

**NAVAL POSTGRADUATE SCHOOL**  
**Monterey, California**

1

**AD-A273 148**



**DTIC**  
**ELECTE**  
**NOV 30 1993**  
**A**

**THESIS**

**DEVELOPMENT OF A MAINTENANCE ADVISOR  
EXPERT SYSTEM FOR THE  
MK 92 MOD 2 FIRE CONTROL SYSTEM:  
FC-1 DESIGNATION - TIME, RANGE, BEARING FC-1  
ACQUISITION, FC-1 TRACK - RANGE, BEARING, AND FC-2  
DESIGNATION - TIME, RANGE, BEARING, FC-2  
ACQUISITION, FC-2 TRACK - RANGE, BEARING, AND FC-4  
AND FC-5.**

by

**Claude David Smith**

**September, 1993**

**Thesis Advisor:**

**Magdi Kamel**

**Approved for public release; distribution is unlimited.**

1388

**93-29177**



93 11 23 052

# REPORT DOCUMENTATION PAGE

1a Report Security Classification <b>Unclassified</b>			1b Restrictive Markings		
2a Security Classification Authority			3 Distribution/Availability of Report		
2b Declassification/Downgrading Schedule			Approved for public release; distribution is unlimited.		
4 Performing Organization Report Number(s)			5 Monitoring Organization Report Number(s)		
6a Name of Performing Organization Naval Postgraduate School		6b Office Symbol AS	7a Name of Monitoring Organization Naval Postgraduate School		
6c Address (city, state, and ZIP code) Monterey CA 93943-5000			7b Address (city, state, and ZIP code) Monterey CA 93943-5000		
8a Name of Funding/Sponsoring Organization Port Hueneme Division, Naval Surface Warfare Center		6b Office Symbol Code 4W32	9 Procurement Instrument Identification Number		
Address (city, state, and ZIP code) Port Hueneme, CA 93043			10 Source of Funding Numbers		
Program Element No		Project No	Task No	Work Unit Accession No	
11 Title (include security classification)/DEVELOPMENT OF A MAINTENANCE ADVISOR EXPERT SYSTEM FOR THE MK 92 MOD 2 FIRE CONTROL SYSTEM: FC-1 DESIGNATION - TIME, RANGE, BEARING, FC-1 ACQUISITION, FC-1 TRACK - RANGE, BEARING, AND FC-2 DESIGNATION - TIME, RANGE, BEARING, FC-2 ACQUISITION, FC-2 TRACK - RANGE, BEARING, AND FC-4 AND FC-5.					
12 Personal Author(s) Claude David Smith					
13a Type of Report Master's Thesis		13b Time Covered From To	14 Date of Report (year, month, day) September 10, 1993		15 Page Count 133
16 Supplementary Notation The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.					
17 Cosali Codes			18 Subject Terms (continue on reverse if necessary and identify by block number)		
Field	Group	Subgroup	Expert Systems, Knowledge Acquisition, Knowledge Representation, Knowledge Implementation, MK 92 MOD 2 Fire Control System		
19 Abstract (continue on reverse if necessary and identify by block number)					
<p>The MK 92 MOD 2 Fire Control System is a complex weapons system based on 1970's technology. It is a maintenance intensive system, requiring extensive technical trouble-shooting and, occasionally, supplemental shore based support. Development of an expert maintenance system for the MK 92 MOD 2 Fire Control System offers a viable solution to the labor intensive efforts of the technicians, reduces the number of visits by shore based support staff, and provides relief to an already overburdened maintenance budget. It will also significantly reduce the depot repair "no fault evident" rate which is the result of good parts replaced because of defective trouble-shooting.</p> <p>This thesis addresses the first iteration of prototype development of the performance channels of the MK 92 MOD 2 Maintenance Advisor Expert System. Specific issues covered include the scope of the project, hardware selection, system shell selection, knowledge acquisition, knowledge representation, knowledge implementation and lessons learned in the process.</p>					
Distribution/Availability of Abstract X unclassified/unlimited — same as report — DTIC users			21 Abstract Security Classification Unclassified		
22a Name of Responsible Individual Magdi Kamel			22b Telephone (include Area Code) (408) 656-2494		22c Office Symbol AS/Ka

Approved for public release; distribution is unlimited.

Development of a Maintenance Advisor Expert System for the  
MK 92 MOD 2 Fire Control System: FC-1 Designation - Time, Range, Bearing, FC-1  
Acquisition, FC-1 Track - Range, Bearing, and FC-2 Designation - Time, Range, Bearing, FC-2  
Acquisition, FC-2 Track - Range, Bearing, and FC-4 and FC-5.

by

Claude David Smith  
Lieutenant, United States Navy  
B.S., University of South Carolina, 1985

Submitted in partial fulfillment  
of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

from the

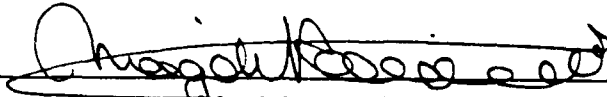
NAVAL POSTGRADUATE SCHOOL  
September 1993

Author:

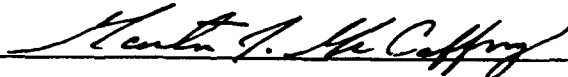


Claude David Smith

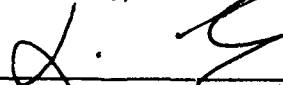
Approved by:



Magdi Kamel, Thesis Advisor



Martin J. McCaffrey, Associate Advisor



David R. Whipple, Chairman  
Department of Administrative Sciences

## ABSTRACT

The MK 92 MOD 2 Fire Control System is a complex weapons system based on 1970's technology. It is a maintenance intensive system, requiring extensive technical trouble-shooting and, occasionally, supplemental shore based support. Development of an expert maintenance system for the MK 92 MOD 2 Fire Control System offers a viable solution to the labor intensive efforts of the technicians, reduces the number of visits by shore based support staff, and provides relief to an already overburdened maintenance budget. It will also significantly reduce the depot repair "no fault evident" rate which is the result of good parts replaced because of defective trouble-shooting.

This thesis addresses the first iteration of prototype development of the performance channels of the MK 92 MOD 2 Maintenance Advisor Expert System. Specific issues covered include the scope of the project, hardware selection, system shell selection, knowledge acquisition, knowledge representation, knowledge implementation, and lessons learned in the process.

DEFENSE QUALITY INSPECTED 8

Accession For	
NTIS	CRA&I <input checked="" type="checkbox"/>
DTIC	PA&S <input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution	
Availability Codes	
Dist	Avail and/or Special
A-1	

## **TABLE OF CONTENTS**

<b>I. INTRODUCTION</b>	<b>1</b>
A. BACKGROUND	1
B. OBJECTIVES	2
C. SCOPE	2
D. METHODOLOGY	2
E. THESIS ORGANIZATION	3
 <b>II. BACKGROUND</b>	 <b>5</b>
A. MK 92 MOD 2 SYSTEM COMPONENTS	5
B. TROUBLE-SHOOTING PROCEDURES AND PROBLEMS	8
C. EXPERT SYSTEM AS A POSSIBLE SOLUTION	9
 <b>III. EXPERT SYSTEMS DEVELOPMENT CYCLE</b>	 <b>11</b>
A. INITIAL PHASES	11
1. Project Start-Up	12
2. Selection of the Domain	13
3. Selection of the Development Environment	14
B. CORE DEVELOPMENT PHASES	14

1.	Development of a Feasibility Prototype System . . . . .	15
2.	Development of a Full Prototype System . . . . .	16
C.	FINAL DEVELOPMENT AND DEPLOYMENT PHASES ..	17
1.	Development of a Production System . . . . .	17
2.	System Deployment . . . . .	18
D.	MK 92 MOD 2 MAINTENANCE ADVISOR EXPERT SYSTEM DEVELOPMENT CYCLE . . . . .	20
IV.	KNOWLEDGE ACQUISITION . . . . .	23
A.	KNOWLEDGE ACQUISITION PROCESS . . . . .	23
1.	PERSONAL INTERVIEWS . . . . .	24
2.	PERSONAL NOTEBOOKS . . . . .	25
3.	ROLE PLAYING . . . . .	25
4.	PICTURES OR DRAWINGS . . . . .	25
5.	MULTIPLE EXPERTS . . . . .	26
6.	QUESTIONNAIRES . . . . .	26
B.	MK 92 MOD 2 KNOWLEDGE ACQUISITION . . . . .	27
V.	KNOWLEDGE REPRESENTATION . . . . .	29
A.	KNOWLEDGE REPRESENTATION METHODS . . . . .	29
1.	Production Rules . . . . .	30
2.	Frames . . . . .	31

B.	SEMANTIC NETWORKS .....	34
C.	PROCEDURES .....	35
D.	MK 92 MOD 2 KNOWLEDGE REPRESENTATION .....	37
VI.	KNOWLEDGE IMPLEMENTATION .....	39
A.	KNOWLEDGE ACQUISITION PROCEDURES AND IMPLEMENTATION .....	39
B.	IMPLEMENTING AND DEBUGGING .....	40
C.	DOCUMENTATION .....	40
D.	UNIFORMITY OF STYLE .....	41
E.	VALIDATION AND VERIFICATION .....	41
F.	MK 92 MOD 2 MAINTENANCE ADVISOR EXPERT SYSTEM PROCEDURE BUILDER ISSUES .....	42
G.	DISPLAY BUILDER ISSUES .....	43
1.	Screen Layout .....	44
a.	Main Title Bar .....	44
b.	Procedure Box .....	44
c.	Action Box .....	44
2.	Colors .....	45
a.	Background .....	46
b.	Foreground .....	46
3.	Conventions .....	47
a.	Naming Conventions .....	47

b. Screen Conventions .....	48
4. Fonts .....	48
 VII. LESSONS LEARNED .....	 50
A. KNOWLEDGE EXPERT AS KNOWLEDGE ACQUIRER ...	50
B. PROCEDURES PARADIGM .....	51
C. EXPERT SYSTEM TOOLS--ADEPT .....	51
D. SELLING THE PRODUCT .....	54
 APPENDIX A .....	 56
APPENDIX B .....	110
LIST OF REFERENCES .....	120
BIBLIOGRAPHY .....	122
INITIAL DISTRIBUTION LIST .....	123



## **ACKNOWLEDGMENTS**

I wish to express my sincere gratitude to Tess Miller who assisted in the quality assurance of the expert system program, thesis proofreading, typing and editing, and most of all for being a good friend.

## **I. INTRODUCTION**

### **A. BACKGROUND**

The MK 92 MOD 2 Fire Control System (FCS) is a complex weapons system based on 1970's technology. Maintaining such an old system to peak performance requires an extensive maintenance effort by the technicians. Often, they are unable to correctly identify malfunctioning components when attempting to isolate system failures. This leads to a waste of valuable man hours and replacement of perfectly good components and often results in extended system down time. Ships frequently request technical assistance from shore based commands which necessitates sending an expert potentially long distances to assist in fault isolation of the fire control system.

Due to the current trends in downsizing, the number of senior, experienced technicians is decreasing and funds for technical support visits are not as readily available. The development of a maintenance advisor expert system for the MK 92 MOD 2 FCS has the potential to substantially reduce the number of requests for technical assistance from shore based technicians and significantly reduce the dollars and time spent on misdiagnosing system faults.

## **B. OBJECTIVES**

This thesis addresses the design and implementation of a prototype MK 92 MOD 2 Maintenance Advisor Expert System. It deals with all phases of development, with specific emphasis on knowledge acquisition, representation and implementation. It is not intended to be a comprehensive guide for systems development, but a discussion of the process followed in this prototype development.

## **C. SCOPE**

This thesis develops a full prototype system rather than a fully operational system of the MK 92 FCS Daily System Operability Test (DSOT). Specifically, this thesis addresses the diagnosis and trouble-shooting of the performance components of the system. These include: FC-1 Designation - Time, Range, Bearing, FC-1 Acquisition, FC-1 Track - Range, Bearing, and FC-2 Designation - Time, Range, Bearing, FC-2 Acquisition, FC-2 Track - Range, Bearing, and FC-4 and FC-5.

## **D. METHODOLOGY**

The system development closely followed the four step methodology outlined by Prerau. (Prerau, 1990, p.14) Step one involves selecting the domain for the system. The domain was defined to include only the performance portion of the DSOT. Step two involves identification of the domain expert or experts. It was decided to use a Paramax Corporation

engineer as the primary knowledge expert. Step three involves the actual knowledge acquisition. The Paramax knowledge expert used his own expertise, as well as a number of other resources in crafting his knowledge. Step four involves actual system development, including hardware and software selection, knowledge representation, implementation and programming.

An initial feasibility prototype of the MK 92 MOD 2 was quickly developed and presented to the department heads and engineers of the Tartar Systems Department at Port Hueneme. The presentation was well received and a decision was reached to continue funding the project.

Several meetings between the students and the NSWC engineers were held at Port Hueneme and the Naval Postgraduate School. An additional meeting was conducted at the Fleet Training Center in San Diego, California, where a portion of the system was demonstrated to instructors and student fire control technicians who maintain the MK 92 MOD 2 System. This provided valuable feedback from experienced personnel on the utility of the project.

## **E. THESIS ORGANIZATION**

This thesis is organized in the following manner:

Chapter II contains a description of the MK 92 MOD 2 FCS components and their interrelationships. Expert system technology is introduced as a

means of assisting shipboard technicians in trouble-shooting and maintaining their assigned equipment.

Chapter III addresses the methodologies of the expert system development cycle, with particular emphasis on the paradigm discussed by Prerau. Additional discussion of the MK 92 MOD 2 prototype development cycle is also included in this chapter.

Chapter IV discusses knowledge acquisition methodologies and the specific processes utilized. Selection of domain experts and interface between the experts and students is also addressed in this chapter.

Chapter V discusses knowledge representation paradigms. Special emphasis is placed on the method used by the MK 92 MOD 2 knowledge expert and procedure based knowledge representation.

Chapter VI covers knowledge implementation procedures, program architecture, and display design, including screen layouts, colors, fonts and graphics. Procedural logic and representation is depicted via a series of structured diagrams for the entire prototype.

Chapter VII discusses lessons learned from the system prototype development. Special attention is given to unique insights gained from the use of the expert system development tool.

Appendix A is a user's manual intended to provide the user with instructions on how to install and run the program. Appendix B provides procedural function descriptions and program logic diagrams.

## **II. BACKGROUND**

The MK 92 MOD 2 Fire Control System (FCS) is a lightweight, high performance multi-purpose Fire Control System. It can be found on board the United States Navy's Oliver Hazard Perry class Guided Missile Frigates (FFG's) and Patrol Hydrofoil Missile class (PHM's), U.S. Coast Guard High and Medium Endurance Cutters, and Australian Anzac and FFG 7 class ships. The MK 92 MOD 2 is part of an integrated system which includes separate air search radar (AN/SPS-49) and surface search radar (AN/SPS-55). The data from these search radars is combined with MK 92 MOD 2 fire information and displayed to the system operators via the FCS consoles. The system is capable of tracking air and surface contacts and providing fire control solutions for the gun and missile. To effect a fire control solution the system must perform the following tasks: locate and track air, surface, and shore targets; anticipate future target positions with respect to own ship's course and speed; and train the gun and missile launcher to intercept and destroy the designated target.

### **A. MK 92 MOD 2 SYSTEM COMPONENTS**

As Figure 2-1 shows, the major components of the system are a Univac AN/UYK-7 digital computer, a MK 75 (76mm) gun, a medium range

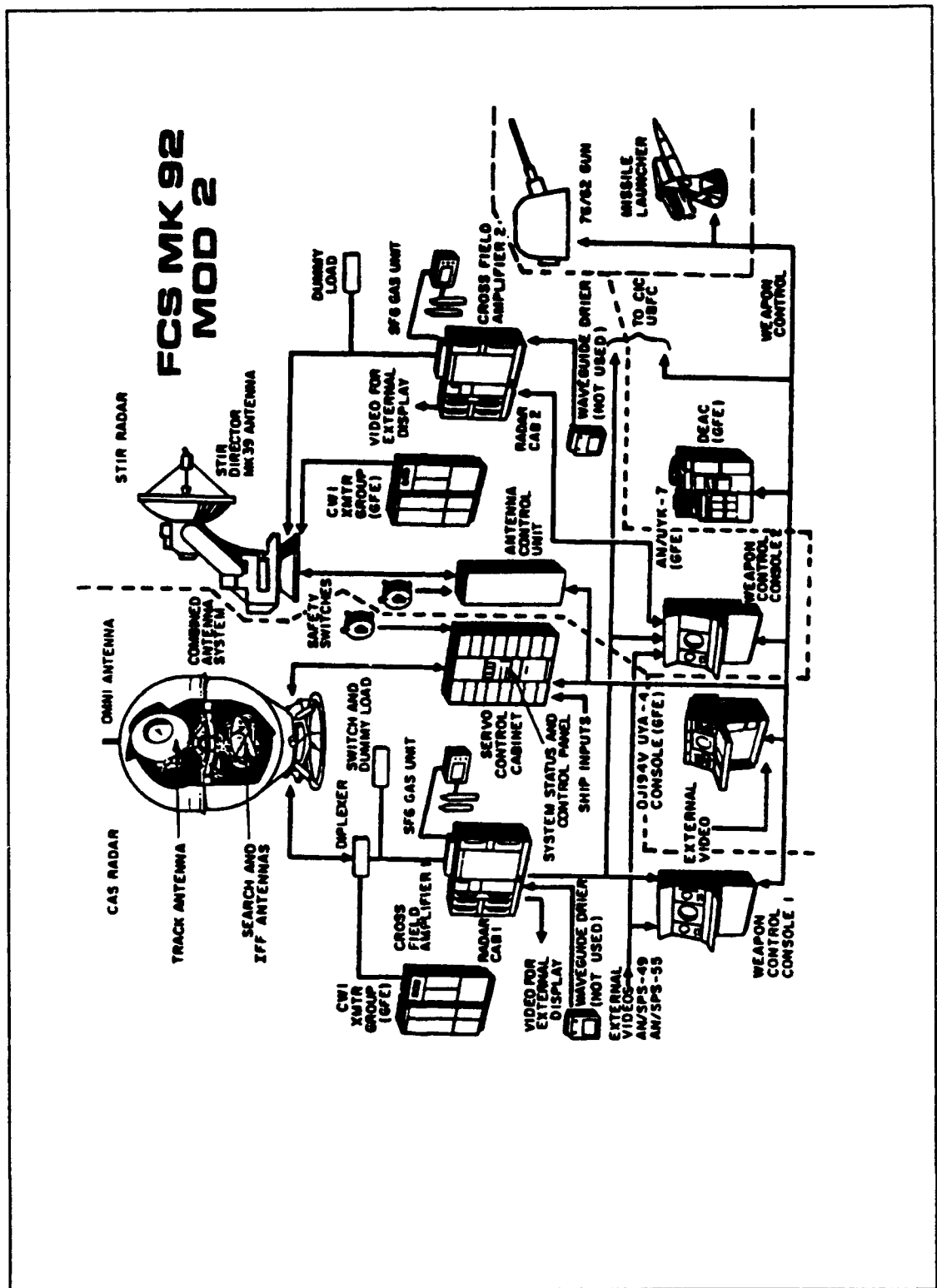


FIGURE 2-1 MK 92 MOD 2 SYSTEM DIAGRAM

standard missile (SM1 MR), a MK 13 guided missile launcher system(GLMS), two weapon control consoles (WCC1 and WCC2), a MK 39 separate track illuminating radar (STIR), and a MK 53 combined antenna system (CAS), consisting of the track antenna and search/identify friend or foe (IFF) antenna. U.S. Coast Guard cutters, while not equipped with missiles, use the system in firing their 76mm gun.

Air targets can be engaged by either gun or missile depending on the mode selected by the operators. Likewise, either weapon can be directed against surface contacts. Firing Channels (FC) are used to differentiate various modes of usage. FC1 and FC2 are for guns and missiles and are assigned air contacts. FC4 and FC5 are for the gun only and are normally assigned surface contacts, although these contacts could be assigned to FC1 or FC2.

An associated sub-system is the Daily System Operability Test(DSOT) set. The DSOT provides a rapid, extensive assessment of the operational readiness of the MK 92 MOD 2 system. This automated test injects signals to thoroughly evaluate the system responsiveness to programmed target parameters. The operator is provided with an equipment summary via the alphanumeric display (TOTE) or with hard copy printouts via the Data Exchange Auxiliary Console (DEAC). Normally, these tests are conducted daily while underway, operational activity allowing. As an added safety



measure, if an actual target is detected, the DSOT system automatically terminates. Additional maintenance and system checks are accomplished as part of the preventative maintenance system (PMS) program and are scheduled either according to system usage or time interval.

## **B. TROUBLE-SHOOTING PROCEDURES AND PROBLEMS**

The DSOT output can indicate one or more NOGO's. NOGO's from the DSOT indicate the system is not functioning properly. The technician, usually a Navy Petty Officer or Chief Petty Officer, begins analyzing the trouble area based on his expert knowledge of the system and with the help of technical maintenance manuals on board.

If the ship's technicians are unable to repair the system, additional technical support may be requested from the Mobile Training Unit (MOTU), NAVSEA, or the Naval Surface Warfare Center at Port Hueneme, California. The Port Hueneme engineers may respond via a message recommending procedures to remedy the problem or may elect to dispatch technicians to the ship to effect the repair.

Needless to say, dispatching an expert sometimes halfway around the world is expensive and time consuming. If the weapon system, however, is mission critical, it must be repaired at any cost. No commanding officer can afford to enter into combat with a system that may not function properly.

Shipboard technicians often trouble-shoot the problem down to a circuit card that appears to be defective, only to have the part returned from depot level repair as "no fault evident" (NFE). Whether or not the suspected component card is defective, the command must still bear the transportation costs and cost of repairing a perfectly good unit. At other times, the same card may be replaced multiple times before the actual source of the problem is isolated. Not only are replacement costs incurred, but valuable time is wasted by initial improper trouble-shooting. In many cases, the maintenance manuals only isolate a problem to a group of cards.

A recent study of Casualty Reports (CASREPs), from 1 July 1990 to 30 June 1991, submitted by the fifty MK 92 MOD 2 equipped ships in the U.S. Navy, found that \$1,475,692 was spent in replacing unnecessary parts. This figure represents 11 percent of total funding these units spent supporting their FCS during fiscal year 1991. (Powell, 1993, p. 38)

### **C. EXPERT SYSTEM AS A POSSIBLE SOLUTION**

Artificial intelligence (AI), through the use of expert systems, offers a potential solution to the problems discussed above. Expert systems are advanced computer software programs that emulate the expertise of human experts in a specific domain. These systems use knowledge techniques, heuristics, and other problem solving techniques used by human experts to solve such problems. Expert systems are particularly useful in design,

process monitoring, and diagnostic applications. (Leonard-Barton, 1988, p. 91) An expert system can provide the shipboard technician with expert consulting anywhere in the world, at any time, and at minimal cost. Such advisory systems are already in place in business, manufacturing, and even in health care.

Fault isolation offers an excellent opportunity for employment of an expert advisor system. Such a system might be able to locate unlikely causes of faults that human trouble-shooters do not investigate because the odds of finding such unlikely causes do not usually warrant the time needed for analysis. Capturing the knowledge of true experts may reduce fault localization time, since these individuals, based on years of experience, know the shortcuts to finding those faults.

Although expert system technology cannot be the panacea for all trouble-shooting problems, there are potential savings to be realized in implementing a maintenance advisor for the MK 92 MOD 2 FCS. If the system can improve the technicians trouble-shooting skills by one third, yearly savings may be in the tens of thousands of dollars. More importantly, to the operational commander, the improved system readiness through less system down time cannot be measured in simple dollar terms. The implementation of an expert system to assist the technician provides an opportunity to significantly reduce these problems.

### **III. EXPERT SYSTEMS DEVELOPMENT CYCLE**

Several approaches for developing expert systems have been proposed in the literature. Prerau breaks the process into three main phases: Initial, Core Development, and Final Development and Deployment. (Prerau, 1990, p. 30) Waterman describes five stages of the expert system evolution process: Demonstration Prototype, Research Prototype, Field Prototype, Production Model and Commercial System. (Waterman, 1986, p. 130) Corrico, Girard, and Jones use a more traditional approach in describing eight stages in a knowledge system life cycle: Identification, Conceptualization, Formalization, Implementation, Testing, Evaluation, Maintenance, and Phase Out. (Corrico, 1989, p. 168) Since examining each methodology would be redundant, only the methodology presented by Prerau will be discussed in detail in the following sections.

#### **A. INITIAL PHASES**

The Initial Phase is comprised of three sub-areas: project start-up, selection of the domain, and selection of the development environment. Hardware and software studies are conducted and the ground work is laid during the initial phase. More importantly, managerial approval is granted for the project to continue and the project team is formed.

## **1. Project Start-Up**

During the project start-up the objectives should be understood by all parties involved. These objectives may vary widely and care should be exercised in attempting new technology and in delivering an operational expert system. According to Prerau, some possible objectives include:

- To "do something in artificial intelligence"; to "show that we're in AI."
- To learn about the expert systems field to "see what it can do for us."
- To educate the staff about expert systems technology.
- To encourage the spread of AI technology around the corporation.
- To build a flashy demonstration system to try to ensure additional funding.
- To build a small system for a small but real application with a payoff to the company.
- To develop a major expert system for a real application with a large payoff, either for use internally or as a product to sell.
- To study or discover expert system development techniques.
- To perform theoretical AI research. (Prerau, 1990, p.30)

Management approval is necessary in order for the project to continue. At times, management will be anxious for the project to continue and, in other instances, the project must be sold to gain support. Normally, with the approval of management, resources are obtained for the project including personnel and funding.

Next, the managerial and technical skills necessary to lead the project are identified and the team leader is selected. Team members are assigned by matching their talents to the tasks to be performed. Training may be needed when there is a shortfall between the skills and qualifications possessed by the project personnel and function complexity. Hiring additional experienced personnel may also be necessary to address any staff shortcomings.

A rough schedule is the last step of the start-up. It may be difficult to estimate the amount of work required until the scope of the project is examined in some depth. Schedulers can initially rely on past experience and refine the schedule as the project progresses. It is better to err on the conservative side when setting milestones, rather than setting optimistic completion times and risk falling behind schedule.

## **2. Selection of the Domain**

Domain Selection is the next step after the project start-up. Domain selection depends on the goals and scope of the project, as well as technical and non-technical considerations. Though the domain selection should not begin until the project starts, it is one of the most important aspects of the project. Accordingly, ample time and resources should be dedicated to this critical process. As the size and expense of a project increase, more effort should go in into this phase to decrease the risks involved.

### **3. Selection of the Development Environment**

Development environment refers to computers, engineering software tools, such as expert system shells, and programming languages, such as C, Lisp or ADA, used in the development of the expert system. Like domain selection, the development environment selection should be completed early in the project. Since technology changes so rapidly, each project team should research the current products to select the hardware and software best suited to the project's unique requirements. This, of course, may prove difficult to sell to upper management, especially if such resources are already in place. The selection of the development environment should be done after the domain has been selected, since domain parameters may affect the best choice of environment.

#### **B. CORE DEVELOPMENT PHASES**

The Core Development phases include a feasibility prototype and an operational prototype. Each prototype phase can be further broken down into three smaller subsections: knowledge acquisition, knowledge representation, and knowledge implementation. Knowledge acquisition is the process of acquiring the knowledge from the domain experts. Knowledge representation is the depiction of the acquired knowledge using one or more of the AI paradigms such as rules, frames, procedures, or object oriented programming. Knowledge implementation is the transformation

of the represented knowledge into an operational expert system program. (Prerau, 1990, p. 17) Knowledge acquisition, representation and implementation are discussed in detail in Chapters IV, V, and VI, respectively.

### **1. Development of a Feasibility Prototype System**

The first step in core development is the development of a feasibility prototype system. This is accomplished rapidly to determine whether the project should continue. While the feasibility prototype may or may not provide the framework for the operational prototype, it allows the project team to fine tune the knowledge acquisition, representation, and implementation processes. Non-essential functions may be added to impress the approving authority. Since it is not intended to become operational, input and output may be fictitious. The demonstration audience should be made aware that this is only a demonstration to illustrate potential features of the final expert system.

Some of the purposes of building a feasibility prototype include:

- It allows the project developers to get a good idea of whether it is feasible to attempt an operational prototype.
- It provides a method to study the effectiveness of the knowledge representation and implementation.
- It may disclose important gaps or problems in the proposed system.



- It yields a tangible product early in the development of the project.
- It gives an opportunity to impress management and program funding agents with a flashy demonstration.
- It gives an idea of what the final system will do and will look like.
- It allows the possibility of early course correction based on feedback.
- It provides a first system that can be field tested and, although not a final product, may be deployed on a limited basis. (Prerau, 1990, p. 39)

## **2. Development of a Full Prototype System**

If the feasibility prototype is well received and funding approved, development of the full prototype is the next step. The full prototype may be an expansion of the feasibility prototype or may incorporate the changes recommended during the feasibility demonstration. For smaller systems the final system may be produced from the feasibility prototype, but for larger systems producing a full prototype is recommended.

During the full prototype phase, knowledge acquisition and representation are further refined. Programming is also improved by writing cleaner, more efficient code. Special effort should go into program development during this phase, since much of the code will be used in the final program. Hardware and software selected in the initial phase should be reconsidered based on refinements of the system requirements. (Prerau, 1990, p. 43)

## **C. FINAL DEVELOPMENT AND DEPLOYMENT PHASES**

The final development and deployment phases represent the last steps in system development. Even after the system is deployed, system maintenance continues as information and procedures change and improved features are added. Typical ADP systems incur as much as seventy-five percent of their total life cycle costs during maintenance and system upgrades. Proper foresight during the developmental phases may reduce this expenditure. System sponsors and managers should be made aware of these continuing costs.

### **1. Development of a Production System**

Proceeding with the final production version depends on feedback from users, other experts, management, and an economic analysis of potential sales or savings derived from use of the system. During this phase a viable system is produced that can later be fielded.

Since most of the actual effort on the project occurs during the full prototype development phase, modification at this point is relatively inexpensive. Therefore, the representation and implementation of the knowledge base may be redesigned using completely different software without incurring excessive additional costs. (Prerau, 1990, p. 45)

Hardware decisions made early in the initial phases may be re-evaluated, since the deployment hardware need not necessarily be the same as the development hardware. Of course, compatibility of the software to

run on both systems is a major concern. Converting the software from one platform to another may be an expensive endeavor, both in terms of time and funds. (Prerau, 1990, p. 45)

As in the case of deployment hardware, deployment software need not be the same as development software. Again, compatibility with hardware is an important issue, as is the cost of software conversion. Conversion costs should be evaluated against potential benefits derived from the different software, such as reduced license fees, lower purchase cost, and enhanced performance. (Prerau, 1990, p. 46)

Other system elements should be considered at this stage. Communication interfaces and procedures and input/output formats and mechanisms should be evaluated for efficiency and rewritten or redesigned if needed. (Prerau, 1990, p. 46) Validation testing should be accomplished to determine whether the system addressed the problem for which it was designed. Verification tests ensure the system's expert knowledge has been captured and implemented correctly. Testing should be thorough, covering all possible cases. Documentation for users, maintainers, and systems managers must be written, either in printed manual format or on-line access. (Prerau, 1990, p. 47)

## **2. System Deployment**

Several factors remain to be considered for the deployment phase. The mode of deployment can either be a turn-key system, a separate entity

integrated into the existing user environment, or a service that remotely accesses data and delivers it back to the user. There are many possible variations regarding availability times, operating and maintenance responsibilities, number of user access channels, and service levels.

(Prerau, 1990, p. 48)

Pricing and marketing, though normally associated with commercial projects, are becoming more important given the military's trend toward costing. Pricing might be determined by the accessing costs associated with developing and maintaining the system or by the potential benefits to the system users. Marketing concerns making potential users aware of the system and selling the benefits associated with the use of the system. A major effort may be required when marketing commercial applications, while in-house marketing will usually be much easier. (Prerau, 1990, p. 48)

As the system is deployed, users must still be convinced to accept the new method of operation. While people are forgiving of errors in human experts, they are skeptical of machines that make mistakes. Just as knowledge experts may balk at the idea of expert systems, users may also fear the implementation of such systems for many of the same reasons. Part of the training program should be geared toward getting the user to trust and accept the system. Training may be by formal courses, instruction manuals, or on-line tutorials. (Prerau, 1990, p. 49)

#### **D. MK 92 MOD 2 MAINTENANCE ADVISOR EXPERT SYSTEM DEVELOPMENT CYCLE**

The development cycle used in the MK 92 MOD 2 Maintenance Advisor Expert system closely parallels the Prerau paradigm, terminating with development of a full prototype for evaluating DSOT performance parameters for FC1, FC2, FC4 and FC5 .

The initial phase began with a meeting between the Port Hueneme engineers and Naval Postgraduate School faculty and students. Objectives of the expert system were discussed and agreed upon, and overall system requirements were presented. It was agreed that the scope of the domain would encompass only the performance procedures for this first prototype. These requirements served as the basis for hardware and software selection. For example, a major requirement of the system is a friendly, easy-to-use graphical interface for use by the technicians on board ship.

Further requirements determined what hardware and software was available and would be best suited for the MK 92 MOD 2 Expert System. Additionally, different methods of knowledge acquisition, representation, and implementation were examined to best fit the application. This requirement dictated the use of a software tool with strong graphical display building capabilities.

A feasibility prototype was quickly developed and demonstrated to the MK 92 MOD 2 program management and system technicians at Port

Hueneme, along with a presentation of a plan of development and potential savings to be realized by implementing the system. This presentation was instrumental in convincing management to support further funding of the project and provided valuable feedback from the technicians and users on the layout of the screen graphics.

Shortly after the presentation, the domain for the full prototype was identified and the scope of the project was laid out. Implementation of FC1, FC2, FC4, and FC5 of the DSOT performance parameters were identified as the goal of the first increment, including all associated Help screens. It was also decided that DSOT calibration parameters evaluation, links to a database management system, and multi-media on-line help would be addressed in future prototype iterations.

The development environment was selected based on research and system requirements. Because of the need for a graphical user interface (GUI), a Windows based program was selected. Adept was selected as the developmental software because of the experience the Army had implementing the M1 tank diagnostic system and the ease by which programmers learn the software. A 486, Windows based PC was selected as the development hardware with future plans to implement the final program on a 486 notebook computer.

Knowledge acquisition and partial representation was accomplished at Port Hueneme as discussed in Chapter IV. Further knowledge

representation and implementation were accomplished incrementally at the Naval Postgraduate School. As each knowledge section was implemented, software developed and hard copy printouts of the knowledge based segment were sent to Port Hueneme for validation and verification. Errors were identified and sent back for correction before knowledge segments were combined into the overall system.

Additional demonstrations were given to the chief engineering officer at Port Hueneme and NAVSEA in Washington, D.C. in an effort to bolster support for the system in a time of shrinking budgets and scarce resources.

#### **IV. KNOWLEDGE ACQUISITION**

Knowledge acquisition is the gathering of information, decisions, heuristics, rules and relationships from any available source. From this collection of knowledge evolves the domain necessary to implement the expert system. Knowledge acquisition is generally regarded as the most complicated and involved phase of expert system development.

##### **A. KNOWLEDGE ACQUISITION PROCESS**

Knowledge acquisition traditionally involves interface between a knowledge acquirer and domain expert. This is an iterative process and may involve many different methods of knowledge gathering. Knowledge acquirers, typically, do not have expert status and may, in fact, know nothing of the concepts or terminology associated with the domain. In order to facilitate knowledge gathering, the expert should be able to communicate easily with people from diverse social and other backgrounds. More importantly, the expert should be able to take incomplete, sometimes fragmented, thoughts and represent them via one of several AI paradigms, such as production rules, procedures, frames, or objects.

Domain experts are those individuals who, through training and experience, have mastered a desired skill or task. (Turban, 1990, p. 434)



Experts, in addition to being the best representatives of the technical area, should have other attributes. They should possess good communication skills to impart their knowledge to others. They should be cooperative and eager to work on the assignment, even though the capturing of the knowledge base will be a drawn-out process. (Prerau, 1990, p. 178-179) Ideally, there should be one individual who possesses all the skills necessary upon which to base the expert system. This is often not the case. The knowledge acquirer may have to rely on other methods for acquiring the expert knowledge, such as personal interviews, personal notebooks, role playing, pictures or drawings, multiple experts, or questionnaires.

### **1. PERSONAL INTERVIEWS**

One-on-one interviews are one of the most common and effective methods of acquiring expert knowledge. There are drawbacks to this seemingly simple method. The expert may have some difficulty taking time away from his job to carry out interviews or have trouble verbalizing complex thought processes that, to him, are second nature. The interviewer should have an outline of the area to be covered before the session begins. This will allow an interviewer who may not be familiar enough with the subject to direct his questions in an orderly fashion. The interviewer should be careful not to overtax the expert during a single session. Time between interviews should be spent structuring the information already gathered for verification by the expert. (Corrico, Girard, Jones, 1989, p. 44)

## **2. PERSONAL NOTEBOOKS**

Notebooks are often an excellent source of information that cannot be matched by any other means. People often write down information they know they may not remember, especially if they know the information does not exist in a text or other document. New ideas, insights, tricks and pictures are but a few types of data that may be available for the asking. (Corrico, Girard, Jones, 1989, p. 47)

## **3. ROLE PLAYING**

Another unique method that may prove useful in situations where other approaches are not effective, is role playing. In effect, a game of 20-questions is played, with the knowledge acquirer asking the expert questions concerning the problem. Although this method may provide solutions for very specific problems, it should not be used routinely for problem solving because the process is too time consuming and agonizing. (Corrico, Girard, Jones, 1989, p. 47)

## **4. PICTURES OR DRAWINGS**

Experts frequently use pictures or drawings to maintain domain relationships. Visual representations may take the form of flow diagrams, charts and tables, or graphs. These visual aids may be useful for the knowledge acquirer to gain a better understanding of relationships and processes. Since the expert has already taken the time to document and

organize the data, knowledge acquirers should seek these out early in the acquisition process. The domain expert should explain the formulas used to graph data, as well as implicit knowledge common to the expert, in order to correctly translate the knowledge to the acquirer. (Corrico, Girard, Jones, 1989, p. 49)

## **5. MULTIPLE EXPERTS**

There may be instances where more than one expert provides expert knowledge. For example, where the proposed system overlaps into another expert's area of expertise, or, where a single expert is not available for the length of time required by the project. Although it is preferable to use multiple experts, it does create certain problems. Experts, of course, may not always agree on the best method of accomplishing a task. There are two rules of thumb to use in such instances. When two experts disagree, one must be considered "the" expert. That expert's opinion should overrule the other every time. When three or more experts are used, the consensus approach should be employed. No one expert should be allowed to override the majority of their peers. (Corrico, Girard, Jones, 1989, p. 51)

## **6. QUESTIONNAIRES**

A final knowledge acquisition technique is the use of questionnaires. This method is useful when experts are widely dispersed and their responses can be used as part of the acquisition process or as

verification of the knowledge domain provided by other experts. (Corrico, Girard, Jones, 1989, p. 51)

## **B. MK 92 MOD 2 KNOWLEDGE ACQUISITION**

A relatively unique method was employed in acquiring the knowledge for the Prototype Maintenance Advisor Expert System for the MK 92 MOD 2 Fire Control System. The MK 92 MOD 2 Department at Port Huememe contracted with the Paramax Company for the services of one of their technicians. His mission was to document the steps an expert would take in diagnosing the faults associated with all possible DSOT NOGO's. Relying on his years of experience on the MK 92 MOD 2 FCS, the MK 92 MOD 2 technician manuals, NAVSEA, Paramax, and Navy Engineers, the expert began documenting the knowledge in the form of diagnostic trees.

Because of the expanse and complexity of the MK 92 MOD 2 FCS, the traditional methods of using a knowledge engineer to acquire the domain would have been extremely time consuming. Using students as knowledge acquirers/engineers was also not a feasible option because travel time required missing classes. Having a knowledge engineer unfamiliar with the MK 92 MOD 2 FCS acquire the knowledge from domain experts would, unquestionably, have been very time consuming and perhaps economically unfeasible.

The product of the expert's effort was a series of diagnostic tree diagrams for the DSOT performance channels: FC1, FC2, FC4, and FC5. Additionally, the expert included, as part of the diagnostic trees, textual help to assist the maintenance technicians with difficult procedures. These textual helps may reference sections of the manuals, provide procedural assistance or denote warnings and cautions where necessary.

The expert played the role of knowledge acquirer. The diagnostic tree diagrams developed by the expert represented the acquired knowledge. Further, the knowledge representation paradigm chosen closely matched the diagnostic tree diagrams, thus greatly facilitating the knowledge representation process. This process is discussed in greater detail in the following chapter.

## **V. KNOWLEDGE REPRESENTATION**

Knowledge acquisition is concerned with the gathering of knowledge from experts. Knowledge representation is concerned with how knowledge is illustrated. Structuring tools are needed to capture, illustrate, and inspect the information so that it can be implemented in an expert system. While paradigms describe the way people use or process their knowledge, representation supplies the details of a specific domain of knowledge. (Corrico, 1989, pp.61-62)

### **A. KNOWLEDGE REPRESENTATION METHODS**

There are a number of methods of representing knowledge. They include production rules, frames, semantic nets, procedures, and logic. Each method has its own advantages and disadvantages. An expert system may incorporate multiple representations to better depict the domain. The choice of a particular representation is influenced by the application domain. A knowledge representation method is selected to represent, as naturally as possible, the application domain. The following sections discuss four of the most widely used representation models: production rules, frames, semantic nets, and procedures.

## 1. Production Rules

The most popular representation technique is rules, sometimes called production rules. This strategy is most appropriate in domains where experts have developed associations in the domain through years of experience. Rules are simply a series of IF-THEN statements checked against a series of given facts about the particular situation. When the IF portion of the statement is true, the THEN portion is executed. When the IF is false, the program branches to another IF or ELSE statement. (Waterman, 1986, p.63) Rules can be described as condition/action, where the program gets information about the status of the environment and then provides the appropriate response.

An example of a production rule is the following:

---

*IF a DC voltage is not present at output of  
the power amplifier,  
THEN replace train drive motor  
ELSE continue troubleshooting procedures.*

---

Rules offer several advantages in knowledge representation. There is a high degree of correspondence between acquisition rules and implementation rules, making programming and maintenance easier. Because rules can be written in simple terms, they are easier to program than other methods, such as programming language code. Rules also tend to be modular, thereby making software maintenance easier. Instead of changing multiple lines of code the maintainer can simply change the affected rule. (Prerau, 1990, pp.254-255)

The execution of rules is accomplished by a process called chaining. Chaining may be classified as backward-chaining or forward-chaining. Backward-chaining is a goal driven approach. In backward-chaining the program identifies the result hypothesis and attempts to assert the facts of all rules having that hypothesis as the end result. It is often necessary to test intermediate or sub-hypotheses before the correct conclusion rule can be identified. (Walters, 1988, p. 196) In contrast, forward-chaining is a data driven approach. As information becomes available the program attempts to draw all possible conclusions.

## **2. Frames**

Frames are data structures that hold various types of knowledge. The best analogy is to that of a data record used in programming languages such as Ada, Pascal, or PL/1. Frames can represent physical objects or ideas.

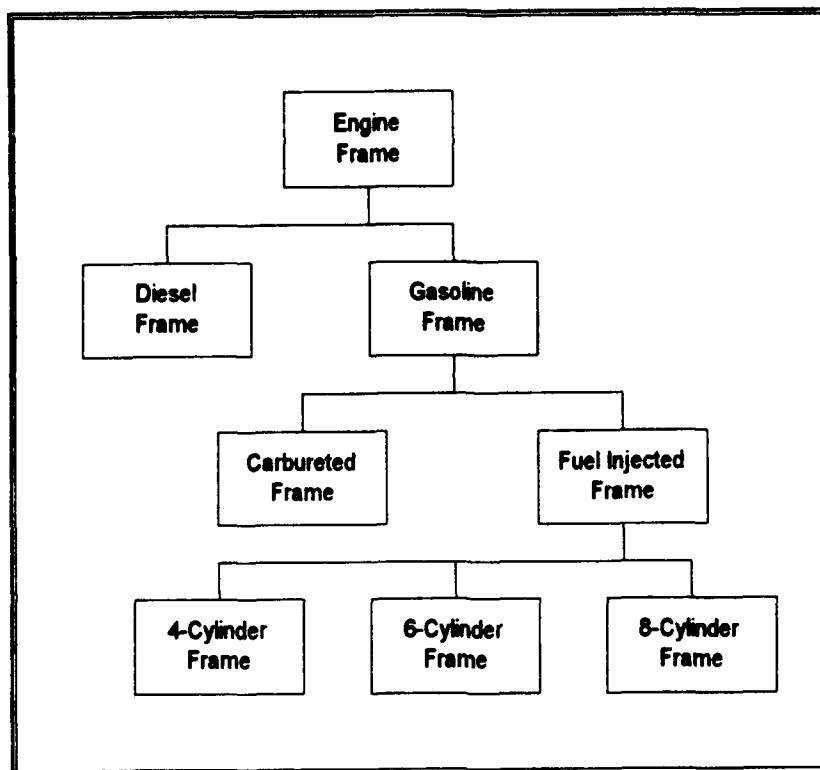


Frames describe characteristics, properties, and behavior of an item. (Walters, 1988, p.210)

Slots provide the internal storage arrangement for frames. Slots can be broken down into sub-slots or facets. For example, in describing a person frame, Jane is a slot, and age and occupation are facets. Though this is a simple example, knowledge representations may be as complex as necessary with frames serving as slots and facets to represent multi-layered structures.

Figure 5-1 depicts a hierarchical set of frames describing knowledge about engines. Figure 5-2 illustrates slots associated with car frames. These slots could, in turn, be frames or facets.

Though reasoning through frames is more complex than reasoning through rules, frames offer several advantages in representing knowledge. Frames provide a relatively simple method of storing and retrieving data. Because frames are hierarchical in nature, relationships can be inherited from other frames. Thus, the data structure need not be reinvented for multiple items. Searches of the knowledge base are faster using the frame structure because of the exact representation of data. Finally, psychologists believe that experts recall information about objects as a group, closely resembling the frame structure. (Badiru, 1992, p.81-82)



**Figure 5-1 HIERARCHY OF FRAMES DESCRIBING ENGINES**

Chevrolet		Ford	
Body Type		Body Type	
Tires		Tires	
Engine Type		Engine Type	
Battery Type		Battery Type	

**Figure 5-2 KNOWLEDGE ACCESS BY FRAME AND SLOT**

## **B. SEMANTIC NETWORKS**

A semantic network is a knowledge representation based on a network structure. Developed to model human intelligence, semantic networks have applications in AI and expert systems. Semantic networks are comprised of nodes connected by links (arcs). Nodes represent objects, ideas, or events. Arcs describe the relationship between nodes. Two common arc examples are "isa" and "hasa" for "is a" and "has a". The use of these types of links is to show the inheritance hierarchy in the net. The lower object in the net can have the same properties as those higher in the net, saving space in the program since the structure does not need to be repeated. (Waterman, 1986, p.70)

Semantic networks are useful in representing knowledge domains with well-defined characteristics, such as decision trees and tables. The primary advantages of semantic networks are inheritance and flexibility. As shown in Figure 5-3, Jane is a mammal and thus inherits the characteristics of all mammals. This ability to take on the characteristics of other related nodes is very useful in describing knowledge.

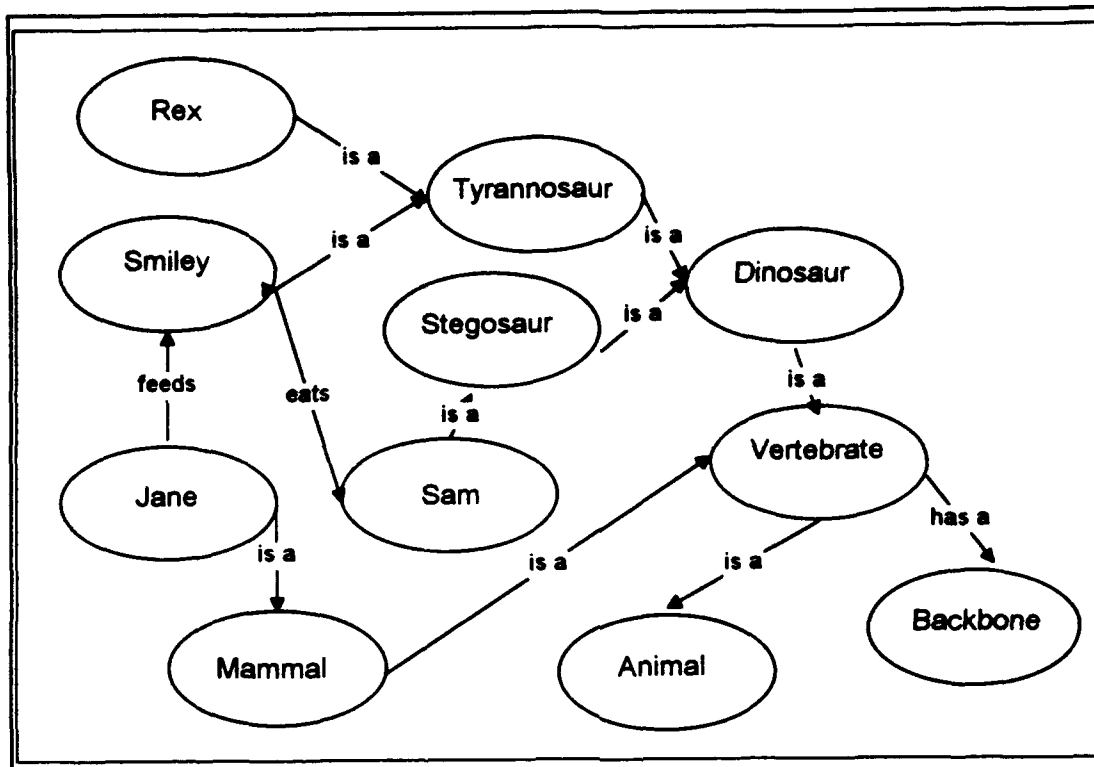


Figure 5-3. SEMANTIC NET

### C. PROCEDURES

Procedural representations provide a method of chaining conditions that represent the domain. Each condition must be unique and used by only that rule for which it was intended.

An example best illustrates this point. Suppose a car will not start. A mechanic may formulate a procedure to arrive at the source of the malfunction, as shown in the example below. Answering no to any of these

questions would lead the mechanic down a sub-procedure to investigate and remedy the situation.

---

**Check electrical system**

- *Does engine turn over?*
- *Does horn sound?*
- *Do lights illuminate?*
- *Do spark plugs fire?*

**Check fuel system**

- *Is there fuel in the tank?*
  - *Is there fuel at the carburetor?*
  - *Is there fuel in the cylinder?*
  - *Is the fuel mixture correct?*
- 

Procedure representations are useful in crafting knowledge for use in diagnostic and production systems. They offer the advantage of modularity in programming, in that each procedure may be constructed individually. This improves system maintainability, as well as coding and debugging ease. However, the procedure based system itself must be considered as a whole. That is, without a given procedure or sub-procedure, the system may not provide correct results. (Georgeff, 1986, pp. 16-18)

#### **D. MK 92 MOD 2 KNOWLEDGE REPRESENTATION**

The logical choice for domain representation for the MK 92 MOD 2 Maintenance Advisor Expert System is procedures. By nature, a diagnostic tree is a procedure that leads a technician through a series of questions based on equipment status to arrive at suggested fault remedies. Well-constructed procedural representations offer flexibility and lead to easy conversion to the knowledge implementation phase. Procedures can be treated modularly and can be added, modified, and deleted as the knowledge domain dictates.

Since DSOT provides the technician with NOGO's that indicate sub-areas of the system that are faulty, the use of a procedure based system allows trouble-shooting, as necessary, in any FC section. The program was constructed to allow entry to any FC and sub-area. Easy access to another FC without exiting the program is made possible with options to return to preceding menus.

Procedure modularity was very useful in the development of the system as it allowed multiple programmers to work simultaneously on different areas. It also allowed each section to be returned to the knowledge experts at Port Hueneme and verified before assimilation into the main program.

One of the critical features of procedural expert systems is that the knowledge is represented in well defined semantics. (Georgeff, 1986, p. 60)

The knowledge represented by the diagnostic procedures supplied by Port Hueneme engineers was well designed and thought out. This made the representation of the knowledge by the programmers much easier and ultimately resulted in a better expert system.

## **VI. KNOWLEDGE IMPLEMENTATION**

Implementing an expert system differs from implementing a conventional program. Specifications for a traditional program are usually complete before programming begins. Specification and implementation for an expert system evolve together. Thus, instead of using a top-down approach, the process tends to be iterative. Segments of the knowledge are programmed separately, then linked together modularly. (Prerau, 1990, pp. 266-267)

The actual programming of an expert system is very much like that of a conventional system in the area of programmer experience. Therefore, it is best to allow the implementors to work with the developmental environment as soon as possible to increase their proficiency. (Prerau, 1990, p. 276)

### **A. KNOWLEDGE ACQUISITION PROCEDURES AND IMPLEMENTATION**

It is evident that there should be a close correspondence between knowledge acquisition procedures and implementation procedures. To make coding easier to follow, the language used in the acquisition procedures should match the language used in the implementation



procedures. This one-to-one correspondence not only aids in development, but assists program maintenance as well. (Prerau, 1990, p. 277)

## **B. IMPLEMENTING AND DEBUGGING**

As expected, debugging an expert system differs from debugging a conventional system. Each module of a conventional system has its own specifications and can be tested independently before it is incorporated into the main program. The same is not true of an expert system. The expert system must be built and debugged incrementally. (Prerau, 1990, p. 279)

Because knowledge acquisition continues throughout the development of the expert system, specifications are constantly evolving. Thus, it may be necessary to modify the program even before coding is completed.

Programmers can usually debug a conventional program by running test case input and arriving at anticipated output. The expert system debugging presents a different problem. Not only must the program yield correct results in respect to the knowledge domain, but, the domain must also be checked for inaccuracies by a knowledge expert. (Prerau, 1990, p. 279)

## **C. DOCUMENTATION**

Just as in conventional programs, documentation is an important part of implementation. Because documenting is not a task most programmers

enjoy, special attention should be given to ensuring that it is done correctly. Expert systems require documentation of both the knowledge domain and the program. Standard features such as input, output, and module purpose should be recorded. Matching the knowledge representation to the implementation by using rule correspondence, naming conventions, and specific references may make the documentation more complete and easier to follow. (Prerau, 1990, p. 280)

#### **D. UNIFORMITY OF STYLE**

In order to ensure that programming style and display screens are uniform, pre-programming conventions should be agreed upon before any coding begins. For a visual programming environment, conventions should address logic flow techniques, such as case handling, *off page connections*, and location of controls and text on the display screen. Conventions enable several programmers to work on the project simultaneously. They should, however, be rigorously enforced to ensure compliance.

#### **E. VALIDATION AND VERIFICATION**

Validation and verification are two important aspects of system evaluation. Validation examines whether the right system was built and whether or not the system will operate at a given level of performance. Verification refers to examining whether the system was built correctly, that

is, whether the system matches the documented expert knowledge. (Prerau, 1990. p.300)

Expert systems development, as described in the preceding chapters, is an iterative process. Therefore, validation and verification testing is completed during each phase of system development.

Validation and verification of the MK 92 MOD 2 Maintenance Advisor Expert System was a unique process. As each procedure was programmed, it was sent to the domain expert for evaluation. This process ensured that the knowledge implementation matched the expert's knowledge representation procedural tree diagrams in both logic flow and wording.

The use of an expert shell programming tool, such as Adept, greatly enhanced the verification effort. Rather than worrying about thousands of lines of code in a programming language such as Lisp or Prolog, the builders only had to concern themselves with correctly matching the expert's representations.

#### **F. MK 92 MOD 2 MAINTENANCE ADVISOR EXPERT SYSTEM PROCEDURE BUILDER ISSUES**

The project's selected knowledge tool uses a graphical tool set to construct individual procedures that define the skeleton, or framework, of an application. The procedures are also "linked", a process that enables the procedures to work together in solving problems, by the tool's graphics set.

Graphical representations and descriptions of the MK 92 MOD 2 Maintenance Advisor Expert System procedures are listed in Appendix A. Each procedure is described in detail including purpose, calling procedure, procedures called, and logic flow relationships of the procedures. The procedures included are FC-1 Designation--Time, Range, and Bearing; FC-1 Acquisition; FC-2 Designation--Time, Range, and Bearing; FC2 Acquisition; FC4; and FC5. The procedures have been implemented as nearly as possible to emulate the expert's original knowledge form to promote future enhancements and simplify maintenance of the system's knowledge base.

#### **G. DISPLAY BUILDER ISSUES**

A display is a collection of graphical objects (i.e., buttons, text fields, and list boxes) that receive information from the user to complete a procedure or present results and instructions. (Himes, 1991, pp. 14) The developmental software provides a comprehensive toolbox that automatically constructs a default display each time the application logic requires user interface. The display builder enables the user to customize the default screen into unique and functional displays. Display builder issues discussed will focus on screen layout, colors, conventions, and fonts. and graphics.

## **1. Screen Layout**

The standard display screen is divided into the Main Title Bar, Procedure Box and Action Box.

### ***a. Main Title Bar***

The title bar, as shown in Section A of Figure 6.1, is located at the top of the display screen. It contains the procedure title, DSOT firing channel NOGO, subtitle, and trouble-shooting location. The variance to this scheme lies in the main and function menus, where only the procedure title is displayed. This section continuously depicts which DSOT NOGO is being evaluated and the location within that NOGO's diagnostic tree.

### ***b. Procedure Box***

The procedure box, as shown in Section B of Figure 6.1, is located in the middle of the display screen. The content of the box varies with each screen, but generally, it contains bitmap objects, procedure and help text, and labeled pushbuttons.

The procedure box is where the expert system requires the user to perform a task, or tasks, and respond to queries. This enables the system to continue its problem diagnosis.

### ***c. Action Box***

The action box, as shown in Section C of Figure 6.1, is located at the bottom of the display screen. This section contains pushbuttons that

enable the user to interact with the expert system. The number of buttons vary with each display screen depending on procedure requirements. Generally, each action box has yes, no, and help buttons. Button properties also vary, depending on procedure requirements. In most situations, yes equates to true, no to false, and help to unknown. Menu selection buttons are also located in the action box. This enables the user to select which procedures he wishes to trouble-shoot or select specific cases based on equipment status or test indication.

## **2. Colors**

The choice of display screen color is a rather difficult task. First, it is important that the chosen colors be complimentary, yet provide enough contrast to be distinctive to the eye. Second, the colors should be soft, but bright enough for the eye to distinguish individual characteristics. The project's selected tool, Adept, includes a color palette of several available colors. The palette enables the user to differentiate between border and fill colors. Also, shading of any selected color is possible through the tool's color editor. It is important for developers to keep in mind that pleasing all users is next to impossible, so they should choose a design and make it standard throughout the application.

The color scheme in this application is divided into background and foreground. A background layout is maintained for all displays, while a foreground layout varies from one display to another.

***a. Background***

Background colors were chosen to be appealing to the eye, yet not overpowering. Sufficient contrast was added to separate the different sections of the display, while still allowing a smooth transition from one section to another. The colors were applied in layers, with the first color applied being the lowest layer. The chosen colors are navy for the overall background, dark green for procedure and action box backgrounds, blue for procedure and action box title bars, aqua for procedure and action box title names, blue green for procedure and action boxes, and soft yellow for menu title bars.

***b. Foreground***

As indicated, the foreground colors are procedure specific. For example, a procedure might have a note associated with one of its diagnostic steps. If so, the "notes" appear on the display screen in blue. The color blue provides sufficient contrast to the blue green color of the procedure box, so it catches the user's eye. Warnings appear in red, bordered in white, while Cautions appear in yellow, also bordered in white. These are standard safety colors, which provide a stark contrast to the surrounding colors, and the user's eye will recognize them as such.

### **3. Conventions**

Screen conventions are important to application standardization. Essentially, conventions are the rules that knowledge implementors must follow when building the individual modules and procedures that make up the expert system. The conventions discussed are naming and screen.

#### ***a. Naming Conventions***

These conventions standardize the labels that are applied to system procedures, pushbuttons, and title bars. An important aspect of naming conventions is the requirement that applied labels be sufficiently unique within separate procedures to prevent logic overlaps and errors during application integration. The naming convention for help pushbuttons covered two different situations. The first involved single help screens with pushbuttons labeled "Return" (returns to DSOT). The second involved multiple help screens with pushbuttons labeled "Return" (returns to DSOT), "Previous" (returns to the previous screen) or "Continue" (continues help), and possibly "Information" (provides explanatory data). A special situation involved help screens that specifically referred to additional help screens by letter. The special help pushbuttons are labeled "Help X" (X equates to the letter assigned).



### ***b. Screen Conventions***

Screen conventions provide standardization on the location of items within each procedure display section. Essentially, the standard screen, as shown in Figure 6.1, becomes a template for the entire expert system development. Information varies, but its location generally remains the same. For example, the "Help" pushbutton usually resides in the Action Box. However, due to the number of sub-procedure pushbuttons in a menu procedure, the "Help" pushbutton may be re-located to the Procedure Box.

Procedure conclusion screens require a separate convention based on single or multiple recommendations. Single recommendations conclude with "Recommend Replacing", while multiple recommendations conclude with "Fault Not Isolated to a Single Card Failure. Recommended Replacement Order is: . . .".

Additionally, Adept can run in either a VGA or SVGA display mode. Either format is usable, however, it is important that multiple-team development occur in the same display mode.

### **4. Fonts**

Wherever possible, the standard application text used was MS Sans Serif (font), bold (font style), 12 (font size), and black (font color), as shown in Figure 6.1. Exceptions to the standard were the use of a 10 point font to fit large amounts of text into a procedure box, title bar heading, excluding "title only" menus, and the Procedure and Action box title bar, which also

substituted aqua for black, as the font color. Additionally, "Warning and Caution" display screens use a 24 point font in the title, and a 14 point font in the text body.

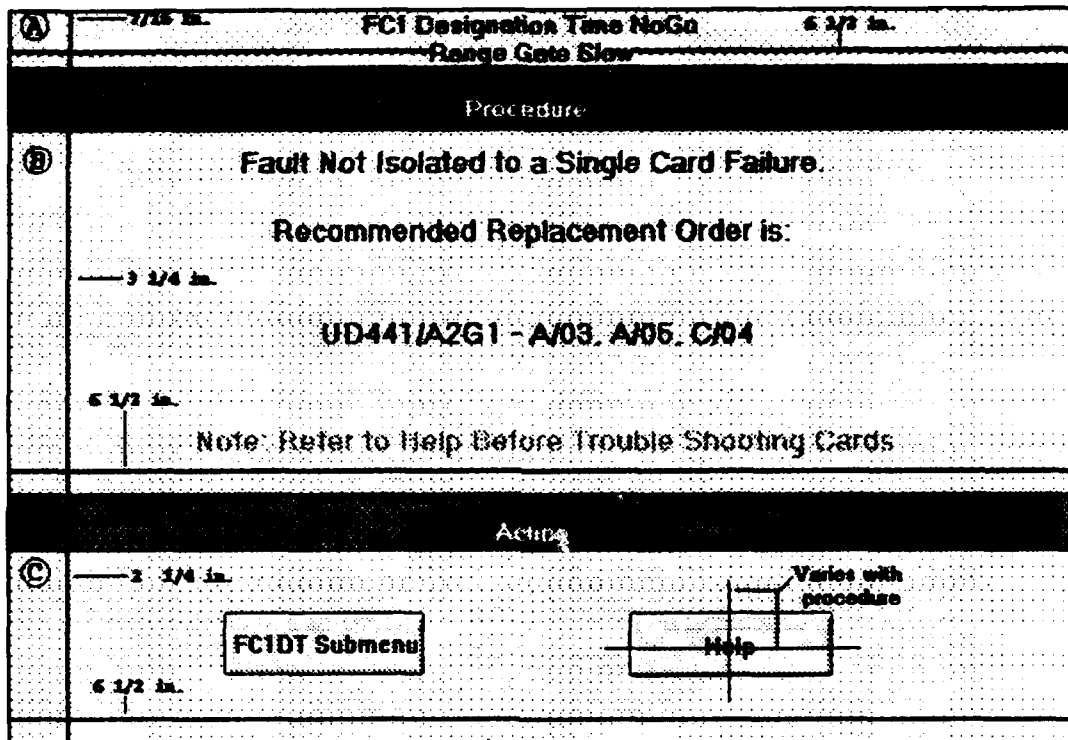


Figure 6-1 FCS MK 92 MOD 2 Display Screen

## **VII. LESSONS LEARNED**

**This chapter documents the experience gained through developing the MK 92 MOD 2 Maintenance Advisor Expert System Prototype. Many of the points mentioned below were not fully developed in the references, and were certainly not fully appreciated until actually encountered first hand.**

### **A. KNOWLEDGE EXPERT AS KNOWLEDGE ACQUIRER**

**The most unique aspect of the development of the MK 92 MOD 2 Maintenance Advisor Expert system was the method of knowledge acquisition. As mentioned in preceding chapters, this was the key to the success of the project and greatly reduced the time required to complete the system. Though the Paramax engineer who acted as the knowledge acquirer had no formal training in knowledge engineering, he represented the trouble-shooting steps using logical, easy to follow diagnostic tree diagrams. These diagrams were later represented in a straightforward manner to divide the domain into logical procedures and implemented using a procedure-based expert system shell. Employing more traditional methods of knowledge acquisition in the case of the MK 92 MOD 2 system would have been expensive and very time consuming.**

## **B. PROCEDURES PARADIGM**

Procedure representation was the logical paradigm for the MK 92 MOD 2 knowledge domain. The inherent nature of diagnostic systems is that of procedures. The diagnostic tree diagrams converted easily into procedures with an almost one-to-one correspondence to implementation.

## **C. EXPERT SYSTEM TOOLS--ADEPT**

There are numerous tools available on the commercial market to assist in building expert systems. There are also a number of conventional languages such as Lisp and C that could be used in coding such systems. Adept, by the Symbolic Corporation, was the software selected to develop the MK 92 MOD 2 Maintenance Advisor Expert System for several reasons.

First, Adept is a procedure based expert system that implements diagnostic procedures in a straightforward manner. For example, multi-path divergence of the knowledge expert's tree diagrams were easily converted into case nodes in Adept. Yes and no responses to troubleshooting questions were paralleled by the arcs connecting the procedures in the Adept program. The use of Adept's "Goal" function allowed programming instances of DSOT multiple NOGO situations.

Second, Adept proved to be easy to learn. Symbolic included a useful tutorial that took the user through a series of lessons geared to develop a working knowledge of the program. Though questions about the

program often arose, they were usually quickly answered by Symbologic's product support staff.

Other software considered required special training by the vendors to become proficient with their tools. Needless to say, this training was not free and greatly increased the price of utilizing their specific applications.

Symbologic charges for each application of the development program. Included in the purchase price is a run-time version of the software that allows user built programs to operate independently of the development program. Under the licensing agreement, Symbologic does not limit the number of run-time versions developers may implement. Other vendors require purchase of a separate run-time version for each expert system fielded. Had this been the case for the MK 92 MOD 2 Maintenance Advisor Expert System, implementation costs would have been much greater.

Another important advantage of Adept is that it combines a procedure builder and a display builder in one package. Some development tools use one program for developing the knowledge base and a separate display builder for building user screens. In addition to being awkward, extra time is necessary to learn two programs, as well as to integrate the output of the two packages.

Adept affords a wide variety of graphical options. Builders can import bitmap format graphics for display in a variety of presentations. For example, the first screen of the MK 92 MOD 2 Maintenance Advisor Expert

System is a title screen with a picture of a FFG 7 class ship in the background. In addition, other graphics/images can be built or modified using a variety of applications. Paintbrush was used by the programmers to create a card extender diagram used in one of the Help displays. A variety of other programs could be used to create and import illustrations into the application.

Adept offers many features for text insertion and editing. The program allows creation of text boxes of any size and offers a variety of font sizes and types. Special characters not resident in the Adept library could be imported from other Windows applications if needed.

The Adept displays are built on a background design that needs to be defined once. While generally useful, there is a drawback to this concept, since the background appears on every display screen throughout the program. The builder must carefully plan a background design that will serve the entire application.

Arrangement of objects on the displays was enhanced by Adept's click and drag feature. Object placement on the screen was aided by a snap option that could be adjusted by the builder from a very fine to a more coarse setting, depending on preference, to maintain a consistent look for all displays. Another useful approach used by the programmers to ensure consistency was simply to copy the entire display over and change only the necessary objects. In fact, the copy feature was very useful in procedure

building, as well as in screen building. Often, procedures were very similar in logic and verbage. For example, FC1 Designation Range is very similar to FC2 Designation Range. Instead of starting from scratch and building an entirely new procedure, the builder only had to create a new procedure shell and copy the old procedure into the new shell using the copy and paste options provided by Adept and then incorporate any necessary modifications into the new procedure. This shortcut alone saved untold hours of programming.

#### **D. SELLING THE PRODUCT**

As addressed in Chapter III, obtaining the support of management is essential to the success of system development. Without management approval it is impossible to continue with the project. This, too, was the case for the MK 92 MOD 2 system. Management wanted to see results and potential system benefits before committing scarce funds to project development. Feasibility prototype demonstrations provided the best forum for demonstrating system capabilities. Demonstrations provided to management at Port Hueneme Division, Naval Surface Warfare Center resulted in continued project funding through fiscal year 1993. A presentation to the Fleet Training Center, San Diego, provided valuable feedback from MK 92 Mod 2 FCS technicians and strengthened the acceptance and support of the program. Finally, a demonstration to the

program sponsor at Naval Sea Systems Command, Washington, D.C., resulted in continued project funding for fiscal year 1994.

Project developers must remember that selling the project to management, funding agencies and, eventually, the end user is essential for continued project development. Selling the project may take the form of presentations, prototype demonstrations, technical reports, meetings, or phone calls. The bottom line is that developers must learn the skills to be marketing and sales agents for their project.



## **APPENDIX A**

### **PROCEDURE FUNCTION DESCRIPTIONS AND DIAGRAMS**

#### **A. MAIN MENU**

**Name:** Main Menu (Figure 1)

**Number:** 0

**Description:** The first menu in the program. Allows selection of the Performance or Calibration portions of the diagnostic program or exits the program.

**Called by:** Starting the program (the first screen the operator sees is an FFG 7 class ship with system developer information and a "CONTINUE" button to start the program.

**Calls:** Performance and Calibration Menus

---

**Name:** Performance Menu

**Number:** 1.0

**Description:** Allows the selection of FC1, FC2, or FC4 and FC5.

**Called by:** Main Menu

**Calls:** FC1, FC2, or FC4 and FC5

**Name:** Calibration Menu

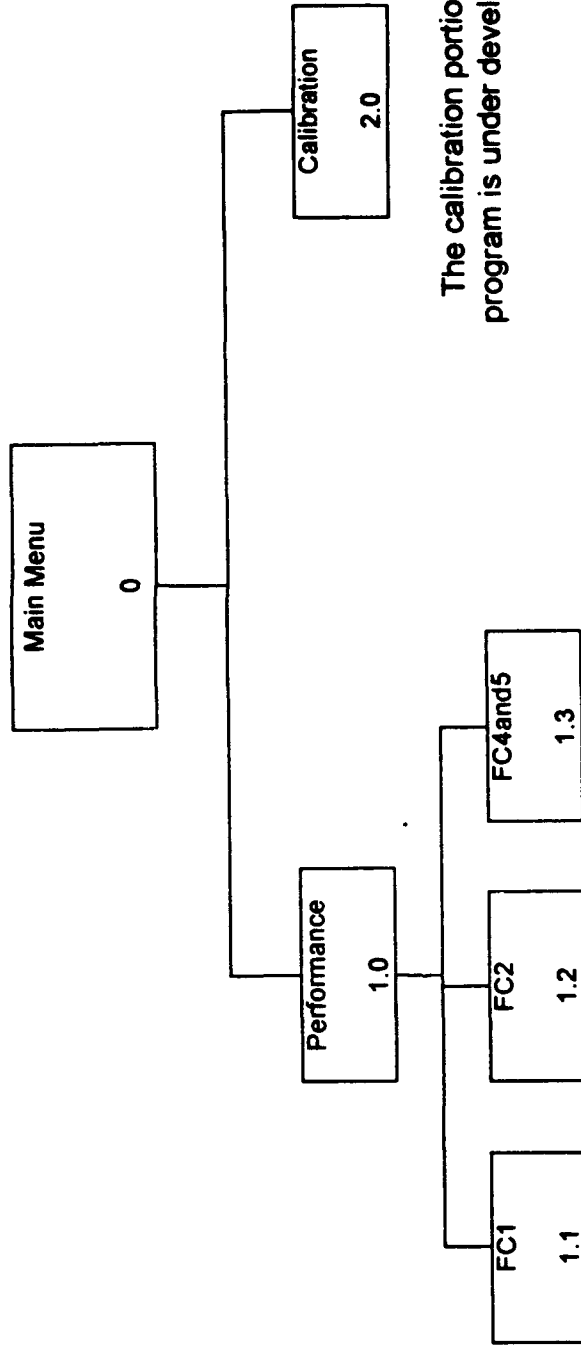
**Number:** 2.0

**Description:** Allows selection of Calibration procedures.

**Called by:** Main Menu

**Calls:** (The Calibration procedures are under development.)

## Main Menu



The calibration portion of the program is under developed.

FIGURE 1

## **B. FC1 MENU**

**Name:** FC1 Menu (Figure 2)

**Number:** 1.1

**Description:** Allows selection of FC1 Designation--Time, Range, and Bearing. FC1 ACQ. FC1 Track--Bearing, Elevation, and Range.

**Called by:** FC1 DTRB Menu

**Calls:** FC1 DTRB, FC1 ACQ, FC1 TBER

---

**Name:** FC1 DTRB Menu

**Number:** 1.1.1

**Description:** Allows selection of FC1 Designation--Time, Range, and Bearing procedures.

**Called by:** FC1 Menu

**Calls:** FC1 DT, FC1 TR, and FC1 TB

---

**Name:** FC1 ACQ

**Number:** 1.1.2

**Description:** Allows selection of FC1 ACQ procedures.

**Called by:** FC1 Menu

**Calls:** See FC1 ACQ Menu

**Name:** FC1 TBER Menu

**Number:** 1.1.3

**Description:** Allows selection of FC1 Track--Bearing Elevation and Range procedures.

**Called by:** FC1 Menu

**Ca's:** FC1 TB, FC1 TE, and FC1 TR

## FC1 Menu

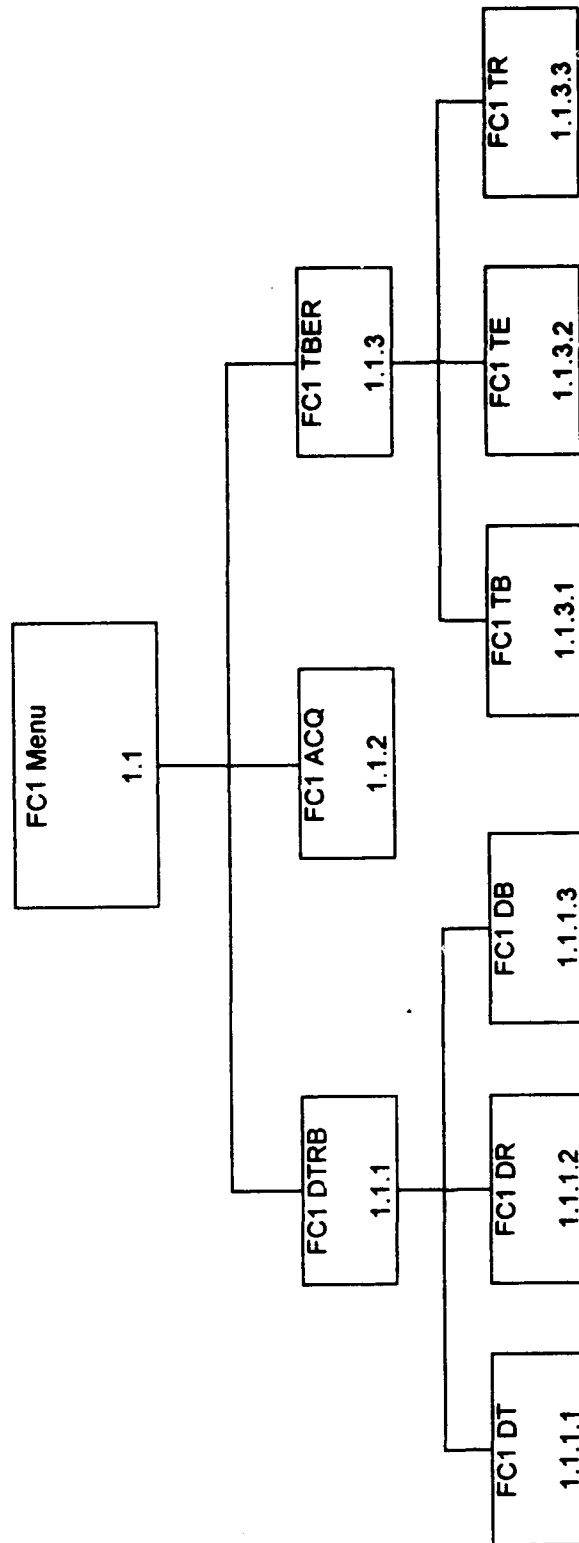


FIGURE 2

### **C. FC1 DT**

**Name:** FC1 DT (Figure 3)

**Number:** 1.1.1.1

**Description:** Allows selection of three FC1 DT cases.  
Case 1--Range Gate approximately 25K yards. Range Reading on TOTE equals zero or is less than 24K yards or greater than 26K yards.  
Case 2--Range Gate approximately 25K yards. Range Reading on TOTE approximately 25K yards.  
Case 3--Range Gate not present or nowhere near 25K yards.

**Called by:** FC1 Menu

**Calls:** FC1 DT Case 1, FC1 DT Case 2, and FC1 DT Case 3

---

**Name:** FC1 DT Case 1

**Number:** 1.1.1.1.1

**Description:** Allows trouble-shooting of FC1 DT Case 1 procedures.

**Called by:** FC1 DT Menu

**Calls:** FC1 DT Case 1A

---

**Name:** FC1 DT Case 1A

**Number:** 1.1.1.1.1.1

**Description:** Continues trouble-shooting of FC1 DT Case 1 procedures.

**Called by:** FC1 DT Case 1

**Calls:** None

**Name:** FC1 DT Case 2  
**Number:** 1.1.1.1.2  
**Description:** Allows trouble-shooting of FC1 DT Case 2 procedures.  
**Called by:** FC1 DT Menu  
**Calls:** FC1 DT--No Track Antenna Movement, Track Antenna Slow, No Range Gate Movement, Range Gate Slow, Both No Movement, and Both Slow.

---

**Name:** FC1 DT No Track Antenna Movement  
**Number:** 1.1.1.1.2.1  
**Description:** Allows trouble-shooting of FC1 DT No Track Antenna procedures.  
**Called by:** FC1 DT Case 2  
**Calls:** FC1 DT No Track Antenna Movement A

---

**Name:** FC1 DT No Track Antenna Movement A  
**Number:** 1.1.1.2.1.1  
**Description:** Continues trouble-shooting of FC1 DT No Track Antenna procedures.  
**Called by:** FC1 DT No Track Antenna Movement  
**Calls:** None



**Name:** FC1 DT Track Antenna Slow  
**Number:** 1.1.1.1.2.2  
**Description:** Allows trouble-shooting of FC1 DT Track Antenna Slow procedures.  
**Called by:** FC1 DT Case 2  
**Calls:** None

---

**Name:** FC1 DT No Range Gate Movement  
**Number:** 1.1.1.1.2.3  
**Description:** Allows trouble-shooting of FC1 DT No Range Gate Movement procedures.  
**Called by:** FC1 DT Case 2  
**Calls:** None

---

**Name:** FC1 DT Range Gate Slow  
**Number:** 1.1.1.1.2.4  
**Description:** Allows trouble-shooting of FC1 DT Range Gate Slow procedures.  
**Called by:** FC1 DT Case 2  
**Calls:** None

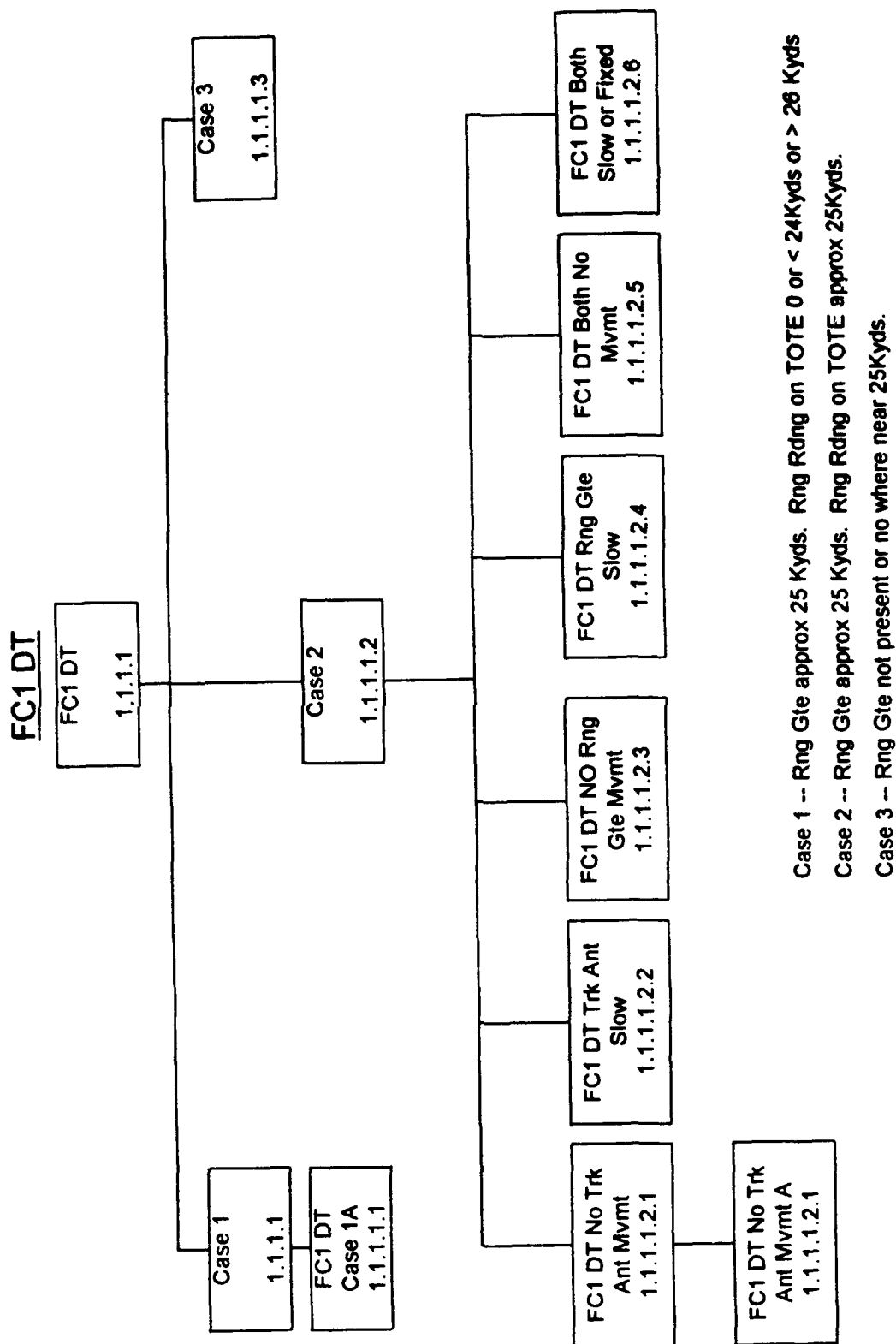
**Name:** FC1 DT Both No Movement  
**Number:** 1.1.1.1.2.5  
**Description:** Allows trouble-shooting of FC1 DT Both No Movement procedures.  
**Called by:** FC1 DT Case 2  
**Calls:** None

---

**Name:** FC1 DT Both Slow  
**Number:** 1.1.1.1.2.6  
**Description:** Allows trouble-shooting of FC1 DT Both Slow procedures.  
**Called by:** FC1 DT Case 2  
**Calls:** None

---

**Name:** FC1 DT Case 3  
**Number:** 1.1.1.1.3  
**Description:** Allows trouble-shooting of FC1 DT Case 3 procedures.  
**Called by:** FC1 DT Menu  
**Calls:** None



**FIGURE 3**

#### **D. FC1 DR**

**Name:** FC1 DR (Figure 4)

**Number:** 1.1.1.2

**Description:** Allows selection of three FC1 DR cases.  
Case 1--Range Rings out of Tolerance in X Axis.  
Case 2-- Range Rings out of Tolerance in X and Y Axis.  
Case 3--Range Rings out of Tolerance in Y Axis.

**Called by:** FC1 DTRB

**Calls:** FC1 DR--Case 1, Case 2, and Case 3

---

**Name:** FC1 DR Case 1

**Number:** 1.1.1.2.1

**Description:** Allows trouble-shooting of FC1 DR Case 1 procedures.

**Called by:** FC1 DR

**Calls:** None

---

**Name:** FC1 DR Case 2

**Number:** 1.1.1.2.2

**Description:** Allows trouble-shooting of FC1 DR Case 2 procedures.

**Called by:** FC1 DR

**Calls:** None

**Name:** FC1 DR Case 3

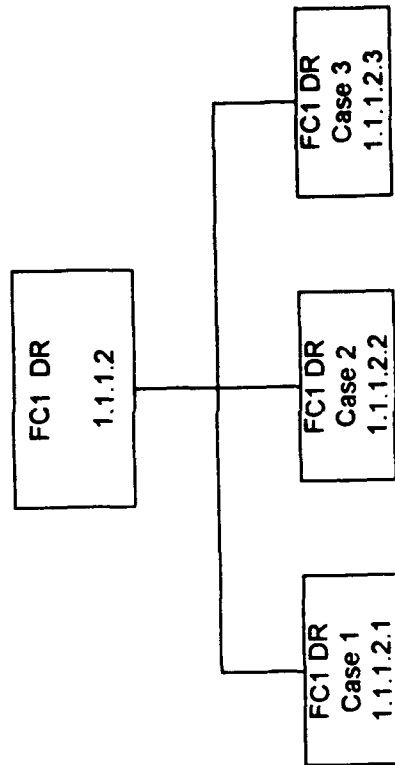
**Number:** 1.1.1.2.3

**Description:** Allows trouble-shooting of FC1 DR Case 3 procedures.

**Called by:** FC1 DR

**Calls:** None

## FC1 DR



- Case 1 -- Range rings out of tolerance in X axis.
- Case 2 -- Range rings out of tolerance in both X and Y axis.
- Case 3 -- Range rings out of tolerance in Y axis.

FIGURE 4

## **E. FC1 DB**

**Name:** FC1 DB (Figure 5)

**Number:** 1.1.1.3

**Description:** Allows selection of three FC1 DB procedures by case.  
(NOTE: these are the same cases called by FC1 DR.)  
Case 1--Range Rings out of Tolerance in X Axis.  
Case 2--Range Rings out of Tolerance in X and Y Axis.  
Case 3--Range Rings out of Tolerance in Y Axis.

**Called by:** FC1 DTRB

**Calls:** FC1 DR--Case 1, Case 2, and Case 3

---

**Name:** FC1 DR Case 1

**Number:** 1.1.1.2.1

**Description:** Allows trouble-shooting of FC1 DR Case 1 procedures.

**Called by:** FC1 DB

**Calls:** None

---

**Name:** FC1 DR Case 2

**Number:** 1.1.1.2.2

**Description:** Allows trouble-shooting of FC1 DR Case 2 procedures.

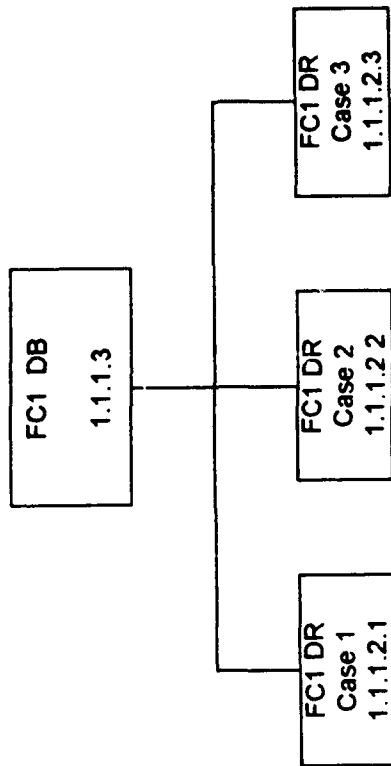
**Called by:** FC1 DB

**Calls:** None

**Name:** FC1 DR Case 3  
**Number:** 1.1.1.2.3  
**Description:** Allows trouble-shooting of FC1 DR Case 2 procedures.  
**Called by:** FC1 DB  
**Calls:** None



FC1 DB



NOTE

FC1 DB calls cases from FC1 DR.

FIGURE 5

## **F. FC1 ACQ**

**Name:** FC1 ACQ (Figure 6)  
**Number:** 1.1.2  
**Description:** Allows selection of FC1 ACQ procedures.  
**Called by:** FC1 Menu  
**Calls:** FC1 ACQ--No Elevation Scan, Low XTAL Current, and Settle Time

---

**Name:** FC1 ACQ No Elevation Scan  
**Number:** 1.1.2.1  
**Description:** Allows trouble-shooting of FC1 ACQ No Elevation Scan procedures.  
**Called by:** FC1 ACQ Menu  
**Calls:** FC1 ACQ D and FC1 ACQ E

---

**Name:** FC1 ACQ D  
**Number:** 1.1.2.1.1  
**Description:** Allows trouble-shooting of FC1 ACQ D procedures.  
**Called by:** FC1 ACQ No Elevation Scan  
**Calls:** None

**Name:** FC1 ACQ E  
**Number:** 1.1.2.2.1  
**Description:** Allows trouble-shooting of FC1 ACQ E procedures.  
**Called by:** FC1 ACQ No Elevation Scan, Low XTAL Current, and  
**Settle** Time. (NOTE: FC1 ACQ E is common to each of these  
procedures.)  
**Calls:** FC1 ACQ A, FC1 ACQ Ea, and FC1 ACQ Eb

---

**Name:** FC1 ACQ A  
**Number:** 1.1.2.2.1.1  
**Description:** Allows trouble-shooting of FC1 ACQ A procedures.  
**Called by:** FC1 ACQ E  
**Calls:** FC1 ACQ Aa

---

**Name:** FC1 ACQ Aa  
**Number:** 1.1.2.2.1.1.1  
**Description:** Continues trouble-shooting of FC1 ACQ A procedures.  
**Called by:** FC1 ACQ A  
**Calls:** None

**Name:** FC1 ACQ Ea  
**Number:** 1.1.2.2.1.2  
**Description:** Continues trouble-shooting of FC1 ACQ E procedures.  
**Called by:** FC1 ACQ E  
**Calls:** FC1 ACQ C

---

**Name:** FC1 ACQ C  
**Number:** 1.1.2.2.1.2.1  
**Description:** Allows trouble-shooting of FC1 ACQ C procedures.  
**Called by:** FC1 ACQ Ea  
**Calls:** None

---

**Name:** FC1 ACQ Eb  
**Number:** 1.1.2.2.1.3  
**Description:** Continues trouble-shooting of FC1 ACQ E procedures.  
**Called by:** FC1 ACQ E  
**Calls:** None

**Name:** FC1 ACQ Low XTAL Current  
**Number:** 1.1.2.2  
**Description:** Allows trouble-shooting of FC1 ACQ Low XTAL Current procedures.  
**Called by:** FC1 ACQ Menu  
**Calls:** FC1 ACQ E

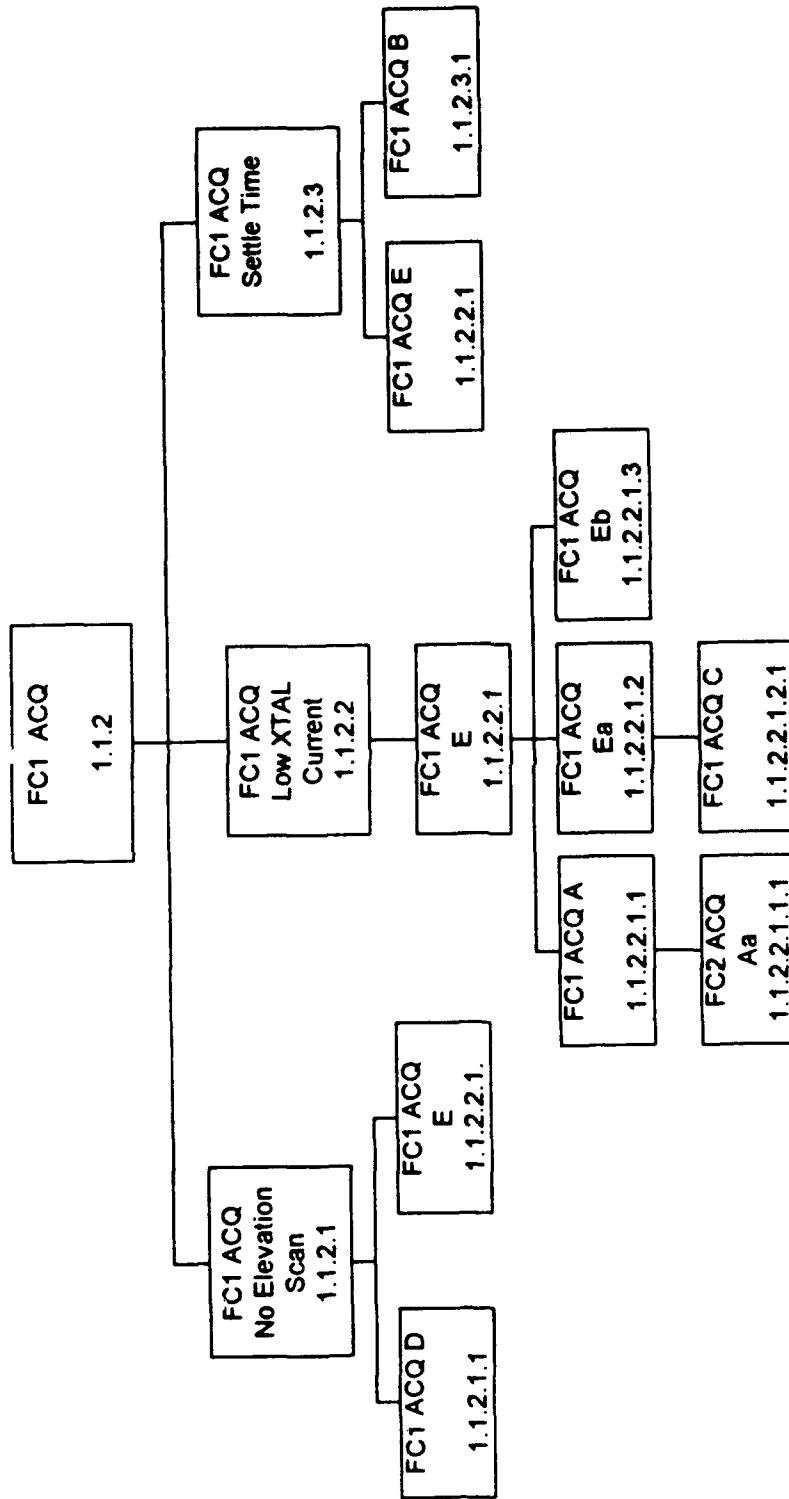
---

**Name:** FC1 ACQ Settle Time  
**Number:** 1.1.2.3  
**Description:** Allows trouble-shooting of FC1 ACQ Settle Time procedures.  
**Called by:** FC1 ACQ Menu  
**Calls:** FC1 ACQ E and FC1 ACQ B

---

**Name:** FC1 ACQ B  
**Number:** 1.1.2.3.1  
**Description:** Allows trouble-shooting of FC1 ACQ B procedures.  
**Called by:** FC1 ACQ Settle Time  
**Calls:** None

## FC1 ACQ



### NOTE

FC1 ACQ E and sub-procedures are common to FC1 ACQ No Elevation Scan, FC1 Low XTAL Current, and FC1 ACQ Settle Time. FC2 ACQ Ed feeds into FC2 ACQ E.

FIGURE 6

## **G. FC2 MENU**

**Name:** FC2 Menu (Figure 7)

**Number:** 1.2

**Description:** Allows selection of FC2 Designation--Time, Range, and Bearing. FC2 ACQ. FC2 Track--Bearing, Elevation, and Range.

**Called by:** FC2 Performance Menu

**Calls:** FC2 DTRB, FC2 ACQ, and FC2 TBER

---

**Name:** FC2 DTBR

**Number:** 1.2.1

**Description:** Allows selection of FC2 Designation--Time, Range, and Bearing procedures.

**Called by:** FC2 Menu

**Calls:** FC2 DT, FC2 TR, and FC2 TB

---

**Name:** FC2 ACQ

**Number:** 1.2.2

**Description:** Allows selection of FC2 ACQ procedures.

**Called by:** FC2 Menu

**Calls:** See FC2 ACQ

**Name:** FC2 TBER Menu

**Number:** 1.2.3

**Description:** Allows selection of FC2 Track--Bearing Elevation and  
**Range** procedures.

**Called by:** FC2 Menu

**Calls:** FC2 TB, FC2 TE, and FC2 TR



## FC2 Menu

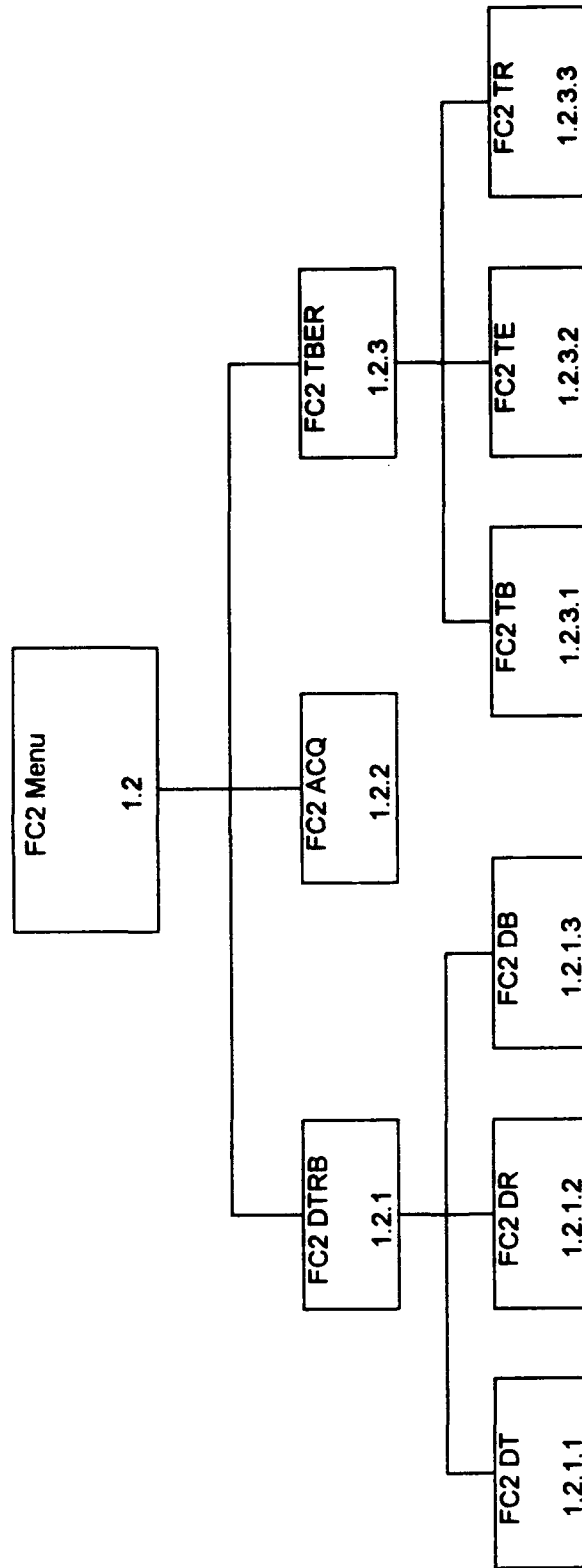


FIGURE 7

## **H. FC2 DT**

**Name:** FC2 DT (Figure 8)

**Number:** 1.2.1.1

**Description:** Allows selection of three FC2 DT cases.  
Case 1--Range Gate approximately 25K yards. Range Reading on TOTE equals zero or is less than 24K yards or greater than 26K yards.  
Case 2--Range Gate approximately 25K yards. Range reading on TOTE approximately 25K yards.  
Case 3--Range Gate not present or nowhere near 25K yards.

**Called by:** FC2 DTRB Menu

**Calls:** FC2 DT Case 1, FC2 Case 2, and FC2 DT Case 3

---

**Name:** FC2 DT Case 1

**Number:** 1.2.1.1.1

**Description:** Allows trouble-shooting of FC2 DT Case 1 procedures.

**Called by:** FC2 DT Menu

**Calls:** FC2 DT G

---

**Name:** FC2 DT G

**Number:** 1.2.1.1.1.1

**Description:** Allows trouble-shooting of FC2 DT G procedures.

**Called by:** FC2 DT Case 1

**Calls:** FC2 DT Ga

**Name:** FC2 DT Ga  
**Number:** 1.2.1.1.1.1  
**Description:** Continues trouble-shooting of FC2 DT G procedures.  
**Called by:** FC2 DT G  
**Calls:** None

---

**Name:** FC2 DT Case 2  
**Number:** 1.2.1.1.2  
**Description:** Allows trouble-shooting of FC2 DT Case 2 procedures.  
**Called by:** FC2 Menu  
**Calls:** FC2 DT--No Track Antenna Movement, Track Antenna Slow, Settle Time, Range Gate Does Not Move, Range Gate Slow, Both Slow or Fixed.

---

**Name:** FC2 DT No Track Antenna Movement (Figure 8A)  
**Number:** 1.2.1.1.2.1  
**Description:** Allows selection of FC2 DT No Track Antenna Movement procedures.  
**Called by:** FC2 DT Case 2  
**Calls:** FC2 DT No Track Antenna Movement--A, B, C, D, and E

**Name:** FC2 DT No Track Antenna Movement A  
**Number:** 1.2.1.1.2.1.1  
**Description:** Allows trouble-shooting of FC2 DT No Track Antenna Movement A procedures.  
**Called by:** FC2 DT No Track Antenna Movement  
**Calls:** FC2 DT No Track Antenna Movement Aa

---

**Name:** FC2 DT No Track Antenna Movement Aa  
**Number:** 1.2.1.1.2.1.1.1  
**Description:** Allows trouble-shooting of FC2 DT No Track Antenna Movement Aa procedures.  
**Called by:** FC2 DT No Track Antenna Movement A  
**Calls:** FC2 DT No Track Antenna Movement Ab and FC2 DT No Track Antenna Movement Ac

---

**Name:** FC2 DT No Track Antenna Movement Ab  
**Number:** 1.2.1.1.2.1.1.1.1  
**Description:** Allows trouble-shooting of FC2 DT No Track Antenna Movement Ab procedures.  
**Called by:** FC2 DT No Track Antenna Movement Aa  
**Calls:** None

**Name:** FC2 DT No Track Antenna Movement Ac  
**Number:** 1.2.1.1.2.1.1.1.2  
**Description:** Allows trouble-shooting of FC2 DT No Track Antenna Movement Ac procedures.  
**Called by:** FC2 DT No Track Antenna Movement Aa  
**Calls:** None

---

**Name:** FC2 DT No Track Antenna Movement B  
**Number:** 1.2.1.1.2.1.2  
**Description:** Allows trouble-shooting of FC2 DT No Track Antenna Movement B procedures.  
**Called by:** FC2 DT No Track Antenna Movement  
**Calls:** FC2 DT No Track Antenna Movement Ba and FC2 DT No Track Antenna Movement Bb

---

**Name:** FC2 DT No Track Antenna Movement Ba  
**Number:** 1.2.1.1.2.1.2.1  
**Description:** Allows trouble-shooting of FC2 DT No Track Antenna Movement Ba procedures.  
**Called by:** FC2 DT No Track Antenna Movement B  
**Calls:** None

**Name:** FC2 DT No Track Antenna Movement Bb  
**Number:** 1.2.1.1.2.1.2.2  
**Description:** Allows trouble-shooting of FC2 DT No Track Antenna Movement Bb procedures.  
**Called by:** FC2 DT No Track Antenna Movement B  
**Calls:** None

---

**Name:** FC2 DT No Track Antenna Movement C  
**Number:** 1.2.1.1.2.1.3  
**Description:** Allows trouble-shooting of FC2 DT No Track Antenna Movement C procedures.  
**Called by:** FC2 DT No Track Antenna Movement  
**Calls:** None

---

**Name:** FC2 DT No Track Antenna Movement D  
**Number:** 1.2.1.1.2.1.4  
**Description:** Allows trouble-shooting of FC2 DT No Track Antenna Movement D procedures.  
**Called by:** FC2 DT No Track Antenna Movement  
**Calls:** None

**Name:** FC2 DT No Track Antenna Movement E  
**Number:** 1.2.1.1.2.1.5  
**Description:** Allows trouble-shooting of FC2 DT No Track Antenna Movement E procedures.  
**Called by:** FC2 DT No Track Antenna Movement  
**Calls:** None

---

**Name:** FC2 DT Track Antenna Slow (Figure 8B)  
**Number:** 1.2.1.1.2.2  
**Description:** Allows trouble-shooting of FC2 DT Track Antenna Slow procedures.  
**Called by:** FC2 DT Case 2  
**Calls:** FC2 DT Track Antenna Slow F

---

**Name:** FC2 DT Track Antenna Slow F  
**Number:** 1.2.1.1.2.2.1  
**Description:** Allows trouble-shooting of FC2 DT Track Antenna Slow F procedures.  
**Called by:** FC2 DT Track Antenna Slow  
**Calls:** FC2 DT Track Antenna Slow Fa and FC2 DT Track Antenna Slow Fb

**Name:** FC2 DT Track Antenna Slow Fa  
**Number:** 1.2.1.1.2.2.1.1  
**Description:** Allows trouble-shooting of FC2 DT Track Antenna Slow Fa procedures.  
**Called by:** FC2 DT Track Antenna Slow F  
**Calls:** None

---

**Name:** FC2 DT Track Antenna Slow Fb  
**Number:** 1.2.1.1.2.2.1.2  
**Description:** Allows trouble-shooting of FC2 DT Track Antenna Slow Fb procedure.  
**Called by:** FC2 DT Track Antenna Slow F  
**Calls:** None

---

**Name:** FC2 DT Settle Time (See Figure 8)  
**Number:** 1.2.1.1.2.3  
**Description:** Allows trouble-shooting of FC2 DT Settle Time procedures.  
**Called by:** FC2 DT Case 2  
**Calls:** None



**Name:** FC2 DT Range Gate Does Not Move  
**Number:** 1.2.1.1.2.4  
**Description:** Allows trouble-shooting of FC2 DT Range Gate Does Not Move procedures.  
**Called by:** FC2 DT Case 2  
**Calls:** None

---

**Name:** FC2 DT Range Gate Slow  
**Number:** 1.2.1.1.2.5  
**Description:** Allows trouble-shooting of FC2 DT Range Gate Slow procedures.  
**Called by:** FC2 DT Case 2  
**Calls:** None

---

**Name:** FC2 DT Both Slow or Fixed  
**Number:** 1.2.1.1.2.6  
**Description:** Allows trouble-shooting of FC2 DT Both Slow or Fixed procedures.  
**Called by:** FC2 DT Case 2  
**Calls:** None

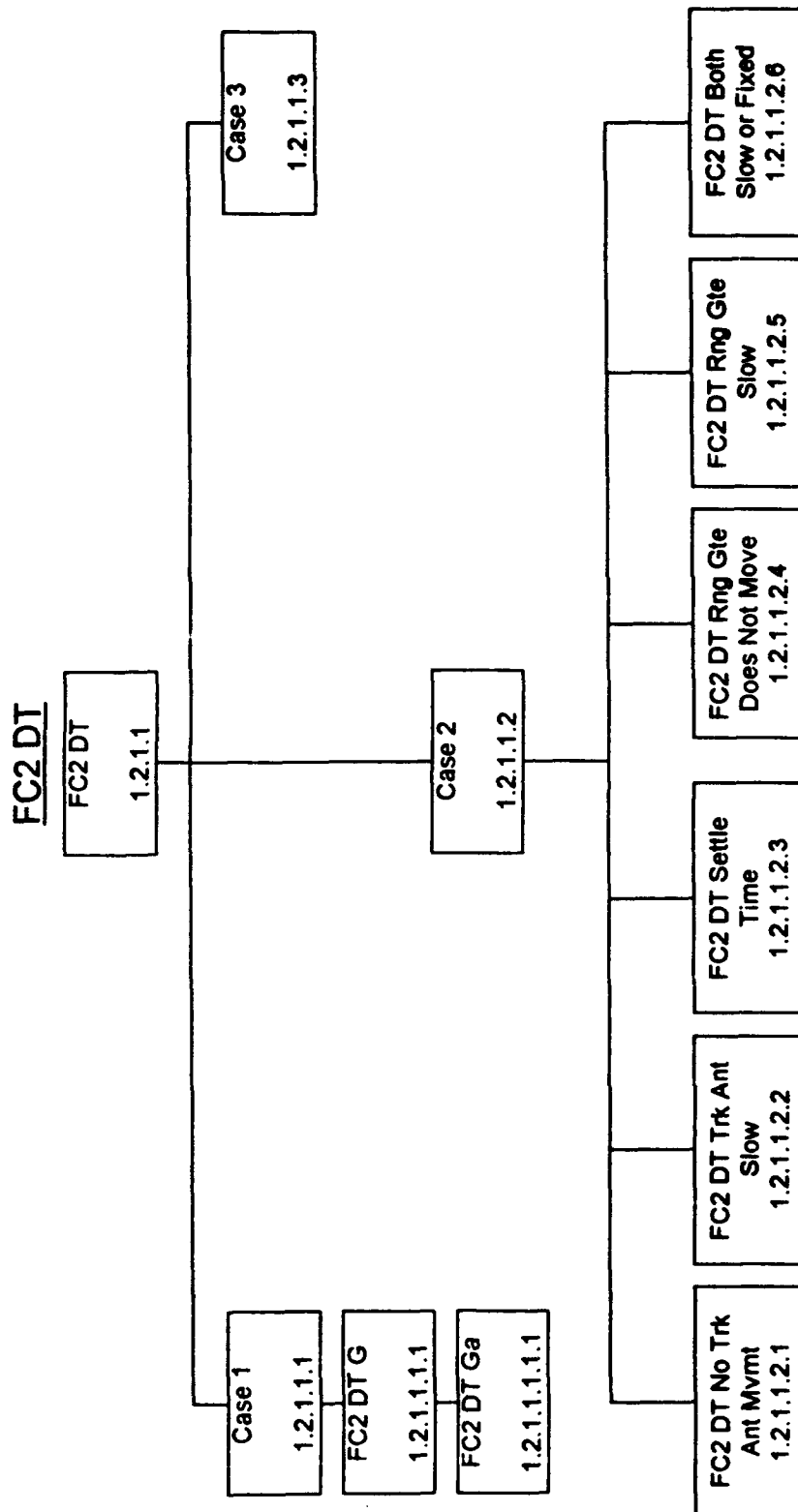
**Name:** FC2 DT Case 3

**Number:** 1.2.1.1.3

**Description:** Allows trouble-shooting of FC2 DT Case 3 procedures.

**Called by:** FC2 DT Menu

**Calls:** None



Case 1 -- Rng Gte approx 25 Kyds. Rng Rding on TOTE 0 or < 24Kyds or > 26 Kyds

Case 2 -- Rng Gte approx 25 Kyds. Rng Rding on TOTE approx 25Kyds.

Case 3 -- Rng Gte not present or no where near 25Kyds.

**FIGURE 8**

FC2 DT No Trk Ant Mvmt

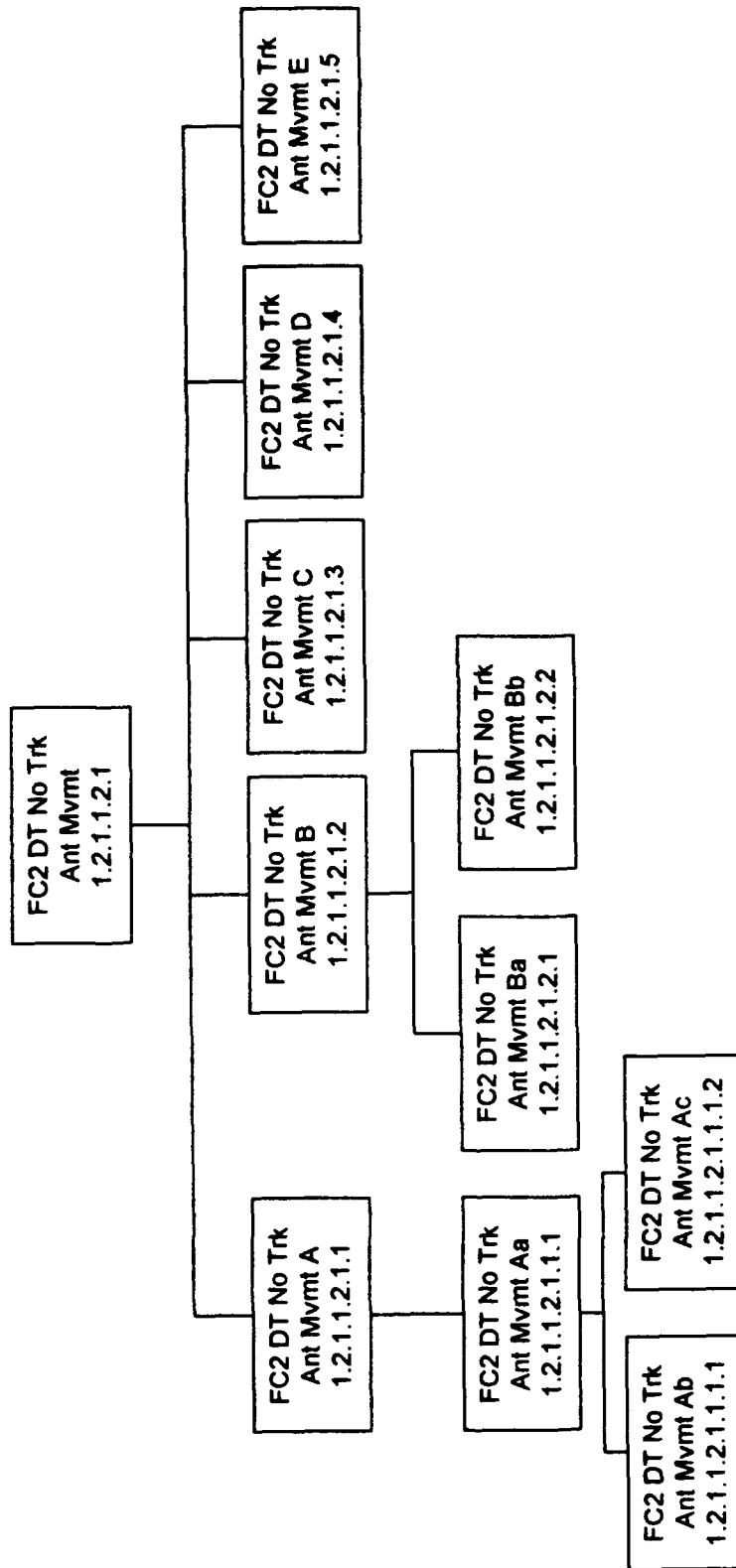


FIGURE 8A

FC2 DT Trk Ant Slow

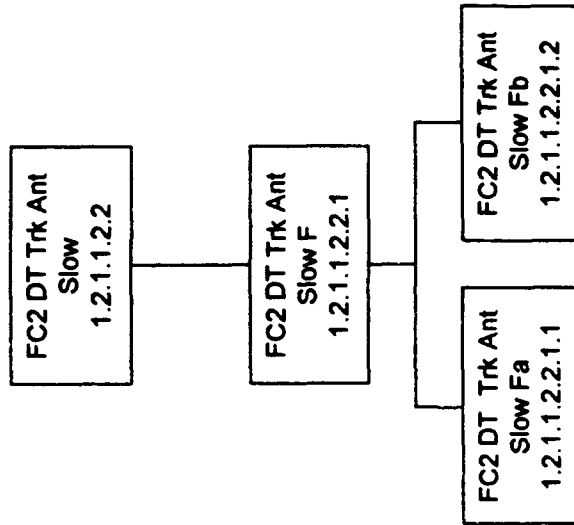


FIGURE 8B

## **L FC2 DR**

**Name:** FC2 DR (Figure 9)

**Number:** 1.2.1.2

**Description:** Allows selection of three FC2 DR cases.  
Case 1--Range Rings out of Tolerance in X Axis.  
Case 2--Range Rings out of Tolerance in X and Y Axis.  
Case 3--Range Rings out of Tolerance in Y Axis.

**Called by:** FC2 DTRB

**Calls:** FC2 DR--Case 1, Case 2, and Case 3

---

**Name:** FC2 DR Case 1

**Number:** 1.2.1.2.1

**Description:** Allows trouble-shooting of FC2 DR Case 1 procedures.

**Called by:** FC2 DR

**Calls:** None

---

**Name:** FC2 DR Case 2

**Number:** 1.2.1.2.2

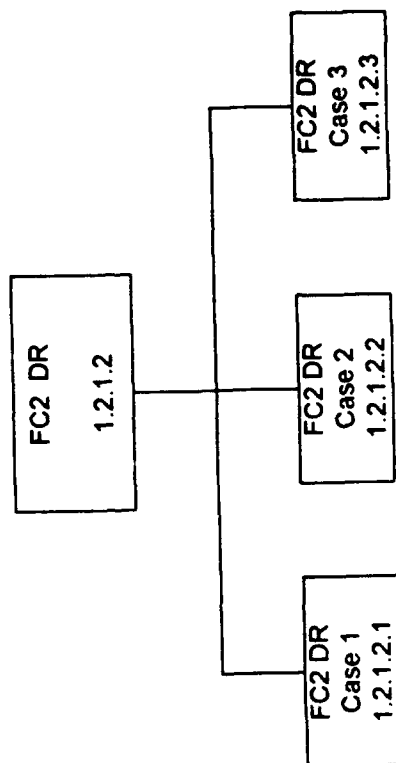
**Description:** Allows trouble-shooting of FC2 DR Case 2 procedures.

**Called by:** FC2 DR

**Calls:** None

**Name:** FC2 DR Case 3  
**Number:** 1.2.1.2.3  
**Description:** Allows trouble-shooting of FC2 DR Case 3 procedures.  
**Called by:** FC2 DR  
**Calls:** None

## FC2 DR



- Case 1 -- Range rings out of tolerance in X axis.
- Case 2 -- Range rings out of tolerance in both X and Y axis.
- Case 3 -- Range rings out of tolerance in Y axis.

FIGURE 9



## **J. FC2 DB**

**Name:** FC2 DB Menu (Figure 10)

**Number:** 1.2.1.3

**Description:** Allows selection of FC2 DB cases.  
(NOTE: These are the same cases called by FC2 DR.)  
Case 1--Range Rings out of Tolerance in X Axis.  
Case 2--Range Rings out of Tolerance in X and Y Axis.  
Case 3--Range Rings out of Tolerance in Y Axis.

**Called by:** FC2 DTRB

**Calls:** FC2 DR--Case 1, Case 2, and Case 3

---

**Name:** FC2 DR Case 1

**Number:** 1.2.1.2.1

**Description:** Allows trouble-shooting of FC2 DR Case 1 procedures.

**Called by:** FC2 DB

**Calls:** None

---

**Name:** FC2 DR Case 2

**Number:** 1.2.1.2.2

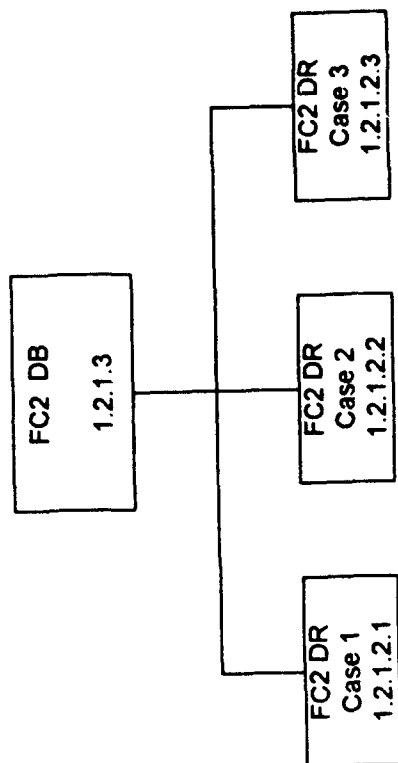
**Description:** Allows trouble-shooting of FC2 DR Case 2 procedures.

**Called by:** FC2 DB

**Calls:** None

**Name:** FC2 DR Case 3  
**Number:** 1.2.1.2.3  
**Description:** Allows trouble-shooting of FC2 DR Case 3 procedures.  
**Called by:** FC2 DB  
**Calls:** None

FC2 DB



NOTE

FC2 D3 calls cases from FC2 DR.

FIGURE 10

## **K FC2 ACQ**

**Name:** FC2 ACQ (Figure 11)

**Number:** 1.2.2

**Description:** Allows selection of FC2 ACQ procedures.

**Called by:** FC2 Menu

**Calls:** FC2 ACQ--No Elevation Scan, Low XTAL Current, and Weak or No Video

---

**Name:** FC2 ACQ No Elevation Scan

**Number:** 1.2.2.1

**Description:** Allows trouble-shooting of FC2 ACQ No Elevation Scan procedures.

**Called by:** FC2 ACQ

**Calls:** FC2 ACQ A and FC2 ACQ E

---

**Name:** FC2 ACQ A

**Number:** 1.2.2.1.1

**Description:** Allows trouble-shooting of FC2 ACQ A procedures.

**Called by:** FC2 ACQ No Elevation Scan

**Calls:** FC2 ACQ Aa, FC2 ACQ Ac, and FC2 ACQ Ad

**Name:** FC2 ACQ Aa  
**Number:** 1.2.2.1.1.1  
**Description:** Allows trouble-shooting of FC2 ACQ Aa procedures.  
**Called by:** FC2 ACQ A  
**Calls:** FC2 ACQ Ab

---

**Name:** FC2 ACQ Ab  
**Number:** 1.2.2.1.1.1.1  
**Description:** Continues trouble-shooting of FC2 ACQ Aa procedures.  
**Called by:** FC2 ACQ Aa  
**Calls:** None

---

**Name:** FC2 ACQ Ac  
**Number:** 1.2.2.1.1.2  
**Description:** Allows trouble-shooting of FC2 ACQ Ac procedures.  
**Called by:** FC2 ACQ A  
**Calls:** None

**Name:** FC2 ACQ Ad  
**Number:** 1.2.2.1.1.3  
**Description:** Allows trouble-shooting of FC2 ACQ Ad procedures.  
**Called by:** FC2 ACQ A  
**Calls:** None

---

**Name:** FC2 ACQ E  
**Number:** 1.2.2.2.1  
**Description:** Allows trouble-shooting of FC2 ACQ E procedures.  
**Called by:** FC2 ACQ No Elevation Scan, Low XTAL Current, and Weak or No Video. (NOTE: FC2 ACQ E is common to each of these procedures.)  
**Calls:** FC2 ACQ B, FC2 ACQ Ea, and FC2 ACQ Eb

---

**Name:** FC2 ACQ B  
**Number:** 1.2.2.2.1.1  
**Description:** Allows trouble-shooting of FC2 ACQ B procedures.  
**Called by:** FC2 ACQ E  
**Calls:** FC2 ACQ Ba

**Name:** FC2 ACQ Ba  
**Number:** 1.2.2.2.1.1.1  
**Description:** Continues trouble-shooting of FC2 ACQ B procedures.  
**Called by:** FC2 ACQ B  
**Calls:** None

---

**Name:** FC2 ACQ Ea  
**Number:** 1.2.2.2.1.2  
**Description:** Allows trouble-shooting of FC2 ACQ E procedures.  
**Called by:** FC2 ACQ E  
**Calls:** FC2 ACQ C

---

**Name:** FC2 ACQ C  
**Number:** 1.2.2.2.1.2.1  
**Description:** Allows trouble-shooting of FC2 ACQ C procedures.  
**Called by:** FC2 ACQ Ea  
**Calls:** None

**Name:** FC2 ACQ Ec  
**Number:** 1.2.2.2.1.3.1  
**Description:** Continues trouble-shooting of FC2 ACQ Eb procedures.  
**Called by:** FC2 ACQ Eb  
**Calls:** None

---

**Name:** FC2 ACQ Low XTAL Current  
**Number:** 1.2.2.2  
**Description:** Allows trouble-shooting of FC2 ACQ Low XTAL Current procedures.  
**Called by:** FC2 ACQ Menu  
**Calls:** FC2 ACQ E

---

**Name:** FC2 ACQ Weak or No Video  
**Number:** 1.2.2.3  
**Description:** Allows trouble-shooting of FC2 ACQ Weak or No Video procedures.  
**Called by:** FC2 ACQ Menu  
**Calls:** FC2 ACQ Ed



**Name:** FC2 ACQ Ed

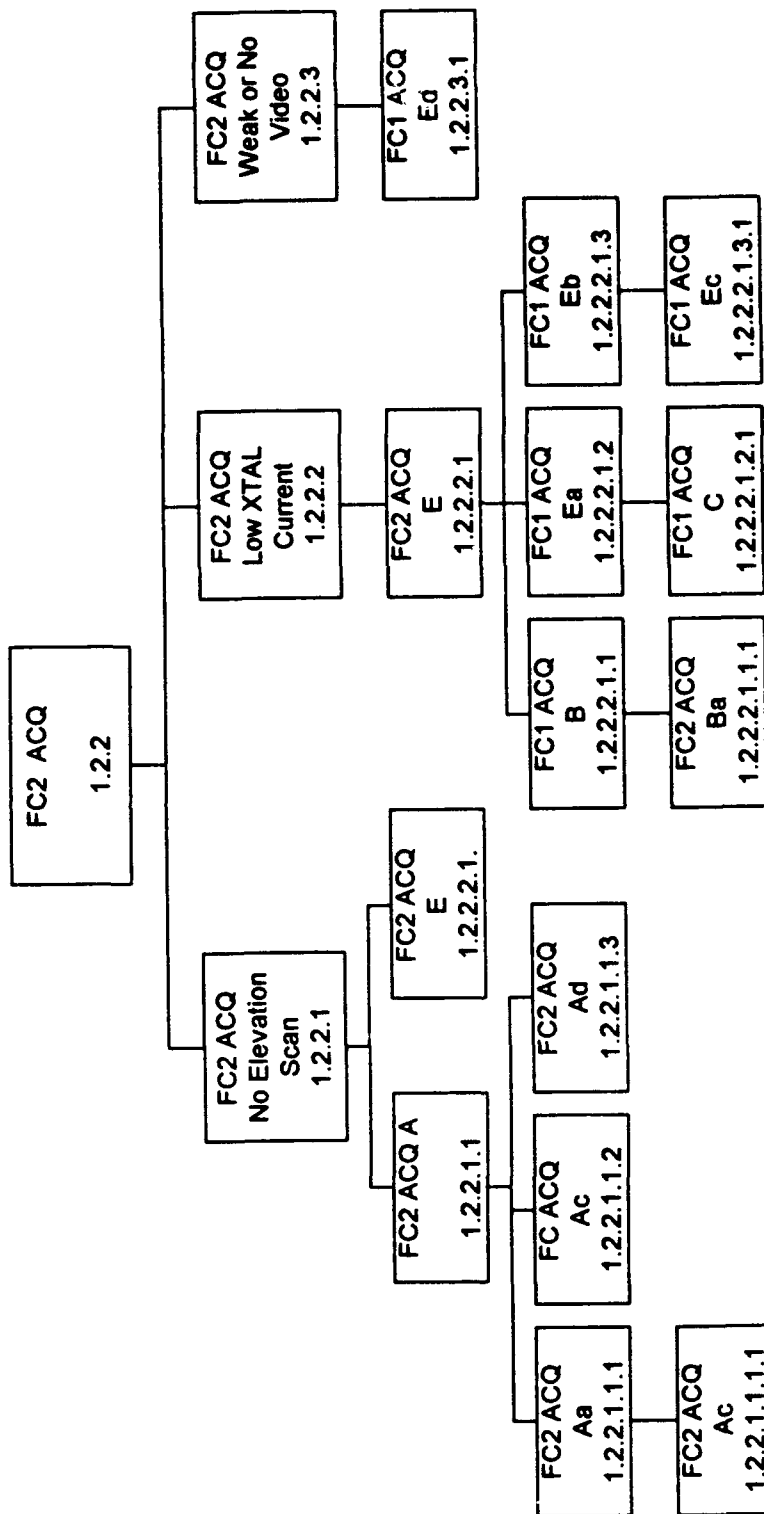
**Number:** 1.2.2.3.1

**Description:** Allows trouble-shooting of FC2 ACQ Ed procedures.

**Called by:** FC2 ACQ Weak or No Video

**Calls:** FC2 ACQ E. (NOTE: FC2 ACQ Ed links into FC2 ACQ E.)

## FC2 ACQ



### NOTE

FC2 ACQ E and sub-procedures are common to No Elevation Scan, Weak or No Video. FC2 ACQ Ed feeds into FC2 ACQ E.

FIGURE 11

## **L FC4 AND FC5**

**Name:** FC4 and FC5 Menu (Figure 12)

**Number:** 1.3

**Description:** Allows selection of FC4 and FC5 Track Bearing Menu, FC4 and FC5 Designation Time Menu, or FC4 and FC5 Track Range Menu.

**Called by:** Performance Menu

**Calls:** FC4 and FC5 TB, FC4 and FC5 DT, and FC4 and FC5 TR.

---

**Name:** FC4 and FC5 TB Menu

**Number:** 1.3.1

**Description:** Allows trouble-shooting of FC4 and FC5 TB procedures.

**Called by:** FC4 and FC5 Menu

**Calls:** None

---

**Name:** FC4 and FC5 DT Menu

**Number:** 1.3.2

**Description:** Allows trouble-shooting of FC4 and FC5 DT procedures.

**Called by:** FC4 and FC5 Menu

**Calls:** FC4 DT Only, FC4 and FC5 DT, and FC5 DT Only

**Name:** FC4 DT Only  
**Number:** 1.3.2.1  
**Description:** Allows trouble-shooting of FC4 DT Only procedures.  
**Called by:** FC4 and FC5 DT Menu  
**Calls:** None

---

**Name:** FC4 and FC5 DT  
**Number:** 1.3.2.2  
**Description:** Allows trouble-shooting of FC4 and FC5 DT procedures.  
**Called by:** FC4 and FC5 DT Menu  
**Calls:** None

---

**Name:** FC5 DT Only  
**Number:** 1.3.2.3  
**Description:** Allows trouble-shooting of FC5 DT Only procedures.  
**Called by:** FC4 and FC5 DT Menu  
**Calls:** None

**Name:** FC4 and FC5 TR Menu

**Number:** 1.3.3

**Description:** Allows trouble-shooting of FC4 and FC5 TR procedures.

**Called by:** FC4 and FC5 Menu

**Calls:** None

## FC4 and FC5

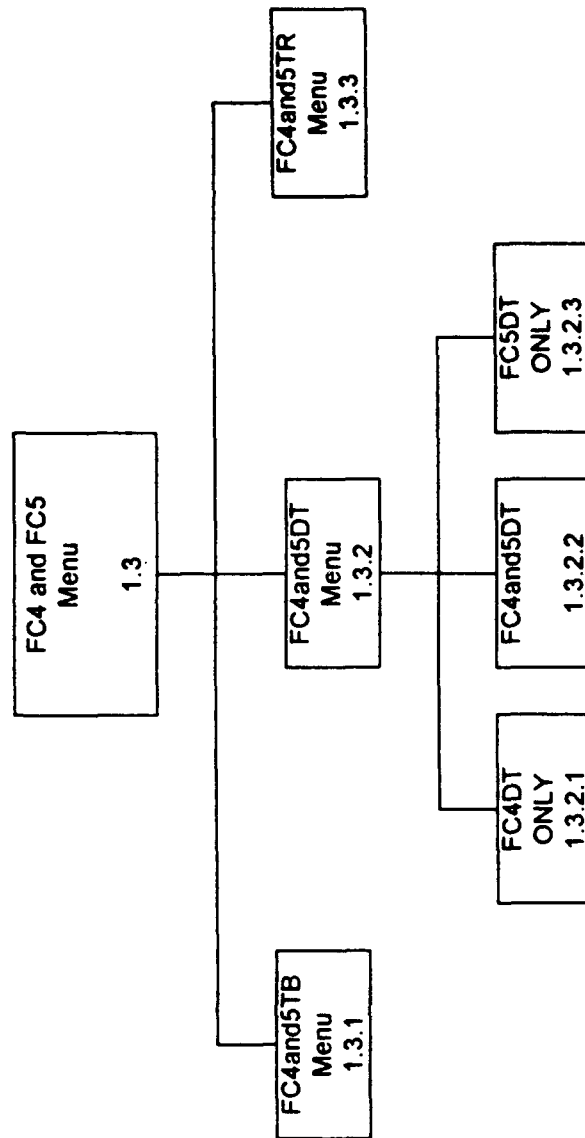


FIGURE 12

**APPENDIX BMK 92 MOD 2 MAINTENANCE ADVISOR EXPERT  
SYSTEM  
USER'S MANUAL**

**M. INTRODUCTION**

The MK 92 MOD 2 Maintenance Advisor Expert System was developed as a joint effort between the Naval Surface Warfare Center, Port Hueneme Division and the Naval Postgraduate School. It was designed to assist the shipboard Fire Control Technician in isolating system faults as indicated by the Daily System Operability Test NOGO's. It is important to note, however, that the expert system is not meant to replace the Fire Control Technician, compete with his knowledge and experience, or replace the MK 92 MOD 2 Maintenance Manuals.

The reasoning contained within the program was designed to be "expert knowledge." Design was based on heuristics (rules of thumb) and probabilities developed through years of experience by several experts. Therefore, the trouble-shooting logic illustrated by the Maintenance Advisor Expert System in many cases does not follow the same logic as the MK 92 MOD 2 maintenance manuals.

It is impossible to foresee every potential malfunction with the system. Therefore, the solutions proposed by the expert system are only recommendations, and are not guaranteed to remedy every fault.

Special consideration went into the design of the system. Because of the dynamic environment of shipboard operations a graphical user interface (GUI), with a point-and-click approach is used rather than keyboard entry. The operator needs only to point the mouse to the appropriate button and click a selection. Compactness and portability were also important considerations. Moving the system from one compartment to another in the trouble-shooting process necessitates using a small notebook type computer rather than a bulkier desktop system.

In order to work effectively with the MK 92 MOD 2 Maintenance Advisor Expert System, a basic understanding of Windows is necessary. The old adage of "keep it simple" was paramount in the design process. The operator should be familiar with the following operations:

- Use of the mouse to point and click.
- Start and quit applications.



## **N. EQUIPMENT NEEDS**

The program that acts as the shell or driver of the Maintenance Advisor Expert System is called Adept, developed by the Symbologic Corporation in Redmond, Washington. It is referred to in some literature as an inference engine. The program was designed to work on any 80286 computer or faster system using Windows 3.0 or any newer version of Windows. The minimum requirements for running the Maintenance Advisor are:

- A Windows compatible micro-computer with 1 MB memory.
- A hard disk drive with at least 6 MB of storage space available.
- Microsoft Windows 3.0 or later version of Windows.
- A 5.25 inch, 1.2 MB floppy drive or 3.5 inch, 1.44 MB floppy drive.
- A VGA color or monochrome monitor.
- A mouse connected to a serial or parallel port.

While these parameters represent the minimum computer capability required to run the MK 92 MOD 2 Maintenance Advisor Expert System, performance will be considerably enhanced utilizing advanced configurations. Four MB of memory is recommended over the 1MB minimum. An 80386 based computer yields a much improved response time

over the minimum 80286, though an 80486 based computers is recommended.

While advanced monitors may provide better resolution, they are not compatible with the program's text layouts. Attempting to run in another mode will jumble the displayed characters rendering the text undecipherable.

#### **O. LOADING THE PROGRAM**

Windows functions make loading the Run-Time program a relatively simple task. Loading the MK 92 MOD 2 Maintenance Advisor Expert System is slightly more complicated. Because of the size of the file it is necessary to do a backup in order to copy it to a floppy disk. The following procedures should be followed to transfer the files to the hard drive:

#### **P. INSTALLING THE RUN-TIME PROGRAM**

- Turn on the computer and select Windows if the system boots to DOS or another menu.
- Select the icon labelled "MAIN."
- Select "FILE MANAGER."
- Place the Run-Time program disk in the floppy drive and close the door.
- Select the floppy drive containing the program disk and observe the files listed on the right of the screen.
- Select the last file, entitled "SETUP.EXE".

- Select "CONTINUE" to load program to C:\ADEPT. The program automatically loads to the "C" drive in the ADEPT Directory.
- Check the AUTOEXEC.BAT file to ensure the appropriate path, as indicated on the screen, is present and select "OK". The Run-Time program is now loaded and the Program Manager is redisplayed.

**Q. RESTORING THE MK 92 MOD 2 MAINTENANCE ADVISOR EXPERT SYSTEM PROGRAM**

- Select the "MS-DOS" icon.
- At the command prompt, "C:\Windows>," type "RESTORE a: c:\windows\adept\MK92."
- The expert system resides on several disks. Simply follow the instructions on the screen to complete the restoration of the file.
- After the restoration is complete, type "EXIT" to return to the Program Manager.

**R. LOADING THE GRAPHICS**

- Select the icon labelled "MAIN".
- Select "FILE MANAGER".
- Place the graphics disk in the floppy drive and close the door.
- Select the floppy drive containing the graphics disk and retrieve the files listed on the right of the screen.
- Select "DISK" and then "DISK COPY". Follow the instructions on the screen and copy the graphic files to "C:" drive.

## **S. RUNNING THE PROGRAM**

- Select the Adept Run-Time program labelled "RUN-ADEPT" at the Program Manager screen.
- Select "APPLICATION" and "OPEN." The Adept files will be displayed.
- Highlight the MK 92 file and select "OK". The program is now ready for use.
- Follow the directions on the screen. The program is self-explanatory.
- The application can be terminated at any time by selecting "APPLICATION" and "EXIT". Selecting "EXIT" within the program at the Main Menu will also terminate the program, but not the application.

## **T. SCREEN LAYOUTS**

Much consideration went into the design of the display screens incorporated in the expert system. Screens are divided into several sections depending on the purpose. While display standardization was an important consideration in building the displays, some deviation was necessary to implement the program. Screen colors were specifically selected to be pleasing to the eye and prevent fatigue after extended use.

At the top of most screens is a Title Bar highlighted in yellow. A title and sub-title, when applicable, are centered on each bar so the operator can readily see where he is in the trouble-shooting process. Below the title bar is a procedure area where the operator finds instructions, information, or pictures, or is presented with a question or case selection. The lower

portion of the display is the action area. Located here are the push buttons, corresponding to the appropriate responses to questions or case selections.

Whenever possible, push buttons are standardized. Menu selections are the major exceptions. Primary push button selections with a brief explanation are listed below:

- YES/NO--Responses to questions. Selects the correct logic path based on responses to questions.
- CONTINUE--Continues with the program or the next help screen.
- PREVIOUS--Returns to the previous screen.
- RETURN--Returns to program from help.
- HELP--Provides information or reference for a specific procedure.

Menu selection is also effected via push buttons. For example, the operator will be given choices of the FC channel he wishes to trouble-shoot: FC1, FC2, FC4, or FC5. FC1 is further broken down into Designation Time (DT), Designation Range (DR), Designation Bearing (DB), Track - Bearing - Elevation - Range (Trk BER), and Acquisition (ACQ). [See figure 1] Presently, the program only covers the performance areas delineated above. The calibration area, FC3, is currently under development.

In keeping with standard Navy color schemes for safety, yellow backgrounds are used for Cautions and red backgrounds are used for Warnings. Important notes are displayed in blue text.

Some HELPS are referred to by letters. For example, the operator will be directed to select HELP A for additional assistance. This lettering scheme is arbitrary and was chosen for the convenience of the programmers. HELPs designated by letters are not necessarily the same across different procedures. For example, HELP A in FC1 Designation Time is not the same as HELP A in FC1 Designation Bearing.

#### **U. RESULT SCREENS**

Result screens at the end of the logic flow recommend components to trouble-shoot for fault correction. Of course, these are recommendations only and are not guaranteed in any way to remedy the problem. When the fault cannot be isolated to a single component, the order in which the components are listed is very important. Parts are listed in order of probability of failure. Thus, replacement should proceed in the order components are listed.

#### **V. CHANGE RECOMMENDATIONS**

As discussed earlier, the logic represented by the Expert System Maintenance Advisor is intended to be "expert" knowledge. Since it is impossible for any one person to be "the expert" there may exist better

methods to diagnose given faults than are represented in this program. Therefore, technicians' recommendations for change are encouraged.

Send your input with a brief description of recommended changes and any necessary references to:

Naval Surface Warfare Center Division  
Code 4W32  
4363 Missile Way  
Port Hueneme CA 93043-4307  
Attn: Mr. Henry Seto

#### **W. ABBREVIATIONS**

Throughout the program standard abbreviations are used comparable to those found in the maintenance manuals and MRC cards. Some of the Menu screens however use non-standard abbreviations. A list of those menu selection abbreviations is provided below:

ACQ.....Acquisition  
No Mvmt.....Both No Movement  
Both No Mvmt or Slow.....Both No Movement or Slow  
Excessive Trk Ant Settle Time.....Excessive Track Antenna Settle Time  
DB.....Designation Bearing  
DR .....Designation Range

DT.....Designation Time  
 No Trk Ant Mvmt.....No Track Antenna Movement  
 No Rng Gate Mvmt.....No Range Gate Movement  
 PAT.....Pulse Amplitude Track  
 PDT.....Pulse Doppler Track  
 Rng Gate Slow.....Range Gate Slow  
 TELEVTN.....Track Elevation  
 TBRNG.....Track Bearing  
 TRNG.....Track Range  
 TRNG GTE CIRCS.....Track Range Gate Circuits  
 Trk Ant Slow .....Track Antenna Slow  
 Trk BER .....Track Bearing, Elevation and Range  
 Settle Time of Trk Ant in Brg.....Settle Time of Track Antenna in Bearing

**NOTE:**

Symbologic and Symbologic Adept are trademarks of Symbologic Corporation.

Microsoft, MS-DOS, and Microsoft Windows are registered trademarks of Microsoft Corporation.



## LIST OF REFERENCES

1. Badiru, Adedeji B. *Expert Systems Applications in Engineering and Manufacturing*. 1992. Prentice Hall, Englewood Cliffs, New Jersey.
2. Biondo, Samuel J. *Fundamentals of Expert Systems Technology: Principles and Concepts*. 1990. Ablex Publishing Company, Norwood, New Jersey.
3. Corrico, Michael A., John E. Girard, and Jennifer P. Jones. *Building Knowledge Systems*. 1989. McGraw-Hill Book Company, New York.
4. Himes, Andrew, and Steven Sperry. 1991. *Symbologic Adept Reference*. Symbologic Corporation.
5. Leonard-Barton, and Dorothy and John J. Sviokla. "Putting Expert Systems to Work," *Harvard Business Review*. v. 66 pp. 91-98, March-April 1988.
6. Port Hueneme Division, Naval Surface Warfare Center. 1992. "System Requirements for the Engineering Development Model (EDM) FCS MK 92 Maintenance Advisor Expert System."
7. Powell, Steven H. *Economic Analysis of the MK 92 MOD 2 Maintenance Advisor Expert System*. Master's Thesis, Naval Postgraduate School, Monterey, California, September 1993.
8. Prerau, David S. *Developing and Managing Expert Systems: Proven Techniques for Business and Industry*. 1990. Addison-Wesley Publishing Company Inc, Menlo Park, California.
9. Turban, Efraim. *Decision Support and Expert Systems: Management Support Systems*. 2nd. ed. 1990. Macmillan Publishing Company, New York.

10. Walters, John R., and Norman R. Neilsen. *Crafting Knowledge-Based Systems: Expert Systems Made Easy Realistic*. 1988. John Wiley & Sons, New York.
11. Waterman, Donald A. *A Guide to Expert Systems*. 1986. Addison-Wesley Publishing Company, Menlo Park, California.

## **BIBLIOGRAPHY**

Eisenstadt, Marc, and Mike Brayshaw. "A Knowledge Engineering Toolkit," *Byte*, v.15, pp. 268-282, October 1990.

Eisenstadt, Marc, and Mike Brayshaw. "A Knowledge Engineering Toolkit Part 2," *Byte*, v.15, pp. 364-370, November 1990.

Halstead, Rodd. "Develop Advanced Expert Systems," *Byte*, v.15, pp. 219-224, January 1990.

Pedersen, Ken. *Expert Systems Programming*. John Wiley and Sons, New York, 1989.

Summers, Eric. "ES: A Public Domain Expert System," *Byte*, v.15, pp. 289-292, October 1990.

Tzafestas, Spyros G. *Knowledge-Based System Diagnosis, Supervision and Control*. 1989. Plenum Press, New York.

### INITIAL DISTRIBUTION LIST

- |    |  |   |
|----|--|---|
| 1. | Defense Technical Information Center<br>Cameron Station<br>Alexandria, Virginia 22304-6145   | 2 |
| 2. | Library, Code 052<br>Naval Postgraduate School<br>Monterey, California 93943-5002  | 2 |
| 3. | Capt. O.H. Perry III<br>Naval Sea Systems Command<br>Code 62Z, NC3, Room 8W06<br>2531 Jefferson Davis Highway<br>Washington, D.C. 22242-5160 | 1 |
| 4. | Mr. Ed McGill<br>Naval Sea Systems Command<br>Code 62ZP, NC3, Room 8W06<br>2531 Jefferson Davis Highway<br>Washington, D.C. 22242-5160       | 1 |
| 5. | FCC Stein<br>Naval Sea Systems Command<br>Code 62ZP, NC3, Room 8W06<br>2531 Jefferson Davis Highway<br>Washington, D.C. 22242-5160           | 1 |
| 6. | CMD A.M. Joseph<br>Port Hueneme Division<br>Naval Surface Warfare Center<br>Code 4A00<br>Port Hueneme, California 93043                      | 1 |
| 7. | Mr. Bill Campbell<br>Port Hueneme Division<br>Naval Surface Warfare Center<br>Code 4W32<br>Port Hueneme, California 93043                    | 1 |

- |     |  |   |
|-----|--|---|
| 8.  | Mr. Henry Seto<br>Port Hueneme Division<br>Naval Surface Warfare Center<br>Code 4W32<br>Port Hueneme, California 93043               | 1 |
| 9.  | Professor Magdi Kamel, Code AS/Ka<br>Naval Postgraduate School<br>Monterey, California 93943-5000                                    | 2 |
| 10. | Professor Martin McCaffrey, Code AS/Mf<br>Naval Postgraduate School<br>Monterey, California 93943-5000                               | 2 |
| 11. | LT Claude David Smith<br>N2, Commander Amphibious Squadron Five<br>FPO AP 96601-5804   | 3 |
| 12. | Mr. W. Bruce Holt<br>American Defense Preparedness Association<br>2101 Wilson Boulevard, Suite 400<br>Arlington, Virginia 22201-3061 | 1 |