then hydr-prob-cause = 'UNKNOWN - NOT PROGRAMMED YET'.


/*-----------UTILITY POWER SYSTEM SYMPTOM:
        CONTINUOUS UTIL HYD HOT WARNING LIGHT              ---- */


whenfound(util-pwr-sys-symptom =
   'Continuous UTIL HYD HOT Warning Light') =
   display([nl,'
   The first check for this symptom is:

Perform functional test of thermal switch, using
   A1-H46AE-450-000, section WP 024 00, para. 10.',nl,nl,nl]).

question(therm-switch) = ['
Is the Thermal Switch operating
properly?'].

legalvals(therm-switch) = ['yes','no'].

rule-40: if util-pwr-sys-symptom =
 'Continuous UTIL HYD HOT Warning Light'
   and therm-switch = 'no'
then hydr-prob-cause = 'Faulty Thermal Switch'.
```

```
whenfound(therm-switch = 'yes') =
   display([nl,'
If you are sure that the Thermal Switch is operating properly, then
the next check for this symptom is:
Abort this consultation (Ctrl-A and "yes"), and start again
 selecting the trouble symptom:
"Fluid Overheating" and troubleshoot as directed.',nl,nl]).
rule-41: if util-pwr-sys-symptom =
 'Continuous UTIL HYD HOT Warning Light'
 and therm-switch = 'yes'
then hydr-prob-cause = 'Fluid overheating'.

/*-----------UTILITY POWER SYSTEM SYMPTOM:
           FAN DOES NOT OPERATE                        ---- */

whenfound(util-pwr-sys-symptom =
   'Fan Does Not Operate') =
   display([nl,'
   The first check for this symptom is:
   Check UT HYD SYS BLO, circuit breakers for 115 vac on
   pins A, B, and C of 085P2.',nl,nl,nl]).
question(blower-cb) = ['
Is there 115 vac on pins A,B,C
of the Utility Hydraulic System
Blower?'].
legalvals(blower-cb) = ['C/Bs are out','Yes, but no fan',
'No voltage present'].
rule-42: if util-pwr-sys-symptom =
   'Fan Does Not Operate'
   and blower-cb = 'C/Bs are out'
   and display([nl,'
   Reset the UT HYD SYS BLO circuit breakers.',nl,nl])
then hydr-prob-cause = 'No AC electrical power - Circuit Breakers Out'.
rule-43: if util-pwr-sys-symptom =
   'Fan Does Not Operate'
   and blower-cb = 'No voltage present'
   and display([nl,'
   Repair or replace the wiring using
   A1-H46AE-420-000, section WP 009 00.',nl,nl])
then hydr-prob-cause = 'Faulty Wiring'.
rule-44: if util-pwr-sys-symptom =
   'Fan Does Not Operate'
   and blower-cb = 'Yes, but no fan'
   and display([nl,'
   Replace the fan using
   A1-H46AE-420-000, section WP 024 00.',nl,nl])
then hydr-prob-cause = 'Faulty Fan'.


/*-----------UTILITY POWER SYSTEM SYMPTOM:
            FLUID OVERHEATING                        ---- */

whenfound(util-pwr-sys-symptom =
   'Fluid Overheating') =
   display([nl,'
   The first thing to do is abort this consultation (Ctrl-A and "yes"),
   and restart the consultation choosing the symptom:
    "Fan Does Not Operate" and troubleshoot as directed.
   The next check after you have troubleshot the above symptom is:
    Inspect the cooler for obstructions.',nl,nl,nl]).
```

```
question(cooler-obstr) = ['
Are there any obstructions in the
cooler, like rags, tools, dead
buzzards?'].

legalvals(cooler-obstr) = ['yes','no'].


rule-45: if util-pwr-sys-symptom =
   'Fluid Overheating'
   and cooler-obstr = 'yes'
then hydr-prob-cause = 'Cooler Air Intake Clogged'.

rule-46: if util-pwr-sys-symptom =
   'Fluid Overheating'
   and cooler-obstr = 'no'
   and line-chatter = 'yes'
   and display([nl,nl,'
   Bleed system at the cooler bleed plug and reservoir bleed plug using
   A1-H46AE-450-000, section WP 013 00.',nl,nl])
then hydr-prob-cause = 'Air Locked in Cooler'.

question(line-chatter) = ['
Can you hear any "line chatter"
when the system is operating?'].

legalvals(line-chatter) = ['yes','no'].

whenfound(line-chatter = 'no') =
   display([nl,'
   If you are sure that the Fan is OK, and
   there are not obstructions in the Cooler Air Intake, and
   there is no line chatter, then:

   Check for temperature rise from pressure to return lines of
   suspected leaking components if internal leakage is not
   audible.  The system must be pressurized to 3000 psi for
   this check.'  ,nl,nl]).

rule-47: if util-pwr-sys-symptom =
   'Fluid Overheating'
   and cooler-obstr = 'no'
   and line-chatter = 'no'
   and internal-sys-temp = 'yes'
then hydr-prob-cause = 'Internal System Leakage'.

question(internal-sys-temp) = ['
Is there a temperature rise from the
pressure to the return lines of the
suspected leaking components?'].

legalvals(internal-sys-temp) = ['yes','no'].

whenfound(internal-sys-temp = 'no') =
   display([nl,'
   If you are sure that the Fan is OK, and
   there are not obstructions in the Cooler Air Intake, and
   there is no line chatter, and
   there is no temperature rise from the pressure to return lines, then:

   Check the reservoir piston rod for actual volume of fluid
   in the tank.',nl,nl]).

rule-48: if util-pwr-sys-symptom =
   'Fluid Overheating'
   and cooler-obstr = 'no'
   and line-chatter = 'no'
   and internal-sys-temp = 'no'
   and res-fluid-lvl = 'low'
then hydr-prob-cause = 'Low Fluid Level'.

question(res-fluid-lvl) = ['
What is the fluid level in the
reservoir?'].

legalvals(res-fluid-lvl) = ['normal','low'].

rule-49: if util-pwr-sys-symptom =
```

```
   'Fluid Overheating'
   and cooler-obstr = 'no'
   and line-chatter = 'no'
   and internal-sys-temp = 'no'
   and res-fluid-lvl = 'normal'
   and display([nl,nl,'
   Sorry, this situation is beyond the scope of this program.',nl,nl])
then hydr-prob-cause = 'UNKNOWN - NOT PROGRAMMED YET'.


/*-----------UTILITY POWER SYSTEM SYMPTOM:
           ACCUMULATOR GAS PRECHARGE BLEEDS OFF               ---- */

whenfound(util-pwr-sys-symptom =
   'Accumulator Gas Precharge Bleeds Off') =
   display([nl,'
   The first check of this symptom is:

   Apply a mild soap and water solution to fittings.
   Watch for bubbles indicating leakage.',nl,nl,nl]).

question(fittings-bubbles) = ['
Are there any bubbles forming
around the fittings?'].

legalvals(fittings-bubbles) = ['yes','no'].

rule-50: if util-pwr-sys-symptom =
  'Accumulator Gas Precharge Bleeds Off'
   and fittings-bubbles = 'yes'
   and display([nl,nl,'
   Lubricate all male fittings (sounds erotic, huh?).  Use antiseize
   compound.  Torque all fittings and line using A1-H46AE-GAI-000,
   section WP 003 00.',nl,nl])
then hydr-prob-cause = 'Leakage Through Line or Component Fittings'.

whenfound(fittings-bubbles = 'no') =
   display([nl,'
   If you are sure that no bubbles formed around any of the fittings,
   then   the next check is:

   Test the accumulator using A1-H46AE-450-000, section WP 049 00.
   ',nl,nl]).


question(accum-test) = ['
Did the Accumulator test "good"
or "bad"?'].

legalvals(accum-test) = ['good','bad'].

rule-51: if util-pwr-sys-symptom =
  'Accumulator Gas Precharge Bleeds Off'
   and fittings-bubbles = 'no'
   and accum-test = 'bad'
   and display([nl,nl,'
   Repair or Replace the Accumulator using
   A1-H46AE-450-000, section WP 023 00.',nl,nl])
then hydr-prob-cause = 'Accumulator Leaks Internally'.

rule-52: if util-pwr-sys-symptom =
  'Accumulator Gas Precharge Bleeds Off'
   and fittings-bubbles = 'no'
   and accum-test = 'good'
   and display([nl,nl,'
   Sorry, this situation is beyond the scope of this program.',nl,nl])
then hydr-prob-cause = 'UNKNOWN - NOT PROGRAMMED YET'.



/*-----------UTILITY POWER SYSTEM SYMPTOM:
           LOW SYSTEM PRESSURE -- TRANSMISSION TURNING            ---- */
```

```
whenfound(util-pwr-sys-symptom =
    'Low System Pressure -- Transmission Turning') =
   display([nl,'
   The first check of this symptom is:

   Check the valve pointer for correct position.',nl,nl,nl]).

question(valve-pointer) = ['
What is the position of the
Depressurizing valve pointer:
  "NORM" or "FILL"?'].

legalvals(valve-pointer) = ['normal','fill'].

rule-53: if util-pwr-sys-symptom =
   'Low System Pressure -- Transmission Turning'
   and valve-pointer = 'fill'
   and display([nl,nl,'
   Turn valve to "NORM".',nl,nl])
then hydr-prob-cause = 'Depressurizing Valve not Turned to NORM'.

whenfound(valve-pointer = 'normal') =
   display([nl,'
   If you are troubleshooting the symptom:
     "Low System Pressure -- Transmission Turning", then
   If you are sure that the valve pointer is in the NORM position, then
   the next check of this symptom is:

   Check system pressure at the accumulator pressure gauge or with a
   direct-reading pressure gauge - Figs. 1 and 2, A1-H46AE-450-000,
   section WP 008 00, p. 16.
   -----------------------------------------------------------

   If you are troubleshooting the symptom:
     "Unable to Pressurize Accumulator With Handpump, then
   the next check is:

     1.  Energize the isolation manifold.
     2.  Pressurize system.
     3.  Check for whether pressure can be maintained.',nl,nl,nl]).

question(sys-press) = ['
What is the direct-reading gauge
pressure?'].

legalvals(sys-press) = ['3000 psi','Below 3000 psi'].

rule-54: if util-pwr-sys-symptom =
   'Low System Pressure -- Transmission Turning'
   and valve-pointer = 'normal'
   and sys-press = '3000 psi'
   and display([nl,nl,'
   You just entered "3000 psi", which means that the system pressure
   is good, but the indicator is not indicating the pressure properly.
   Troubleshoot the pressure indicating system.',nl,nl])
then hydr-prob-cause = 'Faulty Pressure Transmitter or Indicator'.

whenfound(sys-press = 'Below 3000 psi') =
display([nl,'
   If you are sure that the valve pointer is in the NORM position, and
   the  direct-reading pressure on the Accumulator is above or below
   3000 psi,
   then the next check of this symptom is:

   Inspect the pump for blackening of metal plates, discoloration
   of paint, or melting of cadmium plating.',nl,nl,nl-).

question(pump) = ['
Are any of the above symptoms
present on the pump?'].

legalvals(pump) = ['yes','no'].

rule-55: if util-pwr-sys-symptom =
   'Low System Pressure -- Transmission Turning'
   and valve-pointer = 'normal'
```

```
   and sys-press = 'Below 3000 psi'
   and pump = 'yes'
   and display([nl,nl,'
   Overheating is evident, replace pump.',nl,nl])
then hydr-prob-cause = 'Low Pump Output Pressure'.

whenfound(pump = 'no') =
display([nl,'
   If you are sure that the valve pointer is in the NORM position, and
   the  direct-reading pressure on the Accumulator is above or below
   3000 psi, and
   there is no evidence of the pump overheating,
   then the next check of this symptom is:

   Pressurize the system to 3000 psi, using a teststand.',nl,nl,nl]).

question(teststand-press) = ['
Is 3000 psi maintained, or does the
pressure bleed off?'].

legalvals(teststand-press) = ['3000 psi Maintained','Bleed Off'].

rule-56: if util-pwr-sys-symptom =
   'Low System Pressure -- Transmission Turning'
   and valve-pointer = 'normal'
   and sys-press = 'Below 3000 psi'
   and pump = 'no'
   and teststand-press = '3000 psi Maintained'
   and display([nl,nl,'
   If the pressure is maintained at 3000 psi, replace the pump.',nl,nl])
then hydr-prob-cause = 'Low Pump Output Pressure'.

whenfound(teststand-press = 'Bleed Off') =
display([nl,'
   If you are sure that the valve pointer is in the NORM position, and
   the  direct-reading pressure on the Accumulator is above or below
   3000 psi, and
   there is no evidence of the pump overheating, and
   the teststand pressure bleeds off,
   then the next check of this symptom is:

   Check the pressure with a direct-reading gauge installed using
   Figs. 1 and 2, A1-H46AE-450-000, section WP 008 00, p. 16.',nl,nl,nl]).

question(dir-rdg-press) = ['
What does the direct-reading gauge
indicate?'].

legalvals(dir-rdg-press) = ['High Pressure','Low Pressure',
   'Normal','Fluctuating Pressure'].

rule-57: if util-pwr-sys-symptom =
   'Low System Pressure -- Transmission Turning'
   and valve-pointer = 'normal'
   and sys-press = 'Below 3000 psi'
   and pump = 'no'
   and teststand-press = 'Bleed Off'
   and dir-rdg-press = 'Low Pressure'
   and display([nl,nl,'
   If the pressure indicated is low, replace the manifold.',nl,nl])
then hydr-prob-cause = 'Faulty Manifold Pressure Relief Valve'.

rule-58: if util-pwr-sys-symptom =
   'Low System Pressure -- Transmission Turning'
   and valve-pointer = 'normal'
   and sys-press = 'Below 3000 psi'
   and pump = 'no'
   and teststand-press = 'Bleed Off'
   and dir-rdg-press = 'High Pressure'
   or  dir-rdg-press = 'Normal'
   or  dir-rdg-press = 'Fluctuating Pressure'
   and display([nl,nl,'
   If the pressure indicates High, Normal, or Fluctuating,
   then troubleshoot
   the Indicating System, or troubleshoot the symptom:
```

```
    "Power Supply Pressure Fluctuates".',nl,nl])
then hydr-prob-cause = 'CHECK ANOTHER SYMPTOM'.

/*-----------UTILITY POWER SYSTEM SYMPTOM:
        POWER SUPPLY PRESSURE FLUCTUATES                 ---- */

whenfound(util-pwr-sys-symptom =
 'Power Supply Pressure Fluctuates') =
  display([nl,'
  The first check of this symptom is:

  Abort (Crtl-A and "yes") this consultation, and re-boot choosing
  "Utility Pressure Indicating System" to
  Troubleshoot the Pressure Indicating System.

  After troubleshooting the Indicating System:
  Install a pressure gauge using Figs. 1 and 2, A1-H46AE-450-000,
  section WP 008 00, p. 16.',nl,nl,nl]).

rule-59: if util-pwr-sys-symptom =
  'Power Supply Pressure Fluctuates'
  and dir-rdg-press = 'Fluctuating Pressure'
then hydr-prob-cause = 'Faulty Pump Compensator'.

/*

if major-system = 'hydraulic'
  and hydraulic-sub-system = 'Utility Power System'
  and util-pwr-sys-symptom =
  'Power Supply Pressure Fluctuates'
  and dir-rdg-press = 'High Pressure'
  or  dir-rdg-press = 'Normal'
  and display([nl,nl,'
  If the pressure indicates High, Normal, or Fluctuating, then
  troubleshoot the Indicating System, or troubleshoot the symptom:
    "Power Supply Pressure Fluctuates".',nl,nl])
then hydr-prob-cause = 'CHECK ANOTHER SYMPTOM'.
*/

whenfound(dir-rdg-press = 'Normal') =
  display([nl,'
  If you are sure that the Pressure Indicating System is OK, and
  that the direct-reading gauge pressure is normal, then
  the next check of this symptom is:

  Check the Utility System Package Manifold Relief Valve
  Cracking Pressure.',nl,nl]).


rule-60: if util-pwr-sys-symptom =
  'Power Supply Pressure Fluctuates'
  and dir-rdg-press = 'Normal'
  and rel-valv-crkg-press = 'bad'
  and display([nl,nl,'
  Replace the Manifold.',nl,nl])
then hydr-prob-cause = 'Faulty Relief Valve'.

question(rel-valv-crkg-press) = ['
Is the Relief Valve Cracking Pressure
 "good" or "bad"?'].

legalvals(rel-valv-crkg-press) = ['good','bad'].


whenfound(rel-valv-crkg-press = 'good') =
  display([nl,'
  If you are sure that the Pressure Indicating System is OK, and
  that the direct-reading gauge pressure is normal, and
  that the Relief Valve Cracking Pressure is good, then
  the next check of this symptom is:

      1.  Disconnect the line from the aft end of the valve.
      2.  Plug the line.
      3.  Cap the valve fitting.
```

```
     4.  Operate the APU.
     5.  Check for a steady power supply pressure.',nl,nl]).
question(sup-press) = ['
Is the Power Supply Pressure Steady?'].
legalvals(sup-press) = ['yes','no'].
rule-61: if util-pwr-sys-symptom =
   'Power Supply Pressure Fluctuates'
   and dir-rdg-press = 'Normal'
   and rel-valv-crkg-press = 'good'
   and sup-press = 'yes'
   and display([nl,nl,'
   Replace the High Pressure Relief Valve.',nl,nl])
then hydr-prob-cause = 'High Pressure Relief Valve Leaking'.



/*-----------UTILITY POWER SYSTEM SYMPTOM:
                APU WILL NOT START                            ---- */

whenfound(util-pwr-sys-symptom = 'APU Will Not Start') =
   display([nl,'
   The first check of this symptom is:
   Check the Accumulator Pressure Gauge.',nl,nl,nl]).
question(accum-press-ck) = ['
Is the Accumulator Pressure 3000 psi?'].
legalvals(accum-press-ck) = ['yes','no'].
rule-62: if util-pwr-sys-symptom = 'APU Will Not Start'
   and accum-press-ck = 'no'
   and display([nl,nl,'
   Pressurize the Accumulator with the handpump.',nl,nl])
then hydr-prob-cause = 'Low Accumulator Hydraulic Pressure'.
whenfound(accum-press-ck = 'yes') =
   display([nl,'
   If you are sure that the Accumulator pressure is 3000 psi, then
   the next check of this symptom is:
   Depressurize the Accumulator.  Check the Air Charge.',nl,nl,nl]).
question(accum-gas-chg) = ['
Does the Accumulator have a proper
Air charge?'].
legalvals(accum-gas-chg) = ['yes','no'].
rule-63: if util-pwr-sys-symptom = 'APU Will Not Start'
   and accum-press-ck = 'yes'
   and accum-gas-chg = 'no'
   and display([nl,nl,'
   Recharge the Accumulator.',nl,nl])
then hydr-prob-cause = 'Low Accumulator Gas Charge'.
whenfound(accum-gas-chg = 'yes') =
   display([nl,'
   If you are troubleshooting the symptom:
      "APU Will Not Start", then
   If you are sure that the Accumulator pressure is 3000 psi, and
   that the Accumulator has a proper air charge, then
   the next check of this symptom is:
   Disconnect the plug - 136P5 from the pilot valve.
   Check for 24 vdc between pins 2 and 3.
-------------------------------------------------------------
   If you are troubleshooting the symptom:
      "Unable to Pressurize Accumulator With Handpump", then
   If you are sure that Accumulator has a proper gas charge, then
```

```
   Check the Depressurizing Valve pointer position.',nl,nl,nl]).
question(plug-voltage) = ['
Is there 24 vdc between
pins 2 and 3?'].
legalvals(plug-voltage) = ['yes','no'].
rule-64: if util-pwr-sys-symptom = 'APU Will Not Start'
   and accum-press-ck = 'yes'
   and accum-gas-chg = 'yes'
   and plug-voltage = 'no'
   and display([nl,nl,'
   Correct improper installation.
   Check again.  If voltage still not present troubleshoot
   the electrical   system using A1-H46AE-420-000, section WP 009 00.
   ',nl,nl])
then hydr-prob-cause = 'Check Valve Installed Backwards'.
whenfound(plug-voltage = 'yes') =
   display([nl,'
   If you are sure that the Accumulator pressure is 3000 psi, and
   that the Accumulator has a proper air charge, and
   that the Check Valve is installed properly, then
   the next check of this symptom is:
   If voltage is present at the Check Valve from the last check,
    1.  Disconnect the APU signal line at the APU pump-motor.
    2.  Cap the motor opening.
    3.  Install a gauge exceeding 3000 psi capacity into the line.
    4.  Operate the APU start switch.
    5.  Note the gauge pressure.',nl,nl,nl]).
question(gauge-press) = ['
Is there 3000 psi indicating on
 the gauge?'].
legalvals(gauge-press) = ['yes','no'].
rule-65: if util-pwr-sys-symptom = 'APU Will Not Start'
   and accum-press-ck = 'yes'
   and accum-gas-chg = 'yes'
   and plug-voltage = 'yes'
   and gauge-press = 'no'
   and display([nl,nl,'
   If pressure is not read, replace the valve.',nl,nl])
then hydr-prob-cause = 'APU Start Sequence Pilot Valve Not Opening'.

/*-----------UTILITY POWER SYSTEM SYMPTOM:
             SUBSYSTEMS NOT RECEIVING PRESSURE           ---- */

whenfound(util-pwr-sys-symptom = 'Subsystems Not Receiving Pressure') =
   display([nl,'
   The first check of this symptom is:
   Check the position of the Depressurizing Valve.',nl,nl,nl]).
rule-66: if util-pwr-sys-symptom = 'Subsystems Not Receiving Pressure'
   and valve-pointer = 'fill'
   and display([nl,'
   Turn the valve to NORM.',nl,nl])
then hydr-prob-cause = 'Depressurizing Valve Turned to FILL'.
whenfound(valve-pointer = 'normal') =
   display([nl,'
   If you are troubleshooting the symptom:
     "Subsystems Not Receiving Pressure", then
   If you are sure that the valve pointer is in the NORM position, then
   the next check of this symptom is:
   Manually open the No.2 valve.
   Check for pressure at the ramp control.
   -------------------------------------------------------------
   If you are troubleshooting the symptom:
```

```
      "Unable to Pressurize Accumulator With Handpump", then
      ',nl,nl]).
question(ramp-ctrl-press) = ['
Is there pressure at the ramp and hatch
controls?'].
legalvals(ramp-ctrl-press) = ['yes','no'].
rule-67: if util-pwr-sys-symptom = 'Subsystems Not Receiving Pressure'
   and valve-pointer = 'normal'
   and ramp-ctrl-press = 'no'
   and display([nl,'
   Replace the Manifold, using A1-H46AE-450-000, section WP 030 00.
   ',nl,nl])
then hydr-prob-cause =
   'Isolation Manifold No.1 Valve Not Open When De-energized'.
whenfound(ramp-ctrl-press = 'yes') =
   display([nl,'
   If you are sure that the valve pointer is in the NORM position, and
   that there is pressure at the ramp and hatch controls, then
   the next check of this symptom is:
   Check the Accumulator for gas precharge.',nl,nl]).

rule-68: if util-pwr-sys-symptom = 'Subsystems Not Receiving Pressure'
   and valve-pointer = 'normal'
   and ramp-ctrl-press = 'yes'
   and accum-gas-chg = 'no'
   and display([nl,'
   Service the Accumulator using A1-H46AE-450-000, section WP 013 00.
   ',nl,nl])
then hydr-prob-cause = 'No Accumulator Gas Precharge'.
rule-69: if util-pwr-sys-symptom = 'Subsystems Not Receiving Pressure'
   and valve-pointer = 'normal'
   and ramp-ctrl-press = 'yes'
   and accum-gas-chg = 'yes'
   and display([nl,'
   Sorry, this situation is beyond the scope of this program.',nl,nl])
then hydr-prob-cause = 'UNKNOWN - NOT PROGRAMMED YET'.

/*-----------UTILITY POWER SYSTEM SYMPTOM:
       UNABLE TO PRESSURIZE ACCUMULATOR WITH HANDPUMP     ---- */

whenfound(util-pwr-sys-symptom =
   'Unable to Pressurize Accumulator With Handpump') =
   display([nl,'
   The first check of this symptom is:
   Check the Accumulator for a Gas Precharge.',nl,nl,nl]).
rule-70: if util-pwr-sys-symptom =
   'Unable to Pressurize Accumulator With Handpump'
   and accum-gas-chg = 'no'
   and display([nl,nl,'
   Recharge the Accumulator.',nl,nl])
then hydr-prob-cause = 'No Accumulator Gas Precharge'.
rule-71: if util-pwr-sys-symptom =
   'Unable to Pressurize Accumulator With Handpump'
   and accum-gas-chg = 'yes'
   and valve-pointer = 'fill'
   and display([nl,nl,'
   Turn the Depressurizing Valve to NORM.',nl,nl])
then hydr-prob-cause = 'Depressurizing Valve at FILL'.
question(manifold-press) = ['
Is the pressure maintained?'].
legalvals(manifold-press) = ['yes','no'].
rule-72: if util-pwr-sys-symptom =
```

```
   'Unable to Pressurize Accumulator With Handpump'
   and accum-gas-chg = 'yes'
   and valve-pointer = 'normal'
   and manifold-press = 'yes'
   and display([nl,nl,'
   If pressure can be maintained, this is an indication of a
   leak downstream   form the manifold.  Find the leak.',nl,nl])
then hydr-prob-cause = 'Leak in System'.

rule-73: if util-pwr-sys-symptom =
   'Unable to Pressurize Accumulator With Handpump'
   and accum-gas-chg = 'yes'
   and valve-pointer = 'normal'
   and manifold-press = 'no'
   and display([nl,nl,'
   Sorry, this situation is beyond the scope of this program.',nl,nl])
then hydr-prob-cause = 'UNKNOWN - NOT PROGRAMMED YET'.


/*-----------UTILITY POWER SYSTEM SYMPTOM:
       REPEATED EXTENSION OF FILTER CONTAMINATION INDICATORS ---- */


whenfound(util-pwr-sys-symptom =
   'Repeated Extension of Filter Contamination Indicators') =
   display([nl,'
   Troubleshooting procedures for this symptom are:

   Inspect the filter elements for metallic matter.
   If metal is present the first time:
     change the Accumulator and flush the system.  Then,
   inspect the filter elements again.
   If metal is present a second time:
     change the Utility Pump and flush the system.  Then,
   inspect the filter elements again.
   If metal is present the third check:
     changethePump-motorand flush the APUlinesand system.',nl,nl]).

question(filter-elements) = ['
Is there any metallic matter in
the filter elements?'].

legalvals(filter-elements) = ['Yes, Metal present','No Metal'].

rule-74: if util-pwr-sys-symptom =
   'Repeated Extension of Filter Contamination Indicators'
   and filter-elements = 'Yes, Metal present'
then hydr-prob-cause =
'Faulty Accumulator, Utility Pump, and/or Pump-motor'.

rule-75: if util-pwr-sys-symptom =
   'Repeated Extension of Filter Contamination Indicators'
   and filter-elements = 'No Metal'
   and display([nl,nl,'
   If you do not have any metal present in the filter elements
   upon the   first inspection, then the problem is beyond the
   scope of this program.

   If you have already changed the Accumulator and/or the Utility Pump,
   and there is not metal present on subsequent inspections of the filter
   elements, then problem is solved.',nl,nl])
then hydr-prob-cause = 'UNKNOWN - NOT PROGRAMMED YET'.


/*-----------UTILITY POWER SYSTEM SYMPTOM:
    HYDRAULIC CHATTERING OR HIGH FREQUENCY VIBRATIONS IN SYSTEM -- */


whenfound(util-pwr-sys-symptom =
   'Hydraulic Chattering or High Frequency Vibrations in System') =
   display([nl,'
   Troubleshooting procedures for this symptom are:

   1.  Apply electrical power to the helicopter.
```

```
   2.  Set the ROTOR BRAKE switch to OFF.
   3.  Disconnect pressure line from pump and plug port.
   4.  Pressurize system.
   5.  Use a teststand.
   6.  Check for fluid flow from the pressure line.',nl,nl]).

question(press-line-fluid-flow) = ['
Is there any hydraulic fluid flowing
from the pressure line?'].

legalvals(press-line-fluid-flow) = ['yes','no'].

rule-76: if util-pwr-sys-symptom =
    'Hydraulic Chattering or High Frequency Vibrations in System'
   and press-line-fluid-flow = 'yes'
   and display([nl,nl,'
   Replace the Check Valve using A1-H46AE-450-000,
   section WP 022 00.',nl,nl])
then hydr-prob-cause =
'Faulty Check Valve between Pump Pressure Outlet and Pressure Filter'.

rule-77: if util-pwr-sys-symptom =
    'Hydraulic Chattering or High Frequency Vibrations in System'
   and press-line-fluid-flow = 'no'
   and display([nl,nl,'
   Sorry, this situation is beyond the scope of this program.',nl,nl])
then hydr-prob-cause = 'UNKNOWN - NOT PROGRAMMED YET'.


/**********  END OF UTILITY HYDR POWER SYSTEM *********************/

/***********  UTILITY PRESSURE INDICATING SYSTEM  *****************/


whenfound(hydraulic-sub-system =
   'utility pressure indicating system') =
   util-press-ind-symptom.

question(util-press-ind-symptom) = ['
   What are the symptoms of the
   problem ("gripe")?',nl].

legalvals(util-press-ind-symptom) =
   ['no indicator pointer movement',
   'pressure reading high or low',
   'erratic or sluggish pointer movement'].


/*-----------UTILITY PRESSURE INDICATING SYSTEM SYMPTOMS:
                     NO INDICATOR POINTER MOVEMENT              ------*/

whenfound(util-press-ind-symptom =
   'no indicator pointer movement') =
   display([nl,nl,'
   Check the HYDRAULIC BOOST PRESSURE UT
   circuit breakers located on the cockpit overhead console.',nl,nl,nl]).

rule-78: if util-press-ind-symptom = 'no indicator pointer movement'
   and 'circuit breakers are out'
   and display([nl,nl,'
   You need to push in the HYDRAULIC BOOST PRESSURE UT
   circuit breakers located on the cockpit overhead console,
   before you can begin troubleshooting.',nl,nl])
then hydr-prob-cause = 'No AC electrical power'.

whenfound('circuit breakers are out' = 'no') =
   display([nl,nl,nl,nl,'
   If you have checked to make sure that the HYDRAULIC BOOST
   PRESSURE UT circuit breakers located on the
   cockpit overhead panel are in fact IN, now
   check for 26 vac at pin B of indicator plug and pin B of the
   transmitter plug.
   Check wiring continuity.',nl,nl,nl]).
```

```
rule-79: if util-press-ind-symptom = 'no indicator pointer movement'
   and not('circuit breakers are out')
   and 'connector or wiring continuity' = 'bad'
   and display([nl,nl,'
   Repair the wiring or the connector using
   A1-H46AE-420-000, WP 004 00.',nl,nl,nl])
then hydr-prob-cause = 'Faulty Wiring or Connector'.

rule-80: if util-press-ind-symptom = 'no indicator pointer movement'
   and not('circuit breakers are out')
   and 'connector or wiring continuity' = 'good'
   and 'accum gauge press' = 'normal'
   or  'accum gauge press' = 'high'
   or  'accum gauge press' = 'low'
   or  'accum gauge press' = 'fluctuating'
   and display([nl,nl,'
   If the accumulator pressure gauges are reading normal, high,
   low, or fluctuating,
   then the indicator is faulty. Replace the indicator  using
   A1-H46AE-500-000, WP 033 00.',nl,nl,nl]) then hydr-prob-cause =
   'Faulty Flight Control Pressure Indicator'.

question('accum gauge press') = [nl,nl,'
What are the accumulator pressure
gauges reading?',nl].

legalvals('accum gauge press') =
['high','low','normal','fluctuating','no pressure'].

whenfound('accum gauge press' = 'no pressure') =
   display([nl,nl,'
   If you are sure that the accumulator gauge pressure is indicating
   no pressure,
   you must now troubleshoot the hydraulic Utility Power System using
   A1-H46AE-450-000, section WP 008 00.',nl,nl]).

rule-81: if util-press-ind-symptom = 'no indicator pointer movement'
   and not('circuit breakers are out')
   and 'connector or wiring continuity' = 'good'
   and 'accum gauge press' = 'no pressure'
then hydr-prob-cause =
   'No System Pressure - Hydraulic Utility Power System faulty'.

/*----UTILITY PRESSURE INDICATING SYSTEM SYMPTOMS:
                PRESSURE READING HIGH OR LOW                    ------*/

whenfound(util-press-ind-symptom =
   'pressure reading high or low') =
   display([nl,nl,'
   The first troubleshooting step is to:

      Troubleshoot the pressure TRANSMITTER and INDICATOR.
      Troubleshooting procedures for these two parts have not been
      programmed yet.

   Second troubleshooting step:

      Check the wiring continuity.',nl,nl]).

rule-82: if util-press-ind-symptom = 'pressure reading high or low'
   and 'wiring continuity' = 'bad'
   and display([nl,nl,'
   Repair the wiring or the connector using
   A1-H46AE-420-000, WP 004 00.',nl,nl,nl])
then hydr-prob-cause = 'Faulty Wiring or Ground Connection'.

rule-83: if util-press-ind-symptom = 'pressure reading high or low'
   and 'wiring continuity' = 'good'
   and display([nl,nl,'
   Sorry, CADS does not know what the problem is if the wiring
   continuity checks "good".',nl,nl,nl])
then hydr-prob-cause = 'UNKNOWN - NOT PROGRAMMED YET'.


/*----------UTILITY PRESSURE INDICATING SYSTEM SYMPTOMS:
```

SLUGGISH OR ERRATIC POINTER MOVEMENT ------*/

```
whenfound(util-press-ind-symptom =
   'erratic or sluggish pointer movement') =
   display([nl,nl,'
   The first troubleshooting step is to:

      Troubleshoot the pressure TRANSMITTER and INDICATOR.
      Troubleshooting procedures for these two parts have not been
      programmed yet.

   Second troubleshooting step:

      Check the connectors for tightness, and
      wiring for continuity.',nl,nl]).

rule-84: if util-press-ind-symptom =
   'erratic or sluggish pointer movement'
   and 'connector tightness and wiring continuity' = 'no'
   and display([nl,nl,'
   Tighten the connectors.
   Repair the wiring, using
   A1-H46AE-420-000, section WP 004 00.',nl,nl,nl])
then hydr-prob-cause = 'Intermittent Short in Wiring Circuit'.

rule-85: if util-press-ind-symptom =
   'erratic or sluggish pointer movement'
   and 'connector tightness and wiring continuity' = 'yes'
then hydr-prob-cause =
   'Fluctuating Pressure - Hydraulic Flight Control System' cf 60.

/****************  END OF UTILITY SYSTEM MODULE  ***************/
```

```
/*********** MAJOR SYSTEM: HYDRAULIC **********************
     ---------HYDRAULIC SUB-SYSTEM:
                        FLIGHT CONTROL HYDRAULIC SYSTEM-----------*/

goal = hydr-prob-cause.
automaticmenu(ALL).
enumeratedanswers(ALL).
presupposition(flt-ctrl-hydr-sys-symptom) =
   major-system = 'hydraulic'
   and hydraulic-sub-system = 'flight control hydraulic system'.
presupposition(flt-ctrl-press-ind-symptom) =
   major-system = 'hydraulic'
   and hydraulic-sub-system =
   'flight control pressure indicating system'.
question(flt-ctrl-hydr-sys-symptom) = ['
   What are the symptoms of the problem?',nl].
legalvals(flt-ctrl-hydr-sys-symptom) =
   ['low pressure-system no.1','low pressure-system no.2',
   'system no.1 pressure fluctuates',
   'system No.2 pressure fluctuates',
   'System No.1 or No.2 Cannot Be Shut Off',
   'Hardover Condition In Dual Extensible Links'].

/*----SYMPTOM: LOW PRESSURE SYSTEM NO.1--------------------*/

whenfound(flt-ctrl-hydr-sys-symptom = 'low pressure-system no.1') =
   display([nl,'
   Pressurize system with a teststand.  Check system for correct
   pressure.',nl,nl]).
rule-1: if flt-ctrl-hydr-sys-symptom = 'low pressure-system no.1'
   and 'system pressure buildup is correct'
then hydr-prob-cause = 'faulty pump'.
question('system pressure buildup is correct') =
   [nl,'Is the teststand pressure buildup correct?'].
legalvals('system pressure buildup is correct') = ['yes','no'].
whenfound('system pressure buildup is correct' = 'no') =
   display([nl,'
   If you are sure that the pressure buildup is correct using the
   teststand, now

   Check the relief valve cracking pressure.',nl,nl]).
rule-2: if flt-ctrl-hydr-sys-symptom = 'low pressure-system no.1'
   and 'relief valve cracking pressure low'
then hydr-prob-cause = 'relief valve opens at low pressure'.
question('relief valve cracking pressure low') =
   [nl,nl,'
   Is the relief valve cracking at low pressure?',nl].
legalvals('relief valve cracking pressure low') = ['yes','no'].
rule-3: if flt-ctrl-hydr-sys-symptom = 'low pressure-system no.1'
   and not('system pressure buildup is correct')
   or not ('relief valve cracking pressure low')
   and display([nl,nl,'Geez, I do not know what is wrong either!
   It is not in the MIMS, cause I already looked.',nl])
then hydr-prob-cause = 'UNKNOWN'.
```

```
/*---------SYMPTOM:  LOW PRESSURE SYSTEM NO.2--------------*/

whenfound(flt-ctrl-hydr-sys-symptom = 'low pressure-system no.2') =
   display([nl,'
   Connect a direct reading gauge to manifold.  See A1-H46AE-450-
   000, section WP 010 00, Fig. 1 and 2.
   Check for correct pressure.',nl,nl]).

rule-4: if flt-ctrl-hydr-sys-symptom = 'low pressure-system no.2'
   and 'dir rdg gauge manifold press' = 'low'
then hydr-prob-cause = 'faulty pressure reducing valve or relief valve in
   package manifold'.

question('dir rdg gauge manifold press') = [nl,'
   What pressure is the direct reading
   gauge indicating at the manifold?',nl].

legalvals('dir rdg gauge manifold press') =
   ['high','low','fluctuating'].

rule-5: if flt-ctrl-hydr-sys-symptom = 'low pressure-system no.2'
   and 'dir rdg gauge manifold press' = 'high'
   and display([nl,nl,'
   Troubleshoot utility system for low pressure,
   using A1-H46AE-450-000, section WP 008 00.',nl,nl])
then hydr-prob-cause = 'low utility system power supply'.

/*-------SYMPTOM:  SYSTEM NO.1 PRESSURE FLUCTUATES------*/

whenfound(flt-ctrl-hydr-sys-symptom =
     'system no.1 pressure fluctuates') =
   display([nl,'
   Troubleshoot the indicating system using
   A1-H46AE-450-000, section WP 011 00.',nl,nl]).

rule-6: if flt-ctrl-hydr-sys-symptom = 'system no.1 pressure fluctuates'
   and 'faulty indicating system'
then hydr-prob-cause = 'faulty indicating system'.

question('faulty indicating system') = [nl,'
   Does the indicating system have problems?',nl].

legalvals('faulty indicating system') = ['yes','no'].

whenfound('faulty indicating system' = 'no') =
   display([nl,'
   If you have troubleshot the indicating system, and are sure
   that it works properly, then:

   Check system pressure with transmissions turning.  Connect
   a direct reading gauge to PRESS OUT port of system no.1 package
   manifold.
   Use A1-H46AE-450-000, section WP 034 00.',nl,nl]).

rule-7: if flt-ctrl-hydr-sys-symptom = 'system no.1 pressure fluctuates'
   and 'manifold pressure'
then hydr-prob-cause = 'Faulty pump compensator'.

question('manifold pressure') = [nl,'
Does the direct reading gauge indicate
fluctuation more than _+100 psi, or
drops below minimum gauge pressure?'].

legalvals('manifold pressure') = ['yes','no'].

whenfound('manifold pressure' = 'no') =
   display([nl,'
   Check the relief valve cracking pressure.',nl,nl]).

question('relief valve crkg press') = [nl,'
Is the relief valve cracking at the
correct pressure?'].

legalvals('relief valve crkg press') = ['yes','no'].
```

```
rule-8: if flt-ctrl-hydr-sys-symptom = 'system no.1 pressure fluctuates'
   and 'relief valve crkg press' = 'no'
then hydr-prob-cause = 'faulty relief valve'.

whenfound('relief valve crkg press' = 'yes') =
   display([nl,'
   Check vent filter and line for restrictions.',nl,nl]).

question('vent filter and line restricted') = [nl,'
   Is the vent filter and line restricted?',nl].

legalvals('vent filter and line restricted') = ['yes','no'].

rule-9: if flt-ctrl-hydr-sys-symptom = 'system no.1 pressure fluctuates'
   and 'vent filter and line restricted'
then hydr-prob-cause = 'restricted air flow through vent filter'.

rule-10: if flt-ctrl-hydr-sys-symptom = 'system no.1 pressure fluctuates'
   and not('faulty indicating system')
   and 'relief valve crkg press'
   and not('vent filter and line restricted')
   and not('manifold pressure')
   and display([nl,nl,'
   Sorry.  This problem is beyond the scope of this
   program.',nl,nl])
then hydr-prob-cause = 'UNKNOWN - keep troubleshooting'.


/*----------------FLT CTRL SYMPTOMS:  SYS NO.2 PRESSURE
                                       FLUCTUATES----------------*/


whenfound(flt-ctrl-hydr-sys-symptom = 'system No.2 pressure fluctuates') =
   display([nl,'
   Troubleshoot the indicating system, using
   A1-H46AE-450-000, section WP 011 00.',nl,nl]).

rule-11: if flt-ctrl-hydr-sys-symptom = 'system No.2 pressure fluctuates'
   and 'faulty indicating system'
then hydr-prob-cause = 'faulty indicating system'.

whenfound('faulty indicating system' = 'no') =
   display([nl,nl,nl,nl,'
   If you have troubleshot the indicating system, and are sure
   that it works properly, then:

   Connect a direct reading gauge to manifold.  See A1-H46AE-450-
   000, section WP 010 00, Fig. 1 and 2.
   Check for correct pressure.',nl,nl]).

rule-12: if flt-ctrl-hydr-sys-symptom = 'system No.2 pressure fluctuates'
   and 'dir rdg gauge manifold press' = 'high'
   or  'dir rdg gauge manifold press' = 'low'
then hydr-prob-cause = 'faulty pressure reducing valve in
   package manifold'.

rule-13: if flt-ctrl-hydr-sys-symptom = 'system No.2 pressure fluctuates'
   and 'dir rdg gauge manifold press' = 'fluctuating'
   and display([nl,nl,nl,'
   Since you entered "fluctuating" for the direct reading gauge
   pressure check, now:

   Troubleshoot the utility system.',nl,nl,nl])
then hydr-prob-cause = 'Utility system power supply fluctuating'.


/*--------------FLT CTRL SYMPTOMS:  SYS NO.1 OR NO.2 CANNOT
                BE SHUT OFF-----------------------------------------*/


whenfound(flt-ctrl-hydr-sys-symptom =
   'System No.1 or No.2 Cannot Be Shut Off') =
   display([nl,'
   Your first troubleshooting check will be:

   Check for 28 vdc at pin 2 of module connector 033P1.',nl,nl]).
```

```
question('module connector voltage') = [nl,'
Is 28 vdc present at pin 2 of the
module connector 033P1?'].

legalvals('module connector voltage') = ['yes','no'].

rule-14: if flt-ctrl-hydr-sys-symptom =
   'System No.1 or No.2 Cannot Be Shut Off'
   and not('module connector voltage')
then hydr-prob-cause = 'System No.1 module pilot valve control
   circuits are open'.

whenfound('module connector voltage' = 'yes') =
   display([nl,'
   Your next troubleshooting check will be:

   Check for 28 vdc at pin A of package manifold plug
   P765 - 033P2.',nl,nl]).

question('manifold plug P765 voltage') = [nl,'
Is 28 vdc present at pin A of the
package manifold plug P765?'].

legalvals('manifold plug P765 voltage') = ['yes','no'].

rule-15: if flt-ctrl-hydr-sys-symptom = 'System No.1 or
   No.2 Cannot Be Shut Off'
   and 'module connector voltage'
   and not('manifold plug P765 voltage')
then hydr-prob-cause = 'System No.2 package manifold solenoid
   valve control circuit is open'.

whenfound('manifold plug P765 voltage' = 'yes') =
   display([nl,'
   Your next troubleshooting check will be:

   Check for continuity of ground circuit through opposite system
   pressure switch.',nl,nl]).

question('manifold solenoid ground') = [nl,'
Is there continuity of the ground circuit
through the opposite system pressure switch?'].

legalvals('manifold solenoid ground') = ['yes','no'].

rule-16: if flt-ctrl-hydr-sys-symptom =
   'System No.1 or No.2 Cannot Be Shut Off'
   and 'module connector voltage'
   and 'manifold plug P765 voltage'
   and not('manifold solenoid ground')
then hydr-prob-cause = 'Package Manifold Solenoid
   valve ground circuit is open'.

whenfound('manifold solenoid ground' = 'yes') =
   display([nl,'
   Your next troubleshooting check will be:

   With 28 vdc applied and ground circuit completed, the valve
   should operate properly.',nl,nl]).

question('manifold solenoid valve operation') = [nl,'
Does the module solenoid valve in the
package manifold operate properly?'].

legalvals('manifold solenoid valve operation') = ['yes','no'].

rule-17: if flt-ctrl-hydr-sys-symptom =
   'System No.1 or No.2 Cannot Be Shut Off'
   and 'module connector voltage'
   and 'manifold plug P765 voltage'
   and 'manifold solenoid ground'
   and not('manifold solenoid valve operation')
then hydr-prob-cause = 'Faulty Package Manifold Module Solenoid Valve'.
```

```
rule-18: if flt-ctrl-hydr-sys-symptom =
   'System No.1 or No.2 Cannot Be Shut Off'
   and 'module connector voltage'
   and 'manifold plug P765 voltage'
   and 'manifold solenoid ground'
   and 'manifold solenoid valve operation'
   and display([nl,'
   All the checks you did were good.
   Sorry, the problem is beyond the scope of this program.',nl,nl])
then hydr-prob-cause = 'UNKNOWN - not programmed yet'.

/*---------------FLT CTRL SYMPTOMS:  HARDOVER CONDITION
                     IN DUAL EXTENSIBLE LINKS----------------------*/

whenfound(flt-ctrl-hydr-sys-symptom =
   'Hardover Condition In Dual Extensible Links') =
   display([nl,'
   Check the Dual Extensible Links filter elements for evidence of
   contamination.',nl,nl]).

question('dual ext link filters') = [nl,'
Are all three extensible link filter
elements contaminated?'].

legalvals('dual ext link filters') = ['yes','no'].

rule-19: if flt-ctrl-hydr-sys-symptom =
   'Hardover Condition In Dual Extensible Links'
   and 'dual ext link filters'
then hydr-prob-cause = 'Contamination in affected system'.



/*******  HYDRAULIC SUB-SYSTEM:
         FLIGHT CONTROL PRESSURE INDICATING SYSTEM  *****************/



question(flt-ctrl-press-ind-symptom) = ['
   What are the symptoms of the
   problem ("gripe")?',nl].

legalvals(flt-ctrl-press-ind-symptom) =
   ['indicator pointer does not move',
   'low pressure indication','high pressure indication',
   'erratic or sluggish pointer movement'].


/*----FLIGHT CONTROL PRESSURE INDICATING SYSTEM SYMPTOMS:
                  INDICATOR POINTER DOES NOT MOVE           -----

The following troubleshooting procedure "Checking the HYDRAULIC
BOOST PRESSURE Circuit Breakers are in, is not considered a
troubleshooting procedure by the NAESU tech reps.  It is
commented out of this program.  It appears here only because it
took allot of programming work to put it in!! ----------------

whenfound(flt-ctrl-press-ind-symptom =
   'indicator pointer does not move') =
   display([nl,nl,'
   Check the HYDRAULIC BOOST PRESSURE NO.1 and NO.2 circuit
   breakers located on the cockpit overhead panel.',nl,nl,nl]).
```

```
rule-24: if flt-ctrl-press-ind-symptom =
   'indicator pointer does not move'
   and 'circuit breakers are out'
   and display([nl,nl,'
   You need to push in the HYDRAULIC BOOST PRESSURE NO.1 and NO.2
   circuit breakers located on the cockpit overhead panel,
   before you can begin troubleshooting.',nl,nl])
then hydr-prob-cause = 'no AC electrical power'.

question('circuit breakers are out') = [nl,'
Are the flight control pressure
indicator circuit breakers out?',nl].

legalvals('circuit breakers are out') = ['yes','no'].--------*/

whenfound(flt-ctrl-press-ind-symptom =
   'indicator pointer does not move') =
   display([nl,nl,'
   The first troubleshooting check of this symptom is:

   check for 26 vac at pin B of indicator socket and pin B of the
   transmitter.
   Check wiring continuity.',nl,nl,nl]).

rule-25: if flt-ctrl-press-ind-symptom =
   'indicator pointer does not move'
   and 'connector or wiring continuity' = 'bad'
   and display([nl,nl,'
   Repair the wiring or the connector using
   A1-H46AE-420-000, WP 004 00.',nl,nl,nl])
then hydr-prob-cause = 'Faulty Wiring or Connector'.

question('connector or wiring continuity') = [nl,'
If either of these is "bad" enter "bad":
   26 vac at pin B of the indicator
   socket and pin B of transmitter
  OR
   wiring continuity.
If both are "good" enter "good".'].

legalvals('connector or wiring continuity') = ['good','bad'].

whenfound('connector or wiring continuity' = 'good') =
   display([nl,nl,'
   If you are sure that the connector and that wiring
   continuity are both "good", do the following:

          check the pressure with a direct reading
          gauge at the pressure transmitter.',nl,nl]).

rule-26: if flt-ctrl-press-ind-symptom =
   'indicator pointer does not move'
   and 'connector or wiring continuity' = 'good'
   and 'dir rdg press' = 'normal'
   and display([nl,nl,'
   If the direct reading pressure at the transmitter is reading
   normally, then the indicator is faulty.
   Replace the indicator  using A1-H46AE-500-000, WP 012 00.',nl,nl,nl])
then hydr-prob-cause = 'Faulty Flight Control Pressure Indicator'.

question('dir rdg press') = [nl,nl,'
What is the direct reading pressure
at the transmitter indicating?',nl].

legalvals('dir rdg press') =
['high','low','normal','fluctuating'].
```

```
/*----FLIGHT CONTROL PRESSURE INDICATING SYSTEM SYMPTOMS:
                     LOW PRESSURE INDICATION                   ------*/

whenfound(flt-ctrl-press-ind-symptom =
   'low pressure indication') =
   display([nl,nl,'
   Check the reading on the hydraulic system accumulators (direct
   reading gauges).
   If the hydraulic accumulators are indicating normal pressures in
   the hydraulic systems, then the indicators or the transmitters
   are faulty.',nl,nl]).

rule-29: if flt-ctrl-press-ind-symptom= 'low pressure indication'
   and 'dir rdg press' = 'normal'
   and display([nl,nl,'
   If the direct reading gauge indicated normal pressure,
   then replace the indicator.',nl])
then hydr-prob-cause =
   'Faulty indicator or transmitter'.

/*----FLIGHT CONTROL PRESSURE INDICATING SYSTEM SYMPTOMS:
                     HIGH PRESSURE INDICATION                   ------*/

whenfound(flt-ctrl-press-ind-symptom =
   'high pressure indication') =
   display([nl,nl,'
   The first troubleshooting step is to:

      Troubleshoot the pressure TRANSMITTER and INDICATOR.
      Compare the hydraulic accumulator pressure (direct reading
      gauge) to the indicator.  If the accumulators are reading
      normal pressure, then the indicators or transmitter is
      faulty.

   Second troubleshooting step:

   Check the wiring continuity.',nl,nl]).

rule-30: if flt-ctrl-press-ind-symptom = 'high pressure indication'
   and 'wiring continuity' = 'bad'
   and display([nl,nl,'
   Repair the wiring using
   A1-H46AE-420-000, WP 004 00.',nl,nl,nl])
then hydr-prob-cause = 'Incorrectly Wired Indicator or Transmitter'.

/*----FLIGHT CONTROL PRESSURE INDICATING SYSTEM SYMPTOMS:
                     SLUGGISH OR ERRATIC POINTER MOVEMENT        ------*/

whenfound(flt-ctrl-press-ind-symptom =
   'erratic or sluggish pointer movement') =
   display([nl,nl,'
      Troubleshoot the pressure TRANSMITTER and INDICATOR.
      Compare the hydraulic accumulator pressure (direct reading
      gauge) to the indicator.  If the accumulators are reading
      normal pressure, then the indicators or transmitter is
      faulty.',nl,nl]).

rule-32: if flt-ctrl-press-ind-symptom =
   'erratic or sluggish pointer movement'
   and 'dir rdg press' = 'normal'
   and display([nl,nl,'
   If the direct reading gauge indicated normal pressure,
   then replace the indicator or the transmitter.',nl])
then hydr-prob-cause =
   'Faulty indicator or transmitter'.

/*****************  END OF HYDRAULICS MODULE  *****************/
```

LIST OF REFERENCES

1. Chief of Naval Operations, OPNAVINST 4790.2B: *The Naval Aviation Maintenance Program*, v. 1, 1 October 1986.

2. Chief of Naval Operations, OPNAVINST 4790.2B: *The Naval Aviation Maintenance Program*, v. 2, 1 October 1986.

3. Commander, Naval Air Systems Command, NAVAIR A1-H46AE-420-000: *Technical Manual: Organizational/Intermediate Maintenance with Illustrated Parts Breakdown, Electrical Systems, Navy Model H-46A/D/E Helicopters*, sec. WP 008 00, 15 May 1986.

4. Harmon, P., and D. King, *Expert Systems: Artificial Intelligence in Business*, John Wiley & Sons, Inc., 1985.

5. Hayes-Roth, F., D.A. Waterman, and D.B. Lenat, eds., *Building Expert Systems*, Addison-Wesley Publishing Co., Inc., 1983.

6. Barr, A. and E. Feigenbaum, *The Handbook of Artificial Intelligence, Volume II*, William Kaufmann, Inc., 1982.

7. Teknowledge, Inc., *M.1 Reference Manual for Software Version 2.1, March 1986*, Teknowledge, Inc., 1986.

8. Yavorsky, J. S., *Development of a Prototype Multichannel Communication Network Maintenance Expert System*, M.S. Thesis, Naval Postgraduate School, Monterey, California, March 1987.

9. Teknowledge, Inc., *M.1 Sample Knowledge Systems, March 1986*, Teknowledge, Inc., 1986.

10. Grover, M. D., "A Pragmatic Knowledge Acquisition Methodology", *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, 1983, pp. 436-438.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Director of Information Systems Office of Chief of Naval Operations (OP-0945) Navy Department Washington, D.C. 20350-2000	1
4.	Professor Nancy C. Roberts Code 54Rc Naval Postgraduate School Monterey, CA 93943	1
5.	Professor Judith H. Lind Code 55Li Naval Postgraduate School Monterey, CA 93943	1
6.	Mr. Norm Clark P.O. Box 1270 Tustin, CA 92681	1
7.	Commander, Naval Air Systems Command Ext. PMA-261 Washington, D.C. 20361-1261	1
8.	Commanding Officer Helicopter Combat Support Squadron Three (HC-3) Naval Air Station, North Island San Diego, CA 92135-5199	1
9.	Commanding Officer Helicopter Combat Support Squadron Five (HC-5) U.S. Naval Air Station FPO San Francisco 96637-2714	1
10.	Commanding Officer Helicopter Combat Support Squadron Six (HC-6) Naval Air Station Norfolk, VA 23511	1

11. Commanding Officer 1
Helicopter Combat Support Squadron Eight (HC-8)
Naval Air Station
Norfolk, VA 23511-5693

12. Commanding Officer 1
Helicopter Combat Support Squadron Eleven (HC-11)
Naval Air Station, North Island
San Diego, CA 92135-5200

13. Thomas J. Gadzala, LCDR,USN 3
9016 Capcano Rd
San Diego, CA 92126

14. Dr. Jude Franklin 1
Navy Center for Applied Research in AI
U.S. Naval Research Laboratory
Code 7510
Washington, D.C. 20375

15. Dr. Ken DeJong 1
Navy Center for Applied Research in AI
U.S. Naval Research Laboratory
Code 7510
Washington, D.C. 20375

Gadzala
Development of a prototype H-46 helicopter diagnostic expert system.